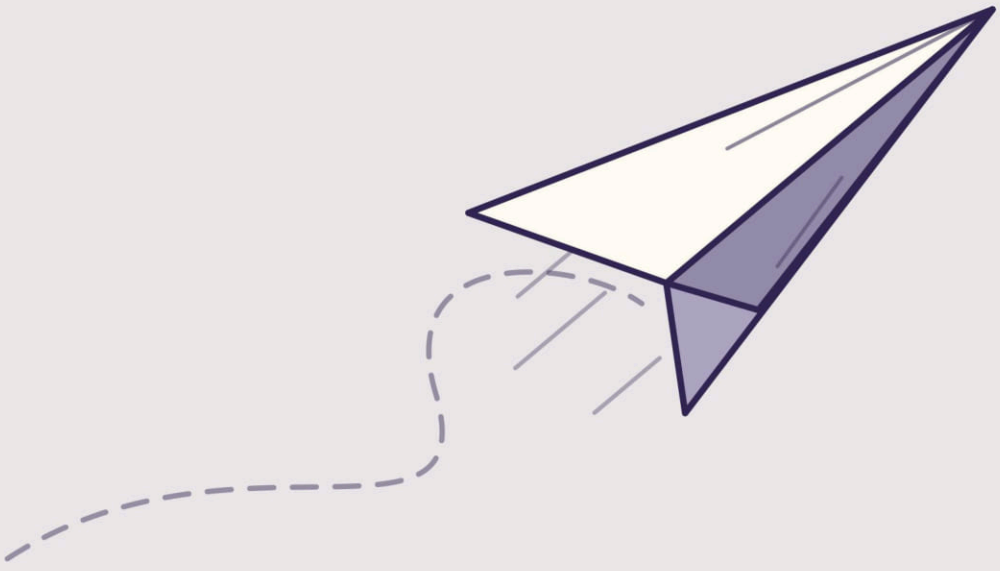
TKA SMK

PENGEMBANGAN PERANGKAT LUNAK DAN GIM

Materi, Strategi, dan Simulasi



Handaka Lumu
by Next Level Study Indonesia

2026/2027 · SMKPPLG

TKA SMK PENGEMBANGAN PERANGKAT LUNAK DAN GIM

Materi, Strategi, dan Simulasi

Handaka Lumu

by Next Level Study Indonesia

The Dancer of the Silent Stage

"Semakin kau mengenalku, semakin banyak yang kamu tidak tahu"



KETERANGAN BUKU

TKA SMK Pengembangan Perangkat Lunak dan Gim: Materi, Strategi, dan Simulasi

Penulis: Handaka Lumu

Penyusun soal: Tim Next Level Study Indonesia

Penerbit: Next Level Study Indonesia

Edisi: Edisi 1 · Gratis untuk pelajar Indonesia · 2026/2027 · Kode buku SMKPPLG

© 2026 Next Level Study Indonesia. Hak cipta dilindungi undang-undang.

GRATIS · TIDAK UNTUK DIPERJUALBELIKAN

Buku ini dibagikan gratis sebagai dedikasi Next Level Study Indonesia untuk bangsa. Boleh diperbanyak dan dibagikan untuk belajar, tetapi tidak untuk diperjualbelikan.

Sumber soal asli: Belum ada soal asli TKA untuk mapel ini; bentuk soal mengikuti soal asli TKA SMK 2025 Projek Kreatif dan Kewirausahaan (Pusat Asesmen Pendidikan, Kemendikdasmen).

Kata Pengantar

Kamu yang sedang bersiap,

Terima kasih sudah membuka buku ini. Hampir semua yang kamu pegang hari ini lahir dari baris-baris kode: aplikasi di ponselmu, gim yang menemani istirahatmu, sampai kasir di warung dekat rumah. Orang-orang yang menulisnya dulu juga pernah bingung kenapa programnya tidak mau jalan, lalu belajar sedikit demi sedikit sampai bisa.

Buku ini lahir dari satu kenyataan. Soal TKA jurusan Pengembangan Perangkat Lunak dan Gim jarang bertanya "apa kepanjangan istilah ini". Soal-soalnya bercerita tentang siswa yang sedang praktik kerja lapangan, studio gim kecil yang mengejar jadwal rilis, atau aplikasi yang tiba-tiba gagal bekerja. Lalu kamu diminta memutuskan tindakan apa yang tepat, apa keluaran program ini, atau di mana letak kesalahannya. Karena itu, yang kamu butuhkan bukan hafalan, melainkan pemahaman yang cukup kuat untuk dipakai di situasi baru.

Isinya mengikuti kisi-kisi resmi Pusmendik dalam lima bab: wawasan dunia kerja, kecakapan kerja dan K3, jaringan komputer, pemrograman terstruktur, dan pemrograman berorientasi objek. Setiap sub bab menjelaskan materinya sampai tuntas, lengkap dengan gambar dan contoh yang diselesaikan langkah demi langkah, baru kemudian strategi dan jebakan. Di akhir setiap bab ada kuis, dan di akhir buku ada simulasi TKA berwaktu. Belum ada soal asli TKA untuk mapel ini, jadi saya tidak menjanjikan angka apa pun. Yang bisa saya janjikan hanyalah penjelasan yang jujur dan latihan yang terarah.

Bacalah berurutan, dan jangan hanya dengan mata. Jalankan kode di komputermu, ubah satu barisnya, lalu lihat apa yang berubah. Bila skor kuismu belum memuaskan, kembalilah ke sub bab yang ditunjuk tabel anjuran. Latihan dan kuis yang sama juga ada di LMS NLS lewat kode QR.

Program yang baik dibangun dari ribuan keputusan kecil yang dikerjakan dengan teliti dan jujur, bahkan saat tidak ada yang memeriksa. Belajarlah dengan cara yang sama. Berjanjilah pada dirimu untuk tidak menyerah ketika kode pertamamu error. Persembahkan kesungguhanmu untuk orang tua yang menunggu kabar baik, untuk guru yang sabar membimbingmu, dan untuk negeri yang kelak membutuhkan tenaga serta pikiranmu. Belajarlah dengan sepenuh jiwa dan raga, karena ilmu yang kamu kumpulkan hari ini adalah baktimu esok hari.

Selamat belajar. Saya percaya kamu bisa.

Oktober 2026

Handaka Lumu

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Panduan Memakai Buku

Buku ini dipakai berurutan: kenali ujiannya (Bagian A), pelajari 5 bab materi (Bagian B), uji kesiapanmu dengan simulasi TKA berwaktu (Bagian C), lalu cocokkan jawaban dan pelajari pembahasannya (Bagian D).

Peta belajar

Langkah	Kamu mengerjakan	Hasil
1	Bagian A: kenali bentuk soal dan peta kisi-kisi	Tahu apa yang diuji
2	Bab demi bab: baca sub bab, kerjakan latihan, lalu Kuis Akhir Bab	Skor kuis per bab
3	Ulangi sub bab yang ditunjuk tabel anjuran di akhir kuis	Titik lemah berkurang
4	Bagian C: 3 simulasi TKA berwaktu	Skor simulasi
5	Bagian D: kunci dan pembahasan semua soal	Paham alasan, bukan hanya jawaban

Tanda di dalam buku

Tanda	Artinya
◎ APA YANG DIUJI	Tujuan belajar sub bab dan indikator resmi yang diuji
◆ KONSEP INTI	Materi lengkap: penjelasan runtut, gambar, dan contoh terselesaikan
▲ STRATEGI DAN JEBAKAN	Langkah menjawab dan pola pengecoh yang sering dipasang
● BEDAH SOAL	Soal dibahas langkah demi langkah
✎ LATIHAN	Soal untuk dikerjakan sendiri; kunci singkat di bawahnya
✔ PERIKSA DIRI	Pertanyaan refleksi dan hal yang perlu diingat
Kotak QR	Membuka latihan dan kuis yang sama di LMS NLS

Kode

SMKPPLG-1.2 = sub bab 1.2; SMKPPLG-KB1 = kuis akhir bab 1; SMKPPLG-FIN1 = simulasi 1. Kode yang sama dipakai di LMS.

Belajar lewat PDF atau LMS

Semua latihan, kunci, dan pembahasan sudah ada di buku ini, jadi kamu bisa belajar hanya dengan PDF atau cetakan. LMS menambahkan latihan dan kuis interaktif, serta menyimpan skormu.

Buka buku ini di LMS NLS

Pindai untuk membuka course buku ini.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Daftar Isi

Kata Pengantar ii
Panduan Memakai Buku iii
Bagian A · Mengenal TKA Pengembangan Perangkat Lunak dan Gim 1
Bagian B · Materi 8
Bab 1 · Wawasan Dunia Kerja Bidang PPLG 8
Sub bab 1.1 · Profesi dan peluang usaha PPLG 9
Sub bab 1.2 · Manajemen proyek dan budaya mutu 23
Kuis Akhir Bab 1 39
Bab 2 · Kecakapan Kerja Dasar, K3, dan Budaya Kerja 51
Sub bab 2.1 · K3LH dan budaya kerja profesional 52
Sub bab 2.2 · Pengelolaan aset fisik dan digital 66
Sub bab 2.3 · Pengelolaan file proyek, source code, dan dokumentasi 79
Kuis Akhir Bab 2 92
Bab 3 · Teknologi Jaringan Komputer untuk PPLG 103
Sub bab 3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi 104
Sub bab 3.2 · Konfigurasi jaringan dasar 118
Sub bab 3.3 · Layer TCP/IP dan fungsi protokol 133
Sub bab 3.4 · Mekanisme komunikasi jaringan 145
Kuis Akhir Bab 3 159
Bab 4 · Pemrograman Terstruktur 171
Sub bab 4.1 · Tipe data dan struktur data 172
Sub bab 4.2 · Struktur kontrol percabangan 186
Sub bab 4.3 · Struktur kontrol perulangan 201
Sub bab 4.4 · Fungsi, prosedur, dan modularisasi 215
Kuis Akhir Bab 4 230
Bab 5 · Pemrograman Berorientasi Objek 242
Sub bab 5.1 · Class, object, dan prinsip dasar OOP 243
Sub bab 5.2 · Enkapsulasi, access modifier, dan getter–setter 259
Sub bab 5.3 · Pewarisan 276
Sub bab 5.4 · Polymorphism 293
Kuis Akhir Bab 5 308
Bagian C · Simulasi TKA 323
Simulasi 1 — Prediksi TKA 2026 324
Simulasi 2 — Prediksi TKA 2026 339
Simulasi 3 — Prediksi TKA 2026 354
Bagian D · Kunci dan Pembahasan 367
Penutup 458
Lampiran 460

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bagian A · Mengenal TKA Pengembangan Perangkat Lunak dan Gim

Apa itu TKA SMK Pengembangan Perangkat Lunak dan Gim

Tes Kemampuan Akademik (TKA) untuk jenjang SMK/MAK memiliki mata pelajaran **pilihan** sesuai bidang keahlian. **Pengembangan Perangkat Lunak dan Gim (PPLG)** adalah salah satunya. Menurut halaman kisi-kisi Pusat Asesmen Pendidikan (Pusmendik) Kemendikdasmen, soal PPLG mengukur tiga tingkat kemampuan: **pengetahuan dan pemahaman** (mengenali, menjelaskan), **penerapan dan implementasi** (mengonfigurasi, menerapkan konsep pada situasi nyata), dan **penalaran** (menganalisis, mengevaluasi, mengklasifikasi). Materinya mengacu pada Capaian Pembelajaran bidang PPLG. Acuan kisi-kisi buku ini: halaman Pusmendik `https://pusmendik.kemendikdasmen.go.id/tka/tka/view/mata-pelajaran-pilihan/sma/pengembangan-perangkat-lunak-dan-gim-smk-mak` (diakses 16 Oktober 2026).

Ada dua hal yang perlu kamu ketahui sejak awal, dan keduanya jujur apa adanya.

> **Jumlah soal dan waktu.** Halaman kisi-kisi Pusmendik tidak mencantumkan jumlah soal dan durasi. Buku ini memakai bentuk yang sama dengan mapel pilihan lain: **25 soal dalam 60 menit**, atau rata-rata sekitar **2,4 menit** untuk setiap soal. Pastikan ketentuan resminya lewat pengumuman Pusmendik dan sekolahmu menjelang ujian.

> **Soal asli.** Untuk mapel PPLG **belum ada soal asli TKA** yang dapat dimuat. Satu-satunya soal asli TKA SMK 2025 yang tersedia adalah mapel Projek Kreatif dan Kewirausahaan. Karena itu, bentuk dan kesulitan soal di buku ini mengikuti soal asli tersebut dan enam contoh soal resmi PPLG di halaman Pusmendik, sedangkan **isinya** mengikuti kisi-kisi PPLG. Buku ini tidak memuat angka "persentase benar nasional" untuk PPLG karena datanya memang belum ada, dan angka seperti itu tidak boleh dikarang.

Apa yang diuji: lima elemen

Kisi-kisi resmi membagi muatan TKA PPLG menjadi **lima elemen**, dan buku ini memakainya sebagai lima bab. Kolom terakhir memuat sebaran soal pada paket latihan buku ini (setiap paket 25 soal), supaya kamu tahu bab mana yang paling banyak keluar.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Elemen kisi-kisi	Isi menurut kisi-kisi	Bab di buku ini	Soal per 25
Wawasan dunia kerja bidang PPLG	jenis profesi dan tugas utamanya; kewirausahaan teknologi; tahapan manajemen proyek, proses bisnis, dan budaya mutu	Bab 1	4
Kecakapan kerja dasar, K3, dan budaya kerja	K3LH; etika dan budaya kerja profesional; pengelolaan aset fisik dan digital (perangkat, berkas proyek, kode sumber, media, basis data, dokumentasi)	Bab 2	4–5
Teknologi jaringan komputer	pengelolaan berkas dan direktori serta konfigurasi lingkungan pengembangan pada sistem operasi; konfigurasi jaringan dasar; layer TCP/IP dan mekanisme komunikasi	Bab 3	5
Pemrograman terstruktur	tipe dan struktur data; struktur kontrol percabangan dan perulangan; fungsi, prosedur, dan modularisasi	Bab 4	6–7
Pemrograman berorientasi objek	class, object, dan prinsip OOP; enkapsulasi dan access modifier; pewarisan; polymorphism	Bab 5	5

Dari tabel itu terlihat bahwa **Bab 4 dan Bab 5 bersama-sama hampir separuh soal**. Pemrograman adalah inti jurusan ini, dan di kedua bab itulah kamu paling sering diminta menelusuri potongan kode. Namun, Bab 1 sampai Bab 3 tidak boleh dilewati: soalnya lebih pendek, jawabannya sering ada di akal sehat kerja yang benar, dan di sanalah banyak poin yang bisa kamu amankan dengan cepat.

Kisi-kisi memuat **empat belas sub-elemen**, sedangkan Peta Topik Bank Soal NLS memecahnya menjadi **24 subtopik**. Agar setiap sub bab cukup utuh untuk dipelajari, subtopik yang bertetangga digabung menjadi **17 sub bab**. Tabel silang di Lampiran menunjukkan kaitan sub bab, subtopik, dan soal.

Tiga tingkat kemampuan

Tingkat	Yang diminta	Contoh bentuknya
Pengetahuan dan pemahaman	mengenali, menjelaskan	menentukan profesi dari uraian tugasnya; menjelaskan fungsi sebuah protokol
Penerapan dan implementasi	mengonfigurasi, menerapkan	memilih alamat IP yang tepat untuk sebuah komputer; menentukan perintah untuk memperbaiki hak akses berkas
Penalaran	menganalisis, mengevaluasi, mengklasifikasi	menelusuri keluaran sebuah program; menilai apakah pembagian modul sudah efektif; mengklasifikasi mana pernyataan yang benar tentang enkapsulasi

Pada paket latihan buku ini, soal sulit paling banyak ada di **Bab 4 dan Bab 5**, karena pada kedua bab itu kisi-kisi memang meminta penalaran. Soal sulit di sini jarang berarti "konsepnya sangat

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

tinggi". Biasanya soal itu sulit karena kamu harus **teliti**: satu tanda, satu angka batas, atau satu baris kode yang terlewat sudah mengubah jawabannya.

Konteks soal: kasus dunia kerja

Hampir setiap soal berbentuk **kasus**. Tokohnya siswa PKL, programmer di sebuah studio gim kecil, tim pembuat aplikasi kasir untuk warung, atau admin lab sekolah. Contoh resmi Pusmendik memperlihatkan hal yang sama: seorang developer menghadapi pesan galat di komputer, sebuah tim menggambar topologi jaringan aplikasi, sebuah program menghitung jumlah siswa yang lulus.

Konteks	Contoh yang akan kamu temui
Dunia kerja dan usaha	uraian tugas yang harus dicocokkan dengan profesi; rincian biaya usaha jasa aplikasi; jadwal kegiatan proyek
Keselamatan, etika, dan pengelolaan aset	UPS yang berasap di lab; kesalahan kode yang ditemukan setelah rilis; tabel peminjaman perangkat uji; berkas rahasia yang ikut terunggah
Jaringan dan sistem operasi	gambar topologi lab; pesan galat saat menjalankan server lokal; alamat IP dan subnet; alur komunikasi aplikasi dengan server
Pemrograman	potongan kode Python pendek (5–15 baris); alur logika sebuah gim atau aplikasi kasir; rancangan class untuk karakter gim

Cerita pembukanya bisa cukup panjang. Sebagian besar hanya latar, tetapi **petunjuk penting sering diselipkan di dalamnya**: angka batas, nama berkas, urutan langkah. **Baca kalimat pertanyaannya dulu**, lalu kembali ke cerita untuk mencari petunjuk yang dibutuhkan.

Bentuk soal dan cara menjawabnya

Kisi-kisi resmi menyebut tiga bentuk soal.

Bentuk	Cara menjawab
Pilihan ganda (PG)	pilih **satu** jawaban dari lima pilihan A–E
PG kompleks kategori (Benar/Salah)	tentukan **setiap** pernyataan Benar atau Salah
PG kompleks pilih lebih dari satu (MCMA)	pilih **semua** pilihan yang benar; jumlahnya tidak disebutkan

Tiga contoh berikut sengaja dibuat sendiri dan bukan bagian dari paket latihan mana pun. Tujuannya memperlihatkan bentuk soal dan cara berpikir menghadapinya.

Contoh 1 — pilihan ganda

> Dito menjalankan server uji aplikasi di sebuah komputer lab dengan alamat 192.168.10.5 dan subnet mask 255.255.255.0. Laptop Dito memiliki alamat 192.168.11.20 dengan subnet mask yang sama. Kabel dan Wi-Fi laptop menyala normal, tetapi laptop tidak dapat membuka server itu. Penyebab yang paling mungkin adalah ....
>
> A. alamat IP server dan laptop berada pada jaringan yang berbeda

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

B. subnet mask harus diganti menjadi 255.255.0.0 pada server saja
C. laptop tidak boleh menggunakan alamat IP dengan angka 20
D. kabel yang dipakai laptop pasti rusak karena Wi-Fi menyala
E. server harus dimatikan lebih dulu sebelum dapat diakses

Cara berpikir. Dengan mask 255.255.255.0 (ditulis /24), tiga angka pertama alamat menunjukkan jaringan. Server berada di 192.168.**10**.x, laptop di 192.168.**11**.x, jadi keduanya berada di jaringan yang berbeda dan tidak dapat saling menjangkau tanpa router yang mengatur jalurnya. Jawaban: **A**. Pilihan B mengubah mask di satu sisi saja, sehingga kedua perangkat tidak sepakat tentang batas jaringan. Pilihan C dan E tidak ada dasarnya. Pilihan D menyimpulkan terlalu jauh dari satu petunjuk.

Contoh 2 — Benar/Salah dengan kode

Perhatikan program Python berikut.

```
stok = {"pensil": 12, "buku": 0, "penghapus": 5}
for barang in stok:
    if stok[barang] == 0:
        print(barang, "habis")
```

Tentukan setiap pernyataan Benar atau Salah.

1. Program mencetak tepat satu baris.

2. Ekspresi `stok[barang]` mengambil jumlah barang berdasarkan namanya.

3. Jika stok penghapus diubah menjadi 0, tulisan "penghapus habis" dicetak sebelum "buku habis".

Cara berpikir. Telusuri satu per satu seperti komputer. Perulangan `for` pada kamus mengambil **kunci** satu per satu menurut urutan penyisipannya: pensil, buku, penghapus. Hanya "buku" yang bernilai 0, jadi hanya satu baris yang dicetak: pernyataan 1 **Benar**. Pernyataan 2 **Benar**: nama barang adalah kunci dan jumlahnya adalah nilai. Pada pernyataan 3, urutan kuncinya tetap pensil, buku, penghapus, jadi "buku habis" dicetak **lebih dulu**, lalu "penghapus habis". Pernyataan 3 **Salah**. Perhatikan bahwa pernyataan 3 benar-benar memakai istilah yang akrab, tetapi urutannya terbalik. Telusuri, jangan menebak.

Contoh 3 — pilih lebih dari satu (MCMA)

Pada Kamis sore, UPS yang menyuplai beberapa komputer di lab RPL tiba-tiba mengeluarkan asap tipis dan bau hangus. Belum tampak api. Tindakan yang tepat adalah (pilih **semua** yang benar) ....

A. memutus aliran listrik dari saklar utama bila dapat dilakukan dengan aman
B. menyiram UPS dengan seember air agar cepat dingin
C. memakai alat pemadam api ringan jenis serbuk kering atau CO_2 bila api muncul
D. menjauhkan orang dari area itu dan melapor kepada penanggung jawab lab

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

> E. merekam video kejadian dari jarak dekat lebih dulu sebelum menyentuh apa pun

Cara berpikir. Nilai pilihan satu per satu. A tepat karena sumber bahaya harus dihentikan lebih dulu. B salah karena air pada peralatan yang masih bertegangan berbahaya. C tepat karena alat pemadam jenis itu aman untuk api yang melibatkan listrik. D tepat karena keselamatan orang lebih dahulu. E salah karena menunda tindakan dan menempatkan orang terlalu dekat dengan sumber bahaya. Jawaban: **A, C, dan D**. Jangan menebak jumlahnya, dan jangan berhenti pada jawaban pertama yang kamu temukan.

Ciri khas soal PPLG

1. Kode Python yang pendek

Potongan kode di soal memakai gaya **Python**: pendek, biasanya 5 sampai 15 baris. Kamu tidak perlu menulis program. Yang diminta adalah **menelusuri** (apa keluarannya), **menganalisis** (apa yang salah atau tidak efisien), atau **mengklasifikasi** (pernyataan mana yang benar tentang konsep yang ditunjukkan kode). Soal kode tidak menuntut pustaka rumit. Soal hanya memakai konsep dasar yang dibahas di Bab 4 dan Bab 5.

2. Gambar yang berisi informasi

Beberapa soal memakai gambar topologi jaringan, diagram alir, atau diagram class. Gambarnya **memuat informasi yang dibutuhkan untuk menjawab**. Alamat IP, arah panah, dan nama class yang tercantum di gambar bukan hiasan. Baca semua label sebelum membaca pilihan jawaban.

3. Tabel sederhana

Rincian biaya, jadwal kegiatan, catatan peminjaman perangkat, atau daftar aset. Hitungannya hanya aritmetika dasar: menjumlah, mengalikan, dan menghitung persentase. Yang menjatuhkan siswa biasanya bukan hitungannya, melainkan **salah memilih baris** yang tepat.

4. Pilihan yang berupa keputusan

Pada soal kasus, pilihan jawabannya sering berupa tindakan: "apa yang paling tepat dilakukan tim?". Beberapa tindakan **benar tetapi bukan yang paling tepat**, atau benar tetapi tidak menyasar akar masalah. Pilih yang menyelesaikan penyebabnya, bukan hanya yang meredakan gejalanya.

Cara menelusuri kode: tabel jejak

Keterampilan terpenting di Bab 4 dan Bab 5 adalah **menelusuri kode dengan tabel jejak**: sebuah tabel kecil yang mencatat nilai setiap variabel setiap kali berubah. Dengan cara ini kamu tidak perlu "merasakan" jawabannya.

Perhatikan program berikut.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
total = 0
for angka in range(1, 6):
    if angka % 2 == 0:
        total = total + angka
print(total)
```

angka	angka % 2 == 0?	total setelah baris itu
(awal)	–	0
1	tidak (sisa 1)	0
2	ya	2
3	tidak	2
4	ya	6
5	tidak	6

Keluarannya adalah **6**, yaitu jumlah bilangan genap 2 dan 4. Perhatikan dua hal yang sering terlewat. Pertama, `range(1, 6)` menghasilkan 1 sampai **5**, angka 6 tidak ikut. Kedua, `print(total)` ada di luar perulangan, sehingga hanya dijalankan **sekali** di akhir. Bab 4.3 membahas ini lebih dalam.

Keterampilan lintas bab

Membaca topologi dan alamat IP

Mulailah dari **gateway** dan **subnet mask** yang disebut di soal. Dua perangkat berada pada jaringan yang sama bila bagian jaringan alamat mereka sama menurut mask itu. Periksa juga nilai tiap angka alamat: setiap bagian alamat IPv4 hanya berkisar 0 sampai 255, sehingga angka seperti 256 atau 300 menandakan alamat tidak valid. Bab 3.2 mengajarkannya dengan teliti.

Membaca pesan galat

Pesan galat adalah petunjuk, bukan musuh. "Permission denied" berarti **izin** yang kurang, "command not found" berarti perintah tidak ditemukan di tempat yang dicari sistem, dan "Address already in use" berarti nomor port sedang dipakai program lain. Setiap pesan menunjuk ke satu akar masalah. Pilih tindakan yang menyasar akar masalah itu.

Memisahkan "terlihat benar" dari "benar menurut kode"

Banyak pengecoh soal kode adalah keluaran yang **terdengar masuk akal**: angka yang kamu harapkan dari program itu, bukan angka yang benar-benar dihasilkannya. Komputer hanya mengerjakan apa yang tertulis. Jangan memakai maksud pembuat program sebagai dasar jawaban; pakai bunyi kodenya.

Strategi 60 menit

1. **Putaran pertama (±30 menit):** kerjakan soal yang jawabannya langsung terlihat: profesi, K3, aset, konsep jaringan, dan soal kode yang pendek. Tandai soal kode panjang dan soal MCMA yang masih ragu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. **Putaran kedua (±25 menit):** kerjakan soal yang ditandai. Untuk soal kode, buat **tabel jejak** di kertas coretan. Untuk soal kasus, tulis dahulu apa yang sebenarnya ditanyakan dengan satu kalimat pendek.
3. **Lima menit terakhir:** periksa ulang soal Benar/Salah dan MCMA baris demi baris, lalu pastikan tidak ada soal yang kosong.
4. **Waspadai kata mutlak yang tidak didukung soal**, seperti "selalu", "tidak pernah", atau "pasti". Namun, jangan memilih atau menolak jawaban hanya karena kata-katanya. Selalu cocokkan dengan informasi di soal.

Cara memakai buku ini untuk PPLG

- **Ikuti urutan bab.** Bab 3 membantu Bab 2 dan Bab 4; Bab 5 memakai semua yang kamu pelajari di Bab 4, terutama fungsi dan struktur data.
- **Jalankan kodenya.** Instal Python 3 di komputermu (atau pakai editor daring), ketik ulang kode di setiap contoh, lalu ubah satu hal dan lihat akibatnya. Hasil yang kamu amati sendiri lebih awet daripada hasil yang hanya dibaca.
- **Prioritaskan sub bab bertanda ★.** Sub bab ini punya bobot soal besar atau soal sulit yang banyak.
- **Kerjakan kuis akhir bab dalam 60 menit**, sama dengan waktu ujian, lalu ikuti tabel anjuran di bawahnya.
- **Simpan Simulasi TKA untuk akhir.** Tiga simulasi berisi soal yang belum pernah kamu temui di sub bab mana pun dan menjadi ukuran kesiapanmu yang sebenarnya.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bagian B · Materi

Bagian ini memuat 5 bab. Tiap bab diakhiri kuis; kerjakan berurutan.

Bab 1 · Wawasan Dunia Kerja Bidang PPLG

SMKPPLG-B1

Setelah mempelajari bab ini, kamu dapat:

- mencocokkan tugas dalam tim pengembang perangkat lunak dan gim dengan profesi yang tepat
- menjelaskan prinsip kewirausahaan teknologi dan menghitung harga jual serta titik impas dari rincian biaya
- menyusun tahapan dan jadwal proyek, termasuk jalur kritisnya, dan memilih model pengembangan yang sesuai
- menerapkan prinsip budaya mutu, seperti kriteria penerimaan, code review, dan pedoman penulisan kode

Peta bab

Sub bab	Judul
1.1	Profesi dan peluang usaha PPLG
1.2	Manajemen proyek dan budaya mutu



Bab 1 di LMS

Buka seluruh latihan bab ini di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-B1

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 1.1 · Profesi dan peluang usaha PPLG

SMKPPLG-1.1

◎ APA YANG DIUJI

Kisi-kisi resmi Pusmendik meminta kamu **mengidentifikasi jenis profesi bidang PPLG beserta tugas utamanya** dan **menjelaskan prinsip dasar kewirausahaan teknologi (technopreneurship)**, termasuk peluang usaha di bidang perangkat lunak dan gim. Levelnya pengetahuan dan pemahaman: kamu harus mengenali, menjelaskan, dan menghubungkan, bukan menghitung rumit.

Dalam soal, bagian ini hampir selalu berbentuk kasus pendek. Sebuah studio butuh "seseorang yang mengerjakan ini dan itu", dan kamu menentukan profesinya. Atau dua lulusan SMK ingin membuka usaha, dan kamu menimbang langkah mana yang masuk akal atau menghitung apakah usahanya sudah untung.

Setelah mempelajari sub bab ini, kamu dapat:

- menyebutkan profesi utama di industri perangkat lunak dan gim beserta tugas dan hasil kerjanya;
- membedakan profesi yang tugasnya mirip lewat kata kerja dan objek kerjanya;
- menjelaskan ciri wirausaha teknologi, cara menemukan peluang, dan cara menguji gagasan dengan murah;
- menghitung biaya, harga jual, laba, dan titik impas sebuah usaha jasa atau produk digital sederhana.

◆ KONSEP INTI

Hampir semua yang kamu pakai setiap hari adalah perangkat lunak: aplikasi ojek daring, gim di ponsel, kasir di minimarket, bahkan jam di dinding yang tersambung internet. Di balik setiap produk itu ada sekelompok orang dengan tugas berbeda-beda. Jurusan PPLG menyiapkan kamu untuk masuk ke salah satunya, atau membuat usaha sendiri. Sub bab ini memperkenalkan dua hal: **siapa mengerjakan apa**, dan **bagaimana sebuah gagasan menjadi usaha yang bisa hidup**.

1. Produk dan tempat bekerja di bidang PPLG

Produk bidang ini secara garis besar ada empat macam.

- **Aplikasi web**: dibuka lewat peramban, seperti situs belanja daring atau sistem absensi sekolah.
- **Aplikasi seluler (mobile)**: dipasang di ponsel, seperti aplikasi dompet digital.
- **Aplikasi desktop**: dipasang di komputer, seperti program kasir toko atau aplikasi pengolah gambar.

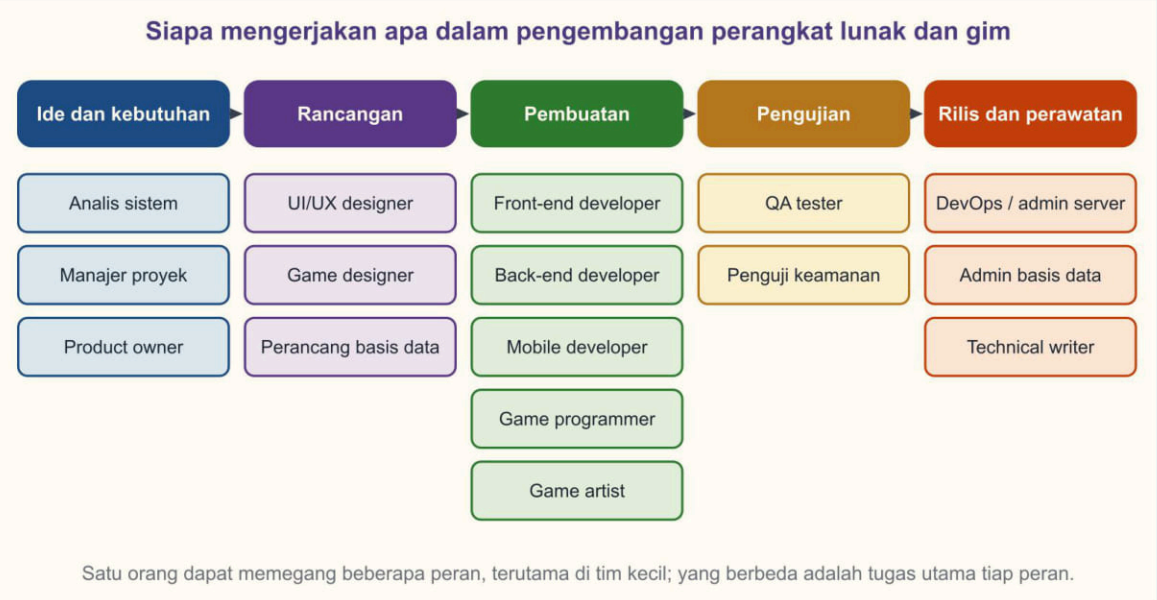
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **Gim (game)**: perangkat lunak hiburan atau edukasi, baik di ponsel, komputer, maupun konsol.

Orang bidang PPLG bekerja di berbagai tempat. Ada **perusahaan perangkat lunak** yang membuat produk atau menerima pesanan, **studio gim** yang memproduksi gim, **startup** yang membangun satu produk digital dari nol, dan **bagian teknologi informasi** di bank, rumah sakit, atau sekolah. Ada juga **pekerja lepas (freelancer)** yang menerima proyek dari banyak klien, dan **wirausahawan** yang membuka usaha jasa atau produk digital sendiri.

2. Siapa mengerjakan apa: profesi dan tugas utamanya

Perangkat lunak yang besar tidak dibuat satu orang. Pekerjaan dibagi dalam beberapa tahap, dan tiap tahap punya profesi yang tugas utamanya berbeda. Gambar 1.1 memperlihatkan petanya, dan tabel berikut menjelaskan tiap profesi.



Gambar 1.1 Peta profesi dalam lima tahap pengembangan perangkat lunak dan gim. Sumber: digambar oleh Tim Next Level Study Indonesia.

Profesi	Tugas utama	Hasil kerja
Analis sistem	menggali kebutuhan klien dan pengguna, lalu menuliskannya sebagai kebutuhan sistem	dokumen kebutuhan
UI/UX designer	merancang tampilan (UI) dan pengalaman pemakaian (UX), mewawancarai dan mengamati calon pengguna	sketsa layar, prototipe yang dapat diklik
Front-end developer	menulis kode tampilan yang dilihat dan disentuh pengguna di peramban	halaman web yang berfungsi
Back-end developer	menulis logika di sisi server, mengolah data, dan menyediakan layanan untuk aplikasi	server dan layanan data
Mobile developer	membuat aplikasi untuk ponsel Android atau iOS	aplikasi ponsel
Perancang dan administrator basis data	merancang tabel, menjaga cadangan, kecepatan akses, dan hak akses data	basis data yang aman dan cepat

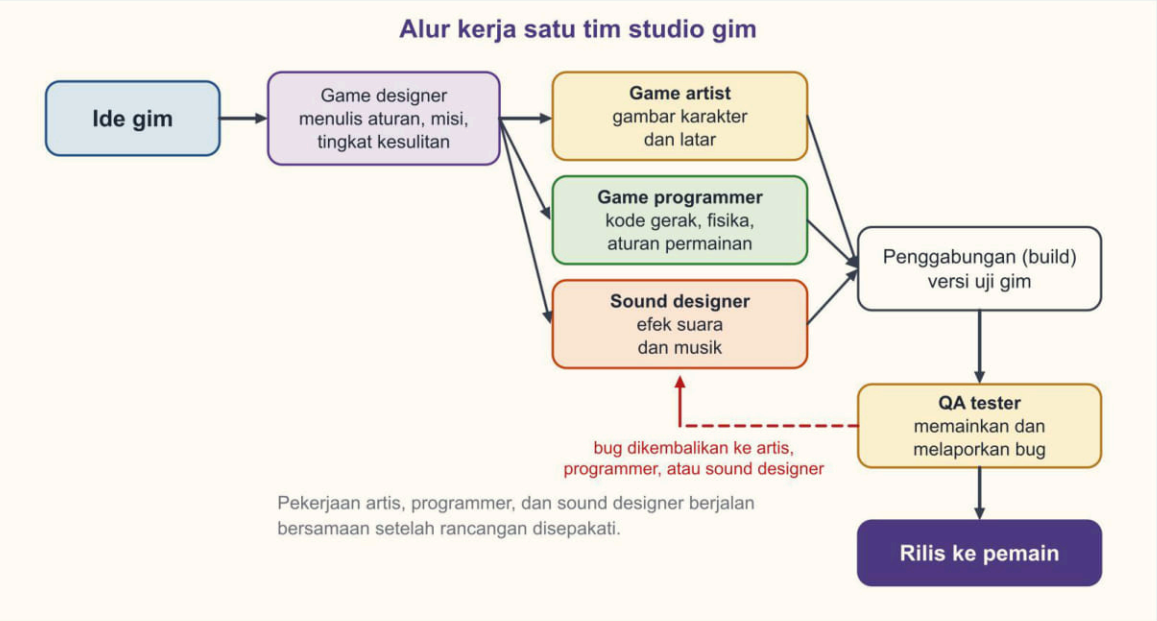
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

QA tester (quality assurance)	menguji perangkat lunak, melaporkan kesalahan (bug) dengan langkah memunculkannya, lalu menguji ulang	laporan bug, hasil uji
DevOps / administrator sistem	menyiapkan server, mengemas dan merilis versi baru, memantau agar layanan tetap hidup	server yang berjalan, jalur rilis
Manajer proyek	mengatur jadwal, tim, dan biaya agar proyek selesai sesuai rencana	rencana dan laporan proyek
Technical writer	menyusun buku panduan, dokumentasi, dan petunjuk pemakaian	dokumentasi

Dunia gim punya profesi tambahan.

Profesi gim	Tugas utama
Game designer	merancang aturan main, misi, dan alur permainan; **level designer** merancang susunan rintangan dan tingkat kesulitan tiap level
Game programmer	menulis kode gerak karakter, fisika, kecerdasan musuh, dan aturan permainan
Game artist / animator	menggambar karakter, latar, dan efek, lalu menganimasikannya
Sound designer	membuat efek suara dan musik latar
QA tester gim	memainkan versi uji gim berulang-ulang untuk menemukan bug

Alur kerja tim gim bukan garis lurus. Gambar 1.2 menunjukkan bahwa setelah rancangan disepakati, artis, programmer, dan sound designer bekerja **bersamaan**, lalu hasilnya digabung menjadi versi uji. QA tester memainkan versi itu dan melaporkan bug, dan bug dikembalikan kepada yang membuatnya sampai versi layak dirilis.



Gambar 1.2 Alur kerja tim studio gim dari ide sampai rilis. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Cara membedakan profesi yang mirip

Banyak profesi tampak berdekatan, dan soal sering memanfaatkannya. Pegangan yang aman: tentukan **kata kerja utama** dan **objek kerja** pada uraian tugas.

- **Merancang** tampilan atau alur pakai → UI/UX designer. **Merancang** aturan dan misi → game designer. **Merancang** tabel data → perancang basis data.
- **Menulis kode** tampilan → front-end developer. **Menulis kode** server dan data → back-end developer. **Menulis kode** gerak karakter dan aturan gim → game programmer.
- **Menguji dan melaporkan bug** → QA tester. Memperbaiki bug adalah tugas pembuat kodenya, bukan penguji.
- **Menyiapkan server, merilis, memantau** → DevOps atau administrator sistem. **Menjaga cadangan dan hak akses data** → administrator basis data.
- **Menggali kebutuhan dari klien** → analis sistem. **Mengatur jadwal dan biaya** → manajer proyek.

Satu orang di tim kecil sering memegang beberapa peran sekaligus, misalnya *full-stack developer* yang mengerjakan front-end sekaligus back-end. Namun, soal selalu menanyakan **tugas utama** dari uraian yang diberikan, bukan siapa yang kebetulan bisa mengerjakan semuanya.

4. Kewirausahaan teknologi (technopreneurship)

Kewirausahaan adalah kemampuan melihat peluang, mengambil risiko yang sudah dihitung, lalu menciptakan nilai lewat usaha. **Technopreneurship** adalah kewirausahaan yang memakai teknologi, misalnya membuat aplikasi, gim, atau jasa pembuatan situs. Wirausahawan teknologi umumnya memiliki ciri berikut:

- **peka terhadap masalah**: memperhatikan hal yang merepotkan orang lain, bukan hanya hal yang menarik bagi dirinya;
- **inovatif**: menawarkan cara baru yang lebih mudah, cepat, atau murah;
- **berani menanggung risiko yang terukur**: mencoba kecil-kecilan dulu, bukan menghabiskan seluruh modal sekaligus;
- **gigih dan mau belajar** dari kegagalan;
- **bertanggung jawab** terhadap pelanggan, karyawan, dan aturan (termasuk hak cipta).

Peluang usaha sering datang dari masalah nyata yang belum punya jalan keluar yang layak. Pemilik warung mencatat utang pelanggan di buku tulis dan sering lupa; pengelola bank sampah di sebuah RW kesulitan menjumlah tabungan warga; pemilik bengkel kecil tidak punya cara mengatur antrean. Masing-masing adalah peluang untuk aplikasi sederhana.

5. Dari gagasan ke usaha: validasi dengan MVP

Kesalahan paling mahal bagi pemula adalah **membangun produk lengkap lebih dulu**, baru berharap ada yang mau memakainya. Cara yang lebih hemat:

1. **Temukan masalah** dan pastikan masalah itu benar-benar dialami orang, misalnya dengan **mewawancarai** beberapa calon pengguna.
2. **Rancang solusi paling sederhana** yang menyelesaikan masalah utamanya.

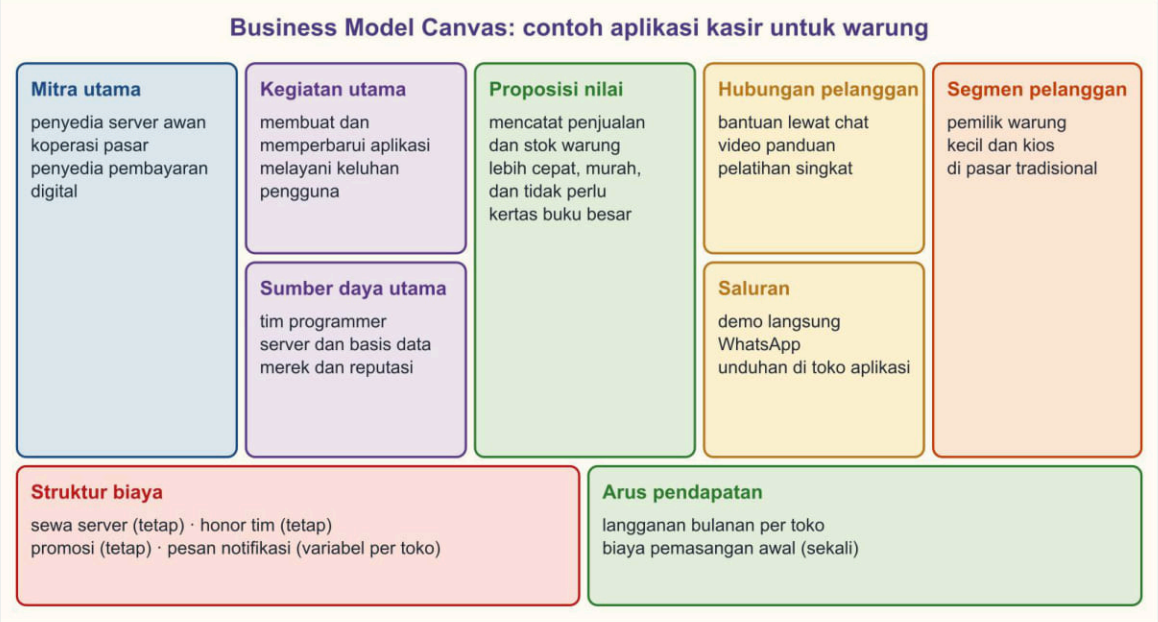
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. **Buat versi awal yang minimal, disebut MVP (minimum viable product)**: hanya fitur inti, cukup untuk dicoba.
4. **Uji pada sedikit pengguna** dan catat tanggapannya, termasuk kesediaan mereka membayar.
5. **Perbaiki atau ubah arah** (disebut *pivot*) berdasarkan data, lalu kembangkan fitur berikutnya.

Dengan cara ini modal kecil tetap terlindungi: bila gagasan tidak diminati, kerugiannya hanya sebuah MVP, bukan produk lengkap.

6. Model bisnis dan cara usaha menghasilkan uang

Agar usaha jelas arahnya, para wirausahawan sering memetakan **model bisnis** dengan *Business Model Canvas* (BMC). BMC terdiri atas sembilan blok, yang dapat dibaca dari "siapa pelanggannya" sampai "dari mana uangnya". Gambar 1.3 memperlihatkan contoh untuk aplikasi kasir warung.



Gambar 1.3 Sembilan blok Business Model Canvas dengan contoh aplikasi kasir untuk warung. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Segmen pelanggan**: siapa yang dilayani (pemilik warung kecil).
- **Proposisi nilai**: masalah apa yang diselesaikan dan mengapa lebih baik dari cara lama.
- **Saluran** dan **hubungan pelanggan**: cara menjangkau dan menjaga hubungan dengan pelanggan.
- **Arus pendapatan**: dari mana uang masuk.
- **Sumber daya utama**, **kegiatan utama**, dan **mitra utama**: apa, siapa, dan kegiatan apa yang dibutuhkan.
- **Struktur biaya**: apa saja yang harus dibayar.

Produk digital punya beberapa model pendapatan yang lazim:

Model	Cara kerjanya	Contoh

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Jual putus / lisensi	pelanggan membayar sekali untuk memakai	aplikasi kasir yang dijual per lisensi
Langganan	pelanggan membayar tiap bulan atau tahun	aplikasi pembukuan untuk UMKM
Freemium	fitur dasar gratis, fitur lanjutan berbayar	gim dengan hadiah tambahan berbayar
Pembelian dalam aplikasi	item atau fitur tambahan dibeli di dalam aplikasi	kostum karakter di gim
Iklan	pengguna gratis, pengiklan yang membayar	gim kasual berisi iklan
Jasa pembuatan	klien membayar proyek pembuatan	situs untuk UMKM atau sekolah

7. Hitungan dasar usaha: biaya, harga jual, dan titik impas

Soal kewirausahaan memakai hitungan yang sederhana, tetapi harus teliti. Ada tiga istilah yang harus jelas.

- **Biaya tetap**: jumlahnya sama berapa pun yang terjual, misalnya sewa ruang, internet, honor bulanan, dan sewa server.
- **Biaya variabel**: bertambah bersama jumlah yang terjual, misalnya biaya lisensi per pesanan atau biaya pengiriman pesan per pelanggan.
- **Biaya total** = biaya tetap + (biaya variabel per unit × jumlah unit).

Dua hitungan yang paling sering muncul:

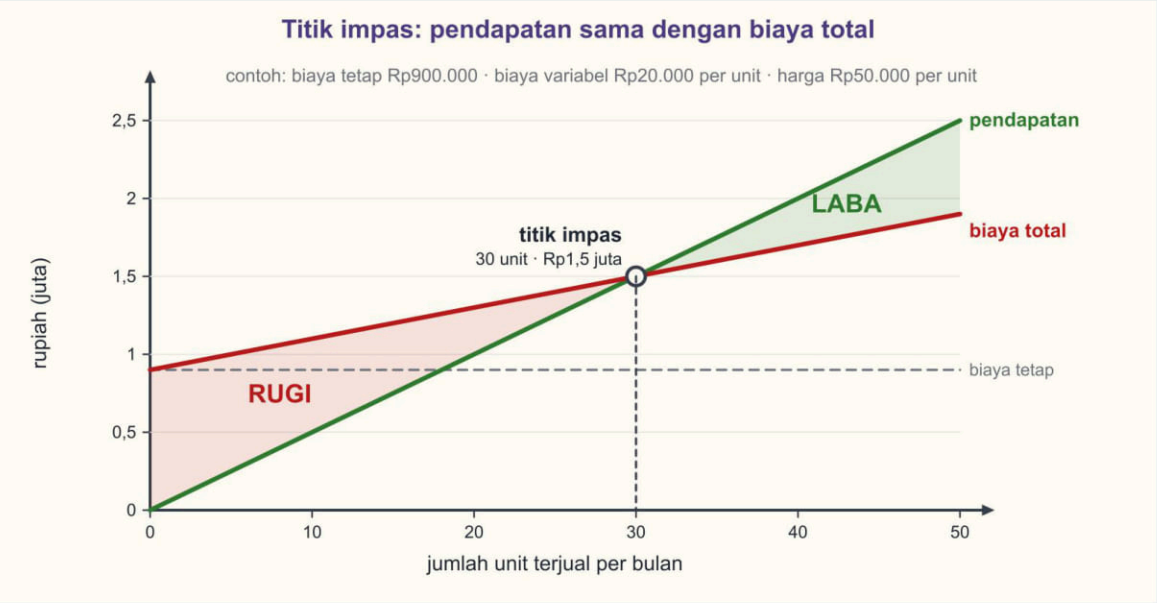
Harga jual dari biaya dan persentase laba. Bila biaya pembuatan satu unit (biaya pokok) adalah B dan usaha menginginkan laba p %, maka harga jual = B + (p % × B).

Titik impas (*break-even point*) adalah jumlah penjualan ketika pendapatan **tepat sama** dengan biaya total, sehingga usaha tidak rugi dan belum untung.

> Titik impas (unit) = biaya tetap ÷ (harga jual per unit − biaya variabel per unit)

Selisih "harga jual − biaya variabel" disebut **sumbangan per unit**: bagian tiap penjualan yang tersisa untuk menutup biaya tetap. Di bawah titik impas usaha rugi, di atasnya untung, dan laba = sumbangan per unit × jumlah − biaya tetap. Gambar 1.4 memperlihatkan hubungan itu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 1.4 Garis pendapatan memotong garis biaya total pada titik impas. Contoh: biaya tetap Rp900.000, biaya variabel Rp20.000, harga jual Rp50.000 per unit, sehingga titik impas 900.000 ÷ 30.000 = 30 unit. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perhatikan dua hal pada gambar. Garis biaya total **tidak mulai dari nol**, tetapi dari biaya tetap, karena biaya itu sudah keluar sebelum ada satu pun penjualan. Dan bila biaya variabel naik sementara harga jual tetap, garis biaya menjadi lebih curam, sumbangan per unit mengecil, dan titik impas bergeser ke kanan.

8. Etika dan risiko

Usaha teknologi memikul tanggung jawab. Pakai hanya aset, perangkat lunak, dan gambar yang lisensinya jelas; aplikasi bajakan dan musik tanpa izin memang menekan biaya, tetapi melanggar hak cipta dan bisa berakhir pada tuntutan. Jaga data pengguna agar tidak bocor, dan jujur tentang kemampuan produk. Risiko usaha digital juga datang dari pasar (produk tidak diminati), teknis (sistem gagal bekerja), dan keuangan (modal habis sebelum pelanggan cukup), sehingga uji kecil dan hitungan biaya yang jujur sangat penting.

CONTOH TERSELESAIKAN

Contoh 1 — Mencocokkan lima uraian tugas dengan profesi

Pada 2027, sebuah klinik hewan di Malang membuat aplikasi jadwal praktik dan rekam medis hewan. Lima orang dalam proyek itu memiliki uraian tugas berikut. Tentukan profesi tiap orang dari pilihan: UI/UX designer, QA tester, back-end developer, analis sistem, DevOps. *(Tokoh dan lembaga fiktif.)*

1. Bu Sekar mewawancarai dokter hewan dan petugas pendaftaran, lalu menuliskan apa saja yang harus bisa dilakukan aplikasi.
2. Dimas menggambar tampilan formulir pendaftaran pasien dan mengamati pegawai klinik mencobanya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Rafi menulis kode yang menyimpan rekam medis ke basis data dan menyediakan datanya bagi aplikasi.
4. Nadia menjalankan skenario pendaftaran berulang kali, mencatat tombol yang tidak berfungsi beserta langkah munculnya, lalu mengujinya lagi setelah diperbaiki.
5. Andre menyiapkan server di awan, mengemas versi baru, dan memastikan layanan tetap hidup.

Langkah: tentukan kata kerja dan objek kerja tiap orang.

1. "Mewawancarai" dan "menuliskan apa yang harus bisa dilakukan aplikasi" → menggali kebutuhan → **analis sistem**.
2. "Menggambar tampilan" dan "mengamati pengguna mencoba" → merancang tampilan dan pemakaian → **UI/UX designer**.
3. "Menulis kode yang menyimpan dan menyediakan data" → logika dan data di sisi server → **back-end developer**.
4. "Menjalankan skenario, mencatat bug, menguji ulang" → **QA tester**.
5. "Menyiapkan server, mengemas versi, menjaga layanan" → **DevOps**.

Hasil: 1 analis sistem, 2 UI/UX designer, 3 back-end developer, 4 QA tester, 5 DevOps. Perhatikan Nadia hanya menguji dan melapor; memperbaiki bug adalah tugas Rafi.

Contoh 2 — Harga jual dari rincian biaya dan laba 25 %

Pada 2027, Studio Cahaya Desa menerima pesanan situs profil untuk sebuah koperasi. Biaya yang dikeluarkan untuk satu situs: honor desainer Rp600.000, honor programmer Rp1.400.000, sewa domain dan hosting setahun Rp400.000. Studio ingin laba 25 % dari biaya. Berapa harga jual situs itu, dan berapa laba studio? *(Data fiktif.)*

1. Jumlahkan biaya pokok: 600.000 + 1.400.000 + 400.000 = Rp2.400.000.
2. Hitung laba 25 %: 25 % × 2.400.000 = Rp600.000.
3. Harga jual = biaya pokok + laba = 2.400.000 + 600.000 = Rp3.000.000.
4. Cek: laba terhadap biaya = 600.000 ÷ 2.400.000 = 0,25 = 25 % ✓.

Jawaban: harga jual Rp3.000.000, laba Rp600.000. Hati-hati: laba 25 % **dari biaya**, bukan dari harga jual. Kalau laba 25 % dihitung dari harga jual, harga jualnya menjadi 2.400.000 ÷ 0,75 = Rp3.200.000, angka yang berbeda.

Contoh 3 — Titik impas aplikasi pengaduan RT

Tim Lapor Warga (2027) membuat aplikasi pengaduan untuk pengurus RT. Pelanggan membayar langganan Rp50.000 per RT per bulan. Biaya tetap bulanan Rp2.400.000 (server dan honor admin), dan biaya variabel Rp10.000 per RT per bulan (pesan notifikasi). (a) Berapa RT yang harus berlangganan agar tim impas? (b) Berapa laba bila 80 RT berlangganan? (c) Bagaimana titik impas bila biaya notifikasi naik menjadi Rp15.000? *(Data fiktif.)*

1. (a) Sumbangan per RT = 50.000 − 10.000 = Rp40.000. Titik impas = 2.400.000 ÷ 40.000 = **60 RT**.
2. (b) Laba = 80 × 40.000 − 2.400.000 = 3.200.000 − 2.400.000 = **Rp800.000**.

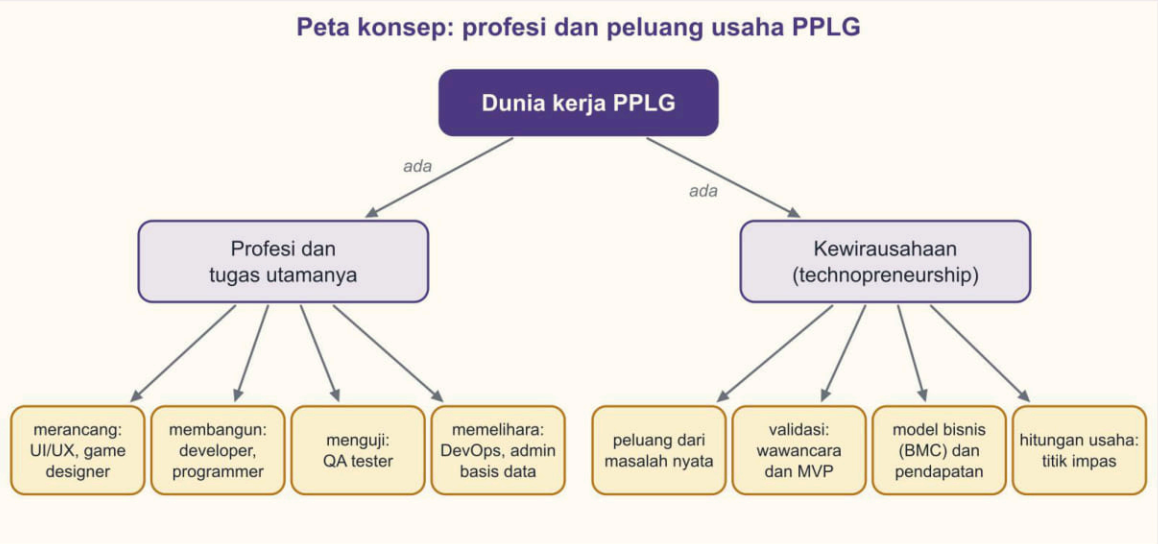
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. (c) Sumbangan baru = 50.000 − 15.000 = 35.000. Titik impas = 2.400.000 ÷ 35.000 ≈ 68,57. Jumlah RT harus bilangan bulat, jadi pembulatan **ke atas**: **69 RT**. Pada 68 RT masih rugi (68 × 35.000 = 2.380.000 < 2.400.000).
4. Cek (a): 60 × 50.000 = 3.000.000; biaya total = 2.400.000 + 60 × 10.000 = 3.000.000 ✓.

Jawaban: (a) 60 RT, (b) Rp800.000, (c) 69 RT. Pembulatan titik impas selalu ke atas, sebab pembulatan ke bawah masih menyisakan kerugian.

❖ PETA KONSEP

Gambar 1.5 merangkum hubungan antarkonsep pada sub bab ini: dunia kerja PPLG bercabang ke profesi dan ke kewirausahaan.



Gambar 1.5 Peta konsep profesi dan peluang usaha PPLG. Sumber: digambar oleh Tim Next Level Study Indonesia.

Yang harus kamu ingat:

- Profesi dibedakan oleh **kata kerja utama dan objek kerjanya**: merancang, membangun, menguji, atau memelihara; tampilan, server, data, atau aturan gim.
- QA tester **menguji dan melapor**; yang memperbaiki bug adalah pembuat kodenya.
- Wirausaha teknologi memulai dari **masalah nyata**, memvalidasinya dengan **wawancara dan MVP**, lalu memetakan model bisnisnya.
- **Biaya tetap** tidak berubah menurut jumlah terjual; **biaya variabel** ikut bertambah.
- **Titik impas = biaya tetap ÷ (harga jual − biaya variabel)**, dibulatkan ke atas; laba = sumbangan per unit × jumlah − biaya tetap.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal profesi:

1. **Garis bawahi kata kerja pada uraian tugas** (merancang, menulis kode, menguji, mengelola, menggali).
2. **Tentukan objek kerjanya**: tampilan, server, basis data, bug, jadwal, aturan gim.
3. **Cocokkan dengan profesi yang tugas utamanya persis itu**. Profesi yang "juga bisa" melakukannya tetap bukan jawaban.

Langkah menjawab soal usaha:

1. **Tulis lebih dulu apa yang diketahui**: biaya tetap, biaya variabel per unit, harga per unit.
2. **Pisahkan biaya tetap dan variabel** sebelum menghitung. Biaya per pelanggan per bulan adalah variabel; sewa dan honor bulanan adalah tetap.
3. **Untuk pilihan langkah usaha**, pilih langkah yang murah, menguji pasar lebih dulu, dan berbasis data, bukan yang membangun semuanya lebih dulu.

Jebakan yang sering dipasang:

- **Profesi yang berdekatan tertukar**, misalnya QA tester dengan developer (penguji tidak memperbaiki kode), game designer dengan game programmer (satu merancang aturan, satu menulis kodenya), UI dengan UX.
- **Opsi "yang juga memakai basis data"**. Back-end developer memang berhubungan dengan basis data, tetapi bila uraian tugas berpusat pada cadangan dan hak akses, jawabannya administrator basis data.
- **Biaya variabel dianggap tetap** sehingga titik impas dihitung dengan angka yang salah.
- **Laba dianggap sama dengan penjualan.** Laba = penjualan dikurangi seluruh biaya, termasuk biaya tetap.
- **Membangun semua fitur sebelum ada pengguna** tampak "lengkap", tetapi tidak sejalan dengan prinsip MVP.
- **Menjiplak aset atau memakai perangkat lunak bajakan demi hemat** tidak pernah menjadi langkah yang benar.

> Contoh salah tafsir: sebuah usaha jasa membuat aplikasi memiliki biaya tetap Rp2.000.000 dan sumbangan per pesanan Rp80.000. Seorang siswa menghitung titik impas sebagai 2.000.000 ÷ 80.000 = 25 pesanan, lalu menyimpulkan "pada pesanan ke-25 usaha mulai untung". Pada pesanan ke-25 usaha tepat impas, belum untung. Keuntungan baru ada mulai pesanan ke-26.

● BEDAH SOAL PREDIKSI

Soal berikut meminta jumlah toko pelanggan agar sebuah usaha aplikasi kasir tidak rugi. Perhatikan bagaimana empat komponen biaya di tabel harus **dipilah** menjadi biaya tetap dan biaya variabel lebih dulu (bagian 7), dan bagaimana tiap pilihan jawaban merupakan hasil dari satu kesalahan memilah.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Startup lulusan SMK menjual aplikasi kasir berlangganan untuk warung dengan harga Rp75.000 per toko per bulan (2027). Rincian biaya bulanannya seperti pada tabel.

Komponen biaya	Besar biaya
Sewa server cloud	Rp1.200.000 per bulan
Domain, email, dan layanan lain	Rp300.000 per bulan
Honor admin paruh waktu	Rp3.000.000 per bulan
Pesan notifikasi WhatsApp	Rp15.000 per toko per bulan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Agar pendapatan bulanan dapat menutup seluruh biaya bulanan (tidak rugi), jumlah toko pelanggan paling sedikit adalah ....

A. 50 toko
B. 60 toko
C. 61 toko
D. 75 toko
E. 300 toko

Pembahasan **Kunci: D**

Konsep: Titik impas (tidak untung, tidak rugi) tercapai saat pendapatan = biaya. Setiap pelanggan menyumbang (harga − biaya per pelanggan) untuk menutup biaya tetap: jumlah pelanggan = biaya tetap ÷ (harga − biaya per pelanggan).

Langkah:
1. Biaya tetap = 1.200.000 + 300.000 + 3.000.000 = Rp4.500.000 per bulan.
2. Sumbangan per toko = 75.000 − 15.000 = Rp60.000.
3. Jumlah toko = 4.500.000 ÷ 60.000 = 75 toko.
4. Cek: 75 × 75.000 = 5.625.000; biaya = 4.500.000 + 75 × 15.000 = 5.625.000 → impas.

Analisis setiap opsi:
- **A.** ✗ Jebakan: biaya WhatsApp ditambahkan ke harga: 4.500.000 ÷ 90.000 = 50.
- **B.** ✗ Jebakan: biaya WhatsApp per toko dilupakan: 4.500.000 ÷ 75.000 = 60.
- **C.** ✗ Biaya WhatsApp dianggap Rp15.000 untuk semua toko: 4.515.000 ÷ 75.000 = 60,2 → 61 toko.
- **D.** ✓ Biaya tetap Rp4.500.000 dibagi sumbangan per toko Rp60.000 = 75 toko.
- **E.** ✗ Biaya tetap dibagi biaya WhatsApp per toko: 4.500.000 ÷ 15.000 = 300.

Jawaban: D.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk soal profesi, garis bawahi kata kerja pada uraian tugas lebih dulu. Untuk soal usaha, pisahkan biaya tetap dan biaya variabel.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. Studio gim Lentera Interaktif akan merilis gim edukasi *Petualangan Aksara* pada Juni 2027. Selama dua minggu sebelum rilis, studio membutuhkan seseorang yang memainkan versi uji gim mengikuti daftar skenario, mencatat setiap kesalahan beserta langkah untuk memunculkannya kembali dan tangkapan layarnya, melaporkannya ke sistem pelacak bug, lalu memeriksa ulang setelah programmer menyatakan kesalahan itu sudah diperbaiki. *(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)*
Profesi yang paling sesuai dengan tugas tersebut adalah ....

A. Game designer yang merancang aturan, misi, dan tingkat kesulitan
B. UI/UX designer yang merancang tata letak layar dan alur pemakaian
C. QA tester yang menguji perangkat lunak dan melaporkan kesalahan
D. Backend developer yang membangun server dan basis data gim
E. Release engineer yang mengemas dan mengunggah versi rilis gim

2. Studio gim Rumah Piksel sedang membuat gim platformer *Si Tupai Lincah* (2027). Studio membutuhkan orang yang menentukan letak pijakan, rintangan, musuh, dan koin pada setiap level; mengatur agar tingkat kesulitan naik bertahap dari level 1 sampai level 10; serta mencoba alur setiap level memakai prototipe kotak-kotak sederhana sebelum gambar karakter dan latar selesai dibuat.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Profesi yang paling sesuai dengan kebutuhan studio tersebut adalah ....

A. Game artist yang menggambar karakter, latar, dan aset visual gim
B. Game programmer yang menulis kode gerak karakter dan fisika gim
C. Sound designer yang membuat efek suara dan musik latar gim
D. Level designer yang merancang susunan rintangan setiap level
E. QA tester yang menguji gim dan melaporkan kesalahan program

3. Aplikasi bank sampah RW 05 akan dipakai banyak warga lanjut usia (2027). Sebelum programmer mulai menulis kode, tim membutuhkan orang yang mewawancarai beberapa warga, menggambar rancangan layar dan prototipe yang dapat diklik, lalu mengamati warga mencoba prototipe itu untuk menemukan tombol atau alur yang membingungkan. *(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)*
Profesi yang paling sesuai untuk tugas tersebut adalah ....

A. Frontend developer yang menulis kode tampilan dengan HTML dan CSS
B. UI/UX designer yang merancang tampilan dan menguji kemudahan pakai
C. Database administrator yang merancang tabel dan hak akses data
D. Technical writer yang menyusun buku panduan pemakaian aplikasi
E. Mobile developer yang memprogram dan mengunggah aplikasi Android

4. Tiga lulusan SMK jurusan PPLG berencana membuat aplikasi antrean servis untuk bengkel motor kecil pada tahun 2027. Modal mereka hanya Rp6.000.000 dan mereka belum tahu

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

apakah pemilik bengkel bersedia membayar langganan. Seorang mentor inkubator bisnis meminta mereka menyusun langkah awal usaha yang hemat modal tetapi tetap berpeluang berkembang.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Langkah awal yang sesuai dengan prinsip kewirausahaan di bidang perangkat lunak adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Mewawancarai beberapa pemilik bengkel untuk memastikan masalah antrean benar-benar mereka alami.

☐ B. Membangun semua fitur, yaitu antrean, stok suku cadang, dan akuntansi, sebelum ada pengguna.

☐ C. Membuat versi awal (MVP) berisi fitur antrean saja, lalu mengujinya di dua bengkel.

☐ D. Menghitung harga langganan dari biaya server bulanan dan kesediaan bengkel membayar.

☐ E. Memakai ikon dan gambar dari aplikasi pesaing agar tampilan aplikasi cepat selesai.

5. Dua lulusan SMK PPLG di Kecamatan Sukamaju ingin membuka jasa pembuatan website untuk UMKM (2027). Hasil survei mereka: dari 40 UMKM, 28 belum punya website; kebanyakan hanya ingin katalog produk dan tombol pesan WhatsApp; banyak yang tidak sanggup membayar di muka lebih dari Rp1.500.000. Satu jasa serupa di kota mematok Rp4.000.000 per website.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Kesimpulan dan rencana yang tepat berdasarkan survei adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Peluang usaha kurang menarik karena sudah ada pesaing di kota, sehingga rencana sebaiknya dibatalkan.

☐ B. Paket sederhana berisi katalog dan tombol WhatsApp sesuai kebutuhan yang paling banyak ditemukan.

☐ C. Biaya dapat dibagi menjadi pembayaran awal kecil ditambah biaya perawatan bulanan agar terjangkau.

☐ D. Template website berbayar versi bajakan dapat dipakai agar harga ditekan serendah mungkin.

☐ E. Harga di bawah pesaing kota menjadi keunggulan selama tetap menutup biaya dan waktu kerja.

Kunci singkat: 1. C 2. D 3. B 4. A, C, D 5. B, C, E

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa seorang QA tester yang menemukan bug tidak langsung memperbaikinya, dan siapa yang seharusnya memperbaiki?
- Apa yang membedakan biaya tetap dari biaya variabel, dan bagaimana keduanya masuk ke rumus titik impas?
- **Yang harus kuingat:** profesi dikenali dari kata kerja dan objek kerjanya; usaha dimulai dari masalah nyata dan diuji dengan MVP; titik impas = biaya tetap ÷ (harga − biaya variabel), dibulatkan ke atas.

Latihan interaktif di LMS

Sub bab 1.1: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-1.1

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 1.2 · Manajemen proyek dan budaya mutu

SMKPPLG-1.2

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **mengidentifikasi tahapan dasar manajemen proyek**, memahami **proses bisnis** pengembangan perangkat lunak dan gim, serta **menjelaskan prinsip budaya mutu**. Levelnya pengetahuan dan pemahaman, tetapi pada soal kasus kamu diminta memilih keputusan yang tepat: apa yang dilakukan ketua tim ketika klien menambah permintaan, jadwal mulai terlambat, atau fitur belum memenuhi kriteria.

Soal pada sub bab ini hampir selalu memakai kasus tim PKL atau studio kecil. Sulitnya jarang pada istilah. Sulitnya ada pada **memilih tindakan yang paling tepat** di antara beberapa tindakan yang masuk akal, atau pada **menelusuri jadwal** dari tabel kegiatan.

Setelah mempelajari sub bab ini, kamu dapat:

- menyebutkan lima fase manajemen proyek dan kegiatan khas tiap fase;
- membedakan model waterfall dan agile (scrum) dan memilih yang sesuai dengan kebutuhan proyek;
- mengelola ruang lingkup dan permintaan perubahan dengan cara yang benar;
- menentukan jalur kritis dan lama proyek dari tabel kegiatan;
- menjelaskan kriteria penerimaan, definisi selesai, tinjauan kode, pedoman penulisan kode, dan siklus perbaikan berkelanjutan sebagai bagian dari budaya mutu.

◆ KONSEP INTI

Membuat aplikasi untuk satu orang yang butuh cepat berbeda dengan membuat aplikasi untuk klien yang membayar, dengan tim berempat, tenggat enam minggu, dan banyak keinginan yang berubah di tengah jalan. Tanpa pengaturan, proyek seperti itu mudah molor, membengkak biayanya, atau menghasilkan produk yang tidak dipakai. Pengaturan inilah yang disebut **manajemen proyek**. Sedangkan **budaya mutu** menjamin bahwa produk yang diserahkan memang baik dan bahwa tim terus memperbaiki cara kerjanya.

1. Apa itu proyek, dan apa yang harus dijaga

Proyek adalah kegiatan **sementara** dengan **tujuan khusus** dan **hasil yang unik**: memiliki awal dan akhir, berbeda dengan pekerjaan rutin seperti menjaga server tetap hidup setiap hari. Membuat aplikasi presensi untuk sebuah sekolah adalah proyek; membalas keluhan pengguna setiap hari adalah pekerjaan rutin.

Pengelola proyek harus menjaga tiga hal yang saling menekan, sering disebut **segitiga kendala**:

- **ruang lingkup**: apa saja yang harus dihasilkan;

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **waktu**: kapan harus selesai;
- **biaya**: berapa sumber daya yang tersedia.

Ketiganya terikat dengan **mutu** di tengahnya. Kalau satu sisi diubah, sisi lain terpengaruh. Menambah fitur (ruang lingkup naik) tanpa menambah waktu atau biaya memaksa mutu turun, misalnya pengujian dipangkas. Karena itu perubahan harus dibicarakan, bukan diam-diam ditelan oleh tim.

2. Lima fase manajemen proyek

Proyek berjalan melalui lima fase (Gambar 1.6).



Gambar 1.6 Lima fase manajemen proyek. Fase pemantauan dan pengendalian berjalan selama perencanaan dan pelaksanaan. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Inisiasi**: menilai apakah proyek layak dikerjakan, mengenali pihak yang berkepentingan (klien, pengguna, tim), dan menyepakati **piagam proyek** yang memuat tujuan dan batas wewenang.
2. **Perencanaan**: menetapkan **ruang lingkup** secara tertulis, memecah pekerjaan menjadi kegiatan, menyusun **jadwal** dan **anggaran**, membagi tugas, dan memikirkan **risiko**.
3. **Pelaksanaan**: mengerjakan produk (merancang, membuat, menguji) sambil berkoordinasi dengan klien.
4. **Pemantauan dan pengendalian**: membandingkan kemajuan nyata dengan rencana, lalu mengambil **tindakan perbaikan** bila menyimpang. Fase ini berjalan sepanjang perencanaan dan pelaksanaan, bukan hanya di akhir.
5. **Penutupan**: serah terima hasil, dokumentasi, dan **evaluasi** untuk pelajaran proyek berikutnya.

Kesalahan yang sering muncul pada soal: menukar inisiasi dengan perencanaan. Bila kegiatannya menilai kelayakan dan menyepakati tujuan, itu inisiasi. Bila kegiatannya menyusun jadwal, anggaran, dan pembagian tugas, itu perencanaan.

3. Proses bisnis pengembangan perangkat lunak

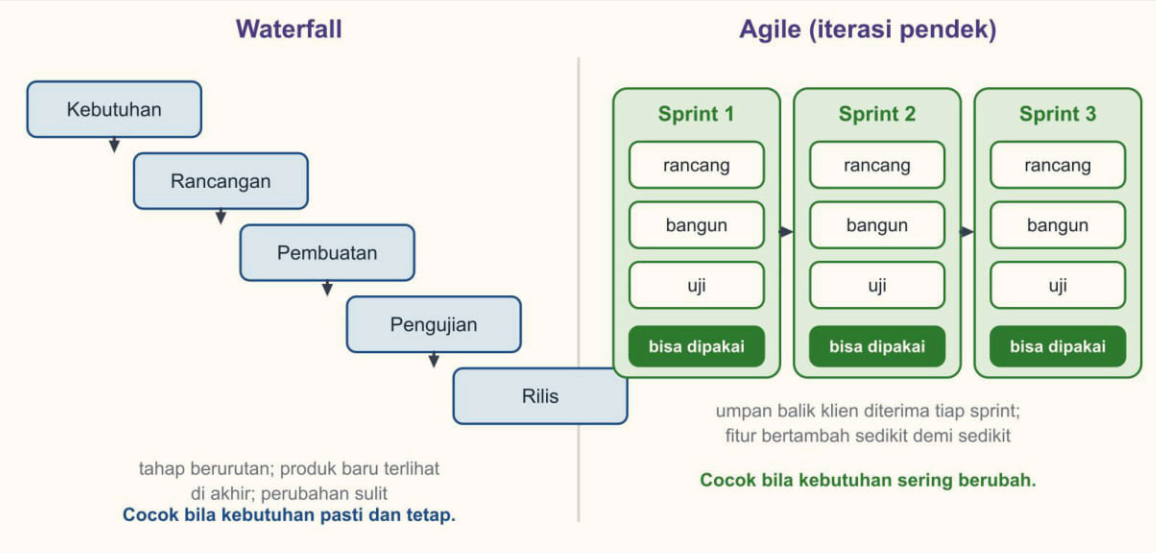
Proses bisnis adalah rangkaian kegiatan yang saling terkait untuk menghasilkan nilai bagi pelanggan. Pada studio yang menerima pesanan aplikasi, rangkaiannya kurang lebih:

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

permintaan klien → penawaran dan kesepakatan (kontrak) → pengerjaan proyek → pengujian → serah terima dan pembayaran → dukungan setelah rilis. Pada studio gim yang menjual produk sendiri: **ide → rancangan → produksi → uji → rilis di toko aplikasi → pembaruan berdasarkan ulasan pemain**. Memahami proses bisnis membantumu melihat bahwa pekerjaan teknis hanya satu bagian dari rangkaian yang lebih besar, termasuk perjanjian dengan klien dan layanan sesudah rilis.

4. Memilih cara kerja: waterfall dan agile

Model pengembangan mengatur urutan kegiatan. Dua model yang perlu kamu kenal ditunjukkan pada Gambar 1.7.

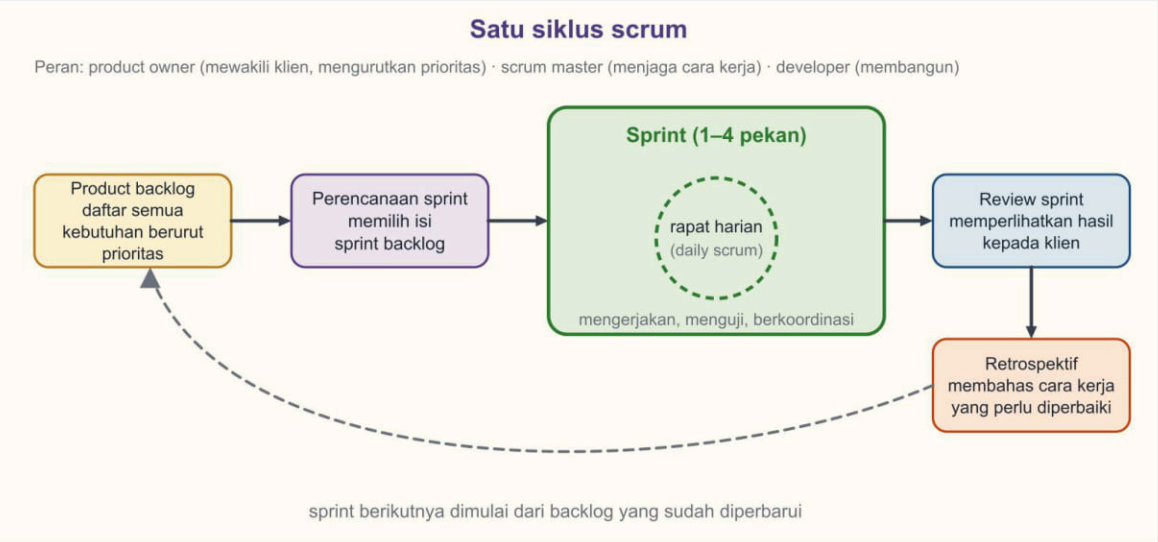


Gambar 1.7 Waterfall mengerjakan tahap demi tahap; agile mengulang siklus pendek yang menghasilkan bagian produk yang bisa dipakai. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Waterfall (air terjun)**: tahap dikerjakan berurutan, yaitu kebutuhan → rancangan → pembuatan → pengujian → rilis. Tahap berikutnya baru mulai setelah tahap sebelumnya selesai. Model ini cocok bila **kebutuhan sudah pasti dan jarang berubah**, tetapi produk baru terlihat di akhir dan perubahan di tengah jalan mahal.
- **Agile**: pekerjaan dipecah menjadi putaran pendek (iterasi) yang masing-masing menghasilkan bagian produk yang dapat dicoba. Klien memberi tanggapan tiap putaran. Model ini cocok bila **kebutuhan belum jelas atau sering berubah**, misalnya gim baru atau aplikasi untuk pasar yang belum dikenal.

Kerangka agile yang paling populer adalah **scrum** (Gambar 1.8).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 1.8 Satu siklus scrum: dari product backlog, perencanaan sprint, sprint, review, sampai retrospektif. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Product backlog**: daftar semua kebutuhan produk yang diurutkan menurut prioritas, dikelola oleh **product owner**, orang yang mewakili klien.
- **Sprint**: putaran kerja dengan panjang tetap, biasanya satu sampai empat pekan. Sebelum sprint dimulai, tim memilih sebagian backlog sebagai **sprint backlog** pada rapat **perencanaan sprint**.
- **Rapat harian (daily scrum)**: pertemuan singkat tiap hari agar tim saling mengetahui kemajuan dan hambatan.
- **Review sprint**: tim memperlihatkan hasil sprint kepada klien untuk ditanggapi.
- **Retrospektif**: tim membahas cara kerjanya sendiri: apa yang sudah baik, apa yang perlu diperbaiki.
- **Scrum master** menjaga agar tim bekerja sesuai aturan scrum dan membantu menyingkirkan hambatan; **developer** membangun produknya.

Tim scrum biasanya mengukur pekerjaan dengan **poin tugas**. **Kecepatan tim (velocity)** adalah jumlah poin yang benar-benar selesai tiap sprint. Bila rencana 30 poin tetapi realisasi hanya 22 dan 21 poin pada dua sprint berturut-turut, tim jangan berharap kecepatannya akan naik dengan sendirinya: sisa pekerjaan harus **diprioritaskan** dengan klien (fitur wajib lebih dulu, fitur tambahan menyusul), bukan memaksa semuanya muat.

5. Ruang lingkup dan permintaan perubahan

Ruang lingkup adalah batas pekerjaan yang disepakati: fitur apa yang dikerjakan dan apa yang tidak. Ia ditulis di dokumen yang disetujui klien dan tim. Masalah muncul ketika pekerjaan **bertambah sedikit demi sedikit tanpa kesepakatan baru**, yang disebut ***scope creep***.

Klien berhak meminta perubahan. Yang benar adalah mengelolanya dengan langkah berikut:

1. **Catat** permintaan sebagai permintaan perubahan.
2. **Nilai dampaknya** terhadap waktu, biaya, dan mutu.

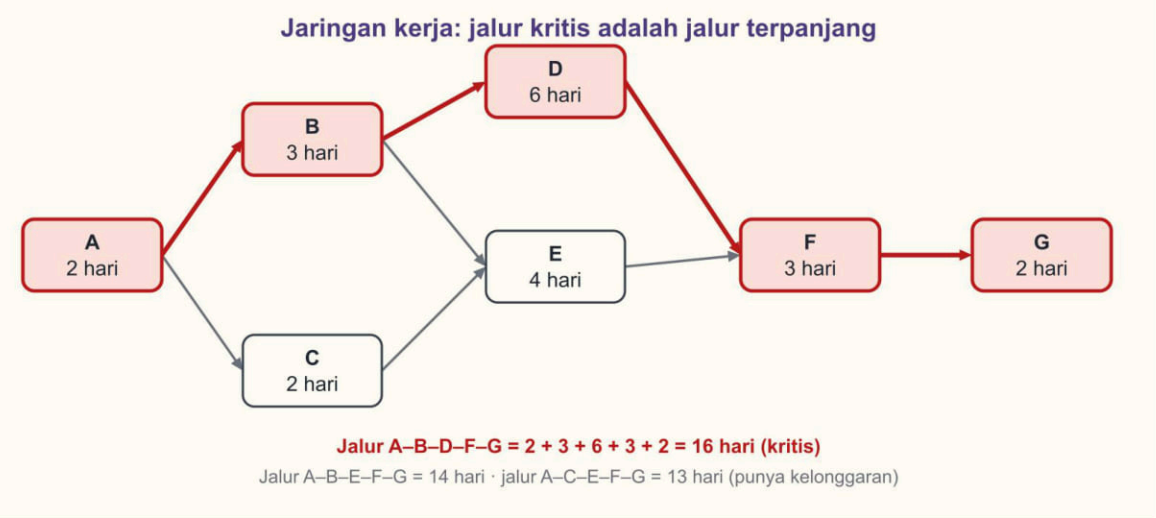
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. **Bicarakan pilihan dengan klien**: menambah waktu atau biaya, menukar dengan fitur lain, atau menjadikannya bagian versi berikutnya.
4. **Putuskan bersama**, lalu **perbarui** dokumen ruang lingkup dan jadwal.

Tindakan yang keliru: menerima semuanya dengan lembur diam-diam, menolak mentah-mentah tanpa membicarakannya, memangkas pengujian agar jadwal muat, atau menunda pembicaraan sampai produk selesai.

6. Jadwal kegiatan, jaringan kerja, dan jalur kritis

Proyek dipecah menjadi **kegiatan** yang masing-masing punya lama pengerjaan dan **pendahulu** (kegiatan yang harus selesai lebih dulu). Kegiatan yang tidak saling menunggu dapat dikerjakan **bersamaan** oleh orang berbeda. Hubungan kegiatan digambar sebagai **jaringan kerja** (Gambar 1.9).



Gambar 1.9 Jaringan kerja tujuh kegiatan. Jalur merah adalah jalur terpanjang (jalur kritis). Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Cara membacanya:

1. Daftar **semua jalur** dari kegiatan awal ke kegiatan akhir.
2. **Jumlahkan lama** kegiatan pada tiap jalur.
3. Jalur **terpanjang** disebut **jalur kritis**. Panjangnya adalah **lama proyek paling cepat**.
4. Keterlambatan kegiatan pada jalur kritis **langsung** memundurkan penyelesaian proyek hari demi hari.
5. Kegiatan di luar jalur kritis punya **kelonggaran**: selisih antara panjang jalur kritis dan panjang jalur yang memuat kegiatan itu. Selama keterlambatannya tidak melebihi kelonggaran, proyek tidak mundur.

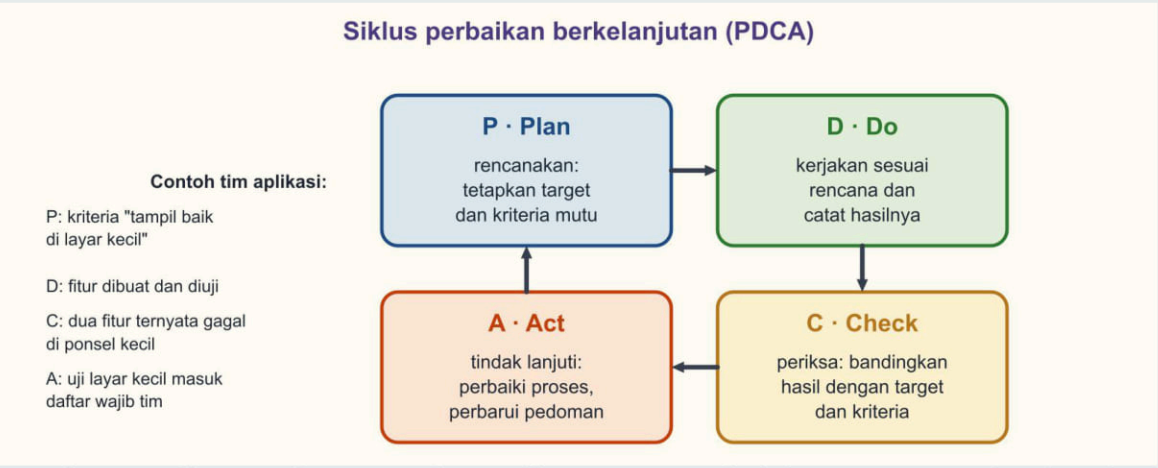
Perhatikan bahwa lama proyek **bukan** jumlah semua kegiatan, karena sebagian kegiatan berjalan paralel. Kegiatan yang punya dua pendahulu (misalnya E menunggu B dan C) baru dapat mulai setelah **keduanya** selesai, jadi yang menentukan adalah pendahulu yang selesai paling lambat.

7. Budaya mutu

Mutu perangkat lunak berarti produk memenuhi kebutuhan pemakai, bebas dari cacat yang mengganggu, dan mudah dirawat. **Budaya mutu** adalah kebiasaan seluruh tim yang menjaga mutu itu sejak awal, bukan pemeriksaan di menit terakhir oleh satu orang. Ada beberapa praktik utama.

- **Kriteria penerimaan (acceptance criteria)**: pernyataan yang **dapat diperiksa dengan jawaban ya atau tidak** tentang kapan sebuah fitur diterima klien, misalnya "total belanja berubah otomatis ketika jumlah barang diubah". Kriteria yang baik jelas dan terukur, bukan "tampilannya bagus".
- **Definisi selesai (definition of done)**: daftar syarat yang disepakati tim agar sebuah pekerjaan boleh disebut selesai, misalnya sudah diuji, sudah ditinjau rekan, dan sudah didokumentasikan. Syarat itu **tidak dilewati demi mengejar jadwal**.
- **Pengujian**: menjalankan skenario untuk menemukan cacat, baik manual maupun otomatis. Semakin awal cacat ditemukan, semakin murah memperbaikinya.
- **Tinjauan kode (code review)**: rekan satu tim membaca perubahan kode sebelum digabung, sehingga kesalahan tertangkap dan pengetahuan menyebar.
- **Pedoman penulisan kode**: kesepakatan gaya, misalnya nama variabel dan fungsi yang deskriptif, keterangan singkat pada fungsi, dan format seragam. Kode yang **sudah berjalan benar tetap perlu mengikuti pedoman**, karena kode dibaca jauh lebih sering daripada ditulis.
- **Perbaikan berkelanjutan**: setelah masalah terjadi, tim mencatat penyebab, memperbaiki proses, dan memeriksa hasilnya. Siklus ini terkenal sebagai **PDCA** (Plan–Do–Check–Act), seperti pada Gambar 1.10.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 1.10 Siklus PDCA dengan contoh tim aplikasi yang memperbaiki kriteria ujinya. Sumber: digambar oleh Tim Next Level Study Indonesia.

Intinya, mutu adalah **tanggung jawab bersama**. Budaya yang sehat tidak menghukum orang yang melakukan kesalahan, tetapi memperbaiki prosesnya agar kesalahan itu tidak berulang.

CONTOH TERSELESAIKAN

Contoh 1 — Jalur kritis dan akibat keterlambatan

Koperasi Desa Sumber Rejeki (2027) memesan aplikasi inventaris gudang. Tim menyusun kegiatan berikut. Kegiatan yang tidak saling menunggu dikerjakan bersamaan. *(Data fiktif.)*

Kegiatan	Lama (hari)	Dikerjakan setelah
A. Wawancara kebutuhan	2	–
B. Desain basis data	3	A
C. Desain antarmuka	2	A
D. Pemrograman modul stok	6	B
E. Pemrograman modul laporan	4	B dan C
F. Pengujian	3	D dan E
G. Pelatihan pengguna	2	F

Tentukan (a) lama proyek paling cepat dan jalur kritisnya, (b) lama proyek bila E terlambat 3 hari, (c) lama proyek bila D dipercepat 2 hari.

1. Daftar jalur: A–B–D–F–G = 2 + 3 + 6 + 3 + 2 = **16**; A–B–E–F–G = 2 + 3 + 4 + 3 + 2 = **14**; A–C–E–F–G = 2 + 2 + 4 + 3 + 2 = **13**.
2. (a) Jalur terpanjang = A–B–D–F–G, jadi lama proyek paling cepat **16 hari** dan jalur kritisnya A–B–D–F–G (Gambar 1.9).
3. Kelonggaran E: jalur yang memuat E paling panjang 14, jadi kelonggarannya 16 − 14 = 2 hari.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. (b) E terlambat 3 hari, melebihi kelonggaran 2 hari. Jalur A–B–E–F–G menjadi 14 + 3 = 17 hari, lebih panjang dari 16. Proyek selesai dalam **17 hari**, mundur 1 hari.
5. (c) D menjadi 4 hari: A–B–D–F–G = 2 + 3 + 4 + 3 + 2 = 14, sedangkan A–B–E–F–G tetap 14. Proyek selesai dalam **14 hari**, bukan 12. Sekarang ada **dua** jalur kritis; mempercepat D lebih jauh tidak berguna lagi sebelum E ikut dipercepat.

Jawaban: (a) 16 hari, jalur A–B–D–F–G; (b) 17 hari; (c) 14 hari.

Contoh 2 — Waterfall atau agile?

Dua proyek ditawarkan kepada sebuah studio kecil pada 2027. Proyek 1: sistem pencatatan retribusi pasar yang aturannya sudah ditetapkan peraturan daerah, dengan formulir dan perhitungan yang tidak boleh berubah. Proyek 2: gim edukasi yang belum pernah dibuat studio itu dan idenya baru separuh jadi, sehingga pemilik ingin mencoba versi awal ke anak-anak, lalu menyesuaikannya. Model mana yang lebih cocok untuk masing-masing? *(Kasus fiktif.)*

1. Tanyakan **seberapa pasti dan stabil kebutuhannya**.
2. Proyek 1: kebutuhan **pasti dan tetap** (diatur peraturan). Pekerjaan berurutan dengan rencana lengkap sejak awal cocok → **waterfall**.
3. Proyek 2: kebutuhan **belum pasti** dan butuh tanggapan pengguna di tengah jalan. Putaran pendek dengan uji coba tiap putaran lebih cocok → **agile (scrum)**.
4. Tambahkan catatan: pilihan model bukan aturan mutlak. Proyek 1 pun boleh memakai putaran pendek bila klien meminta, tetapi alasan utamanya tetap kepastian kebutuhan.

Jawaban: proyek 1 → waterfall; proyek 2 → agile (scrum).

Contoh 3 — Memeriksa fitur terhadap kriteria penerimaan

Aplikasi toko daring sebuah usaha batik memiliki empat kriteria penerimaan untuk halaman **Keranjang** (2027).

- K1. Setiap baris menampilkan nama barang, harga satuan, jumlah, dan subtotal.
- K2. Tombol "−" tidak dapat menurunkan jumlah di bawah 1.
- K3. Total belanja berubah otomatis ketika jumlah diubah.
- K4. Tombol "Checkout" nonaktif bila keranjang kosong.

Hasil uji tim: baris memuat nama, harga, dan jumlah, tetapi tidak ada kolom subtotal; tombol "−" dapat menurunkan jumlah sampai 0; total berubah otomatis saat jumlah diubah; tombol "Checkout" nonaktif ketika keranjang kosong. Apa simpulan dan keputusan tim? *(Kasus fiktif.)*

1. Cocokkan hasil uji dengan tiap kriteria **satu per satu**.
2. K1: subtotal tidak ada → **belum terpenuhi**.
3. K2: jumlah bisa 0 → **belum terpenuhi**.
4. K3: terpenuhi. K4: terpenuhi.
5. Dua kriteria belum terpenuhi, sehingga fitur **belum memenuhi definisi selesai**. Keputusan: perbaiki K1 dan K2, uji ulang, baru serahkan. Menyerahkan lebih dulu "lalu memperbaikinya di versi berikutnya" mengabaikan kriteria yang sudah disepakati.

Jawaban: K3 dan K4 terpenuhi, K1 dan K2 belum; fitur belum boleh diserahkan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Contoh 4 — Permintaan fitur baru di tengah proyek

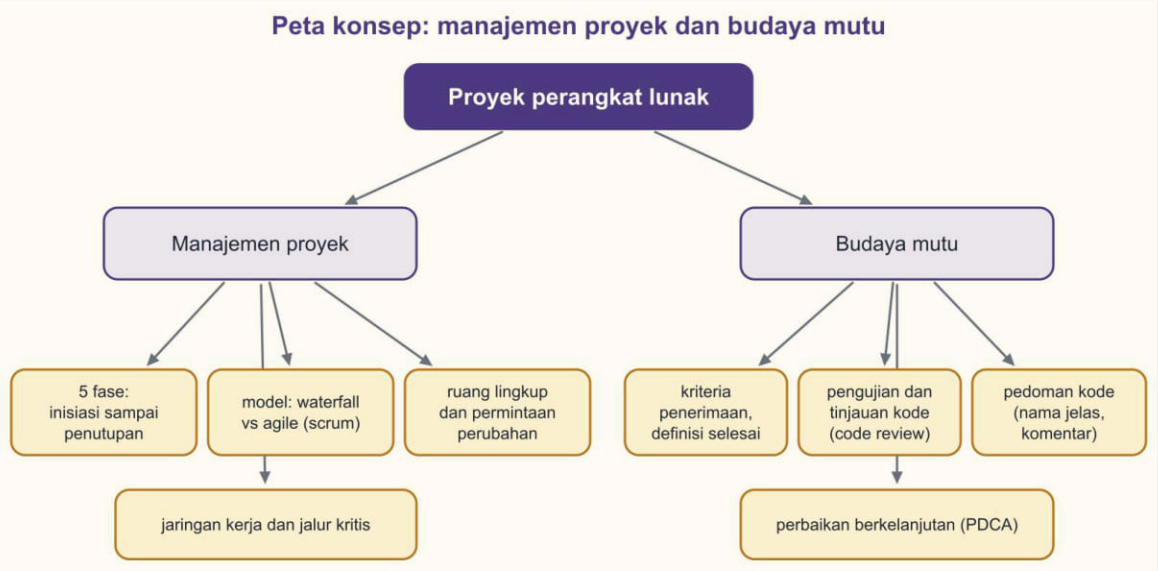
Pada minggu ke-5 dari proyek delapan minggu aplikasi presensi sebuah lembaga kursus (2027), pemilik meminta tambahan laporan kehadiran otomatis berformat PDF. Tim memperkirakan fitur itu membutuhkan sepuluh hari kerja, padahal sisa waktu hanya tiga pekan dan sebagian besar sudah terisi pengujian. Apa langkah yang tepat? *(Kasus fiktif.)*

1. **Catat** sebagai permintaan perubahan; jangan langsung menerima atau menolak.
2. **Nilai dampak**: sepuluh hari kerja melebihi sisa waktu yang longgar. Bila dipaksakan, pengujian terpotong dan mutu turun.
3. **Susun pilihan untuk dibicarakan**: (i) menambah waktu atau biaya; (ii) menukar dengan fitur lain yang kurang penting; (iii) merilis tanpa laporan PDF dan memasukkannya ke versi berikutnya.
4. **Putuskan bersama pemilik**, lalu perbarui dokumen ruang lingkup dan jadwal.

Jawaban: catat, hitung dampak, bicarakan pilihan, dan putuskan bersama pemilik. Jangan lembur diam-diam, menolak tanpa bicara, memotong pengujian, atau menunda sampai rilis.

❖ PETA KONSEP

Gambar 1.11 merangkum hubungan antarkonsep: proyek perangkat lunak dikelola lewat manajemen proyek dan dijaga lewat budaya mutu.



Gambar 1.11 Peta konsep manajemen proyek dan budaya mutu. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Yang harus kamu ingat:

- Lima fase: **inisiasi → perencanaan → pelaksanaan → pemantauan dan pengendalian → penutupan**. Kelayakan dan piagam proyek = inisiasi; jadwal dan anggaran = perencanaan.
- **Waterfall** untuk kebutuhan yang pasti; **agile/scrum** untuk kebutuhan yang berubah. Scrum: backlog → sprint → review → retrospektif.
- Permintaan perubahan: **catat, nilai dampak, bicarakan, putuskan bersama**, lalu perbarui ruang lingkup.
- **Jalur kritis = jalur terpanjang** = lama proyek paling cepat. Keterlambatan di luar jalur kritis tidak memundurkan proyek selama tidak melebihi kelonggarannya.
- Budaya mutu: **kriteria penerimaan, definisi selesai, pengujian, tinjauan kode, pedoman kode, dan PDCA**, dilakukan seluruh tim sejak awal.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal jadwal:

1. **Tulis semua jalur** dari kegiatan awal ke akhir dan jumlahkan lamanya.
2. **Tandai jalur terpanjang** sebagai jalur kritis.
3. **Untuk keterlambatan atau percepatan**, ubah lamanya lalu hitung ulang **semua** jalur. Jalur kritis bisa berpindah.

Langkah menjawab soal keputusan tim:

1. **Tentukan apa yang terjadi**: permintaan perubahan, jadwal melenceng, kriteria belum terpenuhi, atau kesalahan setelah rilis.
2. **Pilih tindakan yang memperbaiki proses dan melibatkan klien atau seluruh tim.** Tindakan itu biasanya mencatat, menilai dampak, membicarakan, lalu memutuskan bersama.
3. **Coret** tindakan yang diam-diam menambah beban (lembur tanpa bicara), memangkas pengujian, menunda bicara, atau menyalahkan satu orang.

Jebakan yang sering dipasang:

- **Lama proyek dijumlah seluruhnya**, padahal ada kegiatan paralel.
- **Jalur kritis dikira kegiatan terpanjang tunggal.** Yang kritis adalah jalur dengan jumlah terpanjang, bukan satu kegiatan.
- **Pemantauan dikira hanya di akhir**, padahal berjalan sepanjang proyek.
- **Inisiasi dan perencanaan tertukar.**
- **"Kode sudah berjalan, jadi tidak perlu mengikuti pedoman."** Pedoman berlaku untuk kode yang benar sekalipun.
- **Tinjauan kode dipakai untuk mencari siapa yang salah.** Tujuannya menangkap cacat dan menyebarkan pengetahuan.
- **Kriteria dilewati karena dikejar jadwal.** Kriteria yang sudah disepakati tidak dihapus demi cepat rilis.

> Contoh salah tafsir: pada jaringan kerja, kegiatan X berlama 5 hari dan berada di jalur yang panjangnya 12 hari, sedangkan jalur kritis 15 hari. Seorang siswa menyimpulkan "X berlama 5 hari, jadi kalau X terlambat 2 hari, proyek pasti mundur 2 hari". Kesimpulan itu keliru: jalur X punya kelonggaran 3 hari, sehingga keterlambatan 2 hari tidak memundurkan proyek.

● BEDAH SOAL PREDIKSI

Soal berikut menyajikan tabel kegiatan dengan satu kegiatan yang terlambat. Perhatikan bahwa keterlambatan di luar jalur kritis tidak otomatis memundurkan proyek, dan bahwa jalur kritis dapat berpindah (bagian 6).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Tim PKL membuat aplikasi perpustakaan untuk SMK Cendekia (2027) dengan rencana kegiatan pada tabel. Kegiatan yang tidak saling menunggu dikerjakan bersamaan oleh anggota yang berbeda.

Kode	Kegiatan	Lama (hari kerja)	Dimulai setelah
K1	Analisis kebutuhan	3	–
K2	Desain antarmuka	4	K1 selesai
K3	Desain basis data	2	K1 selesai
K4	Pemrograman	6	K2 dan K3 selesai
K5	Pengujian	3	K4 selesai
K6	Pelatihan pustakawan	2	K4 selesai

Ternyata desain basis data (K3) terlambat 3 hari dari rencana, sedangkan kegiatan lain sesuai rencana.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Seluruh proyek paling cepat selesai pada akhir hari kerja ke- ....

A. 16
B. 17
C. 19
D. 20
E. 23

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Pembahasan **Kunci: B**

Konsep: Pada jadwal proyek, kegiatan yang menunggu beberapa kegiatan lain baru dapat dimulai setelah kegiatan yang paling akhir selesai. Keterlambatan hanya memundurkan proyek sebesar bagian yang melampaui kegiatan sejajarnya.

Langkah:

1. K1 selesai hari ke-3.
2. K2 selesai 3 + 4 = hari ke-7; K3 (terlambat) selesai 3 + 2 + 3 = hari ke-8.
3. K4 menunggu K2 dan K3 → mulai setelah hari ke-8, selesai 8 + 6 = hari ke-14.
4. K5 selesai 14 + 3 = 17; K6 selesai 14 + 2 = 16. Proyek selesai hari ke-17.
5. Cek: tanpa keterlambatan proyek selesai hari ke-16, jadi keterlambatan 3 hari hanya memundurkan 1 hari.

Analisis setiap opsi:

- **A.** ✗ Jebakan: K3 dianggap tidak memengaruhi jadwal karena lebih pendek daripada K2; padahal setelah terlambat K3 selesai hari ke-8, sehari setelah K2.
- **B.** ✓ K4 mulai setelah K3 selesai hari ke-8, sehingga K5 selesai hari ke-17.
- **C.** ✗ Jebakan: seluruh keterlambatan 3 hari ditambahkan ke jadwal awal 16 hari.
- **D.** ✗ Semua lama kegiatan dijumlahkan (3 + 4 + 2 + 6 + 3 + 2), seolah-olah tidak ada yang dikerjakan bersamaan.
- **E.** ✗ Jumlah semua kegiatan ditambah keterlambatan: 20 + 3.

Jawaban: B.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk soal jadwal, tulis semua jalur dan jumlahnya. Untuk soal keputusan, tanyakan: apakah tindakan itu melibatkan klien atau seluruh tim, dan apakah memperbaiki proses?

1. Tim PKL beranggotakan empat siswa membangun aplikasi pemesanan untuk Toko Kue Bu Rina. Dokumen ruang lingkup yang disepakati berisi katalog kue, keranjang belanja, dan pengiriman pesanan lewat WhatsApp, dengan jadwal 6 minggu. Pada minggu ke-4 tahun 2027, saat pemrograman baru selesai sebagian, Bu Rina meminta tambahan pembayaran QRIS dan tetap ingin aplikasi rilis pada akhir minggu ke-6.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan ketua tim yang paling tepat adalah ....

A. Mengerjakan QRIS dengan lembur setiap malam agar Bu Rina puas dan jadwal tetap
B. Menolak permintaan QRIS karena tidak tercantum dalam dokumen ruang lingkup awal
C. Mengurangi waktu pengujian agar fitur QRIS dapat masuk dalam jadwal 6 minggu
D. Menunda pembahasan QRIS sampai aplikasi rilis, lalu membicarakannya dengan Bu Rina
E. Mencatat permintaan, menghitung dampaknya, lalu memutuskannya bersama Bu Rina

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Tim pengembang aplikasi pendaftaran koperasi sekolah bekerja dalam sprint dua mingguan (2027). Aplikasi wajib rilis pada akhir sprint ke-5 karena akan dipakai saat penerimaan siswa baru. Laporan pemantauan setelah sprint ke-3:

Sprint	Rencana (poin tugas)	Realisasi (poin tugas)
1	30	30
2	30	22
3	30	21

Sisa pekerjaan 77 poin: 40 poin fitur wajib (pendaftaran dan pembayaran) dan 37 poin fitur tambahan (tema warna dan statistik).
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keputusan ketua tim yang paling tepat adalah ....

A. Menambah tiga programmer baru pada sprint ke-5 agar 77 poin dapat selesai
B. Mengurangi pengujian di dua sprint terakhir agar semua fitur dapat masuk
C. Tetap memakai rencana awal karena realisasi diperkirakan naik dengan sendirinya
D. Menyepakati dengan klien bahwa fitur wajib didahulukan, fitur tambahan menyusul
E. Memindahkan tanggal rilis ke sprint ke-7 tanpa membicarakannya dengan sekolah

3. Tim pengembang aplikasi presensi SMK Nusa Bangsa menetapkan "definisi selesai": setiap fitur baru harus ditinjau rekan (code review), lolos uji otomatis, dan dicoba di ponsel Android versi lama sebelum dirilis. Pada rilis versi 1.2 bulan Maret 2027, tombol presensi tidak muncul di ponsel beberapa guru yang memakai Android versi lama. Setelah ditelusuri, uji di ponsel Android lama dilewati karena tim dikejar jadwal rilis.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Masalah terjadi karena salah satu kriteria dalam definisi selesai tidak dijalankan sebelum rilis.	○	○
Agar rilis berikutnya lebih cepat, kriteria uji di ponsel Android lama sebaiknya dihapus dari definisi selesai.	○	○
Setelah memperbaiki tombol, tim sebaiknya menambah kasus uji untuk tombol presensi agar kesalahan serupa terdeteksi sebelum rilis.	○	○

4. Sebelum diserahkan ke sekolah, setiap halaman aplikasi diperiksa terhadap kriteria penerimaan yang disepakati dengan klien (2027). Kriteria untuk halaman Masuk:
(1) Kolom kata sandi menampilkan titik-titik dan memiliki ikon mata untuk melihat isinya.
(2) Ada tautan "Lupa kata sandi?" di bawah kolom kata sandi.
(3) Setiap kolom isian memiliki label di atas kolom, bukan hanya teks contoh di dalam

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

kolom.
(4) Tombol "Masuk" berada di bawah kedua kolom dan selebar kolom isian.
Hasil kerja tim ditunjukkan pada gambar.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kesimpulan pemeriksaan yang tepat adalah ....
(Jawaban benar lebih dari satu.)

Wireframe halaman Masuk — aplikasi Perpustakaan SMK Cendekia, versi 0.3



Pilih semua jawaban yang benar.

☐ A. Kriteria (1) sudah terpenuhi oleh rancangan pada gambar.
☐ B. Kriteria (2) sudah terpenuhi karena ada tautan di bawah formulir.
☐ C. Kriteria (3) belum terpenuhi pada kolom kata sandi.
☐ D. Kriteria (4) sudah terpenuhi oleh rancangan pada gambar.
☐ E. Halaman boleh diserahkan sekarang, lalu kekurangannya diperbaiki di versi berikutnya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. Tim aplikasi kasir koperasi (2027) sepakat memakai pedoman penulisan kode: nama variabel dan fungsi deskriptif dengan huruf kecil dan garis bawah, setiap fungsi diberi keterangan singkat, dan setiap perubahan ditinjau satu rekan sebelum digabung. Peninjau menemukan fungsi bernama `h` dengan variabel `x1` dan `x2` tanpa keterangan apa pun. Penulisnya menjawab, "Kodenya sudah berjalan benar, jadi tidak perlu diubah."
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Kode yang sudah berjalan benar tidak perlu disesuaikan dengan pedoman karena hasilnya tetap sama.	○	○
Mengganti nama h menjadi hitung_diskon dan x1 menjadi harga memudahkan anggota tim lain merawat kode.	○	○
Peninjauan rekan membantu menemukan kesalahan dan menjaga semua kode mengikuti pedoman yang sama.	○	○

Kunci singkat: 1. E 2. D 3. 1B 2S 3B 4. A, C, D 5. 1S 2B 3B

✔ PERIKSA DIRI

- Mengapa lama proyek tidak sama dengan jumlah semua lama kegiatan, dan apa yang membuat sebuah kegiatan "kritis"?
- Apa yang seharusnya dilakukan tim ketika klien meminta fitur baru di tengah proyek, dan mengapa menerima semuanya dengan lembur bukan jawaban yang tepat?
- **Yang harus kuingat:** lima fase proyek; waterfall untuk kebutuhan pasti, agile untuk kebutuhan berubah; jalur kritis = jalur terpanjang; budaya mutu = proses bersama (kriteria penerimaan, definisi selesai, tinjauan kode, PDCA), bukan pemeriksaan satu orang.

Latihan interaktif di LMS

Sub bab 1.2: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-1.2

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kuis Akhir Bab 1

SMKPPLG-KB1

Jumlah soal: 25 **Waktu anjuran:** 60 menit

Kerjakan sendiri dan berwaktu. Setelah selesai, cocokkan dengan kunci di Bagian D, isi tabel skor, lalu lihat anjuran di bawahnya.

Kerjakan Kuis Bab 1 di LMS

Skor dan progresmu tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-KB1

1. Mitra Digital, perusahaan aplikasi toko daring (2027), memiliki seorang staf dengan tugas harian berikut: menjadwalkan salinan cadangan basis data pelanggan setiap malam, memantau kueri yang berjalan lambat lalu memperbaiki kecepatan aksesnya, mengatur siapa yang boleh membaca atau mengubah tabel tertentu, dan menguji bahwa cadangan benar-benar dapat dipulihkan. Staf itu tidak menulis fitur aplikasi.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Profesi yang paling sesuai dengan tugas tersebut adalah ....

A. Pengembang aplikasi sisi server
B. Analis data bisnis
C. Administrator sistem operasi server
D. Administrator basis data
E. Penguji keamanan aplikasi

2. Kantor desa meminta sebuah studio kecil membuat aplikasi antrean layanan surat (2027). Pekan ini tim belum menyusun jadwal dan belum menulis kode. Mereka hanya menilai apakah proyek layak dikerjakan, mencatat pihak yang berkepentingan (kepala desa, petugas loket, warga), dan menyepakati piagam proyek yang memuat tujuan serta batas wewenang.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Fase manajemen proyek yang sedang dikerjakan tim adalah ....

A. Pelaksanaan
B. Inisiasi
C. Perencanaan
D. Pemantauan dan pengendalian
E. Penutupan

3. Tim pembuat aplikasi posyandu digital (2027) beranggotakan empat orang. Alya mewawancarai kader posyandu lalu menuliskan kebutuhan fitur. Bram menulis kode

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

tampilan dan interaksi di sisi ponsel. Citra menyusun tabel data balita dan mengatur siapa yang boleh mengubahnya. Dani menjalankan semua fitur pada beberapa ponsel dan melaporkan setiap kesalahan yang ia temukan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang sesuai dengan pembagian peran mereka adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Alya berperan sebagai analis sistem karena ia menggali dan menuliskan kebutuhan fitur
☐ B. Bram berperan sebagai administrator basis data karena ia menulis kode yang memakai data
☐ C. Citra berperan sebagai administrator basis data karena ia mengelola tabel dan hak akses data
☐ D. Dani berperan sebagai QA tester dan karena itu juga yang harus memperbaiki kode yang salah
☐ E. Kesalahan yang dilaporkan Dani sebaiknya diteruskan kepada pembuat bagian itu untuk diperbaiki

4. Dua alumni SMK PPLG menyusun Business Model Canvas untuk aplikasi "Tukang Terdekat" (2027), yang menghubungkan warga dengan tukang bangunan di kota mereka. Mereka menuliskan beberapa catatan, lalu memasukkannya ke blok yang sesuai pada kanvas.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pemasangan catatan dengan blok kanvas yang tepat adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Warga perkotaan yang sulit menemukan tukang tepercaya termasuk blok segmen pelanggan
☐ B. Biaya sewa server dan honor para pengembang setiap bulan termasuk blok sumber pendapatan
☐ C. Komisi 10 persen dari setiap pesanan yang selesai termasuk blok sumber pendapatan
☐ D. Pencarian dan pemesanan tukang lewat ponsel dalam lima menit termasuk blok proposisi nilai
☐ E. Iklan berbayar di media sosial untuk mengajak warga mengunduh aplikasi termasuk blok kegiatan utama

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. Tim pengembang aplikasi pemesanan untuk jaringan toko roti (2027) membagi pekerjaan seperti pada tabel berikut.

Anggota	Tugas yang dikerjakan
Eka	menjadwalkan cadangan basis data pelanggan dan mengatur siapa yang boleh mengubah tabel
Farhan	menulis kode server yang menyediakan data pesanan bagi aplikasi
Gita	mewawancarai pemilik toko lalu menuliskan kebutuhan fitur aplikasi
Hadi	menyiapkan server awan dan mengemas versi baru untuk dirilis
Indah	merancang alur layar dan mengamati calon pengguna mencobanya

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pasangan anggota dan profesi yang tepat adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Eka sebagai administrator basis data
☐ B. Farhan sebagai back-end developer
☐ C. Gita sebagai penguji perangkat lunak
☐ D. Hadi sebagai DevOps engineer
☐ E. Indah sebagai analis sistem

6. Rudi, lulusan SMK PPLG, mengerjakan pembuatan toko daring untuk seorang klien sendirian (2027). Ia menggambar rancangan tampilan, menulis kode tampilan dan server, memasangnya di layanan hosting, lalu mencoba semua fitur dan mencatat kesalahan yang ia temukan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Selama proyek itu, Rudi memegang beberapa peran sekaligus, misalnya perancang tampilan, developer, dan penguji.	○	○
Pekerjaan menguji tetap merupakan tugas yang berbeda dari menulis kode walaupun dikerjakan oleh orang yang sama.	○	○
Karena Rudi mampu mengerjakan tampilan dan server, ia otomatis disebut administrator basis data.	○	○

7. Studio Dua Langkah (2027) menjual dua produk: paket situs dan lisensi aplikasi kasir. Data bulan ini ditunjukkan pada tabel berikut. Biaya tetap bulan ini Rp8.200.000.

Produk	Harga per unit	Biaya variabel per unit	Terjual
Paket situs	Rp1.500.000	Rp400.000	4
Lisensi aplikasi kasir	Rp300.000	Rp50.000	20

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Hasil usaha studio bulan ini adalah ....

A. laba Rp3.800.000
B. laba Rp1.200.000
C. rugi Rp3.800.000
D. laba Rp9.400.000
E. rugi Rp1.200.000

8. Pada minggu ke-3 dari proyek enam minggu pembuatan aplikasi inventaris sebuah pesantren (2027), programmer inti tim jatuh sakit dan diperkirakan tidak dapat bekerja selama dua minggu. Jadwal rilis sudah disepakati dengan klien.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang tepat dilakukan ketua tim adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Menilai dampaknya pada kegiatan di jalur kritis lalu menyampaikannya kepada klien
☐ B. Menambah jam kerja anggota lain secara diam-diam agar jadwal tidak perlu dibicarakan
☐ C. Menghapus tahap pengujian akhir supaya sisa waktu tetap cukup sampai tanggal rilis
☐ D. Memindahkan sebagian tugas atau menunda fitur kurang penting setelah disepakati klien
☐ E. Memperbarui rencana dan jadwal agar sesuai dengan kondisi tim yang terbaru

9. Kursus daring Sinar Ilmu (2027) memiliki biaya tetap Rp3.600.000 per bulan (platform dan honor pengajar) dan biaya variabel Rp20.000 per peserta (modul cetak dan sertifikat). Harga kursus semula Rp80.000 per peserta. Untuk menarik lebih banyak peserta, pengelola menurunkan harga menjadi Rp70.000 per peserta.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Agar tidak rugi pada harga yang baru, jumlah peserta minimum per bulan adalah ....

A. 52 peserta
B. 60 peserta
C. 72 peserta
D. 90 peserta
E. 180 peserta

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

10. Dinas desa memesan aplikasi pelayanan surat kepada sebuah studio (2027). Format surat dan aturan penomoran sudah ditetapkan peraturan dan tidak akan berubah. Namun, tampilan dan alur layar belum jelas, dan dinas ingin warga mencobanya lebih dulu lalu memberi masukan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pendekatan pengembangan yang paling tepat adalah ....

A. Memakai waterfall untuk seluruh proyek karena sebagian aturannya sudah baku dan tidak berubah
B. Memakai agile untuk seluruh proyek dan tidak mencatat aturan apa pun selama pengerjaan
C. Mengerjakan aturan baku berurutan dan tampilan lewat putaran pendek yang diuji warga
D. Menunda proyek sampai tampilan dan alur layar seluruhnya diputuskan dan disetujui dinas
E. Menyerahkan penentuan tampilan kepada programmer tanpa melibatkan warga sebagai pengguna

11. Studio aplikasi Nusa Digital (2027) ingin merilis pembaruan aplikasi setiap Jumat tanpa mengemas dan mengunggahnya secara manual. Seorang anggota tim menyiapkan jalur otomatis: kode yang lolos uji dikemas dan dipasang ke server, kondisi server dipantau, dan ia mendapat pemberitahuan bila layanan mati.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Profesi yang paling sesuai dengan tugas anggota tim tersebut adalah ....

A. DevOps engineer yang mengelola jalur rilis dan server
B. Front-end developer yang menulis kode tampilan halaman
C. UI/UX designer yang merancang alur layar pemakaian
D. Analis sistem yang menggali kebutuhan dari klien
E. Technical writer yang menyusun panduan pemakaian

12. Studio Jendela Web menerima pesanan situs profil sebuah sekolah (2027). Rincian biaya satu situs ditunjukkan pada tabel berikut.

Komponen biaya	Besar
Honor desainer	Rp700.000
Honor programmer	Rp1.500.000
Domain dan hosting setahun	Rp300.000
Foto dan konten	Rp500.000

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Studio menetapkan laba sebesar 30 % dari **biaya** pembuatan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Harga jual situs tersebut adalah ....

A. Rp4.285.714
B. Rp3.600.000
C. Rp3.030.000
D. Rp3.900.000
E. Rp3.300.000

13. Tim PKL menyusun jadwal pembuatan aplikasi kasir koperasi (2027). Kegiatan, lama pengerjaan, dan urutannya ditunjukkan pada tabel berikut. Kegiatan yang tidak saling menunggu dikerjakan bersamaan oleh orang yang berbeda.

Kegiatan	Lama (hari)	Dikerjakan setelah
A. Analisis kebutuhan	3	–
B. Rancang basis data	4	A
C. Rancang tampilan	2	A
D. Pemrograman fitur	5	B
E. Pembuatan antarmuka	3	C
F. Pengujian gabungan	2	D dan E

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Seorang anggota mengusulkan menambah tenaga agar kegiatan E selesai 1 hari lebih cepat supaya aplikasi dirilis lebih awal. Penilaian yang paling tepat adalah ....

A. Efektif, karena E adalah kegiatan terakhir sebelum F sehingga E menentukan kapan F dapat dimulai
B. Efektif, karena lama proyek adalah jumlah semua kegiatan sehingga setiap kegiatan yang dipercepat memangkas total
C. Tidak efektif, karena jalur A–C–E–F hanya 10 hari, sedangkan jalur A–B–D–F selama 14 hari yang menentukan rilis
D. Tidak efektif, karena hanya kegiatan terpanjang di seluruh proyek, yaitu D, yang menentukan lama proyek
E. Tidak efektif, karena kegiatan yang dipercepat seharusnya yang paling awal, yaitu A, agar semua kegiatan maju

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

14. Seorang peserta magang menyusun bagan alur kerja proyek aplikasi di sebuah studio (2027), seperti pada gambar. Pembimbingnya mengatakan bahwa dua profesi pada bagan tertukar tempatnya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perbaikan yang tepat agar bagan itu benar adalah ....



A. menukar profesi pada tahap Analisis dan tahap Rancangan
B. menukar profesi pada tahap Rancangan dan tahap Pembuatan
C. menukar profesi pada tahap Pengujian dan tahap Rilis
D. menukar profesi pada tahap Pembuatan dan tahap Pengujian
E. menukar profesi pada tahap Analisis dan tahap Rilis

15. Dua alumni SMK PPLG membuka usaha jasa undangan digital bernama Kode Kampung (2027). Rincian biaya dan harga ditunjukkan pada tabel berikut.

Komponen	Besar
Biaya tetap per bulan (internet dan sewa ruang)	Rp1.200.000
Biaya variabel per undangan (lisensi desain)	Rp15.000
Harga jual per undangan	Rp75.000

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	**Benar**	**Salah**
Titik impas tercapai ketika terjual 20 undangan dalam sebulan.	○	○
Jika terjual 30 undangan dalam sebulan, laba usahanya Rp1.800.000.	○	○
Jika biaya lisensi naik menjadi Rp25.000 per undangan dan harga jual tetap, titik impas menjadi 24 undangan.	○	○

16. Gim teka-teki Bunga Nusa (2027) dapat diunduh dan dimainkan tanpa membayar. Pemain yang ingin tampil berbeda dapat membeli kostum karakter di dalam gim, dan pemain yang kehabisan koin dapat membeli paket koin tambahan. Studio tidak menjual gim itu per salinan dan tidak menampilkan iklan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Model pendapatan yang dipakai studio adalah ....

A. pembelian dalam aplikasi
B. langganan bulanan untuk semua pemain
C. jual putus gim per salinan unduhan
D. penayangan iklan kepada pemain
E. jasa pembuatan gim pesanan klien

17. Tim aplikasi pemesanan lapangan futsal (2027) menetapkan definisi selesai: kode ditinjau rekan, uji otomatis lolos, dan dokumentasi fitur diperbarui. Salah satu kriteria penerimaan fitur "cari lapangan" yang ditulis klien berbunyi: "Pencarian harus cepat dan terlihat bagus."
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Fitur yang sudah lolos uji otomatis tetapi belum ditinjau rekan sudah memenuhi definisi selesai.	○	○
Kriteria "cepat dan terlihat bagus" kurang baik karena tidak dapat diperiksa dengan jawaban ya atau tidak.	○	○
Kriteria itu dapat diperbaiki dengan menyebut batas waktu, misalnya hasil tampil paling lama dua detik.	○	○

18. Anggota tim sebuah studio aplikasi (2027) menulis kode dengan gaya yang berbeda-beda: sebagian memakai nama variabel satu huruf, sebagian memakai huruf besar dan kecil bercampur. Akibatnya, rekan sulit membaca kode satu sama lain dan sering terjadi salah paham saat memperbaiki fitur.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang sejalan dengan prinsip budaya mutu adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Membiarkan setiap anggota memakai gayanya sendiri karena kodenya tetap berjalan
☐ B. Menyepakati pedoman penulisan kode tertulis, misalnya nama deskriptif dengan huruf kecil dan garis bawah
☐ C. Memeriksa kepatuhan pada pedoman saat tinjauan kode sebelum perubahan digabungkan
☐ D. Menjatuhkan sanksi kepada anggota yang gaya penulisannya berbeda dari mayoritas
☐ E. Menjalankan alat pemeriksa gaya penulisan otomatis sebelum perubahan digabungkan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

19. Tim PKL menyusun rencana pembuatan aplikasi parkir kampus (2027) dalam bentuk jaringan kerja seperti pada gambar; angka pada kotak adalah lama kegiatan. Kegiatan yang tidak saling menunggu dikerjakan bersamaan. Karena tenggat dimajukan, tim menambah dua programmer sehingga kegiatan S selesai dalam 2 hari, bukan 5 hari. Kegiatan lain tidak berubah.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Lama proyek paling cepat sekarang adalah ... hari.



A. 11
B. 13
C. 14
D. 12
E. 10

20. Tim scrum merencanakan sprint dua pekan untuk aplikasi toko daring (2027). Product backlog yang sudah diurutkan product owner dari prioritas tertinggi adalah: login (8 poin), keranjang belanja (13 poin), pembayaran (13 poin), ulasan produk (5 poin), dan daftar keinginan (3 poin). Dari dua sprint sebelumnya, kecepatan tim rata-rata 21 poin per sprint.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang perencanaan sprint ini adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Login dan keranjang belanja (21 poin) dapat dimasukkan ke sprint ini sesuai urutan prioritas
☐ B. Daftar keinginan sebaiknya dimasukkan lebih dulu karena poinnya paling kecil
☐ C. Memasukkan pembayaran bersama login dan keranjang (34 poin) melebihi kecepatan tim
☐ D. Product owner yang bertanggung jawab mengurutkan prioritas isi product backlog
☐ E. Bila 21 poin selesai, tim wajib menambah 13 poin lagi tanpa berdiskusi dengan product owner

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

21. Tim aplikasi penggajian (2027) menerapkan pedoman: nama variabel harus deskriptif, setiap fungsi memiliki keterangan singkat, dan setiap fungsi memiliki uji otomatis yang lolos. Hasil tinjauan kode empat fungsi yang diajukan ke cabang utama ditunjukkan pada tabel berikut.

Fungsi	Nama variabel	Keterangan fungsi	Uji otomatis
hitung_gaji	deskriptif	ada	lolos
cek	x dan y	ada	lolos
simpan_data	deskriptif	tidak ada	lolos
kirim_slip	deskriptif	ada	belum dibuat

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Fungsi yang boleh digabung ke cabang utama saat ini adalah ....

A. hitung_gaji saja
B. hitung_gaji dan cek
C. hitung_gaji dan simpan_data
D. hitung_gaji dan kirim_slip
E. keempat fungsi karena semuanya sudah berjalan

22. Dua hari sebelum rilis aplikasi pemesanan tiket wisata sebuah desa (2027), Bram menemukan bahwa fitur pembayaran belum pernah diuji di ponsel berlayar kecil, padahal uji itu tercantum dalam kriteria penerimaan yang disepakati dengan klien. Sebagian besar pengunjung desa memakai ponsel seperti itu. Ketua tim menawarkan beberapa keputusan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keputusan yang paling tepat adalah ....

A. merilis sesuai jadwal lalu menguji setelah ada keluhan pengguna di lapangan
B. mengubah kriteria uji layar kecil menjadi pilihan tambahan yang boleh dilewati
C. menguji fitur itu lebih dulu, lalu membicarakan penundaan bila hasilnya gagal
D. meminta klien mencoba sendiri fitur itu setelah aplikasi dirilis ke pengunjung
E. mengurangi pengujian fitur lain agar waktu untuk jadwal tetap cukup sampai rilis

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

23. Seusai rilis aplikasi jadwal piket sekolah (2027), tiga fitur ternyata tampil rusak di ponsel berlayar kecil karena tidak pernah diuji pada ukuran layar itu. Tim ingin memperbaiki cara kerjanya agar kejadian serupa tidak terulang.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang sejalan dengan prinsip budaya mutu adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Menambahkan uji tampilan pada beberapa ukuran layar ke dalam kriteria penerimaan setiap fitur
☐ B. Menunjuk satu anggota paling senior untuk menguji seluruh fitur sendirian menjelang rilis
☐ C. Mencatat penyebab masalah, memperbarui pedoman tim, lalu memeriksa hasilnya pada rilis berikutnya
☐ D. Memberi sanksi kepada pembuat fitur yang rusak agar semua anggota lebih berhati-hati
☐ E. Mewajibkan tinjauan kode oleh rekan satu tim sebelum setiap fitur digabungkan ke cabang utama

24. Tim aplikasi toko daring (2027) sedang menulis kriteria penerimaan untuk fitur "cari barang". Kriteria yang baik harus jelas dan dapat diperiksa penguji dengan jawaban ya atau tidak.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kriteria penerimaan yang paling baik adalah ....

A. Fitur pencarian harus berfungsi dengan baik dan terasa cepat saat dipakai pengguna
B. Mengetik kata kunci menampilkan barang yang memuat kata itu dalam waktu paling lama dua detik
C. Pengguna merasa puas dengan hasil pencarian setelah mencoba fitur itu beberapa kali
D. Hasil pencarian harus sesuai dengan harapan semua pengguna pada setiap kesempatan
E. Tampilan kolom pencarian harus dirancang indah oleh desainer sebelum fiturnya diuji

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

25. Sebuah studio menerima pesanan aplikasi dari klien (2027). Tim ingin menuliskan proses bisnisnya sebagai urutan kegiatan dari awal sampai setelah produk dipakai.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Urutan kegiatan yang paling tepat adalah ....

A. penawaran dan kesepakatan, permintaan klien, pengerjaan, uji, serah terima dan pembayaran, dukungan
B. permintaan klien, pengerjaan, penawaran dan kesepakatan, uji, serah terima dan pembayaran, dukungan
C. permintaan klien, penawaran dan kesepakatan, uji, pengerjaan, serah terima dan pembayaran, dukungan
D. permintaan klien, penawaran dan kesepakatan, pengerjaan, uji, dukungan, serah terima dan pembayaran
E. permintaan klien, penawaran dan kesepakatan, pengerjaan, uji, serah terima dan pembayaran, dukungan

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 1.1 · Profesi dan peluang usaha PPLG	1, 3, 4, 5, 6, 7, 9, 11, 12, 14, 15, 16	___ / 12	Ulangi hlm. 9
Sub bab 1.2 · Manajemen proyek dan budaya mutu	2, 8, 10, 13, 17, 18, 19, 20, 21, 22, 23, 24, 25	___ / 13	Ulangi hlm. 23

Skor 70 % atau lebih: lanjut ke bab berikutnya. Skor 50–69 %: ulangi sub bab pada tabel yang kurang dari separuh benar. Skor di bawah 50 %: baca ulang bab ini dari awal. (Angka ini anjuran belajar, bukan prediksi nilai TKA.)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bab 2 · Kecakapan Kerja Dasar, K3, dan Budaya Kerja

SMKPPLG-B2

Setelah mempelajari bab ini, kamu dapat:

- mengenali bahaya di lingkungan kerja teknologi informasi dan memilih pengendalian serta tindakan yang tepat
- bersikap profesional dan etis, termasuk saat menemukan kesalahan pada sistem yang sedang dipakai
- mengelola aset fisik dan digital, dari inventaris, penyusutan, sampai lisensi dan hak cipta
- mengelola berkas proyek, versi kode dengan Git, cadangan, dan dokumentasi dengan benar

Peta bab

Sub bab	Judul
2.1	K3LH dan budaya kerja profesional
2.2	Pengelolaan aset fisik dan digital
2.3	Pengelolaan file proyek, source code, dan dokumentasi

Bab 2 di LMS

Buka seluruh latihan bab ini di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-B2

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 2.1 · K3LH dan budaya kerja profesional

SMKPPLG-2.1

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menjelaskan prinsip K3LH** (keselamatan dan kesehatan kerja serta lingkungan hidup) dan **karakteristik budaya kerja profesional** di lingkungan kerja bidang PPLG. Levelnya pengetahuan dan pemahaman. Pada soal, kamu biasanya diberi satu kejadian di ruang kerja atau tim, lalu diminta memilih tindakan yang tepat.

Dua pola soal paling sering muncul. Pertama, **kasus bahaya di ruang kerja TI**: stopkontak bertumpuk, UPS berasap, layar terlalu rendah, atau limbah elektronik yang menumpuk. Kedua, **kasus sikap profesional**: kesalahan kode yang ditemukan setelah rilis, data pelanggan yang harus dijaga, atau kode klien lama yang ingin dipakai ulang untuk klien baru. Pilihan jawabannya sering terdengar praktis tetapi tidak profesional.

Setelah mempelajari sub bab ini, kamu dapat:

- menyebutkan bahaya khas di ruang kerja teknologi informasi dan cara mencegahnya;
- menilai risiko dan memilih pengendalian yang paling efektif;
- memilih tindakan yang tepat saat terjadi insiden listrik atau kebakaran kecil;
- menerapkan sikap kerja profesional: jujur, bertanggung jawab, menjaga kerahasiaan, menghormati hak cipta, dan melapor saat membuat kesalahan.

◆ KONSEP INTI

Orang sering mengira pekerjaan di depan komputer itu aman. Pada kenyataannya, ruang kerja TI punya bahaya sendiri: kabel yang kusut, listrik dari banyak perangkat, mata yang lelah, badan yang kaku, dan perangkat bekas yang menjadi limbah berbahaya. Sub bab ini membahas cara mengenali dan mengendalikannya, lalu sikap kerja yang membuat kamu dipercaya di dunia kerja.

1. Apa itu K3LH

K3LH adalah singkatan dari **keselamatan, kesehatan kerja, dan lingkungan hidup**. Tiga tujuannya:

- **Keselamatan kerja**: mencegah kecelakaan, misalnya korsleting atau kebakaran.
- **Kesehatan kerja**: mencegah penyakit akibat pekerjaan, misalnya nyeri leher dan mata lelah.
- **Lingkungan hidup**: mencegah pekerjaan merusak lingkungan, misalnya membuang baterai dan perangkat bekas sembarangan.

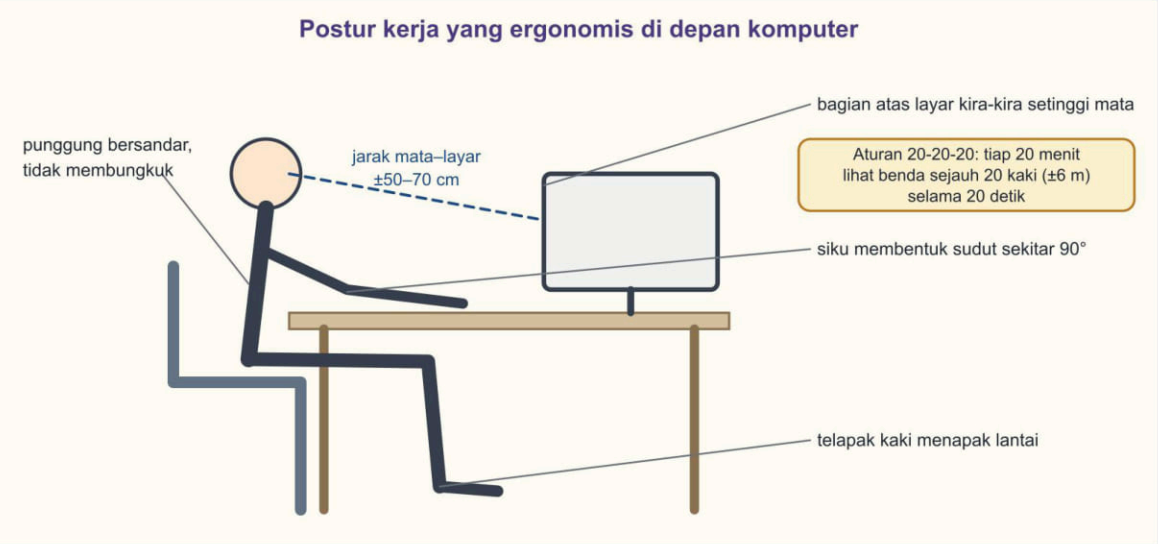
Prinsip dasarnya sederhana: **bahaya dikenali lebih dulu, lalu dicegah, dan bila terjadi,**

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

ditangani dengan benar dan dilaporkan.

2. Bahaya di ruang kerja teknologi informasi

a. Ergonomi dan kesehatan. Bekerja berjam-jam dengan posisi buruk menimbulkan nyeri leher, bahu, punggung, dan pergelangan tangan, serta mata lelah dan kering. Gambar 2.1 memperlihatkan postur yang dianjurkan: bagian atas layar kira-kira setinggi mata, jarak mata dengan layar sekitar 50–70 cm, punggung bersandar, siku sekitar 90°, dan telapak kaki menapak lantai. Untuk mata, dikenal **aturan 20-20-20**: setiap 20 menit, lihat benda sejauh sekitar 6 meter selama 20 detik. Berdiri dan bergerak sebentar secara berkala juga membantu.

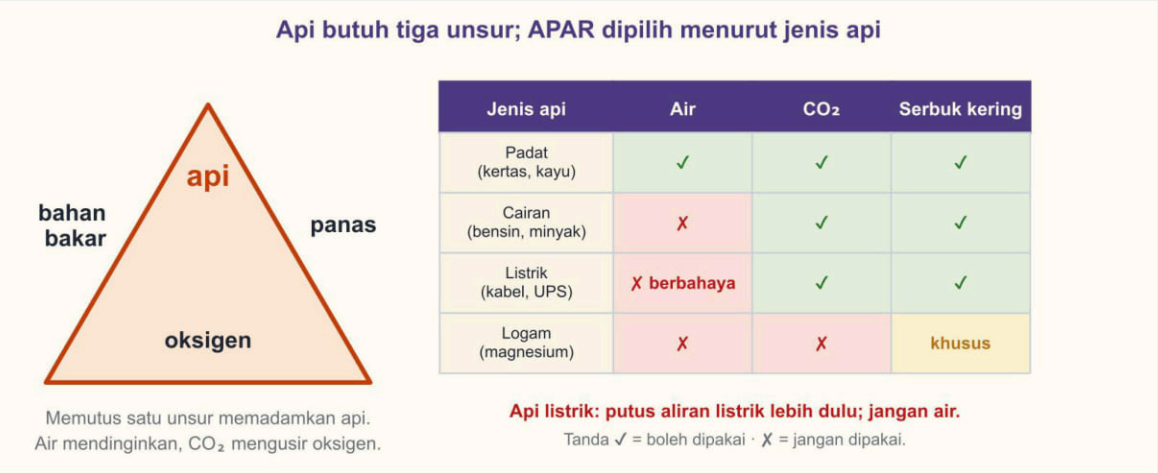


Gambar 2.1 Postur kerja yang ergonomis di depan komputer. Sumber: digambar oleh Tim Next Level Study Indonesia.

b. Listrik. Bahaya listrik meliputi kabel terkelupas, colokan yang kendur, dan beban berlebih karena **stopkontak bertumpuk** (satu stopkontak dipakai banyak perangkat lewat banyak sambungan). Beban berlebih membuat kabel panas dan dapat menyulut api. Pencegahan: gunakan kabel dan stopkontak berstandar, jangan menumpuk sambungan, rapikan kabel agar tidak terinjak, pastikan **grounding** (pentanahan) terpasang, dan pasang **UPS** serta pemutus arus untuk melindungi perangkat dari lonjakan.

c. Kebakaran. Api membutuhkan tiga unsur yang sering digambarkan sebagai **segitiga api**: bahan bakar, panas, dan oksigen (Gambar 2.2). Memutus salah satu unsur memadamkan api. **APAR** (alat pemadam api ringan) dipilih menurut jenis api. Pada api yang berasal dari peralatan listrik, **jangan memakai air**: air menghantarkan listrik sehingga berisiko menyetrum dan dapat memperluas korsleting. Pilih APAR **CO_2** atau **serbuk kering**, dan bila aman, **putus aliran listrik** lebih dulu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 2.2 Segitiga api dan pemilihan APAR menurut jenis api. Sumber: digambar oleh Tim Next Level Study Indonesia.

d. Limbah elektronik (e-waste). Monitor, laptop, dan komponen lama mengandung logam berat; **baterai** (termasuk baterai UPS) mengandung bahan kimia berbahaya dan dapat bocor atau terbakar bila rusak. Perangkat bekas tidak boleh dibuang ke tempat sampah biasa. Jalur yang benar adalah **menyerahkan ke pengelola limbah elektronik atau limbah berbahaya yang berizin**. Perangkat yang rusak atau baterai yang menggembung ditarik dari pemakaian, bukan dipinjamkan lagi.

Perhatikan bahwa kata "bekas" tidak berarti "tidak berbahaya". Baterai UPS yang sudah soak, misalnya, masih menyimpan muatan dan bahan kimia yang dapat mengalami korsleting bila tertekan atau basah. Karena itu perangkat bekas disimpan di tempat kering, terpisah dari bahan mudah terbakar, dan tidak ditumpuk di bawah meja atau di jalur jalan sambil menunggu diangkut.

e. Bahaya lain. Cairan di dekat peralatan (karena itu ada larangan membawa minuman ke ruang server), jalur kabel yang menghalangi jalan, barang yang menumpuk di jalur evakuasi, dan ruang server yang terlalu panas.

3. Menilai risiko dan memilih pengendalian

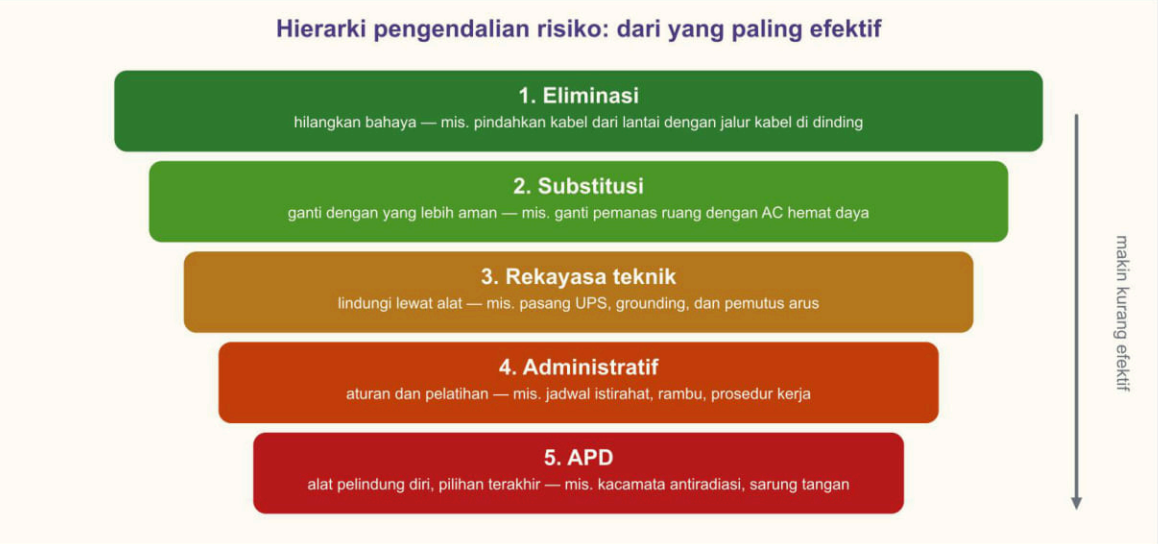
Tidak semua bahaya perlu diperbaiki lebih dulu. **Risiko** ditentukan oleh dua hal: **kemungkinan** bahaya terjadi dan **akibat** yang ditimbulkannya. Cara sederhana adalah memberi nilai 1 (rendah) sampai 3 (tinggi) untuk keduanya, lalu mengalikannya: **tingkat risiko = kemungkinan × akibat**. Risiko dengan nilai tertinggi ditangani lebih dulu.

Setelah risiko ditentukan, pengendaliannya dipilih menurut **hierarki pengendalian** (Gambar 2.3), dari yang paling efektif:

1. **Eliminasi**: menghilangkan bahayanya, misalnya memindahkan kabel dari lantai ke jalur kabel di dinding.
2. **Substitusi**: mengganti dengan yang lebih aman.
3. **Rekayasa teknik**: melindungi lewat alat, misalnya UPS, grounding, dan pemutus arus.
4. **Administratif**: aturan, pelatihan, rambu, jadwal istirahat, prosedur kerja.
5. **APD** (alat pelindung diri): pilihan terakhir, karena bahayanya tetap ada dan hanya

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

penggunanya yang dilindungi.



Gambar 2.3 Hierarki pengendalian risiko. Sumber: digambar oleh Tim Next Level Study Indonesia.

Jadi, bila ada kabel melintang di jalur jalan, cara terbaik bukan memasang tulisan "hati-hati" (administratif) atau menyuruh orang memakai sepatu khusus (APD), tetapi **menyingkirkan kabel itu dari jalur jalan** (eliminasi).

4. Rambu K3

Rambu keselamatan memakai warna dan bentuk yang seragam sehingga bisa dipahami sekilas (Gambar 2.4).

- **Larangan**: lingkaran merah dengan garis coret (dilarang membawa minuman).
- **Wajib**: lingkaran biru dengan simbol putih (wajib memakai gelang antistatik).
- **Peringatan**: segitiga kuning berbingkai hitam (bahaya listrik).
- **Kondisi aman/informasi**: persegi atau persegi panjang hijau (jalur evakuasi, titik kumpul, P3K).



Gambar 2.4 Empat jenis rambu K3 menurut warna dan bentuknya. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. Tindakan saat insiden listrik atau kebakaran kecil

Urutan berpikir yang aman saat ada asap atau percikan dari peralatan listrik:

1. **Utamakan keselamatan orang**: jauhkan orang dari area bahaya.
2. **Putus sumber listrik** bila dapat dijangkau dengan aman.
3. **Gunakan APAR yang sesuai** bila ada api kecil (CO_2 atau serbuk kering untuk api listrik); jangan air.
4. **Lapor** kepada guru, petugas, atau penanggung jawab ruang; bila api membesar, **evakuasi** lewat jalur yang ditandai.
5. Jangan membuka casing perangkat yang sedang berasap dan jangan menyentuh perangkat bertegangan dengan tangan basah.

Evakuasi dan pertolongan pertama. Setiap ruang kerja sebaiknya memiliki **jalur evakuasi** yang tidak terhalang, **titik kumpul** di luar gedung, dan **kotak P3K** (pertolongan pertama pada kecelakaan) yang isinya diperiksa berkala. Saat alarm berbunyi atau diperintahkan evakuasi, tinggalkan pekerjaan, jangan membawa barang besar, jangan memakai lift pada kebakaran, dan berkumpul di titik kumpul agar petugas dapat memastikan semua orang selamat. Tersengat listrik ringan atau luka bakar kecil tetap dilaporkan; jangan menyentuh korban yang masih menempel pada sumber listrik sebelum aliran listrik diputus.

Lingkungan hidup di kantor teknologi. Huruf "LH" pada K3LH mengingatkan bahwa pekerjaan digital juga meninggalkan jejak pada lingkungan: listrik untuk komputer dan server yang menyala sepanjang hari, kertas cetakan, dan limbah elektronik. Prinsip **3R** berlaku di sini: ***reduce*** (kurangi: matikan perangkat yang tidak dipakai, cetak seperlunya), ***reuse*** (pakai ulang: perangkat lama yang masih layak dipakai untuk keperluan lain), dan ***recycle*** (daur ulang lewat pengelola yang berizin).

6. Budaya kerja profesional

Selain aman, pekerja yang baik **dapat dipercaya**. Beberapa nilai pokok:

- **5R**: ringkas, rapi, resik, rawat, rajin. Ruang kerja yang tertata mengurangi kesalahan dan kecelakaan.
- **Disiplin dan tanggung jawab**: menepati jadwal, menyelesaikan tugas, dan bertanggung jawab atas hasil kerja sendiri.
- **Jujur**: mengakui kesalahan dan tidak mengklaim pekerjaan orang lain.
- **Menjaga kerahasiaan**: data pelanggan, kata sandi, kode klien, dan dokumen proyek tidak disebarkan tanpa izin. Pada data pribadi (nama, alamat, nilai), kerjakan dengan **salinan yang datanya disamarkan** bila memungkinkan dan jangan memindahkannya ke perangkat atau layanan pribadi.
- **Perjanjian kerahasiaan (NDA)**: banyak perusahaan dan klien meminta pekerjanya menandatangani perjanjian bahwa informasi proyek, data pelanggan, dan rancangan produk tidak boleh dibocorkan, bahkan setelah proyek selesai. Melanggarnya dapat berujung pada tuntutan, jadi tanyakan dulu bila ragu apakah sesuatu boleh dibagikan.
- **Menghormati hak cipta**: kode yang dibuat untuk klien tergantung perjanjian kerja; bila

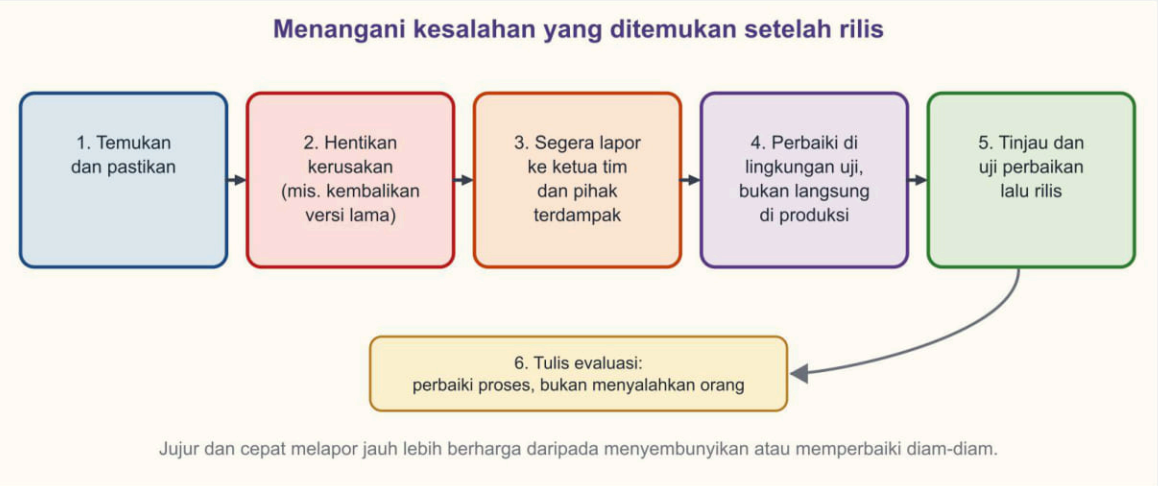
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

kontrak menyebut kode menjadi milik klien, kode itu tidak boleh disalin untuk klien lain. Tulis kode baru atau minta izin tertulis.

- **Menghormati kewenangan dan kerja tim**: tidak mengubah pekerjaan rekan tanpa memberi tahu, dan meminta bantuan lewat jalur yang tepat, bukan membagikan berkas rahasia ke forum publik.
- **Komunikasi yang terbuka**: menyampaikan hambatan lebih awal. Tim yang tahu bahwa sebuah fitur akan terlambat dua hari masih bisa mengatur ulang rencananya; tim yang baru tahu pada hari terakhir tidak bisa.
- **Terbuka terhadap kritik dan tinjauan**: hasil kerja ditinjau rekan untuk menyempurnakan produk, bukan untuk menilai pribadi seseorang. Pekerja profesional menerima masukan, menanyakan bagian yang belum jelas, dan memperbaikinya.
- **Tepat waktu dan menepati janji**: menyampaikan perkiraan yang jujur, bukan perkiraan yang menyenangkan tetapi tidak mungkin ditepati.
- **Menjaga nama baik pekerjaan**: tidak membicarakan kelemahan klien atau rekan di tempat yang tidak semestinya dan tidak memakai sumber daya kantor (komputer, internet, lisensi) untuk kepentingan pribadi tanpa izin.

7. Ketika kesalahan terjadi pada sistem yang sedang dipakai

Pada pekerjaan nyata, kesalahan akan terjadi. Yang membedakan pekerja profesional adalah **cara menanganinya** (Gambar 2.5).



Gambar 2.5 Alur menangani kesalahan yang ditemukan setelah rilis. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Pastikan** kesalahannya dan perkirakan dampaknya.
2. **Hentikan kerusakan**, misalnya mengembalikan versi lama atau menonaktifkan fitur yang bermasalah, dengan sepengetahuan tim.
3. **Segera laporkan** kepada ketua tim dan pihak yang terdampak. Menyembunyikan kesalahan memperbesar kerugiannya.
4. **Perbaiki di lingkungan uji**, bukan langsung di produksi, lalu **tinjau dan uji** sebelum dirilis.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. **Evaluasi tanpa menyalahkan**: cari penyebab dalam proses (misalnya uji yang kurang) supaya tidak terulang.

CONTOH TERSELESAIKAN

Contoh 1 — Menilai risiko tiga bahaya di lab

Lab komputer sebuah SMK (2027) memiliki tiga temuan. Beri nilai kemungkinan dan akibat (1 = rendah, 2 = sedang, 3 = tinggi), lalu tentukan urutan penanganan. *(Kasus fiktif.)*

Temuan	Kemungkinan	Akibat
P. Kabel ekstensi terkelupas dan melintang di lantai dekat pintu	3	3
Q. Kursi beberapa komputer terlalu rendah sehingga siswa menunduk	3	1
R. Tumpukan kardus menutup sebagian jalur evakuasi	1	3

1. Hitung tingkat risiko = kemungkinan × akibat. P = 3 × 3 = **9**. Q = 3 × 1 = **3**. R = 1 × 3 = **3**.
2. Urutkan dari yang tertinggi: P (9), lalu Q dan R (sama-sama 3).
3. Untuk P pilih pengendalian yang paling efektif: **eliminasi**, yaitu menyingkirkan kabel dari lantai dan menggantinya dengan kabel baru yang rapi; memasang tulisan peringatan saja tidak cukup.
4. Q dan R sama-sama bernilai 3, tetapi R menyangkut keselamatan saat darurat, jadi kardus dipindahkan **segera** (hampir tanpa biaya), sementara kursi dijadwalkan untuk disesuaikan.

Jawaban: urutan P → R → Q; P diatasi dengan eliminasi.

Contoh 2 — Percikan api di kabel ekstensi

Pada 2027, di ruang server kecil sebuah sekolah, kabel ekstensi yang memasok dua komputer memercikkan api kecil dan mulai berasap. Tersedia APAR serbuk kering, APAR CO_2, dan selang air. Seorang siswa berada paling dekat. Apa yang sebaiknya dilakukan? *(Kasus fiktif.)*

1. Tentukan jenis api: berasal dari **peralatan listrik**.
2. Tentukan alat yang boleh dipakai: serbuk kering atau CO_2 (tidak menghantarkan listrik). **Selang air tidak boleh.**
3. Susun urutan: bila saklar atau panel dapat dijangkau dengan aman, **putus aliran listrik** lebih dulu; bila api kecil, gunakan APAR; jauhkan orang dan **lapor** kepada guru atau petugas; bila api membesar, **evakuasi**.
4. Jangan membuka atau menyentuh kabel yang berasap dengan tangan.

Jawaban: putus listrik bila aman, padamkan dengan APAR serbuk kering atau CO_2, jauhkan orang, lalu lapor; jangan memakai air.

Contoh 3 — Diskon ganda ditemukan saat penjualan berlangsung

Sari adalah programmer di sebuah perusahaan toko daring (2027). Siang hari saat penjualan promo, ia sadar perubahan kode yang ia rilis pagi tadi membuat diskon terhitung dua kali

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

sehingga harga menjadi terlalu murah. Penjualan masih ramai dan ketua tim sedang rapat tetapi dapat dihubungi. Tindakan apa yang tepat? *(Kasus fiktif.)*

1. **Pastikan** kesalahan dan perkirakan dampak: setiap menit ada pesanan dengan harga salah.
2. **Hentikan kerusakan** dengan sepengetahuan tim: segera hubungi ketua tim, usulkan menonaktifkan promo atau mengembalikan versi sebelumnya.
3. **Lapor** jujur: apa yang salah, sejak kapan, dan perkiraan jumlah pesanan terdampak.
4. **Perbaiki di lingkungan uji**, lalu minta rekan meninjau sebelum dirilis.
5. **Evaluasi**: tambahkan uji perhitungan diskon agar kasus ini tertangkap sebelum rilis.

Jawaban: lapor segera, hentikan dampaknya bersama tim, perbaiki dengan peninjauan, lalu evaluasi proses. Memperbaiki diam-diam atau menunggu rapat selesai memperbesar kerugian.

Contoh 4 — Tangkapan layar data pelanggan untuk portofolio

Bima magang di sebuah studio aplikasi klinik (2027) dan ingin memajang tangkapan layar aplikasi buatannya di portofolio pribadi. Layar itu menampilkan nama dan nomor telepon pasien sungguhan. Apa yang tepat? *(Kasus fiktif.)*

1. Data pasien adalah **data pribadi** milik klinik dan pasien; Bima tidak berhak menyebarkannya.
2. Karya dan data klien juga terikat perjanjian kerja.
3. Pilihan yang benar: memakai **data contoh yang disamarkan**, dan meminta izin tertulis dari studio sebelum menampilkan karya.

Jawaban: samarkan atau ganti data dengan data contoh dan minta izin; jangan memajang data pasien asli.

❖ PETA KONSEP

Gambar 2.6 merangkum K3LH dan budaya kerja profesional.



Gambar 2.6 Peta konsep K3LH dan budaya kerja profesional. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Yang harus kamu ingat:

- **Bahaya ruang TI**: ergonomi (layar setinggi mata, aturan 20-20-20), listrik (jangan menumpuk stopkontak), api (jangan air pada api listrik), dan limbah elektronik (serahkan ke pengelola berizin).
- **Risiko = kemungkinan × akibat**; pengendalian terbaik adalah **eliminasi**, APD yang terakhir.
- Saat insiden: **selamatkan orang, putus listrik bila aman, APAR yang sesuai, lapor**.
- Profesional itu **jujur, melapor kesalahan, menjaga kerahasiaan, dan menghormati hak cipta**.
- Perbaikan kesalahan dilakukan **di lingkungan uji dan dengan sepengetahuan tim**, lalu dievaluasi tanpa menyalahkan.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal K3LH:

1. **Tentukan bahayanya**: ergonomi, listrik, api, atau limbah.
2. **Pilih tindakan yang menghilangkan atau mengendalikan sumber bahaya**, bukan yang hanya menyiasatinya.
3. **Untuk kebakaran listrik**, ingat urutan: orang aman → listrik diputus → APAR sesuai → lapor.

Langkah menjawab soal sikap profesional:

1. **Tanyakan**: siapa yang dirugikan bila tindakan ini dilakukan (klien, pengguna, tim)?
2. **Pilih tindakan yang jujur, terbuka, dan sepengetahuan tim.** Tindakan "diam-diam" hampir selalu salah.
3. Untuk kode atau data klien, pilih jalur **izin** atau **menulis ulang**, bukan menyalin.

Jebakan yang sering dipasang:

- **Air untuk api listrik** atau menyiram perangkat yang berasap.
- **Stopkontak bertumpuk dianggap hemat.**
- **Perangkat bekas dijual ke pengepul tanpa izin** atau dibuang ke tempat sampah umum.
- **Meminjamkan lagi perangkat yang rusak** (misalnya baterai menggembung).
- **Menyembunyikan atau memperbaiki kesalahan diam-diam** agar "tidak merepotkan".
- **Menyalin kode klien lama dengan mengganti nama variabel** dianggap hasil karya baru.
- **Mengunggah data pelanggan ke perangkat pribadi** agar bisa dikerjakan di rumah.
- **APD dianggap pengendalian utama** padahal paling akhir.

> Contoh salah tafsir: seorang siswa menyimpulkan bahwa menempelkan tulisan "Awas kabel!" di dekat kabel yang melintang sudah menyelesaikan risiko. Tulisan itu hanya pengendalian administratif. Kabelnya tetap ada dan orang tetap bisa tersandung; pengendalian yang lebih efektif adalah menyingkirkan atau merapikan kabel itu.

• BEDAH SOAL PREDIKSI

Soal berikut menyajikan kasus UPS berasap di lab. Perhatikan bagaimana tiap pilihan diuji dengan urutan "orang aman → listrik diputus → alat yang sesuai → lapor" (bagian 5).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Saat praktik di lab RPL pada Februari 2027, UPS yang memasok tiga komputer mengeluarkan bau hangus dan asap tipis. Belum terlihat api. Di dinding lab tersedia APAR jenis CO2, dan panel listrik lab berada di dekat pintu.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan yang tepat dilakukan adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Mematikan aliran listrik lab dari panel bila panel dapat dijangkau dengan aman.
☐ B. Menyiram UPS dengan air dari galon agar asap cepat hilang dan UPS mendingin.
☐ C. Membuka casing UPS untuk mencari komponen yang terbakar sebelum melapor.
☐ D. Memakai APAR CO2 bila api muncul, lalu segera melapor kepada guru atau petugas.
☐ E. Menyimpan semua pekerjaan dulu, lalu menunggu asap hilang sebelum melapor.

Pembahasan **Kunci: A, D**

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pada perangkat listrik yang berasap atau terbakar: putus sumber listrik bila aman, jangan memakai air, gunakan APAR yang sesuai (CO2 atau serbuk untuk kebakaran listrik), jauhkan orang, dan laporkan segera.

Analisis setiap pernyataan:
• **Pernyataan 1 ✓** Memutus listrik menghentikan pemanasan dan mencegah api muncul.
• **Pernyataan 2 ✗** Air menghantarkan listrik dan dapat menyebabkan sengatan listrik serta korsleting.
• **Pernyataan 3 ✗** Membuka perangkat yang masih bertegangan sangat berbahaya; pemeriksaan dilakukan teknisi setelah listrik aman.
• **Pernyataan 4 ✓** APAR CO2 aman untuk kebakaran peralatan listrik dan tidak meninggalkan residu.
• **Pernyataan 5 ✗** Jebakan: data memang penting, tetapi menunda tindakan saat ada asap membahayakan orang di lab.

Jawaban: pernyataan 1, 4.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap soal, tulis dulu apakah kasusnya tentang bahaya (K3LH) atau tentang sikap profesional, lalu pilih tindakan yang menghilangkan sumber bahaya atau yang jujur dan terbuka.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. Tim pengembang aplikasi di sebuah startup sering bekerja lebih dari 8 jam di depan laptop. Saat pemeriksaan K3 bulan Februari 2027, petugas menemukan stopkontak bertumpuk di bawah meja, monitor yang terlalu rendah sehingga pegawai menunduk, dan dus berisi baterai laptop bekas di gudang. Ketua tim lalu membuat daftar tindakan perbaikan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan yang sesuai dengan prinsip K3LH adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Menyambung tiga stopkontak secara bertumpuk agar semua laptop dapat diisi dari satu sumber.
☐ B. Mengatur tinggi monitor sehingga bagian atas layar kira-kira sejajar dengan mata.
☐ C. Memangku laptop di atas kasur lipat saat lembur agar posisi kerja lebih santai.
☐ D. Menjual baterai laptop bekas bersama kertas bekas kepada pengepul rongsok terdekat.
☐ E. Mengistirahatkan mata secara berkala dengan melihat benda jauh setelah lama menatap layar.

2. Sebuah studio aplikasi mengganti 15 monitor lama dan 6 baterai UPS yang sudah soak pada Januari 2027. Gudang kantor sempit, sehingga barang-barang itu harus segera dikeluarkan. Para pegawai mengusulkan beberapa cara.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Cara yang paling sesuai dengan prinsip K3LH adalah ....

A. Membuang monitor dan baterai ke tempat sampah umum di depan kantor
B. Menyimpan baterai UPS di bawah meja kerja sampai jadwal angkut sampah
C. Menyerahkan monitor ke pengelola berizin, sedangkan baterai dibuang ke TPS
D. Menjual baterai UPS ke pedagang keliling tanpa izin pengolahan limbah
E. Menyerahkan semuanya ke pengelola limbah elektronik berizin

3. Rina, programmer lepas, membuat aplikasi kasir untuk Toko Maju pada tahun 2026. Kontraknya menyebutkan bahwa seluruh kode sumber aplikasi menjadi milik Toko Maju. Pada Januari 2027, Toko Sejahtera meminta Rina membuat aplikasi kasir dengan fitur yang mirip dan memberi waktu satu bulan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Rina yang paling sesuai dengan etika kerja profesional adalah ....

A. Menyalin kode Toko Maju, lalu mengganti nama variabel dan warna tampilannya
B. Menulis kode baru untuk Toko Sejahtera tanpa menyalin kode milik Toko Maju
C. Menyalin kode Toko Maju karena Rina sendiri yang dahulu menulis kode itu
D. Menyalin kode Toko Maju, lalu meminta izin setelah aplikasi baru selesai
E. Menjual salinan kode Toko Maju kepada Toko Sejahtera dengan harga murah

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Tika, programmer junior di perusahaan pembuat aplikasi rapor digital, ditugasi memperbaiki kesalahan perhitungan nilai rata-rata pada Mei 2027. Basis data produksi aplikasi itu berisi nama, alamat, dan nilai 1.200 siswa. Tenggat perbaikan tinggal dua hari, dan modul perhitungan nilai ditulis oleh rekannya, Galih.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Tika yang sesuai dengan etika kerja profesional adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Mengunduh data nilai seluruh siswa ke flashdisk pribadi agar dapat dikerjakan di rumah.
☐ B. Menguji perbaikan memakai salinan basis data yang nama dan alamat siswanya disamarkan.
☐ C. Meminta bantuan di forum publik dengan melampirkan berkas konfigurasi aplikasi apa adanya.
☐ D. Mengubah modul milik Galih tanpa memberi tahunya agar perbaikan cepat selesai sebelum tenggat.
☐ E. Segera melapor kepada ketua tim bila tanpa sengaja mengubah nilai siswa di basis data.

5. Jumat sore (2027), Dimas, programmer di perusahaan pembuat aplikasi stok barang, menyadari bahwa perubahan kode yang ia unggah pagi tadi membuat jumlah stok di aplikasi milik toko klien tidak sesuai. Toko klien tetap buka sampai malam. Ketua tim sudah pulang, tetapi masih dapat dihubungi lewat telepon.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Dimas yang paling sesuai dengan etika kerja profesional adalah ....

A. Memperbaikinya sendiri Senin pagi sebelum toko buka, tanpa perlu melapor
B. Mengirim perbaikan langsung ke produksi malam itu tanpa peninjauan rekan
C. Segera melapor ke ketua tim, menjelaskan dampaknya, dan ikut memperbaiki
D. Membatalkan perubahannya diam-diam di produksi tanpa memberi tahu tim
E. Mematikan aplikasi klien tanpa pemberitahuan sampai ia sempat memperbaiki

Kunci singkat: 1. B, E 2. E 3. B 4. B, E 5. C

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa air tidak dipakai untuk memadamkan api pada peralatan listrik, dan alat apa yang dipakai?
- Mengapa menyembunyikan kesalahan kode yang sudah dirilis justru memperbesar kerugian?
- **Yang harus kuingat:** risiko = kemungkinan × akibat; eliminasi lebih efektif daripada APD; saat insiden listrik: orang aman, putus listrik, APAR yang sesuai, lapor; profesional itu jujur, melapor, menjaga rahasia, dan menghormati hak cipta.

Latihan interaktif di LMS

Sub bab 2.1: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-2.1

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 2.2 · Pengelolaan aset fisik dan digital

SMKPPLG-2.2

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menjelaskan dan menerapkan prinsip serta prosedur pengelolaan aset fisik dan aset digital** dalam skenario proyek. Cakupannya: perangkat kerja, berkas proyek, direktori, aset digital, kode sumber, media, basis data, dan dokumentasi. Sub bab ini membahas bagian aset itu sendiri (siapa memilikinya, sampai kapan berlaku, bagaimana dicatat, dan apa yang boleh dilakukan dengannya), sedangkan pengelolaan berkas proyek, Git, dan cadangan dibahas di sub bab 2.3.

Soal biasanya memberi **tabel**: daftar aset beserta masa berlaku, catatan peminjaman perangkat, atau lisensi sebuah aset kreatif. Kamu diminta menentukan tindak lanjut yang benar. Hitungannya sederhana; yang menentukan adalah ketelitian membaca baris yang bermasalah.

Setelah mempelajari sub bab ini, kamu dapat:

- membedakan aset fisik dan aset digital serta menyebut contohnya;
- menjelaskan daur hidup aset dan mencatat inventaris serta peminjaman;
- menghitung penyusutan aset dengan metode garis lurus;
- mengelola masa berlaku domain, sertifikat, dan lisensi, serta akun dan hak akses;
- menjelaskan hak cipta dan memilih tindakan yang sah terhadap aset berlisensi (proprietary, sumber terbuka, Creative Commons).

◆ KONSEP INTI

Aset adalah segala sesuatu yang bernilai bagi sebuah usaha atau organisasi dan perlu dijaga. Bagi studio aplikasi, aset bukan hanya komputer. Domain situs, lisensi perangkat lunak, kode sumber, dan basis data pelanggan pun aset, kadang jauh lebih bernilai daripada perangkat kerasnya. Tanpa pengelolaan yang baik, aset mudah hilang, kedaluwarsa, atau dipakai dengan cara yang melanggar hukum.

1. Aset fisik dan aset digital

Gambar 2.7 membagi aset menjadi dua kelompok.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 2.7 Aset fisik dan aset digital pada sebuah studio aplikasi. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Aset fisik** berwujud dan dapat disentuh: laptop dan komputer, ponsel dan tablet uji, monitor, UPS, router, switch, kabel, dan furnitur kerja. Aset fisik bisa rusak, hilang, dan menyusut nilainya karena pemakaian.
- **Aset digital** berupa data atau hak untuk memakai sesuatu: **domain** dan **hosting**, **sertifikat HTTPS**, **lisensi perangkat lunak**, **kode sumber**, **basis data**, **aset kreatif** (gambar, musik, font, model 3D), serta **akun dan hak akses** ke layanan. Aset digital bisa kedaluwarsa, bocor, atau tidak boleh dipakai bila lisensinya tidak sesuai.

2. Daur hidup aset

Setiap aset melalui lima tahap yang sama (Gambar 2.8).



Gambar 2.8 Daur hidup aset dari perencanaan sampai penghapusan. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Perencanaan**: menentukan kebutuhan dan anggaran.
2. **Pengadaan**: membeli atau menyewa, lalu **mencatat di inventaris**.
3. **Pemakaian**: aset dipakai atau dipinjamkan dengan penanggung jawab yang jelas.
4. **Pemeliharaan**: perawatan berkala, pembaruan, dan perpanjangan masa berlaku.
5. **Penghapusan**: aset yang tidak layak pakai dihapus dari pemakaian dan inventaris dengan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

cara yang aman.

3. Inventaris dan peminjaman perangkat

Inventaris adalah daftar aset yang memuat, minimal, **kode aset, nama, lokasi, kondisi, penanggung jawab, dan tanggal penting** (pembelian atau akhir garansi). Setiap aset diberi **label kode** agar mudah dicocokkan dengan catatan. Perangkat yang dipakai bergantian, seperti ponsel uji, dicatat juga dalam **catatan peminjaman**: siapa meminjam, tanggal pinjam, batas kembali, tanggal kembali, dan kondisi saat dikembalikan (Gambar 2.9).

Contoh catatan inventaris dan peminjaman perangkat uji

A. Inventaris aset

Kode	Aset	Lokasi	Kondisi	Penanggung jawab	Garansi s.d.
LPT-01	Laptop pengembang	Ruang tim	baik	Bu Wulan	Mar 2028
HP-02	Ponsel uji Android	Lemari uji	baterai menggembung	Pak Anto	habis
UPS-01	UPS server	Ruang server	baik	Pak Anto	Jun 2027

B. Catatan peminjaman

Kode	Peminjam	Tanggal pinjam	Batas kembali	Kembali	Kondisi saat kembali
LPT-01	Dewi	2 Mar 2027	5 Mar 2027	5 Mar	baik
HP-03	Rio	8 Mar 2027	10 Mar 2027	belum	–
HP-04	Sela	1 Mar 2027	4 Mar 2027	4 Mar	layar retak kecil (dicatat)

Tiap perubahan dicatat: siapa, kapan, dan kondisinya. Perangkat yang terlambat dihubungi; yang rusak ditarik atau diperbaiki.

Gambar 2.9 Contoh lembar inventaris dan catatan peminjaman perangkat uji. Sumber: digambar oleh Tim Next Level Study Indonesia.

Kode aset sebaiknya mengikuti pola tetap yang mudah dibaca, misalnya **jenis-nomor urut** (`LPT-01` untuk laptop pertama, `HP-02` untuk ponsel kedua, `UPS-01` untuk UPS pertama). Pola seperti ini membuat daftar mudah diurutkan, dan kode yang ditempel di perangkat dapat langsung dicocokkan dengan catatan. Ruang penyimpanan juga ikut dicatat: lemari uji, ruang server, atau ruang tim, supaya perangkat yang hilang tahu harus dicari dari mana.

Cara membaca catatan seperti ini untuk menentukan tindak lanjut:

- **Melewati batas kembali dan belum kembali** → hubungi peminjam.
- **Masih dalam batas peminjaman** → belum ada tindakan; jangan menarik perangkat yang sah dipinjam.
- **Rusak ringan tetapi berfungsi** → **catat kondisinya** supaya terpantau dan tidak dianggap kesalahan peminjam berikutnya.
- **Rusak berbahaya** (misalnya baterai menggembung) → **tarik dari peminjaman** dan tangani sebagai limbah berbahaya (lihat sub bab 2.1). Perangkat seperti itu tidak boleh dipinjamkan lagi, walaupun diisi penuh.

4. Pemeliharaan dan penghapusan yang aman

Pemeliharaan meliputi pembersihan berkala, pembaruan perangkat lunak dan firmware, pemeriksaan kondisi, dan perpanjangan masa berlaku. Garansi yang sudah habis **tidak otomatis**

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

berarti perangkat harus diganti; yang menentukan adalah kondisi dan kebutuhan.

Pemeliharaan juga mencakup **pencatatan hasilnya**: tanggal perawatan terakhir, kerusakan yang ditemukan, dan siapa yang mengerjakannya. Catatan ini berguna ketika garansi diklaim atau ketika tim memutuskan apakah sebuah perangkat masih layak.

Penghapusan aset harus aman, terutama untuk perangkat yang pernah menyimpan data. Pada tahap ini:

- **Hapus data dengan aman** sebelum perangkat dilepas. Menghapus berkas biasa atau memformat cepat tidak cukup karena data masih bisa dipulihkan; gunakan penghapusan menyeluruh atau enkripsi sebelumnya.
- **Serahkan perangkat bekas ke pengelola limbah elektronik berizin**, bukan ke tempat sampah umum atau pedagang tanpa izin.
- **Catat penghapusan** di inventaris supaya daftar aset tetap benar.

Kapan sebuah perangkat perlu diganti? Jangan memutuskan hanya dari satu tanda seperti garansi yang habis. Pertimbangkan sekaligus: **kondisi dan keamanan** (apakah berbahaya, misalnya baterai menggembung), **kebutuhan** (apakah masih mampu menjalankan pekerjaan), **biaya perbaikan dibandingkan biaya penggantian**, serta **dukungan pembaruan** (perangkat yang tidak lagi mendapat pembaruan keamanan berisiko dipakai untuk data penting). Perangkat dengan garansi habis tetapi masih sehat dan cukup untuk kebutuhan tidak harus diganti.

Beli atau sewa. Aset fisik dapat dibeli (menjadi milik, dicatat dan disusutkan) atau disewa (biaya rutin, pemeliharaan sering ditanggung penyewa). Pilihan bergantung pada kebutuhan jangka panjang dan modal yang tersedia; usaha rintisan yang modalnya terbatas sering menyewa server atau layanan cloud dibanding membeli perangkat sendiri.

5. Penyusutan aset: metode garis lurus

Nilai aset fisik turun seiring waktu. Cara paling sederhana menghitung penurunannya adalah **metode garis lurus**, yang membagi penurunan sama besar tiap tahun:

> Penyusutan per tahun = (harga perolehan − nilai sisa) ÷ umur manfaat

- **Harga perolehan**: harga saat aset dibeli.
- **Nilai sisa**: perkiraan nilai di akhir masa pakai.
- **Umur manfaat**: lama aset diperkirakan dapat dipakai (tahun).
- **Nilai buku** pada suatu tahun = harga perolehan − penyusutan yang sudah dikumpulkan.

6. Mengelola aset digital

Berbeda dengan aset fisik, aset digital tidak "rusak karena dipakai", tetapi tetap berisiko: **hilang** karena tidak dicadangkan, **bocor** karena akses yang longgar, atau **tidak sah dipakai** karena lisensi tidak sesuai. Pengelolaannya karena itu berfokus pada masa berlaku, akses, dan keabsahan pemakaian.

Masa berlaku. Banyak aset digital punya tanggal kedaluwarsa: **domain** (setahun atau lebih, lalu

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

harus diperpanjang), **hosting**, **sertifikat HTTPS** (bila lewat, peramban menampilkan peringatan kepada pengunjung), dan **lisensi berlangganan**. Praktik yang baik: **catat semua tanggal itu di satu kalender**, aktifkan perpanjangan otomatis bila ada, dan periksa secara berkala. Prioritas penanganan: yang paling dekat tenggatnya dan yang akibatnya paling besar bila lewat.

Lisensi dan jumlah pemakai. Lisensi perangkat lunak sering dihitung per **kursi (seat)**, yaitu per pengguna. Bila lisensi berisi 3 kursi tetapi dipakai 4 orang, studio **melanggar ketentuan lisensi**; solusinya menambah lisensi atau mengurangi pemakai, bukan membiarkan.

Bentuk lisensi perangkat lunak. Ada beberapa cara lisensi diberikan. **Lisensi sekali beli (perpetual)** memberi hak memakai versi tertentu tanpa batas waktu; **langganan** memberi hak selama pembayaran berjalan dan berhenti bila tidak diperpanjang. Menurut hitungannya, ada lisensi **per kursi** (per pengguna), **per perangkat**, dan **per organisasi** (tim atau kantor). Membaca jenis hitungan ini penting agar jumlah pemakai tidak melebihi hak yang dibeli.

Akun dan hak akses. Setiap orang memakai **akun sendiri**, tidak berbagi satu akun administrator. Akun terpisah membuat aktivitas dapat ditelusuri dan akses dapat dicabut ketika orang keluar dari tim. Lindungi akun penting dengan **kata sandi yang panjang dan unik** serta **verifikasi dua langkah** (kode tambahan selain kata sandi), terutama untuk akun domain, server, dan repositori yang memegang aset paling berharga. Berikan **hak akses seperlunya saja** (prinsip hak akses minimum): programmer yang hanya perlu membaca basis data tidak diberi hak menghapusnya.

7. Hak kekayaan intelektual: hak cipta dan lisensi

Hak kekayaan intelektual (HAKI) melindungi hasil karya. Karya digital seperti program, gambar, dan musik dilindungi **hak cipta** secara otomatis begitu diciptakan. Orang lain boleh memakainya hanya sesuai **lisensi** yang diberikan pencipta. Aturan dasarnya: **dapat diunduh dan dipakai gratis tidak sama dengan bebas dipakai untuk apa saja**.

Jenis lisensi yang perlu kamu kenal (Gambar 2.10):

- **Hak cipta penuh (proprietary)**: hanya sesuai perjanjian, misalnya perangkat lunak berbayar. Tidak boleh disalin atau diubah tanpa izin.
- **Sumber terbuka (open source)**, misalnya **MIT**: boleh dipakai, termasuk untuk produk berbayar, dan diubah, dengan menyertakan pemberitahuan lisensi.
- **Creative Commons (CC)**: keluarga lisensi untuk karya kreatif. **BY** berarti wajib mencantumkan nama pencipta. **NC (non-commercial)** berarti **tidak boleh dipakai untuk tujuan komersial**. **ND (no derivatives)** berarti tidak boleh diubah. **SA (share alike)** berarti hasil turunan harus memakai lisensi yang sama.
- **Domain publik / CC0**: bebas dipakai tanpa syarat.
- **Gratis (freeware)** dan **uji coba (trial)**: boleh dipakai sesuai syaratnya, yang sering membatasi pemakaian komersial atau lamanya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Apa yang boleh dilakukan menurut jenis lisensi?

Lisensi	Boleh dipakai untuk produk berbayar	Wajib mencantumkan nama pencipta	Boleh diubah
Hak cipta penuh / proprietary (berbayar)	sesuai ketentuan perjanjian	sesuai perjanjian	tidak, kecuali diizinkan
Sumber terbuka MIT	ya	ya (sertakan pemberitahuan lisensi)	ya
Creative Commons BY	ya	ya	ya
Creative Commons BY-NC	TIDAK (non-komersial)	ya	ya
Creative Commons BY-ND	ya	ya	TIDAK (tanpa turunan)
Domain publik / CC0	ya	tidak wajib	ya

Mencantumkan nama pencipta TIDAK menggantikan izin: lisensi NC tetap melarang produk berbayar.

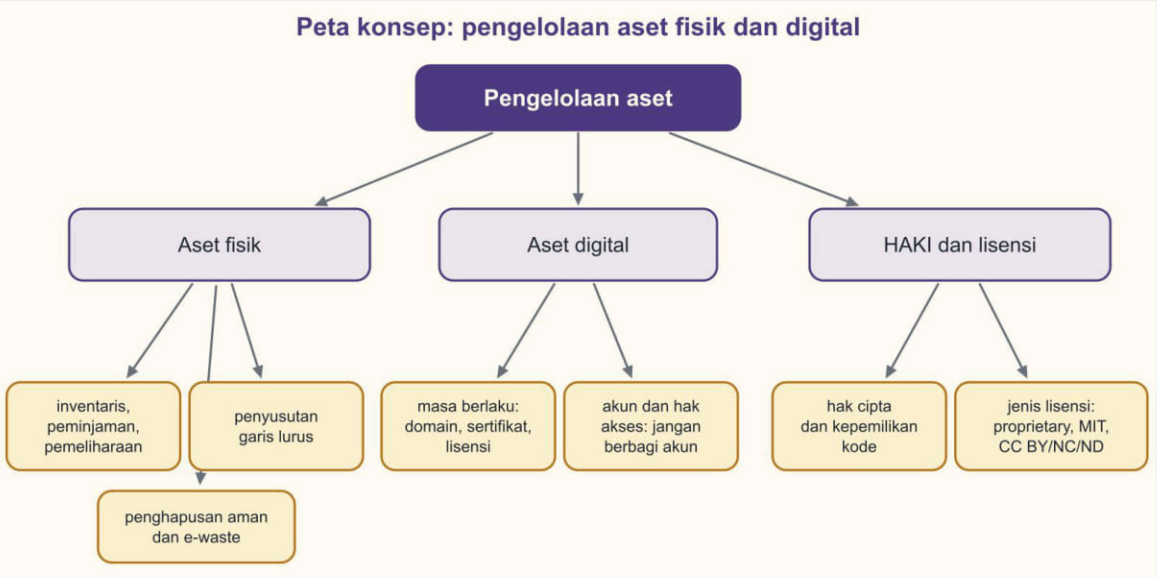
Gambar 2.10 Perbandingan apa yang boleh dilakukan menurut jenis lisensi. Sumber: digambar oleh Tim Next Level Study Indonesia.

Dua pelajaran penting. **Pertama**, mencantumkan nama pencipta tidak menggantikan izin: karya berlisensi NC tetap tidak boleh dipakai pada produk yang dijual walaupun namanya dicantumkan. **Kedua**, mengubah sedikit karya orang lain (mengganti nama variabel, mengubah warna, mengubah tempo musik) **tidak membuatnya menjadi karya sendiri**.

Akibat melanggar lisensi. Pemilik karya dapat menagih ganti rugi, meminta produk ditarik dari peredaran, atau menuntut secara hukum. Selain kerugian uang, studio kehilangan kepercayaan klien dan pemain. Karena itu, sebelum memakai aset dari luar, biasakan membaca keterangan lisensinya, mencatat sumbernya, dan menyimpan buktinya di dokumentasi proyek.

Kepemilikan kode. Kode yang kamu tulis atas pesanan klien menjadi milik siapa ditentukan oleh **kontrak**. Bila kontrak menyatakan seluruh kode menjadi milik klien, kode itu tidak boleh disalin untuk klien lain tanpa izin. Yang boleh dipakai ulang adalah **pengetahuan dan kebiasaan kerja** kamu, bukan kode milik klien.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 2.11 Peta konsep pengelolaan aset fisik dan digital. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Penyusutan garis lurus

Studio Pelita Digital membeli laptop pengembang seharga Rp9.600.000 pada awal 2027. Perkiraan nilai sisa Rp1.200.000 dan umur manfaat 4 tahun. Hitung penyusutan per tahun dan nilai buku pada akhir tahun kedua. *(Data fiktif.)*

1. Penyusutan per tahun = (9.600.000 − 1.200.000) ÷ 4 = 8.400.000 ÷ 4 = **Rp2.100.000**.
2. Penyusutan terkumpul sampai akhir tahun ke-2 = 2 × 2.100.000 = 4.200.000.
3. Nilai buku akhir tahun ke-2 = 9.600.000 − 4.200.000 = **Rp5.400.000**.
4. Cek: pada akhir tahun ke-4, nilai buku = 9.600.000 − 4 × 2.100.000 = 1.200.000, sama dengan nilai sisa ✓.

Jawaban: penyusutan Rp2.100.000 per tahun; nilai buku akhir tahun kedua Rp5.400.000.

Contoh 2 — Aset mana yang boleh dipakai dalam gim yang dijual?

Studio gim Lumbung Piksel akan menjual gim berbayar pada 2027. Tiga aset akan dipakai: (1) pustaka kode berlisensi MIT, (2) foto latar berlisensi CC BY, (3) paket ikon berlisensi CC BY-NC. Apa tindak lanjut yang benar untuk tiap aset? *(Kasus fiktif.)*

1. (1) MIT: boleh dipakai pada produk berbayar. Tindak lanjut: **sertakan pemberitahuan lisensi** MIT.
2. (2) CC BY: boleh dipakai pada produk berbayar. Tindak lanjut: **cantumkan nama pencipta** (misalnya di halaman kredit).
3. (3) CC BY-NC: **tidak boleh** dipakai untuk tujuan komersial. Karena gim dijual, paket ikon ini tidak boleh dipakai tanpa izin. Pilihan: minta **izin komersial** kepada pencipta, beli lisensi komersial, atau ganti dengan ikon lain yang lisensinya sesuai.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Mencantumkan nama pencipta pada aset (3) **tidak cukup**.

Jawaban: (1) pakai dan sertakan lisensi; (2) pakai dan cantumkan pencipta; (3) minta izin komersial atau ganti.

Contoh 3 — Mengurutkan tindak lanjut dari kalender aset digital

Pada 1 April 2027, Bu Ratna memeriksa aset digital studionya. *(Data fiktif.)*

Aset	Keterangan
Sertifikat HTTPS situs	berlaku sampai 3 April 2027
Domain studio	berlaku sampai 20 April 2027; perpanjangan otomatis tidak aktif
Lisensi editor video	3 kursi, dipakai 4 orang
Hosting situs	berlaku sampai 31 Agustus 2027

Tentukan urutan tindak lanjut.

1. Cari tenggat paling dekat dan akibatnya: sertifikat HTTPS berakhir **2 hari lagi**. Bila lewat, pengunjung melihat peringatan keamanan. **Diurus pertama.**
2. Domain berakhir 19 hari lagi dan tidak diperpanjang otomatis. Bila lewat, situs bisa mati. **Diurus kedua**, sekaligus mengaktifkan perpanjangan otomatis.
3. Lisensi editor video sudah **melanggar** ketentuan karena kursinya kurang. Walaupun tidak punya tenggat tanggal, ini perlu segera diselesaikan (tambah satu kursi atau kurangi pemakai). **Diurus ketiga.**
4. Hosting masih berlaku lima bulan. **Belum perlu tindakan**, cukup tercatat di kalender.

Jawaban: sertifikat HTTPS → domain → lisensi editor video; hosting belum perlu tindakan.

Contoh 4 — Satu akun administrator untuk semua

Tiga programmer sebuah studio berbagi satu akun administrator repositori dan layanan server agar biaya akun tidak bertambah (2027). Suatu hari sebuah berkas penting terhapus dan tidak ada yang tahu siapa pelakunya. Apa masalahnya dan apa perbaikannya? *(Kasus fiktif.)*

1. Masalah: akun bersama membuat tindakan **tidak dapat ditelusuri** ke satu orang, dan hak akses terlalu besar untuk semua.
2. Bila salah satu programmer keluar, kata sandi akun harus diganti dan semua orang terganggu.
3. Perbaikan: beri **akun terpisah** untuk tiap programmer, dengan **hak akses seperlunya** (misalnya hanya satu orang berhak menghapus), dan simpan riwayat aktivitas.

Jawaban: pakai akun individual dengan hak akses minimum; hemat biaya akun tidak sebanding dengan risiko keamanan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

❖ PETA KONSEP

Gambar 2.11 di atas merangkum konsepnya. Yang harus kamu ingat:

- **Aset fisik** (perangkat) dan **aset digital** (domain, sertifikat, lisensi, kode, data, akun) sama-sama harus dicatat dan dikelola dalam **daur hidup**: rencana, pengadaan, pakai, pelihara, hapus.
- **Inventaris dan catatan peminjaman** memuat kode, kondisi, penanggung jawab, dan tanggal; tindak lanjut dipilih menurut kondisi nyata (terlambat → hubungi; rusak berbahaya → tarik).
- **Penyusutan garis lurus** = (harga perolehan − nilai sisa) ÷ umur manfaat.
- **Masa berlaku** dicatat di satu kalender dan diurutkan menurut tenggat dan akibatnya.
- **Lisensi menentukan apa yang boleh**: NC tidak boleh untuk produk berbayar, nama pencipta saja tidak cukup, dan kode milik klien tidak boleh disalin untuk klien lain.
- **Satu orang satu akun**, hak akses seperlunya.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal tabel aset:

1. **Baca setiap baris** dan tandai yang menyimpang: lewat batas, mendekati tenggat, rusak berbahaya, melebihi jumlah kursi.
2. **Cocokkan tiap pilihan dengan baris yang tepat**, bukan dengan kesan umum.
3. **Pilih tindakan yang menyelesaikan penyebab masalah**, bukan yang sekadar menutupi gejalanya.

Langkah menjawab soal lisensi:

1. **Sebutkan jenis lisensinya** dan syaratnya: komersial atau tidak, wajib atribusi atau tidak, boleh diubah atau tidak.
2. **Cocokkan dengan rencana pemakaian** (produk dijual atau gratis, diubah atau tidak).
3. Bila tidak cocok, pilih **izin, beli lisensi, atau ganti aset**.

Jebakan yang sering dipasang:

- **"Garansi habis, jadi harus diganti."** Garansi bukan satu-satunya dasar mengganti perangkat.
- **"Lisensi dipakai lebih banyak orang daripada kursinya tidak apa-apa."**
- **Nama pencipta dicantumkan = boleh dipakai komersial.** Lisensi NC tetap melarang.
- **Mengubah sedikit aset orang dianggap karya baru.**
- **Memformat perangkat dianggap sudah menghapus data dengan aman.**
- **Perangkat dengan baterai menggembung dipinjamkan lagi** setelah diisi penuh.
- **Berbagi satu akun administrator demi hemat.**
- **Perpanjangan otomatis dianggap selalu aktif**, padahal bisa dimatikan.

> Contoh salah tafsir: sebuah tabel memuat tablet uji yang dipinjam hingga 12 Mei 2027, dan hari ini 10 Mei 2027. Seorang siswa menyimpulkan "tablet harus segera ditarik karena sudah dipinjam seminggu". Tablet itu masih dalam batas peminjaman; tidak ada tindak lanjut sampai batasnya terlewati.

● BEDAH SOAL PREDIKSI

Soal berikut menyajikan tabel aset sebuah studio. Perhatikan cara menyaring satu per satu baris yang bermasalah (bagian 3 dan 6) dan bagaimana pilihan yang "terdengar hemat" ternyata bukan tindak lanjut yang tepat.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Studio aplikasi Rotiku memeriksa daftar asetnya pada 1 Februari 2027. Hasilnya seperti pada tabel.

Aset	Jenis	Keterangan
Domain rotiku.id	digital	berlaku sampai 12 Februari 2027; perpanjangan otomatis tidak aktif
Laptop L-03	fisik	garansi berakhir Januari 2027; kondisi baik, dipakai setiap hari
Lisensi aplikasi desain	digital	5 kursi (seat), dipakai 5 desainer, berlaku sampai Desember 2027
Cadangan basis data	digital	jadwal: setiap Minggu; cadangan terakhir: 3 Januari 2027
Akun admin server cloud	digital	satu akun dipakai bersama oleh 3 programmer

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tindak lanjut yang tepat berdasarkan tabel adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Laptop L-03 perlu segera diganti dengan laptop baru karena masa garansinya sudah habis.

☐ B. Domain rotiku.id perlu diperpanjang sebelum 12 Februari 2027 karena perpanjangan otomatis mati.

☐ C. Lisensi aplikasi desain perlu ditambah karena dipakai oleh lima desainer sekaligus.

☐ D. Jadwal cadangan basis data perlu diperiksa karena cadangan terakhir dibuat hampir sebulan lalu.

☐ E. Ketiga programmer sebaiknya tetap berbagi satu akun admin agar biaya akun tidak bertambah.

Pembahasan **Kunci: B, D**

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pengelolaan aset meliputi aset fisik (perangkat keras: kondisi, garansi, perawatan) dan aset digital (domain, lisensi, data, akun). Aset digital perlu dipantau masa berlaku, jumlah pemakai lisensi, jadwal cadangan, dan hak aksesnya.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: garansi habis tidak berarti rusak; laptop masih dalam kondisi baik sehingga cukup dipantau dan dirawat.

• **Pernyataan 2** ✓ Domain tinggal 11 hari dan tidak diperpanjang otomatis; bila kedaluwarsa, situs dan email studio berhenti.

• **Pernyataan 3** ✗ Lisensi 5 kursi dipakai 5 orang, jadi masih sesuai.

• **Pernyataan 4** ✓ Seharusnya ada cadangan setiap Minggu; cadangan terakhir 3 Januari berarti sekitar 4 cadangan terlewat.

• **Pernyataan 5** ✗ Akun bersama menyulitkan pelacakan siapa melakukan apa dan berisiko bila kata sandinya bocor; setiap orang sebaiknya punya akun sendiri.

Jawaban: pernyataan 2, 4.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

Kerjakan tanpa melihat kunci. Tandai dulu baris atau fakta yang menyimpang dalam tabel atau kasus, lalu cocokkan dengan pilihan.

1. Studio gim Bintang Timur akan menjual gim *Dayung Nusantara* seharga Rp25.000 di toko aplikasi pada Juli 2027. Musik latarnya diunduh dari situs musik gratis dengan lisensi Creative Commons BY-NC 4.0. Menurut keterangan lisensi itu, musik boleh dipakai dan diubah asalkan nama penciptanya dicantumkan dan tidak dipakai untuk tujuan komersial. *(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)*
Tindakan studio yang paling tepat terhadap musik tersebut adalah ....

A. Mencantumkan nama pencipta di halaman kredit, lalu tetap menjual gim itu
B. Mengubah tempo dan nada musik agar dianggap karya baru milik studio
C. Meminta izin komersial dari pencipta atau memakai musik lain yang sah
D. Tetap memakai musik itu karena sudah dibagikan gratis di situs musik
E. Menulis di deskripsi gim bahwa musiknya diambil dari situs musik gratis

2. Lab PPLG memiliki beberapa ponsel untuk menguji aplikasi siswa. Catatan peminjaman pada 10 Maret 2027 seperti pada tabel.

Kode	Perangkat	Peminjam	Batas kembali	Catatan
HP-01	Android 10	Raka	8 Maret 2027	belum kembali
HP-02	Android 14	–	–	baterai menggembung
HP-03	iPhone uji	Sela	16 Maret 2027	–
HP-04	Android 12	–	–	retak kecil di sudut layar, masih berfungsi

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindak lanjut yang tepat berdasarkan tabel adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Petugas perlu menghubungi Raka karena HP-01 sudah melewati batas kembali.
☐ B. HP-02 masih dapat dipinjamkan asalkan baterainya diisi penuh terlebih dahulu.
☐ C. HP-03 perlu segera ditarik karena masa peminjamannya sudah berakhir.
☐ D. HP-02 ditarik dari peminjaman dan baterainya ditangani sebagai limbah berbahaya.
☐ E. Retak di layar HP-04 dicatat di kartu aset agar kondisinya dapat dipantau.

Kunci singkat: 1. C 2. A, D, E

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa mencantumkan nama pencipta tidak cukup untuk memakai karya berlisensi CC BY-NC pada gim yang dijual?
- Mengapa perangkat dengan baterai menggembung ditarik dari peminjaman dan tidak sekadar diisi penuh?
- **Yang harus kuingat:** aset fisik dan digital dikelola dalam daur hidup dan dicatat; penyusutan garis lurus = (harga − sisa) ÷ umur; masa berlaku diurutkan menurut tenggat; lisensi menentukan apa yang boleh; satu orang satu akun.

Latihan interaktif di LMS

Sub bab 2.2: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-2.2

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 2.3 · Pengelolaan file proyek, source code, dan dokumentasi

SMKPPLG-2.3

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menerapkan prosedur pengelolaan file proyek, kode sumber (source code), media, basis data, dan dokumentasi proyek**. Pada soal, bagian ini sangat sering memakai kasus yang terasa nyata: berkas rahasia yang ikut terunggah ke repositori publik, nama berkas desain yang kacau, konflik saat menggabungkan cabang Git, atau cadangan yang ternyata tidak aman. Ini sub bab yang bobot soalnya besar dan banyak soal sulitnya, karena pilihan yang terdengar praktis hampir selalu salah.

Setelah mempelajari sub bab ini, kamu dapat:

- menata folder dan nama berkas proyek serta memberi nomor versi dengan benar;
- menjelaskan alur kerja Git (repositori, commit, cabang, merge, pull request) dan menyelesaikan konflik penggabungan;
- mencegah dan menangani kebocoran berkas rahasia seperti `.env`;
- merancang cadangan dengan aturan 3-2-1 dan menguji pemulihannya;
- menyebut dokumentasi proyek yang wajib ada dan fungsinya.

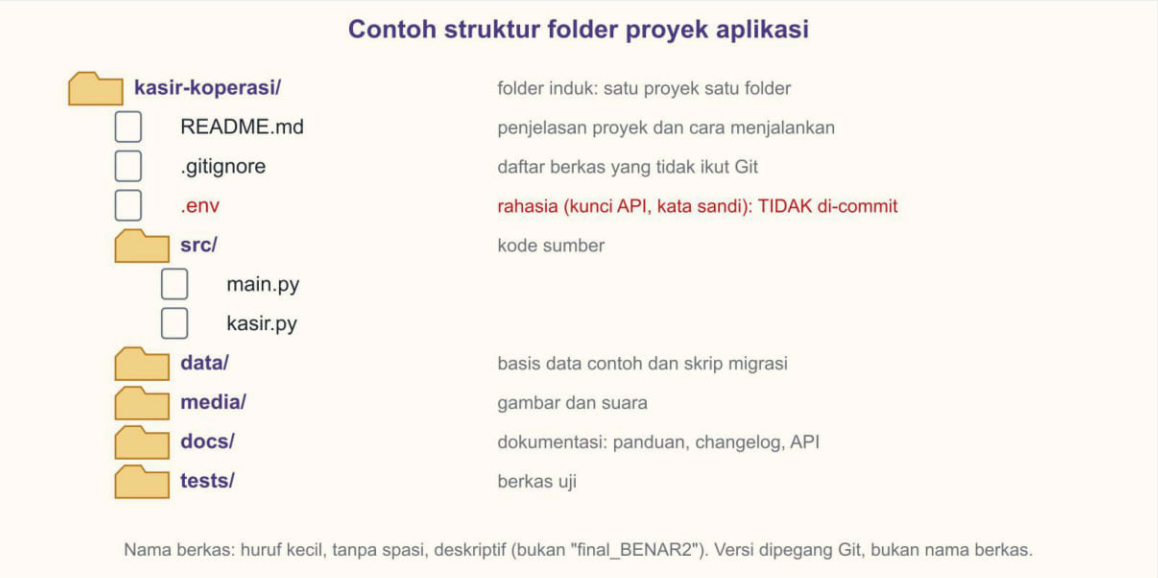
◆ KONSEP INTI

Proyek perangkat lunak menghasilkan banyak berkas: kode, gambar, suara, basis data, dan dokumen. Bila tidak dikelola, tim cepat kehilangan arah. Mereka tidak tahu versi mana yang terbaru, perubahan siapa yang menimpa pekerjaan siapa, atau di mana cadangannya ketika komputer rusak. Sub bab ini membahas cara mengelolanya dengan rapi dan aman.

1. Struktur folder dan penamaan berkas

Satu proyek memiliki satu folder induk dengan susunan yang konsisten (Gambar 2.12). Kode di folder tersendiri, media di folder tersendiri, dokumentasi di folder tersendiri. Susunan yang sama di setiap proyek membuat orang baru cepat paham.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 2.12 Contoh struktur folder proyek aplikasi. Sumber: digambar oleh Tim Next Level Study Indonesia.

Aturan penamaan yang baik:

- **deskriptif**: `laporan_penjualan.py`, bukan `a1.py` atau `baru.py`;
- **konsisten**: huruf kecil, kata dipisah garis bawah atau tanda hubung, tanpa spasi;
- **tidak memakai kata "final"** dan sejenisnya (`desain_final_BENAR2.fig`), karena tidak jelas mana yang terbaru;
- bila terpaksa memakai versi pada nama berkas yang tidak dikelola Git, pakai **nomor versi dan tanggal**, misalnya `desain_v04_2027-03-14.fig`; nomor yang naik teratur mudah diurutkan.

Berkas versi lama **jangan dihapus sembarangan**. Versi lama adalah riwayat yang bisa dibutuhkan ketika sesuatu salah. Untuk kode, riwayat itu dipegang Git sehingga nama berkas tidak perlu memuat versi.

2. Versi perangkat lunak: MAJOR.MINOR.PATCH

Rilis perangkat lunak diberi **nomor versi** tiga bagian, `MAJOR.MINOR.PATCH`, misalnya `2.4.1` (Gambar 2.13).



Gambar 2.13 Arti tiga bagian nomor versi. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **MAJOR** naik bila ada perubahan yang **tidak kompatibel** dengan versi lama (misalnya format data berubah sehingga berkas lama tidak bisa dibuka).
- **MINOR** naik bila **menambah fitur** yang masih kompatibel.
- **PATCH** naik bila hanya **memperbaiki bug kecil**.

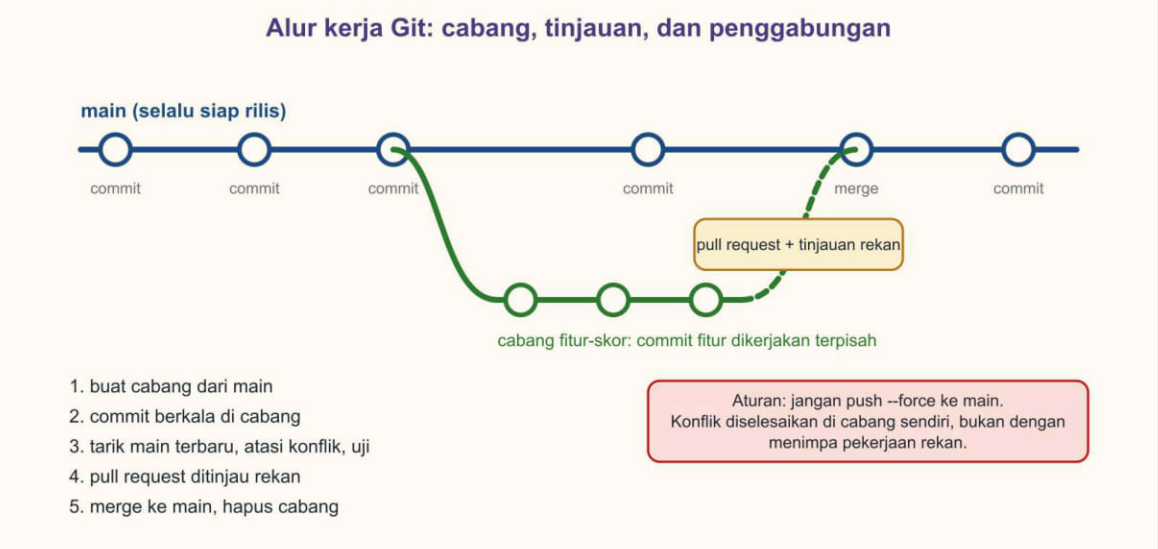
Bila angka yang lebih besar naik, angka di kanannya kembali ke 0. Jika satu rilis memuat beberapa jenis perubahan, **angka yang tingkatnya paling besar yang dinaikkan**.

3. Git: mengelola versi kode bersama

Git adalah sistem kontrol versi: ia mencatat setiap perubahan kode, siapa yang membuatnya, dan kapan, sehingga tim dapat bekerja bersamaan dan kembali ke keadaan sebelumnya bila perlu. Istilah dasarnya:

- **Repositori (repository)**: tempat kode beserta seluruh riwayat perubahannya.
- **Commit**: satu catatan perubahan, lengkap dengan pesan singkat yang menjelaskan isinya.
- **Cabang (branch)**: jalur kerja terpisah. Cabang **main** biasanya memuat kode yang selalu siap rilis, sedangkan setiap fitur dikerjakan di cabangnya sendiri.
- **Merge**: menggabungkan perubahan dari satu cabang ke cabang lain.
- **Pull request**: permintaan menggabungkan cabang fitur ke main, yang **ditinjau rekan** dulu.
- **Konflik**: terjadi bila dua cabang mengubah **bagian yang sama** dari berkas yang sama, sehingga Git tidak dapat memilih sendiri.

Alur kerja yang sehat ditunjukkan pada Gambar 2.14: buat cabang dari main, kerjakan fitur dengan commit berkala, tarik perubahan terbaru dari main, ajukan pull request, ditinjau, lalu di-merge ke main.



Gambar 2.14 Alur kerja Git dengan cabang fitur, tinjauan, dan penggabungan. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perintah dasar yang dipakai sehari-hari di terminal (tanda $ hanya menandai awal perintah):

```
$ git status                    # melihat berkas yang berubah
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
$ git add kasir.py                # memilih berkas yang akan dicatat
$ git commit -m "Tambah diskon"   # mencatat perubahan dengan pesan
$ git checkout -b fitur-skor      # membuat dan pindah ke cabang baru
$ git merge main                  # menggabungkan main ke cabang ini
$ git push origin fitur-skor      # mengirim cabang ke repositori bersama
```

Nama cabang juga sebaiknya jelas dan konsisten, misalnya `fitur-skor`, `perbaikan-pajak`, atau `uji-login`, sehingga rekan langsung tahu isi dan tujuan cabang tanpa membukanya. Cabang yang sudah selesai digabung dihapus agar daftar tidak penuh cabang usang.

Pesan commit yang baik singkat dan menyebut perubahannya ("Perbaiki hitung pajak pada struk"), bukan "update" atau "perbaikan". Pesan yang jelas membuat riwayat berguna ketika kamu mencari kapan sebuah kesalahan masuk.

Menyelesaikan konflik. Bila konflik muncul saat menggabungkan, selesaikan **di cabang sendiri**: tarik main terbaru ke cabangmu, buka berkas yang bertanda konflik, putuskan isi akhirnya (sering kali menyimpan keduanya), hapus tanda konflik, jalankan uji, lalu commit. Git menandai konflik seperti ini:

```
<<<<<<< HEAD
LEVEL_AWAL = 1
=======
MAKS_PEMAIN = 6
>>>>>>> main
```

Bagian di atas `=======` berasal dari cabangmu, bagian di bawahnya dari cabang yang digabung. Cara menyelesaikan yang benar bukan memilih sembarang satu sisi, melainkan **memahami perubahan keduanya**, menggabungkannya, lalu **menguji ulang**.

Cara yang **salah**: memakai `push --force` ke main (menimpa pekerjaan rekan), menghapus berkas agar tidak bentrok, menyalin folder secara manual untuk menimpa folder lain, atau meminta rekan menghapus commit mereka.

4. Berkas rahasia: `.env`, `.gitignore`, dan kebocoran

Aplikasi sering memerlukan **rahasia**: kunci API, kata sandi basis data, token. Rahasia ini disimpan di berkas `.env` atau variabel lingkungan dan **tidak boleh masuk ke repositori**.

Cara mencegahnya: tulis nama berkas rahasia di `.gitignore`, yaitu daftar berkas yang tidak dilacak Git. **Tetapi `.gitignore` hanya berlaku untuk berkas yang belum pernah di-commit.** Berkas yang sudah terlanjur dilacak tetap dilacak.

Cara yang lebih baik daripada menyimpan rahasia di dalam kode adalah **membacanya dari luar kode**: kode hanya menyebut nama pengaturannya, sedangkan nilai sebenarnya dimasukkan di komputer atau server tempat aplikasi berjalan. Dengan begitu, siapa pun yang melihat kode tidak ikut melihat rahasianya, dan rahasia bisa diganti tanpa mengubah kode.

Bila rahasia terlanjur terunggah ke repositori, terutama yang publik, urutannya (Gambar 2.15):

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 2.15 Langkah menangani kunci rahasia yang terlanjur terunggah. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Cabut dan ganti kunci seketika.** Rahasia yang pernah publik **dianggap sudah diketahui orang lain**, walaupun hanya dua jam.
2. Keluarkan berkas dari pelacakan Git dan tambahkan ke `.gitignore`.
3. Bersihkan **riwayat** bila perlu, karena commit lama masih menyimpan isi berkas.
4. Periksa penyalahgunaan (tagihan, catatan akses).
5. Simpan rahasia di tempat yang benar, misalnya variabel lingkungan atau pengelola rahasia.

Yang **tidak cukup**: menghapus berkas lewat commit baru (riwayat tetap ada), menjadikan repositori privat setelah terlanjur publik, atau hanya menambahkan `.gitignore`. Itulah sebabnya langkah pertama selalu **mencabut dan mengganti kunci**.

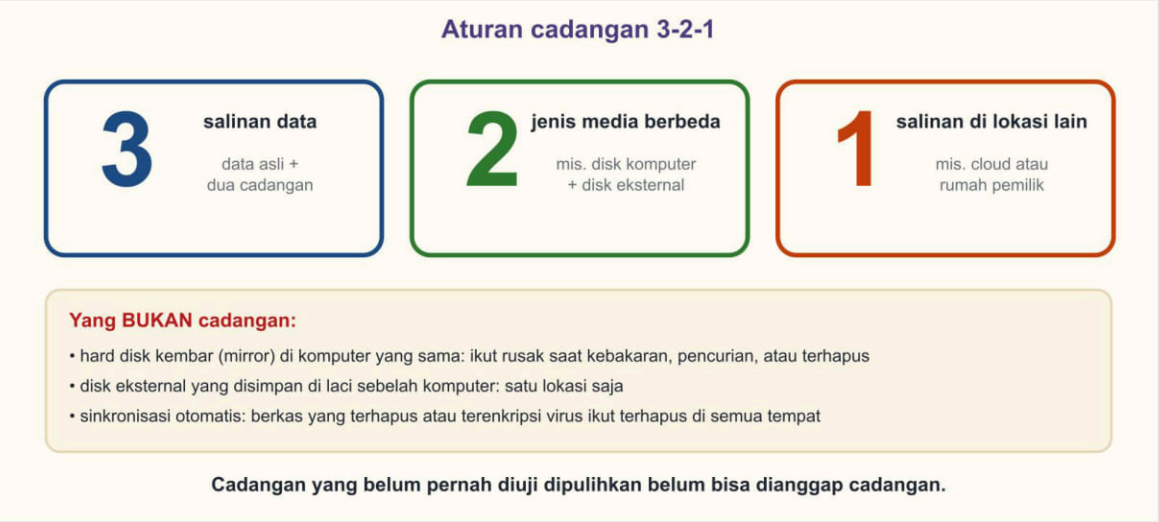
Ringkasnya, tiga kebiasaan menjaga kode tetap sehat: **commit kecil dan sering** (mudah ditelusuri dan dibatalkan), **tarik main terbaru sebelum mengajukan pull request** (konflik kecil lebih mudah dibereskan daripada konflik besar yang menumpuk), dan **jangan pernah menulis ulang riwayat di cabang bersama**.

5. Cadangan (backup): aturan 3-2-1

Cadangan adalah salinan data yang dipakai untuk memulihkan sesuatu yang hilang karena kerusakan perangkat, kebakaran, pencurian, kesalahan manusia, atau virus. Aturan umum yang mudah diingat adalah **3-2-1** (Gambar 2.16):

- **3** salinan data (data asli ditambah dua cadangan);
- **2** jenis media yang berbeda (misalnya disk komputer dan disk eksternal);
- **1** salinan di **lokasi lain** (misalnya penyimpanan cloud atau rumah pemilik).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 2.16 Aturan 3-2-1 dan hal yang bukan cadangan. Sumber: digambar oleh Tim Next Level Study Indonesia.

Yang sering dikira cadangan padahal bukan: **dua disk kembar (mirror) di komputer yang sama** (melindungi dari satu disk rusak, tetapi tidak dari kebakaran atau terhapus tak sengaja), **disk eksternal di laci samping komputer** (satu lokasi), dan **sinkronisasi otomatis** (berkas yang terhapus ikut terhapus di semua tempat).

Dua praktik penting lainnya: tentukan **jadwal** yang sesuai (seberapa banyak data boleh hilang? bila jawabannya "tidak lebih dari sehari", cadangan harian), dan **uji pemulihan** secara berkala, karena cadangan yang belum pernah dicoba dipulihkan belum tentu bisa dipakai. **Cadangan basis data** biasanya dibuat dengan mengekspor isi basis data (*dump*) pada jadwal tetap dan menyimpannya di luar server yang sama.

6. Dokumentasi proyek

Dokumentasi membuat proyek bisa dipahami dan dirawat orang lain, termasuk dirimu sendiri enam bulan kemudian. Bentuknya:

- **README**: pengantar proyek, cara memasang, dan cara menjalankan. Berkas pertama yang dibaca orang.
- **Komentar kode** dan **keterangan fungsi**: menjelaskan *mengapa* kode ditulis begitu, bukan mengulang apa yang sudah jelas dari kodenya.
- **Changelog**: daftar perubahan tiap versi.
- **Dokumentasi API dan basis data**: daftar layanan, masukan dan keluarannya, serta rancangan tabel.
- **Panduan pengguna** dan **catatan rapat** atau keputusan proyek.

Kerangka README yang sederhana tetapi berguna memuat: nama dan tujuan proyek, cara memasang, cara menjalankan, cara mengisi pengaturan (tanpa menuliskan rahasia sungguhan), cara menjalankan uji, dan nama pengelola proyek.

Satu contoh komentar yang baik menjelaskan alasan, bukan mengulang kode:

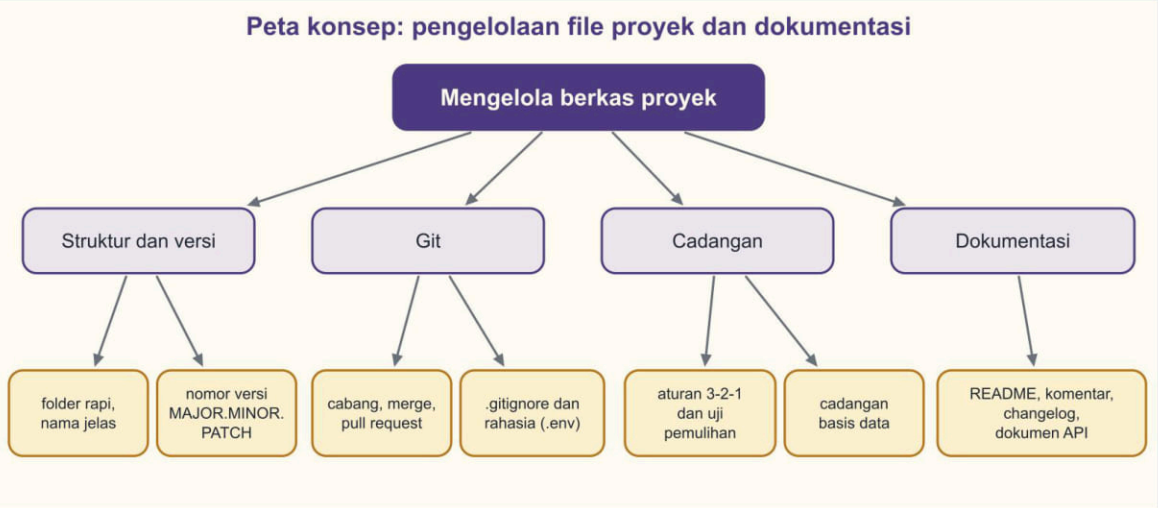
```
# tunggu 2 detik: server pembayaran menolak permintaan yang datang lebih cepat
jeda = 2
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Komentar `# jeda sama dengan 2` hanya mengulang isi baris dan tidak berguna.

7. Media dan basis data dalam proyek

Aset media gim (gambar, suara, model) berukuran besar. Simpan di folder terpisah dan **jangan campur** dengan kode; untuk berkas besar, gunakan penyimpanan khusus atau fitur khusus Git. Rancangan **basis data** dikelola sebagai **skrip migrasi** (perubahan skema yang tercatat dan berurutan) serta data contoh, sedangkan data produksi yang berisi data pribadi tidak dimasukkan ke repositori.



Gambar 2.17 Peta konsep pengelolaan berkas proyek dan dokumentasi. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menentukan nomor versi berikutnya

Aplikasi Catatan Kelas berada pada versi **1.4.2**. Tentukan nomor versi berikutnya untuk tiga rilis terpisah, lalu untuk satu rilis yang memuat ketiganya sekaligus. *(Kasus fiktif.)*

- (a) Memperbaiki kesalahan ketik dan bug kecil pada perhitungan rata-rata.
- (b) Menambah fitur ekspor catatan ke PDF tanpa mengubah fitur lain.
- (c) Mengubah format penyimpanan sehingga berkas catatan lama tidak dapat dibuka lagi.

1. (a) Hanya perbaikan bug → **PATCH** naik: 1.4.2 → **1.4.3**.
2. (b) Fitur baru yang kompatibel → **MINOR** naik, PATCH kembali 0: 1.4.2 → **1.5.0**.
3. (c) Tidak kompatibel → **MAJOR** naik, yang lain kembali 0: 1.4.2 → **2.0.0**.
4. Satu rilis memuat (a), (b), dan (c): naikkan yang tingkatnya paling besar, yaitu MAJOR → **2.0.0**.

Jawaban: (a) 1.4.3; (b) 1.5.0; (c) 2.0.0; gabungan 2.0.0.

Contoh 2 — Menyelesaikan konflik penggabungan

Rani mengerjakan fitur level awal di cabang `fitur-level`. Ketika ia menarik `main` terbaru, Git menandai konflik pada berkas `pengaturan.py`:

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
<<<<<<< HEAD
LEVEL_AWAL = 3
=======
MAKS_PEMAIN = 8
>>>>>>> main
```

Cabang Rani menambah `LEVEL_AWAL`, sedangkan rekannya di `main` menambah `MAKS_PEMAIN`; keduanya baris baru yang berbeda. Bagaimana isi akhir berkas yang benar? *(Kasus fiktif.)*

1. Pahami kedua sisi: sisi atas (cabang Rani) menambah `LEVEL_AWAL = 3`; sisi bawah (main) menambah `MAKS_PEMAIN = 8`.
2. Tanyakan apakah keduanya diperlukan: ya, keduanya adalah pengaturan berbeda yang dibutuhkan.
3. Gabungkan dan hapus tiga baris tanda konflik. Hasilnya:

```
LEVEL_AWAL = 3
MAKS_PEMAIN = 8
```

1. Jalankan uji, lalu commit. Mengambil satu sisi saja akan menghilangkan pekerjaan salah satu pihak.

Jawaban: simpan kedua baris, hapus penanda konflik, uji, dan commit.

Contoh 3 — Merancang cadangan 3-2-1 untuk studio kecil

Sebuah studio animasi kecil menyimpan 300 GB berkas proyek di satu komputer, dan menyalinnya tiap Minggu ke disk eksternal di rak sebelah komputer (2027). Rancang cadangan yang memenuhi 3-2-1, dan perkirakan lama unggah pertama ke cloud bila kecepatan unggah 10 Mbps. *(Kasus fiktif.)*

1. **Periksa kondisi sekarang**: ada 2 salinan (komputer dan disk eksternal), 2 media, tetapi **satu lokasi**. Belum memenuhi 3-2-1 karena tidak ada salinan di lokasi lain.
2. **Tambah salinan ketiga di lokasi lain**: unggah ke penyimpanan cloud (atau disk eksternal kedua yang disimpan di rumah pemilik). Sekarang: data asli + disk eksternal + cloud = 3 salinan, 2 media (disk dan cloud), 1 di lokasi lain.
3. **Jadwal**: cadangan tambahan harian ke cloud (hanya berkas yang berubah), cadangan penuh mingguan ke disk eksternal. **Uji pemulihan** satu kali tiap bulan.
4. **Lama unggah pertama**: 300 GB = 300 × 8 = 2.400 gigabit = 2.400.000 megabit. Waktu = 2.400.000 ÷ 10 = 240.000 detik = 240.000 ÷ 3.600 ≈ **67 jam**, sekitar tiga hari. Unggah berikutnya jauh lebih ringan karena hanya berkas yang berubah.

Jawaban: tambahkan salinan di cloud atau lokasi lain, jadwalkan cadangan, uji pemulihan; unggah pertama ±67 jam.

Contoh 4 — Token basis data tertulis di README

Tim Lapor Warga menulis alamat dan token basis data langsung di `README.md` lalu mengunggahnya ke repositori publik pada 2027. Tiga puluh menit kemudian seorang anggota menyadarinya. Apa urutan tindakan yang tepat? *(Kasus fiktif.)*

1. Tindakan pertama: **cabut atau ganti token basis data**, karena token yang pernah publik dianggap sudah diketahui orang lain.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Hapus token dari README dan pindahkan ke variabel lingkungan; riwayat commit lama masih menyimpannya, jadi **bersihkan riwayat** bila perlu.
3. **Periksa log akses basis data** untuk melihat apakah ada akses mencurigakan.
4. Setelah itu baru hal lain, misalnya memperbarui dokumentasi cara mengisi pengaturan.

Jawaban: ganti token lebih dulu; menghapus teks dari README atau menjadikan repositori privat saja tidak cukup.

❖ PETA KONSEP

Gambar 2.17 di atas merangkum pengelolaan berkas proyek. Yang harus kamu ingat:

- **Folder rapi dan nama deskriptif**; versi kode dipegang Git; versi rilis memakai **MAJOR.MINOR.PATCH**.
- **Git**: commit, cabang, merge, pull request; **konflik diselesaikan di cabang sendiri** dengan memahami dan menggabungkan perubahan, lalu diuji. Jangan `push --force` ke main.
- **`.gitignore` hanya untuk berkas yang belum dilacak.** Rahasia yang terlanjur publik **diganti dulu**; menghapus berkas dari commit baru atau menjadikan repositori privat tidak cukup.
- **Cadangan 3-2-1** (3 salinan, 2 media, 1 lokasi lain) dan **uji pemulihan**; mirror dan sinkronisasi bukan cadangan.
- **Dokumentasi**: README, komentar, changelog, dokumen API, panduan.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal kebocoran dan keamanan berkas:

1. **Tanyakan: apakah rahasia sudah terlihat orang lain?** Bila ya, anggap bocor dan ganti lebih dulu.
2. **Pilih langkah yang menghilangkan akibat, bukan sekadar tampilan.** Menyembunyikan berkas dari tampilan tidak menarik kembali rahasia yang sudah diketahui.

Langkah menjawab soal Git:

1. **Tentukan apa yang terjadi**: konflik, salah cabang, atau perubahan menimpa pekerjaan lain.
2. **Pilih tindakan yang menjaga pekerjaan semua pihak**: tarik main terbaru, atasi konflik di cabang sendiri, uji, lalu ajukan pull request.

Langkah menjawab soal cadangan:

1. **Hitung salinan, media, dan lokasi.** Salinan di tempat yang sama tidak menambah perlindungan dari kebakaran atau pencurian.
2. **Pilih penambahan yang menutup kekurangan 3-2-1**, biasanya salinan di lokasi lain.

Jebakan yang sering dipasang:

- **Menghapus `.env` lewat commit baru** dianggap menyelesaikan kebocoran.
- **`.gitignore` dianggap menghapus berkas yang sudah di-commit.**
- **Repositori dijadikan privat** setelah terlanjur publik dianggap menarik kembali rahasia.
- **`push --force` untuk "menghilangkan konflik".**
- **Dua disk kembar di satu komputer** dianggap cadangan.
- **Sering menyalin ke disk yang sama** dianggap lebih aman.
- **Nama berkas "final_BENAR" sebagai manajemen versi**, atau menghapus semua versi lama.
- **Angka versi salah dinaikkan**: bug kecil memakai MAJOR, atau fitur baru memakai PATCH.

> Contoh salah tafsir: sebuah tim sudah menambahkan `.env` ke `.gitignore` pada hari ke-10 proyek, padahal `.env` sudah di-commit pada hari ke-3. Seorang siswa menyimpulkan "karena sudah ada di `.gitignore`, rahasia di dalamnya aman". Berkas itu sudah dilacak sejak hari ke-3 dan tetap ada di riwayat; `.gitignore` tidak menghapusnya. Rahasia itu harus diganti.

• BEDAH SOAL PREDIKSI

Soal berikut adalah kasus kunci rahasia yang terlanjur terunggah. Perhatikan bagaimana kelima langkah tampak "menyelesaikan masalah" tetapi hanya satu yang menarik kembali bahaya sebenarnya (bagian 4).

►. Pada Maret 2027, Fajar mengunggah (push) proyek aplikasi cuaca ke repositori GitHub publik. Dua jam kemudian rekannya melihat bahwa berkas `.env` ikut terunggah. Berkas itu berisi kunci API berbayar milik tim yang terhubung ke kartu kredit kantor. Ketua tim

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

meminta Fajar segera mengambil langkah pertama yang paling penting.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah pertama yang paling tepat adalah ....

A. Mencabut kunci API lama, membuat kunci baru, lalu mengeluarkan .env dari repositori
B. Menghapus berkas .env dengan commit baru agar berkas itu tidak terlihat lagi di GitHub
C. Mengubah repositori menjadi privat agar orang lain tidak dapat melihat isinya lagi
D. Menambahkan .env ke .gitignore agar berkas itu otomatis terhapus dari GitHub
E. Mengganti nama berkas .env menjadi .env.lama, lalu mengunggah ulang proyeknya

Pembahasan **Kunci: A**

Konsep: Riwayat Git menyimpan setiap versi berkas. Kunci yang pernah terunggah ke repositori publik harus dianggap bocor: kunci dicabut dan diganti lebih dulu, baru berkasnya dikeluarkan dari repositori dan ditambahkan ke .gitignore.

Langkah:
1. Repositori publik selama 2 jam → kunci mungkin sudah disalin orang lain.
2. Menyembunyikan berkas tidak membatalkan kunci yang sudah bocor.
3. Mencabut kunci membuat salinan yang bocor tidak bisa dipakai lagi.

Analisis setiap opsi:
• **A.** ✓ Kunci lama tidak berlaku lagi sehingga salinan yang bocor tidak berguna; setelah itu berkas dibersihkan.
• **B.** ✗ Jebakan: commit baru hanya menghapus berkas dari versi terbaru; versi lama masih ada di riwayat dan kuncinya masih aktif.
• **C.** ✗ Kunci mungkin sudah disalin selama dua jam repositori publik; menjadikannya privat tidak membatalkan kunci itu.
• **D.** ✗ Jebakan: .gitignore hanya mencegah berkas BARU dilacak; berkas yang sudah terunggah tidak terhapus.
• **E.** ✗ Isi berkas tetap terunggah dengan nama baru dan versi lamanya masih ada di riwayat.

Jawaban: A.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap soal, tulis dulu apa yang berisiko hilang atau bocor, lalu pilih langkah yang menghilangkan risiko itu, bukan yang hanya menyamarkannya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. Folder bersama tim desain aplikasi Kopi Kita (Maret 2027) berisi berkas berikut: `desain_final.fig`, `desain_final_revisi.fig`, `desain_final_BENAR.fig`, dan `desain_final_BENAR2.fig`. Anggota tim sering membuka versi yang salah, dan minggu lalu desain lama terkirim ke klien. Ketua tim ingin memperbaiki cara pengelolaan berkas desain.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Pola nama berkas seperti itu menyulitkan anggota tim mengetahui versi yang paling baru.	○	○
Pola nama dengan nomor versi dan tanggal, misalnya desain_v04_2027-03-14.fig, membantu mengurutkan versi.	○	○
Agar tidak membingungkan, berkas versi lama sebaiknya dihapus begitu versi baru dibuat.	○	○

2. Tim gim *Ombak Biru* (2027) memakai aturan Git: cabang main harus selalu siap rilis; setiap fitur dikerjakan di cabang sendiri lalu digabung lewat pull request setelah ditinjau rekan. Reza menyelesaikan fitur papan skor di cabang `fitur-skor`. Saat ia membuka pull request, GitHub menampilkan konflik pada berkas `config.py` karena selama itu ada tiga commit baru dari rekan di main yang juga mengubah berkas tersebut.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Reza yang paling tepat adalah ....

A. Menarik main terbaru ke fitur-skor, membereskan konflik, lalu menguji ulang
B. Mendorong fitur-skor langsung ke main dengan push --force agar konflik hilang
C. Menghapus config.py dari cabang fitur-skor agar tidak bentrok lagi dengan main
D. Menyalin seluruh folder proyek fitur-skor menimpa folder main secara manual
E. Meminta rekan menghapus tiga commit mereka di main agar pull request bersih

3. Studio gim kecil menyimpan seluruh berkas proyek (kode, gambar, suara; 300 GB) di satu komputer kerja (2027). Setiap Jumat, berkas disalin ke hard disk eksternal yang disimpan di laci meja di samping komputer itu. Pemilik studio khawatir semua data hilang bila terjadi kebakaran atau pencurian, lalu meminta prosedur cadangan diperbaiki.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perbaikan yang paling tepat untuk kekhawatiran tersebut adalah ....

A. Menambah hard disk eksternal kedua yang disimpan di laci yang sama
B. Memasang dua hard disk kembar (mirror) di dalam komputer kerja
C. Mengubah jadwal cadangan menjadi setiap hari ke hard disk yang sama
D. Memindahkan semua berkas ke hard disk eksternal agar komputer ringan
E. Menambah salinan di lokasi lain, misalnya penyimpanan cloud

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kunci singkat: 1. 1B 2B 3S 2. A 3. E

✔ PERIKSA DIRI

- Mengapa menghapus berkas rahasia lewat commit baru tidak cukup, dan apa langkah pertama yang tepat?
- Mengapa dua disk kembar di komputer yang sama dan sinkronisasi otomatis bukan cadangan yang memenuhi 3-2-1?
- **Yang harus kuingat:** rahasia yang pernah publik diganti lebih dulu; `.gitignore` hanya berlaku untuk berkas yang belum dilacak; konflik diselesaikan di cabang sendiri lalu diuji; cadangan 3-2-1 dengan satu salinan di lokasi lain dan diuji pemulihannya; MAJOR untuk perubahan tak kompatibel, MINOR untuk fitur, PATCH untuk perbaikan bug.

Latihan interaktif di LMS

Sub bab 2.3: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-2.3

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kuis Akhir Bab 2

SMKPPLG-KB2

Jumlah soal: 25 **Waktu anjuran:** 60 menit

Kerjakan sendiri dan berwaktu. Setelah selesai, cocokkan dengan kunci di Bagian D, isi tabel skor, lalu lihat anjuran di bawahnya.

Kerjakan Kuis Bab 2 di LMS

Skor dan progresmu tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-KB2

1. Ruang praktik perakitan komputer di SMK Cahaya Mandiri (2027) akan dilengkapi rambu untuk menandai titik kumpul saat evakuasi. Empat rambu yang tersedia di gudang perlengkapan ditunjukkan pada gambar.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Rambu yang tepat untuk menandai titik kumpul adalah ....



A. Rambu P
B. Rambu Q
C. Rambu R
D. Rambu S
E. Tidak ada rambu pada gambar yang sesuai

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Dani, pemrogram magang di sebuah studio gim (2027), bekerja delapan jam sehari di depan laptop. Belakangan lehernya nyeri dan matanya terasa kering. Pembimbingnya menyarankan agar ia memperbaiki cara kerjanya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Saran yang sesuai dengan prinsip ergonomi adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Mengatur layar agar bagian atasnya kira-kira setinggi mata pada jarak sekitar 50–70 cm
☐ B. Meletakkan layar serendah mungkin di pangkuan agar leher dapat menunduk dengan nyaman
☐ C. Setiap 20 menit melihat benda sejauh sekitar 6 meter selama kira-kira 20 detik
☐ D. Duduk membungkuk mendekati layar supaya punggung bertumpu pada tepi meja
☐ E. Menjaga siku sekitar 90° dan telapak kaki menapak lantai selama bekerja

3. Hendra, pemrogram di sebuah studio (2027), baru menyelesaikan fitur laporan untuk klien A. Kontrak dengan klien A menyebutkan bahwa seluruh kode hasil pekerjaan menjadi milik klien A. Kini klien B meminta fitur yang hampir sama, dan Hendra merasa kode klien A sangat cocok dipakai.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Hendra boleh menyalin kode itu ke proyek klien B karena ia sendiri yang menulisnya.	○	○
Hendra dapat menulis kode baru untuk klien B atau meminta izin tertulis dari klien A.	○	○
Menyampaikan rencana ini kepada ketua tim termasuk komunikasi terbuka yang profesional.	○	○

4. Seorang bendahara studio (2027) menyusun daftar aset perusahaan menjadi dua kelompok: aset fisik (berwujud) dan aset digital (tidak berwujud tetapi bernilai).
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Benda yang termasuk aset digital adalah ....

A. lisensi berlangganan perangkat lunak desain
B. meja kerja ergonomis untuk para pemrogram studio
C. monitor dua puluh empat inci di meja kerja
D. router nirkabel yang dipasang di kantor
E. proyektor yang dipasang di ruang rapat

5. Pada akhir pekan, klien menelepon Eko karena layanan aplikasi mati dan perlu dinyalakan ulang dari server. Eko tidak punya akses ke server itu. Administrator server sedang cuti. Ia

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

tahu kata sandi akun administrator karena pernah tertulis di buku catatan bersama tim.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Eko yang paling tepat adalah ....

A. memakai akun administrator itu karena kata sandinya sudah diketahui seluruh anggota tim di kantor
B. meminta rekan mengirimkan kata sandi lewat pesan singkat lalu menghapus pesannya
C. membuat akun administrator baru sendiri tanpa memberi tahu siapa pun
D. menunda pekerjaan sampai administrator kembali tanpa melapor kepada ketua tim
E. menghubungi ketua tim agar akses sementara diberikan pihak berwenang sesuai prosedur

6. Studio Warna Lapis (2027) memiliki dua aset yang disusutkan dengan metode garis lurus. Data kedua aset ditunjukkan pada tabel berikut.

Aset	Harga perolehan	Nilai sisa	Umur manfaat	Sudah dipakai
Laptop desain	Rp15.000.000	Rp3.000.000	4 tahun	2 tahun
Server mini	Rp20.000.000	Rp2.000.000	6 tahun	3 tahun

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Jumlah nilai buku kedua aset saat ini adalah ....

A. Rp15.000.000
B. Rp17.500.000
C. Rp20.000.000
D. Rp35.000.000
E. Rp5.000.000

7. Tim sebuah studio aplikasi (2027) ingin membangun kebiasaan kerja yang membuat klien dan rekan percaya kepada mereka.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kebiasaan yang mencerminkan budaya kerja profesional adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Melaporkan sejak awal bahwa sebuah fitur akan terlambat dua hari
☐ B. Menerima masukan dari tinjauan kode dan menanyakan bagian yang belum jelas
☐ C. Menyembunyikan kesalahan kecil agar tim tidak panik selama masih bisa diperbaiki sendiri
☐ D. Memberi perkiraan waktu yang jujur walaupun kurang menyenangkan bagi klien
☐ E. Memakai lisensi perangkat lunak kantor untuk proyek pribadi saat jam istirahat

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

8. Saat memindahkan perangkat di gudang sebuah studio (2027), Rio tersengat listrik dari kabel perpanjangan yang kulitnya terkelupas, dan tangannya menempel pada kabel itu sehingga ia tidak dapat melepaskan diri. Rani berada di dekatnya, dan saklar utama gudang dapat dijangkau dengan aman.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan yang tepat dilakukan Rani adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Menarik tubuh Rio dengan tangan kosong agar ia cepat terlepas dari kabel
☐ B. Memutus aliran listrik dari saklar utama lebih dulu, baru mendekati Rio
☐ C. Memanggil bantuan dan melapor kepada guru atau petugas penanggung jawab
☐ D. Menyiram kabel dengan air untuk mendinginkannya sebelum Rio diperiksa
☐ E. Setelah aman, memeriksa keadaan Rio, memberi pertolongan pertama, dan melaporkan kejadian walau lukanya ringan

9. Tim sebuah studio (2027) memakai pustaka sumber terbuka berlisensi MIT, ikon berlisensi Creative Commons BY, dan satu lisensi berbayar editor kode yang dibeli untuk satu perangkat.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Pustaka berlisensi MIT dapat dipakai pada produk berbayar selama pemberitahuan hak ciptanya dipertahankan.	○	○
Ikon berlisensi CC BY boleh dipakai tanpa menyebut nama pembuatnya.	○	○
Lisensi editor kode untuk satu perangkat boleh dipasang di semua komputer kantor.	○	○

10. Di Studio Gim Rimba Pixel (2027), cahaya matahari dari jendela jatuh langsung ke layar beberapa pemrogram sehingga mata mereka cepat lelah. Pengelola studio mempertimbangkan empat tindakan pengendalian seperti pada tabel berikut.

Kode	Tindakan
P	Memasang tirai penahan cahaya pada jendela di samping layar
Q	Memindahkan posisi meja sehingga layar tidak lagi menghadap jendela dan tidak terkena cahaya langsung
R	Menetapkan jadwal istirahat layar lima menit setiap jam disertai pengumuman di dinding
S	Meminta setiap pemrogram memakai kacamata penyaring cahaya

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Urutan tindakan dari yang paling efektif ke yang kurang efektif menurut hierarki pengendalian risiko adalah ....

A. P – Q – R – S
B. Q – R – P – S
C. S – R – P – Q
D. Q – P – R – S
E. R – Q – P – S

11. Sebuah studio mengelola repositori kode, akun layanan awan, dan folder desain bersama. Dua pekan lagi tiga peserta magang akan bergabung, dan seorang pemrogram lama akan berhenti bekerja.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah pengelolaan aset digital yang tepat adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Memberi seluruh peserta magang akun administrator agar tidak perlu menunggu izin
☐ B. Memberi akses sesuai tugas, misalnya hanya hak baca pada repositori utama
☐ C. Meninjau daftar pemilik akses secara berkala dan mencabut akses orang yang sudah keluar
☐ D. Memakai satu kata sandi yang sama untuk semua akun supaya mudah diingat
☐ E. Mengaktifkan verifikasi dua langkah pada akun layanan awan studio

12. Sebuah studio (2027) memakai kumpulan ikon berlisensi Creative Commons BY-SA 4.0 pada aplikasi gratis buatannya. Tim mengganti warna dan bentuk beberapa ikon agar cocok dengan tema aplikasi, lalu berencana mengunggah ikon hasil ubahan itu ke repositori terbuka milik studio.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Syarat yang harus dipenuhi studio menurut lisensi tersebut adalah ....

A. cukup mencantumkan nama pembuat; ikon hasil ubahan boleh dilisensikan bebas sesuai keinginan studio
B. mencantumkan nama pembuat asli dan membagikan ikon hasil ubahan dengan lisensi sama, BY-SA
C. tidak boleh mengubah ikon sama sekali karena lisensi SA melarang perubahan
D. tidak perlu memenuhi syarat apa pun karena aplikasinya dibagikan gratis
E. harus meminta izin tertulis dan membayar karena SA berarti sumber dayanya berbayar

13. Rani bekerja paruh waktu di sebuah studio dan mengerjakan aplikasi pendaftaran pasien untuk sebuah klinik (2027). Ia telah menandatangani perjanjian kerahasiaan (NDA) dengan studio. Seorang temannya meminta contoh data pasien asli untuk "bahan latihan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

membuat aplikasi".
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Sikap Rani yang paling tepat adalah ....

A. memberikannya karena temannya calon pengembang yang tidak akan menyalahgunakan data itu
B. memberikannya lewat layanan berbagi berkas pribadi yang katanya aman
C. menolak karena data pasien dilindungi perjanjian kerahasiaan
D. memberikan sebagian kecil saja agar pelanggarannya tidak terlalu besar
E. memberikannya asalkan temannya berjanji menghapusnya setelah dipakai

14. Tim sebuah studio memutuskan menyamakan cara menamai berkas proyek agar mudah dicari dan diurutkan. Mereka akan menamai berkas daftar harga produk yang disimpan sebagai lembar kerja dan tidak dikelola Git.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Nama berkas yang paling sesuai dengan prinsip penamaan yang baik adalah ....

A. daftar harga baru final.xlsx
B. DataHarga(2)_baru_sekali.xlsx
C. file1.xlsx
D. Daftar_Harga_FINAL_BENAR_revisi.xlsx
E. daftar_harga_v03_2027-02-10.xlsx

15. Sebuah studio memiliki enam baterai UPS bekas yang sudah tidak dapat menyimpan daya. Baterai itu menumpuk di bawah meja, di dekat tumpukan kardus bekas kemasan perangkat.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan yang paling tepat terhadap baterai tersebut adalah ....

A. menyimpannya di tempat kering lalu menyerahkannya kepada pengelola limbah berizin
B. membuangnya ke tempat sampah biasa agar ikut diangkut oleh petugas kebersihan gedung
C. membakarnya di halaman belakang agar volume limbahnya berkurang dengan cepat
D. membongkar sel baterai untuk mengambil logam di dalamnya lalu menjualnya per kilogram kepada pengepul
E. menguburnya di tanah kosong di belakang gedung agar tidak terlihat oleh siapa pun

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

16. Sebuah studio akan menjual laptop kantor lama (2027) yang masih berisi proyek klien, akun layanan awan yang masuk otomatis, dan lisensi perangkat lunak yang terpasang. Laptop itu telah dihapus dari daftar pemakaian.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat sebelum laptop dilepas adalah ....

A. menghapus semua berkas ke tempat sampah lalu mengosongkan tempat sampah itu sepenuhnya
B. memformat cepat laptop itu lalu langsung menjualnya kepada pembeli yang berminat
C. menghapus hanya berkas pribadi dan membiarkan data proyek klien tetap di dalamnya
D. mencadangkan data yang perlu, mencabut lisensi dan akun, lalu menghapus data dengan aman
E. menjual laptop lebih dulu lalu meminta pembeli menghapus sendiri data di dalamnya

17. Petugas inventaris sebuah studio (2027) memeriksa daftar aset berikut.

Kode	Aset	Kondisi	Penanggung jawab	Lokasi
LAP-01	Laptop	baik	Rani	Lab 1
LAP-02	Laptop	layar retak	(kosong)	Lab 1
PRJ-01	Proyektor	baik	Tono	Lab 2
PRJ-01	Proyektor	rusak berat	Tono	Gudang
SW-01	Lisensi editor kode	aktif sampai Des 2027	Rani	(tidak berlaku)

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang daftar tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Kode PRJ-01 dipakai dua proyektor sehingga salah satunya perlu diberi kode baru
☐ B. LAP-02 perlu ditetapkan penanggung jawabnya agar perbaikan dan pertanggungjawabannya jelas
☐ C. Proyektor rusak berat dapat dihapus dari daftar tanpa keputusan atau persetujuan siapa pun
☐ D. Masa berlaku lisensi SW-01 perlu dipantau karena lisensi termasuk aset digital
☐ E. Lisensi tidak perlu dicatat dalam inventaris karena tidak berwujud

18. Saat meninjau pull request di repositori privat tim (2027), Gita menemukan berkas pelanggan_backup.sql berisi nama, alamat, dan nomor telepon asli pelanggan. Pembuat pull request beralasan bahwa berkas itu perlu agar tim dapat menguji aplikasi dengan data nyata.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keputusan Gita yang paling tepat adalah ....

A. menyetujui karena repositorinya privat sehingga hanya anggota tim yang dapat melihat dan mengunduh berkasnya
B. menolak penggabungan, mengeluarkan berkas dari cabang dan riwayat, dan memakai data contoh yang disamarkan
C. menyetujui lalu meminta berkas dihapus lewat commit berikutnya setelah selesai dipakai untuk menguji
D. menyetujui dengan syarat nama berkas ditambahkan ke .gitignore setelah penggabungan dilakukan
E. menyetujui dengan syarat nama berkas diubah agar isinya tidak mudah ditebak orang lain

19. Gita akan menambahkan fitur "riwayat pesanan" pada aplikasi tim (2027). Kegiatan yang ia pertimbangkan adalah:
1. membuat cabang fitur dari main;
2. mengajukan pull request;
3. mengerjakan fitur dengan commit kecil secara berkala;
4. menggabungkan (merge) ke main setelah disetujui;
5. menarik perubahan terbaru dari main ke cabang fitur dan menjalankan uji;
6. ditinjau oleh rekan.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Urutan kegiatan yang sesuai dengan alur kerja Git yang sehat adalah ....

A. 3 – 1 – 5 – 2 – 6 – 4
B. 1 – 3 – 2 – 5 – 6 – 4
C. 1 – 3 – 5 – 6 – 2 – 4
D. 1 – 5 – 3 – 2 – 4 – 6
E. 1 – 3 – 5 – 2 – 6 – 4

20. Studio Kincir Hijau (2027) mencatat skema cadangan data proyeknya seperti pada tabel berikut.

Salinan	Cara menyimpan	Letak
1 (data asli)	Disk laptop kerja	Meja pemrogram di studio
2	Sinkronisasi otomatis ke penyimpanan awan	Di luar studio (awan)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Evaluasi yang paling tepat terhadap skema tersebut menurut aturan 3-2-1 adalah ....

A. skema sudah memenuhi aturan karena ada dua jenis media dan satu salinan berada di lokasi lain di luar studio
B. salinan di lokasi lain belum ada karena penyimpanan awan tidak termasuk lokasi lain menurut aturan itu
C. jenis media kurang, sehingga salinan awan perlu diganti dengan disk laptop kedua yang disimpan di studio
D. jumlah salinan baru dua dan sinkronisasi ikut menghapus berkas, sehingga perlu cadangan sungguhan di disk eksternal
E. jumlah salinan terlalu banyak, sehingga cukup satu salinan di penyimpanan awan tanpa salinan lain

21. Tim pembuat aplikasi absensi sekolah (2027) akan menulis README untuk repositori mereka agar anggota baru dapat memulai tanpa banyak bertanya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Isi yang tepat dimuat dalam README adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Nama dan tujuan proyek serta nama pengelola proyek
☐ B. Cara memasang dan menjalankan proyek beserta perintahnya
☐ C. Nilai asli kunci API supaya anggota baru tidak perlu bertanya
☐ D. Petunjuk mengisi pengaturan dengan contoh nilai palsu dan cara menjalankan uji
☐ E. Salinan seluruh isi kode program agar tidak perlu membuka repositori

22. Seorang pengelola studio kecil (2027) menyimpan cadangan data proyek pada hard disk yang tersambung ke komputer kerja yang sama dan tidak pernah mencoba memulihkannya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Cadangan di komputer yang sama dengan data asli tetap aman bila komputer itu rusak.	○	○
Cadangan perlu diuji pemulihannya secara berkala agar diketahui dapat dipakai.	○	○
Salinan di lokasi lain melindungi data dari bencana yang menimpa satu gedung.	○	○

23. Tim membuat repositori Git untuk aplikasi pemesanan makanan (2027). Isi folder proyek: berkas pengaturan rahasia (.env, berisi kunci API pembayaran), README.md, folder build/ berisi hasil kompilasi sementara, berkas pelanggan_backup.sql berisi data asli pelanggan, dan folder migrasi/ berisi skrip perubahan skema basis data secara berurutan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Berkas atau folder yang sebaiknya dimasukkan ke .gitignore adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. .env
☐ B. README.md
☐ C. build/
☐ D. pelanggan_backup.sql
☐ E. migrasi/

24. Versi terbaru pustaka pembaca berkas milik sebuah studio (2027) bernomor 1.9.3. Rilis berikutnya memuat tiga perubahan seperti pada tabel berikut.

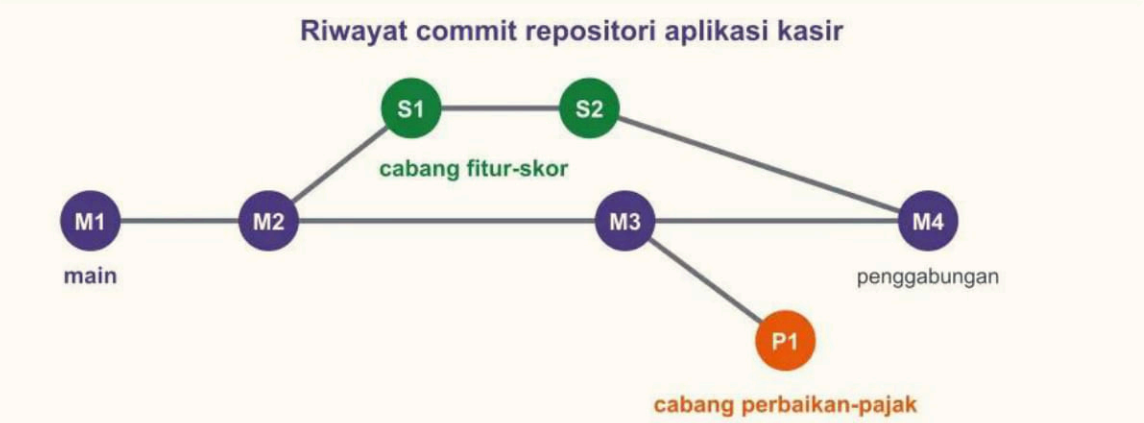
No	Perubahan	Keterangan
1	Perbaikan kesalahan hitung pada pembacaan angka	kompatibel dengan versi lama
2	Penambahan fitur ekspor ke PDF	kompatibel dengan versi lama
3	Penghapusan parameter lama sehingga kode pemakai lama harus diubah	tidak kompatibel

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Nomor versi rilis berikutnya menurut penomoran MAJOR.MINOR.PATCH adalah ....

A. 2.0.0
B. 1.10.4
C. 2.10.4
D. 1.10.0
E. 2.0.1

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

25. Tim aplikasi kasir (2027) melihat riwayat commit repositorinya seperti pada gambar. Lingkaran adalah commit, garis menghubungkan commit dengan commit induknya, dan M4 adalah commit hasil penggabungan (merge). Ketua tim hendak merilis dari cabang main.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perubahan yang **belum** ada di cabang main adalah ....



A. S1 dan S2 dari cabang fitur-skor
B. S2 dari fitur-skor dan P1 dari perbaikan-pajak
C. P1 dari cabang perbaikan-pajak
D. M3 dari cabang main
E. tidak ada, karena semua cabang sudah digabung ke main

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 2.1 · K3LH dan budaya kerja profesional	1, 2, 3, 5, 7, 8, 10, 13, 15	___ / 9	Ulangi hlm. 52
Sub bab 2.2 · Pengelolaan aset fisik dan digital	4, 6, 9, 11, 12, 16, 17	___ / 7	Ulangi hlm. 66
Sub bab 2.3 · Pengelolaan file proyek, source code, dan dokumentasi	14, 18, 19, 20, 21, 22, 23, 24, 25	___ / 9	Ulangi hlm. 79

Skor 70 % atau lebih: lanjut ke bab berikutnya. Skor 50–69 %: ulangi sub bab pada tabel yang kurang dari separuh benar. Skor di bawah 50 %: baca ulang bab ini dari awal. (Angka ini anjuran belajar, bukan prediksi nilai TKA.)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bab 3 · Teknologi Jaringan Komputer untuk PPLG

SMKPPLG-B3

Setelah mempelajari bab ini, kamu dapat:

- mengelola berkas, direktori, dan lingkungan pengembangan pada sistem operasi
- menentukan alamat IP dan konfigurasi jaringan dasar agar perangkat dapat saling terhubung
- menjelaskan fungsi layer TCP/IP dan protokol di dalamnya
- menganalisis mekanisme komunikasi klien–server, HTTP, port, dan API untuk menemukan kendala konektivitas

Peta bab

Sub bab	Judul
3.1	Berkas, direktori, dan lingkungan pengembangan pada sistem operasi
3.2	Konfigurasi jaringan dasar
3.3	Layer TCP/IP dan fungsi protokol
3.4	Mekanisme komunikasi jaringan



Bab 3 di LMS

Buka seluruh latihan bab ini di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-B3

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi

SMKPPLG-3.1

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menerapkan pengelolaan file dan direktori proyek pada sistem operasi** serta **menerapkan konfigurasi dasar lingkungan pengembangan**. Contoh soal resmi Pusmendik untuk bagian ini adalah pesan galat "Permission denied" pada berkas di folder web server lokal, dengan pertanyaan tindakan yang tepat.

Pada soal, kamu biasanya melihat potongan terminal: beberapa perintah berurutan, sebuah pesan galat, atau gambar susunan folder. Lalu kamu diminta menentukan **di mana berkas berada setelah perintah dijalankan**, **perintah mana yang tepat**, atau **langkah apa yang menyelesaikan masalah**. Soal paling sulit biasanya menyangkut **path relatif**: satu tanda `..` yang salah sudah membuat jawabannya berbeda.

Setelah mempelajari sub bab ini, kamu dapat:

- menelusuri path absolut dan relatif dari folder tempat kamu berada;
- memakai perintah dasar untuk membuat, memindah, menyalin, dan menghapus berkas dan folder di Linux dan Windows;
- membaca dan mengubah izin berkas (`chmod`) serta menangani pesan "Permission denied";
- menjelaskan fungsi variabel `PATH` dan membenahi galat "command not found";
- memakai virtual environment, `pip`, dan server lokal dengan benar, termasuk menangani "Address already in use".

Educational Institutions

◆ KONSEP INTI

Seorang pengembang menghabiskan banyak waktunya bukan di depan kode, melainkan di sekitarnya: memindahkan berkas, mengatur folder proyek, memasang alat, menjalankan program dari terminal. Kalau semua itu belum lancar, kode yang benar pun tidak akan jalan. Sub bab ini mengajarkan dasar-dasarnya.

1. Sistem operasi dan terminal

Sistem operasi (operating system, OS) adalah perangkat lunak yang mengelola komputer: menjalankan program, mengatur memori, mengelola berkas, dan menghubungkan perangkat keras. Contohnya **Windows**, **Linux**, dan **macOS**. Pengembang sering memakai **terminal** (antarmuka teks, disebut juga *command line* atau *shell*) karena banyak pekerjaan lebih cepat dan lebih mudah diulang dengan perintah: kamu mengetik perintah, menekan Enter, dan sistem

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

menjawab dengan teks.

Di Linux dan macOS, terminal biasanya memakai bahasa perintah **bash**. Di Windows ada **Command Prompt** dan **PowerShell**. Banyak perintah dasarnya mirip, tetapi namanya berbeda, seperti tabel pada bagian 3.

2. Sistem berkas, folder, dan path

Berkas disimpan dalam **struktur pohon** yang bertingkat: sebuah folder (direktori) dapat berisi berkas dan folder lain. Puncaknya disebut **root**: `/` di Linux dan macOS, atau huruf drive seperti `C:\` di Windows. Pemisah antarfolder adalah `/` di Linux dan `\` di Windows. Di Linux, huruf besar dan kecil dianggap berbeda (`Data` dan `data` adalah dua folder berbeda).

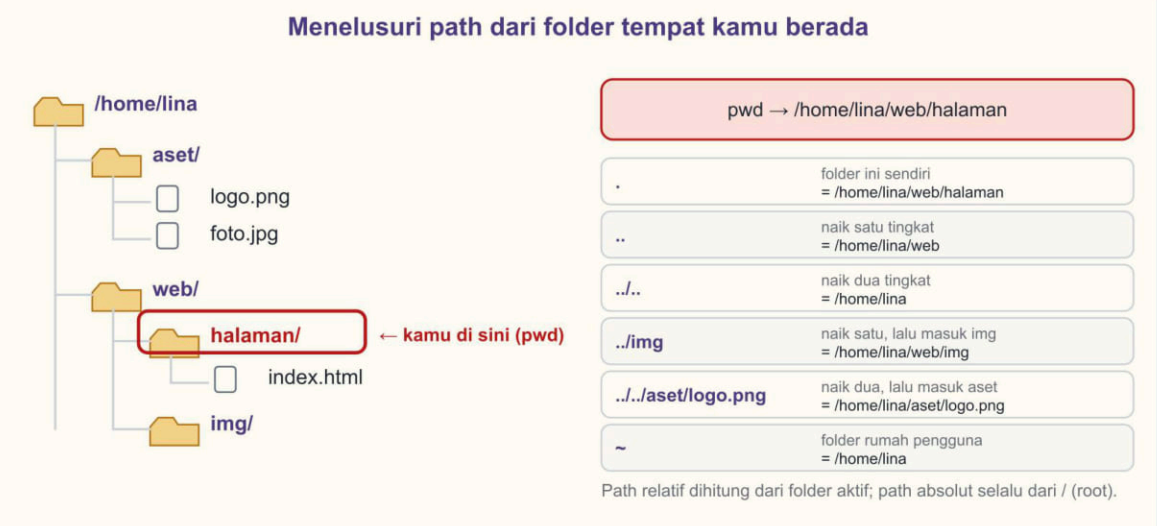
Path adalah alamat sebuah berkas atau folder. Ada dua jenis.

- **Path absolut** ditulis lengkap dari root, misalnya `/home/lina/aset/logo.png`. Alamatnya selalu sama dari mana pun kamu berada.
- **Path relatif** ditulis **dari folder tempat kamu berada sekarang** (folder aktif). Alamatnya berubah bila folder aktif berubah.

Ada tiga simbol penting untuk path relatif:

- `.` berarti folder aktif itu sendiri;
- `..` berarti folder **induk** (naik satu tingkat);
- `~` berarti folder rumah pengguna (misalnya `/home/lina`).

Folder aktif ditampilkan oleh perintah `pwd` (*print working directory*) di Linux atau `cd` tanpa argumen di Windows. Gambar 3.1 memperlihatkan cara membacanya.



Gambar 3.1 Folder aktif /home/lina/web/halaman dan path-path relatif yang dihitung darinya. Sumber: digambar oleh Tim Next Level Study Indonesia.

Pada gambar itu, kamu berada di `halaman`. Maka `..` adalah `/home/lina/web`, `../..` adalah `/home/lina`, dan `../../aset/logo.png` berarti "naik dua tingkat, masuk `aset`, ambil `logo.png`". Cara paling aman menelusuri path relatif adalah **memecahnya per segmen dari kiri ke kanan**, tiap `..` naik satu tingkat dan tiap nama masuk ke dalam folder itu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Satu hal yang sangat sering menjebak: **program membaca path relatif dari folder tempat program dijalankan, bukan dari folder tempat berkas program disimpan.** Bila `main.py` membuka `open("data/menu.csv")`, program itu mencari folder `data` di folder aktif saat dijalankan. Bila kamu menjalankannya dari folder lain, galat `FileNotFoundError` muncul walaupun berkas `menu.csv` ada dan tidak rusak. Perbaikannya adalah **pindah ke folder yang tepat** lebih dulu, bukan mengubah kode.

3. Perintah dasar mengelola berkas dan folder

Tujuan	Linux/macOS	Windows (Command Prompt)
Menampilkan folder aktif	`pwd`	`cd`
Melihat isi folder	`ls` (rinci: `ls -l`)	`dir`
Pindah folder	`cd folder`	`cd folder`
Membuat folder	`mkdir nama`	`mkdir nama`
Menyalin berkas	`cp asal tujuan`	`copy asal tujuan`
Memindah atau mengganti nama	`mv asal tujuan`	`move` / `ren`
Menghapus berkas	`rm nama`	`del nama`
Menampilkan isi berkas	`cat nama`	`type nama`

Hal yang perlu kamu pahami dari tiap perintah:

- **`cp asal tujuan`** membuat **salinan**; berkas asal tetap ada. Untuk folder, tambahkan `-r` (*recursive*): `cp -r folder tujuan/`.
- **`mv asal tujuan` memindah** berkas; di tempat asal berkas hilang. Perintah yang sama dipakai untuk **mengganti nama**: `mv lama.txt baru.txt` memindah ke nama baru di folder yang sama.
- **`rm nama`** menghapus **permanen** (tidak ada tempat sampah). Untuk folder, `rm -r`. Periksa dua kali sebelum menekan Enter.
- Tujuan berupa **folder** (diakhiri `/`) berarti berkas dimasukkan ke folder itu dengan nama yang sama; tujuan berupa **nama berkas** berarti berkas diberi nama itu.
- Tanda bintang * mewakili banyak karakter: `cp *.png gambar/` menyalin semua berkas berakhiran `.png`.

Perhatikan urutan perintah. Tiap `cd` mengubah folder aktif, sehingga path relatif pada perintah **berikutnya** dihitung dari folder yang baru. Menelusuri beberapa perintah berurutan berarti selalu menulis ulang folder aktif sesudah setiap `cd`.

4. Izin berkas dan "Permission denied"

Di Linux dan macOS, setiap berkas dan folder punya **izin** untuk tiga golongan pengguna: **pemilik**, **grup**, dan **lainnya**. Tiap golongan punya tiga kemungkinan izin:

- **r** (*read*): boleh membaca;

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **w** (*write*): boleh mengubah;
- **x** (*execute*): boleh menjalankan (untuk folder: boleh memasuki).

Perintah `ls -l` menampilkan izin dalam sembilan huruf setelah satu huruf jenis (`-` untuk berkas, `d` untuk folder):

```
$ ls -l
-rwxr-x--- 1 lina lina 1520 Mar  3 10:12 deploy.sh
-rw-r----- 1 lina lina  310 Mar  3 10:15 config.ini
```

Gambar 3.2 menguraikan `-rwxr-x---`: pemilik boleh membaca, menulis, dan menjalankan; grup boleh membaca dan menjalankan; yang lain tidak boleh apa-apa.



Gambar 3.2 Izin rwx diubah menjadi angka dengan nilai r = 4, w = 2, x = 1. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perintah **`chmod`** mengubah izin. Ada dua cara menulisnya.

- **Angka**: tiap golongan dijumlah dari r = 4, w = 2, x = 1. Jadi `rwx` = 7, `r-x` = 5, `rw-` = 6, `r--` = 4, dan `---` = 0. Contoh `chmod 750 deploy.sh` memberi pemilik 7, grup 5, lainnya 0.
- **Simbol**: `chmod u+x skrip.sh` menambah izin jalankan untuk pemilik (`u` = user, `g` = grup, `o` = lainnya).

Pesan **"Permission denied"** berarti pengguna yang menjalankan perintah **tidak punya izin** yang dibutuhkan (membaca, menulis, atau menjalankan berkas itu). Tindakan yang tepat adalah **memeriksa izin dengan `ls -l`, lalu mengubah izin atau pemilik seperlunya**, misalnya `chmod u+x` agar skrip bisa dijalankan, atau memperbaiki kepemilikan folder web server. Jawaban "beri izin 777 supaya semua bisa" dihindari karena membuka berkas untuk siapa saja dan berbahaya.

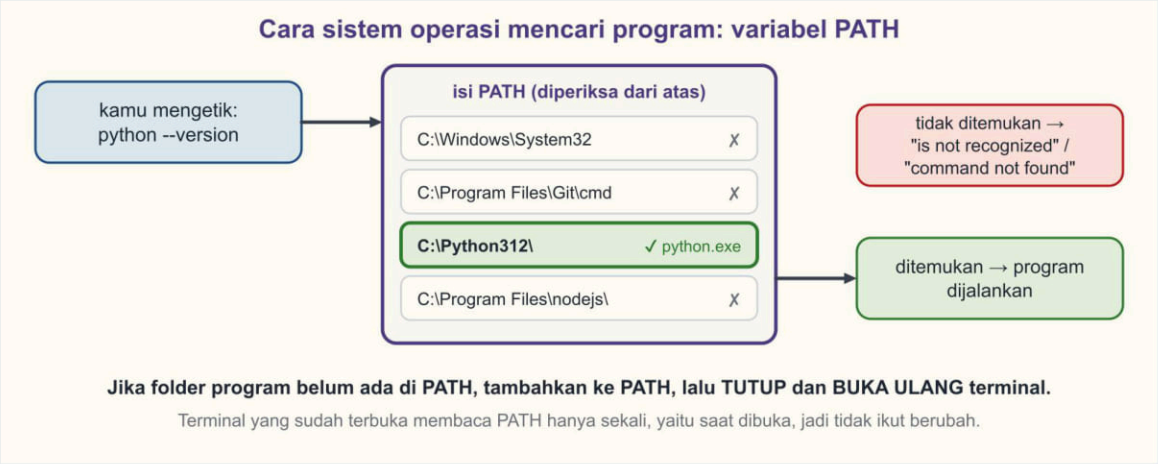
Windows mengatur izin lewat jendela **Properties → Security**, tetapi alasannya sama: pengguna harus punya izin yang sesuai.

5. Variabel lingkungan dan PATH

Variabel lingkungan adalah pasangan nama dan nilai yang disimpan sistem operasi untuk dipakai program. Yang paling penting bagi pengembang adalah **PATH**: daftar folder yang

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

diperiksa sistem, dari atas ke bawah, setiap kali kamu mengetik nama sebuah program (Gambar 3.3). Bila program ditemukan di salah satu folder itu, program dijalankan. Bila tidak, muncul pesan "command not found" (Linux) atau "is not recognized" (Windows).



Gambar 3.3 Sistem memeriksa folder-folder pada PATH untuk menemukan program. Sumber: digambar oleh Tim Next Level Study Indonesia.

Itulah sebabnya, ketika memasang Python, ada pilihan **"Add python.exe to PATH"**. Bila terlewat, perintah `python` tidak dikenali dari terminal. Cara membenahinya: tambahkan folder instalasi program ke PATH (atau pasang ulang dengan pilihan itu dicentang).

Satu detail yang sering diuji: **terminal yang sudah terbuka membaca PATH hanya sekali, saat dibuka.** Setelah PATH diubah, terminal yang sudah berjalan masih memakai daftar lama. Karena itu, **tutup lalu buka ulang terminal** (atau editor yang memuat terminal itu) agar perubahan terbaca. Menambahkan folder ke PATH sekali lagi atau memasang ulang program tidak ada gunanya bila terminalnya belum dibuka ulang.

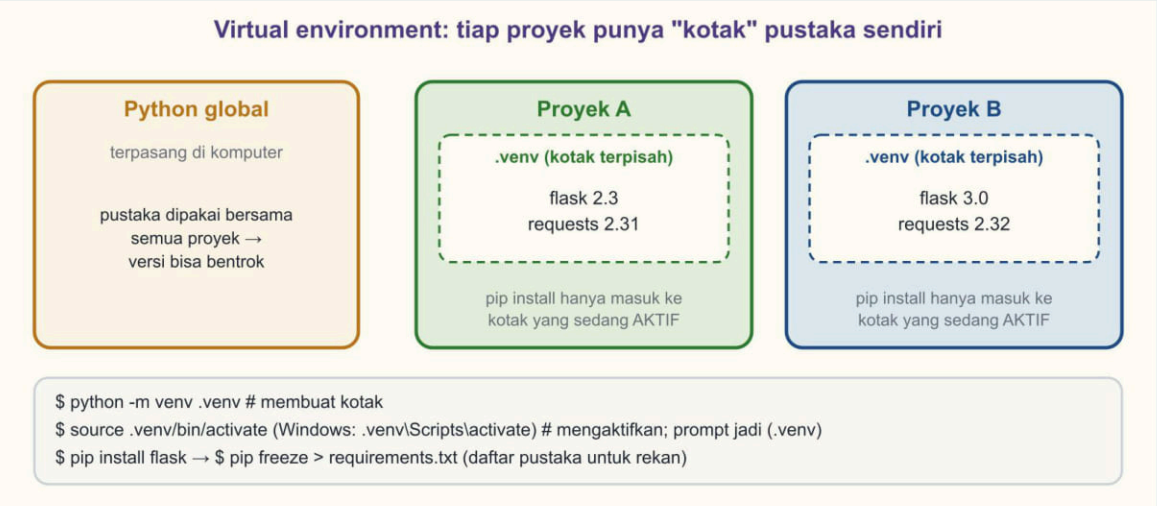
6. Memasang alat dan memeriksa versi

Lingkungan pengembangan terdiri atas beberapa alat: **interpreter atau kompilator** (misalnya Python), **editor atau IDE** (misalnya VS Code), **Git**, dan pustaka yang dibutuhkan proyek. Setelah memasang, kamu memeriksa bahwa alat itu terbaca dengan mencetak versinya: `python --version`, `git --version`. Bila versi muncul, alat itu ada di PATH dan siap dipakai. Pada beberapa sistem Linux, perintahnya `python3`, bukan `python`.

7. Virtual environment dan pip

Proyek Python memakai **pustaka** (library) tambahan yang dipasang dengan **`pip`**. Masalah muncul bila proyek A memerlukan pustaka versi 2 dan proyek B memerlukan versi 3, padahal keduanya memakai Python yang sama. Penyelesaiannya adalah **virtual environment (venv)**: sebuah folder terpisah yang berisi salinan lingkungan Python lengkap dengan pustakanya sendiri, satu untuk tiap proyek (Gambar 3.4).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 3.4 Setiap proyek memiliki kotak pustaka sendiri sehingga versi tidak bentrok. Sumber: digambar oleh Tim Next Level Study Indonesia.

Langkah memakainya:

```
$ python -m venv .venv                # membuat venv di folder .venv
$ source .venv/bin/activate             # mengaktifkan
(Windows: .venv\Scripts\activate)
(.venv) $ pip install flask             # memasang pustaka DI DALAM venv
(.venv) $ pip freeze > requirements.txt  # menyimpan daftar pustaka
(.venv) $ deactivate                    # keluar dari venv
```

Beberapa aturan yang harus dipahami:

- Setelah venv **aktif**, terminal menampilkan nama venv, misalnya `(.venv)`.
- **`pip install` memasang ke lingkungan yang sedang aktif.** Bila kamu menjalankan `pip install flask` **sebelum** mengaktifkan venv, `flask` terpasang di Python global, bukan di venv. Akibatnya, setelah venv diaktifkan, muncul `ModuleNotFoundError: No module named 'flask'`. Perbaikannya: dengan venv **aktif**, jalankan `pip install flask` lagi.
- **`requirements.txt`** mencatat pustaka proyek supaya rekan dapat memasangnya dengan `pip install -r requirements.txt`.
- Folder `.venv` **tidak** dimasukkan ke Git (cukup `requirements.txt`).

8. Server lokal dan port

Untuk menguji situs web atau aplikasi, pengembang sering menjalankan **server lokal** di komputernya sendiri, misalnya:

```
$ python3 -m http.server 8000
```

Perintah itu menayangkan isi folder aktif lewat alamat `http://localhost:8000`. **`localhost`** (alamat `127.0.0.1`) berarti komputer ini sendiri. Angka **8000** adalah **port**, pintu bernomor tempat sebuah program menunggu sambungan. Satu port pada satu waktu hanya dapat dipakai **satu program**.

Karena itu, bila kamu menjalankan perintah yang sama ketika server lama masih berjalan (misalnya di jendela terminal lain yang belum ditutup), muncul `OSError: [Errno 98] Address already in use`. Penyebabnya bukan Python yang rusak atau alamat IP yang

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

bentrok. Port 8000 **sedang dipakai** oleh proses sebelumnya. Cara menanganinya: **hentikan server lama** (tekan `Ctrl + C` di terminal tempatnya berjalan, atau hentikan proses yang memakai port itu) atau **pakai port lain**, misalnya `python3 -m http.server 8001`.

9. Galat umum dan cara membacanya

Pesan galat adalah petunjuk yang menunjuk ke satu penyebab. Gambar 3.5 merangkum lima yang paling sering muncul.

Pesan galat → penyebab → tindakan

Pesan galat	Penyebab umum	Tindakan yang tepat
Permission denied	hak akses berkas atau folder kurang	periksa dengan ls -l; ubah izin atau pemilik seperlunya (chmod, chown), bukan 777
command not found / is not recognized	program tidak ada di PATH	tambahkan folder program ke PATH, lalu buka ulang terminal
No such file or directory	path salah atau folder aktif keliru	periksa pwd dan ls; sesuaikan path atau pindah folder
ModuleNotFoundError	pustaka belum terpasang di lingkungan aktif	aktifkan venv yang benar, lalu pip install pustakanya
Address already in use	port sedang dipakai proses lain	hentikan proses lama atau jalankan di port lain

Gambar 3.5 Lima pesan galat umum beserta penyebab dan tindakan yang tepat. Sumber: digambar oleh Tim Next Level Study Indonesia.



Gambar 3.6 Peta konsep berkas, direktori, dan lingkungan pengembangan. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menyalin dan mengganti nama dengan path relatif

Lina berada di `/home/lina/web/halaman` (lihat Gambar 3.1). Ia ingin **menyalin** `/home/lina/aset/logo.png` ke folder `/home/lina/web/img`, tanpa menghapus berkas asal,

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

lalu mengganti nama salinannya menjadi `logo-sekolah.png`. Tulis perintahnya dengan path relatif. *(Kasus fiktif.)*

1. Tentukan folder aktif: `/home/lina/web/halaman`.
2. Hitung path berkas asal: `..` = `/home/lina/web`; `../..` = `/home/lina`; jadi `../../aset/logo.png` = `/home/lina/aset/logo.png` ✓.
3. Hitung path tujuan: `../img/` = `/home/lina/web/img/` ✓.
4. Karena berkas asal harus tetap ada, pakai `cp`, bukan `mv`:

```
$ cp ../../aset/logo.png ../img/
```

1. Ganti nama salinan: `mv` dengan tujuan nama baru di folder yang sama:

```
$ mv ../img/logo.png ../img/logo-sekolah.png
```

Jawaban: `cp ../../aset/logo.png ../img/` lalu `mv ../img/logo.png ../img/logo-sekolah.png`. Bila `mv` dipakai pada langkah 4, berkas `logo.png` di `aset` akan hilang.

Contoh 2 — Menentukan izin untuk tiga berkas

Tentukan perintah `chmod` untuk tiga kebutuhan berikut. *(Kasus fiktif.)*

- (a) `deploy.sh`: pemilik boleh membaca, mengubah, dan menjalankan; grup boleh membaca dan menjalankan; lainnya tidak boleh apa pun.
- (b) `config.ini`: pemilik boleh membaca dan mengubah; grup boleh membaca; lainnya tidak boleh apa pun.
- (c) Folder `unggahan`: pemilik boleh semuanya; grup dan lainnya boleh membaca isi dan memasuki folder, tetapi tidak boleh mengubah.

1. (a) Pemilik `rwx` = 4 + 2 + 1 = 7; grup `r-x` = 4 + 1 = 5; lainnya `---` = 0 → **750**: `chmod 750 deploy.sh`.
2. (b) Pemilik `rw-` = 4 + 2 = 6; grup `r--` = 4; lainnya 0 → **640**: `chmod 640 config.ini`.
3. (c) Pemilik 7; grup `r-x` = 5; lainnya `r-x` = 5 → **755**: `chmod 755 unggahan`. Izin `x` pada folder berarti boleh memasukinya.
4. Cek arah sebaliknya: `-rw-r-----` dibaca pemilik `rw-` (6), grup `r--` (4), lainnya `---` (0) → 640 ✓.

Jawaban: (a) 750, (b) 640, (c) 755.

Contoh 3 — "node is not recognized" setelah pemasangan

Setelah memasang Node.js di laptop Windows pada 2027, Arman membuka terminal dan mengetik `node --version`, tetapi muncul `'node' is not recognized as an internal or external command`. Ia ingat tidak mencentang pilihan penambahan ke `PATH` saat memasang. Tentukan langkah yang benar. *(Kasus fiktif.)*

1. Baca galatnya: sistem tidak menemukan program `node`. Penyebab: folder instalasinya **tidak ada di `PATH`**.
2. Bukan masalah program rusak, jadi memasang ulang bukan langkah pertama.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Tambahkan folder instalasi (misalnya `C:\Program Files\nodejs\`) ke variabel lingkungan `PATH` lewat pengaturan sistem.
4. **Tutup dan buka ulang terminal**, karena terminal yang sudah terbuka masih memakai `PATH` lama.
5. Periksa dengan `node --version`.

Jawaban: tambahkan folder Node.js ke `PATH`, lalu buka ulang terminal. Hanya menambahkan lagi atau memasang ulang tanpa membuka ulang terminal tidak menolong.

Contoh 4 — Pustaka tidak ditemukan padahal sudah dipasang

Bintang memasang pustaka dengan `pip install requests`, lalu mengaktifkan venv proyeknya dan menjalankan `python tugas.py`. Muncul `ModuleNotFoundError: No module named 'requests'`. Apa penyebab dan perbaikannya? *(Kasus fiktif.)*

1. Perhatikan urutan: `pip install` dijalankan **sebelum** venv diaktifkan.
2. Karena itu `requests` terpasang di Python global, sedangkan program sekarang berjalan di venv yang terpisah dan belum memuatnya.
3. Perbaikan: dengan venv **aktif** (prompt menampilkan `(.venv)`), jalankan `pip install requests`, lalu jalankan ulang programnya.

Jawaban: pasang pustaka ketika venv aktif.

❖ PETA KONSEP

Gambar 3.6 merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Path relatif dihitung dari folder aktif**; `..` naik satu tingkat; program membaca path relatif dari folder tempat ia **dijalankan**.
- **`cp` menyalin (asal tetap ada); `mv` memindah atau mengganti nama (asal hilang)**.
- **Izin `rwx` = 4-2-1**; `chmod 750` = pemilik 7, grup 5, lainnya 0. "Permission denied" diatasi dengan memeriksa dan mengubah izin seperlunya, bukan 777.
- **`PATH`** menentukan program yang dikenali; setelah mengubahnya, **buka ulang terminal**.
- **`pip install` masuk ke lingkungan yang sedang aktif**; aktifkan venv dulu.
- **"Address already in use"** = port sedang dipakai; hentikan proses lama atau pakai port lain.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal perintah dan path:

1. **Tulis folder aktif** di awal. Sesudah setiap `cd`, perbarui tulisan itu.
2. **Pecah path relatif per segmen**; tiap `..` naik satu tingkat.
3. **Tentukan tujuan perintah**: menyalin (`cp`, asal tetap), memindah atau mengganti nama (`mv`, asal hilang), atau menghapus (`rm`).
4. **Periksa tujuan**: berakhiran `/` berarti folder, tanpa itu berarti nama berkas.

Langkah menjawab soal galat:

1. **Baca pesan galat sampai habis** dan tentukan satu penyebabnya (izin, PATH, path, pustaka, port).
2. **Pilih tindakan yang menyasar penyebab itu.** Memasang ulang Python, mengganti IP, atau mematikan firewall jarang menjadi jawaban benar.

Jebakan yang sering dipasang:

- **`mv` dikira menyalin**, padahal berkas asal hilang.
- **Path relatif dihitung dari folder tempat berkas program berada**, bukan dari folder aktif tempat program dijalankan.
- **`../` dihitung kurang atau lebih satu tingkat** karena folder aktif tidak ditulis dulu.
- **`chmod 777` sebagai solusi** untuk semua masalah izin.
- **`PATH` diubah tetapi terminal tidak dibuka ulang.**
- **`pip install` sebelum venv aktif**, lalu bingung kenapa pustaka tidak ditemukan.
- **"Address already in use" dikira Python rusak atau IP bentrok.**
- **Rumus izin tertukar**: r = 4, w = 2, x = 1; `r-x` adalah 5, bukan 6.

> Contoh salah tafsir: pada folder aktif `/home/rudi/gim/level`, seorang siswa menulis `cp ../../aset/musik.mp3 .` untuk menyalin `/home/rudi/gim/aset/musik.mp3`. Ia menghitung dua tingkat naik, padahal `..` dari `level` sudah `/home/rudi/gim`. Dua tingkat naik menuju `/home/rudi`, tempat folder `aset` tidak ada. Path yang benar adalah `../aset/musik.mp3`.

● BEDAH SOAL PREDIKSI

Soal berikut memperlihatkan struktur folder dalam gambar dan sebuah program yang gagal membuka berkas relatif. Perhatikan dua hal yang harus kamu pisahkan: **dari mana program dijalankan**, dan **dari mana path relatif di dalam program dibaca** (bagian 2).

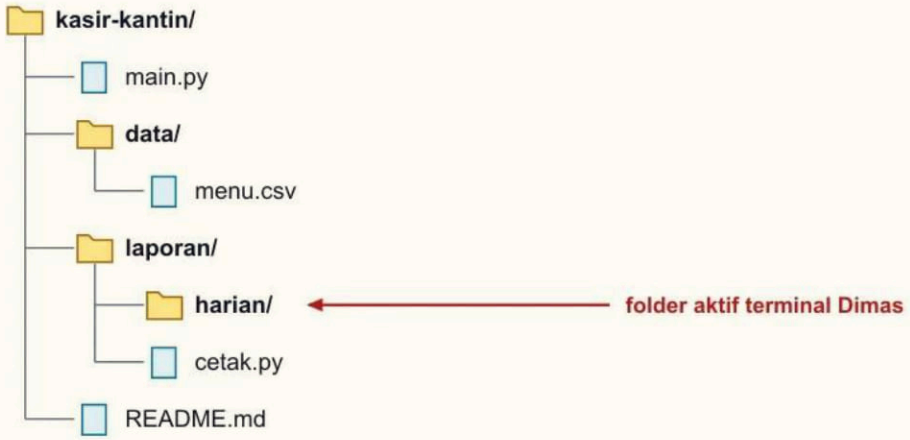
►. Gambar menunjukkan struktur folder proyek aplikasi kasir kantin. Di dalam `main.py`, daftar menu dibuka dengan perintah `open("data/menu.csv")`. Dimas menjalankan program dari folder yang ditunjuk panah pada gambar dan mendapat pesan berikut.

```
> python ../../main.py
FileNotFoundError: [Errno 2] No such file or directory: 'data/menu.csv'
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Berkas menu.csv sebenarnya ada dan tidak rusak.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Cara yang tepat agar program berjalan tanpa mengubah isi main.py adalah ....

Struktur folder proyek kasir-kantin di laptop Dimas (2027)



A. Menyalin main.py ke folder harian, lalu menjalankan `python main.py` di sana
B. Tetap di folder harian, lalu menjalankan `python ../main.py`
C. Pindah dengan `cd ../../data`, lalu menjalankan `python ../main.py`
D. Membuat folder data di dalam laporan, lalu menyalin menu.csv ke sana
E. Pindah dengan `cd ../..`, lalu menjalankan `python main.py`

Pembahasan **Kunci: E**

Konsep: Path relatif seperti "data/menu.csv" dicari mulai dari folder aktif (current working directory) terminal saat program dijalankan, bukan dari folder tempat berkas program berada.

Langkah:
1. Folder aktif: kasir-kantin/laporan/harian → program mencari kasir-kantin/laporan/harian/data/menu.csv (tidak ada).
2. Berkas ada di kasir-kantin/data/menu.csv → folder aktif harus kasir-kantin.
3. cd ../.. dari harian → laporan → kasir-kantin, lalu python main.py.

Analisis setiap opsi:
• **A.** ✗ Folder aktif tetap harian, jadi program mencari harian/data/menu.csv yang tidak ada.
• **B.** ✗ Jebakan: ../main.py menunjuk laporan/main.py yang tidak ada; berkas program pun tidak ditemukan.
• **C.** ✗ Jebakan: main.py memang ditemukan, tetapi folder aktif menjadi data sehingga dicari data/data/menu.csv.
• **D.** ✗ Folder aktif masih harian; program mencari harian/data/menu.csv, bukan laporan/data/menu.csv.
• **E.** ✓ Folder aktif menjadi kasir-kantin sehingga data/menu.csv ditemukan.

Jawaban: E.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk soal terminal, tulis folder aktif setelah setiap perintah. Untuk soal galat, tulis dulu satu kalimat penyebabnya.

1. Dina bekerja di terminal Linux. Perintah `pwd` menampilkan `/home/dina/proyek/kasir`. Ia ingin MENYALIN berkas `config.json` dari folder `/home/dina/template` ke folder tempat ia berada sekarang. Berkas di folder template tidak boleh berpindah atau terhapus karena masih dipakai proyek lain (2027).

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Perintah yang tepat adalah ....

A. `mv ../../template/config.json .`
B. `cp ../template/config.json .`
C. `cp config.json ../../template/`
D. `cp ../../template/config.json .`
E. `cp ~/proyek/template/config.json .`

2. Raka merapikan folder proyek gim di terminal Linux (2027). Awalnya berkas `musik.mp3` berada di `/home/raka/gim`. Ia menjalankan perintah berikut secara berurutan.

```
cd /home/raka/gim
mkdir aset
cd aset
mkdir suara
mv ../musik.mp3 suara/
cd ..
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Setelah perintah terakhir, letak berkas musik.mp3 dan folder aktif terminal adalah ....

A. berkas di /home/raka/gim/aset/suara; folder aktif /home/raka/gim
B. berkas di /home/raka/gim dan aset/suara; folder aktif /home/raka/gim
C. berkas di /home/raka/gim/aset/suara; folder aktif /home/raka
D. berkas di /home/raka/gim/suara; folder aktif /home/raka/gim
E. berkas di /home/raka/aset/suara; folder aktif /home/raka/gim

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Bayu memasang Python 3.12 di laptop Windows untuk proyek PKL (2027), tetapi lupa mencentang pilihan "Add python.exe to PATH". Di terminal VS Code yang sudah terbuka sejak pagi, perintah `python --version` menghasilkan pesan "python is not recognized". Bayu lalu menambahkan folder instalasi Python ke variabel PATH lewat menu Environment Variables dan menekan OK. Namun, di terminal VS Code yang sama pesan itu masih muncul.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah berikutnya yang paling tepat adalah ....

A. Memasang ulang Python karena berkas python.exe kemungkinan rusak
B. Menutup lalu membuka ulang VS Code agar terminal membaca PATH baru
C. Menambahkan folder Python ke PATH sekali lagi lalu menekan OK
D. Menjalankan VS Code sebagai administrator agar perintahnya dikenali
E. Mengganti nama python.exe menjadi python3.exe agar dikenali

4. Tim PKL memakai virtual environment (venv) terpisah untuk setiap proyek agar versi pustaka antarproyek tidak saling mengganggu; aturan tim: pustaka dipasang di dalam venv proyek (2027). Pagi tadi Wulan menjalankan `pip install flask` sebelum mengaktifkan venv. Setelah itu ia mengaktifkan venv, sehingga terminal menampilkan `(.venv)`, lalu menjalankan `python app.py` dan muncul pesan `ModuleNotFoundError: No module named 'flask'`.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat adalah ....

A. Memasang ulang Python versi terbaru, lalu membuat venv yang baru
B. Menyalin folder flask dari instalasi Python global ke folder proyek
C. Menjalankan pip install flask saat venv aktif, lalu menjalankan ulang
D. Menonaktifkan venv dan menjalankan semua proyek dengan Python global
E. Menambahkan folder .venv ke variabel PATH Windows secara permanen

5. Nadia menguji halaman web di laptop Linux-nya dengan perintah `python3 -m http.server 8000` (2027). Kemarin ia menjalankan perintah yang sama di jendela terminal lain, lalu laptop hanya ditutup layarnya tanpa dimatikan, sehingga terminal itu masih berjalan. Hari ini perintah tersebut gagal dengan pesan `OSError: [Errno 98] Address already in use`.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat adalah ....

A. Memasang ulang Python karena modul http.server diduga rusak
B. Mengganti alamat IP laptop agar port 8000 kosong kembali
C. Mematikan firewall laptop agar port 8000 dapat dibuka lagi
D. Menghentikan server lama di terminal itu, atau memakai port lain
E. Menghapus folder proyek, lalu menyalinnya ulang dari repositori

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kunci singkat: 1. D 2. A 3. B 4. C 5. D

✔ PERIKSA DIRI

- Mengapa `mv` tidak cocok untuk menyalin berkas yang masih dipakai proyek lain, dan perintah apa yang tepat?
- Mengapa terminal yang sudah terbuka tidak mengenali program setelah PATH diubah?
- **Yang harus kuingat:** path relatif dihitung dari folder aktif (`..` naik satu tingkat); `cp` menyalin dan `mv` memindah; izin rwx = 4-2-1; setelah mengubah PATH buka ulang terminal; pip memasang ke lingkungan aktif; satu port hanya untuk satu program.



Latihan interaktif di LMS

Sub bab 3.1: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-3.1

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 3.2 · Konfigurasi jaringan dasar

SMKPPLG-3.2

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **mengimplementasikan konfigurasi jaringan dasar agar sistem terhubung dalam lingkungan pengembangan**. Cakupannya: perangkat jaringan dan fungsinya, alamat IP, subnet mask, gateway, DNS, serta kemampuan memeriksa mengapa dua perangkat tidak dapat saling terhubung. Contoh soal resmi Pusmendik untuk bagian ini memperlihatkan gambar topologi: sebuah server aplikasi, penyimpanan berkas, dan basis data. Kamu diminta memilih alamat IP untuk satu titik agar berada dalam satu jaringan, dan salah satu pilihannya adalah alamat yang tidak valid karena memuat angka 300.

Ini salah satu sub bab dengan persentase soal sulit tertinggi. Kesulitannya jarang berasal dari konsep yang rumit. Kesulitannya terletak pada **ketelitian**: satu angka pada alamat, satu gateway yang berbeda, atau satu perangkat yang ternyata tersambung ke Wi-Fi lain. Bacalah tabel dan gambar dengan teliti, bandingkan alamat per oktet, dan jangan mengandalkan kesan.

Setelah mempelajari sub bab ini, kamu dapat:

- membedakan fungsi switch, router, dan access point;
- menentukan apakah dua perangkat berada pada satu jaringan dari alamat IP dan subnet mask-nya;
- menentukan ID jaringan, rentang host, dan alamat broadcast pada jaringan /24;
- memilih alamat IP statis yang valid, belum dipakai, dan tidak bentrok dengan rentang DHCP;
- menjelaskan peran gateway, DNS, dan DHCP serta menelusuri penyebab perangkat tidak terhubung.

◆ KONSEP INTI

Jaringan komputer adalah sekumpulan perangkat yang saling terhubung agar dapat bertukar data dan berbagi sumber daya. Bagi pengembang, jaringan menentukan apakah laptop dapat membuka server uji, apakah aplikasi dapat mengakses basis data, dan apakah ponsel yang sedang diuji dapat memanggil layanan di komputer. Sub bab ini membahas bagaimana perangkat-perangkat itu disambungkan dan diberi alamat.

1. Jenis jaringan

Menurut jangkauannya, jaringan dibedakan menjadi:

- **LAN** (*Local Area Network*): jaringan di satu lokasi terbatas seperti lab, kantor, atau rumah.

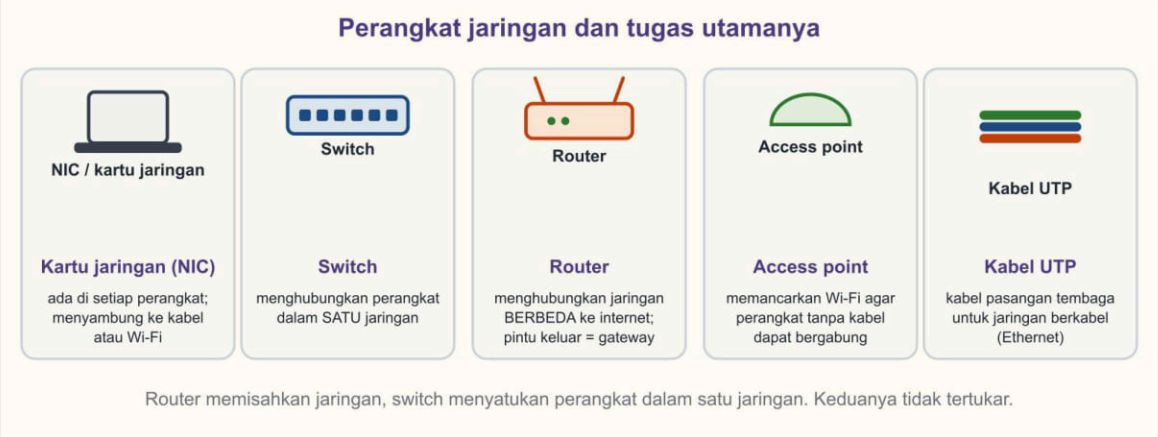
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **WAN** (*Wide Area Network*): jaringan yang menghubungkan lokasi yang berjauhan, misalnya dua cabang perusahaan di kota berbeda.
- **Internet**: jaringan dari jaringan yang menghubungkan seluruh dunia.

Lingkungan pengembangan sehari-hari biasanya LAN: laptop pengembang, server uji, dan beberapa ponsel uji berada pada jaringan yang sama.

2. Perangkat jaringan dan tugasnya

Gambar 3.7 memperlihatkan perangkat yang paling sering ditemui.



Gambar 3.7 Perangkat jaringan dan tugas utamanya. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Kartu jaringan (NIC)** ada di setiap perangkat yang terhubung. Ia menyambungkan perangkat ke kabel atau ke Wi-Fi.
- **Kabel UTP** menyalurkan data pada jaringan berkabel (Ethernet).
- **Switch** menghubungkan banyak perangkat **dalam satu jaringan**. Ia meneruskan data ke perangkat tujuan di dalam jaringan itu.
- **Router** menghubungkan **jaringan yang berbeda**, termasuk jaringan lab dengan internet. Alamat router yang dipakai perangkat sebagai pintu keluar disebut **gateway**.
- **Access point** memancarkan sinyal Wi-Fi agar perangkat tanpa kabel dapat bergabung ke jaringan.
- **Hub** adalah pendahulu switch yang meneruskan data ke semua port; kini jarang dipakai.

Cara mengingat: switch **menyatukan** perangkat dalam satu jaringan; router **menghubungkan** satu jaringan dengan jaringan lain. Keduanya tidak boleh tertukar.

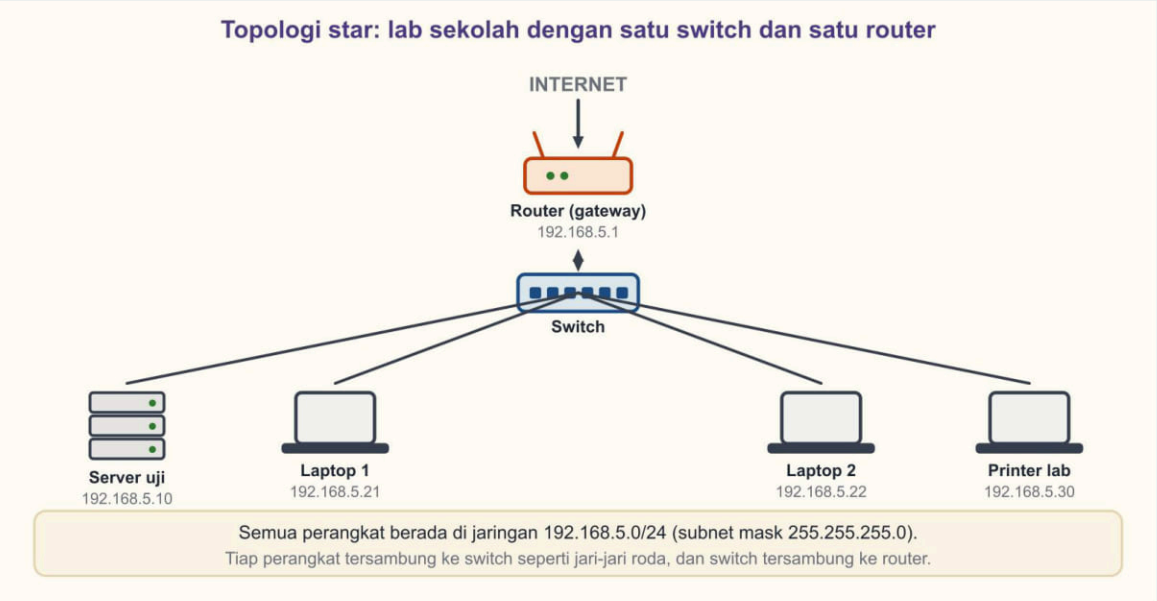
Bagi pengembang, memahami perangkat ini membantu membaca laporan masalah dengan cepat. Bila hanya satu laptop yang tidak terhubung, curigai kabel, NIC, atau pengaturan laptop itu. Bila semua perangkat di satu switch bermasalah, curigai switch atau kabel ke router. Bila semua perangkat bisa saling membuka tetapi tidak ada yang dapat membuka internet, curigai router atau sambungan keluarnya.

3. Topologi star

Topologi adalah cara perangkat disusun dan disambungkan. Yang paling umum di lab dan kantor adalah **topologi star** (bintang): setiap perangkat tersambung ke satu pusat, yaitu switch,

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

seperti jari-jari roda. Switch kemudian tersambung ke router untuk keluar ke internet (Gambar 3.8). Kelebihannya, kerusakan satu kabel hanya memutus satu perangkat, dan menambah perangkat cukup dengan menancapkan satu kabel lagi.



Gambar 3.8 Topologi star pada jaringan lab dengan satu switch dan satu router. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Alamat IP versi 4

Agar data sampai ke perangkat yang tepat, setiap perangkat diberi **alamat IP**. Alamat **IPv4** terdiri atas **empat angka**, disebut **oktet**, yang dipisahkan titik. Setiap oktet berkisar **0 sampai 255**. Contoh yang valid: `192.168.20.30`. Contoh yang **tidak valid**: `192.168.20.256` (oktet 256 melebihi 255), `192.168.300.5` (300 melebihi 255), dan `192.168.20` (hanya tiga oktet).

Beberapa golongan alamat perlu kamu kenal:

- **Alamat privat**, dipakai di jaringan lokal dan tidak dapat dirutekan langsung di internet: `10.0.0.0`–`10.255.255.255`, `172.16.0.0`–`172.31.255.255`, dan `192.168.0.0`–`192.168.255.255`. Jaringan lab hampir selalu memakai alamat privat.
- **Alamat publik**: dipakai di internet, diberikan oleh penyedia layanan.
- **Alamat loopback `127.0.0.1`** (nama `localhost`): selalu berarti **komputer ini sendiri**. Perangkat lain tidak dapat dijangkau lewat alamat ini.

5. Subnet mask: membagi bagian jaringan dan bagian host

Setiap alamat IP terdiri atas dua bagian: **bagian jaringan** (menunjuk jaringan mana) dan **bagian host** (menunjuk perangkat mana di jaringan itu). **Subnet mask** menunjukkan di mana batas keduanya. Dalam bentuk biner, angka **1** pada mask menandai bagian jaringan dan angka **0** menandai bagian host (Gambar 3.9).



Gambar 3.9 Dengan mask 255.255.255.0, tiga oktet pertama adalah bagian jaringan dan oktet terakhir adalah bagian host. Sumber: digambar oleh Tim Next Level Study Indonesia.

Mask yang paling sering dipakai adalah **255.255.255.0**, ditulis juga **/24** (24 bit angka 1). Artinya, **tiga oktet pertama** adalah bagian jaringan dan **oktet terakhir** adalah bagian host. Dari sini kamu mendapat aturan terpenting:

> **Dua perangkat berada pada satu jaringan bila bagian jaringan alamatnya sama.** Pada mask /24, tiga oktet pertama harus **sama persis**.

Contoh: `192.168.20.30` dan `192.168.20.77` satu jaringan (192.168.20). `192.168.20.30` dan `192.168.2.30` **bukan** satu jaringan (192.168.**20** berbeda dengan 192.168.**2**). Telitilah pada

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

oktet ketiga, karena `20` dan `2` mudah terbaca sama.

Pada jaringan /24, misalnya `192.168.20.0/24`:

- **ID jaringan** adalah alamat dengan bagian host semuanya 0: `192.168.20.0`. Alamat ini menamai jaringan dan tidak diberikan kepada perangkat.
- **Alamat broadcast** adalah alamat dengan bagian host semuanya 1 (255): `192.168.20.255`. Alamat ini dipakai untuk menyapa semua perangkat dan tidak diberikan kepada perangkat.
- **Rentang host** adalah `192.168.20.1` sampai `192.168.20.254`, **254 alamat** yang dapat dipakai perangkat (256 dikurangi 2).

Mask lain juga ada. Pada `255.255.0.0` (/16), **dua oktet pertama** yang menjadi bagian jaringan, sehingga `10.0.1.5` dan `10.0.2.5` satu jaringan. Pada mask `255.255.255.0`, kedua alamat itu berbeda jaringan. Hasilnya bergantung pada mask, jadi selalu baca mask-nya.

6. Gateway, DNS, dan DHCP

Selain alamat IP dan mask, sebuah perangkat biasanya memerlukan tiga hal lain.

- **Gateway** adalah alamat router yang dipakai perangkat untuk **keluar dari jaringannya**: ke jaringan lain atau ke internet. Gateway harus berada **di jaringan yang sama** dengan perangkat (misalnya `192.168.20.1` untuk perangkat `192.168.20.x`). **Dalam satu jaringan, gateway tidak dipakai**: dua laptop satu jaringan tetap bisa saling berhubungan walaupun gateway salah satunya keliru. Yang rusak hanyalah akses ke luar jaringan, seperti internet.
- **DNS** (*Domain Name System*) menerjemahkan nama seperti `smkbudi.sch.id` menjadi alamat IP. Tanpa DNS yang benar, nama situs tidak dapat dibuka walaupun internet tersambung; alamat IP-nya langsung masih bisa.
- **DHCP** adalah layanan yang **memberi alamat IP otomatis** kepada perangkat yang bergabung ke jaringan. Layanan ini memakai **rentang alamat tertentu** (misalnya `.100`–`.200`). Alamat yang diberikan manual disebut **alamat statis**.

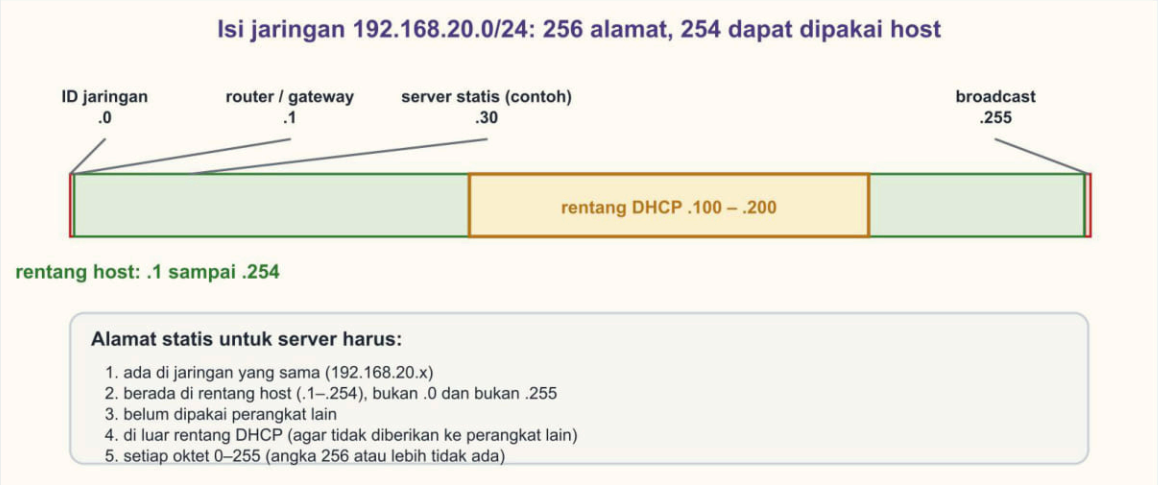
Server, printer, dan perangkat yang harus selalu dapat dijangkau di alamat yang sama diberi **alamat statis**. Perangkat biasa seperti laptop dan ponsel memakai **DHCP** agar tidak perlu diatur satu per satu. **Konflik IP** terjadi bila dua perangkat memakai alamat yang sama, dan salah satunya (atau keduanya) kehilangan sambungan. Bila sebuah perangkat dibiarkan memakai DHCP tetapi layanan DHCP tidak menjawab, sistem sering memberi alamat darurat berawalan **169.254**. Alamat itu tanda bahwa perangkat gagal mendapat alamat dari jaringan, sehingga yang perlu diperiksa adalah layanan DHCP atau sambungan ke jaringannya, bukan pengaturan aplikasi.

Wi-Fi dan jaringan tamu. Sebuah access point dapat memancarkan lebih dari satu nama Wi-Fi (SSID), dan tiap nama dapat terhubung ke jaringan yang berbeda. Banyak sekolah dan kantor memisahkan **Wi-Fi tamu** dari **jaringan internal**: perangkat di Wi-Fi tamu bisa membuka internet tetapi tidak dapat mengakses server internal. Karena itu, sebelum menyimpulkan ada kerusakan, pastikan perangkat tersambung ke Wi-Fi yang tepat.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

7. Memilih alamat statis yang benar

Gambar 3.10 merangkum syarat sebuah alamat statis untuk server pada jaringan `192.168.20.0/24`.



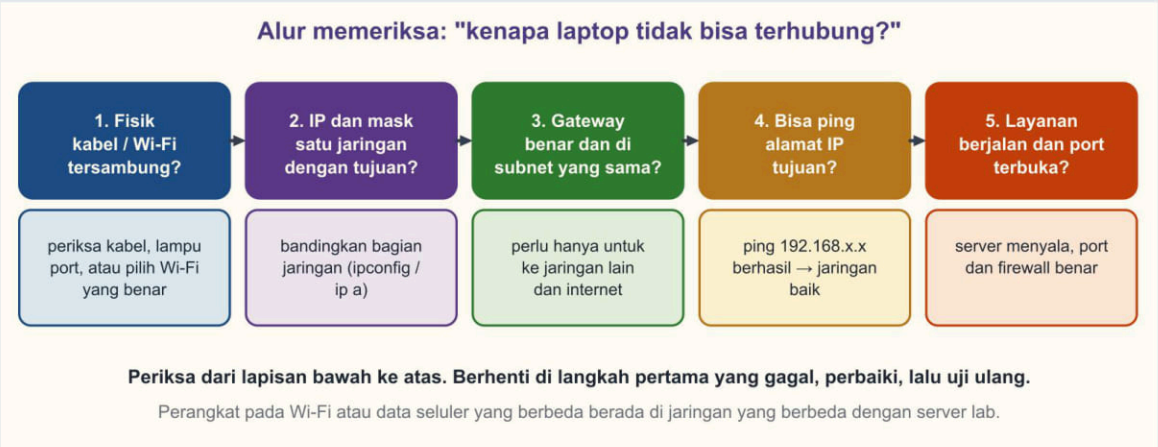
Gambar 3.10 Isi jaringan /24 dan lima syarat alamat statis. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Satu jaringan**: bagian jaringan sama dengan perangkat lain (192.168.20.x).
2. **Di dalam rentang host**: bukan `.0` dan bukan `.255`.
3. **Belum dipakai**: periksa daftar alamat yang sudah terpasang.
4. **Di luar rentang DHCP**: kalau tidak, DHCP kelak dapat memberikan alamat yang sama kepada perangkat lain.
5. **Valid**: setiap oktet 0 sampai 255.

Gateway dan mask harus **mengikuti jaringan itu** (gateway `192.168.20.1`, mask `255.255.255.0`).

8. Memeriksa konektivitas

Bila perangkat tidak dapat terhubung, periksa secara **berurutan dari bawah ke atas** (Gambar 3.11).



Gambar 3.11 Alur memeriksa mengapa sebuah perangkat tidak dapat terhubung. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. **Fisik**: kabel tertancap, lampu port menyala, atau Wi-Fi yang dipilih benar.
2. **IP dan mask**: lihat konfigurasi dengan `ipconfig` (Windows) atau `ip a` (Linux), lalu bandingkan **bagian jaringan** dengan tujuan.
3. **Gateway**: perlu benar hanya bila tujuan di jaringan lain atau internet.
4. **Ping**: `ping 192.168.20.30` menguji apakah perangkat tujuan menjawab. Berhasil berarti jaringan baik.
5. **Layanan dan port**: server berjalan dan port-nya terbuka, bukan diblokir firewall (dibahas di sub bab 3.4).

Contoh keluaran perintah yang sering dipakai (tampilan dapat sedikit berbeda antarsistem):

```
C:\> ipconfig
   IPv4 Address. . . . . . . . : 192.168.20.45
   Subnet Mask . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . : 192.168.20.1

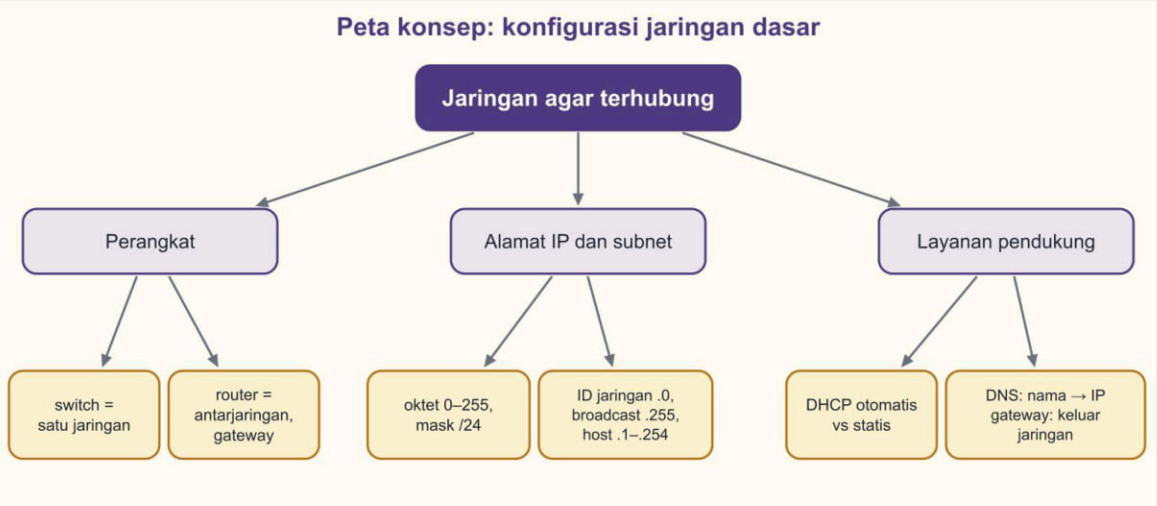
C:\> ping 192.168.20.30
Reply from 192.168.20.30: bytes=32 time=2ms TTL=64
```

Baris **IPv4 Address**, **Subnet Mask**, dan **Default Gateway** inilah yang dibandingkan dengan perangkat tujuan. Balasan "Reply from" berarti perangkat tujuan menjawab, sedangkan "Request timed out" berarti tidak ada jawaban: bisa karena beda jaringan, perangkat mati, atau firewall menolak ping.

Pengayaan. Mask tidak selalu 255.255.255.0. Notasi **CIDR** menulis jumlah bit jaringan setelah garis miring: `/24` memberi 254 host, `/25` memberi 126 host, `/26` memberi 62 host, dan `/16` memberi 65.534 host. Semakin besar angka setelah garis miring, semakin kecil jaringannya. Soal TKA pada sub bab ini mengandalkan /24 dan /16.

Satu jebakan penting: **perangkat di Wi-Fi atau jaringan yang berbeda tidak dapat membuka server di jaringan lab.** Ponsel yang memakai data seluler atau Wi-Fi tamu berada di jaringan lain, sehingga alamat server lab tidak dapat dijangkau. Solusinya menyambungkan ponsel ke jaringan yang sama dengan server, bukan mengganti port atau mengubah alamat server. Dan alamat `localhost` di ponsel menunjuk ponsel itu sendiri, bukan server.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 3.12 Peta konsep konfigurasi jaringan dasar. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Satu jaringan atau bukan?

Tentukan apakah pasangan perangkat berikut berada pada satu jaringan. *(Data fiktif.)*

- (a) `192.168.7.15` dan `192.168.7.200`, mask `255.255.255.0`.
- (b) `192.168.7.15` dan `192.168.70.15`, mask `255.255.255.0`.
- (c) `10.0.1.5` dan `10.0.2.5`, mask `255.255.255.0`.
- (d) `10.0.1.5` dan `10.0.2.5`, mask `255.255.0.0`.

1. Mask `255.255.255.0` berarti tiga oktet pertama harus sama; mask `255.255.0.0` berarti dua oktet pertama.
2. (a) Tiga oktet pertama `192.168.7` sama → **satu jaringan**.
3. (b) Oktet ketiga `7` dan `70` berbeda → **berbeda jaringan**.
4. (c) Oktet ketiga `1` dan `2` berbeda → **berbeda jaringan**.
5. (d) Dua oktet pertama `10.0` sama → **satu jaringan**.

Jawaban: (a) ya, (b) tidak, (c) tidak, (d) ya. Alamat yang sama dapat satu jaringan atau tidak, bergantung pada mask.

Contoh 2 — Memilih alamat statis untuk server

Jaringan kantor sebuah studio (2027): `172.16.4.0/24`, gateway `172.16.4.1`, rentang DHCP `172.16.4.100`–`172.16.4.200`. Alamat yang sudah dipakai: `.1` (router), `.10` (printer), `.20` (penyimpanan jaringan). Sebuah server baru perlu alamat statis. Mana yang tepat dari lima calon berikut? *(Data fiktif.)*

- A. `172.16.4.10`
- B. `172.16.4.150`
- C. `172.16.5.30`

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- D. `172.16.4.30`
- E. `172.16.4.256`

1. Periksa syarat satu per satu.
2. A: berada di jaringan benar dan rentang host, tetapi **sudah dipakai printer** → tolak (konflik IP).
3. B: **di dalam rentang DHCP** (.100–.200) → tolak.
4. C: bagian jaringan `172.16.5` **berbeda** dari `172.16.4` → tolak.
5. D: jaringan benar, di rentang host, belum dipakai, di luar rentang DHCP, oktet valid → **lolos**.
6. E: oktet `256` **tidak valid** → tolak.

Jawaban: D, `172.16.4.30`.

Contoh 3 — Laptop tidak dapat membuka server uji

Pada 2027, Wira tidak dapat membuka `http://192.168.3.40:8000` dari laptopnya. Hasil `ipconfig` dan konfigurasi server tercatat sebagai berikut. *(Data fiktif.)*

Perangkat	Alamat IP	Subnet mask	Gateway
Laptop Wira	192.168.30.40	255.255.255.0	192.168.30.1
Server uji	192.168.3.40	255.255.255.0	192.168.3.1

1. Periksa fisik: laptop menampilkan Wi-Fi tersambung; lanjut ke langkah berikutnya.
2. Bandingkan bagian jaringan dengan mask /24: laptop `192.168.30`, server `192.168.3`. **Berbeda** (30 dan 3).
3. Gateway masing-masing sudah sesuai jaringannya sendiri, jadi bukan itu masalahnya.
4. Kesimpulan: laptop dan server **berada di jaringan berbeda**, sehingga tidak dapat saling menjangkau tanpa rute antarjaringan.
5. Perbaikan: sambungkan laptop dan server ke jaringan yang sama, misalnya memberi server alamat di jaringan `192.168.30.x` yang valid dan belum dipakai.

Jawaban: penyebabnya perbedaan jaringan (192.168.30 dan 192.168.3). Mengganti port atau memasang ulang aplikasi tidak menyelesaikannya.

Contoh 4 — Gateway salah, apa akibatnya?

Pada jaringan `192.168.30.0/24`, komputer C memiliki alamat `192.168.30.60`, mask `255.255.255.0`, dan gateway `192.168.31.1`. Server uji berada di `192.168.30.10`. (a) Dapatkah C membuka server uji? (b) Dapatkah C membuka situs di internet? *(Data fiktif.)*

1. (a) C dan server uji satu jaringan (192.168.30), jadi mereka saling berhubungan **tanpa gateway** → **dapat**.
2. (b) Untuk internet, C harus lewat gateway. Gateway `192.168.31.1` berada di jaringan lain (31) sehingga **tidak dapat dijangkau** → **tidak dapat**.

Jawaban: (a) ya; (b) tidak sampai gateway diperbaiki menjadi `192.168.30.1`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

❖ PETA KONSEP

Gambar 3.12 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Switch menyatukan perangkat dalam satu jaringan; router menghubungkan jaringan yang berbeda** dan menjadi gateway.
- **IPv4** = empat oktet **0–255**; angka 256 atau lebih tidak valid.
- **Mask /24 (255.255.255.0)**: tiga oktet pertama = jaringan; **dua perangkat satu jaringan bila bagian jaringan sama**.
- **ID jaringan** `.0` dan **broadcast** `.255` tidak diberikan kepada perangkat; **host** `.1–.254` (254 alamat).
- **Gateway hanya diperlukan untuk keluar dari jaringan**; **DNS** mengubah nama menjadi IP; **DHCP** memberi alamat otomatis, sedangkan server memakai **alamat statis di luar rentang DHCP**.
- Periksa konektivitas **berurutan**: fisik → IP dan mask → gateway → ping → layanan dan port.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal alamat dan jaringan:

1. **Tulis mask-nya** dan tentukan oktet mana yang menjadi bagian jaringan.
2. **Bandingkan bagian jaringan per oktet** antara perangkat yang dibandingkan.
3. **Untuk memilih alamat statis**, periksa lima syarat satu per satu: jaringan, rentang host, belum dipakai, di luar DHCP, oktet valid.
4. **Untuk soal "tidak dapat terhubung"**, tentukan di langkah mana urutan pemeriksaan pertama kali gagal.

Jebakan yang sering dipasang:

- **Oktet ketiga terbaca sama** padahal berbeda (`20` dan `2`, atau `7` dan `70`).
- **Alamat bernilai 256 atau lebih** dikira valid.
- **Alamat `.0` atau `.255`** diberikan kepada perangkat.
- **Alamat di dalam rentang DHCP** dipilih sebagai alamat statis.
- **Gateway salah dikira menghalangi akses satu jaringan.** Gateway hanya berpengaruh pada akses keluar jaringan.
- **Mengubah port atau mematikan firewall** ketika masalahnya adalah beda jaringan.
- **Ponsel di data seluler atau Wi-Fi lain** dikira satu jaringan dengan server.
- **`localhost` di perangkat lain** dikira menunjuk ke server.
- **Switch dan router tertukar.**

> Contoh salah tafsir: pada mask `255.255.255.0`, seorang siswa menyimpulkan bahwa `192.168.5.10` dan `192.168.6.10` satu jaringan karena "angka depannya sama dan hanya satu angka yang berbeda". Satu oktet yang berbeda pada bagian jaringan sudah cukup membuat keduanya berbeda jaringan (5 dan 6).

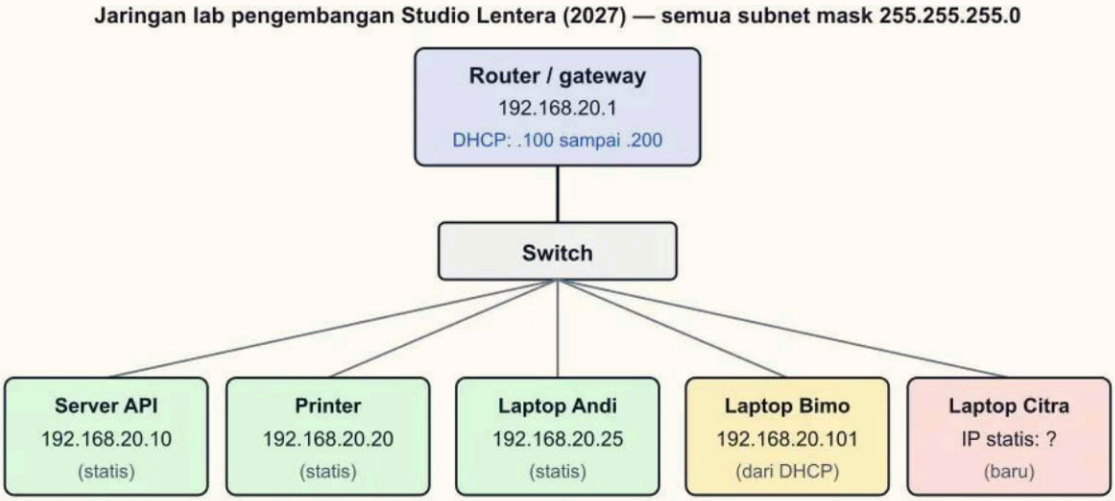
Educational Institutions

• BEDAH SOAL PREDIKSI

Soal berikut menyajikan gambar jaringan lab dan memintamu memilih alamat statis untuk sebuah laptop yang akan menjadi server. Perhatikan bagaimana kelima syarat alamat statis (bagian 7) menyingkirkan empat pilihan satu per satu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Gambar menunjukkan jaringan lab pengembangan Studio Lentera. Citra akan menjalankan server uji aplikasi di laptop barunya, sehingga laptop itu harus memakai alamat IP statis yang dapat diakses semua perangkat di gambar dan tidak bentrok dengan perangkat lain, termasuk perangkat yang kelak mendapat alamat dari DHCP. Subnet mask 255.255.255.0 dan gateway 192.168.20.1 sudah diisi.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Alamat IP yang paling tepat untuk laptop Citra adalah ....



A. 192.168.20.20
B. 192.168.2.30
C. 192.168.20.150
D. 192.168.20.30
E. 192.168.20.256

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Pembahasan **Kunci: D**

Konsep: Dengan subnet mask 255.255.255.0, perangkat satu jaringan memiliki tiga angka depan yang sama (192.168.20.x) dan angka terakhir 1–254. Alamat statis tidak boleh sama dengan perangkat lain dan sebaiknya di luar rentang yang dibagikan DHCP.

Langkah:

1. Syarat 1 — satu jaringan: harus 192.168.20.x.
2. Syarat 2 — angka terakhir 1–254 dan belum dipakai: .1, .10, .20, .25, .101 sudah dipakai.
3. Syarat 3 — di luar rentang DHCP .100–.200 agar tidak bentrok di kemudian hari.
4. Yang memenuhi ketiganya: 192.168.20.30.

Analisis setiap opsi:

- **A.** ✗ Alamat .20 sudah dipakai printer sehingga terjadi bentrok alamat.
- **B.** ✗ Jebakan: angka terakhirnya belum dipakai, tetapi 192.168.2.x berbeda jaringan dengan 192.168.20.x.
- **C.** ✗ Jebakan: saat ini belum dipakai, tetapi .150 berada di rentang DHCP sehingga suatu saat dapat dibagikan ke perangkat lain.
- **D.** ✓ Satu jaringan, belum dipakai, dan di luar rentang DHCP.
- **E.** ✗ Angka 256 melebihi batas 255 sehingga bukan alamat IP yang sah.

Jawaban: D.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap soal, tulis mask-nya, bandingkan bagian jaringan per oktet, lalu tentukan pengaruh gateway hanya bila tujuan di luar jaringan.

1. Di lab sekolah (2027), server uji aplikasi dan empat laptop siswa tersambung ke satu switch yang terhubung ke router 192.168.5.1 untuk akses internet. Hasil pemeriksaan konfigurasi dan uji membuka server uji ditunjukkan pada tabel.

Perangkat	Alamat IP	Subnet mask	Gateway	Membuka http://192.168.5.10:8000
Server uji	192.168.5.10	255.255.255.0	192.168.5.1	–
Laptop 1	192.168.5.21	255.255.255.0	192.168.5.1	berhasil
Laptop 2	192.168.5.22	255.255.255.0	192.168.15.1	berhasil
Laptop 3	192.168.6.23	255.255.255.0	192.168.5.1	gagal

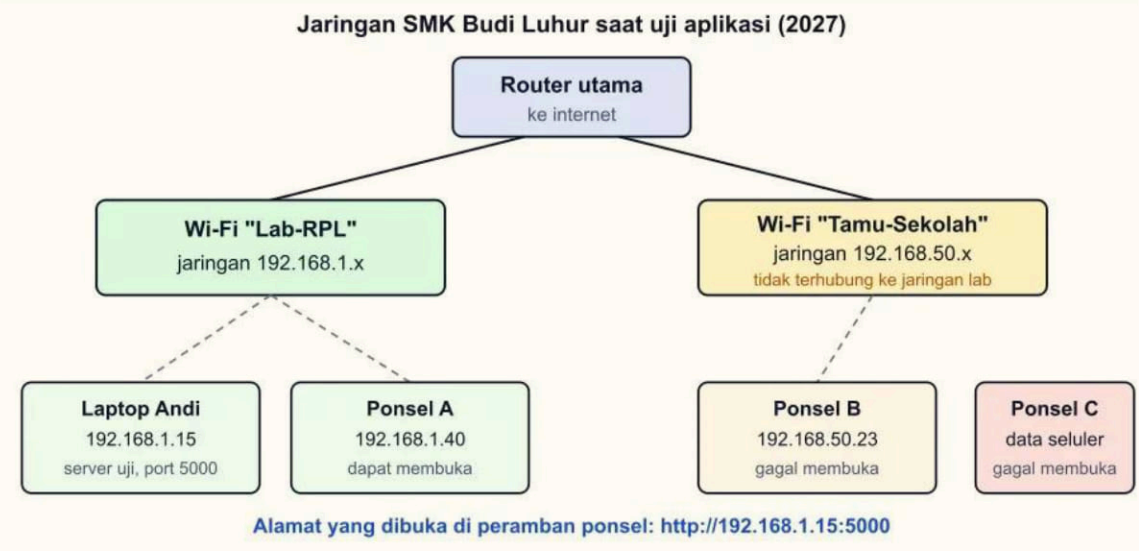
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat berdasarkan tabel adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Laptop 3 dapat diperbaiki dengan mengganti gateway-nya menjadi 192.168.6.1.
☐ B. Laptop 3 gagal karena alamat IP-nya berada di jaringan yang berbeda dengan server uji.
☐ C. Laptop 2 tetap dapat membuka server uji karena satu jaringan dengannya walau gateway-nya salah.
☐ D. Alamat server uji perlu diubah menjadi 192.168.6.10 agar semua laptop dapat membukanya.
☐ E. Laptop 2 akan gagal membuka situs di internet selama gateway-nya belum diperbaiki.

2. Andi menjalankan server uji aplikasi web di laptopnya. Tiga ponsel dipakai untuk menguji tampilan aplikasi, dan susunan jaringannya ditunjukkan pada gambar. Ponsel A dapat membuka aplikasi, sedangkan ponsel B dan C gagal. Server berjalan normal dan firewall laptop sudah mengizinkan port 5000.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan yang paling tepat agar ponsel B dan C dapat ikut menguji adalah ....



A. Mengganti port aplikasi dari 5000 menjadi 80 agar terbuka bagi semua ponsel
B. Menyambungkan ponsel B dan C ke Wi-Fi Lab-RPL, lalu membuka alamat yang sama
C. Mengubah alamat laptop Andi menjadi 192.168.50.15 tanpa memindah jaringannya
D. Meminta ponsel B dan C membuka http://localhost:5000 di peramban ponsel
E. Mematikan data seluler ponsel C agar ponsel itu otomatis masuk ke jaringan lab

Kunci singkat: 1. B, C, E 2. B

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa dua laptop yang satu jaringan masih dapat saling berhubungan walaupun gateway salah satunya keliru?
- Mengapa sebuah alamat statis untuk server tidak boleh berada di dalam rentang DHCP?
- **Yang harus kuingat:** satu jaringan bila bagian jaringan sama menurut mask; oktet 0–255; host `.1`–`.254` pada /24; gateway hanya untuk keluar jaringan; alamat statis di luar DHCP; periksa konektivitas dari bawah ke atas.



Latihan interaktif di LMS

Sub bab 3.2: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-3.2

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 3.3 · Layer TCP/IP dan fungsi protokol

SMKPPLG-3.3

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **mengimplementasikan dan menganalisis konsep layer TCP/IP** serta **fungsi protokol pada setiap layer**. Pada soal, kamu biasanya diberi satu kasus komunikasi, misalnya pesan yang dikirim lewat aplikasi obrolan, login ke gim daring, atau pengelolaan situs sekolah di server sewaan. Lalu kamu diminta menjelaskan **di layer mana sesuatu terjadi**, **protokol apa yang tepat**, atau **mana pernyataan yang benar tentang protokol itu**.

Soal bagian ini lebih sering berbentuk **pernyataan benar atau salah** (Benar/Salah dan pilih lebih dari satu), karena kesalahan yang paling sering terjadi adalah menukar tugas layer atau menukar sifat TCP dan UDP.

Setelah mempelajari sub bab ini, kamu dapat:

- menyebutkan empat layer TCP/IP dan tugas masing-masing, serta padanannya pada model OSI;
- menempatkan protokol (HTTP, HTTPS, DNS, FTP, SSH, TCP, UDP, IP, Ethernet, Wi-Fi) pada layer yang tepat;
- menjelaskan enkapsulasi dan di layer mana port, alamat IP, dan alamat MAC ditentukan;
- membedakan TCP dan UDP serta memilih yang sesuai untuk sebuah fitur;
- menjelaskan fungsi HTTPS, SSH, SFTP, dan DNS, termasuk hal yang tidak dapat dilakukannya.

NEXT LEVEL STUDY

◆ KONSEP INTI

Ketika kamu mengirim pesan "Tugas sudah dikumpulkan" lewat aplikasi, banyak hal terjadi dalam sekejap: pesan disusun, ditujukan ke program yang benar di server, dikirim lewat jaringan yang berpindah-pindah, lalu dikirim sebagai sinyal lewat kabel atau udara. Pekerjaan sebesar itu tidak dikerjakan satu program. Ia dibagi menjadi **lapisan (layer)**, dan tiap lapisan hanya mengurus tugasnya sendiri. Pembagian ini membuat pengembang aplikasi tidak perlu memikirkan sinyal listrik, dan pembuat kartu jaringan tidak perlu memikirkan isi pesan.

1. Model TCP/IP empat layer

Keluarga protokol **TCP/IP** yang dipakai internet dibagi menjadi **empat layer**, dari atas ke bawah (Gambar 3.13).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Empat layer TCP/IP, protokolnya, dan satuan datanya

Layer	Protokol contoh	Satuan data	Tugas utama	Padanan
Aplikasi	HTTP, HTTPS, DNS, FTP/SFTP, SMTP/IMAP, SSH, DHCP	data / pesan	berhubungan langsung dengan program; menyusun dan membaca isi pesan	OSI 5–7: aplikasi, penyajian, sesi
Transport	TCP, UDP (nomor port)	segmen (TCP) / datagram (UDP)	mengantar antarprogram; menentukan port; TCP menjamin keandalan	OSI 4: transport
Internet	IP, ICMP (ping) (alamat IP)	paket	menentukan alamat IP tujuan dan merutekan paket antarjaringan	OSI 3: jaringan
Akses jaringan	Ethernet, Wi-Fi (alamat MAC)	frame / bit	mengirim bit lewat kabel atau udara ke perangkat di jaringan yang sama	OSI 1–2: data link, fisik

dikirim: dari atas ke bawah · diterima: dari bawah ke atas

Gambar 3.13 Empat layer TCP/IP, contoh protokolnya, satuan datanya, dan padanan pada model OSI. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Layer aplikasi**: berhubungan langsung dengan program. Di sinilah pesan disusun menurut aturan sebuah protokol aplikasi, misalnya permintaan **HTTP(S)** untuk membuka situs. **DNS**, **FTP/SFTP**, **SMTP/IMAP** (surel), **SSH**, dan **DHCP** juga protokol layer aplikasi.
2. **Layer transport**: mengantarkan data **antara program di dua perangkat**. Di sinilah **nomor port** ditentukan dan di sinilah **TCP** atau **UDP** bekerja.
3. **Layer internet**: menentukan **alamat IP** pengirim dan tujuan serta **merutekan paket antarjaringan**. Protokolnya **IP**; **ICMP** (dipakai oleh `ping`) juga di sini.
4. **Layer akses jaringan**: mengirim data sebagai bit lewat kabel atau udara ke perangkat **di jaringan yang sama**, memakai **Ethernet** atau **Wi-Fi** dan **alamat MAC** (alamat tetap kartu jaringan).

Model lain yang sering disebut adalah **OSI** dengan **tujuh layer**. Padanannya: tiga layer paling atas OSI (aplikasi, penyajian, sesi) setara dengan layer aplikasi TCP/IP; transport setara dengan transport; jaringan setara dengan layer internet; dan *data link* serta *fisik* setara dengan layer akses jaringan. Soal TKA memakai model TCP/IP, jadi hafalkan keempat layer di atas beserta tugasnya.

Mengapa model berlapis berguna? Pertama, **tiap lapisan dapat diganti tanpa mengubah yang lain**: aplikasi web tetap sama baik dikirim lewat kabel maupun lewat Wi-Fi. Kedua, **kesalahan lebih mudah dilacak**: bila halaman tidak terbuka, kamu memeriksa dari layer bawah ke atas. Ketiga, **protokol dapat dikembangkan secara terpisah** oleh pihak berbeda, asalkan mengikuti kesepakatan antarlayer.

Cara mengingat tugas tiap layer dengan satu pertanyaan:

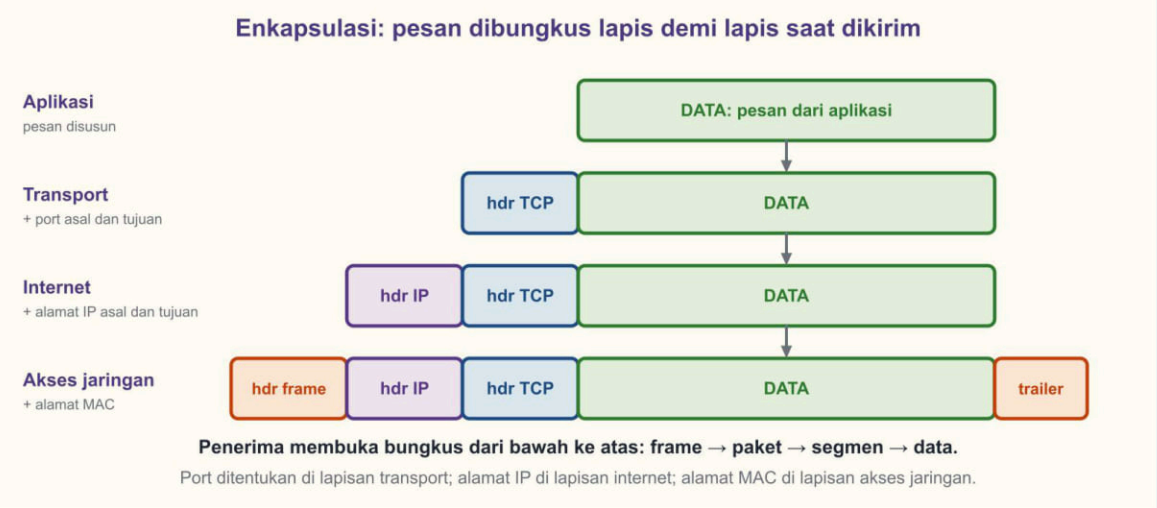
- Aplikasi: **apa isi pesannya dan menurut aturan apa?**
- Transport: **untuk program yang mana (port), dan dikirim dengan cara apa (TCP atau UDP)?**
- Internet: **ke perangkat mana di dunia (alamat IP), lewat jalur mana?**

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- Akses jaringan: **bagaimana bit sampai ke perangkat sebelah (MAC, kabel, Wi-Fi)?**

2. Enkapsulasi: pesan dibungkus lapis demi lapis

Saat dikirim, pesan melewati layer dari atas ke bawah, dan **tiap layer menambahkan bungkus (header)** berisi informasi yang dibutuhkannya. Proses ini disebut **enkapsulasi** (Gambar 3.14).



Gambar 3.14 Enkapsulasi pesan dari layer aplikasi sampai layer akses jaringan. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Aplikasi** menyusun data (misalnya permintaan HTTPS).
2. **Transport** menambah header berisi **port asal dan port tujuan** (misalnya tujuan 443). Hasilnya disebut **segmen** (pada TCP).
3. **Internet** menambah header berisi **alamat IP asal dan IP tujuan**. Hasilnya disebut **paket**.
4. **Akses jaringan** menambah header berisi **alamat MAC** dan penutup (trailer). Hasilnya disebut **frame**, yang dikirim sebagai bit.

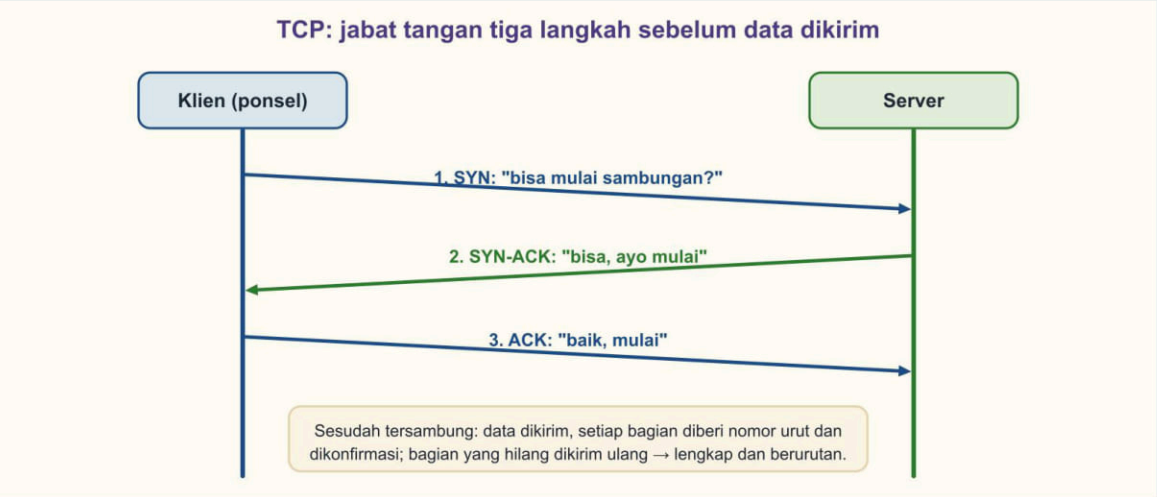
Di perangkat penerima prosesnya terbalik: frame dibuka menjadi paket, lalu segmen, lalu data, dan akhirnya sampai ke program yang dituju. Dari proses ini kamu mendapat aturan yang penting untuk soal Benar/Salah: **nomor port ditentukan di layer transport, bukan di layer internet; alamat IP ditentukan di layer internet; alamat MAC di layer akses jaringan.**

3. TCP dan UDP

Layer transport menyediakan dua protokol dengan sifat berbeda.

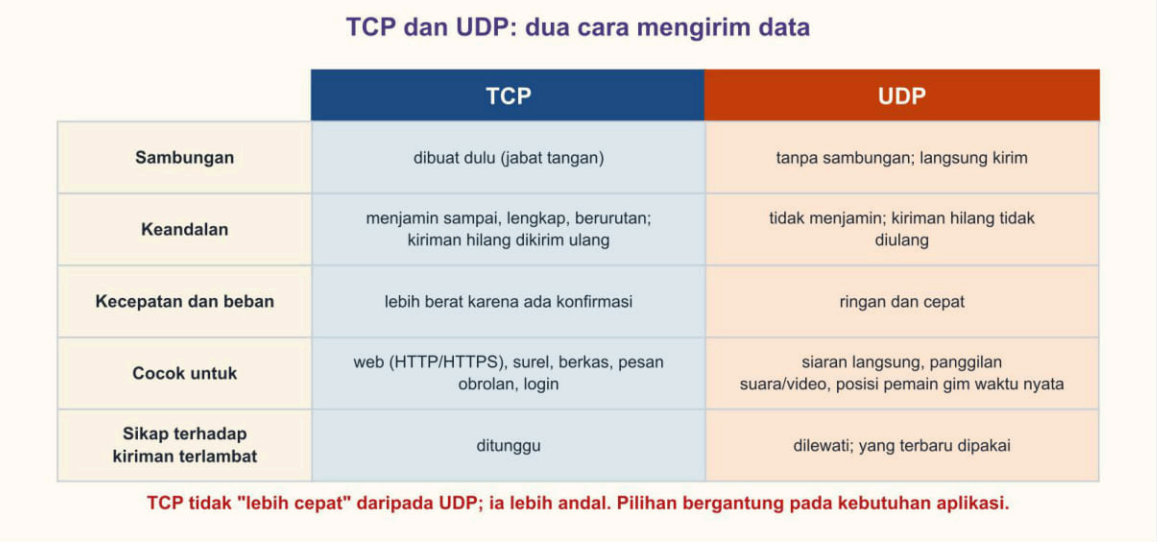
TCP (*Transmission Control Protocol*) adalah protokol **berorientasi sambungan**. Sebelum data dikirim, kedua pihak melakukan **jabat tangan tiga langkah** (Gambar 3.15): klien mengirim **SYN** ("bisa mulai sambungan?"), server menjawab **SYN-ACK** ("bisa, ayo mulai"), klien membalas **ACK** ("baik"). Sesudah tersambung, data dipecah menjadi bagian bernomor urut dan tiap bagian **dikonfirmasi**. Bagian yang hilang **dikirim ulang**. Hasilnya, data sampai **lengkap dan berurutan**.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 3.15 Jabat tangan tiga langkah TCP. Sumber: digambar oleh Tim Next Level Study Indonesia.

UDP (*User Datagram Protocol*) **tidak membuat sambungan** dan tidak menjamin apa pun: data langsung dikirim, dan bagian yang hilang tidak diulang. Karena tidak ada konfirmasi dan pengulangan, UDP **ringan dan cepat**. Perbandingan keduanya dirangkum pada Gambar 3.16.



Gambar 3.16 Perbandingan TCP dan UDP. Sumber: digambar oleh Tim Next Level Study Indonesia.

Dua catatan tambahan: pertanyaan DNS yang singkat biasanya memakai UDP karena cepat dan cukup diulang sendiri oleh aplikasi bila jawabannya tidak datang; dan sebuah aplikasi dapat memakai **kedua** protokol sekaligus untuk fitur yang berbeda, seperti gim yang memakai TCP untuk login dan UDP untuk gerak karakter.

Pilihan di antara keduanya bergantung pada kebutuhan aplikasi:

- **TCP** untuk yang tidak boleh kurang atau salah urutan: membuka situs, mengunduh berkas, surel, **pesan obrolan**, dan **login**.
- **UDP** untuk yang **lebih baik terlambat dilewati daripada ditunggu**: siaran langsung, panggilan suara dan video, dan **posisi karakter dalam gim waktu nyata**. Bila satu kiriman posisi terlambat, posisi terbaru yang berikutnya lebih berguna daripada menunggu yang lama.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Satu pelurusan yang sering diuji: **TCP tidak "lebih cepat" daripada UDP.** TCP lebih **andal**, UDP lebih **ringan**.

Alamat MAC dan alamat IP: apa bedanya? Keduanya sama-sama alamat, tetapi bekerja pada layer berbeda. **Alamat MAC** melekat pada kartu jaringan dan dipakai **di dalam satu jaringan** (layer akses jaringan). **Alamat IP** dapat berganti menurut jaringan tempat perangkat berada dan dipakai untuk **menjangkau jaringan lain** (layer internet). Ibaratnya, alamat IP seperti alamat rumah lengkap dengan kota, sedangkan alamat MAC seperti nama penghuni yang dikenali tetangga satu gang. Protokol **ARP** menghubungkan keduanya: saat laptop tahu alamat IP tujuan di jaringan yang sama, ARP menanyakan alamat MAC pemiliknya.

Port umum. Nomor port membantu server membedakan layanan yang berjalan di satu alamat IP. Beberapa nomor yang lazim: **80** (HTTP), **443** (HTTPS), **22** (SSH dan SFTP), **21** (FTP), **25** (SMTP), dan **53** (DNS). Pembahasan port lebih lengkap ada di sub bab 3.4.

Ping dan ICMP. Perintah `ping` mengirim pesan kecil dengan protokol **ICMP** ke sebuah alamat IP dan menunggu balasan. Pesan ini berada di layer internet. Bila ping berhasil, berarti perangkat tujuan **dapat dijangkau di layer internet**; itu belum membuktikan layanan di atasnya (misalnya situs web) berjalan.

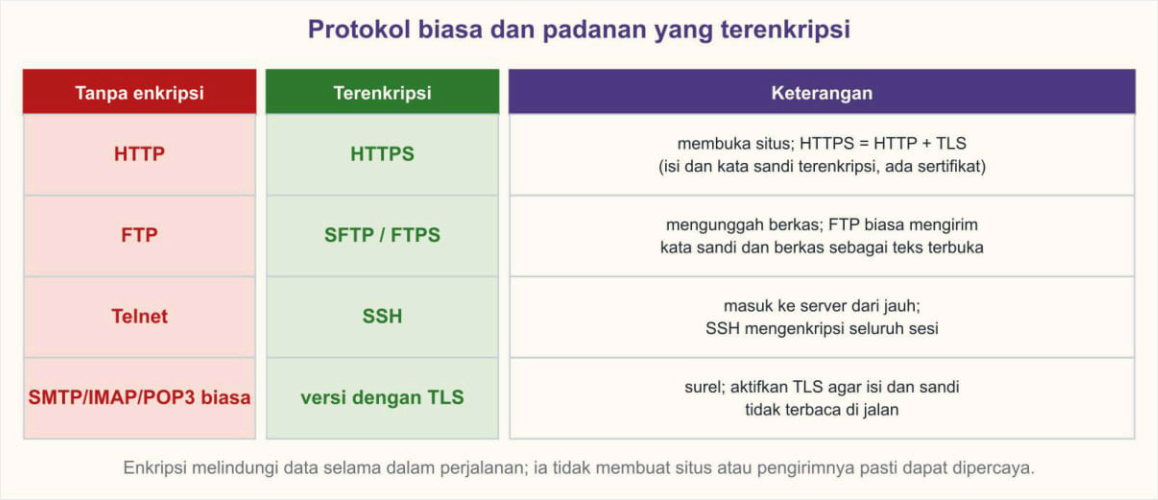
4. Protokol layer aplikasi yang perlu dikenal

- **HTTP** dan **HTTPS**: protokol web. Peramban mengirim permintaan dan server membalas dengan halaman atau data.
- **DNS**: menerjemahkan **nama** (misalnya `smkbudi.sch.id`) menjadi **alamat IP**. DNS adalah protokol **layer aplikasi**, bukan layer internet. Bila alamat IP server berubah (misalnya server dipindahkan), **catatan DNS perlu diperbarui** agar nama itu menunjuk ke alamat yang baru.
- **FTP** dan **SFTP**: memindahkan berkas ke dan dari server.
- **SMTP**, **IMAP**, **POP3**: mengirim dan menerima surel.
- **SSH**: masuk ke server dari jauh dan menjalankan perintah, dengan **sesi terenkripsi**.
- **DHCP**: memberi alamat IP otomatis kepada perangkat (sub bab 3.2). Walau terkait jaringan, DHCP juga protokol layer aplikasi.

5. Protokol terenkripsi dan batasnya

Beberapa protokol lama mengirim data sebagai **teks terbuka**: siapa pun yang dapat menyadap jaringan dapat membacanya, termasuk kata sandi. Karena itu dipakai padanan yang terenkripsi (Gambar 3.17).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 3.17 Protokol tanpa enkripsi dan padanannya yang terenkripsi. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **HTTPS = HTTP + TLS**. Isi permintaan, termasuk nama pengguna dan kata sandi, **terenkripsi** selama dalam perjalanan, dan server membuktikan identitasnya dengan **sertifikat**. Pengguna lain di Wi-Fi yang sama tidak dapat membaca kata sandi yang dikirim lewat HTTPS.
- **FTP biasa** mengirim kata sandi dan berkas **tanpa enkripsi**; gunakan **SFTP** (berjalan di atas SSH) atau FTPS.
- **SSH** menggantikan **Telnet** yang tanpa enkripsi.

Cara kerja HTTPS secara ringkas: (1) peramban meminta server membuktikan identitasnya; (2) server mengirim **sertifikat** yang diterbitkan lembaga tepercaya; (3) peramban memeriksa bahwa sertifikat itu sah, belum kedaluwarsa, dan cocok dengan nama situs; (4) keduanya menyepakati **kunci rahasia** untuk sesi itu; (5) seluruh data sesudahnya dienkripsi dengan kunci tersebut. Itulah sebabnya sertifikat yang kedaluwarsa (sub bab 2.2) membuat peramban menampilkan peringatan kepada pengunjung.

Enkripsi punya batas. Ia melindungi data **selama dalam perjalanan**, tetapi **tidak membuktikan bahwa situs atau pengirimnya pasti baik**, dan tidak melindungi data yang sudah tersimpan di server bila server itu dibobol.

6. Menentukan layer yang bermasalah dari gejala

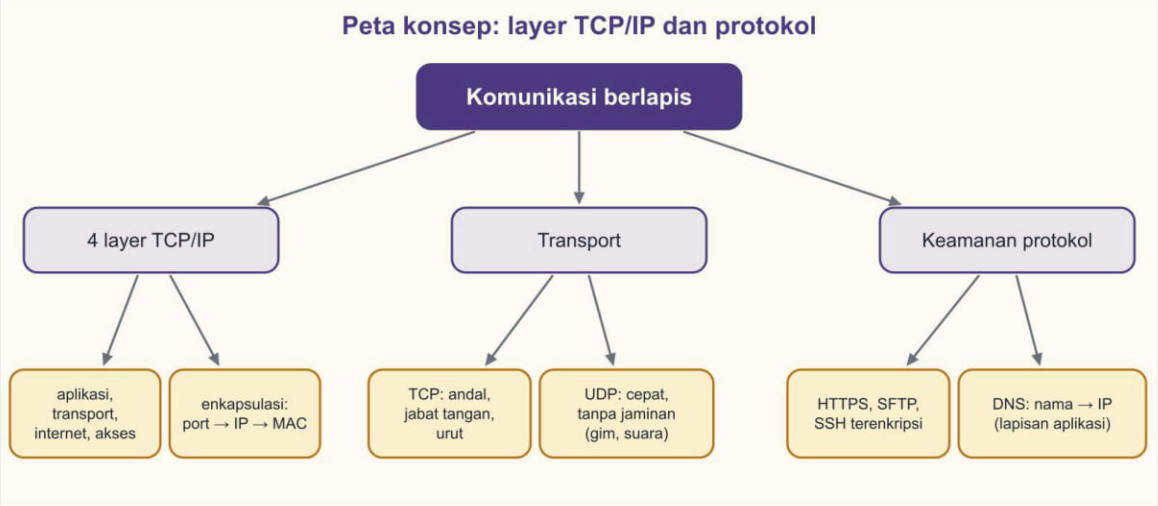
Memahami layer membantu menelusuri gangguan. Pegangan sederhananya:

Gejala	Layer yang dicurigai
Wi-Fi atau kabel terputus; lampu port mati	akses jaringan
Ping ke alamat IP tujuan gagal padahal kabel baik; alamat salah atau rute tidak ada	internet
Alamat IP dapat di-ping, tetapi sambungan ke port tertentu gagal (diblokir atau layanan tidak berjalan)	transport
Ping ke alamat IP berhasil, tetapi membuka lewat **nama** gagal	aplikasi (DNS)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Server menjawab, tetapi dengan kesalahan seperti 404 atau 500	aplikasi

Gangguan di layer bawah menghentikan semua layer di atasnya. Karena itu urutan pemeriksaan selalu **dari bawah ke atas**: pastikan sambungan fisik dan Wi-Fi, lalu alamat IP, lalu port dan layanan, lalu isi respons aplikasi. Kamu menghemat waktu bila tidak membongkar kode aplikasi sebelum memastikan kabel tertancap. Bila Wi-Fi laptop terputus (layer akses jaringan), tidak ada data dari layer di atasnya yang dapat terkirim.



Gambar 3.18 Peta konsep layer TCP/IP dan protokol. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Perjalanan sebuah foto melewati empat layer

Putri mengirim foto tugas lewat aplikasi galeri kelas ke server `galeri.sch.id` memakai HTTPS di port 443. Jelaskan apa yang dilakukan tiap layer di laptop Putri. *(Kasus fiktif.)*

1. **Aplikasi**: menyusun permintaan HTTPS yang membawa foto. Nama `galeri.sch.id` diterjemahkan lebih dulu oleh DNS menjadi alamat IP server.
2. **Transport**: memakai **TCP**, membuat sambungan dengan jabat tangan tiga langkah, lalu menambah header dengan **port tujuan 443** (dan port asal acak).
3. **Internet**: menambah header dengan **alamat IP laptop** dan **alamat IP server**, lalu menentukan paket dikirim ke router (gateway) karena server di luar jaringan.
4. **Akses jaringan**: menambah **alamat MAC** (laptop dan router) lalu mengirim **frame** lewat Wi-Fi sebagai bit.
5. Di sisi server urutannya dibalik: frame → paket → segmen → data, sampai ke program yang melayani HTTPS.

Jawaban: port 443 ditambahkan di layer transport, alamat IP di layer internet, alamat MAC di layer akses jaringan; Wi-Fi bekerja di layer akses jaringan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Contoh 2 — Memilih TCP atau UDP

Tentukan protokol layer transport yang paling sesuai untuk empat fitur sebuah aplikasi kelas daring (2027). *(Kasus fiktif.)*

- (a) Mengunduh berkas laporan nilai.
- (b) Siaran video langsung guru.
- (c) Mengirim pesan di kolom obrolan kelas.
- (d) Panggilan suara antara dua siswa.

1. Tanyakan: apakah kekurangan atau salah urutan bisa diterima?
2. (a) Berkas harus lengkap, tidak boleh rusak → **TCP**.
3. (b) Siaran langsung: potongan yang terlambat tidak berguna lagi, lebih baik dilewati → **UDP**.
4. (c) Pesan harus lengkap dan berurutan → **TCP**.
5. (d) Suara waktu nyata: lebih baik ada jeda kecil daripada menunggu potongan lama → **UDP**.

Jawaban: (a) TCP; (b) UDP; (c) TCP; (d) UDP.

Contoh 3 — Gejala dan layer yang bermasalah

Tentukan layer yang paling mungkin bermasalah pada tiap gejala. *(Kasus fiktif.)*

- (a) Laptop menampilkan "tidak ada Wi-Fi tersambung".
- (b) `ping 203.0.113.25` berhasil, tetapi `ping api.toko.id` gagal dengan pesan "tidak dapat menemukan host".
- (c) Browser membuka `https://toko.id/produk` dan menerima halaman "500 Internal Server Error".

1. (a) Perangkat belum tersambung ke media jaringan → **akses jaringan**.
2. (b) Alamat IP dapat dijangkau, sedangkan nama tidak dapat diterjemahkan → **DNS, layer aplikasi**.
3. (c) Sambungan sampai ke server dan server menjawab, tetapi dengan kesalahan aplikasi → **layer aplikasi** (sisi server).

Jawaban: (a) akses jaringan; (b) aplikasi (DNS); (c) aplikasi.

Contoh 4 — Memperbaiki protokol yang tidak aman

Tim kecil mengelola situs di server sewaan pada 2027. Mereka masuk ke server memakai Telnet dan mengunggah berkas dengan FTP biasa. Seorang anggota meminta agar seluruh sambungan diamankan. Protokol apa yang sebaiknya dipakai? *(Kasus fiktif.)*

1. Telnet dan FTP biasa mengirim sandi dan data sebagai **teks terbuka**.
2. Pengganti Telnet: **SSH**, yang mengenkripsi seluruh sesi.
3. Pengganti FTP biasa: **SFTP** (di atas SSH) atau FTPS.
4. Situs untuk pengunjung memakai **HTTPS** agar data login terenkripsi.

Jawaban: SSH untuk masuk server, SFTP untuk unggah berkas, HTTPS untuk situs.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

❖ PETA KONSEP

Gambar 3.18 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Empat layer**: aplikasi → transport → internet → akses jaringan. Saat dikirim turun dan dibungkus (enkapsulasi), saat diterima naik dan dibuka.
- **Port** di layer transport; **alamat IP** di layer internet; **alamat MAC** di layer akses jaringan.
- **TCP andal** (jabat tangan, konfirmasi, urut); **UDP ringan** (tanpa jaminan). TCP bukan "lebih cepat".
- **DNS** (nama → IP) ada di layer aplikasi dan catatannya diperbarui bila IP server berubah.
- **HTTPS, SFTP, SSH** mengenkripsi data dalam perjalanan; enkripsi tidak menjamin situs pasti terpercaya.
- Gangguan di layer bawah menghentikan semua layer di atasnya.

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal layer dan protokol:

1. **Tulis keempat layer** dan tentukan di layer mana kegiatan pada soal terjadi.
2. **Cocokkan dengan tugas layer**: aplikasi (isi pesan), transport (port), internet (IP), akses (MAC, kabel, Wi-Fi).
3. **Untuk TCP atau UDP**, tanyakan apakah data boleh kurang atau terlambat.
4. **Untuk soal gangguan**, mulai dari layer terbawah dan naik.

Jebakan yang sering dipasang:

- **Port dan alamat IP ditempatkan di layer yang sama.** Port di transport, IP di internet.
- **DNS ditempatkan di layer internet.** DNS adalah protokol layer aplikasi.
- **"TCP lebih cepat daripada UDP"** atau "UDP tidak boleh dipakai karena tidak andal".
- **HTTPS dikira membuat situs pasti aman dan terpercaya.** Ia mengenkripsi data selama dalam perjalanan.
- **FTP biasa dikira terenkripsi seperti HTTPS.**
- **Alamat MAC dan alamat IP tertukar**: MAC tetap pada kartu jaringan dan dipakai di jaringan yang sama, IP dapat berganti dan dipakai antarjaringan.
- **Gangguan Wi-Fi dianggap di layer aplikasi** padahal layer akses jaringan.
- **Nama tidak dapat dibuka dianggap kabel rusak** padahal alamat IP dapat di-ping (DNS).

> Contoh salah tafsir: seorang siswa menyimpulkan bahwa "karena situs memakai HTTPS, data pelanggan di server pasti aman". HTTPS hanya melindungi data selama dalam perjalanan antara peramban dan server. Bila server dibobol atau kata sandi basis datanya bocor, HTTPS tidak melindungi data yang tersimpan di server.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• BEDAH SOAL PREDIKSI

Soal berikut menyajikan rancangan komunikasi sebuah gim daring: login, posisi karakter, dan pesan obrolan. Perhatikan bagaimana tiap fitur dipasangkan dengan protokol yang tepat menurut sifat **keandalan dan keterlambatan** (bagian 3), dan bagaimana satu pilihan menyelipkan "TCP lebih cepat" sebagai jebakan.

►. Tim pengembang gim daring *Arena Nusantara* (2027) menjelaskan rancangan komunikasi gimnya. Pemain masuk (login) dengan nama pengguna dan kata sandi lewat HTTPS ke server.arena.id. Selama bertanding, posisi karakter dikirim 20 kali per detik; bila satu kiriman terlambat, kiriman itu dilewati dan dipakai posisi terbaru. Pesan obrolan antarpemain harus sampai lengkap dan berurutan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang rancangan tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. HTTPS dipakai untuk login agar nama pengguna dan kata sandi terenkripsi selama dikirim.
☐ B. UDP cocok untuk posisi karakter karena kiriman yang terlambat lebih baik dilewati daripada ditunggu.
☐ C. DNS bekerja di lapisan internet karena tugasnya mengantarkan paket ke alamat server tujuan.
☐ D. Pesan obrolan sebaiknya dikirim dengan TCP agar sampai lengkap dan sesuai urutan.
☐ E. Posisi karakter sebaiknya juga memakai TCP karena TCP lebih cepat daripada UDP.

Pembahasan **Kunci: A, B, D**

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Model TCP/IP terdiri atas lapisan aplikasi (HTTP/HTTPS, DNS), transport (TCP andal dan berurutan; UDP cepat tanpa kirim ulang), internet (IP: alamat dan rute paket), dan akses jaringan (kabel/Wi-Fi).

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ HTTPS = HTTP yang dienkripsi (TLS), sehingga data login tidak terbaca pihak lain di jaringan.
• **Pernyataan 2** ✓ UDP tidak mengirim ulang data yang hilang; cocok untuk data waktu nyata yang cepat basi.
• **Pernyataan 3** ✗ Jebakan: DNS menerjemahkan nama server.arena.id menjadi alamat IP dan bekerja di lapisan aplikasi; pengantaran paket adalah tugas IP.
• **Pernyataan 4** ✓ TCP mengirim ulang data yang hilang dan menjaga urutan, sesuai kebutuhan obrolan.
• **Pernyataan 5** ✗ TCP justru menunggu dan mengirim ulang data yang hilang sehingga lebih lambat; itu tidak cocok untuk posisi waktu nyata.

Jawaban: pernyataan 1, 2, 4.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap pernyataan, tulis dulu layer atau protokol yang dibicarakan, lalu periksa tugasnya.

1. Raka mengirim pesan "Tugas sudah dikumpulkan" lewat aplikasi obrolan sekolah SapaKelas dari laptopnya yang tersambung Wi-Fi (2027). Aplikasi mengirim pesan itu ke server sapakelas.sch.id melalui HTTPS di port 443. Guru TJKT meminta siswa PPLG menjelaskan proses pengiriman itu dengan model TCP/IP.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Aplikasi menyusun pesan dalam format permintaan HTTP(S); tahap ini terjadi di lapisan aplikasi.	○	○
Nomor port tujuan 443 dan alamat IP server sama-sama ditentukan di lapisan internet.	○	○
Bila Wi-Fi laptop terputus, gangguan berada di lapisan akses jaringan sehingga data dari lapisan di atasnya tidak dapat terkirim.	○	○

2. Tim PKL mengelola situs sekolah di server sewaan (2027). Pengunjung membuka situs lewat `https://smkbudi.sch.id`. Tim masuk ke server untuk menjalankan perintah lewat SSH dan selama ini mengunggah berkas memakai FTP biasa. Pekan lalu server dipindahkan sehingga alamat IP-nya berganti.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang protokol yang dipakai tim adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. DNS menghubungkan nama smkbudi.sch.id dengan alamat IP server, sehingga catatannya perlu diperbarui.
☐ B. FTP biasa aman dipakai karena kata sandinya ikut terenkripsi seperti pada HTTPS.
☐ C. SSH dipakai untuk masuk dan menjalankan perintah di server dari jauh secara terenkripsi.
☐ D. Unggahan berkas sebaiknya beralih ke SFTP agar berkas dan kata sandinya terenkripsi.
☐ E. HTTPS melindungi data yang dikirim antara peramban pengunjung dan server situs.

Kunci singkat: 1. 1B 2S 3B 2. A, C, D, E

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Di layer mana nomor port, alamat IP, dan alamat MAC masing-masing ditentukan?
- Mengapa posisi karakter dalam gim waktu nyata lebih cocok memakai UDP, sedangkan pesan obrolan memakai TCP?
- **Yang harus kuingat:** empat layer dan tugasnya; enkapsulasi menambah port → IP → MAC; TCP andal dan UDP ringan; DNS di layer aplikasi; HTTPS, SFTP, dan SSH terenkripsi tetapi tidak menjamin kepercayaan situs.

Latihan interaktif di LMS

Sub bab 3.3: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-3.3

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 3.4 · Mekanisme komunikasi jaringan

SMKPPLG-3.4

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **mengimplementasikan dan menganalisis mekanisme komunikasi berbasis TCP/IP**, dan muatannya juga menyebut HTTP/HTTPS, model client–server, serta akses basis data atau API lewat jaringan. Kalau sub bab 3.3 membahas *apa* yang dilakukan tiap layer, sub bab ini membahas *bagaimana sebuah aplikasi benar-benar berkomunikasi*: siapa yang meminta, ke mana, lewat pintu nomor berapa, dan apa arti jawaban yang kembali.

Pada soal, kamu biasanya diberi **sebuah gejala dan hasil uji**: tabel hasil `nslookup`, `ping`, dan pembukaan beberapa alamat; kode status 404 atau 500; atau susunan satu server yang menjalankan dua program. Kamu diminta menentukan **di mana letak gangguan** atau **pernyataan mana yang benar**. Soal-soal ini sulit bukan karena istilahnya, tetapi karena jawaban yang benar harus **ditarik dari bukti pada hasil uji**, bukan dari dugaan.

Setelah mempelajari sub bab ini, kamu dapat:

- menjelaskan model client–server dan siklus permintaan serta balasan HTTP;
- membaca kode status HTTP dan menentukan apakah kesalahan ada di sisi klien atau server;
- menguraikan bagian-bagian URL dan menjelaskan peran DNS serta nomor port;
- menjelaskan REST API, JSON, dan arsitektur aplikasi tiga lapis (klien, server aplikasi, basis data);
- membaca hasil `nslookup`, `ping`, dan uji alamat untuk menemukan letak gangguan, serta memilih port yang aman dibuka pada firewall.

◆ KONSEP INTI

Hampir semua aplikasi modern tidak berjalan sendirian. Aplikasi pemesanan makanan memanggil server, server mengambil data dari basis data, dan gim daring bertukar posisi dengan pemain lain. Semua itu bergantung pada pola komunikasi yang sama. Dengan memahaminya, kamu dapat menjelaskan mengapa sebuah fitur gagal dan di sisi mana perbaikannya harus dilakukan.

1. Model client–server

Dalam komunikasi jaringan, dua peran selalu hadir.

- **Client (klien)** adalah pihak yang **meminta**: peramban di laptop, aplikasi di ponsel, atau program lain.
- **Server** adalah program yang **melayani**: ia menunggu permintaan, memprosesnya, lalu

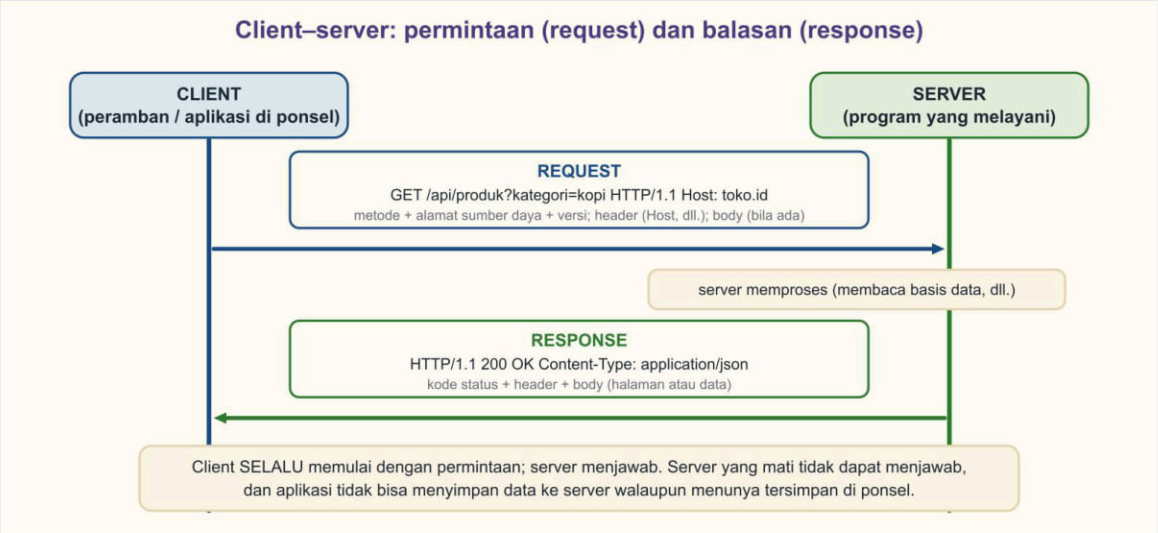
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

menjawab. "Server" menunjuk program, walaupun sering juga dipakai untuk komputer tempat program itu berjalan.

Pola dasarnya **klien memulai, server menjawab**. Klien selalu yang pertama mengirim permintaan. Konsekuensinya penting untuk soal: bila server mati, **aplikasi klien tidak dapat menyimpan atau mengambil data dari server**, walaupun tampilan menu tersimpan di ponsel. Data yang harus dibagi, seperti pesanan, hanya bisa tersimpan melalui server.

2. HTTP: permintaan dan balasan

HTTP (*HyperText Transfer Protocol*) adalah protokol layer aplikasi yang dipakai web dan sebagian besar API. Komunikasinya berupa pasangan **request** (permintaan) dan **response** (balasan) (Gambar 3.19).



Gambar 3.19 Siklus permintaan dan balasan antara klien dan server. Sumber: digambar oleh Tim Next Level Study Indonesia.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sebuah **request** memuat:

- **metode**, yang menyatakan maksud permintaan;
- **alamat sumber daya** yang diminta (path);
- **header** (keterangan tambahan seperti nama host) dan, bila perlu, **body** (isi yang dikirim, misalnya data formulir).

Metode yang paling sering dipakai:

Metode	Maksud	Contoh
GET	membaca atau mengambil data	menampilkan daftar produk
POST	membuat data baru atau mengirim data	mengirim pesanan, login
PUT	memperbarui data	mengubah alamat pelanggan
DELETE	menghapus data	menghapus item keranjang

Sebuah **response** memuat **kode status**, header, dan body (halaman atau data). Kode status adalah angka tiga digit yang memberi tahu hasil permintaan.

Contoh sebuah request dan response dalam bentuk teksnya (disederhanakan):

```
GET /api/v1/produk?kategori=kopi HTTP/1.1
Host: toko.id
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 118
```

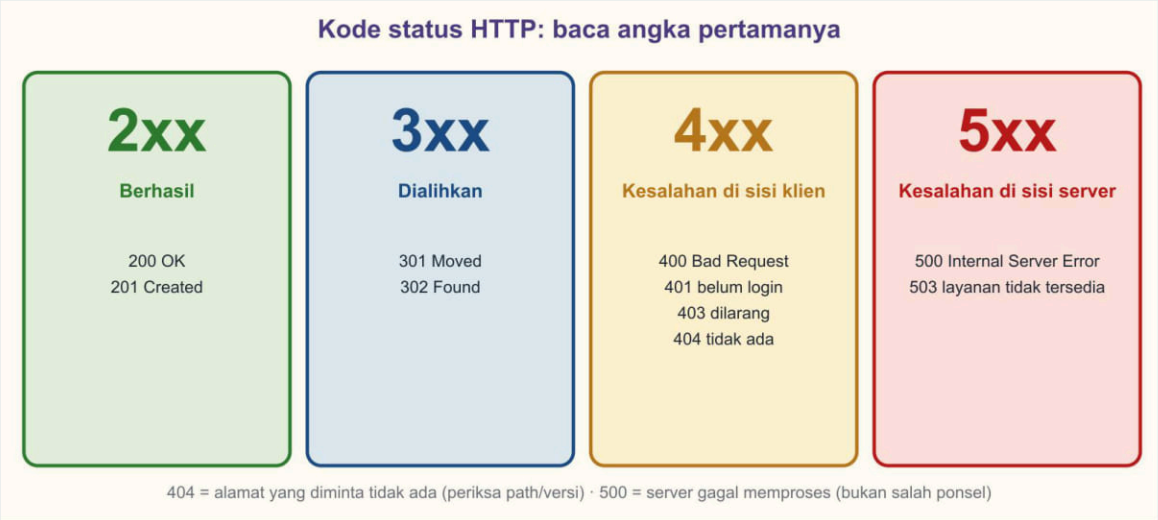
Baris pertama request memuat **metode, path, dan versi HTTP**; baris berikutnya adalah **header**. Response diawali **kode status** lalu header, dan sesudahnya baru body berisi datanya.

HTTP bersifat **tanpa ingatan (stateless)**: tiap permintaan berdiri sendiri, dan server tidak otomatis mengingat siapa yang baru saja meminta. Agar pengguna tetap dikenali sesudah login, server memberi **cookie** atau **token** yang kemudian dikirim klien pada setiap permintaan berikutnya. Itulah sebabnya data login yang terkirim lewat `http://` berbahaya: cookie dan kata sandinya dapat dibaca orang di jaringan yang sama.

3. Kode status HTTP

Angka **pertama** kode status menunjukkan keluarganya (Gambar 3.20).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 3.20 Keluarga kode status HTTP. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **2xx** berhasil: `200 OK`, `201 Created` (data baru berhasil dibuat).
- **3xx** dialihkan: `301`/`302` meminta klien memakai alamat lain.
- **4xx kesalahan di sisi klien** (permintaan yang salah): `400 Bad Request` (permintaan tidak dimengerti), `401` (belum masuk/login), `403 Forbidden` (tidak punya hak), dan **`404 Not Found`** (alamat yang diminta tidak ada di server).
- **5xx kesalahan di sisi server** (server gagal memproses permintaan yang sah): **`500 Internal Server Error`** dan `503 Service Unavailable` (layanan sedang tidak tersedia).

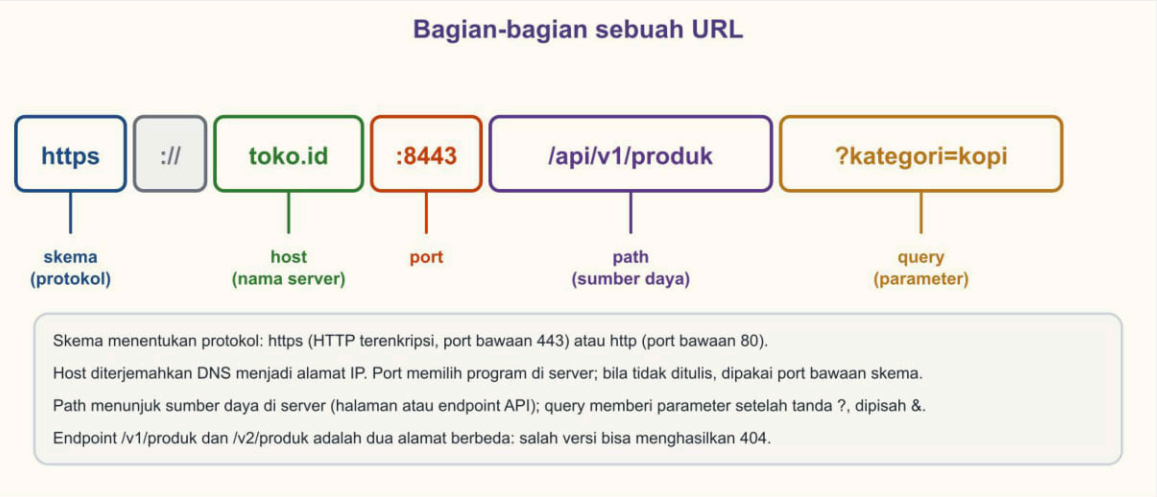
Kode `503` sering muncul ketika server kewalahan melayani banyak permintaan sekaligus, misalnya saat pendaftaran siswa baru dibuka serentak. Di sini pun kesalahannya ada di sisi server: kapasitas atau konfigurasinya perlu ditingkatkan, bukan ponsel pengguna yang diperbaiki.

Pelajaran penting: **404 bukan berarti server mati.** Server menjawab, dan jawabannya "alamat itu tidak ada". Sering sebabnya klien memanggil alamat yang salah, misalnya endpoint versi lama. Sebaliknya, **500 berarti klien sudah meminta dengan benar tetapi server gagal memprosesnya**, jadi perbaikannya di server (kode, basis data, atau beban terlalu berat), bukan di ponsel.

4. URL dan DNS

URL adalah alamat lengkap sebuah sumber daya di web. Gambar 3.21 menguraikan bagian-bagiannya.

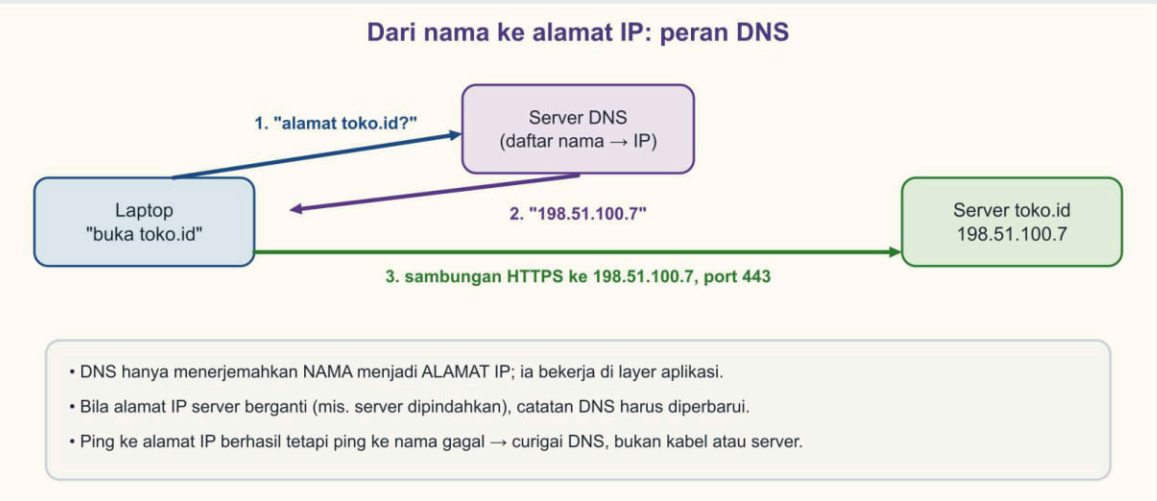
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 3.21 Bagian-bagian sebuah URL. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Skema** menentukan protokol (`https` atau `http`) dan port bawaannya.
- **Host** adalah nama server (`toko.id`) atau alamat IP.
- **Port** (setelah titik dua) memilih program di server; bila tidak ditulis, dipakai port bawaan skema (443 untuk `https`, 80 untuk `http`).
- **Path** menunjuk sumber daya, misalnya `/api/v1/produk`.
- **Query** (setelah tanda `?`) berisi parameter, dipisah dengan `&`, misalnya `?kategori=kopi&urut=harga`.

Manusia memakai nama, tetapi jaringan memakai alamat IP. Penerjemahnya adalah **DNS**. Sebelum membuka `toko.id`, laptop bertanya kepada server DNS "alamat IP-nya berapa?", mendapat jawaban, lalu menyambung ke alamat itu (Gambar 3.22). Bila alamat server berganti, catatan DNS harus diperbarui. Bila nama tidak dapat dibuka tetapi alamat IP-nya dapat di-ping, masalahnya ada di DNS.



Gambar 3.22 Peran DNS menerjemahkan nama menjadi alamat IP. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. Port: membedakan layanan di satu alamat IP

Satu server dengan satu alamat IP sering menjalankan **beberapa program**. Cara jaringan membedakannya adalah **nomor port** (0–65535). Alamat IP menunjuk perangkat; **port menunjuk program di perangkat itu**. Pasangan "alamat IP + port" mengarahkan data tepat ke program yang dituju.

Port yang lazim:

Port	Layanan
80	HTTP
443	HTTPS
22	SSH dan SFTP
21	FTP
25	SMTP (surel keluar)
53	DNS
3306	basis data MySQL
5432	basis data PostgreSQL
3000, 5000, 8000, 8080	sering dipakai server pengembangan

Karena itu, server yang menjalankan aplikasi web di port 80 dan MySQL di port 3306 dengan alamat IP yang sama tidak membingungkan permintaan: permintaan ke `192.168.1.5:80` sampai ke aplikasi web, dan permintaan ke `192.168.1.5:3306` sampai ke MySQL. Membuka `http://192.168.1.5` **tidak** menyambungkan laptop ke MySQL; jalur yang dipakai ditentukan port, bukan urutan program dinyalakan. Dan HTTP, seperti protokol aplikasi lainnya, berjalan **di atas TCP** (layer transport).

6. API, REST, dan JSON

API (*Application Programming Interface*) adalah cara program meminta layanan dari program lain. Pada web, banyak API berbentuk **REST API**: server menyediakan **endpoint**, yaitu alamat tertentu, yang dipanggil memakai metode HTTP. Contoh endpoint untuk produk: `GET https://toko.id/api/v1/produk` mengambil daftar produk. Balasannya biasanya berformat **JSON**, teks terstruktur yang mudah dibaca program:

```
{
  "id": 7,
  "nama": "Kopi Gayo 200 g",
  "harga": 45000,
  "stok": 12
}
```

Mengapa memakai API dan bukan membiarkan aplikasi membuka basis data sendiri? Karena API menjadi **pintu tunggal** yang bisa diperiksa: server memeriksa siapa yang meminta, apakah ia berhak, dan apakah datanya masuk akal sebelum menyentuh basis data. Klien yang berbeda

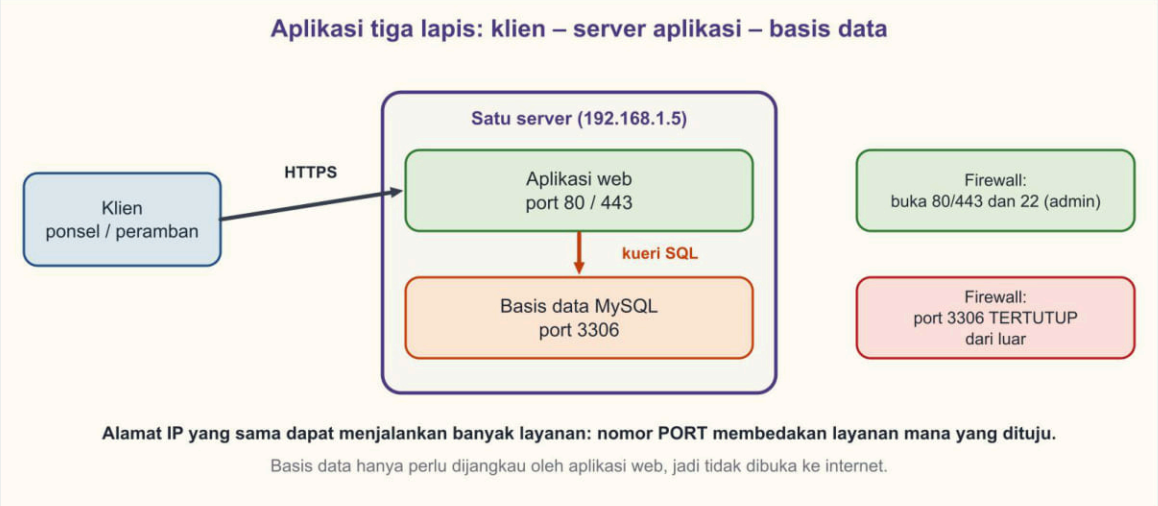
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(aplikasi ponsel dan situs web) dapat memakai endpoint yang sama.

Angka versi pada alamat (`/v1/`, `/v2/`) memungkinkan server menyediakan beberapa versi API sekaligus. Bila aplikasi masih memanggil versi lama yang sudah dihentikan, server menjawab **404** walaupun servernya sehat. Perbaikannya menyesuaikan endpoint yang dipanggil aplikasi dengan versi yang tersedia.

7. Aplikasi tiga lapis: klien, server aplikasi, basis data

Aplikasi yang menyimpan data biasanya terdiri atas tiga bagian (Gambar 3.23).



Gambar 3.23 Klien, server aplikasi, dan basis data beserta port dan firewall. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Klien**: tampilan yang dipakai pengguna (peramban atau ponsel).
2. **Server aplikasi**: menerima permintaan dari klien, menerapkan aturan aplikasi, dan berbicara dengan basis data.
3. **Basis data**: menyimpan data (misalnya MySQL di port 3306).

Klien **tidak** berbicara langsung dengan basis data. Klien memanggil server aplikasi, dan server aplikasilah yang mengirim kueri ke basis data lalu mengembalikan hasilnya. Susunan ini lebih aman: basis data tidak perlu dijangkau dari internet.

8. Firewall dan port yang dibuka

Firewall adalah penyaring lalu lintas yang mengizinkan atau menolak sambungan menurut aturan, biasanya berdasarkan alamat dan **port**. Prinsipnya: **buka hanya port yang memang diperlukan, dan tutup sisanya.** Untuk server aplikasi web yang diurus lewat SSH:

- **buka** port 443 (dan 80 bila perlu dialihkan ke 443) agar pengguna dapat memakai aplikasi;
- **buka** port 22 hanya untuk administrator (idealnya dibatasi alamat tertentu);
- **tutup** port basis data (3306, 5432) dari internet, karena hanya server aplikasi yang membutuhkannya.

Port basis data yang dibuka ke publik adalah celah keamanan yang sering dieksploitasi: penyerang dapat mencoba menebak kata sandi berulang kali dari mana saja. Dengan menutupnya dari luar, satu-satunya jalan ke basis data adalah lewat server aplikasi yang sudah

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

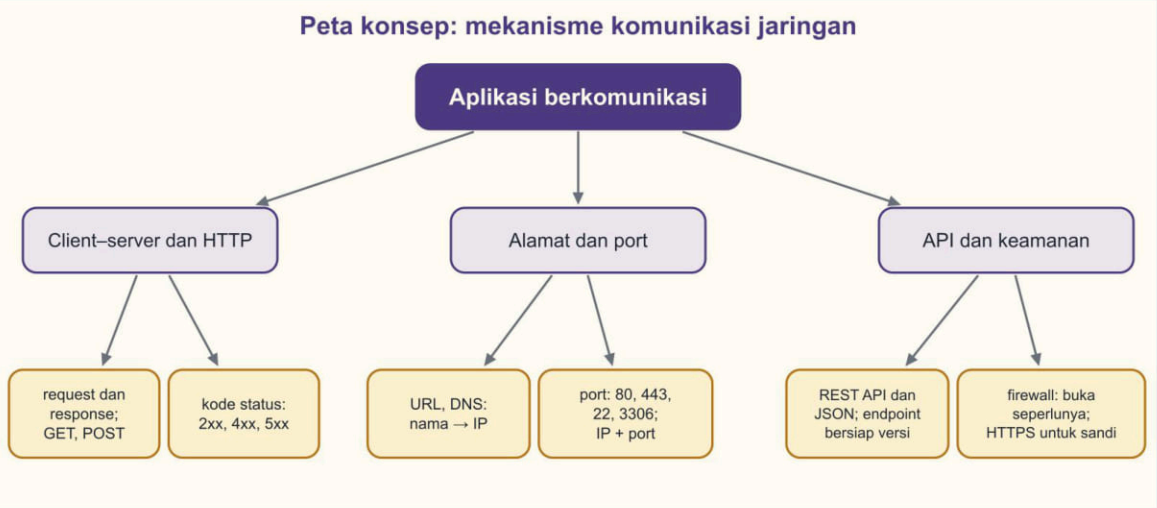
memeriksa permintaan. Hal lain: kata sandi tidak boleh dikirim lewat `http://`; pakai `https://` agar data terenkripsi dalam perjalanan (sub bab 3.3).

9. Membaca hasil uji untuk menemukan gangguan

Alat sederhana untuk menelusuri komunikasi:

- **`nslookup nama`**: menanyakan DNS alamat IP sebuah nama. Gagal → DNS bermasalah.
- **`ping alamat`**: menguji apakah perangkat tujuan menjawab (layer internet).
- **Membuka alamat di peramban atau `curl`**: menguji layanan HTTP; kode status memberi petunjuk.

Cara membacanya adalah **menyaring dari bawah ke atas**. Bila nama tidak dapat diterjemahkan tetapi ping ke alamat IP berhasil, **DNS** yang bermasalah. Bila ping berhasil dan alamat dapat dibuka tetapi halaman tertentu **404**, **alamat yang dipanggil salah** (path atau versi), bukan jaringannya. Bila sambungan ke sebuah port ditolak, layanan di port itu mungkin tidak berjalan atau firewall menutupnya. Satu cara menyimpulkan yang sehat adalah bertanya pada setiap hasil uji: "apa yang dibuktikan hasil ini, dan apa yang belum?". Ping yang berhasil membuktikan perangkat terjangkau, tetapi belum membuktikan layanan web berjalan. Dua kata yang sering muncul: **timeout** (tidak ada jawaban sama sekali, mungkin diblokir atau perangkat tidak terjangkau) dan **connection refused** (perangkat terjangkau tetapi tidak ada program yang mendengarkan di port itu).



Gambar 3.24 Peta konsep mekanisme komunikasi jaringan. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menguraikan sebuah URL

Uraikan `https://toko.id:8443/api/v1/produk?kategori=kopi&urut=harga`. Port berapa yang dipakai bila `:8443` tidak ditulis? Apa yang dikirim ke server pada bagian `?`? *(Contoh fiktif.)*

1. **Skema**: `https` → protokol HTTP terenkripsi.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. **Host**: `toko.id` → nama server, diterjemahkan DNS menjadi alamat IP.
3. **Port**: `8443` → program di port 8443, bukan port bawaan.
4. **Path**: `/api/v1/produk` → endpoint daftar produk, API versi 1.
5. **Query**: `kategori=kopi` dan `urut=harga` → dua parameter, dipisah &.
6. Bila `:8443` tidak ditulis, dipakai **port bawaan `https`, yaitu 443**.

Jawaban: bagian `?` mengirim dua parameter (kategori = kopi, urut = harga); tanpa `:8443` dipakai port 443.

Contoh 2 — Menentukan arti kode status

Aplikasi kantin sekolah (2027) menghasilkan kode berikut. Tentukan sisi yang bermasalah dan tindakan yang tepat. *(Kasus fiktif.)*

Kejadian	Kode
(a) Aplikasi memanggil `/api/v1/menu`, tetapi server hanya menyediakan `/api/v2/menu`	404
(b) Aplikasi mengirim pesanan yang benar, tetapi server gagal menyimpan ke basis data	500
(c) Siswa membuka halaman guru tanpa masuk	401
(d) Pesanan berhasil dibuat	201

1. Lihat angka pertama: 4xx = salah di sisi klien, 5xx = salah di sisi server, 2xx = berhasil.
2. (a) **404**: server sehat, tetapi alamatnya tidak ada. Perbaikan di **aplikasi**: panggil `/api/v2/menu`.
3. (b) **500**: permintaan benar, server gagal memproses. Perbaikan di **server**: periksa basis data dan kodenya.
4. (c) **401**: butuh login. Siswa perlu masuk dahulu.
5. (d) **201**: berhasil dibuat.

Jawaban: (a) sisi klien, perbaiki endpoint; (b) sisi server; (c) klien belum login; (d) berhasil.

Contoh 3 — Menentukan port yang dibuka

Sebuah server sekolah (2027) menjalankan tiga layanan: aplikasi web (HTTPS di port 443), SSH untuk administrator (port 22), dan MySQL (port 3306). Aplikasi web berada di server yang sama dengan MySQL. Tentukan aturan firewall dari internet. *(Kasus fiktif.)*

1. Tanyakan tiap layanan: siapa yang perlu menjangkaunya dari luar?
2. **443**: pengguna aplikasi di internet → **buka**.
3. **22**: hanya administrator → **buka terbatas** (idealnya hanya dari alamat administrator).
4. **3306**: hanya aplikasi web yang membutuhkan, dan ia berada di server yang sama → **tutup dari internet**.

Jawaban: buka 443, buka 22 secara terbatas, tutup 3306 dari luar.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Contoh 4 — Gangguan DNS atau jaringan?

Pada 2027, aplikasi pemesanan sebuah kafe tiba-tiba tidak dapat membuka daftar menu. Teknisi melakukan empat uji berikut. *(Data fiktif.)*

No	Uji	Hasil
1	`nslookup api.kafe.id`	tidak dapat menemukan alamat
2	`ping 198.51.100.7` (alamat server yang dikenal)	ada balasan
3	buka `https://api.kafe.id/menu`	gagal: nama tidak dapat dijangkau
4	buka `https://198.51.100.7/menu`	menu tampil

1. Uji 2 menunjukkan server **dapat dijangkau** lewat alamat IP.
2. Uji 4 menunjukkan **layanan web berjalan** dan menjawab.
3. Uji 1 dan 3 gagal **hanya ketika memakai nama**.
4. Kesimpulan: gangguan ada pada **DNS**, yaitu catatan nama `api.kafe.id` tidak terjawab atau tidak menunjuk ke alamat yang benar.
5. Perbaikan: periksa dan perbaiki catatan DNS domain itu. Memasang ulang aplikasi atau mengganti kabel tidak menolong.

Jawaban: gangguan DNS; periksa catatan DNS `api.kafe.id`.

❖ PETA KONSEP

Gambar 3.24 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Klien meminta, server menjawab**; bila server mati, klien tidak dapat menyimpan data ke server.
- **HTTP**: request (metode + alamat) dan response (kode status). **GET** membaca, **POST** membuat atau mengirim.
- **Kode status**: 2xx berhasil, 4xx salah di klien (**404** = alamat tidak ada), 5xx salah di server (**500** = server gagal memproses).
- **DNS** menerjemahkan nama menjadi IP; **port** menunjuk program di server; IP + port = tujuan lengkap.
- **REST API** memakai endpoint dan JSON; **endpoint versi lama dapat 404**.
- **Tiga lapis**: klien → server aplikasi → basis data; **port basis data tidak dibuka ke internet**; kata sandi lewat **HTTPS**.
- Menyaring gangguan **dari bawah ke atas** dari hasil nslookup, ping, dan pembukaan alamat.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal hasil uji:

1. **Tulis hasil tiap uji** dan apa yang dibuktikannya (nama terjawab? alamat IP terjangkau? layanan menjawab? apa kode statusnya?).
2. **Tentukan uji pertama yang gagal** dari bawah ke atas dan hubungkan dengan layer atau layanan itu.
3. **Pilih pernyataan yang didukung bukti pada tabel**, bukan yang sekadar masuk akal.

Langkah menjawab soal firewall dan arsitektur:

1. **Tanyakan: siapa yang perlu menjangkau layanan ini?** Pengguna umum, administrator, atau hanya aplikasi lain.
2. **Buka seperlunya**; tutup yang hanya dipakai di dalam server.

Jebakan yang sering dipasang:

- **404 dikira server mati atau DNS rusak.** Server menjawab; alamat yang dipanggil tidak ada.
- **500 dikira kesalahan ponsel** atau permintaan klien yang salah.
- **Pengguna di Wi-Fi sekolah dapat membaca kata sandi** pada HTTPS. HTTPS mengenkripsinya.
- **Server dianggap tetap berfungsi saat mati** karena menu tersimpan di ponsel.
- **IP yang sama dianggap otomatis tersambung ke semua layanan**, padahal port memilih layanan.
- **Port basis data dibuka ke internet** "supaya mudah".
- **HTTP dikira tidak memakai TCP** atau "tanpa layer transport".
- **Mengganti port, mematikan firewall, atau memasang ulang aplikasi** sebagai jawaban pertama untuk gejala yang jelas menunjuk DNS.

> Contoh salah tafsir: seorang siswa melihat bahwa `ping alamat` berhasil dan `https://alamat/halaman` menghasilkan 404, lalu menyimpulkan "jaringan bermasalah". Ping berhasil dan server menjawab dengan kode 404 membuktikan jaringan dan server sehat. Yang salah adalah alamat halaman yang diminta.

● BEDAH SOAL PREDIKSI

Soal berikut menyajikan tabel hasil uji `nslookup`, `ping`, dan pembukaan dua alamat. Perhatikan bagaimana tiap baris tabel **membuktikan sesuatu** (bagian 9), dan bagaimana pernyataan pertama menggoda untuk memilih "DNS" padahal bukti pada tabel tidak mendukungnya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Aplikasi kasir Warungku tiba-tiba gagal menampilkan daftar produk sejak pagi hari (2027). Programmer melakukan beberapa uji dari laptop kantor dengan hasil seperti pada tabel.

No	Uji	Hasil
1	nslookup api.warungku.id	alamat 203.0.113.25
2	ping 203.0.113.25	ada balasan
3	buka https://api.warungku.id/produk	kode 404 Not Found
4	buka https://api.warungku.id/v2/produk	kode 200 OK, daftar produk tampil

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Gangguan berada pada DNS karena aplikasi tidak dapat menemukan alamat server.	○	○
Server dapat dijangkau dan menanggapi permintaan HTTP yang dikirim kepadanya.	○	○
Perbaikan yang tepat adalah menyesuaikan alamat endpoint yang dipanggil aplikasi menjadi /v2/produk.	○	○

Pembahasan **Kunci: 1S 2B 3B**

Konsep: Komunikasi aplikasi dengan server melalui tahap: DNS menerjemahkan nama menjadi alamat IP, paket diantar ke alamat itu, lalu client mengirim permintaan HTTP dan server membalas dengan kode status (200 berhasil, 404 alamat sumber daya tidak ditemukan, 500 galat di server).

Langkah:

1. Uji 1 berhasil → DNS baik.
2. Uji 2 berhasil → server terjangkau.
3. Uji 3 → server menjawab 404: alamat /produk tidak ada.
4. Uji 4 → alamat /v2/produk berhasil: endpoint sudah berpindah.

Analisis setiap pernyataan:

• **Pernyataan 1 → Salah.** Uji 1 berhasil: nama api.warungku.id diterjemahkan menjadi 203.0.113.25. DNS tidak bermasalah.

• **Pernyataan 2 → Benar.** Uji 2 mendapat balasan, dan uji 3–4 mendapat kode respons dari server. Kode 404 pun adalah jawaban server.

• **Pernyataan 3 → Benar.** Alamat /produk tidak ditemukan (404), sedangkan /v2/produk berhasil (200). Aplikasi masih memanggil alamat lama.

Jawaban: Salah, Benar, Benar.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap soal, tulis apa yang dibuktikan setiap hasil uji atau informasi, lalu periksa pernyataan satu per satu.

1. Aplikasi pemesanan makanan KantinKu (2027) dipasang di ponsel siswa. Ponsel mengirim permintaan pesanan ke alamat `http://kantinku.sch.id/api/pesan`, lalu server menyimpan pesanan dan mengirim balasan. Pada jam istirahat, sebagian pesanan gagal dengan kode status 500. Pemeriksaan keamanan juga menemukan bahwa data login masih dikirim lewat http://, bukan https://.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang aplikasi tersebut adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Ponsel siswa berperan sebagai client yang mengirim permintaan, lalu server mengirim respons.

☐ B. Mengganti http:// menjadi https:// membuat data login terenkripsi selama dikirim.

☐ C. Dengan HTTPS, pengguna lain di Wi-Fi sekolah tetap dapat membaca kata sandi yang dikirim.

☐ D. Saat server mati, aplikasi tetap dapat menyimpan pesanan karena menu tersimpan di ponsel.

☐ E. Kode status 500 menunjukkan bahwa galat terjadi di sisi server, bukan di ponsel siswa.

2. Satu server di SMK Nusantara (alamat IP 192.168.1.5) menjalankan dua program sekaligus: aplikasi web perpustakaan di port 80 dan basis data MySQL di port 3306 (2027). Siswa membuka aplikasi perpustakaan dari laptop dengan alamat `http://192.168.1.5`, sedangkan aplikasi web itu sendiri mengambil data dari MySQL di server yang sama.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Server dapat membedakan permintaan untuk aplikasi web dan untuk basis data dari nomor port tujuannya.	○	○
Karena alamat IP-nya sama, laptop yang membuka http://192.168.1.5 akan terhubung ke MySQL bila MySQL dijalankan lebih dulu.	○	○
Permintaan HTTP ke port 80 dapat dikirim tanpa koneksi TCP karena HTTP tidak membutuhkan protokol lapisan transport.	○	○

Kunci singkat: 1. A, B, E 2. 1B 2S 3S

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa kode 404 tidak berarti server mati, dan siapa yang seharusnya memperbaikinya?
- Mengapa port basis data tidak dibuka ke internet walaupun aplikasi web membutuhkan basis data itu?
- **Yang harus kuingat:** klien memulai dan server menjawab; 4xx salah di klien, 5xx salah di server; DNS menerjemahkan nama, port memilih program; aplikasi memakai arsitektur tiga lapis; buka firewall seperlunya; uji dari bawah ke atas.

Latihan interaktif di LMS

Sub bab 3.4: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-3.4

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kuis Akhir Bab 3

SMKPPLG-KB3

Jumlah soal: 25 **Waktu anjuran:** 60 menit

Kerjakan sendiri dan berwaktu. Setelah selesai, cocokkan dengan kunci di Bagian D, isi tabel skor, lalu lihat anjuran di bawahnya.

Kerjakan Kuis Bab 3 di LMS

Skor dan progresmu tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-KB3

1. Lina membereskan folder tugas gambar di terminal Linux (2027). Awalnya folder aktif berisi berkas dan folder berikut, lalu ia menjalankan tiga perintah.

```
$ ls
a.png  b.png  c.txt  d.png  arsip
$ cp *.png arsip/
$ mv c.txt arsip/catatan.txt
$ rm b.png
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Hasil perintah `ls` pada folder aktif, lalu `ls arsip`, berturut-turut adalah ....

A. a.png arsip d.png lalu a.png d.png catatan.txt

B. a.png arsip c.txt d.png lalu a.png b.png catatan.txt d.png

C. a.png arsip d.png lalu a.png b.png catatan.txt d.png

D. arsip d.png lalu a.png b.png catatan.txt d.png

E. a.png arsip d.png lalu b.png catatan.txt d.png

2. Tomi menjalankan skrip cadangan di terminal Linux (2027), tetapi muncul pesan galat berikut.

```
$ ./cadangan.sh
bash: ./cadangan.sh: Permission denied
$ ls -l cadangan.sh
-rw-r--r-- 1 tomi tomi 842 Jun 12 09:20 cadangan.sh
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang galat tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Berkas belum memiliki izin execute (x) bagi pemilik, sehingga tidak dapat dijalankan
☐ B. Perintah `chmod u+x cadangan.sh` dapat memperbaikinya dengan menambah izin jalankan bagi pemilik
☐ C. Berkas sebaiknya diberi izin 777 supaya semua pengguna dapat menjalankannya tanpa masalah lagi
☐ D. Pemilik berkas memang tomi, sehingga masalahnya terletak pada izin berkas, bukan pada kepemilikan
☐ E. Bash perlu dipasang ulang karena galat tersebut berarti program bash rusak

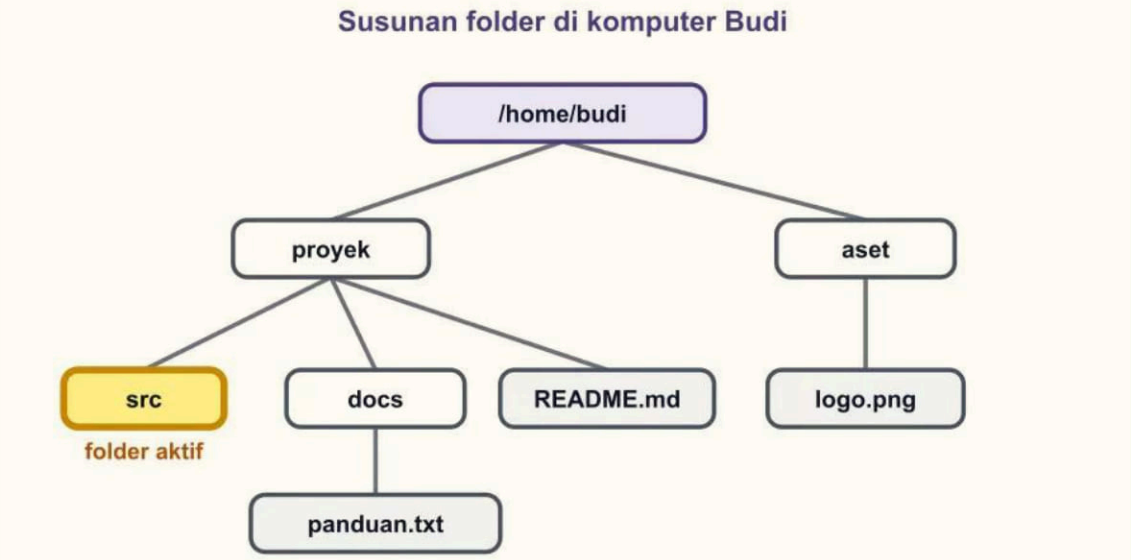
3. Tim proyek Python (2027) sudah menyimpan daftar pustaka beserta versinya dalam berkas requirements.txt di repositori. Seorang anggota baru ingin memasang semua pustaka itu di komputernya sambil lingkungan virtual proyek sedang aktif.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perintah yang tepat dijalankan anggota baru tersebut adalah ....

A. `pip install -r requirements.txt`
B. `pip freeze > requirements.txt`
C. `python -m venv requirements.txt`
D. `pip install requirements.txt`
E. `python requirements.txt`

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Gambar menunjukkan susunan folder di komputer Budi (2027). Terminal Budi sedang berada di folder `src` (ditandai kuning), lalu ia mengetik perintah `cat ../docs/panduan.txt`.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Berkas yang dibaca perintah itu berada pada path absolut ....



A. `/home/budi/proyek/src/docs/panduan.txt`
B. `/home/budi/proyek/docs/panduan.txt`
C. `/home/budi/docs/panduan.txt`
D. `/home/proyek/docs/panduan.txt`
E. `/home/budi/aset/docs/panduan.txt`

5. Administrator server sebuah studio (2027) akan mengatur izin tiga berkas sesuai kebutuhan pada tabel berikut. Nilai izin chmod dihitung dari r = 4, w = 2, x = 1 untuk pemilik, grup, dan lainnya, berurutan.

Berkas	Kebutuhan akses
backup.sh	pemilik: baca, tulis, jalankan; grup: baca dan jalankan; lainnya: tidak ada
catatan.txt	pemilik dan grup: baca dan tulis; lainnya: hanya baca
kunci.pem	pemilik: hanya baca; grup dan lainnya: tidak ada

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Nilai chmod untuk backup.sh, catatan.txt, dan kunci.pem secara berurutan adalah ....

A. 750, 664, 600
B. 755, 664, 400
C. 750, 646, 400
D. 740, 664, 400
E. 750, 664, 400

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

6. Rina menyiapkan proyek Python (2027) dan menjalankan urutan perintah berikut. Setelah itu muncul galat pada baris terakhir.

```
$ pip install requests
$ python -m venv .venv
$ source .venv/bin/activate
(.venv) $ python ambil.py
ModuleNotFoundError: No module named 'requests'
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Penjelasan dan perbaikan yang paling tepat adalah ....

A. pustaka requests rusak atau tidak lengkap, sehingga harus diunduh manual dari situs resminya lalu disalin
B. pip dijalankan sebelum venv aktif, jadi requests terpasang di Python global; ulangi pip install saat venv aktif
C. Python harus dipasang ulang agar dapat mengenali lingkungan virtual yang baru dibuat oleh perintah venv itu
D. folder .venv harus dihapus lalu dibuat ulang dengan nama lain supaya pustaka lama tidak ikut terbaca
E. perintah source seharusnya dijalankan lebih dulu sebelum perintah python -m venv dijalankan sama sekali

7. Pengelola lab komputer sebuah SMK (2027) ingin menghubungkan jaringan lab (192.168.50.0/24) dengan jaringan milik penyedia layanan internet sehingga komputer di lab dapat membuka situs di internet.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perangkat yang tepat untuk menghubungkan dua jaringan yang berbeda itu adalah ....

A. switch
B. hub
C. access point
D. kabel UTP
E. router

8. Sari menjalankan serangkaian perintah berikut di terminal Linux (2027). Folder aktif awalnya `/home/sari/tugas` yang berisi folder `gambar` dan berkas `laporan.txt`.

```
$ pwd
/home/sari/tugas
$ mkdir arsip
$ cp laporan.txt arsip/
$ cd gambar
$ mv ../laporan.txt ../arsip/laporan_lama.txt
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Setelah seluruh perintah dijalankan, keadaan yang benar adalah ....

A. arsip hanya berisi laporan.txt, dan laporan.txt masih ada di folder tugas
B. arsip hanya berisi laporan_lama.txt, dan laporan.txt tidak ada di folder tugas
C. arsip berisi laporan.txt dan laporan_lama.txt, dan laporan.txt masih ada di folder tugas
D. arsip berisi laporan.txt dan laporan_lama.txt, dan laporan.txt tidak ada lagi di folder tugas
E. arsip berisi laporan.txt dan laporan_lama.txt, dan laporan.txt berpindah ke folder gambar

9. Fajar membuat aplikasi web kecil dan menjalankannya di laptop pada port 5000 (2027). Di laptop, alamat `http://localhost:5000` terbuka normal. Ia lalu mencoba membuka alamat yang sama dari ponselnya yang tersambung ke data seluler, tetapi gagal.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Di ponsel, localhost menunjuk ponsel itu sendiri, bukan laptop Fajar
☐ B. Ponsel perlu memakai alamat IP laptop pada jaringan yang sama untuk membuka aplikasi
☐ C. Mengganti port menjadi 80 pasti membuat alamat localhost dapat dibuka dari ponsel
☐ D. Laptop dan ponsel harus tersambung ke jaringan yang sama, bukan data seluler
☐ E. Python perlu dipasang di ponsel agar alamat localhost laptop dapat dibuka

10. Seorang pengembang menguji akses ke server arsip studio (2027) dari laptopnya dan mencatat hasil uji pada tabel berikut.

Uji	Hasil
Lampu port kabel di laptop	menyala
ping 10.30.1.20 (alamat IP server)	balasan diterima
ping arsip.studio.id (nama server)	gagal: nama tidak dikenali
Membuka http://10.30.1.20:8080 di peramban	berhasil

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Layer yang paling mungkin bermasalah adalah ....

A. layer akses jaringan, karena kabel perlu diganti
B. layer internet, karena alamat IP server tidak dapat dijangkau
C. layer transport, karena port 8080 diblokir firewall
D. layer aplikasi, tepatnya penerjemahan nama oleh DNS
E. tidak ada layer yang bermasalah karena sebagian uji berhasil

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

11. Studio Samudra Biru (2027) merancang beberapa fitur gim kapal daring: login pemain, posisi kapal waktu nyata, obrolan teks antarpemain, dan pengunduhan berkas pembaruan gim. Tim memilih protokol transport yang sesuai untuk tiap fitur.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pilihan yang tepat adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Login pemain memakai TCP karena kata sandi tidak boleh hilang atau tertukar urutannya
☐ B. Posisi kapal waktu nyata dapat memakai UDP karena kiriman posisi lama tidak berguna lagi
☐ C. Pengunduhan berkas pembaruan memakai UDP karena UDP paling andal
☐ D. Obrolan teks memakai TCP agar pesan sampai lengkap dan berurutan
☐ E. TCP selalu lebih cepat dari UDP sehingga dipakai untuk semua fitur

12. Aplikasi ponsel pelaporan sampah (2027) memanggil server aplikasi di awan lewat HTTPS. Server aplikasi menyimpan data ke basis data MySQL (port 3306) di server lain, dan administrator masuk ke server aplikasi lewat SSH (port 22). Administrator sedang menyusun aturan firewall.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Aturan yang tepat adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Membuka port 3306 ke seluruh internet agar ponsel dapat menyimpan data langsung ke basis data
☐ B. Membuka port 443 pada server aplikasi agar ponsel dapat memanggilnya
☐ C. Menutup port basis data dari internet sehingga hanya server aplikasi yang dapat menjangkaunya
☐ D. Membuka semua port agar tidak ada sambungan yang gagal
☐ E. Membatasi port 22 hanya untuk alamat milik administrator

13. Aplikasi pemesanan Warung Nusa (2027) mengirim permintaan HTTP berikut ke server.

```
POST /api/v1/pesanan HTTP/1.1
Host: warung.id
Content-Type: application/json

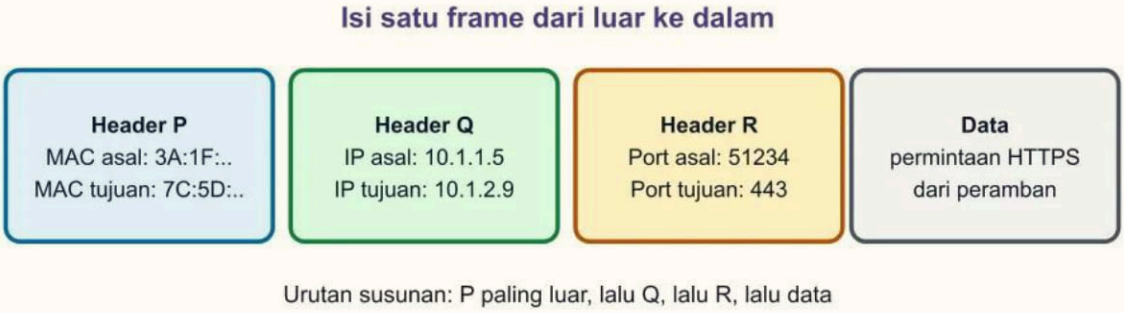
{"menu": "nasi goreng", "jumlah": 2}
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang permintaan tersebut adalah ....

A. memakai metode POST untuk membuat data pesanan baru
B. memakai metode GET untuk membaca daftar pesanan yang tersimpan di server
C. memakai metode DELETE untuk membatalkan pesanan yang sudah dibuat
D. memakai metode PUT untuk membuat data pesanan yang baru di server
E. memakai metode POST untuk membaca isi pesanan yang sudah tersimpan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

14. Seorang pengembang menangkap satu frame data dari aplikasi pemesanan memakai alat analisis jaringan (2027). Isi frame itu, dari luar ke dalam, ditunjukkan pada gambar; nama ketiga header disamarkan menjadi P, Q, dan R.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang gambar tersebut adalah .... (Jawaban benar lebih dari satu.)



Pilih semua jawaban yang benar.

☐ A. Header R ditambahkan pada layer transport karena memuat nomor port
☐ B. Header Q ditambahkan pada layer akses jaringan karena memuat alamat IP
☐ C. Header P ditambahkan paling akhir saat pengiriman dan dibuka paling awal oleh penerima
☐ D. Header R memuat alamat IP sehingga data dapat dirutekan antarjaringan
☐ E. Port tujuan 443 menunjukkan data ditujukan ke layanan HTTPS pada server

15. Seorang siswa menyambungkan laptopnya ke Wi-Fi laboratorium dan menjalankan perintah `ipconfig` (2027). Hasilnya:

```
> ipconfig
   IPv4 Address. . . . . . . : 169.254.31.7
   Subnet Mask . . . . . . . : 255.255.0.0
   Default Gateway . . . . . :
```

Laptop itu diatur untuk memperoleh alamat secara otomatis.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kesimpulan yang paling tepat adalah ....

A. alamat IP laptop diatur manual dengan benar oleh siswa sehingga jaringan seharusnya sudah berfungsi normal
B. server DNS sedang rusak sehingga nama situs tidak dapat dibuka dari laptop itu
C. gateway terisi otomatis dengan benar sehingga internet dapat dipakai seperti biasa
D. laptop gagal memperoleh alamat dari DHCP, sehingga sambungan jaringan dan layanan DHCP yang diperiksa
E. aplikasi peramban di laptop perlu dipasang ulang agar dapat membuka situs

16. Sebuah studio gim (2027) membandingkan protokol transport untuk gim balap daring yang akan dirilis.

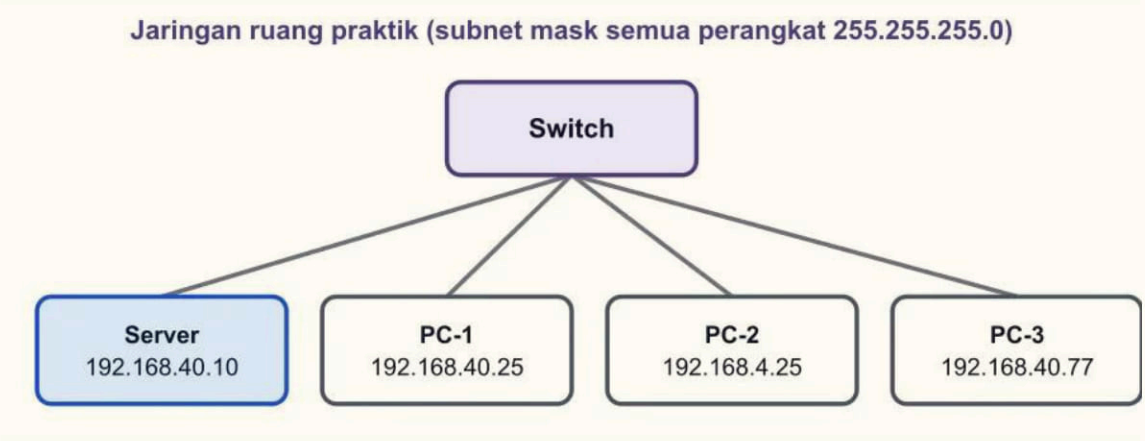
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
TCP selalu lebih cepat daripada UDP karena memastikan semua data sampai.	○	○
Posisi mobil pemain pada gim waktu nyata dapat memakai UDP karena kiriman lama yang terlambat lebih baik dilewati.	○	○
Pada TCP, kedua pihak melakukan jabat tangan tiga langkah (SYN, SYN-ACK, ACK) sebelum data dikirim.	○	○

17. Sebuah ruang praktik (2027) memakai satu switch yang menghubungkan satu server dan tiga komputer, seperti pada gambar. Seluruh perangkat memakai subnet mask 255.255.255.0 dan tidak ada router pada jaringan ini. Teknisi menemukan bahwa satu komputer tidak dapat berhubungan langsung dengan server karena alamat IP-nya berada di jaringan yang berbeda.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Komputer yang harus diperbaiki alamat IP-nya adalah ....



A. PC-2
B. PC-1
C. PC-3
D. Switch
E. Server

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

18. Jaringan kantor sebuah studio (2027) memakai alamat 10.20.8.0/24 dengan gateway 10.20.8.1. Layanan DHCP memberikan alamat dalam rentang 10.20.8.50 sampai 10.20.8.150. Alamat yang sudah dipakai secara statis ditunjukkan pada tabel berikut.

Alamat	Dipakai oleh
10.20.8.1	Router
10.20.8.5	Server berkas
10.20.8.12	Printer
10.20.8.40	Penyimpanan jaringan

Sebuah server pengujian baru memerlukan alamat statis.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Alamat yang paling tepat untuk server pengujian tersebut adalah ....

A. 10.20.8.255
B. 10.20.8.12
C. 10.20.8.30
D. 10.20.8.100
E. 10.20.9.30

19. Seorang siswa yang sudah masuk (login) ke portal sekolah (2027) membuka halaman pengaturan nilai milik guru, lalu peramban menampilkan kode status 403 Forbidden. Alamat halaman itu benar dan server portal berjalan normal.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang paling tepat tentang kode status 403 tersebut adalah ....

A. server portal sedang mati sehingga permintaan siswa tidak dapat diproses sama sekali
B. alamat halaman salah ditulis sehingga halaman itu tidak ditemukan di server
C. jaringan siswa terputus sehingga server tidak pernah menerima permintaannya
D. server gagal memproses permintaan yang sah sehingga perbaikan dilakukan di server
E. permintaan ditolak karena akun siswa tidak memiliki hak mengakses halaman itu

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

20. Seorang pengembang menguji layanan API di server pengujian studio (2027) dari laptopnya dan mencatat hasil berikut: nslookup api.studio.id menjawab 10.5.5.9; ping 10.5.5.9 berhasil; koneksi ke 10.5.5.9 port 8080 ditolak; sedangkan setelah masuk ke server itu lewat SSH, perintah curl ke localhost port 8080 mengembalikan jawaban yang normal.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Penyebab yang paling mungkin adalah ....

A. DNS tidak dapat menerjemahkan nama server menjadi alamat IP yang benar
B. alamat IP server salah sehingga server tidak dapat dijangkau dengan perintah ping
C. firewall server menutup port 8080 dari luar, padahal layanannya berjalan
D. kabel jaringan laptop terputus sehingga tidak ada sambungan sama sekali
E. alamat IP server bentrok dengan perangkat lain di jaringan studio

21. Seorang teknisi jaringan SMK (2027) menjelaskan pengaturan jaringan lab kepada siswa baru: komputer lab memakai mask 255.255.255.0, gateway 192.168.60.1, dan memperoleh alamat otomatis.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Dalam satu jaringan yang sama, dua komputer tetap dapat saling berhubungan walaupun gateway salah satunya keliru.	○	○
DHCP bertugas menerjemahkan nama situs menjadi alamat IP.	○	○
ID jaringan 192.168.60.0 dan alamat broadcast 192.168.60.255 boleh dipasang pada komputer.	○	○

22. Seorang siswa mempelajari cara kerja aplikasi pemesanan daring (2027) yang terdiri atas aplikasi ponsel, server aplikasi, dan basis data. Ia mencatat tiga pernyataan berikut.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Aplikasi ponsel sebaiknya menyimpan pesanan langsung ke basis data tanpa melalui server aplikasi karena lebih aman.	○	○
Pada URL https://toko.id/produk yang tidak menuliskan nomor port, dipakai port bawaan 443.	○	○
Kode status 4xx menandakan kesalahan di sisi klien, misalnya alamat yang diminta tidak ada.	○	○

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

23. Pada model TCP/IP, sebuah pesan melewati beberapa layer saat dikirim. Sebuah peramban mengirim permintaan ke server web dan harus menentukan program mana di server yang dituju, misalnya port 443.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Nomor port ditentukan pada layer ....

A. aplikasi
B. transport
C. internet
D. akses jaringan
E. semua layer sekaligus

24. Teknisi mencatat konfigurasi tiga komputer di jaringan sebuah studio (2027) seperti pada tabel berikut.

Perangkat	Alamat IP	Subnet mask	Gateway
A	192.168.12.10	255.255.255.0	192.168.12.1
B	192.168.12.45	255.255.255.0	192.168.1.1
C	192.168.21.10	255.255.255.0	192.168.21.1

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang konfigurasi tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. A dan B berada pada satu jaringan sehingga dapat saling berhubungan walaupun gateway B keliru
☐ B. Gateway B keliru, sehingga B tidak dapat keluar ke jaringan lain atau internet lewat gateway itu
☐ C. A dan C berada pada satu jaringan karena dua oktet pertama alamatnya sama
☐ D. B dan C dapat bertukar data tanpa router karena keduanya memakai subnet mask yang sama
☐ E. C tidak dapat langsung menjangkau A karena oktet ketiga alamatnya berbeda (21 dan 12)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

25. Log permintaan ke API aplikasi pemesanan Warung Nusa (2027) ditunjukkan pada tabel berikut.

No	Permintaan	Kode status
1	GET /api/v2/menu	200
2	GET /api/v1/menu	404
3	POST /api/v2/pesanan	201
4	GET /api/v2/laporan (tanpa login)	401
5	POST /api/v2/pesanan	503

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang log tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Permintaan 2 gagal karena alamat versi 1 tidak ditemukan di server, bukan karena server mati

☐ B. Permintaan 3 gagal membuat pesanan karena kodenya bukan 200

☐ C. Permintaan 4 ditolak karena pengguna belum masuk (login)

☐ D. Permintaan 5 menunjukkan layanan sedang tidak tersedia di sisi server

☐ E. Kode 200 pada permintaan 1 menandakan kesalahan di sisi klien

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi	1, 2, 3, 4, 5, 6, 8, 9	___ / 8	Ulangi hlm. 104
Sub bab 3.2 · Konfigurasi jaringan dasar	7, 15, 17, 18, 21, 24	___ / 6	Ulangi hlm. 118
Sub bab 3.3 · Layer TCP/IP dan fungsi protokol	10, 11, 14, 16, 23	___ / 5	Ulangi hlm. 133
Sub bab 3.4 · Mekanisme komunikasi jaringan	12, 13, 19, 20, 22, 25	___ / 6	Ulangi hlm. 145

Skor 70 % atau lebih: lanjut ke bab berikutnya. Skor 50–69 %: ulangi sub bab pada tabel yang kurang dari separuh benar. Skor di bawah 50 %: baca ulang bab ini dari awal. (Angka ini anjuran belajar, bukan prediksi nilai TKA.)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bab 4 · Pemrograman Terstruktur

SMKPPLG-B4

Setelah mempelajari bab ini, kamu dapat:

- menganalisis penggunaan tipe data dan struktur data dalam potongan program
- menelusuri alur percabangan dan perulangan serta mengevaluasi efektivitasnya
- membagi program menjadi fungsi dan modul, lalu menilai desain modularnya

Peta bab

Sub bab	Judul
4.1	Tipe data dan struktur data
4.2	Struktur kontrol percabangan
4.3	Struktur kontrol perulangan
4.4	Fungsi, prosedur, dan modularisasi

Bab 4 di LMS

Buka seluruh latihan bab ini di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-B4

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 4.1 · Tipe data dan struktur data

SMKPPLG-4.1

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menganalisis penggunaan tipe data dan struktur data dalam potongan program**, termasuk penerapannya untuk memecahkan masalah. Pada soal, ini hampir selalu berbentuk kasus pendek berisi kode Python: sebuah program kecil dengan beberapa variabel, lalu kamu diminta **menentukan keluarannya**, **menemukan baris yang menimbulkan galat**, atau **memilih struktur data yang paling tepat** untuk sebuah kebutuhan.

Soal-soal ini tampak mudah, tetapi sering menyimpan satu jebakan kecil: teks `"15000"` yang bukan angka, pembagian yang menghasilkan desimal, indeks yang dimulai dari 0, atau `[baris][kolom]` yang terbalik. Kuncinya adalah **menelusuri langkah demi langkah** dan selalu **menentukan tipe tiap nilai**.

Setelah mempelajari sub bab ini, kamu dapat:

- menyebutkan tipe data dasar Python dan menentukan tipe hasil sebuah operasi;
- memakai konversi tipe dan memahami mengapa `input()` selalu menghasilkan teks;
- menghitung dengan operator `/`, `//`, `%`, dan `**`, serta memahami prioritasnya;
- mengakses elemen list, string, dan list dua dimensi dengan indeks dan irisan;
- memakai dictionary, set, dan tuple dengan benar serta memilih struktur data yang tepat untuk sebuah kebutuhan.

◆ KONSEP INTI

Program komputer pada dasarnya mengolah **data**. Data itu harus disimpan dengan cara yang benar, dan cara menyimpan yang benar bergantung pada **jenis** datanya (tipe data) dan **bentuk kumpulannya** (struktur data). Tanpa itu, program menghasilkan keluaran yang keliru atau berhenti dengan galat. Contoh kode di buku ini memakai **Python 3**.

1. Variabel dan tipe data

Variabel adalah nama yang menunjuk sebuah nilai di memori. Di Python, variabel dibuat dengan **tanda sama dengan (=)**, yang berarti "simpan nilai di sisi kanan ke nama di sisi kiri", bukan "sama dengan" dalam arti matematika.

```
nama = "Kancil"
nyawa = 3
kecepatan = 2.5
aktif = True
print(type(nama), type(nyawa), type(kecepatan), type(aktif))
```

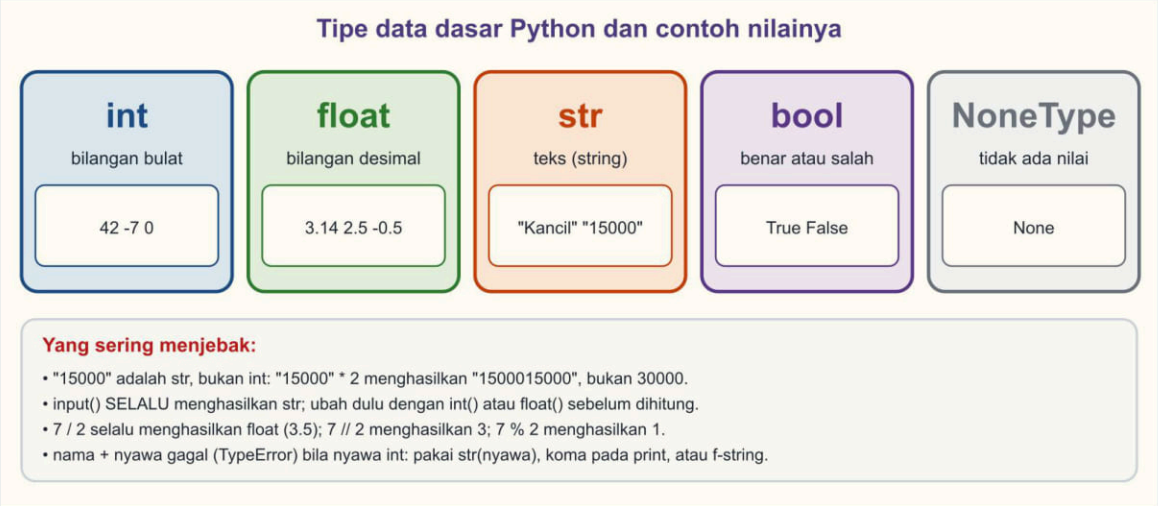
Keluarannya: `<class 'str'> <class 'int'> <class 'float'> <class 'bool'>`. Fungsi

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

`type()` memperlihatkan tipe sebuah nilai. Python menentukan tipe dari nilainya sendiri (tidak perlu ditulis), dan sebuah variabel boleh diisi ulang dengan tipe lain.

Tipe data dasar (Gambar 4.1):

- **`int`**: bilangan bulat, seperti `42`, `-7`, `0`.
- **`float`**: bilangan desimal, seperti `3.14`, `2.5`. Memakai **titik** sebagai pemisah desimal.
- **`str`**: teks (*string*), ditulis dengan tanda kutip: `"Kancil"`, `'halo'`, dan juga `"15000"`.
- **`bool`**: benar atau salah, hanya `True` atau `False` (huruf pertama besar).
- **`None`**: tanda "tidak ada nilai".



Gambar 4.1 Tipe data dasar Python, contoh nilai, dan jebakan yang sering muncul. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perbedaan yang paling sering dipakai soal: `15000` adalah **angka** (`int`), tetapi `"15000"` adalah **teks** yang kebetulan berisi digit. Komputer memperlakukan keduanya sangat berbeda.

2. Konversi tipe

Fungsi **`input()`** membaca ketikan pengguna dan **selalu menghasilkan `str`**, bahkan jika yang diketik adalah angka. Untuk menghitung, nilainya harus diubah dulu dengan fungsi konversi:

- `int("15")` → `15` (teks ke bilangan bulat);
- `float("2.5")` → `2.5`;
- `str(15)` → `"15"` (angka ke teks).

Bila teksnya bukan angka yang sah, konversi gagal dengan **`ValueError`**, misalnya `int("abc")` atau `int("2.5")`.

Cara aman menghitung dari input:

```
harga = int(input("Harga satuan: "))   # dikonversi sekaligus
jumlah = 2
print(harga * jumlah)
```

Bila `harga` tidak dikonversi, `harga * jumlah` bukan perkalian bilangan. Operator `*` pada teks berarti **mengulang teks**:

```
a = "12"
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
print(a * 3)         # teks diulang tiga kali
print(int(a) * 3)    # perkalian bilangan
```

Keluarannya `121212` untuk baris pertama dan `36` untuk baris kedua.

Menggabungkan teks dengan angka juga memerlukan perhatian. Operator `+` pada dua teks **menyambungkannya**, tetapi `+` antara teks dan angka menimbulkan **`TypeError`**:

```
nama = "Kancil"
nyawa = 3
print(nama + " nyawa " + nyawa)        # TypeError: harus str
print(nama + " nyawa " + str(nyawa))   # benar
print(nama, "nyawa", nyawa)            # benar: koma menyambung otomatis
print(f"{nama} nyawa {nyawa}")         # benar: f-string
```

Hanya baris pertama yang menimbulkan galat. Tiga cara lainnya benar: `str()` mengubah angka menjadi teks, koma pada `print` memisahkan bagian dengan spasi dan mengubah tiap bagian menjadi teks, dan **f-string** (teks berawalan `f` dengan `{}` untuk menyisipkan nilai) menyisipkan nilai langsung.

3. Operator aritmetika

Operator	Arti	Contoh	Hasil
`+ - *`	tambah, kurang, kali	`3 * 4`	`12`
`/`	bagi (**selalu `float`**)	`7 / 2`	`3.5`
`//`	bagi bulat (hasil dibulatkan **ke bawah**)	`7 // 2`	`3`
`%`	sisa bagi (*modulus*)	`7 % 2`	`1`
`**`	pangkat	`2 ** 3`	`8`

Beberapa hal yang menentukan jawaban soal:

- **`/` selalu menghasilkan `float`**: `8 / 2` adalah `4.0`, bukan `4`.
- **`//` membagi bulat ke bawah**: `7 // 2` adalah 3, dan pada bilangan negatif `-7 // 2` adalah `-4` (dibulatkan ke arah yang lebih kecil).
- **`%` memberi sisa**: `7 % 2` adalah 1. Pada kasus nyata, `%` dipakai menguji genap atau ganjil (`n % 2 == 0` berarti genap) dan membagi rata (7 voucher untuk 2 kelompok: `7 // 2 = 3` voucher tiap kelompok dan `7 % 2 = 1` voucher sisa).
- **`int` dengan `float` menghasilkan `float`**: `3 * 2.5` adalah `7.5`.
- **Prioritas operator**: `**` lebih dulu, lalu `* / // %`, lalu `+ -`. Operator setingkat dihitung dari kiri ke kanan. Jadi `2 + 3 * 4` adalah `14`, sedangkan `(2 + 3) * 4` adalah `20`. Gunakan **kurung** agar maksudnya jelas.

Operator **perbandingan** (`==`, `!=`, `<`, `>`, `<=`, `>=`) menghasilkan `bool`: `5 > 3` adalah `True`. Perhatikan bahwa `=` menyimpan nilai, sedangkan `==` membandingkan. Operator perbandingan dan logika dibahas lebih dalam pada sub bab 4.2.

4. String: teks sebagai deret karakter

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

String adalah deret karakter yang urutannya bernomor. Nomor itu disebut **indeks** dan **dimulai dari 0**.

```
kata = "Kancil"
print(kata[0], kata[-1], kata[1:4], len(kata))
```

Keluarannya `K l anc 6`. `kata[0]` karakter pertama, `kata[-1]` karakter terakhir, `kata[1:4]` irisan dari indeks 1 sampai **3** (indeks 4 tidak ikut), dan `len()` memberi panjang. Beberapa metode string: `kata.upper()` (huruf besar), `kata.lower()`, `" a ".strip()` (membuang spasi di tepi), `"a,b".split(",")` (memecah menjadi list). String **tidak dapat diubah di tempat**; metode seperti `upper()` menghasilkan string baru.

5. List: kumpulan berurutan yang dapat diubah

List menyimpan banyak nilai berurutan dalam kurung siku, dan **dapat diubah** (tambah, hapus, ganti). Indeksnya sama seperti string: maju dari 0 dan mundur dari -1 (Gambar 4.2).



Gambar 4.2 Indeks maju dan mundur pada list beserta irisan. Sumber: digambar oleh Tim Next Level Study Indonesia.

```
barang = ["pensil", "buku", "penghapus"]
barang.append("penggaris")     # tambah di akhir
print(barang[0], barang[-1], len(barang))
print(barang.pop())            # ambil dan hapus yang TERAKHIR
print(barang.pop(0))           # ambil dan hapus yang PERTAMA
print(barang)
barang.remove("buku")          # hapus berdasarkan nilai
print(barang, "buku" in barang)
```

Keluarannya berurutan: `pensil penggaris 4`, lalu `penggaris`, lalu `pensil`, lalu `['buku', 'penghapus']`, dan terakhir `['penghapus'] False`.

Hal-hal yang harus dipahami:

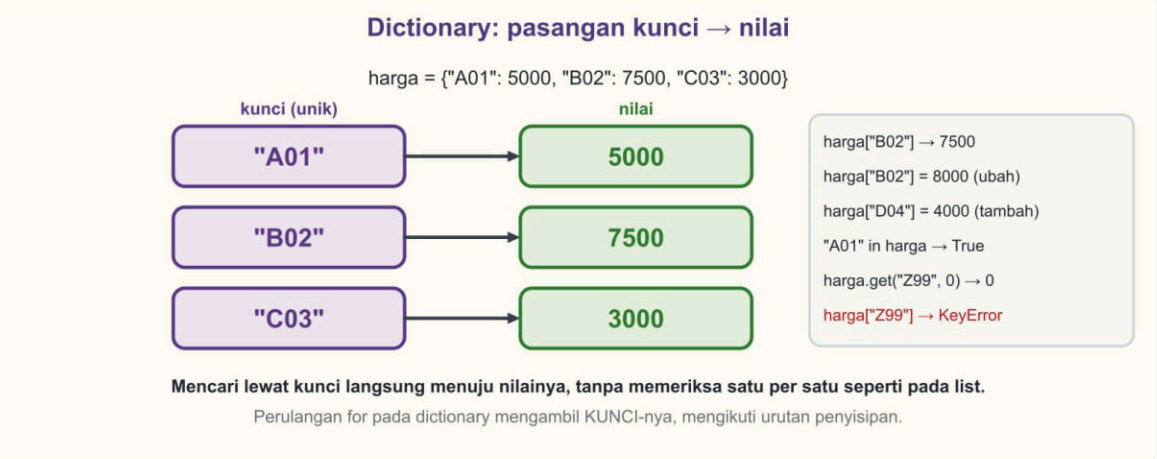
- **`append(x)`** menambah di akhir; **`pop()`** mengambil dan menghapus elemen **terakhir**; **`pop(0)`** mengambil yang **pertama**; **`remove(x)`** menghapus berdasarkan nilai; **`in`** menguji keberadaan nilai.
- **`len(daftar)`** memberi banyak elemen. Indeks terakhir adalah **`len - 1`**. Mengakses indeks yang tidak ada menimbulkan **`IndexError`**.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **Tumpukan (*stack*)**: elemen yang masuk terakhir keluar pertama (`append` dan `pop()`), cocok untuk tombol "Urungkan" yang membatalkan aksi paling akhir. **Antrean (*queue*)**: yang masuk pertama keluar pertama (`append` dan `pop(0)`).

6. Tuple, set, dan dictionary

- **Tuple**, ditulis dengan kurung biasa `(3, 4)`, mirip list tetapi **tidak dapat diubah**. Cocok untuk data tetap seperti koordinat titik.
- **Set**, ditulis dengan kurung kurawal `{1, 2, 3}`, menyimpan **nilai unik tanpa urutan**. Menambah nilai yang sudah ada tidak mengubah apa pun, sehingga cocok untuk daftar yang tidak boleh ganda dan untuk menguji "sudah ada?".
- **Dictionary** menyimpan **pasangan kunci dan nilai**, ditulis `{"A01": 5000}`. **Kunci harus unik**, dan dengan kunci kamu langsung menuju nilainya (Gambar 4.3).



Gambar 4.3 Dictionary menyimpan pasangan kunci dan nilai. Sumber: digambar oleh Tim Next Level Study Indonesia.

```
harga = {"A01": 5000, "B02": 7500}
harga["B02"] = 8000            # mengubah nilai
harga["C03"] = 3000            # menambah pasangan baru
print(harga["B02"], len(harga), "A01" in harga, harga.get("Z99", 0))
for kode in harga:
    print(kode, harga[kode])
```

Keluarannya: `8000 3 True 0`, lalu tiga baris `A01 5000`, `B02 8000`, `C03 3000`. Perhatikan: `harga["Z99"]` pada kunci yang tidak ada menimbulkan **`KeyError`**, sedangkan `harga.get("Z99", 0)` mengembalikan nilai bawaan `0`. Perulangan `for` pada dictionary mengambil **kuncinya**, sesuai urutan penyisipan, dan operator `in` memeriksa **kunci**, bukan nilai.

7. List dua dimensi: kisi dan papan

Elemen sebuah list boleh berupa list lagi. List yang berisi list disebut **list dua dimensi**, cocok untuk kisi seperti papan permainan, peta labirin, atau tabel nilai (Gambar 4.4). Elemennya diakses dengan **dua indeks, `[baris][kolom]`**.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 4.4 List dua dimensi: indeks pertama memilih baris, indeks kedua memilih kolom. Sumber: digambar oleh Tim Next Level Study Indonesia.

```
kebun = [["tomat", "cabai", "terong"],
         ["bayam", "kangkung", "sawi"],
         ["jagung", "kacang", "talas"]]
print(kebun[1][0], kebun[2][2], kebun[0][-1])
print(len(kebun), len(kebun[0]))
for baris in kebun:
    print(baris[1])
```

Keluarannya: `bayam talas terong`, lalu `3 3`, lalu `cabai`, `kangkung`, `kacang`. `kebun[1][0]` adalah baris 1 kolom 0; `len(kebun)` adalah **jumlah baris**; `len(kebun[0])` adalah **jumlah kolom**. Kesalahan yang paling sering: membalik urutan menjadi `[kolom][baris]`.

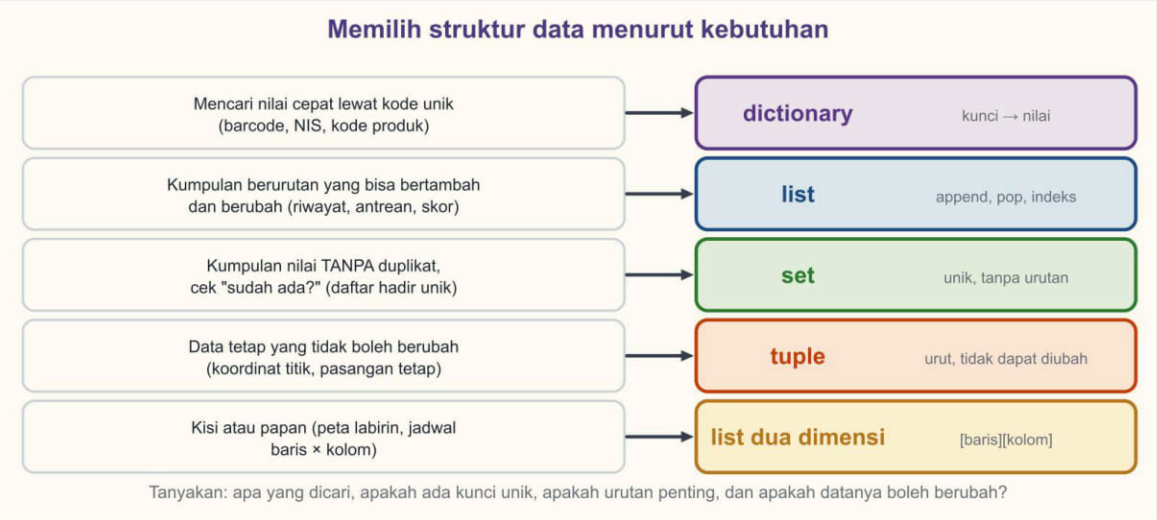
Satu konsep tambahan yang membantu memilih struktur data adalah **dapat diubah (*mutable*) dan tidak dapat diubah (*immutable*)**. **List, dictionary, dan set dapat diubah**: isinya boleh ditambah, dihapus, atau diganti setelah dibuat. **String, tuple, `int`, dan `float` tidak dapat diubah**: bila kamu menulis `kata = kata.upper()`, yang terjadi adalah dibuatnya nilai baru, bukan mengubah yang lama. Karena itu `kata[0] = "k"` pada string menimbulkan `TypeError`. Sifat ini menjelaskan mengapa koordinat tetap dipilih sebagai tuple (tidak sengaja berubah) dan mengapa riwayat aksi memakai list (terus bertambah).

Hal lain yang perlu kamu perhatikan adalah **nama variabel**. Python membedakan huruf besar dan kecil (`Nyawa` dan `nyawa` adalah dua variabel berbeda), nama tidak boleh diawali angka, dan tidak boleh memuat spasi. Nama yang deskriptif seperti `jumlah_voucher` membuat kode mudah dibaca, sebagaimana dibahas pada budaya mutu di sub bab 1.2.

8. Memilih struktur data yang tepat

Struktur data dipilih menurut **apa yang akan dilakukan dengan datanya** (Gambar 4.5).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 4.5 Kebutuhan dan struktur data yang tepat. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Mencari nilai lewat kode unik** (barcode, NIS, kode produk) → **dictionary**. Dengan kunci, kamu langsung menuju datanya; pada list kamu harus memeriksa satu per satu dari awal, dan pada 500 barang itu lambat. Dictionary juga mudah diperbarui, karena mengubah harga cukup dengan mengisi ulang satu kunci.
- **Kumpulan berurutan yang bertambah dan berubah** (riwayat aksi, antrean, daftar skor) → **list**.
- **Kumpulan tanpa ganda** (daftar kode yang sudah dipakai) → **set**.
- **Data tetap** (koordinat) → **tuple**.
- **Kisi atau tabel → list dua dimensi**.

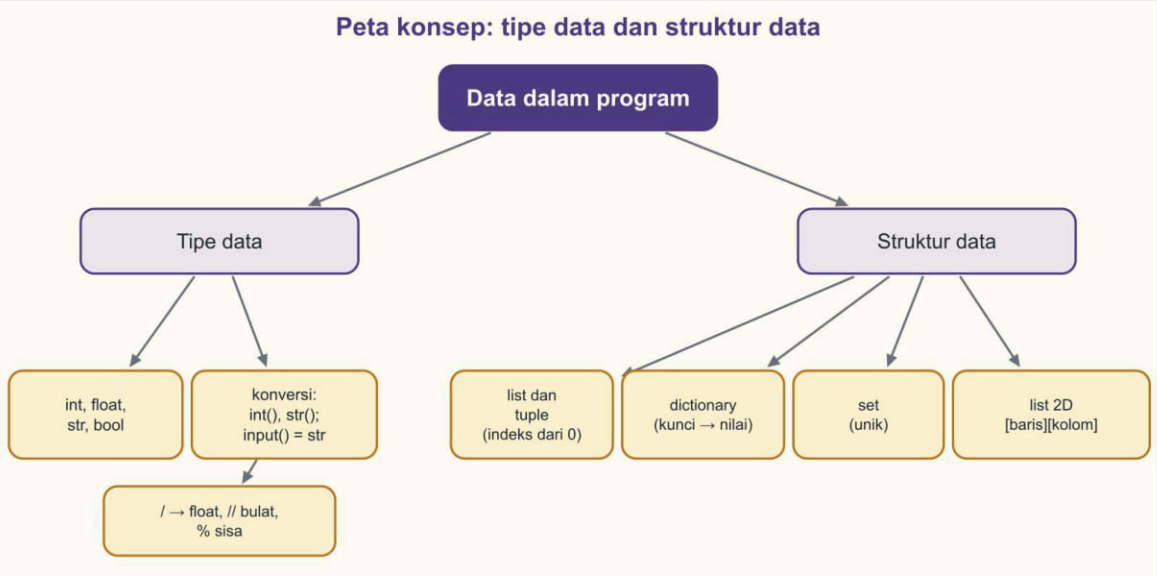
Dua list terpisah yang dicocokkan "menurut urutannya" rawan salah: bila urutan salah satu berubah, pasangannya ikut salah. Dictionary menyatukan pasangan itu sehingga tidak mungkin tertukar.

Satu kebiasaan baik saat menelusuri kode berstruktur data: **gambar datanya di kertas coretan**. Untuk list, tulis kotak-kotak berindeks; untuk dictionary, tulis dua kolom kunci dan nilai; untuk list dua dimensi, gambar kisi dengan nomor baris dan kolom. Dengan gambar itu, setiap operasi seperti `pop()` atau `harga["B02"] = 8000` dapat kamu coret dan tulis ulang, dan kesalahan indeks terlihat sebelum kamu memilih jawaban.

9. Galat yang sering muncul

- **`TypeError`**: operasi dengan tipe yang tidak cocok (misalnya `"a" + 1`).
- **`ValueError`**: nilai salah untuk fungsi (misalnya `int("abc")`).
- **`IndexError`**: indeks list di luar jangkauan.
- **`KeyError`**: kunci dictionary tidak ada.
- **`NameError`**: nama variabel belum dibuat.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 4.6 Peta konsep tipe data dan struktur data. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menelusuri keluaran dengan konversi tipe

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
jumlah = "12"
kali = 3
print(jumlah * kali)
print(int(jumlah) * kali)
print(jumlah + str(kali))
print(int(jumlah) / kali)
print(int(jumlah) // 5, int(jumlah) % 5)
```

1. Tentukan tipe: `jumlah` adalah `str` ("12"), `kali` adalah `int` (3).
2. Baris 1: `str * int` → teks diulang tiga kali → `121212`.
3. Baris 2: `int("12") * 3` → `36`.
4. Baris 3: `"12" + str(3)` → `"12" + "3"` → `123` (menyambung teks).
5. Baris 4: `12 / 3` → `/` selalu `float` → **`4.0`**.
6. Baris 5: `12 // 5` → 2; `12 % 5` → 2; `print` dengan koma → `2 2`.

Jawaban: `121212`, `36`, `123`, `4.0`, `2 2`. Perhatikan `4.0`: hasil `/` tidak pernah `int`.

Contoh 2 — Membaca elemen list dua dimensi

Tentukan keluaran dan jawab: di indeks berapa `"kangkung"` pada `kebun` di atas? *(Kasus fiktif.)*

```
kebun = [["tomat", "cabai", "terong"],
         ["bayam", "kangkung", "sawi"],
         ["jagung", "kacang", "talas"]]
print(kebun[2][1], kebun[0][2], kebun[1][-1])
print(len(kebun[1]))
```

1. `kebun[2][1]`: baris 2 (`["jagung", "kacang", "talas"]`), kolom 1 → `kacang`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. `kebun[0][2]`: baris 0, kolom 2 → `terong`.
3. `kebun[1][-1]`: baris 1, elemen terakhir → `sawi`.
4. `len(kebun[1])`: panjang baris 1 → 3.
5. `"kangkung"` ada pada baris 1, kolom 1: `kebun[1][1]`.

Jawaban: `kacang terong sawi`, lalu 3; kangkung di `kebun[1][1]`.

Contoh 3 — Dictionary: ubah, tambah, dan cari

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
stok = {"kopi": 10, "teh": 4}
stok["teh"] = stok["teh"] + 6
stok["gula"] = 8
print(stok["teh"], len(stok), "kopi" in stok, stok.get("susu", 0))
```

1. Awal: `kopi` 10, `teh` 4.
2. `stok["teh"] = 4 + 6` → teh menjadi 10.
3. `stok["gula"] = 8` → menambah pasangan baru; sekarang tiga pasangan.
4. `stok["teh"]` → 10; `len(stok)` → 3; `"kopi" in stok` → `True`; `stok.get("susu", 0)` → `susu` tidak ada, kembalikan 0.

Jawaban: `10 3 True 0`.

Contoh 4 — Tumpukan dan antrean pada list

Tentukan keluaran dan isi akhir list. *(Kasus fiktif.)*

```
riwayat = ["garis", "kotak", "teks"]
riwayat.append("warna")
print(riwayat.pop())
print(riwayat.pop(0))
print(riwayat)
```

1. Setelah `append`: `["garis", "kotak", "teks", "warna"]`.
2. `pop()` → mengambil yang **terakhir**, `warna`; sisa `["garis", "kotak", "teks"]`.
3. `pop(0)` → mengambil yang **pertama**, `garis`; sisa `["kotak", "teks"]`.
4. `print(riwayat)` → `['kotak', 'teks']`.

Jawaban: `warna`, lalu `garis`, lalu `['kotak', 'teks']`. Tombol "Urungkan" yang membatalkan aksi terakhir memakai `pop()` (tumpukan), bukan `pop(0)`.

Contoh 5 — Memilih struktur data

Pilih struktur data untuk setiap kebutuhan dan alasannya. *(Kasus fiktif.)*

- (a) Mencari harga langsung dari kode produk pada 800 barang.
- (b) Menyimpan nilai ulangan siswa berurutan menurut tanggal, dan nilai baru terus ditambahkan.
- (c) Mencatat NIS siswa yang sudah absen hari ini tanpa boleh tercatat dua kali.
- (d) Menyimpan koordinat tetap titik awal (x, y) gim.

1. (a) Ada kunci unik dan perlu pencarian cepat → **dictionary** (kode → harga).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. (b) Urutan penting dan data bertambah → **list**.
3. (c) Tidak boleh ganda → **set**.
4. (d) Data tetap yang tidak berubah → **tuple**.

Jawaban: (a) dictionary, (b) list, (c) set, (d) tuple.

❖ PETA KONSEP

Gambar 4.6 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- Tipe dasar: **`int`, `float`, `str`, `bool`**. `"15000"` adalah teks; `input()` selalu menghasilkan teks.
- **`/` selalu `float`; `//` membagi bulat ke bawah; `%` memberi sisa.** Prioritas: `**`, lalu `*` `/` `//` `%`, lalu `+` `-`.
- **Indeks dimulai dari 0**; `-1` elemen terakhir; irisan `[a:b]` tidak menyertakan indeks `b`.
- **`append`/`pop()`** = tumpukan; **`pop(0)`** = antrean; **`in`** menguji keberadaan.
- **Dictionary**: kunci unik → nilai; `[]` pada kunci tak ada = `KeyError`, `.get()` aman.
- **List 2D**: `[baris][kolom]`; `len(peta)` jumlah baris.
- Pilih struktur menurut kebutuhan: **dictionary** (cari lewat kunci), **list** (berurutan), **set** (unik), **tuple** (tetap).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal keluaran program:

1. **Tulis tipe tiap variabel** (int, float, str) di samping barisnya.
2. **Telusuri baris demi baris** dengan tabel jejak, mencatat nilai variabel setelah tiap baris.
3. **Hitung operasi sesuai tipenya**: * pada str mengulang, + pada str menyambung, / menghasilkan float.
4. **Cek galat**: adakah operasi tipe tak cocok (str + int), indeks di luar jangkauan, atau kunci yang tidak ada?

Langkah menjawab soal memilih struktur data:

1. **Tanyakan: apa yang dilakukan dengan data itu?** Dicari lewat kunci, diurutkan, dijaga unik, atau dibuat tetap?
2. **Pilih struktur yang membuat operasi utama paling langsung**, bukan yang hanya "bisa" menyimpan.

Jebakan yang sering dipasang:

- **Teks berisi angka dihitung sebagai angka**: `"15000" * 2` menghasilkan `1500015000`, bukan `30000`.
- **`/` dikira menghasilkan bilangan bulat** (`7 / 2` adalah `3.5`; `8 / 2` adalah `4.0`).
- **`//` dikira membulatkan ke terdekat.** Ia membulatkan ke bawah.
- **Indeks dimulai dari 1** atau **irisan menyertakan batas akhir.**
- **`[baris][kolom]` dibalik** menjadi `[kolom][baris]`.
- **`pop()` dan `pop(0)` tertukar.**
- **`print(nama + nyawa)` dikira benar** padahal `TypeError`.
- **Dictionary dicari seperti list** atau `in` dikira memeriksa nilai (ia memeriksa kunci).
- **Dua list terpisah dipilih untuk data berpasangan** padahal dictionary lebih aman.

> Contoh salah tafsir: pada `kata = "Kancil"`, seorang siswa menyimpulkan `kata[1:4]` adalah `"anci"` karena "dari indeks 1 sampai 4". Batas akhir irisan **tidak ikut**, jadi `kata[1:4]` adalah indeks 1, 2, dan 3: `"anc"`.

• BEDAH SOAL PREDIKSI

Soal berikut memperlihatkan program kecil yang menghitung total dari input. Perhatikan bagaimana kamu harus menentukan **tipe** setiap nilai lebih dulu (bagian 1 dan 2), lalu operasinya, sebelum menulis keluaran.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Siswa PKL di sebuah toko alat tulis menulis program Python berikut untuk menghitung total belanja (2027).

```
harga = input("Harga satuan: ")
jumlah = 2
total = harga * jumlah
print("Total:", total)
```

Saat program dijalankan, kasir mengetik `15000` lalu menekan Enter.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program dan penyebabnya adalah ....

A. Total: 1500015000, karena teks "15000" diulang dua kali
B. Total: 30000, karena teks angka diubah otomatis menjadi int
C. Program berhenti dengan TypeError, karena str tidak dapat dikali
D. Total: 15000, karena teks dikali hanya menghasilkan teks semula
E. Total: 30000.0, karena input() membaca angka sebagai float

Pembahasan **Kunci: A**

Konsep: Fungsi `input()` selalu menghasilkan str (teks). Operator * pada str dengan int mengulang teks sebanyak bilangan itu. Agar dihitung sebagai angka, ubah dulu: `int(input(...))`.

Langkah:
1. harga = "15000" (str), jumlah = 2 (int).
2. "15000" * 2 → "15000" diulang 2 kali → "1500015000".
3. Keluaran: Total: 1500015000.

Analisis setiap opsi:
- **A.** ✓ input() menghasilkan str; str * int mengulang teks.
- **B.** ✗ Jebakan: Python tidak mengubah teks menjadi angka secara otomatis; perlu int().
- **C.** ✗ Jebakan: str + int memang TypeError, tetapi str * int sah dan mengulang teks.
- **D.** ✗ Teks yang dikali 2 diulang dua kali, bukan tetap satu kali.
- **E.** ✗ input() menghasilkan str, bukan float.

Jawaban: A.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Sebelum memilih jawaban, tulis tipe tiap nilai dan hasil operasinya sendiri.

1. Panitia lomba gim sekolah ingin membagi 7 voucher secara adil kepada 2 kelompok pemenang (2027). Programmer mencoba tiga operator pembagian Python berikut.

```
voucher = 7
kelompok = 2
print(voucher / kelompok)
print(voucher // kelompok)
print(voucher % kelompok)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran ketiga baris print secara berurutan adalah ....

A. 3.5, lalu 3.5, lalu 1
B. 3, lalu 3, lalu 1
C. 3.5, lalu 3, lalu 0.5
D. 3, lalu 3.5, lalu 1
E. 3.5, lalu 3, lalu 1

2. Programmer gim *Kancil Melesat* (2027) ingin menampilkan status karakter di layar. Variabelnya sebagai berikut.

```
nama = "Kancil"
nyawa = 3
kecepatan = 2.5
```

Ia mencoba beberapa baris perintah untuk menampilkan status.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Baris perintah yang menimbulkan error saat dijalankan adalah ....

A. `print(nama + " nyawa " + nyawa)`
B. `print(nama, "nyawa", nyawa)`
C. `print(nyawa * kecepatan)`
D. `print(nama + " nyawa " + str(nyawa))`
E. `print(f"{nama} nyawa {nyawa}")`

3. Peta labirin gim *Lorong Sunyi* (2027) disimpan dalam list dua dimensi. Tanda "#" berarti dinding, "." berarti jalan, dan "K" berarti kunci.

```
peta = [["#", ".", "#", "."],
        [".", ".", "K", "#"],
        ["#", ".", ".", "."]]
print(peta[1][2], peta[2][0])
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `. #`
B. `K .`
C. `K #`
D. `# K`
E. `. K`

4. Aplikasi kasir minimarket koperasi SMK Bina Karya (2027) mengelola 500 jenis barang. Setiap barang memiliki kode barcode yang unik, nama, dan harga. Saat barcode dipindai, aplikasi harus langsung menampilkan nama dan harga barang itu. Harga barang sesekali berubah dan harus mudah diperbarui oleh petugas.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Struktur data Python yang paling tepat untuk menyimpan data barang adalah ....

A. List berisi nama barang saja, dicari dengan perulangan dari awal
B. Dictionary dengan kunci kode barcode dan nilai nama serta harga
C. Set berisi kode barcode agar tidak ada kode yang tercatat ganda
D. Dua list terpisah untuk nama dan harga, dicocokkan dari urutannya
E. Tuple berisi seluruh harga barang, diurutkan dari yang termurah

5. Tim siswa membuat aplikasi menggambar sederhana untuk kelas desain (2027). Setiap aksi pengguna (menggambar garis, kotak, teks, mengganti warna) disimpan dalam list Python `riwayat` dengan `riwayat.append(aksi)`. Saat tombol "Urungkan" ditekan, aplikasi harus membatalkan aksi yang paling akhir dilakukan, lalu aksi sebelumnya bila tombol ditekan lagi.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Cara mengambil aksi dari list yang tepat untuk tombol "Urungkan" adalah ....

A. `riwayat.pop(0)`, mengambil aksi pertama seperti antrean
B. `riwayat.sort()`, lalu mengambil aksi teratas menurut abjad
C. `riwayat.clear()`, menghapus semua aksi lalu mengulang
D. `riwayat[0]`, membaca aksi pertama tanpa menghapusnya
E. `riwayat.pop()`, mengambil aksi terakhir seperti tumpukan

Kunci singkat: 1. E 2. A 3. C 4. B 5. E

✔ PERIKSA DIRI

- Mengapa `"15000" * 2` tidak menghasilkan 30000, dan bagaimana cara memperbaikinya?
- Mengapa dictionary lebih tepat daripada list untuk mencari harga dari kode produk pada 500 barang?
- **Yang harus kuingat:** `input()` menghasilkan teks; `/` menghasilkan float, `//` membulatkan ke bawah, `%` sisa; indeks dari 0 dan irisan tidak menyertakan batas akhir; `[baris][kolom]`; `pop()` mengambil yang terakhir; dictionary untuk pencarian lewat kunci.

Latihan interaktif di LMS

Sub bab 4.1: kerjakan latihannya dan lihat progresmu.
next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-4.1

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 4.2 · Struktur kontrol percabangan

SMKPPLG-4.2

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menganalisis alur logika struktur kontrol percabangan** dan **mengevaluasi efektivitasnya**. Contoh soal resmi Pusmendik untuk struktur kontrol berbentuk tiga pernyataan Benar/Salah tentang sebuah program yang menghitung berapa nilai lulus dalam sebuah daftar. Soal-soal percabangan di buku ini sama: kamu diberi kode `if–elif–else`, diagram alir, atau aturan bisnis (harga tiket, predikat nilai, ongkos kirim), lalu diminta **menelusuri hasilnya**, **menilai apakah kodenya sesuai aturan**, atau **menyusun kasus uji yang benar**.

Soal bab ini menyandang persentase soal sulit yang tinggi, tetapi sulitnya bukan karena konsepnya rumit. Sulitnya karena **satu tanda yang keliru** (> bukan >=), **urutan syarat yang terbalik**, atau **satu kasus batas yang terlewat** mengubah seluruh jawaban. Karena itu, kebiasaan utama yang harus kamu bangun adalah menelusuri kode dengan nilai nyata, bukan "merasa" kodenya benar.

Setelah mempelajari sub bab ini, kamu dapat:

- membaca dan menggambar diagram alir untuk sebuah program percabangan;
- menelusuri `if`, `if–else`, `elif`, dan `if` bersarang dengan nilai uji tertentu;
- menggunakan operator perbandingan dan operator logika (`and`, `or`, `not`) dengan prioritasnya;
- menemukan kesalahan urutan syarat, cabang yang tidak terjangkau, dan kondisi tumpang tindih;
- menyusun kasus uji dengan nilai batas dan menilai efektivitas sebuah desain percabangan.

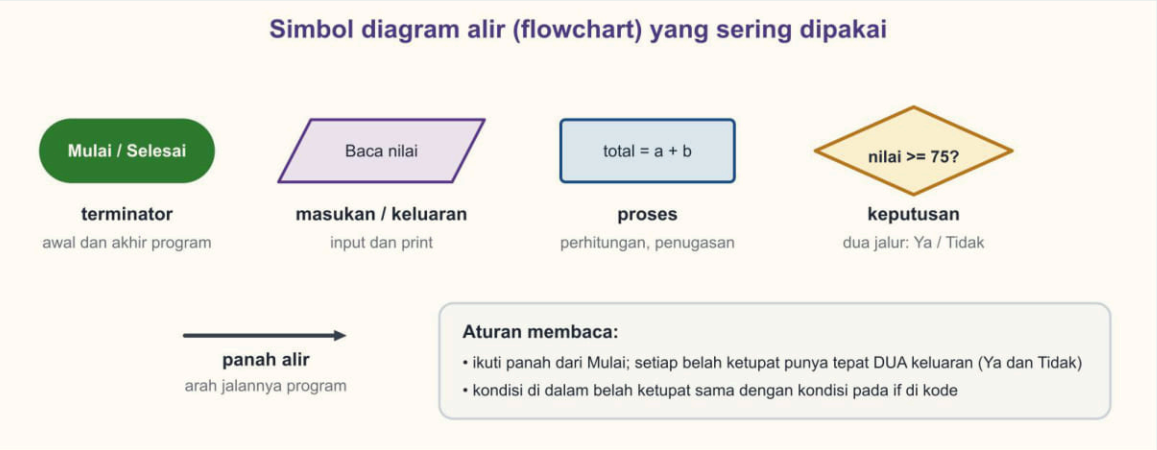
◆ KONSEP INTI

Program yang hanya menjalankan perintah dari atas ke bawah tidak berguna untuk dunia nyata, karena dunia nyata penuh keputusan: jika nilainya 75 ke atas dia lulus, jika belanja lebih dari seratus ribu dia mendapat diskon, jika umurnya di bawah lima tahun tiketnya gratis. **Percabangan** membuat program mampu memilih jalur sesuai **kondisi**. Sub bab ini mengajarkan cara menulis, membaca, dan menguji keputusan semacam itu.

1. Algoritma dan diagram alir

Algoritma adalah urutan langkah yang jelas untuk menyelesaikan suatu masalah. Salah satu cara menggambarkannya adalah **diagram alir** (*flowchart*), yang memakai simbol baku dan panah (Gambar 4.7).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 4.7 Simbol diagram alir dan aturan membacanya. Sumber: digambar oleh Tim Next Level Study Indonesia.

- **Oval** (terminator): awal dan akhir program.
- **Jajar genjang**: masukan dan keluaran (membaca nilai, mencetak hasil).
- **Persegi panjang**: proses (perhitungan, penugasan nilai).
- **Belah ketupat**: **keputusan**; di dalamnya tertulis kondisi, dan dari sana keluar **dua panah**, Ya dan Tidak.
- **Panah**: arah jalannya program.

Setiap belah ketupat pada diagram alir sama dengan sebuah `if` (atau `elif`) pada kode. Membaca diagram alir berarti **mengikuti panah dari Mulai**, menjawab setiap kondisi dengan nilai yang diuji, dan mencatat apa yang dicetak atau dihitung.

2. Struktur if, if–else, dan elif

Bentuk paling sederhana adalah **`if`**: bagian di dalamnya hanya dijalankan **bila kondisinya benar**. Bagian itu ditandai dengan **inden** (menjorok ke dalam, biasanya empat spasi), dan inden itu wajib di Python.

```
suhu = 38.5
if suhu >= 37.5:
    print("Demam")
print("Selesai")
```

Keluarannya `Demam` lalu `Selesai`. Baris `print("Selesai")` tidak menjorok, sehingga dijalankan **selalu**. Bila `suhu = 36.5`, hanya `Selesai` yang tercetak.

`if-else` menyediakan dua jalur: bila kondisi benar jalankan blok pertama, bila tidak jalankan blok `else`.

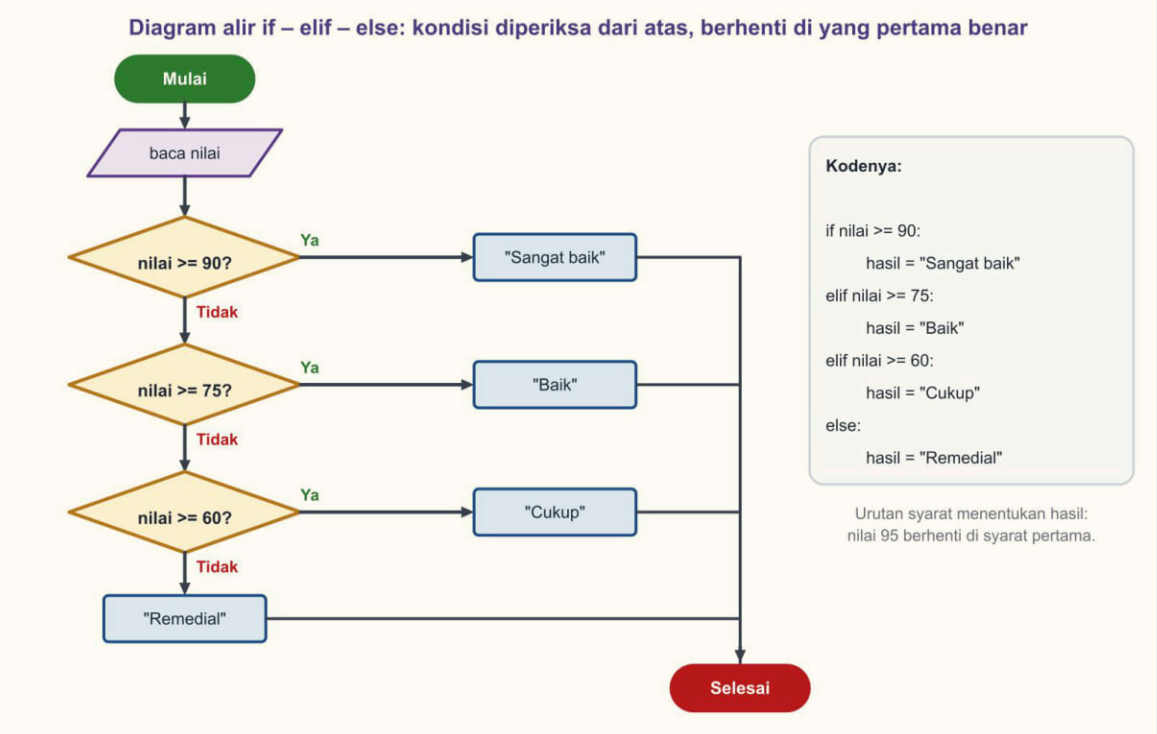
```
total = 80000
if total >= 100000:
    ongkir = 0
else:
    ongkir = 15000
print(ongkir)
```

Keluarannya `15000`, sebab 80000 kurang dari 100000.

`elif` (singkatan *else if*) dipakai untuk **lebih dari dua jalur**. Python memeriksa kondisi **dari atas**

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

ke bawah, menjalankan blok pertama yang kondisinya benar, **lalu melewati sisanya**. Bila tidak ada yang benar, blok `else` yang berjalan (bila ada). Inilah kunci seluruh sub bab ini: **pemeriksaan berhenti pada kondisi pertama yang benar**, sehingga **urutan** kondisi menentukan hasil. Gambar 4.8 memperlihatkan aliran untuk predikat nilai.



Gambar 4.8 Diagram alir if–elif–else untuk predikat nilai: pemeriksaan dari atas ke bawah. Sumber: digambar oleh Tim Next Level Study Indonesia.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Operator perbandingan dan logika

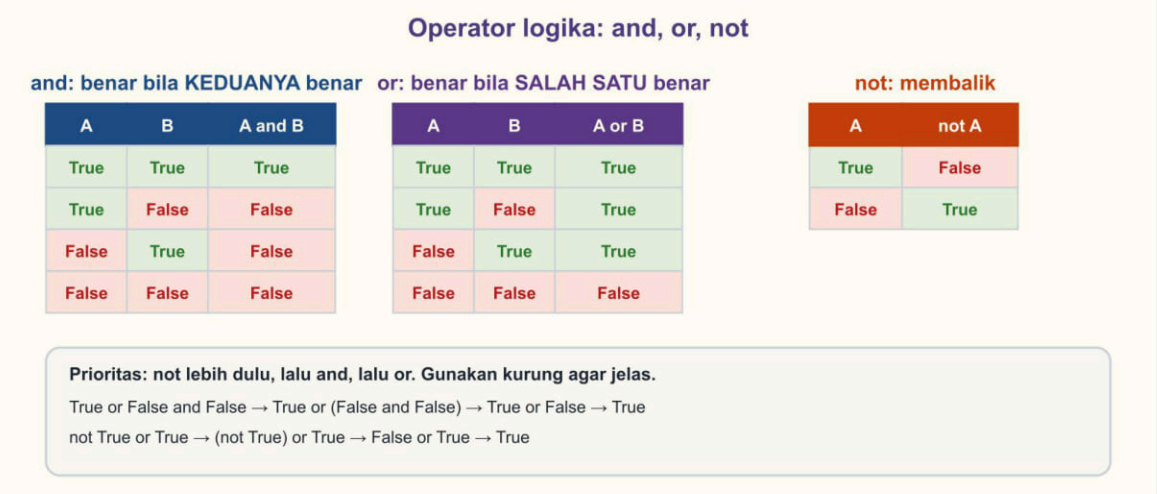
Operator perbandingan membandingkan dua nilai dan menghasilkan `True` atau `False`:

Operator	Arti
==	sama dengan
!=	tidak sama dengan
< >	kurang dari, lebih dari
<= >=	kurang dari atau sama dengan; lebih dari atau sama dengan

Hati-hati: **= menyimpan nilai, sedangkan == membandingkan.** `if x = 5:` adalah galat penulisan; yang dimaksud adalah `if x == 5:`.

Operator logika menggabungkan beberapa kondisi (tabel kebenarannya pada Gambar 4.9):

- **and**: benar bila **kedua** kondisi benar.
- **or**: benar bila **salah satu** (atau keduanya) benar.
- **not**: membalik nilai kebenaran.



Gambar 4.9 Tabel kebenaran operator and, or, dan not beserta prioritasnya. Sumber: digambar oleh Tim Next Level Study Indonesia.

Prioritas operator logika: `not` lebih dulu, lalu `and`, lalu `or`. Misalnya `True or False and False` dihitung sebagai `True or (False and False)`, yaitu `True or False`, yaitu `True`. Bila ragu, **pakai kurung**.

Kondisi bersyarat ganda menjadi lebih jelas dengan operator logika. Contoh: lulus bila nilai teori dan nilai praktik keduanya minimal 70:

```
teori = 80
praktik = 65
if teori >= 70 and praktik >= 70:
    print("Lulus")
else:
    print("Belum lulus")
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Keluarannya `Belum lulus`, sebab kondisi kedua salah sehingga `and` salah.

4. Urutan syarat pada elif menentukan hasil

Karena pemeriksaan berhenti di kondisi pertama yang benar, **kondisi yang lebih umum tidak boleh ditaruh di atas kondisi yang lebih khusus**. Perhatikan program pemberian hadiah berikut yang urutan syaratnya keliru:

```
poin = 120
if poin >= 50:
    hadiah = "stiker"
elif poin >= 100:
    hadiah = "buku"
elif poin >= 200:
    hadiah = "tas"
else:
    hadiah = "tidak ada"
print(hadiah)
```

Keluarannya `stiker`, bukan `buku`. Nilai 120 memenuhi `poin >= 50` yang diperiksa pertama, sehingga pemeriksaan berhenti di sana. Kondisi `poin >= 100` dan `poin >= 200` menjadi **cabang yang tidak terjangkau** oleh nilai mana pun yang 50 atau lebih. Perbaikannya: **urutkan dari yang paling ketat ke yang paling longgar**.

```
poin = 120
if poin >= 200:
    hadiah = "tas"
elif poin >= 100:
    hadiah = "buku"
elif poin >= 50:
    hadiah = "stiker"
else:
    hadiah = "tidak ada"
print(hadiah)
```

Keluarannya sekarang `buku`. Cara lain mencegah kesalahan urutan adalah memakai **rentang yang saling lepas**, misalnya `elif 100 <= poin < 200:`. Python mengizinkan perbandingan berantai seperti ini.

Hal yang berkaitan: **`if` terpisah berbeda dengan `elif`**. Pada tiga `if` berurutan, **semuanya diperiksa** dan beberapa bisa berjalan sekaligus. Pada rangkaian `if–elif–else`, **hanya satu** blok yang berjalan.

Kondisi tidak harus berupa angka. Teks juga dapat dibandingkan, dan operator **`in`** memeriksa apakah sebuah nilai ada di dalam kumpulan:

```
hari = "Sabtu"
if hari == "Sabtu" or hari == "Minggu":
    print("Libur")
if hari in ["Senin", "Selasa", "Rabu", "Kamis", "Jumat"]:
    print("Masuk")
```

Program itu mencetak `Libur` saja untuk `hari = "Sabtu"`. Perbandingan teks peka huruf besar dan kecil: `"sabtu" == "Sabtu"` bernilai `False`. Bila kondisi mengandung beberapa pilihan nilai, bentuk `hari in [...]` lebih ringkas daripada rangkaian `or` yang panjang.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Tabel keputusan membantu merancang kondisi yang kompleks sebelum menulis kode. Kolom kiri memuat kombinasi kondisi, kolom kanan hasilnya. Untuk diskon belanja yang bergantung pada keanggotaan dan jumlah belanja, tabel keputusannya dapat berbentuk:

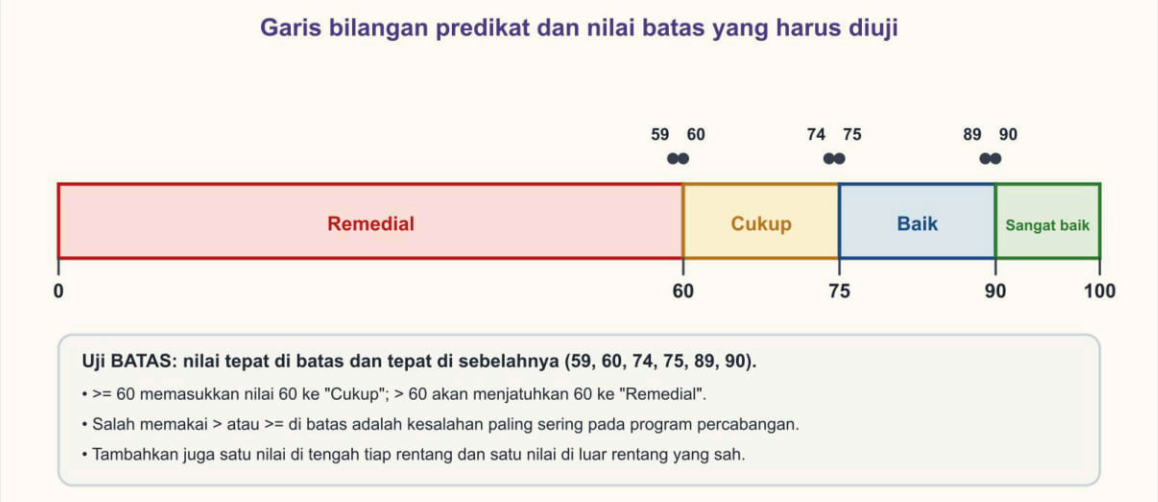
Belanja minimal Rp100.000?	Anggota?	Diskon
ya	ya	10 %
ya	tidak	5 %
tidak	ya	5 %
tidak	tidak	0 %

Dari tabel ini kamu langsung melihat apakah semua kombinasi sudah tertangani. Empat baris berarti empat kemungkinan; bila kodenya hanya menangani tiga, ada kombinasi yang terlewat.

Beberapa kesalahan penulisan yang sering membuat program berhenti dengan galat, dan kerap dipakai sebagai pengecoh pada soal: **lupa titik dua** (`:`) di akhir baris `if`, **inden tidak konsisten** (blok di dalam `if` harus menjorok sama), dan **memakai = pada kondisi**. Galat jenis ini disebut galat **sintaks** (penulisan) dan terdeteksi sebelum program berjalan. Berbeda dengan galat **logika**, yaitu program berjalan tanpa pesan galat tetapi hasilnya salah, misalnya urutan `elif` terbalik atau `>` seharusnya `>=`. Galat logika jauh lebih sulit ditemukan karena tidak ada pesan yang menunjuk lokasinya; itu sebabnya uji dengan nilai batas sangat penting.

5. Nilai batas dan kasus uji

Sebagian besar kesalahan percabangan muncul **di nilai batas**: nilai tepat pada ambang dan nilai tepat di sebelahnya. Misalnya predikat nilai dengan ambang 60, 75, 90 (Gambar 4.10).



Gambar 4.10 Predikat nilai pada garis bilangan dan nilai batas yang harus diuji. Sumber: digambar oleh Tim Next Level Study Indonesia.

Nilai 60 termasuk "Cukup" bila syaratnya `>= 60`, tetapi jatuh ke "Remedial" bila ditulis `> 60`. Karena itu **kasus uji yang baik** memuat: nilai **tepat di batas** (60, 75, 90), nilai **tepat di sebelahnya** (59, 74, 89), satu nilai **di tengah** tiap rentang, dan satu nilai **di luar jangkauan** yang sah (misalnya 0 atau 100 sebagai ujung).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Saat menilai apakah sebuah **kasus uji** sudah benar, ikuti **aturan bisnisnya** dan hitung hasilnya sendiri satu per satu. Hasil yang "seharusnya" harus cocok dengan aturan, bukan dengan tebakan.

Menelusuri dengan tabel jejak. Cara paling dapat diandalkan untuk menjawab soal percabangan adalah menulis **tabel jejak**: satu baris untuk setiap nilai uji, satu kolom untuk setiap kondisi, dan satu kolom hasil. Misalnya untuk `if nilai >= 75: ... elif nilai >= 60: ... else: ...` dengan nilai uji 59, 60, 74, 75:

nilai	`nilai >= 75`	`nilai >= 60`	hasil
59	salah	salah	else
60	salah	benar	cabang kedua
74	salah	benar	cabang kedua
75	benar	(tidak diperiksa)	cabang pertama

Perhatikan kolom terakhir untuk nilai 75: kondisi kedua **tidak diperiksa** karena kondisi pertama sudah benar. Tabel seperti ini membuat jawabanmu berdasarkan bukti, bukan dugaan.

6. if bersarang dan menyederhanakan percabangan

Sebuah `if` boleh berada di dalam `if` lain (**bersarang**). Inden menunjukkan blok mana yang berada di dalam mana.

```
umur = 4
pelajar = False
if umur < 5:
    biaya = 0
else:
    if pelajar:
        biaya = 15000
    else:
        biaya = 25000
print(biaya)
```

Keluarannya `0`. Kode di atas bisa ditulis lebih rata dengan `elif`:

```
if umur < 5:
    biaya = 0
elif pelajar:
    biaya = 15000
else:
    biaya = 25000
```

Hasilnya sama, tetapi lebih mudah dibaca. **Efektivitas** sebuah percabangan dinilai dari: (1) **hasilnya benar untuk semua kasus**, termasuk batas; (2) **tidak ada cabang yang tidak terjangkau**; (3) **tidak ada kondisi yang diulang-ulang** tanpa perlu; dan (4) **mudah dibaca dan diubah** (rata, bukan sarang berlapis).

Contoh lain tentang kondisi yang **berulang tanpa perlu**. Dua kode berikut menghasilkan hasil sama, tetapi kode pertama memeriksa `total >= 100000` dua kali:

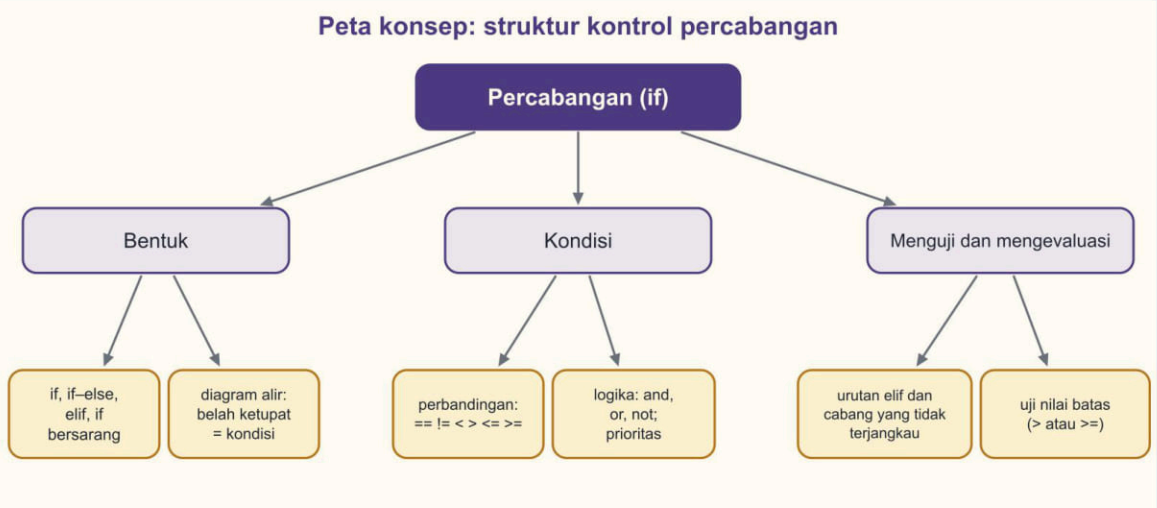
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
if total >= 100000 and anggota:
    diskon = 0.10
if total >= 100000 and not anggota:
    diskon = 0.05
```

Kode kedua lebih efektif karena kondisi yang sama dipecah sekali dan cabangnya jelas:

```
if total >= 100000:
    if anggota:
        diskon = 0.10
    else:
        diskon = 0.05
```

Sebelum menulis kode, banyak programmer menuliskan langkahnya dulu dalam bahasa sehari-hari yang terstruktur (**pseudocode**), misalnya "jika belanja minimal seratus ribu: jika anggota, diskon sepuluh persen, jika bukan, diskon lima persen". Pseudocode mudah dicocokkan dengan diagram alir dan dengan kode, sehingga kesalahan logika ketahuan lebih awal.



Gambar 4.11 Peta konsep struktur kontrol percabangan. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menelusuri tarif parkir dengan lima nilai uji

Tentukan keluaran program tarif parkir berikut. *(Kasus fiktif.)*

```
def tarif_parkir(jam, member):
    if jam <= 1:
        biaya = 2000
    elif jam <= 3:
        biaya = 5000
    else:
        biaya = 8000
    if member:
        biaya = biaya - 1000
    return biaya

for j, m in [(1, False), (2, True), (3, False), (4, True), (0.5, True)]:
    print(j, m, tarif_parkir(j, m))
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. Perhatikan struktur: rangkaian `if–elif–else` pertama menentukan biaya dasar; `if member` **terpisah** sesudahnya, sehingga dijalankan pula.
2. `(1, False)`: `jam <= 1` benar → 2000; member salah → **2000**.
3. `(2, True)`: `2 <= 1` salah; `2 <= 3` benar → 5000; member benar → 5000 − 1000 = **4000**.
4. `(3, False)`: `3 <= 3` benar (batas!) → 5000; tanpa potongan → **5000**.
5. `(4, True)`: 4 lebih dari 3 → `else` → 8000; member → **7000**.
6. `(0.5, True)`: `0.5 <= 1` benar → 2000; member → **1000**.

Jawaban: `1 False 2000`, `2 True 4000`, `3 False 5000`, `4 True 7000`, `0.5 True 1000`. Perhatikan bahwa nilai batas `1` dan `3` jatuh ke rentang yang menggunakan `<=`.

Contoh 2 — Memperbaiki urutan elif

Sebuah toko memberi gratis ongkir bila belanja minimal Rp300.000, diskon Rp20.000 bila belanja minimal Rp150.000, dan diskon Rp5.000 bila belanja minimal Rp50.000. Kode di bawah dianggap sesuai oleh pembuatnya. Periksa untuk belanja Rp200.000 dan perbaiki. *(Kasus fiktif.)*

```
belanja = 200000
if belanja >= 50000:
    hadiah = "diskon 5 ribu"
elif belanja >= 150000:
    hadiah = "diskon 20 ribu"
elif belanja >= 300000:
    hadiah = "gratis ongkir"
else:
    hadiah = "tidak ada"
print(hadiah)
```

1. Telusuri: 200000 ≥ 50000 benar pada syarat **pertama** → `diskon 5 ribu`; pemeriksaan berhenti.
2. Seharusnya belanja Rp200.000 memenuhi syarat Rp150.000 → "diskon 20 ribu". Kodenya **salah**.
3. Penyebab: syarat umum (≥ 50000) ada di atas syarat khusus. Syarat `>= 150000` dan `>= 300000` **tidak terjangkau**.
4. Perbaikan: urutkan dari yang paling ketat: `>= 300000`, lalu `>= 150000`, lalu `>= 50000`.
5. Uji ulang untuk belanja 300000 (batas) → "gratis ongkir"; 150000 → "diskon 20 ribu"; 149999 → "diskon 5 ribu"; 49999 → "tidak ada".

Jawaban: dengan urutan yang benar, Rp200.000 menghasilkan "diskon 20 ribu".

Contoh 3 — Dari diagram alir ke kode

Sebuah diagram alir: **Mulai → baca `suhu` → apakah `suhu > 100`? Ya: cetak "uap" → Selesai. Tidak: apakah `suhu >= 0`? Ya: cetak "cair" → Selesai. Tidak: cetak "beku" → Selesai.** Tulis kodenya, lalu tentukan keluaran untuk suhu 100, 0, dan -5. *(Kasus fiktif.)*

1. Tiap belah ketupat menjadi satu kondisi, dari atas ke bawah.
2. Kode:

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
suhu = 100
if suhu > 100:
    print("uap")
elif suhu >= 0:
    print("cair")
else:
    print("beku")
```

1. Suhu 100: `100 > 100` salah (batas; bukan lebih dari 100); `100 >= 0` benar → **cair**.
2. Suhu 0: `0 > 100` salah; `0 >= 0` benar → **cair**.
3. Suhu -5: `-5 > 100` salah; `-5 >= 0` salah; `else` → **beku**.

Jawaban: `cair`, `cair`, `beku`. Diagram alir ini memakai > pada ambang pertama, jadi suhu tepat 100 jatuh ke "cair". Bila kamu mengubahnya menjadi >=, hasil untuk suhu 100 berubah menjadi "uap".

Contoh 4 — Operator logika dalam satu rangkaian

Toko buku memberi diskon: 15 % bila belanja minimal Rp300.000; 10 % bila belanja minimal Rp100.000 **dan** anggota; 5 % bila belanja minimal Rp100.000 **atau** anggota; selain itu 0. Tentukan keluaran. *(Kasus fiktif.)*

```
def diskon(total, anggota):
    if total >= 300000:
        return 0.15
    elif total >= 100000 and anggota:
        return 0.10
    elif total >= 100000 or anggota:
        return 0.05
    else:
        return 0

for t, a in [(350000, False), (120000, True), (120000, False), (50000, True), (50000,
False)]:
    print(t, a, diskon(t, a))
```

1. `(350000, False)`: ≥ 300000 → **0.15**.
2. `(120000, True)`: tidak ≥ 300000; ≥ 100000 **dan** anggota → **0.1**.
3. `(120000, False)`: syarat kedua gagal (bukan anggota); syarat ketiga `total >= 100000 or anggota`: 120000 ≥ 100000 benar → **0.05**.
4. `(50000, True)`: syarat 1 dan 2 gagal; syarat 3: `False or True` → benar → **0.05**.
5. `(50000, False)`: semua gagal → `else` → **0**.

Jawaban: `0.15`, `0.1`, `0.05`, `0.05`, `0`. Urutan syarat sengaja dari yang paling ketat ke paling longgar; bila syarat `or` ditaruh paling atas, semua orang yang memenuhinya akan berhenti di sana.

Contoh 5 — Menyusun kasus uji dari aturan

Perpustakaan sekolah (2027) menetapkan denda keterlambatan: tidak terlambat tidak ada denda; terlambat 1–3 hari Rp500 per hari; 4–7 hari Rp1.000 per hari; lebih dari 7 hari denda tetap Rp10.000. Tentukan hasil seharusnya untuk keterlambatan 0, 1, 3, 4, 7, dan 8 hari, lalu periksa kodenya. *(Kasus fiktif.)*

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
def denda(hari):
    if hari <= 0:
        return 0
    elif hari <= 3:
        return hari * 500
    elif hari <= 7:
        return hari * 1000
    else:
        return 10000

for h in [0, 1, 3, 4, 7, 8]:
    print(h, denda(h))
```

1. Hitung dulu dari **aturan**, bukan dari kode: 0 hari → 0; 1 hari → 500; 3 hari → 1.500; 4 hari → 4.000; 7 hari → 7.000; 8 hari → 10.000.
2. Telusuri kode: `0 <= 0` benar → 0 ✓; 1 dan 3 masuk `hari <= 3` → 500 dan 1.500 ✓; 4 dan 7 masuk `hari <= 7` → 4.000 dan 7.000 ✓; 8 masuk `else` → 10.000 ✓.
3. Semua nilai batas (0, 1, 3, 4, 7, 8) cocok dengan aturan, jadi kodenya sesuai.

Jawaban: `0 0`, `1 500`, `3 1500`, `4 4000`, `7 7000`, `8 10000`. Perhatikan bahwa hasil seharusnya dihitung dari aturan lebih dulu, kemudian dibandingkan dengan kode, supaya kesalahan kode tidak "tersembunyi".

❖ PETA KONSEP

Gambar 4.11 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **`if/elif/else`** diperiksa **dari atas ke bawah** dan **berhenti di kondisi pertama yang benar**.
- **Urutan syarat**: dari yang paling ketat (khusus) ke yang paling longgar (umum); bila terbalik, cabang bawah tidak terjangkau.
- **`==` membandingkan; `=` menyimpan.** `and` butuh keduanya benar, `or` cukup satu; prioritas **not, and, or**.
- **Uji nilai batas**: tepat di ambang dan di sebelahnya; `>=` dan `>` berbeda tepat di ambang.
- Tiap **belah ketupat** pada diagram alir adalah satu kondisi.
- Percabangan yang efektif: benar untuk semua kasus, tanpa cabang tak terjangkau, tidak berulang, dan mudah dibaca.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal percabangan:

1. **Tulis aturan bisnis** sebagai rentang di garis bilangan, termasuk nilai batas.
2. **Telusuri kode dengan nilai konkret**, mulai dari nilai batas, bukan nilai tengah saja.
3. **Pada `elif`, berhenti di kondisi pertama yang benar**; coret cabang sisanya.
4. **Untuk menilai urutan syarat**, tanyakan: adakah nilai yang berhenti di cabang yang lebih umum padahal seharusnya masuk cabang yang lebih khusus?
5. **Untuk kasus uji**, hitung hasil seharusnya dari aturan bisnis lebih dulu, baru bandingkan dengan pilihan.

Jebakan yang sering dipasang:

- **Urutan `elif` dari longgar ke ketat**, sehingga cabang bawah tidak terjangkau.
- **`>` dan `>=` tertukar di nilai batas.**
- **`if` terpisah dikira `elif`** (dua blok sama-sama berjalan).
- **`and` dan `or` tertukar**, terutama dengan prioritas operator.
- **`=` dipakai di tempat `==`.**
- **Menghitung hanya satu nilai uji.** Satu nilai yang kebetulan benar tidak membuktikan kodenya benar.
- **Mengganti `else` dengan `elif` tanpa kondisi yang menutup semua kasus**, sehingga ada nilai yang tidak diproses.
- **Kasus uji dihitung dari kode**, bukan dari aturan bisnis, sehingga kesalahan kode "terbukti benar".

> Contoh salah tafsir: pada `if belanja >= 100000 or anggota:`, seorang siswa menyimpulkan "anggota dengan belanja Rp10.000 tidak mendapat diskon karena belanjanya kecil". Dengan `or`, cukup salah satu kondisi benar; anggota memenuhinya, sehingga diskon tetap diberikan.

● BEDAH SOAL PREDIKSI

Soal berikut menyajikan aturan predikat nilai dan sebuah fungsi dengan urutan syarat yang bermasalah. Perhatikan bagaimana kamu menguji dengan nilai batas dan nilai yang seharusnya masuk cabang paling ketat (bagian 4 dan 5).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Aplikasi penilaian praktik di SMK Teknologi (2027) seharusnya memberi predikat: nilai 90 ke atas "Sangat baik", 75–89 "Baik", 60–74 "Cukup", dan di bawah 60 "Remedial". Programmer menulis fungsi berikut.

```
def predikat(nilai):
    if nilai >= 60:
        return "Cukup"
    elif nilai >= 75:
        return "Baik"
    elif nilai >= 90:
        return "Sangat baik"
    else:
        return "Remedial"
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang fungsi tersebut adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `predikat(95)` menghasilkan "Cukup".

☐ B. `predikat(75)` menghasilkan "Baik" sesuai rencana predikat.

☐ C. `predikat(59)` menghasilkan "Cukup" seperti nilai 60.

☐ D. Agar sesuai rencana, urutan syarat diubah menjadi >= 90, lalu >= 75, lalu >= 60.

☐ E. Kesalahan diperbaiki dengan mengganti `else` menjadi `elif nilai < 60`.

Pembahasan **Kunci: A, D**

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pada if–elif–else, syarat diperiksa berurutan dari atas dan hanya cabang PERTAMA yang benar yang dijalankan. Pada pemeriksaan batas bawah (>=), syarat yang paling ketat harus diletakkan paling atas.

Langkah:

1. Setiap nilai ≥ 60 langsung masuk cabang pertama ("Cukup").
2. Cabang "Baik" dan "Sangat baik" tidak pernah tercapai.
3. Nilai < 60 tetap masuk else ("Remedial").

Analisis setiap pernyataan:

• **Pernyataan 1 ✓** 95 >= 60 benar pada syarat pertama, sehingga langsung mengembalikan "Cukup".

• **Pernyataan 2 ✗** Jebakan: 75 >= 60 sudah benar di syarat pertama, jadi hasilnya "Cukup"; elif nilai >= 75 tidak pernah diperiksa.

• **Pernyataan 3 ✗** 59 >= 60 salah, 59 >= 75 salah, 59 >= 90 salah → else → "Remedial".

• **Pernyataan 4 ✓** Syarat yang paling ketat diperiksa lebih dulu sehingga setiap rentang mendapat predikat yang benar.

• **Pernyataan 5 ✗** Bagian else sudah benar untuk nilai di bawah 60; masalahnya ada pada urutan syarat di atasnya.

Jawaban: pernyataan 1, 4.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

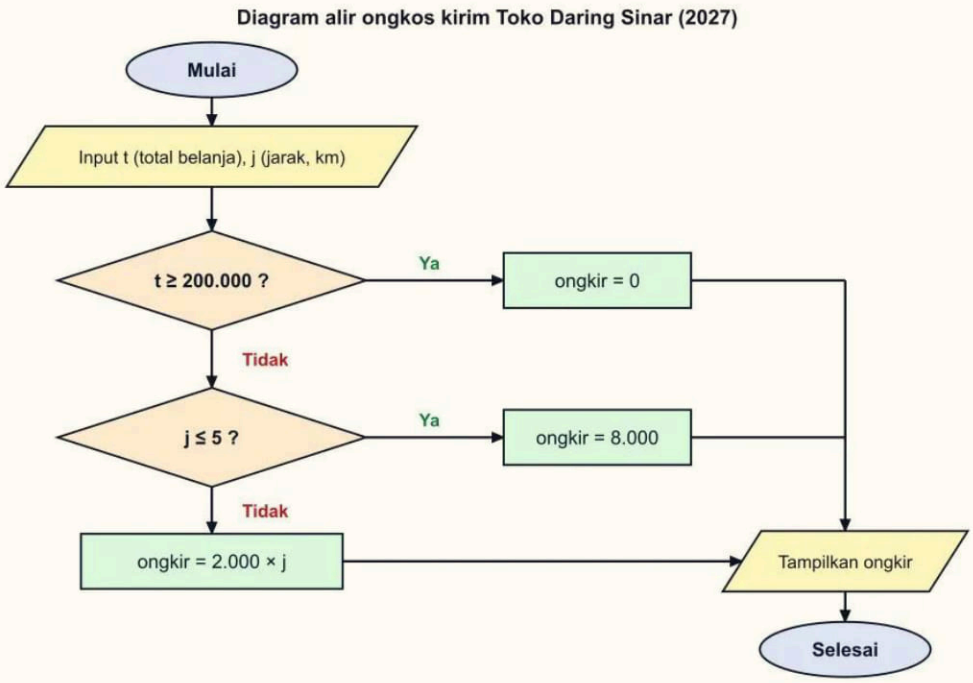
Kerjakan tanpa melihat kunci. Untuk setiap soal, gambar garis bilangan aturannya atau telusuri diagram alirnya dengan nilai batas.

1. Toko Daring Sinar menghitung ongkos kirim (ongkir) setiap pesanan dengan diagram alir pada gambar. Pada suatu hari di tahun 2027 masuk tiga pesanan berikut.

- Pesanan P1: total belanja Rp180.000, jarak 8 km
- Pesanan P2: total belanja Rp200.000, jarak 12 km
- Pesanan P3: total belanja Rp150.000, jarak 5 km

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Jumlah ongkir ketiga pesanan tersebut adalah ....



A. Rp26.000
B. Rp48.000
C. Rp24.000
D. Rp16.000
E. Rp50.000

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Taman edukasi sains Galaksi (2027) menetapkan harga tiket: pengunjung berumur di bawah 5 tahun gratis; pengunjung lain yang menunjukkan kartu pelajar membayar Rp15.000; selain itu Rp25.000. Pada hari Senin, setiap tiket BERBAYAR mendapat potongan Rp5.000. Sebelum aplikasi loket dibuat, programmer menyusun beberapa kasus uji beserta hasil yang seharusnya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kasus uji yang hasil seharusnya sudah benar adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Anak 4 tahun datang hari Senin membayar Rp0.
☐ B. Pelajar 16 tahun dengan kartu pelajar datang hari Senin membayar Rp15.000.
☐ C. Pengunjung dewasa tanpa kartu pelajar datang hari Selasa membayar Rp20.000.
☐ D. Anak tepat 5 tahun tanpa kartu pelajar datang hari Minggu masuk gratis.
☐ E. Pengunjung dewasa tanpa kartu pelajar datang hari Senin membayar Rp20.000.

Kunci singkat: 1. C 2. A, E

✔ PERIKSA DIRI

- Mengapa pada `if-elif-else` urutan syarat dapat mengubah hasil, dan bagaimana cara menyusunnya?
- Mengapa kasus uji harus memuat nilai tepat di batas (misalnya 60, 75, 90) dan bukan hanya nilai tengah?
- **Yang harus kuingat:** pemeriksaan berhenti di kondisi pertama yang benar; urutkan dari paling ketat; uji nilai batas; `and` butuh keduanya, `or` cukup satu; `==` membandingkan.



Latihan interaktif di LMS

Sub bab 4.2: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-4.2

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 4.3 · Struktur kontrol perulangan

SMKPPLG-4.3

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menganalisis alur logika struktur kontrol perulangan** dan **mengevaluasi efektivitasnya**. Pada soal, kamu akan diberi kode Python yang memakai `for` atau `while`, kadang bersarang atau memakai `break` dan `continue`, lalu diminta **menentukan keluarannya**, **menghitung berapa kali isi perulangan dijalankan**, atau **menilai pernyataan** tentang program itu (Benar/Salah atau pilih lebih dari satu).

Ini salah satu bagian paling sulit pada bab pemrograman: pada paket latihan buku ini, soal perulangan **semuanya** berkategori sulit. Alasannya sederhana. Menelusuri perulangan yang berputar belasan kali membutuhkan **ketelitian pencatatan**, dan satu kesalahan kecil (angka akhir `range` yang ikut dihitung, nilai awal yang keliru, atau lupa kapan nilai diperbarui) mengubah seluruh jawaban. Penyelamat utamamu adalah **tabel jejak**: satu baris untuk setiap putaran.

Setelah mempelajari sub bab ini, kamu dapat:

- menjelaskan cara kerja `for` dengan `range(mulai, akhir, langkah)` dan menentukan nilai serta banyaknya putaran;
- menjelaskan cara kerja `while`, syarat berhentinya, dan bahaya perulangan tak berhingga;
- memakai penghitung dan akumulator, serta `break` dan `continue`;
- menelusuri perulangan bersarang dan menghitung jumlah putarannya;
- menelusuri program dengan tabel jejak dan menemukan kesalahan seperti selisih satu (*off-by-one*).

◆ KONSEP INTI

Banyak pekerjaan komputer berupa **mengulang**: menjumlahkan seribu nilai, memeriksa setiap musuh dalam gim, menampilkan setiap barang di daftar, atau menghitung mundur sebelum roket meluncur. Menulis perintah yang sama ribuan kali tidak mungkin. **Perulangan** membuat sekumpulan perintah dijalankan berkali-kali dengan menuliskannya sekali. Python memiliki dua perulangan: **`for`** dan **`while`**.

1. Perulangan for dan fungsi range

`for` dipakai ketika **jumlah putaran sudah diketahui** atau ketika kamu mengolah **setiap elemen** dari sebuah kumpulan. Ia mengambil elemen satu per satu dan menjalankan blok yang menjorok untuk masing-masing.

```
for nama in ["Ayu", "Budi", "Cici"]:
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
    print("Halo", nama)
```

Keluarannya tiga baris: `Halo Ayu`, `Halo Budi`, `Halo Cici`. Untuk mengulang sekian kali dengan angka, dipakai fungsi **`range`**, yang menghasilkan deretan bilangan.

- **`range(akhir)`**: dari 0 sampai `akhir - 1`. `range(5)` menghasilkan `0, 1, 2, 3, 4` (lima angka).
- **`range(mulai, akhir)`**: dari `mulai` sampai `akhir - 1`. `range(2, 6)` menghasilkan `2, 3, 4, 5`.
- **`range(mulai, akhir, langkah)`**: loncat sebesar `langkah`. `range(2, 11, 3)` menghasilkan `2, 5, 8`. Langkah **negatif** menghitung mundur: `range(10, 0, -3)` menghasilkan `10, 7, 4, 1`.

Aturan terpenting (Gambar 4.12): **angka `akhir` tidak pernah ikut.** Perulangan berhenti **sebelum** mencapainya. Itulah sebabnya `range(10, 0, -3)` berakhir di 1, bukan 0, dan `range(1, 5)` hanya menghasilkan 1 sampai 4.



Gambar 4.12 Tiga contoh range pada garis bilangan; lingkaran putus-putus adalah batas "akhir" yang tidak ikut. Sumber: digambar oleh Tim Next Level Study Indonesia.

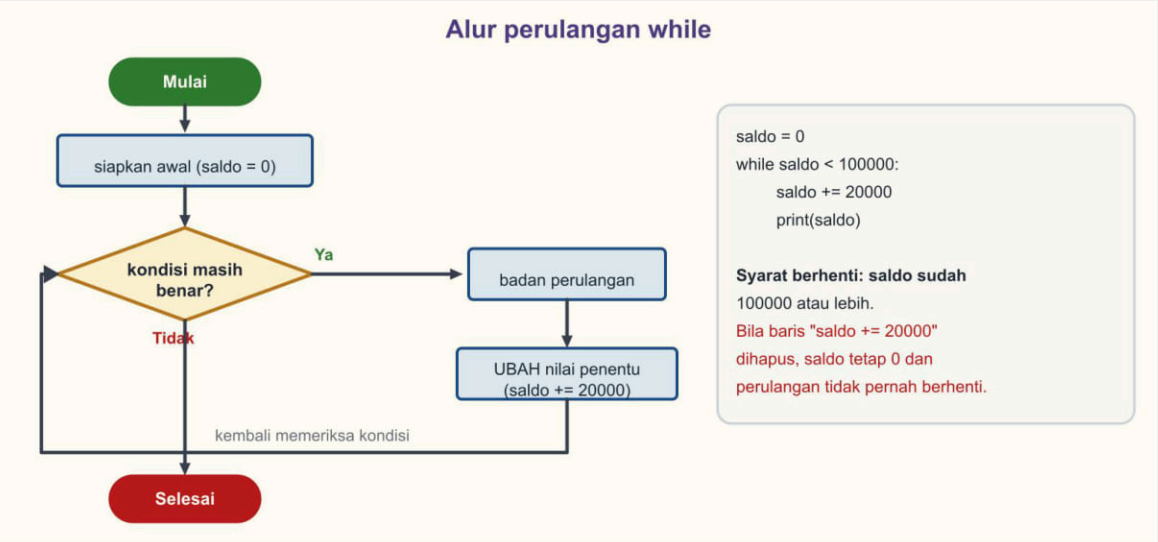
```
for i in range(10, 0, -3):
    print(i, end=" ")
print()
print("selesai")
```

Program itu mencetak `10 7 4 1 ` pada satu baris (karena `end=" "` mengganti pindah baris dengan spasi), lalu pindah baris oleh `print()` kosong, lalu `selesai`. **Banyak putaran** adalah banyak angka yang dihasilkan `range`, yaitu 4. Cara cepat menghitungnya: selisih `mulai` dan `akhir`, dibagi besar langkah, dibulatkan ke atas ($(10 - 0) \div 3 = 3{,}33 \rightarrow 4$).

2. Perulangan while

`while` dipakai ketika **jumlah putaran tidak diketahui di awal**, tetapi ada **syarat** untuk terus berjalan. Selama syaratnya benar, blok diulang. Alurnya ditunjukkan pada Gambar 4.13: syarat diperiksa **sebelum** setiap putaran, dan bila salah, perulangan selesai.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 4.13 Alur perulangan while: kondisi, badan, pembaruan nilai, lalu kembali memeriksa kondisi. Sumber: digambar oleh Tim Next Level Study Indonesia.

```
saldo = 0
minggu = 0
while saldo < 70000:
    minggu += 1
    saldo += 20000
print(minggu, saldo)
```

Program itu mencatat berapa minggu menabung Rp20.000 sampai saldo mencapai paling sedikit Rp70.000. Keluarannya `4 80000`: setelah minggu ke-3 saldo 60000 (masih kurang), setelah minggu ke-4 saldo 80000 (tercapai).

Tiga bagian `while` yang selalu harus kamu periksa:

1. **Nilai awal** variabel yang dipakai dalam syarat (di atas, `saldo = 0`).
2. **Syarat** (`saldo < 70000`).
3. **Pembaruan** yang suatu saat membuat syarat menjadi salah (`saldo += 20000`).

Bila **pembaruan** dihapus atau tidak mengarah ke syarat salah, perulangan **tidak pernah berhenti**. Itu disebut **perulangan tak berhingga** (*infinite loop*). Contohnya `while saldo < 70000:` tanpa baris `saldo += 20000` akan berjalan selamanya karena `saldo` tetap 0.

3. Penghitung dan akumulator

Dua pola yang hampir selalu muncul dalam perulangan:

- **Penghitung (counter)**: variabel yang bertambah 1 tiap kejadian, misalnya `hitung += 1`.
- **Akumulator**: variabel yang mengumpulkan jumlah, misalnya `total += nilai`.

Penulisan `x += 5` berarti `x = x + 5`. **Letak** baris itu penting: bila `total += nilai` berada **di dalam** perulangan, ia dijalankan tiap putaran; bila berada **di luar** (tidak menjorok), ia dijalankan sekali saja setelah perulangan.

```
nilai = [80, 65, 90, 75]
i = 0
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
jumlah = 0
while i < len(nilai):
    jumlah += nilai[i]
    i += 1
print(jumlah, jumlah / len(nilai))
```

Keluarannya `310 77.5`: jumlah keempat nilai 310, rata-ratanya 310 ÷ 4 = 77,5 (hasil `/` selalu `float`). Perhatikan `i += 1` yang menjadi pembaruan: tanpa itu, program berputar tanpa henti pada `nilai[0]`.

4. break dan continue

Dua perintah mengatur alur dari dalam perulangan.

- **`continue`: lewati sisa isi putaran ini** dan langsung lanjut ke putaran berikutnya.
- **`break`: hentikan seluruh perulangan** saat itu juga.

```
angka = [5, 8, 3, 12, 6, 9]
genap_total = 0
for a in angka:
    if a % 2 == 1:
        continue
    genap_total += a
    if genap_total > 15:
        break
print(genap_total)
```

Telusuri dengan tabel jejak (Gambar 4.14). Angka ganjil dilewati (`continue`), angka genap dijumlahkan, dan begitu jumlahnya lebih dari 15, perulangan berhenti. Keluarannya `20`.

Tabel jejak: satu baris untuk setiap putaran perulangan

```
angka = [5, 8, 3, 12, 6, 9]
genap_total = 0
for a in angka:
    if a % 2 == 1:
        continue
    genap_total += a
    if genap_total > 15:
        break
print(genap_total)
```

a	a % 2 == 1?	genap_total	> 15?	tindakan
5	ya	**0**	–	continue
8	tidak	**8**	tidak	lanjut
3	ya	**8**	–	continue
12	**tidak**	**20**	**ya**	**break**
6				tak dijalankan
9				tak dijalankan

continue: lewati sisa isi putaran ini dan lanjut ke elemen berikutnya. break: hentikan seluruh perulangan.

Keluaran: print(genap_total) dijalankan sekali setelah perulangan berhenti → 20.

Gambar 4.14 Tabel jejak untuk program dengan continue dan break. Sumber: digambar oleh Tim Next Level Study Indonesia.

Bedanya, `continue` hanya melompati **satu** putaran, sedangkan `break` **mengakhiri seluruh** perulangan: angka 6 dan 9 pada contoh tidak pernah diproses karena `break` terjadi pada angka 12.

5. Perulangan bersarang

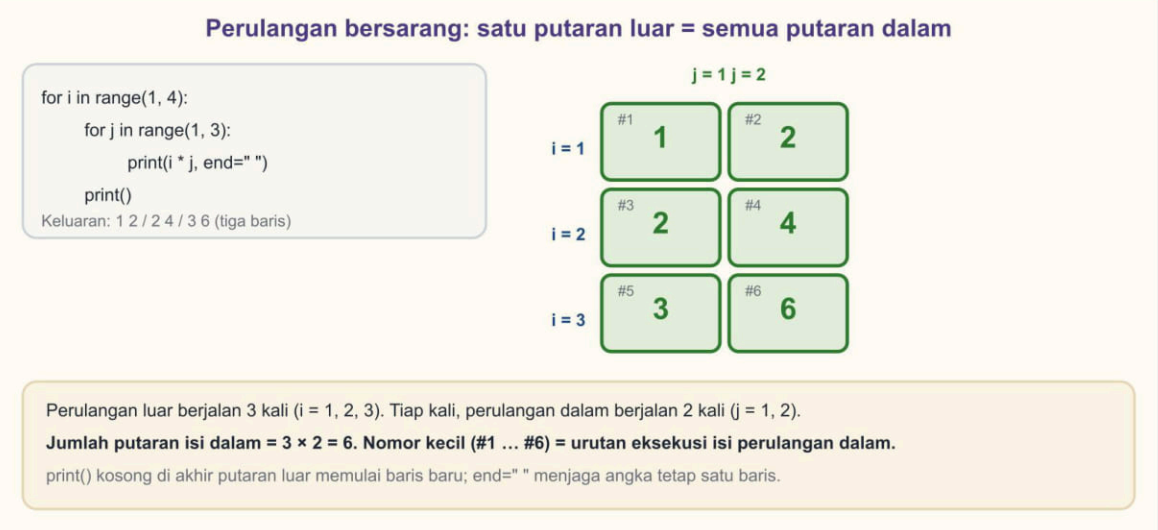
Sebuah perulangan boleh berada di dalam perulangan lain. Pada **perulangan bersarang**, **setiap putaran perulangan luar menjalankan seluruh putaran perulangan dalam** (Gambar 4.15). Jumlah putaran isi paling dalam adalah **hasil kali** jumlah putaran kedua

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

perulangan.

```
for i in range(1, 4):
    for j in range(1, 3):
        print(i * j, end=" ")
    print()
```

Perulangan luar berjalan 3 kali (`i` = 1, 2, 3) dan perulangan dalam 2 kali (`j` = 1, 2) setiap kali. Isi dalam berjalan 3 × 2 = 6 kali. Keluarannya tiga baris: `1 2`, `2 4`, `3 6`. `print()` kosong yang menjorok pada tingkat perulangan luar dijalankan **sekali per putaran luar** untuk pindah baris.



Gambar 4.15 Perulangan bersarang: tiga putaran luar, masing-masing dengan dua putaran dalam. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perulangan bersarang cocok untuk **kisi dua dimensi** (baris dan kolom), misalnya mencetak papan permainan atau memeriksa setiap sel peta. Variabel perulangan dalam sering memakai nilai variabel perulangan luar, seperti `range(baris)`; pada kasus itu jumlah putaran dalam berubah tiap putaran luar dan perlu dihitung satu per satu, bukan dikalikan begitu saja.

6. Menelusuri dengan tabel jejak dan kesalahan umum

Cara paling andal menelusuri perulangan adalah **tabel jejak**: kolom untuk setiap variabel penting, satu baris untuk tiap putaran, dan satu kolom untuk hasil pemeriksaan kondisi. Tuliskan **nilai awal** lebih dulu. Lalu telusuri satu putaran, perbarui kolom, dan jangan membuat lompatan.

Kesalahan yang sering muncul pada soal:

- **Selisih satu (*off-by-one*)**. `range(1, 5)` menghasilkan 1 sampai 4, bukan 5. Untuk menjumlahkan 1 sampai 5 kamu perlu `range(1, 6)`.
- **Angka akhir `range` dihitung ikut**, terutama pada langkah negatif.
- **Letak pembaruan atau akumulator salah** (di dalam atau di luar perulangan).
- **Nilai awal keliru**, misalnya penghitung dimulai dari 1 padahal seharusnya 0.
- **Perulangan tak berhingga** karena pembaruan terlupa atau kondisinya tidak pernah

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

menjadi salah.

- **`break` dan `continue` tertukar.**
- **Syarat `while` memakai `<` padahal seharusnya `<=`** (nilai batas).

7. Pola perulangan yang sering dipakai

Beberapa pola ini muncul berulang kali pada program nyata dan soal.

Mengolah setiap elemen list dengan indeks. Selain `for nilai in daftar`, kamu dapat memakai indeks lewat `range(len(daftar))`. Cara ini perlu bila kamu butuh nomor urutnya:

```
menu = ["nasi", "mie", "soto"]
for i in range(len(menu)):
    print(i + 1, menu[i])
```

Keluarannya `1 nasi`, `2 mie`, `3 soto`. Perhatikan `len(menu)` bernilai 3 sehingga `range(3)` menghasilkan indeks 0, 1, 2, dan `i + 1` dipakai agar nomor tampilannya dimulai dari 1.

Mencari nilai terbesar. Simpan nilai terbesar sementara, lalu perbarui bila ada yang lebih besar:

```
skor = [40, 85, 62, 91, 77]
terbesar = skor[0]
for s in skor:
    if s > terbesar:
        terbesar = s
print(terbesar)
```

Keluarannya `91`. Awalnya `terbesar` diisi elemen pertama, bukan 0, supaya benar juga bila semua nilai negatif.

Meminta masukan sampai benar. `while True` berulang tanpa batas, dan `break` menghentikannya ketika syarat terpenuhi. Pola ini dipakai untuk memvalidasi masukan:

```
# tidak dijalankan (menunggu ketikan pengguna)
while True:
    umur = int(input("Umur: "))
    if umur >= 0:
        break
    print("Umur tidak boleh negatif")
```

Program terus bertanya sampai pengguna mengetik umur yang tidak negatif.

Mengubah for menjadi while dan sebaliknya. `for n in range(1, 4): print(n)` setara dengan `n = 1`, `while n < 4: print(n)`, `n += 1`. Dari sini terlihat bahwa `for` mengerjakan tiga bagian sekaligus: nilai awal, syarat, dan pembaruan. Itulah sebabnya perulangan `for` lebih jarang tak berhingga.

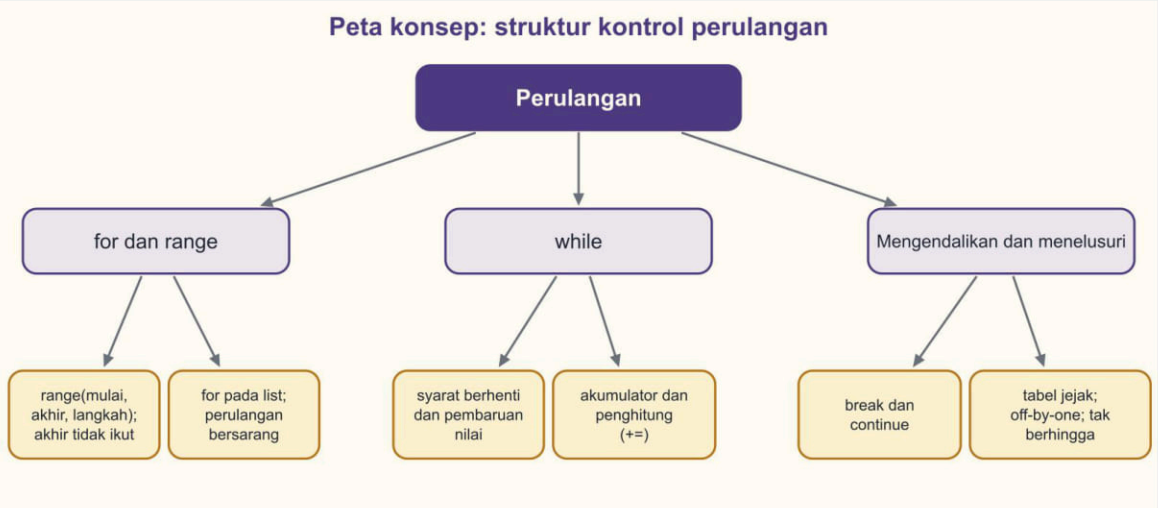
Efektivitas. Perulangan yang baik tidak mengerjakan hal yang sama berulang-ulang tanpa perlu. Perhitungan yang hasilnya tidak berubah (misalnya `len(daftar)` pada daftar yang tidak berubah) sebaiknya dihitung sekali di luar perulangan. Dan bila yang dicari sudah ditemukan, `break` menghentikan pemeriksaan sisa elemen sehingga program tidak membuang waktu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

8. Memilih for atau while

- **`for`** bila jumlah putaran jelas (mengulang n kali, mengolah setiap elemen daftar, melewati rentang angka).
- **`while`** bila yang jelas adalah **syarat berhenti**, bukan jumlah putaran (mengulang sampai saldo cukup, sampai pengguna memasukkan nilai yang benar, atau sampai gim berakhir).

Keduanya dapat saling menggantikan, tetapi pilihan yang tepat membuat kode lebih pendek dan lebih sulit salah. Perulangan `for` mengurus penghitung sendiri, sedangkan `while` mengharuskan kamu menulis nilai awal dan pembaruan sendiri, sehingga lebih rawan perulangan tak berhingga.



Gambar 4.16 Peta konsep struktur kontrol perulangan. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Hitung mundur dengan langkah negatif

Tentukan keluaran dan banyak putaran program berikut. *(Kasus fiktif.)*

```
n_putaran = 0
for n in range(9, 0, -2):
    print(n, end=" ")
    n_putaran += 1
print()
print("putaran:", n_putaran)
```

1. Hitung deret `range(9, 0, -2)`: mulai 9, tiap langkah berkurang 2 → 9, 7, 5, 3, 1. Berikutnya −1 tetapi ia sudah lewat batas 0, jadi berhenti. Angka 0 tidak ikut karena batas akhir.
2. Tiap putaran mencetak `n` dan menambah penghitung → `9 7 5 3 1 ` di satu baris.
3. Banyak putaran: lima.
4. `print()` kosong → pindah baris. Lalu `putaran: 5`.

Jawaban: baris pertama `9 7 5 3 1`, baris kedua `putaran: 5`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Contoh 2 — Jumlah dan rata-rata dengan while

Tentukan keluaran dan jelaskan mengapa baris `i += 1` wajib ada. *(Kasus fiktif.)*

```
nilai = [70, 85, 95]
i = 0
jumlah = 0
while i < len(nilai):
    jumlah += nilai[i]
    i += 1
print(jumlah, jumlah / len(nilai))
```

1. Awal: `i = 0`, `jumlah = 0`, `len(nilai)` = 3.
2. Putaran 1: `i = 0 < 3` benar; `jumlah = 70`; `i = 1`.
3. Putaran 2: `1 < 3` benar; `jumlah = 155`; `i = 2`.
4. Putaran 3: `2 < 3` benar; `jumlah = 250`; `i = 3`.
5. `3 < 3` salah → berhenti. Cetak `250` dan `250 / 3` = 83.33333333333333.
6. Tanpa `i += 1`, `i` tetap 0 sehingga syarat selalu benar dan program berputar selamanya.

Jawaban: `250 83.33333333333333`. Baris `i += 1` adalah pembaruan yang akhirnya membuat syarat salah.

Contoh 3 — Menemukan dan memperbaiki perulangan tak berhingga

Kode berikut ingin mencetak hitung mundur `10 7 4 1` tetapi tidak pernah berhenti. Temukan penyebab dan perbaiki. *(Kasus fiktif.)*

```
# tidak dijalankan (perulangan tak berhingga)
hitung = 10
while hitung > 0:
    print(hitung)
```

1. Periksa tiga bagian `while`: nilai awal `hitung = 10` (ada), syarat `hitung > 0` (ada), **pembaruan** (**tidak ada**).
2. Karena `hitung` tidak pernah berubah, syaratnya selalu benar dan program berulang tanpa henti, mencetak 10 terus-menerus.
3. Perbaikan: kurangi `hitung` sebesar 3 pada tiap putaran:

```
hitung = 10
while hitung > 0:
    print(hitung, end=" ")
    hitung -= 3
```

1. Telusuri: 10 → cetak, jadi 7; 7 → cetak, jadi 4; 4 → cetak, jadi 1; 1 → cetak, jadi −2; `-2 > 0` salah → berhenti.

Jawaban: pembaruan `hitung -= 3` hilang; setelah ditambah, keluaran `10 7 4 1`.

Contoh 4 — Menghitung putaran pada perulangan bersarang

Tentukan keluaran dan berapa kali `print` di dalam dijalankan. *(Kasus fiktif.)*

```
for baris in range(1, 4):
    for kolom in range(2):
        print(baris + kolom, end=" ")
    print()
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. `baris` = 1, 2, 3 (tiga putaran luar); `kolom` = 0, 1 (dua putaran dalam).
2. `baris = 1`: kolom 0 → 1, kolom 1 → 2 → baris pertama `1 2`.
3. `baris = 2`: kolom 0 → 2, kolom 1 → 3 → `2 3`.
4. `baris = 3`: kolom 0 → 3, kolom 1 → 4 → `3 4`.
5. Putaran isi dalam: 3 × 2 = 6.

Jawaban: `1 2`, `2 3`, `3 4`; `print` di dalam dijalankan 6 kali.

Contoh 5 — Mengoreksi selisih satu

Seorang siswa ingin menjumlahkan 1 sampai 5 dengan kode berikut dan mendapat 10, bukan 15. Temukan kesalahan dan perbaiki. *(Kasus fiktif.)*

```
total = 0
for n in range(1, 5):
    total += n
print(total)
```

1. `range(1, 5)` menghasilkan 1, 2, 3, 4 (angka 5 tidak ikut).
2. Jumlah: 1 + 2 + 3 + 4 = 10, sehingga keluarannya `10`.
3. Perbaikan: akhir harus 6 → `range(1, 6)` menghasilkan 1 sampai 5, dan jumlahnya 15.

Jawaban: kesalahan selisih satu; ganti `range(1, 5)` dengan `range(1, 6)`.

Contoh 6 — Skor dan nyawa pada perulangan gim

Dalam sebuah gim (2027), pemain mulai dengan 3 nyawa dan skor 0. Setiap level dari 1 sampai 6 menambah skor sebesar nomor levelnya. Setiap level yang nomornya kelipatan 3 mengurangi satu nyawa. Gim berhenti lebih awal bila nyawa habis. Tentukan skor akhir dan nyawa akhir. *(Kasus fiktif.)*

```
nyawa = 3
skor = 0
for level in range(1, 7):
    skor += level
    if level % 3 == 0:
        nyawa -= 1
        if nyawa == 0:
            break
print(skor, nyawa)
```

1. Siapkan tabel jejak dengan nilai awal `nyawa = 3`, `skor = 0`.
2. Level 1: skor 1; 1 bukan kelipatan 3. Level 2: skor 3.
3. Level 3: skor 6; 3 kelipatan 3, nyawa menjadi 2; belum 0, lanjut.
4. Level 4: skor 10. Level 5: skor 15.
5. Level 6: skor 21; 6 kelipatan 3, nyawa menjadi 1; belum 0, perulangan selesai karena `range(1, 7)` habis.
6. Cetak `21 1`.

Jawaban: `21 1`. Bila nyawa awalnya 2, nyawa menjadi 0 pada level 6 sehingga `break` terjadi, tetapi perulangan toh sudah berada di level terakhir; sedangkan bila nyawa awalnya 1, `break`

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

terjadi pada level 3 dan skor akhir hanya 6. Perubahan satu nilai awal dapat menggeser level tempat perulangan berhenti, sehingga nilai awal selalu dibaca dengan cermat.

Contoh 7 — Memilih for atau while

Pilih perulangan yang paling tepat untuk tiga kebutuhan berikut dan jelaskan alasannya. *(Kasus fiktif.)*

- (a) Menampilkan 30 nomor antrean mulai dari 1.
- (b) Menambahkan Rp25.000 setiap minggu sampai tabungan cukup untuk membeli sepatu seharga Rp400.000.
- (c) Memeriksa setiap barang dalam daftar belanja untuk mencari yang stoknya nol.

1. (a) Jumlah putaran **diketahui** (30) → **`for n in range(1, 31)`**. Perhatikan batas akhirnya 31.
2. (b) Jumlah minggu **tidak diketahui**; yang ada adalah syarat berhenti (tabungan ≥ 400000) → **`while tabungan < 400000`** dengan pembaruan `tabungan += 25000`.
3. (c) Mengolah **setiap elemen** daftar → **`for barang in daftar`**.

Jawaban: (a) `for`, (b) `while`, (c) `for`. Pilihan yang tepat membuat kode lebih ringkas dan memperkecil peluang perulangan tak berhingga.

❖ PETA KONSEP

Gambar 4.16 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **`range(mulai, akhir, langkah)`**: `akhir` **tidak ikut**; langkah negatif menghitung mundur.
- **`while`** butuh tiga hal: **nilai awal, syarat, dan pembaruan**. Pembaruan hilang → perulangan tak berhingga.
- **Penghitung** (`+= 1`) dan **akumulator** (`+= nilai`): perhatikan **di dalam atau di luar** perulangan.
- **`continue`** melewati satu putaran; **`break`** menghentikan seluruh perulangan.
- **Perulangan bersarang**: putaran isi dalam = putaran luar × putaran dalam (bila konstan).
- **Tabel jejak**: satu baris per putaran, tulis nilai awal lebih dulu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menelusuri soal perulangan:

1. **Tulis nilai awal** semua variabel di atas tabel jejak.
2. **Tentukan deretan nilai perulangan** (`range` atau elemen list) sebelum menelusuri isinya.
3. **Telusuri satu putaran penuh**, perbarui tiap kolom, dan jangan melompat.
4. **Cek `break`/`continue`** pada tiap putaran, lalu **perhatikan baris apa yang berada di luar perulangan** (dicetak sekali).
5. **Untuk jumlah putaran**, hitung banyak nilai yang dihasilkan, bukan selisih angka mentah.

Jebakan yang sering dipasang:

- **Batas akhir `range` ikut dihitung**, terutama pada langkah negatif.
- **Jumlah putaran dikira `akhir - mulai`** tanpa memperhitungkan langkah.
- **`continue` dikira `break`**, atau sebaliknya.
- **Baris di luar perulangan dikira berjalan tiap putaran.**
- **Perulangan dalam dihitung sekali** padahal berjalan penuh pada tiap putaran luar.
- **Nilai awal akumulator keliru** (mulai dari 1 untuk jumlah).
- **Syarat `while` tidak pernah salah** karena pembaruan tidak mengarah ke syarat.
- **Pernyataan "angka yang tercetak tetap sama"** pada soal Benar/Salah setelah batas `range` diubah, padahal angka terakhir bisa ikut atau tidak.

> Contoh salah tafsir: pada `range(10, 1, -3)`, seorang siswa menyimpulkan bahwa angka yang tercetak sama dengan pada `range(10, 0, -3)` karena "batas akhirnya hampir sama". Pada `range(10, 0, -3)` hasilnya 10, 7, 4, 1. Pada `range(10, 1, -3)` batas akhir adalah 1 dan **tidak ikut**, sehingga hasilnya 10, 7, 4 saja.

● BEDAH SOAL PREDIKSI

Soal berikut menyajikan hitung mundur dengan `range` bernilai negatif. Perhatikan cara menurunkan deret angkanya lebih dulu dan bagaimana tiap pernyataan Benar/Salah diuji pada deret itu (bagian 1).

►. Gim *Roket Mini* (2027) menampilkan hitung mundur sebelum roket meluncur. Programmer menulis kode berikut.

```
hitung = 0
for i in range(10, 0, -3):
    print(i, end=" ")
    hitung += 1
print()
print("Total:", hitung)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Isi perulangan for dijalankan sebanyak 4 kali.	○	○
Angka terakhir yang tercetak pada baris hitung mundur adalah 1.	○	○
Bila range diubah menjadi range(10, 1, -3), angka yang tercetak tetap 10 7 4 1.	○	○

Pembahasan **Kunci: 1B 2B 3S**

Konsep: `range(awal, akhir, langkah)` dimulai dari awal, bertambah sebesar langkah (boleh negatif), dan berhenti SEBELUM mencapai akhir; nilai akhir tidak ikut.

Langkah:

1. i = 10 → cetak, hitung 1.
2. i = 7 → cetak, hitung 2.
3. i = 4 → cetak, hitung 3.
4. i = 1 → cetak, hitung 4.
5. i berikutnya −2 < batas 0 → berhenti. Keluaran: 10 7 4 1 lalu Total: 4.

Analisis setiap pernyataan:

• **Pernyataan 1 → Benar.** Nilai i: 10, 7, 4, 1. Nilai berikutnya −2 sudah melewati batas 0, jadi berhenti. Total: 4.

• **Pernyataan 2 → Benar.** Urutan tercetak: 10 7 4 1.

• **Pernyataan 3 → Salah.** Jebakan: batas akhir tidak ikut. Dengan batas 1, nilai 1 tidak tercetak; hasilnya 10 7 4.

Jawaban: Benar, Benar, Salah.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Buat tabel jejak untuk setiap soal sebelum memilih jawaban.

1. Programmer gim petualangan menguji aturan nyawa dengan kode Python berikut (2027). Setiap level genap mengurangi satu nyawa.

```
nyawa = 3
level = 1
while nyawa > 0 and level <= 5:
    if level % 2 == 0:
        nyawa -= 1
    level += 1
print(level, nyawa)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Isi perulangan while dijalankan sebanyak 5 kali.	○	○
Bila nilai awal nyawa diubah menjadi 2, perulangan berhenti karena nyawa habis sebelum level melewati 5.	○	○
Keluaran program adalah 5 1.	○	○

2. Dalam gim lari *Kancil Melesat* (2027), skor dihitung dari daftar nilai petak yang dilewati karakter. Petak bernilai 0 adalah petak kosong, sedangkan petak bernilai negatif adalah jebakan yang menghentikan lari. Programmer menulis kode berikut.

```
petak = [3, 0, 5, -1, 4, 2]
skor = 0
for nilai in petak:
    if nilai == 0:
        continue
    if nilai < 0:
        break
    skor += nilai * 2
print(skor)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 28
B. 16
C. 8
D. 26
E. 6

3. Aplikasi celengan digital siswa (2027) mencatat setoran Rp20.000 setiap minggu. Setiap minggu ke-4, ke-8, dan seterusnya, koperasi sekolah memberi bonus Rp5.000. Programmer menulis kode berikut untuk menghitung kapan target Rp100.000 tercapai.

```
tabungan = 0
minggu = 0
while tabungan < 100000:
    minggu += 1
    if minggu % 4 == 0:
        tabungan += 5000
    tabungan += 20000
print(minggu, tabungan)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `4 85000`
B. `5 100000`
C. `6 125000`
D. `5 110000`
E. `5 105000`

4. Dalam gim menyusun menara balok (2027), balok pada baris ke-n bernilai n poin, dan banyak balok pada baris ke-n adalah n buah. Programmer menghitung total poin menara 3 baris dengan kode berikut.

```
poin = 0
for baris in range(1, 4):
    for kolom in range(baris):
        poin += baris
print(poin)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 14
B. 6
C. 9
D. 30
E. 20

Kunci singkat: 1. 1B 2B 3S 2. B 3. E 4. A

✔ PERIKSA DIRI

- Mengapa `range(1, 5)` hanya menghasilkan empat angka, dan bagaimana cara mendapatkan 1 sampai 5?
- Apa perbedaan `break` dan `continue`, dan apa yang terjadi pada `while` bila pembaruan lupa ditulis?
- **Yang harus kuingat:** angka akhir `range` tidak ikut; `while` butuh nilai awal, syarat, dan pembaruan; `continue` melewati satu putaran dan `break` menghentikan perulangan; putaran dalam dijalankan penuh tiap putaran luar; selalu pakai tabel jejak.



Latihan interaktif di LMS

Sub bab 4.3: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-4.3

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 4.4 · Fungsi, prosedur, dan modularisasi

SMKPPLG-4.4

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menganalisis pembagian fungsi dan prosedur** dalam program terstruktur dan **mengevaluasi efektivitas desain modularnya**. Contoh soal resmi Pusmendik untuk modularisasi memperlihatkan sebuah kode dengan fungsi `proses` dan `hitung`, lalu kamu diminta memilih pemanggilan agar keluarannya tepat. Dengan kata lain, kamu harus bisa **menelusuri fungsi yang memanggil fungsi**, memahami apa yang dikembalikan, dan menilai apakah cara program dipecah sudah baik.

Soal bagian ini terbagi dua jenis. Jenis pertama adalah **menelusuri kode**: fungsi dengan parameter bernilai bawaan, fungsi yang tidak mengembalikan apa-apa, atau fungsi yang mengembalikan dua nilai sekaligus. Jenis kedua adalah **menilai desain**: tim memecah aplikasi menjadi modul, lalu kamu menentukan pernyataan mana yang benar tentang manfaat, risiko, dan cara memperbaikinya. Pada jenis kedua, jawaban benar hampir selalu memuat gagasan yang sama: **tulis satu kali, pakai di banyak tempat, dan beri tiap bagian satu tugas yang jelas**.

Setelah mempelajari sub bab ini, kamu dapat:

- menulis dan menelusuri fungsi dengan parameter, argumen, nilai bawaan, dan `return`;
- membedakan `return` dari `print` serta memahami fungsi tanpa `return` yang menghasilkan `None`;
- membedakan variabel lokal dan global, serta menelusuri fungsi yang memanggil fungsi lain;
- memecah program menjadi modul dan memakai `import`;
- menilai efektivitas desain modular dengan konsep kohesi, kopling, dan duplikasi.

◆ KONSEP INTI

Bayangkan program kasir kantin yang ditulis dalam satu berkas sepanjang 600 baris dari atas ke bawah tanpa pembagian. Rumus diskon tertulis di tiga tempat. Ketika aturan diskon berubah, seseorang lupa mengubah satu di antaranya, dan akibatnya total di struk berbeda dengan total di laporan. Masalah seperti ini sangat umum, dan solusinya adalah **modularisasi**: memecah program menjadi bagian-bagian kecil yang masing-masing punya tugas jelas, yaitu **fungsi** dan **modul**.

1. Fungsi: membuat dan memanggil

Fungsi adalah blok perintah yang diberi nama dan dapat dipanggil berulang kali. Fungsi ditulis dengan kata kunci **`def`**. Fungsi menerima masukan lewat **parameter** dan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

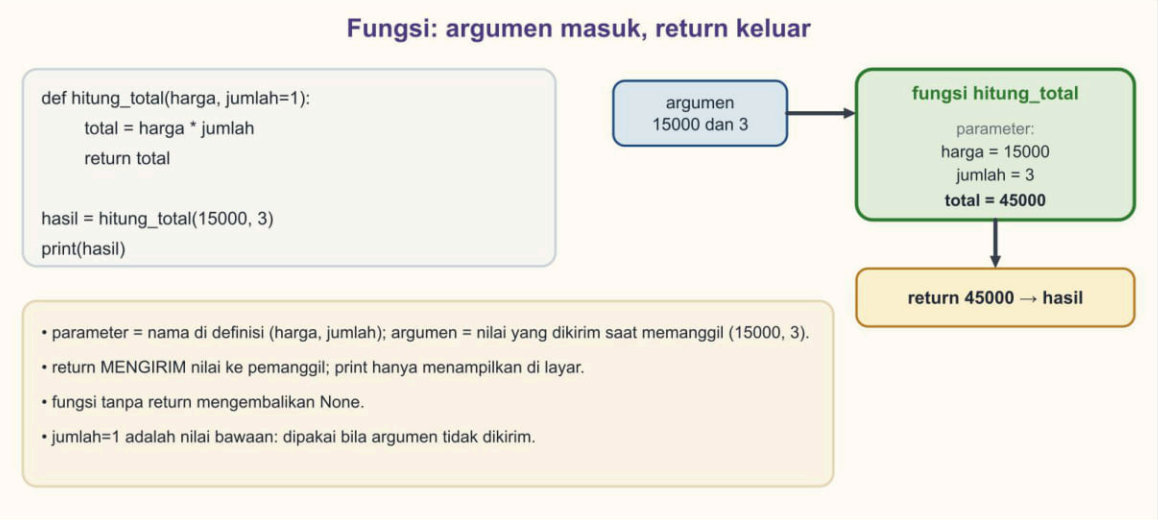
menyerahkan hasilnya lewat **`return`** (Gambar 4.17).

```
def hitung_total(harga, jumlah=1):
    total = harga * jumlah
    return total

hasil = hitung_total(15000, 3)
print(hasil)
```

Keluarannya `45000`. Beberapa istilah yang sering diuji:

- **Parameter** adalah nama di definisi fungsi (`harga`, `jumlah`). **Argumen** adalah nilai yang dikirim saat fungsi dipanggil (`15000`, `3`). Argumen "mengisi" parameter menurut urutannya.
- **Nilai bawaan (default)**: `jumlah=1` berarti bila argumen kedua tidak dikirim, `jumlah` bernilai 1. Jadi `hitung_total(15000)` menghasilkan `15000`.
- **`return`** menghentikan fungsi dan **mengirim nilai kembali** kepada pemanggil.



Gambar 4.17 Argumen masuk ke parameter, return mengirim nilai keluar. Sumber: digambar oleh Tim Next Level Study Indonesia.

Fungsi yang dibuat dengan baik dapat dipanggil dari mana saja dan sebanyak apa pun, sehingga kode yang sama tidak perlu ditulis ulang.

2. return dan print: dua hal yang berbeda

Ini kesalahan konsep yang paling sering diuji. **`print` menampilkan nilai di layar**; **`return` mengirim nilai ke tempat fungsi dipanggil**. Keduanya tidak sama.

```
def hitung_poin(belanja):
    return belanja // 10000

def cetak_struk(nama, belanja):
    poin = hitung_poin(belanja)
    print(nama, "mendapat", poin, "poin")

hasil = cetak_struk("Rani", 83000)
print(hasil)
```

Keluarannya dua baris: `Rani mendapat 8 poin`, lalu `None`. `cetak_struk` hanya mencetak

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

dan **tidak punya `return`**. Fungsi yang tidak memiliki `return` otomatis mengembalikan **`None`**. Karena itu `hasil` berisi `None`, dan `print(hasil)` mencetak `None`. Pemanggilan `cetak_struk` sendiri menghasilkan baris pertama (efek samping), sedangkan baris kedua berasal dari `print(hasil)`.

Fungsi yang **mengerjakan sesuatu tanpa mengembalikan nilai** (mencetak, menyimpan ke berkas) sering disebut **prosedur**. Fungsi yang **mengembalikan nilai** disebut fungsi dalam arti sempit. Di Python keduanya ditulis dengan `def`; yang membedakan adalah ada atau tidaknya `return`.

Satu jebakan lain: nilai yang dikembalikan **tidak akan terpakai** bila tidak ditampung atau dicetak. Menulis `hitung_poin(83000)` sendirian tidak menampilkan apa-apa. Untuk menampilkan, tulis `print(hitung_poin(83000))`.

3. Mengembalikan lebih dari satu nilai

Sebuah fungsi dapat mengembalikan beberapa nilai sekaligus, dipisah koma. Python membungkusnya sebagai **tuple**.

```
def ringkas(nilai):
    tertinggi = max(nilai)
    rata = sum(nilai) / len(nilai)
    return tertinggi, rata

t, r = ringkas([60, 75, 90])
print(t, r)
print(ringkas([60, 75, 90]))
```

Keluarannya `90 75.0` pada baris pertama dan `(90, 75.0)` pada baris kedua. Pada `t, r = ...`, kedua nilai **diuraikan** ke dua variabel. Bila hasilnya dicetak langsung, tampil sebagai tuple dengan kurung. Perhatikan `75.0`: pembagian `/` menghasilkan `float`. Fungsi bawaan yang berguna: `max()`, `min()`, `sum()`, `len()`.

4. Lingkup variabel: lokal dan global

Variabel yang dibuat **di dalam fungsi** hanya ada selama fungsi berjalan dan hanya dapat dipakai di sana (**lokal**). Variabel yang dibuat **di luar fungsi** disebut **global** (Gambar 4.18).

```
diskon = 10

def hitung(harga):
    diskon = 5
    return harga - diskon

print(hitung(100))
print(diskon)
```

Keluarannya `95`, lalu `10`. Di dalam `hitung`, `diskon = 5` membuat variabel **lokal baru** bernama sama; variabel global `diskon` tidak berubah. Sebuah fungsi boleh **membaca** variabel global, tetapi praktik yang baik adalah **mengirim data lewat parameter dan menerima hasil lewat return**. Variabel global yang diubah dari banyak tempat membuat program sulit dilacak: sulit diketahui bagian mana yang mengubah nilainya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 4.18 Variabel lokal berada di dalam fungsi, variabel global di luar fungsi. Sumber: digambar oleh Tim Next Level Study Indonesia.

5. Fungsi yang memanggil fungsi

Fungsi boleh memanggil fungsi lain. Cara menelusurinya adalah mengikuti **urutan pemanggilan** dan **nilai yang dikembalikan** (Gambar 4.19).

```
def harga_akhir(h, d=0):
    return h - d

def total(daftar, potongan):
    jumlah = 0
    for h in daftar:
        jumlah += harga_akhir(h)
    return harga_akhir(jumlah, potongan)

print(total([20000, 15000, 10000], 5000))
```

Telusuri: di dalam perulangan, `harga_akhir(h)` hanya dikirimi **satu** argumen, jadi `d` memakai **nilai bawaan 0** dan hasilnya `h - 0 = h`. Setelah tiga putaran, `jumlah` = 45000. Terakhir `harga_akhir(45000, 5000)` memberi `45000 - 5000 = 40000`. Keluarannya `40000`.



Gambar 4.19 Urutan fungsi yang memanggil fungsi lain, dan nilai yang dikembalikan ke atas. Sumber: digambar oleh Tim Next Level Study Indonesia.

Aturan telusur: fungsi yang **terakhir dipanggil selesai lebih dulu** dan mengembalikan nilainya kepada pemanggilnya, begitu seterusnya sampai ke pemanggil paling awal.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

6. Modul dan import

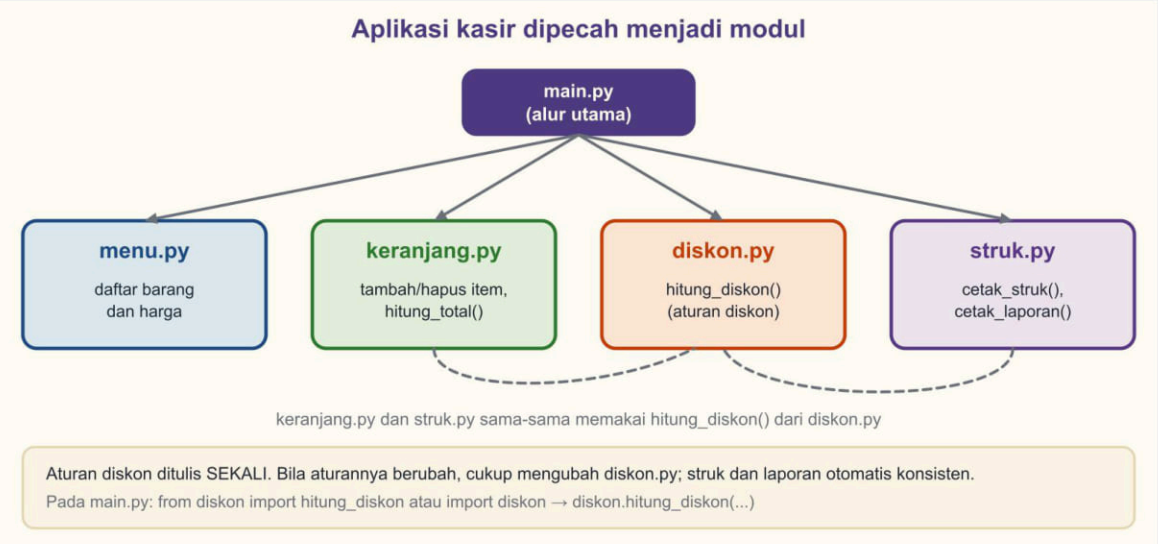
Ketika program membesar, fungsi-fungsinya dikelompokkan menurut tugas dan disimpan dalam **berkas terpisah**. Setiap berkas Python (`.py`) disebut **modul**. Modul lain memakainya dengan **`import`**:

```
# berkas diskon.py
def hitung_diskon(total):
    if total >= 100000:
        return total * 0.1
    return 0

# berkas main.py
from diskon import hitung_diskon
print(hitung_diskon(250000))
```

Keluarannya `25000.0`. Ada dua bentuk `import` yang umum: `import diskon` (dipanggil dengan `diskon.hitung_diskon(...)`) dan `from diskon import hitung_diskon` (dipanggil langsung). Python juga menyediakan modul bawaan, seperti `math` dan `random`.

Gambar 4.20 memperlihatkan aplikasi kasir yang dipecah menjadi modul menurut tugasnya: daftar barang, keranjang, aturan diskon, dan struk. Modul `keranjang.py` dan `struk.py` sama-sama memanggil `hitung_diskon()` dari `diskon.py`, sehingga aturan diskon hanya ada di satu tempat.



Gambar 4.20 Contoh pembagian modul pada aplikasi kasir. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

7. Mengapa modularisasi efektif

Desain modular dinilai dari beberapa kriteria.

- **Tidak ada duplikasi (prinsip DRY: *Don't Repeat Yourself*).** Logika yang sama ditulis **sekali** dalam fungsi dan dipakai di banyak tempat. Bila ditulis di dua tempat, keduanya bisa berkembang berbeda, misalnya satu dibulatkan dan satu tidak, sehingga hasilnya selisih. Bila aturan berubah, cukup mengubah satu fungsi.
- **Mudah diuji.** Fungsi kecil dengan masukan dan keluaran jelas dapat diuji sendiri dengan beberapa contoh sebelum digabungkan. Pengujian satu fungsi sekaligus melindungi semua bagian yang memakainya.
- **Mudah diubah.** Bila format penyimpanan skor diganti dari teks ke JSON, hanya modul penyimpan yang berubah, **selama fungsi yang dipanggil modul lain tetap sama** (nama dan parameternya).
- **Kerja tim.** Anggota tim dapat mengerjakan modul berbeda secara bersamaan.
- **Mudah dibaca dan dipakai ulang.** Fungsi dengan nama yang jelas seperti `hitung_diskon()` menjelaskan dirinya sendiri.

Dua konsep membantu menilai kualitas pembagian modul (Gambar 4.21):

- **Kohesi tinggi**: sebuah fungsi atau modul mengerjakan **satu tugas yang jelas** (modul fisika hanya mengurus gerak dan tabrakan). Fungsi raksasa yang mengerjakan input, hitung, gambar, dan simpan sekaligus berkohesi rendah.
- **Kopling rendah**: modul **sedikit bergantung** pada rincian modul lain; mereka berhubungan lewat **fungsi dengan parameter dan return** yang jelas. Kopling tinggi terjadi bila modul saling mengubah variabel global atau saling mengetahui isi dalam modul lain, sehingga mengubah satu bagian merusak bagian lain.



Gambar 4.21 Desain dengan kohesi tinggi dan kopling rendah dibandingkan dengan desain yang buruk. Sumber: digambar oleh Tim Next Level Study Indonesia.

Contoh penilaian: bila `fisika.py` juga mengambil alih tugas menggambar karakter, kohesinya turun, dan modul itu kini bergantung pada cara menggambar. Bila posisi karakter dijadikan variabel global yang boleh diubah semua modul, koplingnya naik dan sulit dilacak siapa yang mengubahnya. Sebaliknya, bila `kontrol.py` dapat diganti dengan versi layar

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

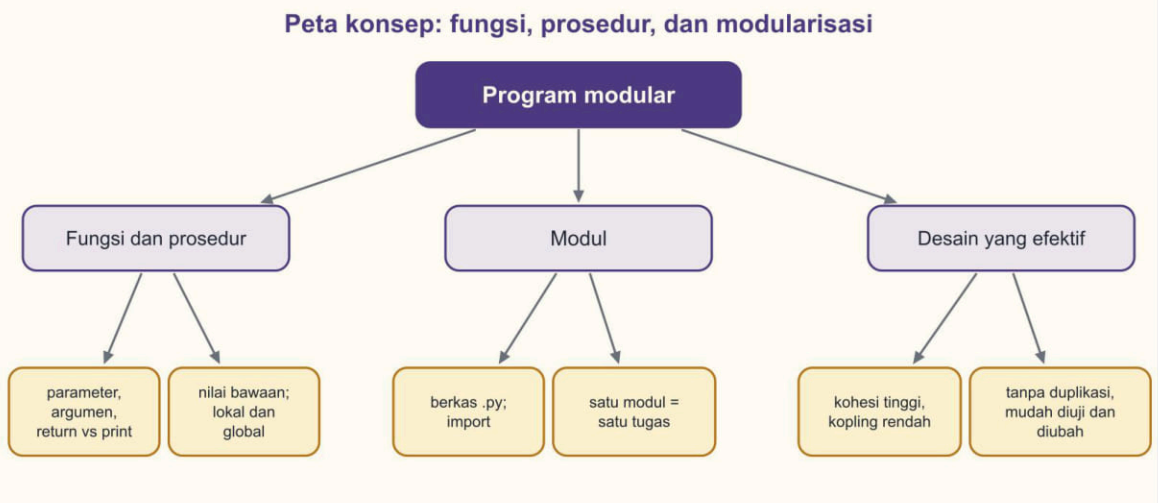
sentuh tanpa mengubah `fisika.py`, itu tanda desain modular yang baik.

8. Kebiasaan menulis fungsi yang baik

- **Satu fungsi, satu tugas.** Bila sulit menamai fungsi tanpa kata "dan", mungkin ia mengerjakan terlalu banyak.
- **Nama yang jelas**, berupa kata kerja: `hitung_diskon`, `cetak_struk`, `simpan_skor`.
- **Parameter dan return**, bukan variabel global.
- **Keterangan singkat (docstring)** di awal fungsi menjelaskan tugas, parameter, dan hasilnya:

```
def hitung_diskon(total):
    """Mengembalikan potongan 10% bila total minimal 100000, selain itu 0."""
    if total >= 100000:
        return total * 0.1
    return 0
```

- **Uji tiap fungsi** dengan beberapa nilai, termasuk nilai batas, sebelum dipakai di bagian lain. Fungsi kecil mudah diuji karena masukan dan keluarannya jelas.
- **Beri nama konstanta** untuk nilai tetap seperti tarif pajak (`TARIF_PAJAK`), supaya perubahannya cukup di satu tempat dan maksudnya terbaca.



Gambar 4.22 Peta konsep fungsi, prosedur, dan modularisasi. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Fungsi tanpa return

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
def hitung_luas(p, l):
    luas = p * l
    print("Luas:", luas)

a = hitung_luas(4, 5)
print(a)
print(a is None)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. Panggilan `hitung_luas(4, 5)` menghitung `luas = 20` dan **mencetak** `Luas: 20`.
2. Fungsi tidak memakai `return`, sehingga mengembalikan `None`; `a` berisi `None`.
3. `print(a)` mencetak `None`.
4. `a is None` bernilai `True`, jadi dicetak `True`.

Jawaban: `Luas: 20`, `None`, `True`. Untuk memakai luasnya, fungsi harus memakai `return luas`.

Contoh 2 — Fungsi yang memanggil fungsi dengan nilai bawaan

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
def tarif(jarak, ekspres=False):
    biaya = jarak * 2000
    if ekspres:
        biaya = biaya + 5000
    return biaya

def total_kirim(jarak_list):
    jumlah = 0
    for j in jarak_list:
        jumlah += tarif(j)
    return jumlah + tarif(2, True)

print(total_kirim([1, 3, 4]))
```

1. Di dalam perulangan, `tarif(j)` hanya dikirimi satu argumen, jadi `ekspres` memakai nilai bawaan `False`.
2. `tarif(1)` = 2000; `tarif(3)` = 6000; `tarif(4)` = 8000. Jumlah = 16000.
3. `tarif(2, True)`: biaya 4000, `ekspres` benar → tambah 5000 → 9000.
4. Hasil: 16000 + 9000 = 25000.

Jawaban: `25000`. Nilai bawaan membuat satu fungsi dapat dipakai dengan dua cara.

Contoh 3 — Dua nilai kembalian dan lingkup variabel

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
poin = 50

def tambah_bonus(skor):
    poin = skor + 10
    return poin, skor

a, b = tambah_bonus(70)
print(a, b, poin)
```

1. `tambah_bonus(70)`: di dalam fungsi, `poin = 70 + 10 = 80` adalah variabel **lokal**.
2. Fungsi mengembalikan `(80, 70)`, diuraikan menjadi `a = 80` dan `b = 70`.
3. Variabel **global** `poin` tidak berubah dan tetap 50.
4. `print(a, b, poin)` → `80 70 50`.

Jawaban: `80 70 50`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Contoh 4 — Menilai dua desain modul

Sebuah tim membuat aplikasi absensi (2027). Desain A: satu fungsi `proses()` memuat pembacaan data, perhitungan kehadiran, pencetakan laporan, dan penyimpanan, dengan variabel global untuk daftar siswa. Desain B: empat fungsi (`baca_data()`, `hitung_hadir()`, `cetak_laporan()`, `simpan()`) yang saling mengirim data lewat parameter dan return. Desain mana yang lebih efektif dan mengapa? *(Kasus fiktif.)*

1. **Kohesi**: A mengerjakan empat tugas dalam satu fungsi (rendah); B satu tugas per fungsi (tinggi).
2. **Kopling**: A mengubah variabel global yang sama dari semua bagian (tinggi); B hanya berhubungan lewat parameter dan return (rendah).
3. **Pengujian**: `hitung_hadir()` pada B dapat diuji dengan data buatan tanpa membaca berkas; pada A tidak dapat dipisahkan.
4. **Perubahan**: bila format laporan berubah, di B cukup mengubah `cetak_laporan()`.

Jawaban: desain B lebih efektif; kohesinya tinggi, koplingnya rendah, mudah diuji dan diubah.

Contoh 5 — Menghilangkan duplikasi dengan fungsi

Sebuah toko menghitung pajak 11 % pada tiga tempat berbeda dalam programnya: struk, laporan harian, dan rekap bulanan. Tuliskan fungsi pengganti, tentukan pajak untuk belanja Rp200.000, dan jelaskan keuntungannya bila tarif pajak berubah menjadi 12 %. *(Kasus fiktif.)*

```
TARIF_PAJAK = 0.11

def hitung_pajak(belanja):
    return belanja * TARIF_PAJAK

print(hitung_pajak(200000))
print(hitung_pajak(50000) + hitung_pajak(150000))
```

1. Rumus yang sama (belanja × 0,11) ditulis tiga kali. Kita memusatkannya dalam satu fungsi `hitung_pajak(belanja)`.
2. `hitung_pajak(200000)` = 200000 × 0,11 = 22000.0.
3. Baris terakhir memanggil fungsi dua kali: 5500.0 + 16500.0 = 22000.0.
4. Bila tarif berubah menjadi 12 %, cukup mengubah satu baris (`TARIF_PAJAK = 0.12`). Struk, laporan, dan rekap otomatis memakai tarif baru dan tidak mungkin saling berbeda.

Jawaban: `22000.0` dan `22000.0`. Keuntungannya adalah satu tempat perubahan, tidak ada selisih antarlaporan, dan fungsi dapat diuji sekali saja.

Contoh 6 — Menguji fungsi sebelum dipakai

Sebelum memasang fungsi `denda_telat(hari)` ke aplikasi, programmer mengujinya dengan beberapa nilai. Fungsi itu seharusnya memberi denda Rp1.000 per hari untuk keterlambatan 1 sampai 5 hari, dan tetap Rp5.000 bila lebih dari 5 hari. *(Kasus fiktif.)*

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
def denda_telat(hari):
    if hari <= 0:
        return 0
    if hari <= 5:
        return hari * 1000
    return 5000

for h in [0, 1, 5, 6, 10]:
    print(h, denda_telat(h))
```

1. Hitung hasil seharusnya dari aturan: 0 → 0; 1 → 1000; 5 → 5000; 6 → 5000; 10 → 5000.
2. Telusuri fungsi: `hari <= 0` untuk 0 → 0. Nilai 1 dan 5 masuk `hari <= 5` → 1000 dan 5000. Nilai 6 dan 10 jatuh ke `return 5000`.
3. Semua hasil cocok, termasuk nilai batas 5 dan 6.

Jawaban: `0 0`, `1 1000`, `5 5000`, `6 5000`, `10 5000`. Karena fungsi sudah teruji pada nilai batas, ia aman dipakai oleh modul lain.

❖ PETA KONSEP

Gambar 4.22 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Parameter** ada di definisi; **argumen** dikirim saat memanggil; **nilai bawaan** dipakai bila argumen tidak dikirim.
- **`return` mengirim nilai ke pemanggil; `print` hanya menampilkan.** Fungsi tanpa `return` menghasilkan **`None`**.
- **`return a, b`** mengembalikan tuple; bisa diuraikan `x, y = ...`.
- **Variabel lokal** hanya ada di dalam fungsi; mengisi nama yang sama membuat variabel lokal baru, global tidak berubah.
- **Modul** = berkas `.py`; dipakai dengan **`import`**.
- **Desain modular yang efektif**: tanpa duplikasi (DRY), **kohesi tinggi**, **kopling rendah**, mudah diuji dan diubah.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menelusuri soal fungsi:

1. **Catat setiap definisi fungsi** dan parameternya, termasuk nilai bawaan.
2. **Mulai dari pemanggilan paling luar**; ikuti ke dalam setiap fungsi yang dipanggil, lalu kembali dengan nilai yang di-return.
3. **Pisahkan apa yang dicetak di dalam fungsi** (efek samping) dari **apa yang dikembalikan** (nilai).
4. **Periksa lingkup**: variabel mana lokal, mana global.

Langkah menilai desain modular:

1. **Cari duplikasi**: adakah logika yang sama ditulis di dua tempat?
2. **Tanyakan: bila aturan atau format berubah, berapa tempat yang harus diubah?** Makin sedikit, makin baik.
3. **Pilih pernyataan yang menyebut tugas tunggal per modul, hubungan lewat parameter dan return, dan pengujian terpisah.**

Jebakan yang sering dipasang:

- **Fungsi tanpa `return` dikira mengembalikan angka** (yang tercetak di dalamnya).
- **`return` dan `print` dianggap sama.**
- **Nilai bawaan lupa dipakai** saat fungsi dipanggil dengan satu argumen.
- **Variabel lokal dikira mengubah variabel global** bernama sama.
- **Nilai kembalian dua elemen dicetak tanpa kurung** padahal bentuknya tuple, atau sebaliknya.
- **Satu fungsi besar dianggap "paling mudah dirawat".**
- **Menggabungkan semua modul menjadi satu berkas** untuk menghindari duplikasi.
- **Variabel global dipakai bersama agar "praktis".**
- **Fungsi `fisika` ikut menggambar** karena "lebih sedikit panggilan".

> Contoh salah tafsir: pada `def total(a, b): print(a + b)` lalu `x = total(2, 3)`, seorang siswa menyimpulkan bahwa `x` berisi 5. Fungsi itu hanya mencetak 5 dan tidak memakai `return`, sehingga `x` berisi `None`.

• BEDAH SOAL PREDIKSI

Soal berikut memperlihatkan fungsi yang mencetak sesuatu tanpa mengembalikan nilai, lalu hasil pemanggilannya dicetak. Perhatikan bahwa ada **dua sumber keluaran** yang berbeda: cetakan di dalam fungsi dan cetakan hasil `return` (bagian 2).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Aplikasi kasir koperasi sekolah memberi 1 poin untuk setiap belanja Rp10.000 (2027). Programmer menulis dua fungsi berikut, lalu mengujinya.

```
def hitung_poin(belanja):
    return belanja // 10000

def cetak_struk(nama, belanja):
    poin = hitung_poin(belanja)
    print(nama, "mendapat", poin, "poin")

hasil = cetak_struk("Sari", 57000)
print(hasil)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. `Sari mendapat 5 poin` saja
B. `Sari mendapat 5.7 poin` lalu `None`
C. `Sari mendapat 5 poin` lalu `5`
D. `None` lalu `Sari mendapat 5 poin`
E. `Sari mendapat 5 poin` lalu `None`

Pembahasan **Kunci: E**

Konsep: Fungsi yang memakai `return` mengembalikan nilai. Fungsi tanpa return (prosedur) hanya menjalankan perintah, misalnya mencetak, dan mengembalikan `None`. Operator `//` adalah pembagian bulat.

Langkah:

1. cetak_struk("Sari", 57000) dijalankan: poin = 57000 // 10000 = 5.
2. Di dalam cetak_struk, print mencetak: Sari mendapat 5 poin.
3. cetak_struk tidak memiliki return → hasil = None.
4. print(hasil) mencetak None. Urutan keluaran: Sari mendapat 5 poin, lalu None.

Analisis setiap opsi:

- **A.** ✗ Jebakan: baris print(hasil) juga dijalankan dan mencetak None.
- **B.** ✗ // adalah pembagian bulat, hasilnya 5, bukan 5.7.
- **C.** ✗ Jebakan: cetak_struk tidak mengembalikan poin; nilai 5 hanya dicetak di dalamnya.
- **D.** ✗ Urutannya terbalik: struk dicetak lebih dulu saat cetak_struk dipanggil.
- **E.** ✓ Struk dicetak di dalam fungsi, lalu hasil berisi None karena tidak ada return.

Jawaban: E.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk soal kode, tulis hasil tiap pemanggilan fungsi. Untuk soal desain, tanyakan: apakah pernyataan itu menyebut tugas tunggal, tanpa duplikasi, atau pengujian terpisah?

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. Aplikasi kantin sekolah (2027) menghitung total belanja, lalu memotongnya dengan voucher. Programmer menulis dua fungsi berikut.

```
def harga_akhir(harga, diskon=0):
    return harga - diskon

def total(daftar, potongan):
    jumlah = 0
    for h in daftar:
        jumlah += harga_akhir(h)
    return harga_akhir(jumlah, potongan)

print(total([20000, 15000, 10000], 5000))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `30000`
B. `45000`
C. `15000`
D. `40000`
E. Error, karena `harga_akhir(h)` kurang satu argumen

2. Aplikasi rapor praktik (2027) membutuhkan nilai tertinggi dan rata-rata sekaligus. Programmer menulis fungsi berikut.

```
def ringkas(nilai):
    tertinggi = max(nilai)
    rata = sum(nilai) / len(nilai)
    return tertinggi, rata

t, r = ringkas([70, 80, 90])
print(t, r)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `90 80`
B. `(90, 80.0)`
C. `80.0 90`
D. `90 80.0`
E. Error, karena return hanya boleh satu nilai

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Program kasir kantin SMK Karya (2027) ditulis dalam satu berkas sepanjang 600 baris tanpa fungsi. Rumus diskon yang sama ditulis ulang di tiga tempat, sehingga ketika aturan diskon berubah, satu tempat terlupa diubah dan total belanja di laporan berbeda dengan di struk. Tim memutuskan memecah program menjadi beberapa fungsi dan modul.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang rencana tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Rumus diskon ditulis sekali dalam fungsi hitung_diskon() sehingga perubahan aturan dilakukan di satu tempat.
☐ B. Satu fungsi besar proses_semua() yang memuat input, hitung, dan cetak paling mudah dirawat.
☐ C. Fungsi hitung_diskon() dapat diuji terpisah dengan beberapa contoh belanja sebelum digabungkan.
☐ D. Pembagian modul memungkinkan anggota tim mengerjakan bagian berbeda pada waktu yang sama.
☐ E. Fungsi cetak_struk() dan cetak_laporan() memanggil hitung_diskon() yang sama sehingga hasil diskon keduanya sama.

4. Aplikasi Kantin Online SMK Harapan (2027) dibagi menjadi modul `menu.py`, `pesanan.py`, `pembayaran.py`, dan `laporan.py`. Fungsi `hitung_total()` ditulis terpisah di pesanan.py dan di laporan.py oleh dua anggota tim. Versi di pesanan.py membulatkan total ke ratusan rupiah, sedangkan versi di laporan.py tidak. Akibatnya, jumlah di laporan harian selalu berbeda beberapa rupiah dari jumlah semua pesanan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang masalah dan perbaikannya adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Selisih muncul karena logika yang sama ditulis di dua tempat lalu berkembang berbeda.
☐ B. hitung_total() sebaiknya ditulis di satu modul, lalu dipakai oleh pesanan.py dan laporan.py.
☐ C. Menggabungkan keempat modul menjadi satu berkas besar adalah cara terbaik mencegah duplikasi.
☐ D. Setelah perbaikan, pengujian hitung_total() sekaligus melindungi fitur pesanan dan laporan.
☐ E. Aturan pembulatan dan parameter hitung_total() perlu didokumentasikan agar dipakai dengan benar.

5. Tim gim *Pelari Candi* (2027) membagi kode menjadi empat modul: `kontrol.py` (membaca tombol), `fisika.py` (gerak dan tabrakan), `tampilan.py` (menggambar layar), dan `simpan.py` (menyimpan skor ke berkas). Modul lain memakai fungsi `simpan_skor(nama,`

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

`skor`) untuk menyimpan skor. Anggota tim mengajukan beberapa pendapat.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pendapat yang tepat tentang desain modular tersebut adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Fungsi di fisika.py sebaiknya juga menggambar karakter agar tidak perlu memanggil tampilan.py.

☐ B. Bila format berkas skor diganti ke JSON, perubahan hanya dilakukan di simpan.py selama simpan_skor() tetap sama.

☐ C. Posisi karakter dijadikan variabel global agar semua modul dapat mengubahnya secara langsung.

☐ D. fisika.py dapat diuji dengan data posisi buatan tanpa harus menjalankan tampilan gim.

☐ E. kontrol.py dapat diganti versi layar sentuh untuk ponsel tanpa mengubah fisika.py.

Kunci singkat: 1. D 2. D 3. A, C, D, E 4. A, B, D, E 5. B, D, E

✔ PERIKSA DIRI

- Mengapa fungsi yang hanya memakai `print` tidak dapat dipakai untuk menyimpan hasil hitungannya dalam variabel?
- Mengapa rumus diskon sebaiknya ditulis sekali dalam satu fungsi, bukan di setiap tempat yang membutuhkannya?
- **Yang harus kuingat:** return mengirim nilai, print menampilkan; tanpa return hasilnya None; parameter lawan argumen; variabel lokal tidak mengubah global; modul = berkas `.py` dengan import; desain modular baik bila tanpa duplikasi, kohesi tinggi, kopling rendah.



Latihan interaktif di LMS

Sub bab 4.4: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-4.4

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kuis Akhir Bab 4

SMKPPLG-KB4

Jumlah soal: 25 **Waktu anjuran:** 60 menit

Kerjakan sendiri dan berwaktu. Setelah selesai, cocokkan dengan kunci di Bagian D, isi tabel skor, lalu lihat anjuran di bawahnya.



Kerjakan Kuis Bab 4 di LMS

Skor dan progresmu tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-KB4

1. Perhatikan potongan program Python berikut.

```
a = 17
b = 5
print(a / b)
print(a // b)
print(a % b)
print(a ** 2)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang keluaran program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Baris pertama mencetak 3.4

☐ B. Baris kedua mencetak 4 karena pembagian dibulatkan ke bilangan terdekat

☐ C. Baris ketiga mencetak 2, yaitu sisa pembagian 17 oleh 5

☐ D. Baris keempat mencetak 34 karena 17 dan 2 disambung

☐ E. Baris kedua mencetak 3, yaitu hasil bagi yang dibulatkan ke bawah

2. Panitia lomba membuat program untuk mengolah kode peserta (2027).

```
kode = "SMK-PPLG-2027"
bagian = kode[4:8]
print(bagian.lower() + str(len(bagian)))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. pplg 4

B. pplg4

C. pplg-5

D. plg-4

E. TypeError

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Perhatikan program Python berikut yang mensimulasikan antrean loket (2027).

```
antrean = ["A", "B", "C", "D"]
antrean.append("E")
x = antrean.pop(0)
y = antrean.pop()
antrean.insert(1, "Z")
print(x, y, antrean)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Variabel x berisi "A" karena pop(0) mengambil elemen pertama.	○	○
Variabel y berisi "E" karena pop() tanpa argumen mengambil elemen terakhir.	○	○
Isi akhir antrean adalah ["B", "C", "Z", "D"].	○	○

4. Seorang siswa menguji potongan program Python untuk menghitung belanja di warung kelas (2027).

```
harga = "12500"
jumlah = 3
print(int(harga) * jumlah)
print(harga * jumlah)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. 37500 lalu 125001250012500
B. 37500 lalu 37500
C. 37500 lalu TypeError
D. 125001250012500 lalu 37500
E. TypeError pada baris pertama

5. Perhatikan potongan program Python berikut.

```
buah = ["apel", "jeruk", "mangga", "pisang"]
print(buah[2])
print(buah[-1])
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. jeruk lalu apel
B. mangga lalu apel
C. mangga lalu pisang
D. jeruk lalu pisang
E. pisang lalu mangga

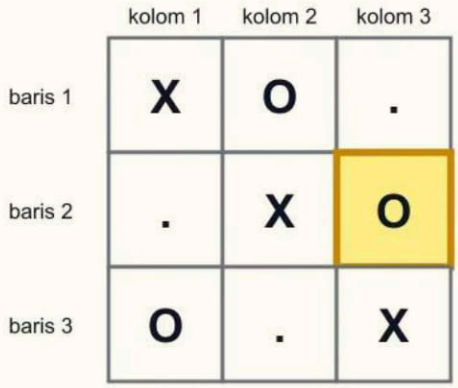
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

6. Sebuah gim sederhana menyimpan papan permainan pada list dua dimensi (2027), seperti pada gambar. Satu sel ditandai kuning.

```
papan = [["X", "O", "."],
         [".", "X", "O"],
         ["O", ".", "X"]]
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Ekspresi yang menghasilkan isi sel yang ditandai kuning (baris ke-2, kolom ke-3 pada gambar) adalah ....

Papan permainan (nomor baris dan kolom pada gambar dihitung mulai dari 1)



A. `papan[2][3]`
B. `papan[3][2]`
C. `papan[2][1]`
D. `papan[1][2]`
E. `papan[1][1]`

7. Sebuah warung (2027) menyimpan stok barang pada dictionary dan memperbaruinya dengan program berikut.

```
stok = {"kopi": 12, "teh": 7, "gula": 20}
stok["teh"] -= 3
stok["susu"] = 5
del stok["gula"]
print(len(stok), stok["teh"])
print("gula" in stok, 5 in stok)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Baris pertama yang dicetak berisi dua angka, yaitu 3 dan 4
☐ B. Baris kedua mencetak False True karena nilai 5 ada di dalam dictionary
☐ C. Baris kedua mencetak False False karena operator in memeriksa kunci, bukan nilai
☐ D. len(stok) bernilai 4 karena kunci gula tetap terhitung setelah dihapus
☐ E. Kunci susu ditambahkan ke dictionary dengan nilai 5

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

8. Tim sebuah studio (2027) meninjau kualitas modul program kasir mereka. Satu modul menghitung total belanja, satu modul mencetak struk, dan satu modul menyimpan data penjualan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Fungsi yang baik mengerjakan satu tugas tunggal dan menerima data lewat parameter.	○	○
Variabel global bersama lebih baik daripada parameter karena membuat fungsi lebih praktis dipanggil.	○	○
Modul yang dapat diuji sendiri memudahkan pencarian kesalahan karena letaknya lebih mudah dipersempit.	○	○

9. Perhatikan potongan program Python berikut.

```
def kuadrat(x):
    return x * x

hasil = kuadrat(6) + kuadrat(2)
print(hasil)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 36
B. 40
C. 64
D. 8
E. 38

10. Perhatikan potongan program Python berikut.

```
suhu = 30
if suhu >= 35:
    print("panas sekali")
elif suhu >= 28:
    print("hangat")
else:
    print("sejuk")
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. panas sekali
B. sejuk
C. panas sekali lalu hangat
D. hangat lalu sejuk
E. hangat

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

11. Sebuah gim sederhana (2027) memakai perulangan berikut untuk mengatur putaran permainan.

```
nyawa = 3
skor = 0
putaran = 0
while nyawa > 0 and putaran < 6:
    putaran += 1
    if putaran % 2 == 0:
        nyawa -= 1
    else:
        skor += 10
print(putaran, nyawa, skor)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Perulangan berhenti setelah putaran ke-6

☐ B. Skor bertambah pada putaran ganjil sehingga skor akhir adalah 30

☐ C. Nilai nyawa pada akhir program adalah 1

☐ D. Perulangan berhenti pada putaran ke-5 karena skor sudah mencapai 30

☐ E. Nyawa berkurang pada putaran genap sehingga nyawa akhir adalah 0

12. Perhatikan potongan program Python berikut.

```
total = 100

def tambah(n):
    total = n + 5
    return total

def reset():
    global total
    total = 0

a = tambah(10)
reset()
print(a, total)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Keluaran program adalah 15 0

☐ B. Pemanggilan tambah(10) mengubah variabel global total menjadi 15

☐ C. Variabel total di dalam fungsi tambah adalah variabel lokal

☐ D. Pemanggilan reset() mengubah total global menjadi 0 karena memakai kata global

☐ E. Tanpa kata global pada reset(), keluaran total tetap 0

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

13. Seorang siswa menulis program untuk mengelompokkan umur peserta kegiatan (2027) dan menjalankannya dengan umur 15.

```
umur = 15
if umur >= 6:
    kategori = "anak"
elif umur >= 13:
    kategori = "remaja"
elif umur >= 18:
    kategori = "dewasa"
else:
    kategori = "balita"
print(kategori)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang keluaran dan perbaikan program adalah ....

A. keluarannya balita, karena umur 15 tidak memenuhi syarat apa pun yang ditulis pada program
B. keluarannya anak; syarat perlu diurutkan dari yang paling ketat agar tiap kategori terjangkau
C. keluarannya remaja, karena cabang kedua dijalankan sesudah cabang pertama selesai
D. keluarannya dewasa, karena ketiga syarat diperiksa sampai yang terakhir bernilai benar
E. keluarannya anak dan remaja, karena dua syarat pertama sama-sama bernilai benar

14. Perhatikan potongan program Python berikut.

```
hitung = 0
for i in range(1, 5):
    for j in range(i):
        hitung += 1
print(hitung)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. 10
B. 16
C. 4
D. 6
E. 15

15. Perhatikan potongan program Python berikut.

```
for i in range(2, 11, 3):
    print(i, end=" ")
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 2 5 8 11
B. 5 8 11
C. 2 3 4 5 6 7 8 9 10
D. 2 5 8
E. 3 6 9

16. Perhatikan potongan program Python berikut.

```
n = 0
total = 0
while total < 20:
    n += 1
    total += n * 2
print(n, total)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 3 12
B. 5 30
C. 4 10
D. 6 21
E. 4 20

17. Sebuah studio memisahkan program kasir menjadi dua berkas (2027). Isi berkas pajak.py dan kasir.py berturut-turut sebagai berikut.

```
# pajak.py
TARIF = 0.11

def hitung_pajak(harga):
    return round(harga * TARIF)
```

```
# kasir.py
from pajak import hitung_pajak

def total_bayar(harga):
    return harga + hitung_pajak(harga)

print(total_bayar(50000))
```

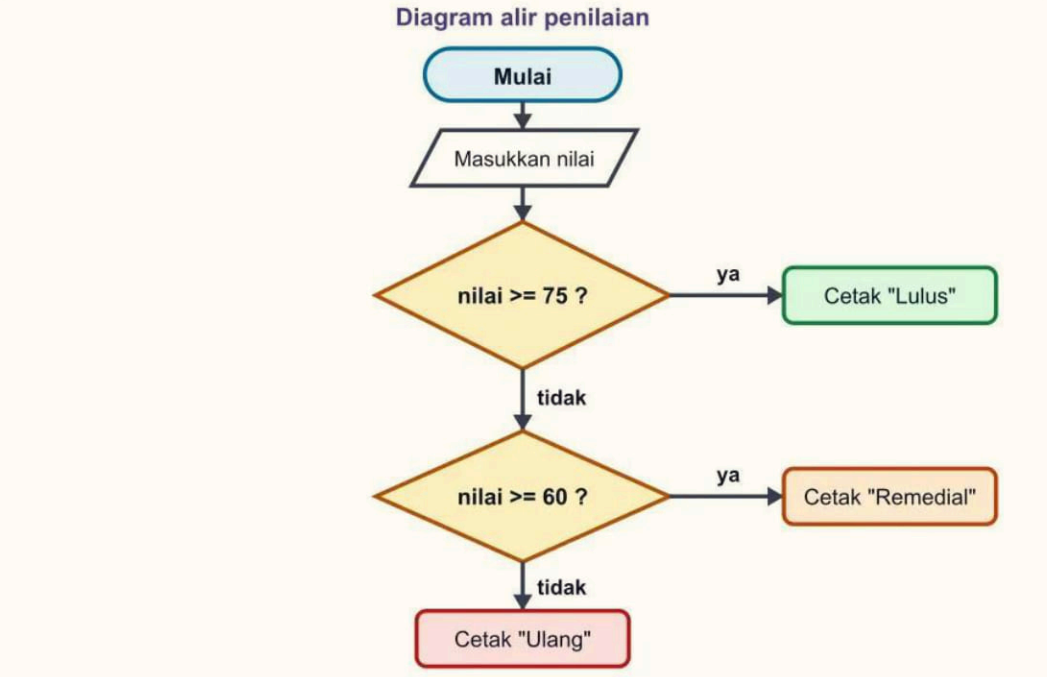

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Keluaran program kasir.py adalah 55500
☐ B. Bila tarif pajak berubah, cukup mengubah nilai TARIF di pajak.py
☐ C. kasir.py harus menyalin isi hitung_pajak agar dapat memakainya
☐ D. hitung_pajak dapat diuji sendiri tanpa menjalankan kasir.py
☐ E. Perintah import membuat semua variabel kasir.py menjadi global di pajak.py

18. Sebuah sekolah (2027) menyusun diagram alir untuk program penilaian kuis, seperti pada gambar. Setelah mencetak salah satu hasil, program selesai. Program dijalankan empat kali berturut-turut dengan nilai masukan 75, 74, 60, dan 59.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program untuk keempat nilai itu, berurutan, adalah ....



A. Lulus, Remedial, Remedial, Ulang
B. Lulus, Lulus, Remedial, Ulang
C. Lulus, Remedial, Ulang, Ulang
D. Remedial, Remedial, Remedial, Ulang
E. Lulus, Remedial, Remedial, Remedial

19. Perhatikan potongan program Python berikut.

```
def sapa(nama):
    print("Halo", nama)

h = sapa("Dewi")
print(h)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. Halo Dewi lalu Dewi
B. Halo Dewi lalu Halo Dewi
C. Halo Dewi lalu None
D. None saja
E. Halo Dewi lalu 0

20. Perhatikan potongan program Python berikut.

```
hasil = 0
for x in range(1, 10):
    if x % 3 == 0:
        continue
    if x > 7:
        break
    hasil += x
print(hasil)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Nilai hasil di akhir program adalah 19
☐ B. Nilai hasil di akhir program adalah 28 karena semua x dari 1 sampai 7 ikut dijumlahkan
☐ C. continue melewati x = 3 dan x = 6, sehingga keduanya tidak dijumlahkan
☐ D. break menghentikan perulangan saat x = 8, sehingga x = 9 tidak diproses
☐ E. break dijalankan pertama kali saat x = 3 karena 3 habis dibagi 3

21. Sebuah toko aplikasi (2027) menetapkan aturan diskon: belanja Rp200.000 atau lebih memperoleh diskon 10 persen; belanja Rp100.000 sampai kurang dari Rp200.000 memperoleh diskon 5 persen; di bawah itu tidak ada diskon. Seorang pemrogram menulis fungsi berikut lalu mencatat hasil uji pada tabel.

```
def diskon(belanja):
    if belanja > 200000:
        return 10
    elif belanja >= 100000:
        return 5
    else:
        return 0
```

Kasus uji	Belanja (Rp)	Diskon yang diharapkan	Hasil program
1	99.999	0	0
2	100.000	5	5
3	200.000	10	5
4	250.000	10	10

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kesimpulan yang paling tepat adalah ....

A. program sudah benar karena tiga dari empat hasil sesuai harapan
B. syarat kedua salah karena seharusnya memakai > dan bukan >= pada nilai 100.000
C. nilai batas 200.000 salah karena syarat pertama memakai >, padahal seharusnya >=
D. diskon 10 persen seharusnya baru diberikan mulai belanja 250.000, bukan 200.000
E. kasus uji 250.000 tidak diperlukan karena tidak mewakili nilai batas apa pun

22. Sebuah toko daring (2027) menghitung harga akhir setelah potongan dengan fungsi berikut.

```
def diskon(harga, persen=10):
    return harga - harga * persen // 100

def total(a, b, persen=5):
    return diskon(a, persen) + diskon(b)

print(total(1000, 2000))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 2850
B. 2700
C. 3000
D. 2750
E. 2800

23. Perhatikan potongan program Python berikut.

```
for k in range(20, 0, -6):
    print(k, end=" ")
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Keluaran program adalah 20 14 8 2.	○	○
Bila batas akhir diubah menjadi 2 (range(20, 2, -6)), angka 2 tetap tercetak.	○	○
Perulangan berjalan sebanyak 4 putaran.	○	○

24. Tim pengembang gim teka-teki (2027) memakai aturan skor di empat bagian: menu utama, layar level, papan peringkat, dan hadiah harian. Setiap kali aturan skor berubah, tim harus mengubah keempat bagian itu dan sering ada yang terlupa. Tim ingin merancang ulang

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

kode agar perubahan aturan skor mudah dan aman.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Rancangan modular yang paling efektif adalah ....

A. menyalin rumus skor ke setiap bagian agar masing-masing bagian dapat berdiri sendiri tanpa saling bergantung
B. menggabungkan seluruh kode gim dalam satu berkas besar supaya rumus skor tidak tersebar ke banyak berkas
C. menyimpan skor sebagai variabel global yang boleh diubah dari semua bagian gim kapan saja diperlukan
D. meminta tiap bagian menyalin hasil hitung dari menu utama secara manual setiap kali aturan berubah
E. menempatkan rumus skor dalam satu fungsi pada modul tersendiri yang dipanggil keempat bagian

25. Ruang server sebuah studio gim (2027) memakai fungsi berikut untuk memutuskan siapa yang boleh masuk.

```
def boleh_masuk(jam, petugas):
    if (jam >= 8 and jam < 17) or petugas:
        return True
    return False
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang hasil pemanggilan fungsi tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `boleh_masuk(8, False)` menghasilkan True
☐ B. `boleh_masuk(17, False)` menghasilkan True karena jam 17 masih termasuk jam kerja
☐ C. `boleh_masuk(20, True)` menghasilkan True karena petugas selalu boleh masuk
☐ D. `boleh_masuk(7, False)` menghasilkan True karena jam 7 mendekati jam kerja
☐ E. `boleh_masuk(16, False)` menghasilkan True

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 4.1 · Tipe data dan struktur data	1, 2, 3, 4, 5, 6, 7	___ / 7	Ulangi hlm. 172
Sub bab 4.2 · Struktur kontrol percabangan	10, 13, 18, 21, 25	___ / 5	Ulangi hlm. 186
Sub bab 4.3 · Struktur kontrol perulangan	11, 14, 15, 16, 20, 23	___ / 6	Ulangi hlm. 201
Sub bab 4.4 · Fungsi, prosedur, dan modularisasi	8, 9, 12, 17, 19, 22, 24	___ / 7	Ulangi hlm. 215

Skor 70 % atau lebih: lanjut ke bab berikutnya. Skor 50–69 %: ulangi sub bab pada tabel yang kurang dari separuh benar. Skor di bawah 50 %: baca ulang bab ini dari awal. (Angka ini anjuran belajar, bukan prediksi nilai TKA.)

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bab 5 · Pemrograman Berorientasi Objek

SMKPPLG-B5

Setelah mempelajari bab ini, kamu dapat:

- menerapkan konsep class, object, atribut, method, dan prinsip dasar OOP dalam rancangan program
- memilih tingkat akses dan enkapsulasi yang tepat untuk atribut dan method
- menerapkan relasi pewarisan antar-class dengan benar
- mengimplementasikan dan menelusuri polymorphism lewat method overriding dan referensi superclass

Peta bab

Sub bab	Judul
5.1	Class, object, dan prinsip dasar OOP
5.2	Enkapsulasi, access modifier, dan getter–setter
5.3	Pewarisan
5.4	Polymorphism

Bab 5 di LMS

Buka seluruh latihan bab ini di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-B5

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 5.1 · Class, object, dan prinsip dasar OOP

SMKPPLG-5.1

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menerapkan konsep class, object, atribut, dan method** serta **prinsip dasar pemrograman berorientasi objek (PBO atau OOP)** dalam merancang struktur program. Pada soal, bagian ini muncul dalam tiga bentuk. Pertama, **menelusuri kode** class sederhana dengan beberapa objek untuk menentukan keluarannya. Kedua, **merancang class** dari sebuah kebutuhan: apa yang menjadi atribut (data) dan apa yang menjadi method (tindakan). Ketiga, **mencocokkan sebuah rancangan dengan prinsip OOP** yang digunakannya (abstraksi, enkapsulasi, pewarisan, polimorfisme).

Sub bab ini adalah fondasi seluruh bab: tiga sub bab berikutnya (enkapsulasi, pewarisan, polimorfisme) membahas prinsip-prinsip yang di sini baru diperkenalkan. Kalau konsep class dan object belum kokoh, tiga bagian berikutnya akan terasa membingungkan, jadi luangkan waktu menjalankan kodenya sendiri.

Setelah mempelajari sub bab ini, kamu dapat:

- membedakan class dan object serta menjelaskan hubungan keduanya;
- menulis class dengan `__init__`, `self`, atribut, dan method, lalu membuat beberapa object darinya;
- menelusuri keluaran program yang memakai beberapa object dengan keadaan berbeda;
- menentukan mana yang menjadi atribut dan mana yang menjadi method dalam sebuah rancangan;
- menyebut empat prinsip OOP dan mencocokkan kasus dengan prinsip yang tepat.

◆ KONSEP INTI

Pada program terstruktur (Bab 4), data dan fungsi dipisahkan: kamu menyimpan data di variabel, lalu menulis fungsi yang mengolahnya. Untuk program kecil itu cukup, tetapi untuk gim dengan puluhan karakter, setiap karakter dengan nyawa, posisi, dan kemampuan sendiri, cara itu cepat berantakan. Kamu butuh cara untuk **menyatukan data dengan tindakan yang berkaitan dengannya**. Itulah ide **pemrograman berorientasi objek**: program dibangun dari **objek**, yaitu benda-benda dalam program yang masing-masing menyimpan datanya sendiri dan tahu cara melakukan tindakannya sendiri.

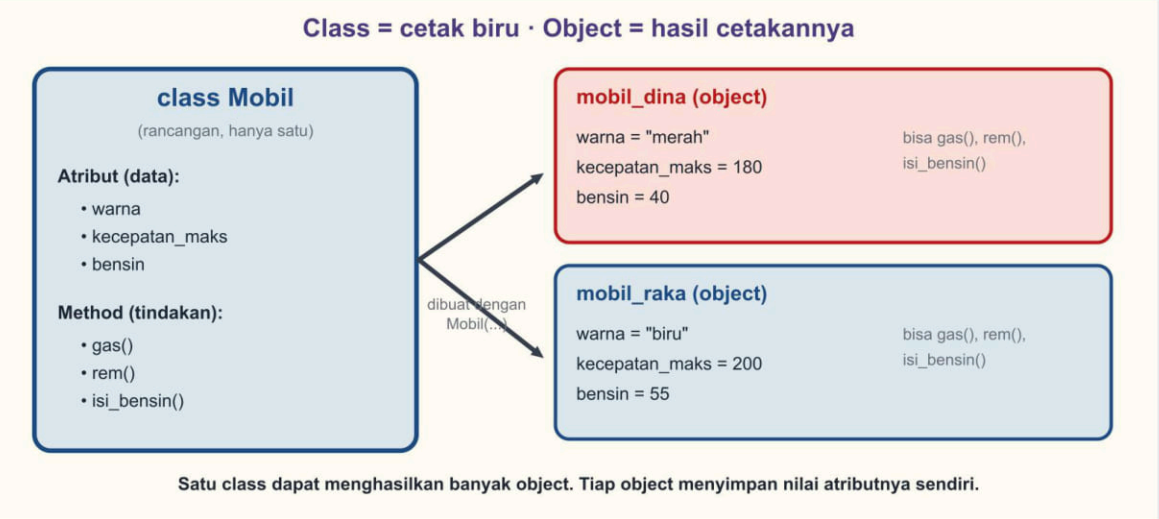
1. Class dan object

Class adalah **cetak biru** atau rancangan: ia mendeskripsikan data apa yang dimiliki dan tindakan apa yang bisa dilakukan oleh sebuah jenis benda. **Object** (objek) adalah **hasil nyata** yang dibuat dari class itu. Hubungannya seperti cetakan kue dan kuenya, atau denah rumah

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

dan rumah yang dibangun darinya (Gambar 5.1).

- Satu class dapat menghasilkan **banyak objek**.
- Tiap objek menyimpan **nilai atribut sendiri** (mobil Dina merah dan mobil Raka biru).
- Membuat objek dari class disebut **membuat instans** (*instansiasi*).



Gambar 5.1 Class Mobil sebagai cetak biru dan dua objek hasilnya. Sumber: digambar oleh Tim Next Level Study Indonesia.

Kesalahan konsep yang sering diuji: **class dan object tertukar.** `Mobil` adalah class; `mobil_dina` dan `mobil_raka` adalah object-nya. Objek bukan "class turunan"; pengertian turunan menyangkut pewarisan antar-class (sub bab 5.3).

Sebagai perbandingan, bayangkan gim dengan 30 musuh. Dengan pendekatan prosedural, kamu mungkin membuat tiga puluh variabel nyawa, tiga puluh variabel posisi, dan fungsi yang menerima banyak parameter agar tahu musuh mana yang dimaksud. Dengan OOP, kamu membuat satu class `Musuh` dan tiga puluh objeknya. Data dan tindakan sudah menyatu, sehingga memanggil `musuh_7.kena_serangan(10)` jelas ditujukan kepada musuh ketujuh. Itulah yang membuat program berskala besar lebih rapi, lebih mudah diubah, dan lebih mudah dikerjakan bersama.

2. Atribut dan method

Sebuah class memiliki dua jenis anggota.

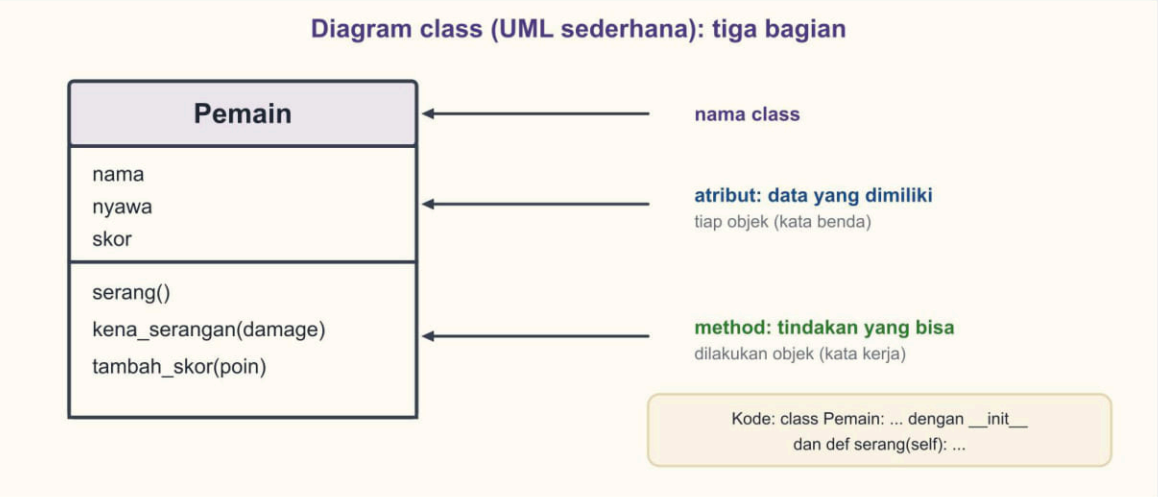
- **Atribut** adalah **data** yang dimiliki objek: `warna`, `kecepatan_maks`, `bensin`, `nyawa`, `saldo`. Atribut biasanya **kata benda** atau sifat.
- **Method** adalah **tindakan** yang dapat dilakukan objek: `gas()`, `rem()`, `serang()`, `setor()`. Method adalah fungsi yang berada di dalam class, biasanya **kata kerja**.

Aturan praktis memilah: tanyakan "apa yang **dimiliki**" (atribut) dan "apa yang **dilakukan**" (method). Jadi pada tiket pentas, nomor kursi, harga, dan status adalah **atribut**, sedangkan memesan dan membatalkan adalah **method**. "Status" adalah keadaan, jadi atribut, bukan tindakan. Jebakan yang sering muncul: menjadikan tindakan sebagai atribut (`pesan`) atau menjadikan data sebagai method (`status()`), atau memecah satu jenis benda menjadi

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

beberapa class padahal cukup satu class dengan banyak objek.

Gambar 5.2 memperlihatkan **diagram class** (UML sederhana): sebuah kotak tiga bagian berisi nama class, daftar atribut, dan daftar method.



Gambar 5.2 Diagram class: nama, atribut, dan method. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Menulis class di Python: `__init__` dan `self`

Berikut class `Dompet` dengan dua atribut dan dua method.

```
class Dompet:
    def __init__(self, pemilik, saldo=0):
        self.pemilik = pemilik
        self.saldo = saldo

    def setor(self, jumlah):
        self.saldo += jumlah

    def tarik(self, jumlah):
        if jumlah <= self.saldo:
            self.saldo -= jumlah

d1 = Dompet("Ayu", 50000)
d2 = Dompet("Budi")
d1.setor(20000)
d2.setor(30000)
d2.tarik(50000)
d1.tarik(10000)
print(d1.saldo, d2.saldo)
```

Keluarannya `60000 30000`. Bagian-bagiannya:

- **`class Dompet:`** mendefinisikan class. Nama class lazimnya diawali huruf besar.
- **`__init__`** (dua garis bawah di tiap sisi) adalah method khusus yang **dijalankan otomatis saat objek dibuat**. Fungsinya memberi nilai awal pada atribut. Disebut **konstruktor**.
- **`self`** adalah **objek yang sedang dipakai**. Setiap method menerimanya sebagai parameter pertama. `self.saldo` berarti "atribut `saldo` milik objek ini". Saat memanggil `d1.setor(20000)`, Python mengisi `self` dengan `d1` otomatis; kamu tidak menulisnya di dalam kurung.
- **`d1 = Dompet("Ayu", 50000)`** membuat objek: argumen `"Ayu"` dan `50000` masuk ke `pemilik` dan `saldo`. `d2 = Dompet("Budi")` memakai nilai bawaan `saldo=0`.
- **`d1.setor(20000)`** memanggil method pada objek `d1`. **`d1.saldo`** membaca atributnya.

Telusuri: `d1` mulai 50000, disetor 20000 → 70000, ditarik 10000 → **60000**. `d2` mulai 0, disetor 30000 → 30000; penarikan 50000 melebihi saldo sehingga **tidak terjadi**, tetap **30000**.

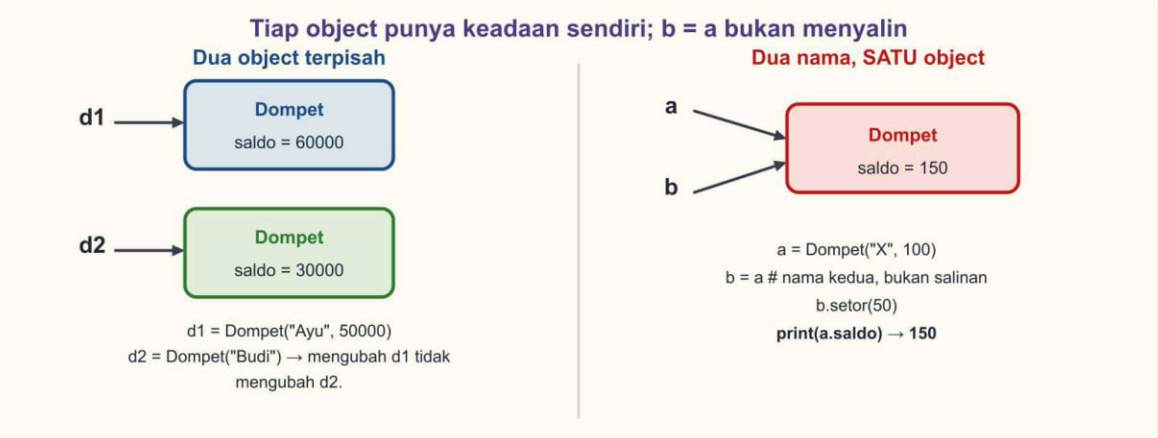
Satu hal lagi yang perlu diketahui: sebuah method dapat **memanggil method lain pada objek yang sama** lewat `self`, dan dapat **mengembalikan nilai** seperti fungsi biasa. Misalnya method `total()` pada class `Pesanan` (contoh di bawah) mengembalikan jumlah harga item tanpa mengubah keadaan objek, sedangkan `bayar()` mengubah atribut `status` tanpa mengembalikan apa pun. Method yang mengubah keadaan objek lazimnya tidak memakai `return`, dan method yang hanya membaca keadaan lazimnya memakai `return`. Pengamatan ini berguna ketika kamu membaca kode dan harus menentukan apa yang berubah setelah sebuah pemanggilan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bila sebuah objek dicetak langsung dengan `print(d1)`, Python hanya menampilkan keterangan singkat berisi nama class dan alamat memori, bukan isi atributnya. Untuk melihat isinya, cetak atributnya satu per satu (`print(d1.saldo)`) atau definisikan method `__str__` pada class. Soal TKA umumnya mencetak atribut atau hasil method, bukan objek utuh.

4. Tiap objek punya keadaan sendiri

Setiap objek membawa **salinan atributnya sendiri**. Mengubah `d1` tidak mengubah `d2` (Gambar 5.3). Inilah kekuatan OOP: kamu dapat membuat 30 objek `Prajurit`, masing-masing dengan nyawanya sendiri, dari satu class.



Gambar 5.3 Dua objek terpisah memiliki keadaan sendiri; dua nama untuk satu objek mengubah objek yang sama. Sumber: digambar oleh Tim Next Level Study Indonesia.

Ada satu jebakan: **`b = a` tidak menyalin objek**, melainkan memberi **nama kedua** untuk objek yang sama.

```
a = Dompet("Xena", 100)
b = a
b.setor(50)
print(a.saldo)
```

Keluarannya `150`: `a` dan `b` menunjuk satu objek, sehingga setoran lewat `b` terlihat lewat `a`. Untuk dua objek terpisah, panggil `Dompet(...)` dua kali.

5. Atribut instans dan atribut class

Atribut yang dibuat di `__init__` dengan `self.` adalah **atribut instans**: tiap objek punya sendiri. Atribut yang ditulis langsung di dalam class (di luar method) adalah **atribut class**: **dipakai bersama** semua objek.

```
class Pemain:
    jumlah = 0
    def __init__(self, nama):
        self.nama = nama
        Pemain.jumlah += 1

p1 = Pemain("Ani")
p2 = Pemain("Beni")
p3 = Pemain("Cici")
```

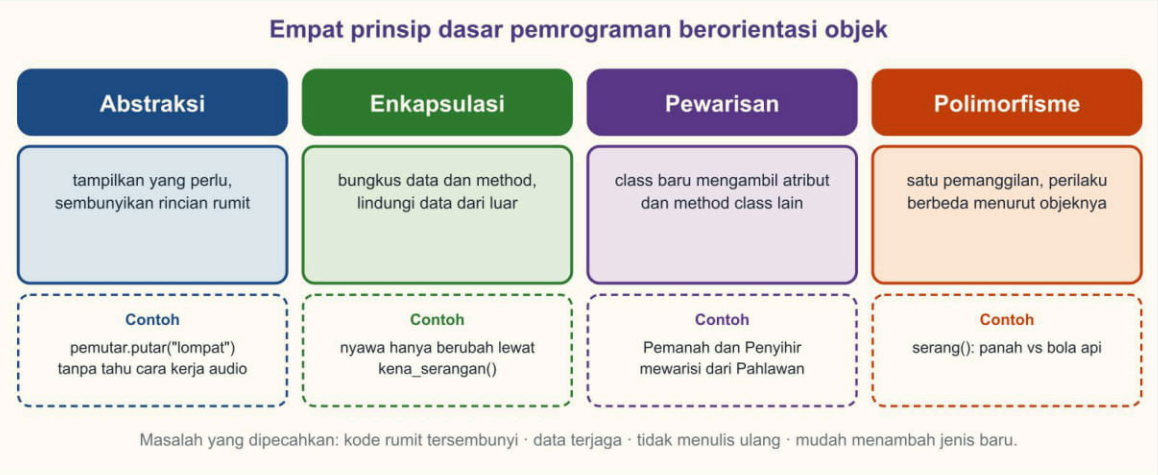

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
print(Pemain.jumlah, p1.jumlah, p2.nama)
```

Keluarannya `3 3 Beni`. `jumlah` adalah atribut class: bernilai 3 sesudah tiga objek dibuat, dan dapat dibaca lewat nama class maupun lewat objek. `nama` adalah atribut instans, jadi `p2.nama` adalah `Beni`. Atribut class cocok untuk nilai yang sama bagi semua objek, misalnya hitungan jumlah objek atau tarif umum.

6. Empat prinsip dasar OOP

OOP mengajarkan empat prinsip utama (Gambar 5.4). Masing-masing dibahas lebih dalam pada sub bab berikutnya; di sini yang penting adalah mengenalinya dan masalah yang dipecahkan.



Gambar 5.4 Empat prinsip dasar OOP, masalah yang dipecahkan, dan contohnya. Sumber: digambar oleh Tim Next Level Study Indonesia.

1. **Abstraksi**: menampilkan hal yang perlu dan **menyembunyikan rincian rumit** di balik antarmuka sederhana. Programmer lain cukup menulis `pemutar.putar("lompat")` tanpa tahu cara kerja berkas audio di dalam class. Cirinya: **banyak detail disembunyikan di balik method yang sederhana**.
2. **Enkapsulasi**: **membungkus data dan method dalam satu class** serta melindungi data dari perubahan sembarangan dari luar. Nyawa pahlawan hanya berubah lewat `kena_serangan()` yang menjaganya tidak negatif (sub bab 5.2).
3. **Pewarisan**: class baru **mengambil atribut dan method** dari class yang sudah ada tanpa menulis ulang. `Pemanah` dan `Penyihir` mewarisi `nama` dan `nyawa` dari `Pahlawan` (sub bab 5.3).
4. **Polimorfisme**: **pemanggilan yang sama menghasilkan perilaku yang berbeda** menurut objek yang menerimanya. `serang()` pada `Pemanah` menembakkan panah, pada `Penyihir` mengeluarkan bola api (sub bab 5.4).

Cara mencocokkan kasus dengan prinsip: tanyakan apa yang dilakukan rancangan itu. Menyembunyikan rincian rumit → abstraksi. Melindungi atau membatasi akses ke data → enkapsulasi. Class baru memakai bagian class lama → pewarisan. Satu pemanggilan, banyak perilaku → polimorfisme. Dan membuat banyak objek dari satu class **bukan pewarisan**; itu **instansiasi**. Satu rancangan boleh memuat lebih dari satu prinsip sekaligus; soal biasanya

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

meminta kamu memasangkan **satu bagian rancangan** dengan **satu prinsip**, jadi baca tiap potongan rancangan secara terpisah dan jangan mencampuradukkannya.

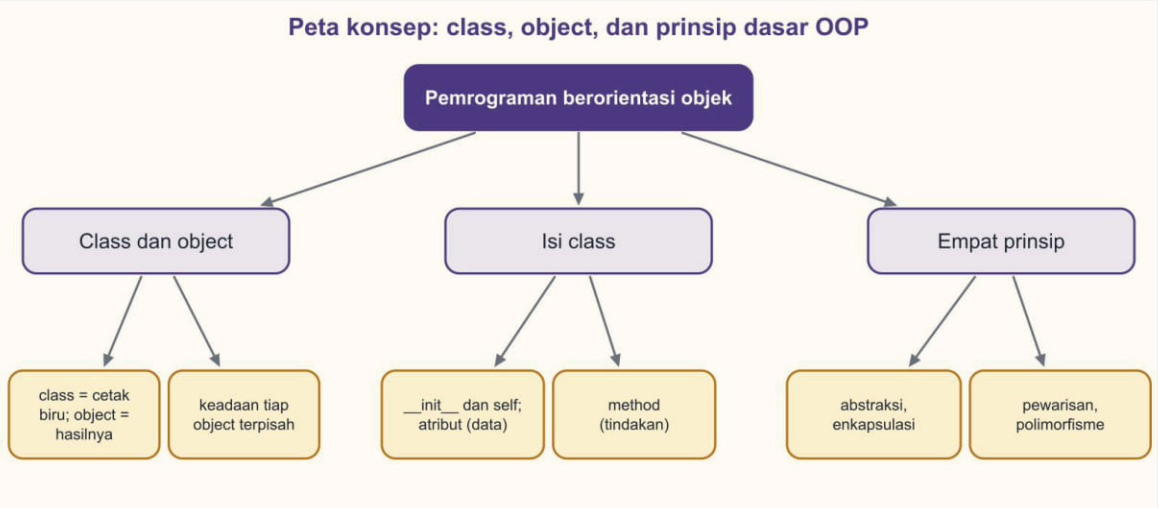
7. Merancang class dari kebutuhan

Saat diminta merancang class dari sebuah deskripsi, ikuti langkah ini.

1. **Tentukan jenis bendanya** (satu class): `Tiket`, `Pesanan`, `Karakter`.
2. **Daftar data yang melekat** pada tiap benda → **atribut** (nomor kursi, harga, status).
3. **Daftar tindakan yang dilakukan atau dialami benda itu → method** (pesan, batal).
4. **Tentukan nilai awal** yang dikirim saat objek dibuat → parameter `__init__`.
5. **Satu objek untuk setiap benda nyata**: satu objek per kursi, per pemain, per buku.

Sebagai contoh, kebutuhan "toko sepatu menyimpan setiap sepatu dengan merek, ukuran, harga, dan jumlah stok; sepatu dapat dijual dan stoknya ditambah" menghasilkan class `Sepatu` dengan atribut `merek`, `ukuran`, `harga`, `stok` dan method `jual()` serta `tambah_stok()`. Setiap pasang sepatu yang berbeda menjadi satu objek. Sebuah daftar yang berisi semua objek `Sepatu` (list) menyimpan seluruh inventaris toko.

Sebuah rancangan yang baik tidak membuat class berbeda untuk benda yang hanya berbeda nilainya (misalnya `MobilMerah` dan `MobilBiru`), tidak menaruh data berubah di method, dan tidak menyimpan "daftar semua kursi" di dalam satu objek tiket tunggal.



Gambar 5.5 Peta konsep class, object, dan prinsip dasar OOP. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Dua objek, satu class

Tentukan keluaran program berikut. *(Kasus fiktif.)*

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
class Kartu:
    def __init__(self, pemilik, poin=0):
        self.pemilik = pemilik
        self.poin = poin

    def belanja(self, jumlah):
        self.poin += jumlah // 10000

k1 = Kartu("Tari", 5)
k2 = Kartu("Umar")
k1.belanja(30000)
k2.belanja(45000)
k2.belanja(9999)
print(k1.poin, k2.poin)
```

1. `k1` mulai dengan 5 poin; `k2` dengan 0 (nilai bawaan).
2. `k1.belanja(30000)`: 30000 // 10000 = 3 → `k1.poin` = 8.
3. `k2.belanja(45000)`: 45000 // 10000 = 4 → `k2.poin` = 4.
4. `k2.belanja(9999)`: 9999 // 10000 = 0 → tetap 4.
5. Cetak `8 4`.

Jawaban: `8 4`. Tiap objek mengubah atributnya sendiri.

Contoh 2 — Merancang class dari deskripsi

Warung makan membuat aplikasi pesanan (2027). Setiap pesanan memiliki nomor pesanan, nama pelanggan, daftar harga item, dan status (baru, lunas, batal). Pesanan dapat ditambah item, dibayar, atau dibatalkan. Rancang class `Pesanan` dan tunjukkan penggunaannya untuk dua pelanggan. *(Kasus fiktif.)*

1. Jenis benda: pesanan → class `Pesanan`.
2. Data yang melekat: `nomor`, `pelanggan`, `item` (daftar harga), `status` → **atribut**.
3. Tindakan: `tambah_item(harga)`, `total()`, `bayar()`, `batal()` → **method**.
4. Nilai awal saat dibuat: nomor dan nama pelanggan; `item` mulai kosong; status "baru".

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
class Pesanan:
    def __init__(self, nomor, pelanggan):
        self.nomor = nomor
        self.pelanggan = pelanggan
        self.item = []
        self.status = "baru"

    def tambah_item(self, harga):
        self.item.append(harga)

    def total(self):
        return sum(self.item)

    def bayar(self):
        self.status = "lunas"

p1 = Pesanan(1, "Wati")
p2 = Pesanan(2, "Yoga")
p1.tambah_item(15000)
p1.tambah_item(8000)
p2.tambah_item(20000)
p1.bayar()
print(p1.total(), p1.status, p2.total(), p2.status)
```

1. Telusuri: `p1.item` = [15000, 8000] → total 23000; `p1.bayar()` mengubah statusnya menjadi "lunas". `p2.item` = [20000] → total 20000; status tetap "baru".

Jawaban: `23000 lunas 20000 baru`. Satu class `Pesanan`, dua objek dengan keadaan berbeda; status adalah atribut, `bayar()` adalah method.

Contoh 3 — Mencocokkan kasus dengan prinsip OOP

Cocokkan tiap kasus dengan prinsip OOP yang paling tepat. *(Kasus fiktif.)*

- (a) Class `Kamera` menyediakan method `foto()`; programmer cukup memanggilnya tanpa tahu cara sensor, fokus, dan penyimpanan bekerja.
- (b) Atribut `saldo` hanya dapat berubah lewat `setor()` dan `tarik()` yang memeriksa aturan.
- (c) Class `Elang` dan `Merpati` memakai atribut `sayap` dan method `terbang()` dari class `Burung` tanpa menulis ulang.
- (d) Daftar berisi objek `Kucing`, `Anjing`, dan `Sapi`; saat `bersuara()` dipanggil pada tiap objek, suaranya berbeda.

1. (a) Rincian rumit disembunyikan di balik method sederhana → **abstraksi**.
2. (b) Data dilindungi, hanya diubah lewat method dengan aturan → **enkapsulasi**.
3. (c) Class baru memakai atribut dan method class lain → **pewarisan**.
4. (d) Satu pemanggilan, perilaku berbeda → **polimorfisme**.

Jawaban: (a) abstraksi, (b) enkapsulasi, (c) pewarisan, (d) polimorfisme.

Contoh 4 — Atribut class dan instans

Tentukan keluaran program berikut. *(Kasus fiktif.)*

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
class Siswa:
    sekolah = "SMK Mandiri"
    jumlah = 0

    def __init__(self, nama):
        self.nama = nama
        Siswa.jumlah += 1

a = Siswa("Rudi")
b = Siswa("Sari")
c = Siswa("Tono")
print(c.nama, a.sekolah, Siswa.jumlah)
```

1. `sekolah` dan `jumlah` adalah atribut class: dipakai bersama semua objek.
2. Setiap kali `Siswa(...)` dibuat, `__init__` menambah `Siswa.jumlah` 1: setelah tiga objek, `jumlah` = 3.
3. `c.nama` adalah atribut instans: `Tono`.
4. `a.sekolah` dibaca dari class: `SMK Mandiri`.

Jawaban: `Tono SMK Mandiri 3`.

Contoh 5 — Dua nama untuk satu objek

Tentukan keluaran program berikut dan jelaskan mengapa. *(Kasus fiktif.)*

```
class Skor:
    def __init__(self, nilai=0):
        self.nilai = nilai

a = Skor(10)
b = a
c = Skor(10)
b.nilai += 5
print(a.nilai, c.nilai)
```

1. `a = Skor(10)` membuat satu objek; `b = a` membuat **nama kedua** untuk objek yang sama.
2. `c = Skor(10)` membuat **objek lain** yang terpisah.
3. `b.nilai += 5` mengubah objek yang ditunjuk `a` dan `b`: nilainya 15.
4. `c` tidak ikut berubah: 10.

Jawaban: `15 10`. Untuk menyalin, buat objek baru; `b = a` tidak menyalin.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

❖ PETA KONSEP

Gambar 5.5 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Class = cetak biru; object = hasilnya.** Satu class, banyak object; tiap object punya keadaan sendiri.
- **Atribut = data (kata benda); method = tindakan (kata kerja).**
- **`__init__`** dijalankan saat objek dibuat; **`self`** adalah objek yang sedang dipakai.
- **`b = a` bukan menyalin objek**; keduanya menunjuk objek yang sama.
- **Empat prinsip**: abstraksi (sembunyikan rincian), enkapsulasi (lindungi data), pewarisan (pakai bagian class lain), polimorfisme (satu pemanggilan, banyak perilaku).
- **Membuat banyak objek dari satu class adalah instansiasi, bukan pewarisan.**

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menjawab soal class dan object:

1. **Tulis objek yang dibuat dan nilai awalnya** untuk tiap objek.
2. **Telusuri pemanggilan method berurutan**, mencatat atribut objek yang dipengaruhi (bukan objek lain).
3. **Pisahkan atribut instans dan atribut class**; yang kedua dibagi semua objek.

Langkah menjawab soal merancang class dan mencocokkan prinsip:

1. **Tanyakan "apa yang dimiliki" dan "apa yang dilakukan"** untuk memilah atribut dan method.
2. **Untuk prinsip**, tanyakan apa yang dilakukan rancangannya: menyembunyikan, melindungi, mewarisi, atau menjawab berbeda.

Jebakan yang sering dipasang:

- **Class dan object tertukar** (menyebut objek sebagai class turunan).
- **Atribut dan method tertukar** (`warna` disebut method, `gas()` disebut atribut).
- **Membuat 30 objek dikira pewarisan**; itu instansiasi.
- **Setiap warna atau ukuran mendapat class sendiri** padahal cukup satu class dengan atribut.
- **Dua objek dikira berbagi atribut** padahal masing-masing punya sendiri.
- **`b = a` dikira menyalin.**
- **`self` terlupa** pada `self.saldo` atau pada parameter method.
- **Abstraksi dan enkapsulasi tertukar**: menyembunyikan rincian rumit adalah abstraksi; melindungi data dari perubahan langsung adalah enkapsulasi.
- **Atribut yang berubah dimasukkan ke dalam daftar method** (status disebut `status()`).

> Contoh salah tafsir: sebuah rancangan menyebut "class `PemutarSuara` berisi kode rumit, dan programmer lain hanya menulis `putar("menang")`", lalu seorang siswa menyimpulkan itu pewarisan karena "dipakai banyak bagian gim". Pewarisan menyangkut class baru yang mengambil isi class lama. Menyembunyikan kode rumit di balik method sederhana adalah abstraksi.

• BEDAH SOAL PREDIKSI

Soal berikut menyajikan rancangan class dari sebuah kebutuhan tiket. Perhatikan cara memisahkan data yang **dimiliki** setiap tiket (atribut) dari tindakan yang **dilakukan** pada tiket (method) (bagian 2 dan 7).

►. Tim PKL membuat aplikasi pemesanan tiket pentas seni sekolah (2027). Kebutuhannya: setiap tiket memiliki nomor kursi, harga, dan status (tersedia atau terjual); tiket dapat dipesan dan dapat dibatalkan. Tim akan membuat class `Tiket`, lalu satu objek dibuat untuk setiap kursi.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Rancangan class Tiket yang paling tepat adalah ....

A. atribut pesan, batal, status; method nomor_kursi() dan harga()
B. atribut nomor_kursi dan harga; method status(), pesan(), batal()
C. atribut kursi_tersedia, kursi_terjual; method pesan(), batal()
D. atribut nomor_kursi, harga, pesan; method status() dan batal()
E. atribut nomor_kursi, harga, status; method pesan() dan batal()

Pembahasan **Kunci: E**

Konsep: Atribut adalah data yang dimiliki setiap objek; method adalah tindakan yang dapat dilakukan objek. Objek-objek tiket dibuat dari class, bukan disimpan sebagai atribut class itu.

Langkah:
1. Data: nomor kursi, harga, status → atribut.
2. Tindakan: memesan, membatalkan → method pesan() dan batal().

Analisis setiap opsi:
• **A.** ✗ Tindakan dijadikan atribut dan data dijadikan method; terbalik.
• **B.** ✗ Jebakan: status adalah data milik tiket yang berubah karena pesan() atau batal(), jadi lebih tepat sebagai atribut.
• **C.** ✗ Jebakan: atribut ini menggambarkan seluruh gedung, bukan satu tiket; harga dan status tiket juga tidak ada.
• **D.** ✗ pesan adalah tindakan, bukan data; status adalah data, bukan tindakan.
• **E.** ✓ Data menjadi atribut dan tindakan menjadi method.

Jawaban: E.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk soal kode, tulis nilai atribut tiap objek setelah setiap pemanggilan. Untuk soal rancangan, tulis dulu apa yang dimiliki dan apa yang dilakukan.

1. Tim gim balap *Sirkuit Kota* (2027) merancang class `Mobil` dengan data `warna`, `kecepatan_maks`, dan `bensin`, serta tindakan `gas()`, `rem()`, dan `isi_bensin()`. Saat balapan dimulai, program membuat `mobil_dina` berwarna merah dengan kecepatan maksimum 180 dan `mobil_raka` berwarna biru dengan kecepatan maksimum 200.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang rancangan tersebut adalah ....

A. mobil_dina dan mobil_raka adalah objek class Mobil dengan nilai atribut berbeda
B. warna dan kecepatan_maks adalah method karena setiap mobil memiliki keduanya
C. gas() adalah atribut karena nilai kecepatan mobil berubah ketika gas() dipanggil
D. Mobil adalah objek, sedangkan mobil_dina dan mobil_raka adalah class turunannya
E. Setiap mobil memerlukan class sendiri karena warna dan kecepatannya berbeda-beda

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Aplikasi perpustakaan kelas (2027) memakai class berikut untuk mencatat stok buku.

```
class Buku:
    def __init__(self, judul, stok):
        self.judul = judul
        self.stok = stok

    def pinjam(self):
        if self.stok > 0:
            self.stok -= 1

b1 = Buku("Dasar Pemrograman", 2)
b2 = Buku("Jaringan Dasar", 1)
b1.pinjam()
b2.pinjam()
b2.pinjam()
print(b1.stok, b2.stok)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. `1 -1`
B. `0 0`
C. `1 0`
D. `2 1`
E. `0 -1`

3. Tim gim RPG *Legenda Nusantara* (2027) menulis catatan rancangan berikut.

• Atribut nyawa pahlawan tidak dapat diubah langsung dari luar; nyawa hanya berkurang lewat method `kena_serangan()` yang memastikan nyawa tidak negatif.

• Class `Pemanah` dan `Penyihir` mewarisi atribut nama dan nyawa dari class `Pahlawan`.

• Saat `serang()` dipanggil, Pemanah menembakkan panah, sedangkan Penyihir mengeluarkan bola api.

• Program membuat 30 objek `Prajurit` dari class Prajurit untuk pasukan musuh.

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pasangan rancangan dan prinsip PBO yang tepat adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Pembuatan 30 objek Prajurit dari class Prajurit adalah contoh pewarisan.
☐ B. Nyawa yang hanya dapat diubah lewat kena_serangan() adalah contoh enkapsulasi.
☐ C. Pemanah dan Penyihir yang mewarisi nama dan nyawa dari Pahlawan adalah contoh pewarisan.
☐ D. Method kena_serangan() yang memastikan nyawa tidak negatif adalah contoh polimorfisme.
☐ E. serang() yang berperilaku berbeda pada Pemanah dan Penyihir adalah contoh polimorfisme.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Gim edukasi *Kebun Ceria* (2027) memiliki fungsi `gerak(hewan)` dan `suara(hewan)` yang masing-masing berisi percabangan if–elif panjang menurut jenis hewan: gajah, burung, ikan, dan seterusnya. Setiap bulan tim menambah dua sampai tiga jenis hewan baru. Setiap kali itu kedua fungsi harus diubah, dan beberapa kali perubahan tersebut merusak gerak hewan lama.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Rancangan yang paling tepat untuk mengatasi masalah tersebut adalah ....

A. Superclass Hewan berisi gerak() dan suara(); tiap jenis hewan menjadi subclass yang menulis ulang
B. Setiap kali ada jenis hewan baru, tim menambahkan cabang elif baru di dalam fungsi gerak() dan suara()
C. Satu class Hewan menyimpan atribut jenis, lalu percabangan if–elif dipindah ke dalam method gerak()
D. Tim membuat fungsi gerak_gajah(), gerak_burung(), dan seterusnya, satu fungsi untuk setiap jenis hewan
E. Seluruh atribut hewan dijadikan variabel global agar fungsi gerak() dan suara() mudah membacanya

5. Tim gim *Rimba Bersuara* (2027) membuat class `PemutarSuara`. Di dalamnya ada kode rumit untuk membaca berkas audio, mengatur volume, dan memilih kanal suara. Programmer lain cukup menulis `pemutar.putar("lompat")` atau `pemutar.putar("menang")` tanpa perlu tahu cara kerja di dalam class itu.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Prinsip PBO yang paling tepat menggambarkan rancangan tersebut adalah ....

A. Pewarisan, karena programmer lain mewarisi kemampuan memutar suara
B. Abstraksi, karena detail rumit disembunyikan di balik method sederhana
C. Polimorfisme, karena putar() dapat memutar banyak nama suara berbeda
D. Instansiasi, karena objek pemutar dibuat dari class PemutarSuara
E. Pewarisan, karena class itu dipakai oleh banyak bagian gim sekaligus

Kunci singkat: 1. A 2. C 3. B, C, E 4. A 5. B

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Apa perbedaan class dan object, dan mengapa membuat 30 objek dari satu class bukan pewarisan?
- Mengapa `b = a` tidak menghasilkan dua objek terpisah?
- **Yang harus kuingat:** class = cetak biru, object = hasilnya; atribut = data, method = tindakan; `__init__` memberi nilai awal dan `self` adalah objek yang dipakai; empat prinsip: abstraksi, enkapsulasi, pewarisan, polimorfisme.

Latihan interaktif di LMS

Sub bab 5.1: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-5.1

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 5.2 · Enkapsulasi, access modifier, dan getter–setter

SMKPPLG-5.2

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menerapkan ketepatan access modifier dan enkapsulasi pada atribut dan method**, termasuk getter dan setter, dengan tingkat berpikir **mengklasifikasi**: kamu memilah mana yang seharusnya dapat diakses dari luar class, mana yang harus dilindungi, dan apa akibat dari pilihan itu. Pada soal, kamu biasanya melihat sebuah class dengan atribut privat (diawali dua garis bawah), method pembaca dan pengubah nilai, lalu beberapa baris yang memakai objeknya. Kamu menentukan **apa hasil pemanggilan itu**, **mana pernyataan yang benar**, atau **bagaimana rancangan class yang tepat** untuk sebuah aturan bisnis.

Sub bab ini paling sering memakai kata-kata **private**, **getter**, **setter**, dan **validasi**. Setiap kali soal memberi aturan seperti "nilai tidak boleh negatif" atau "hanya berkurang lewat pembayaran", ia sedang menyuruhmu berpikir tentang enkapsulasi: data harus dilindungi supaya aturan itu tidak bisa dilanggar.

Setelah mempelajari sub bab ini, kamu dapat:

- menjelaskan enkapsulasi dan alasan data dilindungi;
- membedakan public, protected, dan private, serta cara menuliskannya di Python;
- menelusuri class dengan atribut privat, getter, dan setter berikut hasil pemanggilannya;
- menulis setter yang memeriksa (memvalidasi) nilai sebelum menyimpannya;
- menentukan tingkat akses yang tepat untuk atribut dan method pada sebuah rancangan.

NEXT LEVEL STUDY

◆ KONSEP INTI

Bayangkan sebuah mesin ATM. Kamu tidak bisa membuka mesin dan mengubah saldo rekeningmu langsung. Kamu hanya boleh memakai tombol dan layar: memasukkan PIN, memilih tarik tunai, memasukkan jumlah. Setiap permintaan diperiksa: PIN benar? saldo cukup? batas harian terpenuhi? Aturan itu tidak mungkin dilewati karena data saldo berada di balik "dinding", dan satu-satunya pintu adalah tombol yang menjalankan pemeriksaan. Inilah **enkapsulasi** dalam pemrograman.

1. Mengapa data perlu dilindungi

Enkapsulasi berarti **membungkus data dan method yang mengolahnya dalam satu class**, serta **melindungi data itu dari perubahan sembarangan dari luar**. Bila atribut dibuka sepenuhnya, siapa pun dapat mengisinya dengan nilai apa saja, termasuk nilai yang tidak masuk akal.

```
class AkunTerbuka:
    def __init__(self, nama):
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
        self.nama = nama
        self.koin = 0

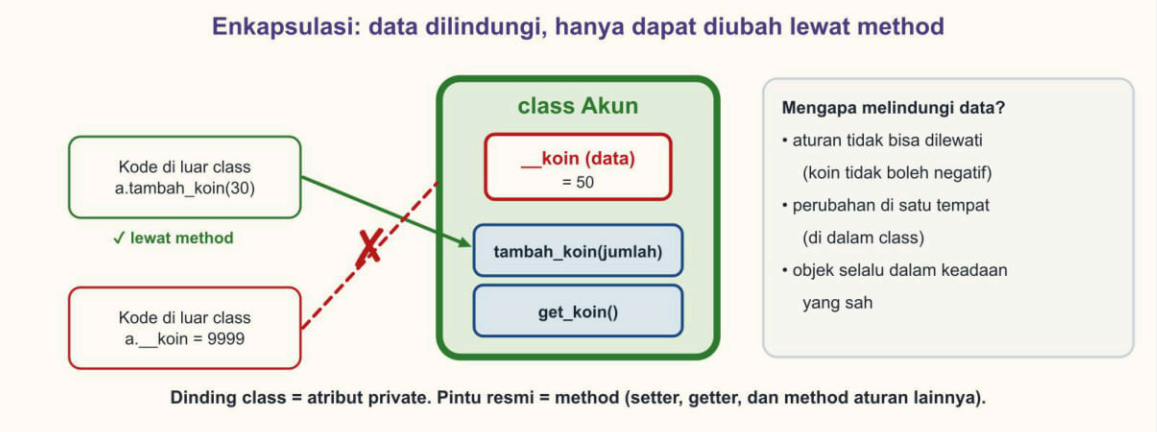
a = AkunTerbuka("Rani")
a.koin = -9999
print(a.koin)
```

Program itu mencetak `-9999`. Tidak ada yang mencegah koin menjadi negatif, padahal aturan gim mungkin menyatakan koin tidak boleh kurang dari nol. Dengan enkapsulasi, atribut `koin` disembunyikan dan hanya dapat berubah lewat method yang memeriksa aturan terlebih dulu. Hasilnya:

- **aturan tidak dapat dilewati**;
- **perubahan aturan cukup dilakukan di satu tempat** (di dalam class);
- **objek selalu berada dalam keadaan yang sah**.

Gambar 5.6 menunjukkan gagasan ini: atribut `__koin` berada di dalam "dinding" class, sehingga kode luar yang mencoba mengisinya langsung ditolak, sedangkan kode yang memanggil `tambah_koin()` diizinkan masuk lewat pintu resmi dan diperiksa lebih dulu.

Alasan lain datang dari **perubahan di masa depan**. Misalkan tim memutuskan bahwa koin disimpan dalam satuan baru, atau setiap penambahan harus dicatat ke riwayat. Bila atribut `koin` dipakai langsung di ratusan tempat, ratusan tempat itu harus diubah dan sebagian pasti terlewat. Bila semua perubahan lewat `tambah_koin()`, cukup satu method yang diubah, dan seluruh program otomatis mengikuti. Itulah sebabnya enkapsulasi juga memudahkan perawatan program jangka panjang.



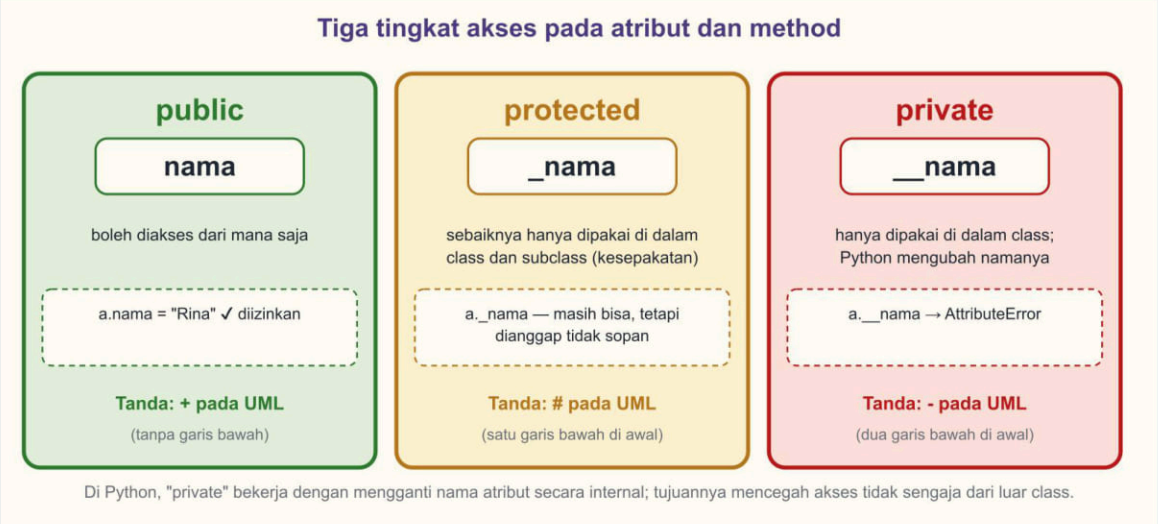
Gambar 5.6 Atribut privat berada di balik "dinding" class; method adalah pintu resminya. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Access modifier: public, protected, private

Access modifier (pengubah akses) menentukan siapa yang boleh mengakses atribut atau method. Ada tiga tingkat (Gambar 5.7).

- **Public**: dapat diakses dari mana saja. Ditulis tanpa awalan, misalnya `nama`.
- **Protected**: sebaiknya hanya dipakai di dalam class itu dan subclass-nya. Di Python ditulis dengan **satu garis bawah** di awal, misalnya `_nama`. Ini hanya **kesepakatan penulisan**; Python tidak melarangnya.
- **Private**: hanya untuk di dalam class. Di Python ditulis dengan **dua garis bawah** di awal, misalnya `__koin`. Python mengubah namanya secara internal sehingga **akses langsung dari luar class menimbulkan `AttributeError`**.

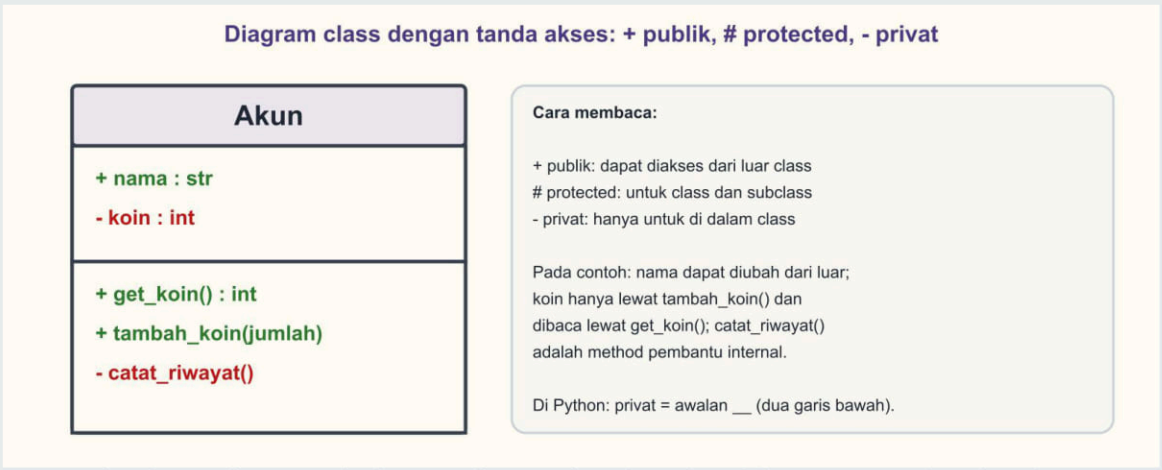


Gambar 5.7 Tiga tingkat akses di Python beserta tanda pada diagram class. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perhatikan perbedaan Python dengan beberapa bahasa lain. Di bahasa seperti Java, `private` adalah larangan yang dipaksakan. Di Python, awalan garis bawah adalah **tanda niat** (protected) atau **penyamaran nama** (private) yang mencegah akses tidak sengaja. Untuk keperluan soal TKA, pegangan yang berlaku: **atribut berawalan __ tidak dapat dibaca atau diubah langsung dari luar class; harus lewat method.**

Pada diagram class (UML), tingkat akses ditandai: **+** untuk public, **#** untuk protected, dan **-** untuk private (Gambar 5.8).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 5.8 Tanda + # - pada diagram class. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Getter dan setter

Bila atribut dibuat private, dari mana kode luar dapat membaca atau mengubahnya? Jawabannya lewat **method publik**.

- **Getter** adalah method yang **membaca** nilai atribut privat dan mengembalikannya. Contoh: `get_koin()`.
- **Setter** adalah method yang **mengubah** nilai atribut privat, **dengan memeriksa nilainya dulu**. Contoh: `set_suhu(nilai)`.

```
class Akun:
    def __init__(self, nama):
        self.nama = nama
        self.__koin = 0

    def get_koin(self):
        return self.__koin

    def tambah_koin(self, jumlah):
        if jumlah > 0:
            self.__koin += jumlah

a = Akun("Rani")
a.tambah_koin(50)
a.tambah_koin(-20)
print(a.get_koin())
a.nama = "Rina"
print(a.nama)
```

Keluarannya `50` dan `Rina`. Telusuri:

1. `__koin` dimulai dari 0 dan **privat**.
2. `tambah_koin(50)`: `50 > 0` benar → koin menjadi 50.
3. `tambah_koin(-20)`: `-20 > 0` salah → **ditolak**, koin tetap 50. Itulah gunanya pemeriksaan di dalam method.
4. `get_koin()` mengembalikan 50.
5. `a.nama` bersifat **publik**, jadi dapat diubah langsung dari luar menjadi `Rina`.

Bila kode luar mencoba `print(a.__koin)`, yang muncul adalah `AttributeError`, bukan 50. Pembacaan harus lewat `a.get_koin()`.

Setter yang memvalidasi adalah inti soal-soal bagian ini. Setter yang baik memeriksa bahwa nilai baru masuk akal, dan sering **mengembalikan `True` atau `False`** untuk memberi tahu apakah perubahan diterima (Gambar 5.9).

```
class Sensor:
    def __init__(self):
        self.__suhu = 25

    def get_suhu(self):
        return self.__suhu

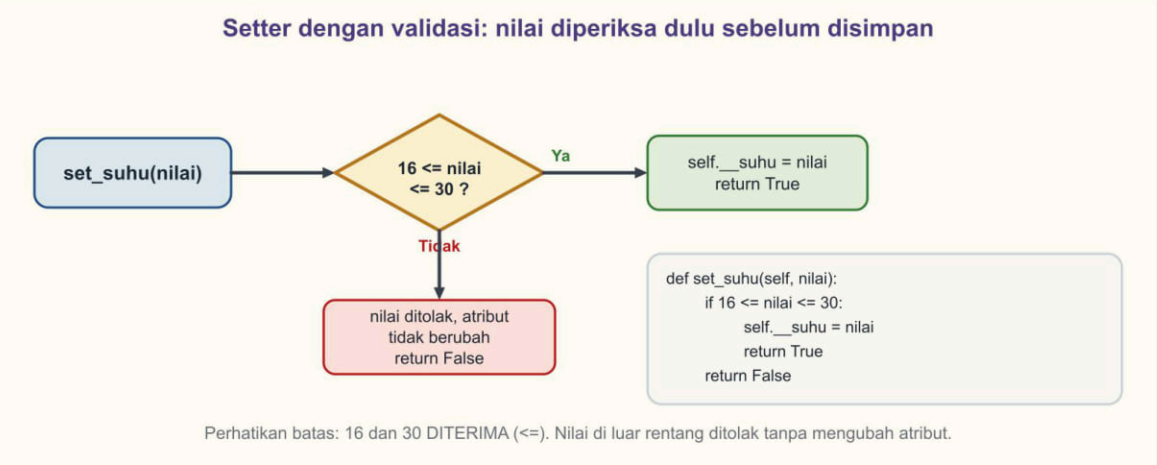
    def set_suhu(self, nilai):
        if 0 <= nilai <= 60:
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
            self.__suhu = nilai
            return True
        return False

s = Sensor()
hasil = s.set_suhu(75)
print(hasil, s.get_suhu())
print(s.set_suhu(60), s.get_suhu())
```

Keluarannya `False 25` lalu `True 60`. `set_suhu(75)` ditolak karena 75 di luar rentang 0–60, sehingga suhu tetap 25 dan hasilnya `False`. `set_suhu(60)` **diterima** karena batas 60 termasuk (`<=`), sehingga suhu menjadi 60 dan hasilnya `True`. Perhatikan lagi nilai batas, yang sering menjadi jebakan.



Gambar 5.9 Setter memeriksa nilai: diterima atau ditolak tanpa mengubah atribut. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Dua pelurusan istilah yang sering diuji:

- **`get_...()` adalah getter**, karena ia hanya **mengembalikan** nilai. Ia **bukan** setter walaupun mengembalikan nilai atribut privat.
- **`set_...()` atau `tambah_...()` yang mengubah nilai adalah setter** (atau method pengubah). Method seperti `tambah_koin()` mengubah nilai dengan aturan, sehingga berperan sebagai setter.

4. Method privat dan kapan atribut boleh publik

Bukan hanya atribut yang dapat dibuat privat; **method** pun bisa. Method privat adalah **pekerjaan di balik layar** yang hanya dipakai method lain di dalam class yang sama.

```
class Kasir:
    def __init__(self):
        self.__total = 0

    def __hitung_pajak(self, jumlah):
        return jumlah * 0.11

    def tambah_belanja(self, jumlah):
        self.__total += jumlah + self.__hitung_pajak(jumlah)

    def get_total(self):
        return self.__total

k = Kasir()
k.tambah_belanja(100000)
print(k.get_total())
```

Keluarannya `111000.0`. Method `__hitung_pajak()` hanya dipanggil dari dalam class, yaitu oleh `tambah_belanja()`. Kode luar tidak perlu dan tidak boleh memanggilnya. Dengan begitu, bila cara menghitung pajak berubah, hanya satu tempat yang diubah dan kode luar tidak terpengaruh.

Apakah semua atribut harus privat? **Tidak.** Atribut yang **tidak punya aturan** dan tidak berbahaya bila diubah, seperti `nama` pemain, boleh publik. Membuat semua atribut privat disertai getter dan setter yang tidak melakukan apa-apa justru menambah kode tanpa manfaat. Pertanyaan penentunya tetap: **adakah aturan yang bisa dilanggar bila atribut ini diubah sembarangan?**

Beberapa kesalahan penulisan juga sering menjadi pengecoh pada soal kode:

- **Lupa `self.`** ketika memakai atribut di dalam method (`__koin += jumlah` tanpa `self.` menimbulkan galat).
- **Salah jumlah garis bawah**: `_koin` (protected) dan `__koin` (private) bukan nama yang sama.
- **Method tanpa parameter `self`** sehingga tidak dapat dipanggil lewat objek.
- **Salah tempat validasi**: pemeriksaan ditaruh di luar class sehingga mudah terlewat oleh kode lain.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. Properti dengan @property (pengayaan ringan)

Python menyediakan cara yang lebih ringkas, yaitu **`@property`**, supaya getter dan setter dapat dipakai seperti atribut biasa sambil tetap memeriksa nilai.

```
class Lampu:
    def __init__(self):
        self.__terang = 0

    @property
    def terang(self):
        return self.__terang

    @terang.setter
    def terang(self, nilai):
        if 0 <= nilai <= 100:
            self.__terang = nilai

l = Lampu()
l.terang = 70
l.terang = 150
print(l.terang)
```

Keluarannya `70`. Penugasan `l.terang = 70` otomatis memanggil setter yang memeriksa rentangnya; `l.terang = 150` ditolak diam-diam karena di luar 0–100. Soal TKA lebih sering memakai bentuk `get_...()` dan `set_...()`, tetapi pola pemeriksaannya sama.

6. Memilih tingkat akses yang tepat

Saat merancang class, tentukan tingkat akses tiap anggota dengan pertanyaan: **siapa yang perlu mengaksesnya, dan bahayanya apa bila orang lain sembarangan mengubahnya?**

Anggota	Tingkat akses	Alasan
`nama` pemain	public	boleh dibaca dan diubah; tidak ada aturan khusus
`nyawa`	private + getter + method pengurang	tidak boleh negatif; hanya berkurang lewat `kena_serangan()`
`saldo` rekening	private + `setor()` dan `tarik()`	harus mematuhi aturan (tidak boleh menarik melebihi saldo)
`tunggakan` siswa	private + getter	halaman laporan hanya perlu membaca; berkurang lewat pembayaran tercatat
method pembantu internal	private	tidak untuk dipanggil dari luar
method layanan (`setor()`, `get_saldo()`)	public	pintu resmi bagi kode lain

Contoh penalaran: pada aplikasi pembayaran SPP, aturan sekolah menyatakan tunggakan tidak boleh negatif dan hanya boleh berkurang lewat pembayaran yang tercatat. Maka atribut `tunggakan` dibuat **private**. Petugas yang ingin "mengoreksi salah ketik" **tidak diberi akses langsung**, melainkan jalur koreksi yang tercatat lewat method khusus; membuka atribut

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

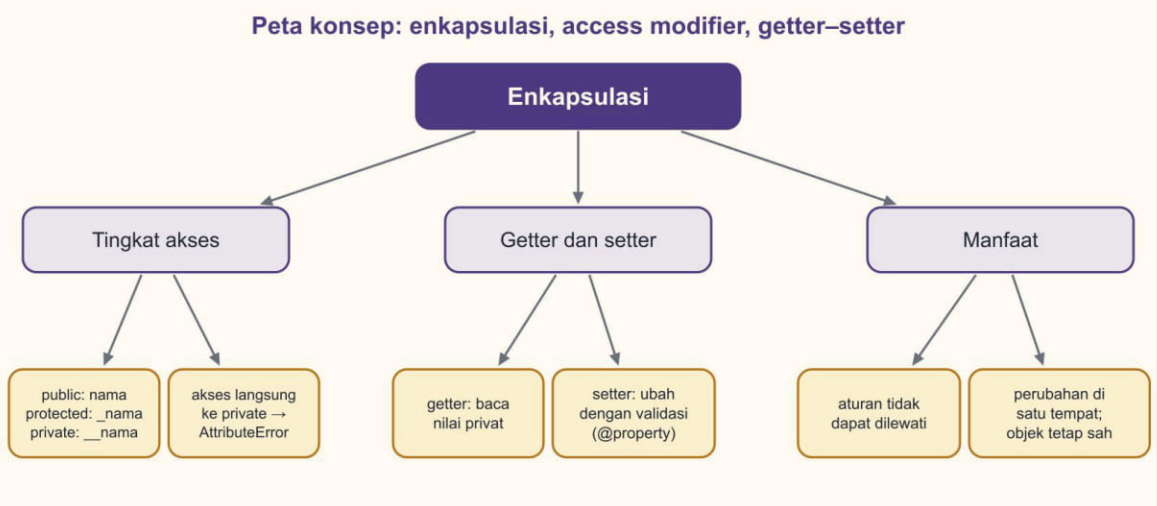
menjadi publik justru melanggar aturan sekolah. Halaman laporan memakai **getter** `get_tunggakan()` untuk membaca tanpa dapat mengubah. Dan method `bayar(jumlah)` menolak jumlah nol, negatif, atau melebihi tunggakan, sehingga aturan dijaga di dalam class.

Catatan tentang "private" di Python. Python tidak benar-benar mengunci atribut. Atribut `__koin` di dalam class `Akun` sebenarnya disimpan dengan nama `_Akun__koin`, dan orang yang sengaja menuliskan nama itu masih bisa mengaksesnya. Tujuan awalan dua garis bawah adalah **mencegah kecelakaan**: akses tidak sengaja dan bentrokan nama. Karena itu, aturan pemrograman yang baik tidak mengandalkan "keamanan" dari awalan garis bawah, tetapi memakainya sebagai tanda bahwa bagian itu bukan untuk diubah dari luar. Dalam soal TKA, pegangan yang berlaku adalah akses langsung `a.__koin` dari luar class menimbulkan `AttributeError`.

7. Ringkasan: ciri rancangan yang baik

Rancangan yang menerapkan enkapsulasi dengan baik memiliki ciri berikut:

- **Atribut yang punya aturan dibuat private**; atribut tanpa aturan boleh public.
- **Setiap perubahan atribut penting lewat method** yang memeriksa nilai.
- **Pembacaan lewat getter** bila kode luar hanya perlu melihat nilainya.
- **Method publik sedikit dan jelas**; rincian internal disembunyikan.
- **Validasi ada di dalam class**, bukan diserahkan kepada setiap kode yang memakai class.



Gambar 5.10 Peta konsep enkapsulasi, access modifier, dan getter–setter. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menelusuri class dengan setter berpengembalian

Tentukan keluaran program berikut. *(Kasus fiktif.)*

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
class Termostat:
    def __init__(self):
        self.__suhu = 22

    def get_suhu(self):
        return self.__suhu

    def set_suhu(self, nilai):
        if 16 <= nilai <= 30:
            self.__suhu = nilai
            return True
        return False

t = Termostat()
a = t.set_suhu(25)
b = t.set_suhu(35)
c = t.set_suhu(16)
print(a, b, c, t.get_suhu())
```

1. Suhu awal 22 (privat).
2. `set_suhu(25)`: `16 <= 25 <= 30` benar → suhu 25 → `True`.
3. `set_suhu(35)`: `35 <= 30` salah → ditolak, suhu tetap 25 → `False`.
4. `set_suhu(16)`: batas bawah **diterima** (<=) → suhu 16 → `True`.
5. `get_suhu()` → 16. Cetak `True False True 16`.

Jawaban: `True False True 16`. Perhatikan bahwa penolakan 35 tidak mengubah suhu, dan nilai batas 16 diterima.

Contoh 2 — Menambahkan validasi pada nilai siswa

Class `Siswa` menyimpan nilai privat dan harus menolak nilai di luar 0–100. Lengkapi dan telusuri. *(Kasus fiktif.)*

```
class Siswa:
    def __init__(self, nama):
        self.nama = nama
        self.__nilai = 0

    def set_nilai(self, nilai):
        if 0 <= nilai <= 100:
            self.__nilai = nilai
            return True
        return False

    def get_nilai(self):
        return self.__nilai

s = Siswa("Dewi")
print(s.set_nilai(85), s.get_nilai())
print(s.set_nilai(120), s.get_nilai())
print(s.set_nilai(0), s.get_nilai())
```

1. `set_nilai(85)`: di dalam rentang → nilai 85, hasil `True` → `True 85`.
2. `set_nilai(120)`: di luar rentang → ditolak, nilai tetap 85, hasil `False` → `False 85`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. `set_nilai(0)`: 0 termasuk batas bawah → diterima, nilai 0 → `True 0`.

Jawaban: `True 85`, `False 85`, `True 0`.

Contoh 3 — Menentukan tingkat akses untuk class Rekening

Tentukan tingkat akses yang tepat untuk anggota class `Rekening`: `nama_pemilik`, `saldo`, `setor()`, `tarik()`, `cek_saldo()`, dan sebuah method pembantu `catat_riwayat()`. *(Kasus fiktif.)*

1. `nama_pemilik`: tidak ada aturan khusus → **public**.
2. `saldo`: harus mematuhi aturan (tidak boleh melebihi penarikan) → **private**.
3. `setor()` dan `tarik()`: pintu resmi untuk mengubah saldo dengan pemeriksaan → **public**.
4. `cek_saldo()`: getter yang hanya membaca → **public**.
5. `catat_riwayat()`: hanya dipakai di dalam class → **private**.

Jawaban: public: `nama_pemilik`, `setor()`, `tarik()`, `cek_saldo()`; private: `saldo` dan `catat_riwayat()`.

Contoh 4 — Akses langsung yang gagal

Tentukan hasil dari tiap baris tambahan setelah class `Akun` pada bagian 3. *(Kasus fiktif.)*

- (a) `print(a.get_koin())` setelah `a.tambah_koin(40)`.
- (b) `print(a.__koin)` dari luar class.
- (c) `a.nama = "Sari"` lalu `print(a.nama)`.

1. (a) Getter mengembalikan nilai koin, yang sekarang 40 (bila sebelumnya 0) → `40`.
2. (b) `__koin` privat; akses langsung dari luar → **`AttributeError`**, bukan 40.
3. (c) `nama` publik → dapat diubah langsung; mencetak `Sari`.

Jawaban: (a) `40`; (b) `AttributeError`; (c) `Sari`.

Contoh 5 — Menilai usulan perubahan pada class

Class `Siswa` untuk aplikasi pembayaran SPP memiliki atribut privat `__tunggakan`, getter `get_tunggakan()`, dan method `bayar(jumlah)` yang menolak jumlah nol, negatif, atau melebihi tunggakan. Seorang petugas mengusulkan dua perubahan: (1) mengubah `__tunggakan` menjadi publik agar salah ketik dapat dikoreksi langsung, dan (2) menghapus pemeriksaan "melebihi tunggakan" supaya pembayaran lebih cepat. Nilai usulan itu. *(Kasus fiktif.)*

1. Aturan sekolah: tunggakan tidak boleh negatif dan hanya berkurang lewat pembayaran yang tercatat.
2. Usulan (1): bila atribut publik, siapa pun dapat mengisinya dengan nilai apa pun tanpa catatan. Itu melanggar aturan "hanya lewat pembayaran yang tercatat". Salah ketik tetap dapat dikoreksi, tetapi lewat method koreksi khusus yang mencatat siapa mengubah dan kapan, bukan dengan membuka atribut.
3. Usulan (2): tanpa pemeriksaan "melebihi tunggakan", pembayaran Rp500.000 pada tunggakan Rp300.000 membuat tunggakan menjadi **−200.000**, yaitu nilai negatif yang dilarang aturan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Kesimpulan: kedua usulan melemahkan enkapsulasi dan sebaiknya ditolak, dengan menawarkan jalur koreksi tercatat sebagai gantinya.

Jawaban: tolak keduanya; atribut tetap private, pemeriksaan tetap di dalam `bayar()`, dan koreksi lewat method khusus yang tercatat.

Contoh 6 — Atribut private pada subclass (gambaran awal)

Class `Pahlawan` memiliki `__nyawa` (privat) dan `nama` (publik). Class `Pemanah(Pahlawan)` dibuat sebagai turunannya. Dapatkah method di dalam `Pemanah` mengubah `self.__nyawa` secara langsung? *(Kasus fiktif.)*

1. Atribut `__nyawa` privat milik class `Pahlawan`.
2. Python mengubah nama atribut privat menjadi bentuk yang terikat pada class pemiliknya, sehingga kode di dalam **subclass tidak dapat memakai `self.__nyawa` secara langsung**.
3. Cara yang benar: subclass memanggil **method milik class induk**, misalnya `self.kena_serangan(10)` atau `self.get_nyawa()`, yang berada di dalam `Pahlawan` dan sah mengakses `__nyawa`.
4. Bila induk ingin mengizinkan subclass mengakses langsung, atributnya dibuat **protected** (`_nyawa`), yang menjadi kesepakatan bahwa hanya class dan subclass yang memakainya.

Jawaban: tidak; subclass harus memakai method induk. Pembahasan lebih lanjut ada di sub bab 5.3.

❖ PETA KONSEP

Gambar 5.10 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Enkapsulasi** = membungkus data dan method serta melindungi data dari perubahan langsung dari luar.
- **Tingkat akses di Python**: `nama` (public), `_nama` (protected, kesepakatan), `__nama` (private; akses langsung dari luar → `AttributeError`). Tanda UML: +, #, -.
- **Getter** membaca; **setter** mengubah **dengan validasi**. `get_...()` bukan setter.
- Setter yang menolak nilai **tidak mengubah atribut**; periksa **nilai batas** (<= atau <).
- Atribut yang punya aturan dibuat **private**; atribut tanpa aturan boleh public.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menelusuri soal enkapsulasi:

1. **Tandai atribut privat** (berawalan `__`) dan atribut publik.
2. **Telusuri tiap pemanggilan method berurutan**, mencatat nilai atribut privat setelah tiap panggilan.
3. **Untuk setter dengan `if`**, periksa kondisi dengan nilai yang dikirim, termasuk nilai batas; bila kondisi salah, atribut **tidak berubah**.
4. **Akses `.__atribut` dari luar class** → `AttributeError`.

Langkah menjawab soal merancang class:

1. **Tanyakan: adakah aturan yang harus dijaga?** Bila ya, atribut private dengan setter yang memeriksa.
2. **Tanyakan: siapa yang perlu membaca saja?** Beri getter.
3. **Tolak solusi yang membuka atribut menjadi publik** demi kemudahan, bila ada aturan yang bisa dilanggar.

Jebakan yang sering dipasang:

- **`__atribut` dikira dapat dibaca langsung dari luar** (`print(a.__koin)` menampilkan nilainya).
- **Getter disebut setter** karena "mengembalikan nilai atribut privat".
- **Nilai yang ditolak dikira tetap tersimpan**, atau setter dikira selalu `True`.
- **Nilai batas** (`60` pada `0 <= nilai <= 60`) dikira ditolak.
- **Atribut publik dikira dilindungi** padahal bisa diubah langsung (`a.nama = "Rina"`).
- **Atribut privat dibuat publik "supaya petugas mudah mengoreksi"** padahal melanggar aturan.
- **`_atribut` dikira benar-benar tertutup.** Itu hanya kesepakatan.
- **Validasi diletakkan di kode pemanggil**, bukan di dalam class.

> Contoh salah tafsir: seorang siswa menyimpulkan bahwa setelah `s.set_suhu(75)` pada sensor dengan rentang 0–60, "suhu menjadi 75 karena method sudah dipanggil". Pemanggilan hanya berarti permintaan; pemeriksaan di dalam method menolak nilai 75, sehingga atribut tidak berubah dan hasilnya `False`.

• BEDAH SOAL PREDIKSI

Soal berikut menyajikan class dengan atribut privat dan setter yang menolak nilai bernilai negatif. Perhatikan cara mengikuti pemanggilan berurutan dan menilai tiap pernyataan, termasuk pernyataan tentang atribut publik `nama` (bagian 3).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Class berikut menyimpan koin pemain dalam gim *Pasar Rempah* (2027). Atribut koin dibuat privat agar hanya diubah lewat method.

```
class Akun:
    def __init__(self, nama):
        self.nama = nama
        self.__koin = 0

    def get_koin(self):
        return self.__koin

    def tambah_koin(self, jumlah):
        if jumlah > 0:
            self.__koin += jumlah

a = Akun("Rani")
a.tambah_koin(50)
a.tambah_koin(-20)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang kode tersebut adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `a.get_koin()` menghasilkan 50 karena tambahan −20 ditolak oleh pemeriksaan if.

☐ B. `a.get_koin()` menghasilkan 30 karena 50 dikurangi 20 sama dengan 30.

☐ C. `print(a.__koin)` dari luar class menampilkan angka 50.

☐ D. Atribut nama dapat diubah langsung dari luar, misalnya `a.nama = "Rina"`.

☐ E. Method `get_koin()` disebut setter karena dipakai untuk mengubah nilai koin.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Pembahasan **Kunci: A, D**

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Enkapsulasi menyembunyikan data penting. Di Python, atribut berawalan dua garis bawah (`__koin`) diperlakukan sebagai privat: tidak dapat dipanggil dengan nama itu dari luar class. Getter membaca nilai; setter/method pengubah mengubah nilai sambil memeriksa aturan.

Langkah:
1. a dibuat: nama "Rani", __koin 0.
2. tambah_koin(50): 50 > 0 → __koin = 50.
3. tambah_koin(−20): −20 > 0 salah → tidak berubah, __koin tetap 50.

Analisis setiap pernyataan:
• **Pernyataan 1 ✓** Panggilan pertama menambah 50; panggilan kedua tidak memenuhi jumlah > 0 sehingga diabaikan. Koin = 50.
• **Pernyataan 2 ✗** Jebakan: −20 tidak pernah dijumlahkan karena syarat jumlah > 0 bernilai salah.
• **Pernyataan 3 ✗** Atribut berawalan __ tidak dapat diakses dengan nama itu dari luar class; perintah ini menimbulkan AttributeError.
• **Pernyataan 4 ✓** nama tidak diberi awalan __ sehingga bersifat publik dan dapat diubah langsung.
• **Pernyataan 5 ✗** get_koin() hanya membaca nilai (getter); yang mengubah koin adalah tambah_koin().

Jawaban: pernyataan 1, 4.

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap soal, tulis nilai atribut privat setelah tiap pemanggilan, lalu periksa tiap pernyataan.

1. Aplikasi pemantau kebun hidroponik sekolah (2027) menyimpan suhu air dari sensor. Nilai yang masuk akal hanya 0–60 °C. Programmer menulis class berikut.

```
class Sensor:
    def __init__(self):
        self.__suhu = 25

    def get_suhu(self):
        return self.__suhu

    def set_suhu(self, nilai):
        if 0 <= nilai <= 60:
            self.__suhu = nilai
            return True
        return False

s = Sensor()
hasil = s.set_suhu(75)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang kode tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Setelah kode dijalankan, `s.get_suhu()` menghasilkan 75 karena set_suhu() sudah dipanggil.
☐ B. Setelah kode dijalankan, hasil bernilai False dan `s.get_suhu()` menghasilkan 25.
☐ C. Pemanggilan `s.set_suhu(-5)` mengembalikan True karena −5 lebih kecil dari 60.
☐ D. Method get_suhu() termasuk setter karena mengembalikan nilai atribut privat.
☐ E. `s.set_suhu(60)` diterima karena batas 60 termasuk dalam rentang yang diizinkan.

2. Tim PKL merancang class `Siswa` untuk aplikasi pembayaran SPP sebuah SMK (2027) dengan atribut nis, nama, dan tunggakan. Aturan sekolah: tunggakan tidak boleh negatif dan hanya boleh berkurang lewat pembayaran yang tercatat; halaman laporan hanya perlu membaca tunggakan. Petugas tata usaha mengusulkan agar tunggakan dapat dikoreksi langsung bila ada salah ketik.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Atribut tunggakan sebaiknya dibuat publik agar petugas dapat langsung mengoreksinya bila ada salah ketik.	○	○
Method bayar(jumlah) yang menolak jumlah nol, negatif, atau melebihi tunggakan menjaga aturan sekolah tetap dipatuhi.	○	○
Getter get_tunggakan() memungkinkan halaman laporan membaca tunggakan tanpa dapat mengubahnya.	○	○

Kunci singkat: 1. B, E 2. 1S 2B 3B

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa atribut dengan aturan (seperti saldo atau nyawa) dibuat private dan hanya diubah lewat method?
- Apa yang terjadi pada atribut bila setter menolak nilai yang dikirim kepadanya?
- **Yang harus kuingat:** `__atribut` tidak dapat diakses langsung dari luar class; getter membaca, setter mengubah dengan validasi; nilai yang ditolak tidak mengubah atribut; periksa nilai batas; atribut yang punya aturan dibuat private.



Latihan interaktif di LMS

Sub bab 5.2: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-5.2

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 5.3 · Pewarisan

SMKPPLG-5.3

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **menerapkan relasi pewarisan antar-class yang sesuai**, dengan tingkat berpikir menerapkan dan **mengklasifikasi**. Contoh soal resmi Pusmendik (polymorphism, dengan class induk dan dua class turunan) pun mengandaikan kamu memahami pewarisan lebih dulu. Pada soal, kamu biasanya menghadapi tiga hal: **menelusuri kode** class induk dan turunannya untuk menentukan keluaran, **membaca diagram class** dan menyebut atribut serta method yang dimiliki sebuah objek, dan **memilih rancangan pewarisan** yang tepat (atau menolak pewarisan yang tidak cocok).

Hampir semua soal pada sub bab ini tergolong sulit, bukan karena istilahnya rumit, tetapi karena **jejak pemanggilan** harus dilacak dengan teliti: method mana yang dijalankan, atribut mana yang sudah ada, dan apakah `super()` sudah dipanggil.

Setelah mempelajari sub bab ini, kamu dapat:

- menjelaskan pewarisan, superclass, dan subclass serta menulis `class Anak(Induk);`
- menyebutkan atribut dan method yang dimiliki sebuah objek subclass, termasuk yang diwarisi dari beberapa tingkat induk;
- menjelaskan method overriding dan fungsi `super()`, termasuk `super().__init__(...);`
- menelusuri keluaran program yang memakai pewarisan;
- membedakan hubungan *is-a* (pewarisan) dan *has-a* (atribut), dan menentukan kapan pewarisan tidak tepat.

◆ KONSEP INTI

Dalam sebuah gim, ada banyak jenis karakter: pemanah, penyihir, prajurit, tabib. Semuanya punya nama dan nyawa, semuanya bisa terkena serangan, tetapi masing-masing punya kemampuan khusus. Bila setiap jenis ditulis sebagai class terpisah dari nol, atribut `nama`, `nyawa`, dan method `kena_serangan()` akan ditulis berulang-ulang. Bila aturan nyawa berubah, semua class harus diubah. **Pewarisan** (*inheritance*) menyelesaikan masalah itu: bagian yang sama ditulis **sekali** dalam sebuah class induk, dan class lain **mewarisinya**.

1. Istilah dan sintaks pewarisan

- **Superclass** (class induk, *parent class*, *base class*) adalah class yang bagian-bagiannya diwarisi.
- **Subclass** (class turunan, *child class*) adalah class yang mewarisi dan boleh menambah anggota baru.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **Pewarisan** berarti subclass **otomatis memiliki** atribut dan method superclass-nya.

Di Python, superclass ditulis **dalam kurung** setelah nama subclass:

```
class Hewan:
    def __init__(self, nama):
        self.nama = nama

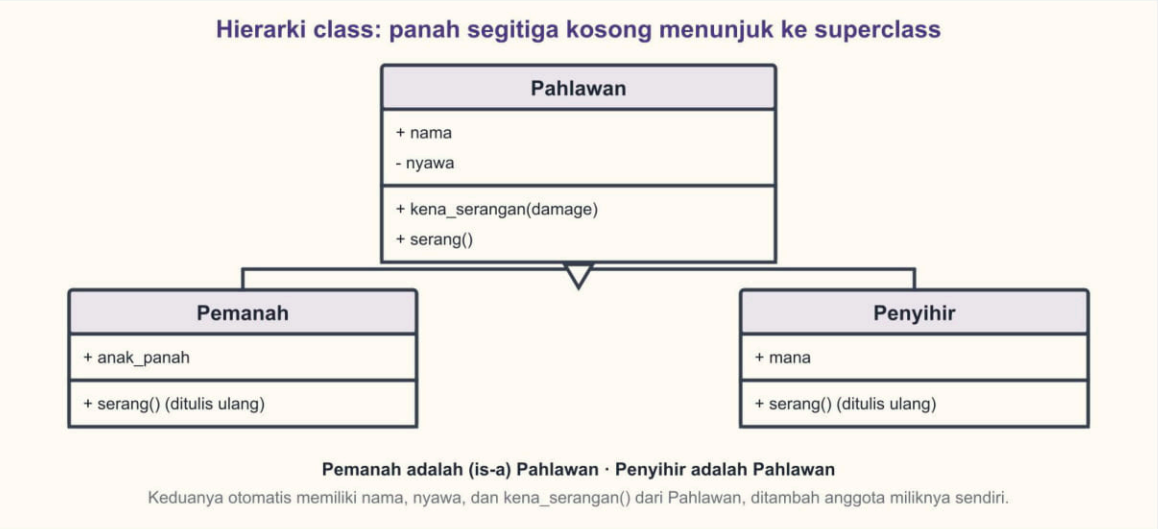
    def info(self):
        return "Hewan " + self.nama

class Kucing(Hewan):
    def suara(self):
        return "meong"

k = Kucing("Mimi")
print(k.info(), k.suara())
```

Keluarannya `Hewan Mimi meong`. `Kucing(Hewan)` berarti `Kucing` mewarisi `Hewan`. `Kucing` tidak menulis `__init__` maupun `info()`, tetapi objek `k` tetap memiliki atribut `nama` dan dapat memanggil `info()` karena **keduanya diwarisi**. Karena `Kucing` tidak punya `__init__` sendiri, `__init__` milik `Hewan` dipakai, dan itu sebabnya `Kucing("Mimi")` mengirim satu argumen (`nama`).

Pada diagram class (UML), pewarisan digambar sebagai garis dengan **segitiga kosong** di ujung yang menunjuk ke **superclass** (Gambar 5.11).



Gambar 5.11 Pemanah dan Penyihir mewarisi Pahlawan; panah segitiga kosong menunjuk ke superclass. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perhatikan bahwa kotak subclass pada diagram biasanya hanya memuat anggota **miliknya sendiri**. Anggota yang diwarisi tidak ditulis ulang di kotaknya, tetapi tetap dimiliki objeknya. Itu sebabnya soal diagram class meminta kamu **menggabungkan** isi kotak subclass dengan isi semua kotak induknya.

Manfaat pewarisan dapat dirangkum menjadi tiga hal. Pertama, **tidak menulis ulang**: bagian yang sama berada di satu tempat sehingga kode lebih pendek. Kedua, **perubahan di satu tempat**: bila aturan nyawa berubah, cukup mengubah `Pahlawan`, dan seluruh

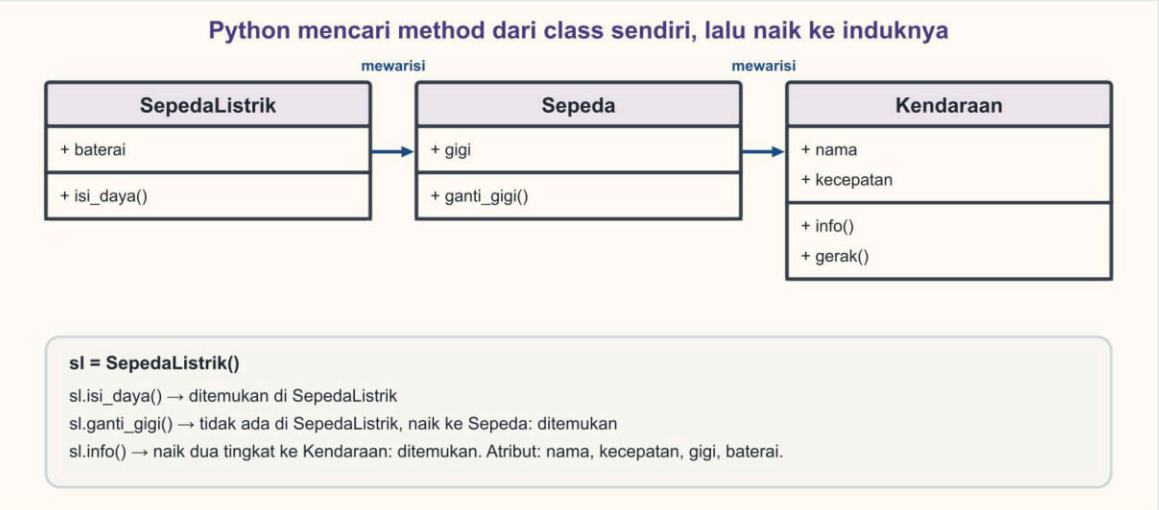
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

turunannya (pemanah, penyihir, tabib) otomatis ikut. Ketiga, **mudah menambah jenis baru**: menambah `Ksatria` cukup dengan menulis subclass yang hanya memuat hal-hal yang berbeda, bukan menyalin semuanya dari awal. Karena itu, pewarisan sering dipakai pada gim, aplikasi dengan banyak jenis pengguna (siswa, guru, admin), dan sistem yang memiliki banyak jenis produk.

Di Python sebuah class bahkan dapat memiliki lebih dari satu induk (*pewarisan ganda*), tetapi bentuk itu di luar cakupan TKA. Fokus soal adalah pewarisan dengan **satu induk**, baik satu tingkat maupun bertingkat.

2. Pewarisan bertingkat dan pencarian method

Pewarisan dapat **bertingkat**: `SepedaListrik` mewarisi `Sepeda`, yang mewarisi `Kendaraan`. Sebuah objek `SepedaListrik` memiliki **semua** atribut dan method dari ketiga tingkat (Gambar 5.12).



Gambar 5.12 Pewarisan tiga tingkat dan cara Python mencari method. Sumber: digambar oleh Tim Next Level Study Indonesia.

Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Saat sebuah method dipanggil lewat objek, Python mencarinya dengan urutan tetap: **pertama di class objek itu sendiri, bila tidak ada, naik ke superclass-nya, lalu ke superclass di atasnya**, sampai ketemu. Karena itu:

- `sl.isi_daya()` ditemukan di `SepedaListrik`.
- `sl.ganti_gigi()` tidak ada di `SepedaListrik`, tetapi ada di `Sepeda`, jadi tetap dapat dipanggil.
- `sl.info()` dicari ke atas sampai `Kendaraan`.

Sebaliknya, **induk tidak memiliki anggota milik turunannya**. Objek `Sepeda` **tidak** dapat memanggil `isi_daya()`, sebab method itu baru ada di `SepedaListrik`. Arah pewarisan hanya ke bawah: turunan memiliki milik induk, bukan sebaliknya. Kesalahan konsep yang sering dipasang pada soal adalah membalik arah ini.

3. Method overriding: menulis ulang method induk

Subclass boleh **menulis ulang** (*override*) sebuah method induk dengan **nama yang sama** bila perilakunya berbeda. Python akan memakai versi milik subclass, karena pencarian dimulai dari class objek itu sendiri.

```
class Tiket:
    def __init__(self, harga):
        self.harga = harga

    def total(self):
        return self.harga

class TiketVIP(Tiket):
    def total(self):
        return super().total() + 25000

class TiketPelajar(Tiket):
    def total(self):
        return self.harga - 5000

print(Tiket(50000).total())
print(TiketVIP(50000).total())
print(TiketPelajar(50000).total())
```

Keluarannya `50000`, `75000`, dan `45000`. `TiketVIP` dan `TiketPelajar` sama-sama menulis ulang `total()`. Pada `TiketVIP`, **`super().total()`** memanggil versi milik induk (50000), lalu menambahkan 25000. Dengan `super()`, subclass **memperluas** perilaku induk, bukan menggantikannya sepenuhnya.

4. super().init: mengisi atribut milik induk

Bagian yang paling sering menjebak adalah `__init__`. Bila sebuah subclass menulis `__init__` sendiri, **`__init__` induk tidak otomatis dijalankan**. Atribut milik induk baru terbentuk bila subclass **memanggilnya secara eksplisit** dengan `super().__init__(...)` (Gambar 5.13).

```
class Karakter:
    def __init__(self, nama, nyawa):
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

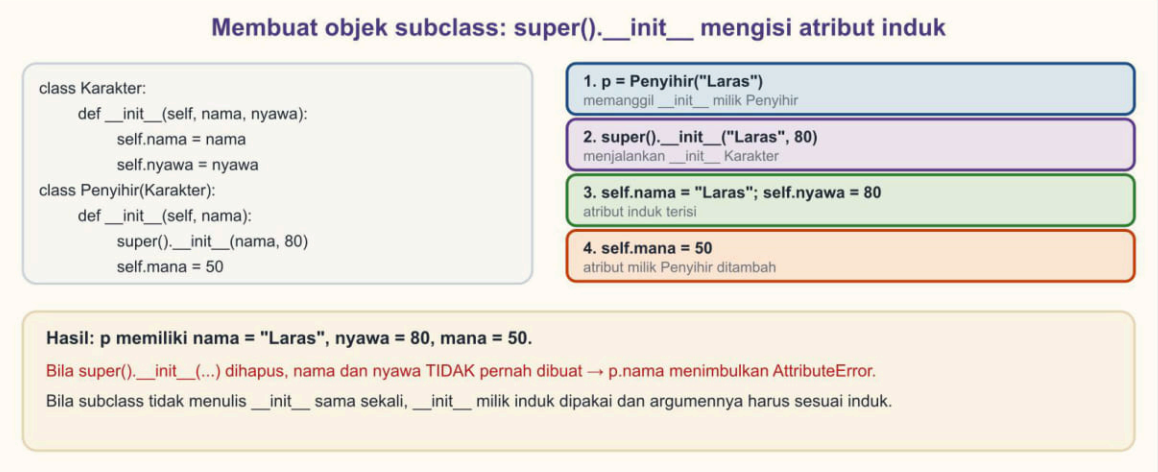
```
        self.nama = nama
        self.nyawa = nyawa

    def info(self):
        return self.nama + " " + str(self.nyawa)

class Pemanah(Karakter):
    def __init__(self, nama):
        super().__init__(nama, 100)
        self.anak_panah = 30

p = Pemanah("Raka")
print(p.info(), p.anak_panah)
```

Keluarannya `Raka 100 30`. Telusuri: `Pemanah("Raka")` menjalankan `__init__` milik `Pemanah`; baris `super().__init__(nama, 100)` menjalankan `__init__` milik `Karakter`, sehingga `nama = "Raka"` dan `nyawa = 100` terbentuk; lalu `anak_panah = 30` ditambahkan. `p.info()` memakai method warisan.



*Gambar 5.13 Urutan pembuatan objek subclass dengan super().**init**. Sumber: digambar oleh Tim Next Level Study Indonesia.*

Educational Institutions

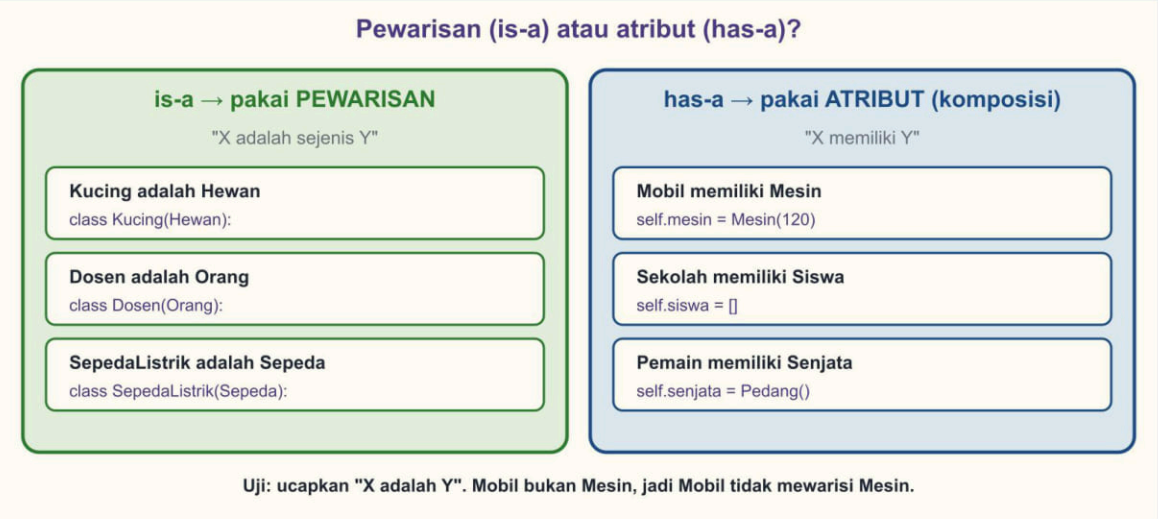
Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Apa yang terjadi bila baris `super().__init__(...)` dihapus? Atribut `nama` dan `nyawa` **tidak pernah dibuat**. Memanggil `p.info()` akan menimbulkan `AttributeError`, karena `self.nama` tidak ada. Dua catatan tambahan:

- Argumen yang dikirim ke `super().__init__(...)` harus **cocok dengan parameter `__init__` induk**. Pada contoh, induk meminta `nama` dan `nyawa`, jadi keduanya dikirim (nyawa diisi tetap 100).
- Subclass yang **tidak** menulis `__init__` memakai `__init__` induk apa adanya, sehingga membuat objeknya harus mengirim argumen sesuai induk.

5. Is-a dan has-a: kapan memakai pewarisan

Pewarisan menyatakan hubungan **"adalah"** (*is-a*): `Pemanah` **adalah** `Karakter`, `Kucing` **adalah** `Hewan`. Bila hubungannya **"memiliki"** (*has-a*), yang tepat adalah **menjadikan objek lain sebagai atribut**, bukan pewarisan (Gambar 5.14).



Gambar 5.14 Pewarisan untuk hubungan "adalah", atribut untuk hubungan "memiliki". Sumber: digambar oleh Tim Next Level Study Indonesia.

```
class Mesin:
    def __init__(self, tenaga):
        self.tenaga = tenaga

class Mobil:
    def __init__(self, tenaga):
        self.mesin = Mesin(tenaga)

m = Mobil(120)
print(m.mesin.tenaga)
```

Keluarannya `120`. `Mobil` **memiliki** `Mesin`, jadi mesin disimpan sebagai atribut. Mobil bukan jenis mesin, sehingga `class Mobil(Mesin)` akan keliru.

Uji cepat: ucapkan "X adalah Y". "Kucing adalah hewan" benar, jadi pewarisan. "Mobil adalah mesin" janggal, jadi bukan pewarisan. "Sekolah memiliki siswa" menunjuk atribut berupa daftar siswa.

Pewarisan juga **tidak tepat** ketika:

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- hanya ingin **memakai ulang beberapa baris kode**, padahal tidak ada hubungan "adalah";
- subclass **menolak atau meniadakan** perilaku induk, sehingga tidak lagi bisa dipakai di mana induknya dipakai;
- hierarkinya menjadi terlalu dalam dan rumit.

Satu pengamatan tambahan tentang **atribut class yang ditimpa**: pada subclass boleh menulis ulang atribut class milik induk (misalnya `batas = 5`). Objek subclass membaca versi miliknya, sedangkan objek induk tetap memakai versi induk. Pengamatan ini berguna bila soal memberi satu nilai bawaan di induk dan mengubahnya di sebuah subclass: kamu cukup menentukan objek mana yang bertanya.

Hati-hati pula dengan **nama method yang sama tetapi parameter berbeda** antara induk dan subclass. Python tidak mengenal *overloading*: bila subclass menulis ulang `hitung(a)` menjadi `hitung(a, b)`, versi subclass yang dipakai, dan pemanggilan dengan satu argumen menimbulkan galat. Aturan praktisnya, method yang ditulis ulang sebaiknya memiliki parameter yang sama dengan method induk.

6. Atribut privat dan pewarisan

Atribut privat induk (berawalan `__`) **tidak dapat dipakai langsung** oleh kode di dalam subclass, karena namanya terikat pada class induk. Subclass memakai **method milik induk** untuk membaca atau mengubahnya, misalnya `self.kena_serangan(10)` atau `self.get_nyawa()`. Bila induk ingin mengizinkan subclass memakai atributnya secara langsung, atribut dibuat **protected** (satu garis bawah, `_nyawa`). Pembahasan akses ini terkait dengan sub bab 5.2, dan keduanya perlu dibaca bersama.

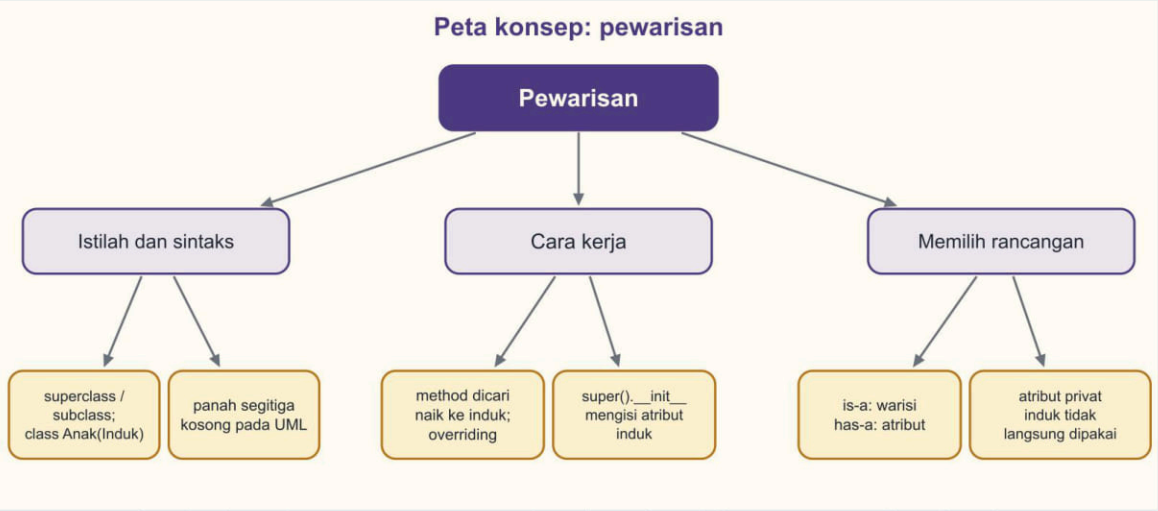
7. isinstance dan kesetaraan subclass dengan induknya

Karena subclass adalah sejenis induknya, sebuah objek subclass **dianggap juga** objek induk. Fungsi `isinstance(objek, Class)` memeriksanya:

```
p = Pemanah("Raka")
print(isinstance(p, Pemanah), isinstance(p, Karakter), isinstance(p, str))
```

Keluarannya `True True False`. Objek `p` adalah `Pemanah` dan juga `Karakter` (induknya), tetapi bukan `str`. Sifat inilah yang membuat sebuah daftar bertipe induk dapat berisi objek turunan, dasar polimorfisme pada sub bab 5.4.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 5.15 Peta konsep pewarisan. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menelusuri atribut dan method warisan

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
class Perangkat:
    def __init__(self, nama, baterai):
        self.nama = nama
        self.baterai = baterai

    def status(self):
        return self.nama + ": " + str(self.baterai) + "%"

class Ponsel(Perangkat):
    def __init__(self, nama):
        super().__init__(nama, 100)
        self.sinyal = 4

    def telepon(self):
        self.baterai -= 10
        return "menelepon"

p = Ponsel("Nova")
print(p.telepon())
print(p.status(), p.sinyal)
```

1. `Ponsel("Nova")` memanggil `__init__` Ponsel, yang memanggil `super().__init__("Nova", 100)`: `nama = "Nova"`, `baterai = 100`. Lalu `sinyal = 4`.
2. `p.telepon()`: `baterai` menjadi 90 dan mengembalikan `"menelepon"` → dicetak `menelepon`.
3. `p.status()` adalah method **warisan** dari Perangkat: `"Nova: 90%"`. `p.sinyal` = 4.
4. Baris terakhir mencetak `Nova: 90% 4`.

Jawaban: `menelepon`, lalu `Nova: 90% 4`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Contoh 2 — Override dengan super()

Tentukan keluaran dan jelaskan peran `super()`. *(Kasus fiktif.)*

```
class Kendaraan:
    def __init__(self, roda):
        self.roda = roda

    def biaya_parkir(self):
        return 1000

class Truk(Kendaraan):
    def biaya_parkir(self):
        return super().biaya_parkir() * 3

class Sepeda(Kendaraan):
    def biaya_parkir(self):
        return 500

print(Kendaraan(4).biaya_parkir(), Truk(6).biaya_parkir(), Sepeda(2).biaya_parkir())
```

1. `Kendaraan(4).biaya_parkir()`: method milik Kendaraan → 1000.
2. `Truk(6).biaya_parkir()`: Truk menulis ulang; `super().biaya_parkir()` memanggil versi Kendaraan (1000), dikalikan 3 → **3000**.
3. `Sepeda(2).biaya_parkir()`: Sepeda menulis ulang tanpa `super()` → **500**.
4. Truk dan Sepeda tidak menulis `__init__`, jadi memakai `__init__` Kendaraan (satu argumen, `roda`).

Jawaban: `1000 3000 500`.

Contoh 3 — Akibat lupa memanggil super().init

Class berikut dijalankan. Apa yang terjadi, dan bagaimana memperbaikinya? *(Kasus fiktif.)*

```
# tidak dijalankan (menimbulkan AttributeError)
class Akun:
    def __init__(self, nama):
        self.nama = nama

class AkunPremium(Akun):
    def __init__(self, nama, bonus):
        self.bonus = bonus

a = AkunPremium("Tia", 10)
print(a.bonus)
print(a.nama)
```

1. `AkunPremium` menulis `__init__` sendiri, sehingga `__init__` milik `Akun` **tidak dijalankan**.
2. Hanya `self.bonus` yang dibuat. `print(a.bonus)` mencetak `10`.
3. `a.nama` tidak pernah dibuat → `AttributeError: 'AkunPremium' object has no attribute 'nama'`.
4. Perbaikan: panggil `super().__init__(nama)` di dalam `__init__` subclass:

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
class Akun:
    def __init__(self, nama):
        self.nama = nama

class AkunPremium(Akun):
    def __init__(self, nama, bonus):
        super().__init__(nama)
        self.bonus = bonus

a = AkunPremium("Tia", 10)
print(a.bonus, a.nama)
```

Keluarannya `10 Tia`.

Jawaban: tanpa `super().__init__`, atribut induk tidak terbentuk; setelah diperbaiki, `10 Tia`.

Contoh 4 — Pewarisan atau atribut?

Untuk tiap pasangan, tentukan apakah memakai pewarisan atau atribut. *(Kasus fiktif.)*

- (a) `Dokter` dan `Pegawai`.
- (b) `Rumah` dan `Kamar`.
- (c) `Burung` dan `Hewan`.
- (d) `Pemain` dan `Senjata`.

1. Ucapkan "X adalah Y" untuk tiap pasangan.
2. (a) Dokter adalah pegawai → **pewarisan** (`Dokter(Pegawai)`).
3. (b) Rumah memiliki kamar → **atribut** (daftar kamar).
4. (c) Burung adalah hewan → **pewarisan**.
5. (d) Pemain memiliki senjata → **atribut**.

Jawaban: (a) pewarisan, (b) atribut, (c) pewarisan, (d) atribut.

Contoh 5 — Membaca diagram tiga tingkat

Diagram: `Kendaraan` (atribut nama, kecepatan; method info(), gerak()), `Sepeda(Kendaraan)` (atribut gigi; method ganti_gigi()), `SepedaListrik(Sepeda)` (atribut baterai; method isi_daya()). Apa yang dimiliki objek `sl = SepedaListrik()`, dan dapatkah objek `sp = Sepeda()` memanggil `isi_daya()`? *(Kasus fiktif.)*

1. Objek `sl` memiliki semua dari tiga tingkat: atribut nama, kecepatan, gigi, baterai; method info(), gerak(), ganti_gigi(), isi_daya().
2. `sl.info()` menjalankan `info()` dari `Kendaraan` (diwarisi dua tingkat).
3. `sp = Sepeda()` hanya memiliki milik `Sepeda` dan `Kendaraan`. `isi_daya()` ada di `SepedaListrik`, yang **turunan** Sepeda, bukan induknya.
4. Karena itu `sp.isi_daya()` tidak dapat dipanggil.

Jawaban: `sl` memiliki seluruh anggota ketiga class; `sp` tidak dapat memanggil `isi_daya()`.

Contoh 6 — Merancang hierarki dari sebuah kebutuhan

Sebuah aplikasi perpustakaan sekolah (2027) mengelola tiga jenis anggota: siswa, guru, dan pustakawan. Semuanya memiliki nama dan nomor anggota serta dapat meminjam buku. Siswa

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

dapat meminjam paling banyak 3 buku, guru 5 buku, dan pustakawan tidak dibatasi. Rancang hierarki class-nya. *(Kasus fiktif.)*

1. Tentukan yang **sama** pada semua jenis anggota: atribut `nama`, `nomor`, dan method `pinjam()`. Bagian ini ditulis sekali di class induk `Anggota`.
2. Tentukan yang **berbeda**: batas pinjam (3, 5, tak terbatas). Ini menjadi atribut atau method yang ditulis ulang di subclass.
3. Uji hubungan: siswa adalah anggota, guru adalah anggota, pustakawan adalah anggota → pewarisan tepat.
4. Susun class:

```
class Anggota:
    batas = 3

    def __init__(self, nama, nomor):
        self.nama = nama
        self.nomor = nomor

    def boleh_pinjam(self, jumlah_sekarang):
        return jumlah_sekarang < self.batas

class Siswa(Anggota):
    pass

class Guru(Anggota):
    batas = 5

class Pustakawan(Anggota):
    def boleh_pinjam(self, jumlah_sekarang):
        return True

print(Siswa("A", 1).boleh_pinjam(3), Guru("B", 2).boleh_pinjam(3), Pustakawan("C",
3).boleh_pinjam(99))
```

1. Telusuri: `Siswa` memakai `batas = 3` dari induk; `3 < 3` salah → `False`. `Guru` mengganti `batas` menjadi 5; `3 < 5` benar → `True`. `Pustakawan` menulis ulang method sehingga selalu `True`.

Jawaban: `False True True`. Hal yang sama ditulis sekali di `Anggota`; hal yang berbeda ditulis di subclass.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

❖ PETA KONSEP

Gambar 5.15 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- `class Anak(Induk)`: subclass otomatis memiliki atribut dan method superclass; **arah pewarisan ke bawah**.
- **Pencarian method**: dari class objek, lalu naik ke induk.
- **Overriding**: subclass menulis ulang method dengan nama sama; `super().method()` memanggil versi induk.
- `super().__init__(...)` wajib dipanggil bila subclass menulis `__init__` dan masih membutuhkan atribut induk.
- **Is-a → pewarisan; has-a → atribut.**
- **Atribut privat induk tidak dapat dipakai langsung di subclass**; pakai method induk.
- **Objek subclass juga objek induknya** (`isinstance`).

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menelusuri soal pewarisan:

1. **Gambar hierarkinya** (siapa induk siapa) dan daftar anggota tiap class.
2. **Untuk membuat objek**, tentukan `__init__` mana yang dijalankan dan apakah `super().__init__` dipanggil dengan argumen yang benar.
3. **Untuk memanggil method**, cari dari class objek lalu naik; tentukan versi mana yang dipakai.
4. **Untuk diagram**, gabungkan isi kotak subclass dengan isi seluruh kotak induknya.

Jebakan yang sering dipasang:

- **`super().__init__` terlewat**, sehingga atribut induk tidak ada (`AttributeError`).
- **Urutan atribut dikira dikirim sendiri** padahal harus dikirim lewat `super().__init__(...)` dengan argumen yang cocok.
- **Arah pewarisan terbalik**: induk dikira memiliki anggota turunannya.
- **Method yang tidak tertulis di kotak subclass** dikira tidak dimiliki (padahal diwarisi).
- **Override dikira menghapus method induk**, padahal `super()` masih dapat memanggilnya.
- **Pewarisan dipakai untuk hubungan "memiliki"** (Mobil mewarisi Mesin).
- **Atribut privat induk** dikira dapat diubah langsung dari subclass.
- **Keluaran dua pemanggilan method dijumlahkan sebelum diperiksa**, padahal harus dihitung versi yang benar untuk tiap objek.

> Contoh salah tafsir: pada `class Penyihir(Karakter)` dengan `__init__` sendiri yang memanggil `super().__init__(nama, 80)`, seorang siswa menyimpulkan "objek Penyihir tidak punya `info()` karena `info()` tidak ditulis di class Penyihir". `info()` diwarisi dari `Karakter`, jadi `Penyihir("Laras").info()` berfungsi dan memakai atribut `nama` serta `nyawa` yang diisi oleh `super().__init__`.

● BEDAH SOAL PREDIKSI

Soal berikut memperlihatkan subclass yang tidak menulis ulang sebuah method induk dan memakai `super().__init__` dengan nilai tetap. Perhatikan urutan pembentukan atribut dan pencarian method yang naik ke induk (bagian 2 dan 4).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Tim gim *Legenda Nusantara* (2027) membuat class berikut. Class Penyihir tidak menulis ulang method info().

```
class Karakter:
    def __init__(self, nama, nyawa):
        self.nama = nama
        self.nyawa = nyawa

    def info(self):
        return self.nama + " " + str(self.nyawa)

class Penyihir(Karakter):
    def __init__(self, nama):
        super().__init__(nama, 80)
        self.mana = 50

p = Penyihir("Laras")
print(p.info(), p.mana)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. `Laras 50 80`
B. Error, karena Penyihir tidak memiliki method `info()`
C. `Laras 80 50`
D. Error, karena nilai nyawa tidak dikirim saat membuat objek
E. `Laras 80` saja

Pembahasan **Kunci: C**

Konsep: Subclass mewarisi atribut dan method superclass. `super().__init__(...)` menjalankan konstruktor superclass dari dalam konstruktor subclass, sehingga atribut superclass ikut terisi.

Langkah:

1. Penyihir("Laras") → konstruktor Penyihir: super().__init__("Laras", 80) → nama = "Laras", nyawa = 80.
2. Lalu self.mana = 50.
3. p.info() diwarisi dari Karakter → "Laras 80".
4. print("Laras 80", 50) → Laras 80 50.

Analisis setiap opsi:

- **A.** ✗ Urutan tertukar: info() menampilkan nama lalu nyawa (80), baru kemudian mana (50).
- **B.** ✗ Jebakan: Penyihir mewarisi info() dari Karakter, jadi method itu ada.
- **C.** ✓ info() menghasilkan "Laras 80", lalu p.mana = 50 dicetak di sebelahnya.
- **D.** ✗ Jebakan: nilai nyawa 80 dikirim oleh konstruktor Penyihir lewat super().
- **E.** ✗ print mencetak dua nilai: hasil info() dan p.mana.

Jawaban: C.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk setiap soal, gambar hierarki class dan tulis anggota yang dimiliki tiap objek sebelum menjawab.

1. Aplikasi penggajian sebuah studio gim (2027) memakai class berikut. Manajer mendapat bonus pegawai biasa ditambah tunjangan tetap.

```
class Pegawai:
    def __init__(self, nama, gaji):
        self.nama = nama
        self.gaji = gaji

    def bonus(self):
        return self.gaji // 10

class Manajer(Pegawai):
    def bonus(self):
        return super().bonus() + 500000

m = Manajer("Sinta", 6000000)
p = Pegawai("Doni", 4000000)
print(m.bonus() + p.bonus())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

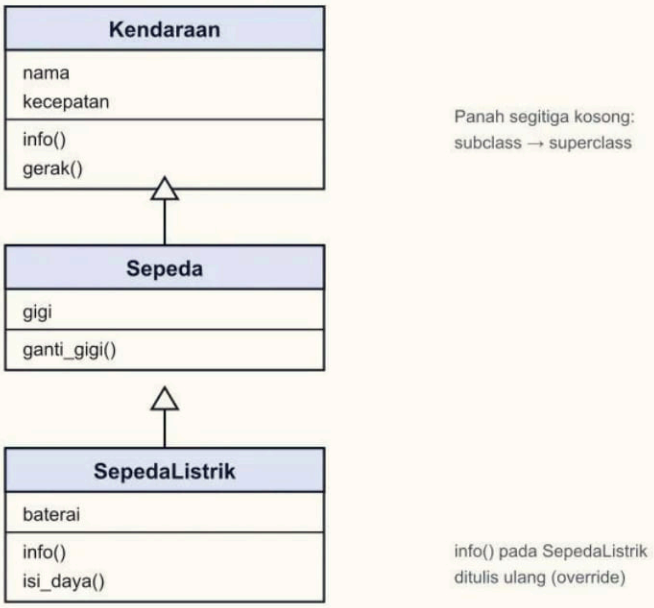
A. `900000`
B. `1500000`
C. `1000000`
D. `2000000`
E. Error, karena Manajer tidak memiliki `__init__`

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Gambar menunjukkan diagram class untuk fitur sewa kendaraan pada gim *Kota Hijau*. Programmer akan membuat objek `sl = SepedaListrik()` dan objek `sp = Sepeda()` sesuai diagram tersebut.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat berdasarkan diagram adalah ....

Diagram class aplikasi sewa kendaraan gim Kota Hijau (2027)



A. Objek sl memiliki atribut nama, kecepatan, gigi, dan baterai
B. Objek sl tidak dapat memanggil ganti_gigi() karena method itu milik Sepeda
C. Pemanggilan sl.info() menjalankan info() yang tertulis di class Kendaraan
D. Objek sp dapat memanggil isi_daya() karena Sepeda adalah induk SepedaListrik
E. Objek sl tidak memiliki gerak() karena method itu tidak tertulis di kotaknya

Kunci singkat: 1. B 2. A

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✔ PERIKSA DIRI

- Mengapa objek subclass dapat memanggil method yang tidak tertulis di kotak subclass pada diagram class?
- Apa yang terjadi pada atribut induk bila subclass menulis `__init__` tetapi tidak memanggil `super().__init__(...)`?
- **Yang harus kuingat:** `class Anak(Induk)`; method dicari dari class objek lalu naik ke induk; `super().__init__` mengisi atribut induk; override menulis ulang method dengan nama sama; is-a pakai pewarisan, has-a pakai atribut.



Latihan interaktif di LMS

Sub bab 5.3: kerjakan latihannya dan lihat progresmu.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-5.3

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab 5.4 · Polymorphism

SMKPPLG-5.4

◎ APA YANG DIUJI

Kisi-kisi resmi meminta kamu **mengimplementasikan polymorphism melalui method overriding atau referensi superclass terhadap objek subclass**. Contoh soal resmi Pusmendik untuk bagian ini berbentuk soal pilih lebih dari satu (PGK MCMA): ada tiga class (A sebagai induk, B dan C sebagai turunan yang menulis ulang method yang sama), sebuah fungsi yang memanggil method itu pada objek-objek dalam perulangan, dan kamu menandai pernyataan yang benar.

Inilah bentuk khas soal polymorphism: **sebuah daftar berisi objek dari beberapa class yang berkerabat, lalu sebuah perulangan memanggil satu method pada semuanya**. Kamu harus menentukan, untuk setiap objek, **versi method mana** yang dijalankan, lalu menjumlahkan atau menyusun hasilnya. Seluruh soal polymorphism pada paket latihan buku ini tergolong sulit, tetapi pola penyelesaiannya sama dan bisa dilatih: **tabel jejak dengan satu baris per objek.**

Setelah mempelajari sub bab ini, kamu dapat:

- menjelaskan polymorphism sebagai satu pemanggilan yang menghasilkan perilaku berbeda menurut objeknya;
- menulis ulang method (*overriding*) pada subclass dan menjelaskan versi mana yang dijalankan;
- menelusuri perulangan yang memanggil satu method pada daftar objek campuran;
- menjelaskan bahwa subclass yang tidak menulis ulang method memakai versi induknya;
- mengganti percabangan `if-elif` panjang dengan rancangan polimorfik dan menjelaskan manfaatnya.

◆ KONSEP INTI

Kata *polymorphism* berasal dari bahasa Yunani yang berarti "banyak bentuk". Dalam pemrograman, artinya **satu nama method dapat berperilaku berbeda tergantung pada objek yang memanggilnya**. Sudah dua kali kamu melihat dasarnya: pada sub bab 5.1 (empat prinsip OOP) dan 5.3 (overriding). Sekarang kamu memakainya sepenuhnya.

1. Satu pemanggilan, banyak perilaku

Bayangkan permainan kebun hewan. Setiap hewan dapat mengeluarkan suara, tetapi suaranya berbeda. Kamu ingin cukup menyuruh "bersuaralah" kepada semua hewan tanpa memeriksa jenisnya satu per satu.

```
class Hewan:
    def suara(self):
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
        return "..."

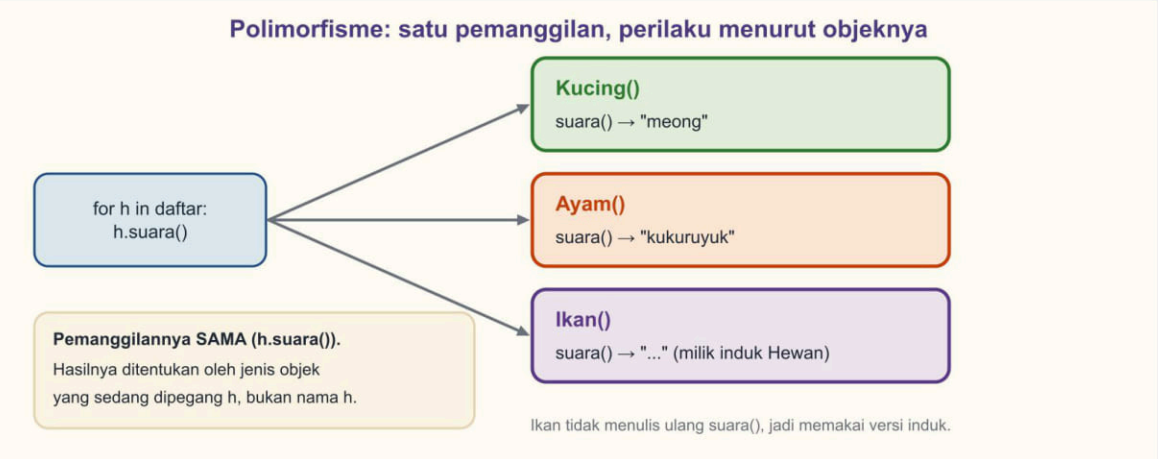
class Kucing(Hewan):
    def suara(self):
        return "meong"

class Ayam(Hewan):
    def suara(self):
        return "kukuruyuk"

class Ikan(Hewan):
    pass

for h in [Kucing(), Ayam(), Ikan(), Kucing()]:
    print(h.suara())
```

Keluarannya empat baris: `meong`, `kukuruyuk`, `...`, `meong`. Perhatikan bahwa **pemanggilan `h.suara()` selalu sama**, tetapi hasilnya berbeda menurut jenis objek yang dipegang `h` (Gambar 5.16). Itulah polimorfisme.



Gambar 5.16 Pemanggilan yang sama menghasilkan perilaku berbeda menurut jenis objek. Sumber: digambar oleh Tim Next Level Study Indonesia.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Ada dua hal yang membuatnya mungkin:

1. **Method overriding**: subclass menulis ulang method induk dengan **nama dan parameter yang sama** tetapi perilaku berbeda (`Kucing.suara`, `Ayam.suara`).
2. **Referensi superclass terhadap objek subclass**: sebuah daftar atau parameter yang diperlakukan sebagai "Hewan" dapat berisi objek `Kucing`, `Ayam`, dan `Ikan`, karena masing-masing **adalah** Hewan (sub bab 5.3).

Satu kasus yang sering dipertanyakan: `Ikan` tidak menulis ulang `suara()`. Apakah program galat? **Tidak.** Python mencari method dari class objek itu sendiri lalu naik ke induk, sehingga `Ikan().suara()` menjalankan versi `Hewan` dan menghasilkan `...`. Subclass yang tidak menulis ulang method otomatis memakai **versi induknya**.

2. Menelusuri perulangan polimorfik

Cara menjawab soal polimorfisme adalah menelusuri **tiap objek dalam daftar**, menentukan class aslinya, mencari versi method yang dipakai, lalu mencatat hasilnya. Pakailah tabel jejak (Gambar 5.17).

```
class Pegawai:
    def __init__(self, gaji):
        self.gaji = gaji

    def tunjangan(self):
        return self.gaji // 10

class Manajer(Pegawai):
    def tunjangan(self):
        return self.gaji // 5

class Magang(Pegawai):
    def tunjangan(self):
        return 0

tim = [Pegawai(4000000), Manajer(6000000), Magang(2000000), Pegawai(2500000)]
total = 0
for p in tim:
    total += p.tunjangan()
print(total)
```

Telusuri satu per satu:

1. `Pegawai(4000000)`: memakai `tunjangan` milik `Pegawai` → 4.000.000 // 10 = 400.000. Total 400.000.
2. `Manajer(6000000)`: `Manajer` menulis ulang `tunjangan` → 6.000.000 // 5 = 1.200.000. Total 1.600.000.
3. `Magang(2000000)`: `Magang` menulis ulang → 0. Total tetap 1.600.000.
4. `Pegawai(2500000)`: versi `Pegawai` → 250.000. Total 1.850.000.

Keluarannya `1850000`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Menelusuri loop polimorfik dengan tabel jejak

```
class Pegawai:
    def __init__(self, gaji): self.gaji = gaji
    def tunjangan(self): return self.gaji // 10
class Manajer(Pegawai):
    def tunjangan(self): return self.gaji // 5
class Magang(Pegawai):
    def tunjangan(self): return 0
tim = [Pegawai(4000000), Manajer(6000000),
       Magang(2000000), Pegawai(2500000)]
total = 0
for p in tim: total += p.tunjangan()
```

m	method yang dipakai	hasil	total
Pegawai(4jt)	versi Pegawai (induk)	4000000 // 10 = 400000	**400000**
Manajer(6jt)	versi Manajer (override)	6000000 // 5 = 1200000	**1600000**
Magang(2jt)	versi Magang (override)	0	**1600000**
Pegawai(2,5jt)	versi Pegawai (induk)	250000	**1850000**

Keluaran: 1850000

Pegawai biasa memakai method induk; Manajer dan Magang memakai versi miliknya. Pemanggilannya sama: p.tunjangan().

Gambar 5.17 Tabel jejak untuk perulangan yang memanggil method yang sama pada objek berbeda. Sumber: digambar oleh Tim Next Level Study Indonesia.

Mengapa cara ini bekerja? Pada saat program berjalan, Python tidak melihat nama variabel (p), melainkan **jenis objek yang sebenarnya dipegangnya**. Karena itu, baris `p.tunjangan()` yang sama persis di dalam perulangan dapat menjalankan tiga method berbeda pada tiga putaran. Pemilihan versi method seperti ini terjadi **saat program dijalankan**, putaran demi putaran, bukan saat kode ditulis.

Tiga hal yang harus kamu periksa pada setiap objek:

- **Class aslinya** (yang tertulis saat objek dibuat), bukan class induknya.
- **Apakah class itu menulis ulang method?** Bila ya, versi itu yang dipakai; bila tidak, naik ke induk.
- **Argumen yang dikirim saat method dipanggil** dan atribut yang dipakai method itu.

3. Mengganti percabangan dengan polimorfisme

Polimorfisme sering dipakai untuk menggantikan rangkaian `if-elif` yang bercabang menurut **jenis** sesuatu. Perhatikan contoh yang berat dirawat berikut:

```
def gerak(jenis):
    if jenis == "gajah":
        return "berjalan"
    elif jenis == "burung":
        return "terbang"
    elif jenis == "ikan":
        return "berenang"
    else:
        return "diam"
```

Tiap kali ada jenis baru, fungsi `gerak` harus diubah. Bila ada fungsi lain yang juga bercabang menurut jenis (suara, makan, tidur), semuanya harus diubah serentak. Satu perubahan yang terlewat atau salah dapat **merusak perilaku jenis lama**. Rancangan polimorfik memecahnya (Gambar 5.18):

```
class Hewan:
    def gerak(self):
        return "diam"

class Gajah(Hewan):
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
    def gerak(self):
        return "berjalan"

class Burung(Hewan):
    def gerak(self):
        return "terbang"

class Ikan(Hewan):
    def gerak(self):
        return "berenang"

for h in [Gajah(), Burung(), Ikan(), Hewan()]:
    print(h.gerak())
```

Keluarannya `berjalan`, `terbang`, `berenang`, `diam`. Untuk menambah `Katak`, cukup menulis **class baru** dengan `gerak()`-nya; perulangan di bawah **tidak perlu diubah sama sekali**. Itulah manfaat utama: **menambah perilaku baru tanpa mengubah kode lama yang sudah berjalan.**



Gambar 5.18 Percabangan menurut jenis dibandingkan dengan rancangan polimorfik. Sumber: digambar oleh Tim Next Level Study Indonesia.

Perlu dicatat juga bahwa kelas induk pada rancangan ini (`Hewan`) menyediakan perilaku **bawaan** (`"diam"`), yang dipakai oleh jenis yang belum menulis ulang. Pola ini aman: jenis baru yang lupa menulis `gerak()` tidak membuat program galat, hanya berperilaku bawaan. Bila induk dimaksudkan semata-mata sebagai kerangka, method-nya dapat dibiarkan kosong atau memberi pesan bahwa subclass harus menulis ulangnya.

Rancangan yang **hanya memindahkan** `if–elif` ke dalam satu method (misalnya satu class `Hewan` dengan atribut `jenis`, lalu `gerak()` bercabang menurut `jenis`) **tidak** menyelesaikan masalah: jenis baru tetap memaksa kamu mengubah method itu. Rancangan yang tepat menjadikan **tiap jenis sebuah subclass yang menulis ulang method**.

4. Referensi superclass, objek subclass

Karena objek subclass **adalah** juga objek superclass-nya, sebuah variabel, daftar, atau parameter fungsi yang diperlakukan sebagai superclass dapat menerima objek turunan apa

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

pun (Gambar 5.19).

```
class Alarm:
    def bunyi(self):
        return "tut"

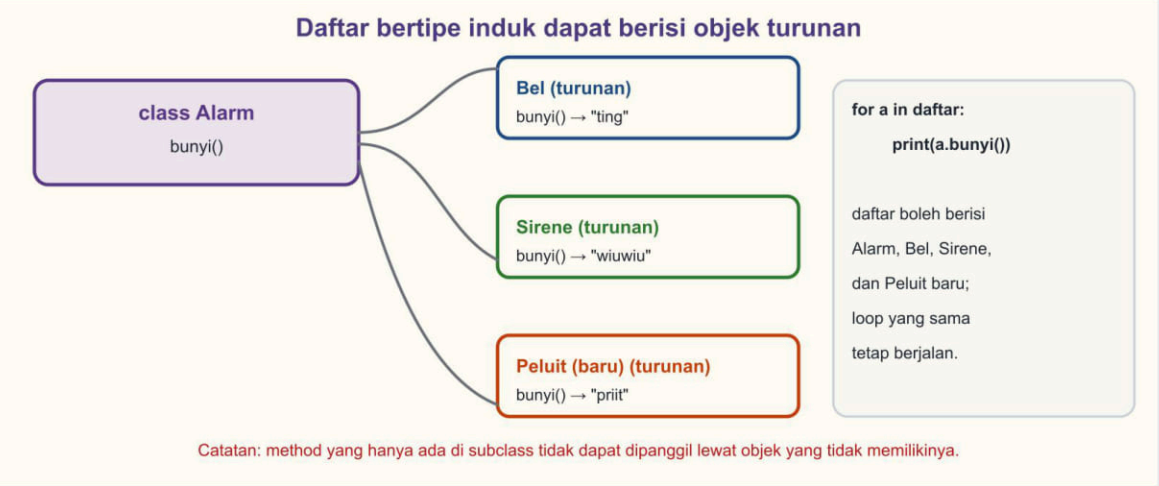
class Bel(Alarm):
    def bunyi(self):
        return "ting"

class Sirene(Alarm):
    def bunyi(self):
        return "wiu" * 2

def nyalakan(alat):
    print(alat.bunyi())

nyalakan(Alarm())
nyalakan(Bel())
nyalakan(Sirene())
```

Keluarannya `tut`, `ting`, dan `wiuwiu`. Fungsi `nyalakan` **tidak peduli** class sebenarnya; ia hanya memanggil `bunyi()`. Bila kelak ditambahkan class `Peluit`, fungsi `nyalakan` tidak perlu diubah.



Gambar 5.19 Sebuah daftar atau fungsi bertipe induk dapat memakai objek turunan apa pun. Sumber: digambar oleh Tim Next Level Study Indonesia.

Ada **batasnya**: lewat referensi induk, kamu hanya dapat memanggil **method yang ada pada class objek itu atau induknya**. Bila sebuah method hanya ada pada satu subclass (misalnya `lompat()` hanya milik `Kucing`), memanggilnya pada objek `Hewan` atau `Ikan` menimbulkan `AttributeError`. Karena itu, method yang ingin dipakai secara polimorfik sebaiknya **dideklarasikan di superclass**, lalu ditulis ulang di subclass.

5. Python dan duck typing, serta tidak adanya overloading

Python juga mengenal bentuk polimorfisme yang lebih longgar, disebut ***duck typing***: "bila berjalan seperti bebek dan bersuara seperti bebek, perlakukan ia sebagai bebek". Perulangan akan bekerja pada objek apa pun yang memiliki method bernama sesuai, **walaupun tidak mewarisi class yang sama**. Dalam soal TKA, bentuk yang diujikan adalah yang berbasis

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

pewarisan seperti di atas.

Satu pelurusan lain: **Python tidak memiliki overloading** (beberapa method bernama sama dengan parameter berbeda dalam satu class seperti pada beberapa bahasa lain). Bila dua method bernama sama ditulis dalam satu class, yang kedua menimpa yang pertama. Variasi jumlah argumen ditangani dengan **parameter bernilai bawaan**. Jadi, "polimorfisme" di sini selalu berarti **overriding** (subclass menulis ulang method induk), bukan overloading.

5b. Polimorfisme bersama super() dan kapan ia tepat

Polimorfisme dapat digabung dengan `super()` (sub bab 5.3): subclass menulis ulang sebuah method **dan** memanggil versi induk di dalamnya, sehingga perilaku induk tetap berjalan lalu ditambah hal baru. Misalnya, `Manajer.tunjangan()` dapat ditulis `return super().tunjangan() + 500000`. Untuk menelusurinya, hitung dulu hasil `super().tunjangan()` lalu tambahkan bagian milik subclass.

Kapan polimorfisme tepat dipakai? Ketika kamu memiliki **beberapa jenis benda yang melakukan hal yang sama dengan cara berbeda**, dan program harus memperlakukan semuanya secara seragam: menghitung gaji berbagai jenis pegawai, menggambar berbagai jenis bentuk, mengirim notifikasi lewat berbagai saluran, atau menggerakkan berbagai jenis musuh. Sebaliknya, bila hanya ada satu jenis benda, atau perilakunya identik, polimorfisme tidak diperlukan. Rancangan yang baik tidak memaksakan pewarisan dan polimorfisme di tempat yang tidak membutuhkannya.

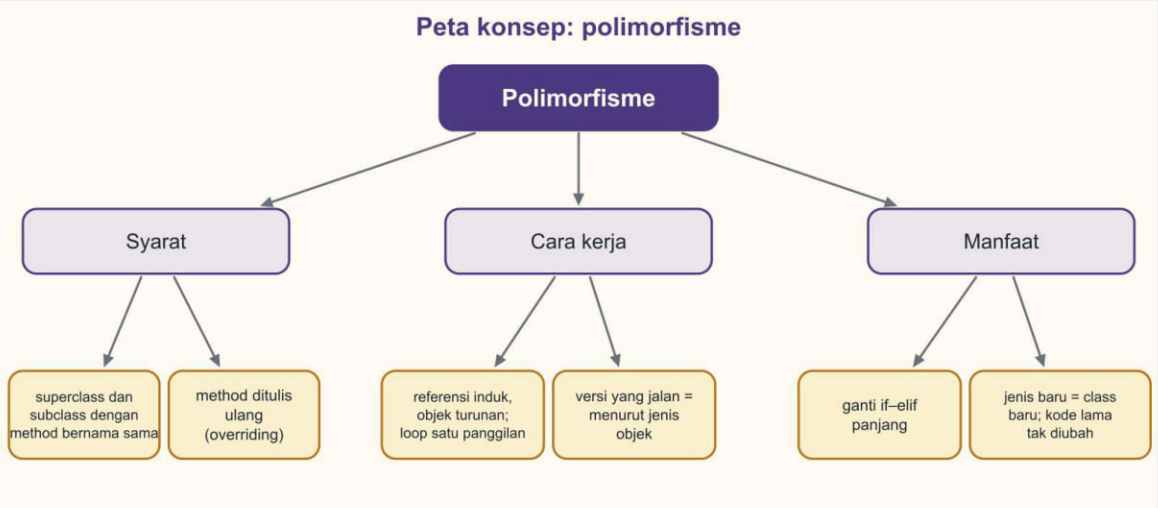
6. Ciri polimorfisme dalam sebuah rancangan

Saat sebuah rancangan ditanyakan prinsipnya, polimorfisme dikenali dari ciri-ciri ini:

- ada **induk dan beberapa turunan** yang menulis ulang method yang sama;
- ada **satu pemanggilan** (dalam perulangan atau fungsi) yang berlaku untuk banyak jenis objek;
- **hasil berbeda menurut jenis objek**;
- **menambah jenis baru tidak mengubah kode pemanggil**.

Bedakan dengan prinsip lain: bila yang ditonjolkan adalah **menyembunyikan rincian**, itu abstraksi; **melindungi data**, enkapsulasi; **mengambil bagian class lain**, pewarisan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan



Gambar 5.20 Peta konsep polimorfisme. Sumber: digambar oleh Tim Next Level Study Indonesia.

CONTOH TERSELESAIKAN

Contoh 1 — Menelusuri perulangan dengan satu versi diwarisi

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
class Kasir:
    def potongan(self, belanja):
        return belanja // 20

class KasirMember(Kasir):
    def potongan(self, belanja):
        return belanja // 10

class KasirTamu(Kasir):
    pass

total = 0
for k in [Kasir(), KasirMember(), KasirTamu(), KasirMember()]:
    total += k.potongan(100000)
print(total)
```

1. `Kasir()`: 100000 // 20 = 5000. Total 5000.
2. `KasirMember()`: ditulis ulang → 100000 // 10 = 10000. Total 15000.
3. `KasirTamu()`: tidak menulis ulang → memakai versi `Kasir` → 5000. Total 20000.
4. `KasirMember()`: 10000. Total 30000.

Jawaban: `30000`. `KasirTamu` tidak menimbulkan galat; ia memakai versi induknya.

Contoh 2 — Mengubah if–elif menjadi class polimorfik

Fungsi berikut menghitung biaya kirim menurut jenis layanan. Ubah menjadi rancangan polimorfik dan tentukan keluaran untuk daftar ["hemat", "ekspres", "hemat"] dengan berat 3 kg. *(Kasus fiktif.)*

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

```
def biaya(jenis, kg):
    if jenis == "hemat":
        return kg * 4000
    elif jenis == "ekspres":
        return kg * 9000 + 5000
    else:
        return kg * 6000
```

1. Tiap jenis menjadi subclass dari `Layanan`, dengan method `biaya(kg)` yang ditulis ulang.
2. Kode:

```
class Layanan:
    def biaya(self, kg):
        return kg * 6000

class Hemat(Layanan):
    def biaya(self, kg):
        return kg * 4000

class Ekspres(Layanan):
    def biaya(self, kg):
        return kg * 9000 + 5000

paket = [Hemat(), Ekspres(), Hemat()]
total = 0
for p in paket:
    total += p.biaya(3)
print(total)
```

1. `Hemat().biaya(3)` = 12000; `Ekspres().biaya(3)` = 27000 + 5000 = 32000; `Hemat().biaya(3)` = 12000.
2. Total: 12000 + 32000 + 12000 = **56000**.
3. Untuk menambah layanan baru, tulis class baru; perulangan tidak diubah.

Jawaban: `56000`. Tiap jenis menjadi class, dan perilaku baru tidak mengubah kode lama.

Contoh 3 — Fungsi yang menerima objek induk

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
class AlatBayar:
    def bayar(self, jumlah):
        return jumlah

class Kartu(AlatBayar):
    def bayar(self, jumlah):
        return jumlah + jumlah // 50

class Voucher(AlatBayar):
    def bayar(self, jumlah):
        return max(jumlah - 10000, 0)

def checkout(alat, jumlah):
    return alat.bayar(jumlah)

print(checkout(AlatBayar(), 80000), checkout(Kartu(), 80000), checkout(Voucher(),
8000))
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

1. `checkout(AlatBayar(), 80000)` → 80000.
2. `checkout(Kartu(), 80000)` → 80000 + 80000 // 50 = 80000 + 1600 = 81600.
3. `checkout(Voucher(), 8000)` → `max(8000 - 10000, 0)` = `max(-2000, 0)` = 0.

Jawaban: `80000 81600 0`. Fungsi `checkout` tidak berubah walaupun alat bayarnya berbeda-beda.

Contoh 4 — Method khusus subclass lewat daftar induk

Perhatikan program berikut, lalu tentukan apa yang terjadi. *(Kasus fiktif.)*

```
# tidak dijalankan (menimbulkan AttributeError pada objek kedua)
class Kendaraan:
    def jalan(self):
        return "bergerak"

class Pesawat(Kendaraan):
    def terbang(self):
        return "mengudara"

for k in [Pesawat(), Kendaraan()]:
    print(k.jalan())
    print(k.terbang())
```

1. Objek pertama `Pesawat()`: `jalan()` diwarisi → `bergerak`; `terbang()` ada di `Pesawat` → `mengudara`.
2. Objek kedua `Kendaraan()`: `jalan()` → `bergerak`; `terbang()` **tidak ada** di `Kendaraan` (hanya milik turunan) → `AttributeError`.
3. Perbaikan: bila `terbang()` ingin dipakai secara polimorfik, tulis di `Kendaraan` (misalnya mengembalikan `"tidak bisa terbang"`) lalu tulis ulang di `Pesawat`.

Jawaban: dua baris pertama tercetak untuk `Pesawat`, kemudian `bergerak` untuk `Kendaraan`, lalu `AttributeError`. Method yang dipakai polimorfik harus ada di induk.

Contoh 5 — Overriding dengan super() dalam perulangan

Tentukan keluaran program berikut. *(Kasus fiktif.)*

```
class Serangan:
    def kerusakan(self, dasar):
        return dasar

class Api(Serangan):
    def kerusakan(self, dasar):
        return super().kerusakan(dasar) + 5

class Es(Serangan):
    def kerusakan(self, dasar):
        return super().kerusakan(dasar) * 2

total = 0
for s in [Serangan(), Api(), Es(), Api()]:
    total += s.kerusakan(10)
print(total)
```

1. `Serangan()`: versi induk → 10. Total 10.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. `Api()`: `super().kerusakan(10)` = 10, ditambah 5 → 15. Total 25.
3. `Es()`: `super().kerusakan(10)` = 10, dikali 2 → 20. Total 45.
4. `Api()`: 15. Total 60.

Jawaban: `60`. Pada `Api` dan `Es`, versi induk tetap dijalankan di dalam versi subclass lewat `super()`, lalu hasilnya diubah.

Contoh 6 — Menilai tiga rancangan

Sebuah aplikasi menggambar harus menghitung luas beberapa bentuk (persegi, segitiga, lingkaran), dan kelak bentuk baru akan terus ditambah. Tiga rancangan diusulkan. *(Kasus fiktif.)*

- (A) Satu fungsi `luas(jenis, ukuran)` dengan `if-elif` untuk tiap bentuk.
- (B) Satu class `Bentuk` dengan atribut `jenis` dan method `luas()` yang bercabang `if-elif` menurut `jenis`.
- (C) Class `Bentuk` dengan method `luas()`, lalu subclass `Persegi`, `Segitiga`, `Lingkaran` yang masing-masing menulis ulang `luas()`.

1. Tanyakan: bila bentuk baru ditambah, apa yang harus diubah?
2. (A): fungsi `luas` harus ditambah cabang `elif`. Kode lama ikut disentuh.
3. (B): method `luas()` harus ditambah cabang. Percabangan hanya dipindah tempat.
4. (C): cukup menulis subclass baru. Kode lama dan perulangan pemanggil **tidak disentuh**.
5. Hanya (C) yang menerapkan polimorfisme dan memenuhi tujuan "mudah ditambah tanpa mengubah kode lama".

Jawaban: rancangan (C). Rancangan (B) tampak berorientasi objek tetapi tetap bercabang menurut jenis sehingga tidak menyelesaikan masalah.

❖ PETA KONSEP

Gambar 5.20 di atas merangkum konsep sub bab ini. Yang harus kamu ingat:

- **Polimorfisme = satu pemanggilan, perilaku berbeda menurut jenis objek**, lewat **overriding** dan **referensi superclass terhadap objek subclass**.
- **Versi method yang jalan** ditentukan oleh **class asli objek**; subclass yang tidak menulis ulang **memakai versi induk**.
- **Telusuri tiap objek dalam daftar** dengan tabel jejak: class asli → versi method → hasil.
- Polimorfisme **menggantikan `if-elif` panjang menurut jenis**; **menambah jenis baru cukup menulis class baru**, tanpa mengubah kode pemanggil.
- Method yang dipakai polimorfik **harus ada di superclass**; method khusus subclass tidak dapat dipanggil lewat objek induk.
- **Python tidak punya overloading**; polimorfisme berarti overriding.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

▲ STRATEGI DAN JEBAKAN

Langkah menelusuri soal polimorfisme:

1. **Gambar hierarki class**: siapa induk siapa, dan method mana yang ditulis ulang.
2. **Untuk tiap objek dalam daftar**, tulis class aslinya (nama sesudah `class` saat dibuat).
3. **Tentukan versi method**: ditulis ulang di class itu? Bila tidak, naik ke induk.
4. **Hitung hasil dengan argumen yang benar** dan perbarui total atau keluaran.
5. **Perhatikan perintah di luar perulangan** (cetak sekali, di akhir).

Jebakan yang sering dipasang:

- **Semua objek dikira memakai method induk** karena "daftar bertipe induk".
- **Subclass yang tidak menulis ulang method dikira galat**, padahal memakai versi induk.
- **Hasil dihitung dari class induk** untuk objek turunan yang menulis ulang method.
- **Method khusus subclass** dikira dapat dipanggil pada semua objek dalam daftar.
- **Overloading dikira ada di Python.**
- **`if-elif` dipindah ke satu method** dikira sama baiknya dengan polimorfisme.
- **Menambah subclass dikira mengharuskan mengubah perulangan.**
- **Prinsip tertukar**: pewarisan (mengambil milik induk) dan polimorfisme (perilaku berbeda) dikira sama.

> Contoh salah tafsir: pada `for m in [Musuh(), Golem()]: total += m.kena(9)` dengan `Golem` yang menulis ulang `kena`, seorang siswa menghitung kedua objek memakai `Musuh.kena` karena "listnya berisi Musuh". Objek kedua adalah `Golem`, sehingga versi `Golem.kena` yang dipakai. Yang menentukan adalah **class asli objek**, bukan nama umum di daftar.

● BEDAH SOAL PREDIKSI

Soal berikut memperlihatkan perulangan atas tiga jenis objek dengan satu jenis yang tidak menulis ulang method. Perhatikan cara tabel jejak menentukan versi method tiap objek, dan bagaimana subclass yang tidak menulis ulang tetap berjalan tanpa galat (bagian 1 dan 2).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

►. Dalam gim strategi *Benteng Terakhir* (2027), setiap musuh menerima serangan 9 poin dengan cara berbeda. Programmer menulis kode berikut untuk menghitung total kerusakan pada satu gelombang musuh.

```
class Musuh:
    def kena(self, dmg):
        return dmg

class Golem(Musuh):
    def kena(self, dmg):
        return dmg // 2

class Hantu(Musuh):
    pass

total = 0
for m in [Musuh(), Golem(), Hantu(), Golem()]:
    total += m.kena(9)
print(total)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. 36
B. 17
C. 27
D. 26
E. Error, karena Hantu tidak punya method kena()

Pembahasan **Kunci: D**

Konsep: Polimorfisme: satu pemanggilan m.kena(9) menjalankan method sesuai class objeknya. Subclass yang menulis ulang method (overriding) memakai versinya sendiri; subclass yang tidak menulis ulang memakai method milik superclass.

Langkah:

1. Musuh() → kena dari Musuh: 9.
2. Golem() → kena dari Golem: 9 // 2 = 4.
3. Hantu() → tidak punya kena sendiri → memakai milik Musuh: 9.
4. Golem() → 4.
5. Total: 9 + 4 + 9 + 4 = 26.

Analisis setiap opsi:

- **A.** ✗ Jebakan: semua objek dianggap memakai kena milik Musuh: 4 × 9 = 36.
- **B.** ✗ Jebakan: Hantu dianggap tidak menghasilkan kerusakan: 9 + 4 + 0 + 4 = 17.
- **C.** ✗ Pembagian dianggap biasa (9 / 2 = 4,5): 9 + 4,5 + 9 + 4,5 = 27.
- **D.** ✓ Musuh 9, Golem 4, Hantu 9 (warisan), Golem 4 → 26.
- **E.** ✗ Hantu mewarisi kena() dari Musuh, jadi tidak terjadi error.

Jawaban: D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

✎ LATIHAN

Kerjakan tanpa melihat kunci. Untuk tiap soal, buat tabel jejak berisi class asli, versi method, dan hasil.

1. Aplikasi tagihan SPP (2027) mengirim notifikasi lewat beberapa saluran. Programmer menulis kode berikut.

```
class Notifikasi:
    def kirim(self, pesan):
        return "[umum] " + pesan

class Email(Notifikasi):
    def kirim(self, pesan):
        return "[email] " + pesan

class WhatsApp(Notifikasi):
    def kirim(self, pesan):
        return "[wa] " + pesan.upper()

for n in [Email(), WhatsApp(), Notifikasi()]:
    print(n.kirim("lunas"))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang kode tersebut adalah ....

(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Ketiga baris keluaran diawali [umum] karena semua objek berasal dari turunan Notifikasi.

☐ B. Baris kedua keluaran adalah `[wa] LUNAS`.

☐ C. Baris pertama keluaran adalah `[email] lunas`.

☐ D. Bila class SMS(Notifikasi) dengan kirim() sendiri ditambahkan ke list, perulangan for tidak perlu diubah.

☐ E. Bila class Email tidak menulis kirim(), objek `Email()` menghasilkan `[umum] lunas`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Gim *Lantai Ajaib* (2027) menghitung luas lantai satu level yang tersusun dari ubin persegi dan ubin segitiga siku-siku dengan sisi tegak sama panjang. Programmer menulis kode berikut.

```
class Ubin:
    def __init__(self, s):
        self.s = s

    def luas(self):
        return self.s * self.s

class Segitiga(Ubin):
    def luas(self):
        return self.s * self.s // 2

total = 0
for u in [Ubin(4), Segitiga(4), Segitiga(3)]:
    total += u.luas()
print(total)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 41
B. 28.5
C. 28
D. 29
E. Error, karena Segitiga tidak memiliki __init__

Kunci singkat: 1. B, C, D, E 2. C

✔ PERIKSA DIRI

- Mengapa `Ikan().suara()` tidak menimbulkan galat padahal class `Ikan` tidak menulis ulang `suara()`?
- Apa keuntungan rancangan polimorfik dibandingkan `if-elif` panjang ketika sebuah jenis baru ditambahkan?
- **Yang harus kuingat:** polimorfisme = satu pemanggilan, perilaku berbeda menurut class asli objek; versi method dicari dari class objek lalu naik ke induk; telusuri tiap objek dengan tabel jejak; menambah jenis baru cukup menulis class baru; Python tidak punya overloading.

Latihan interaktif di LMS

Sub bab 5.4: kerjakan latihannya dan lihat progresmu.
next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-5.4

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Kuis Akhir Bab 5

SMKPPLG-KB5

Jumlah soal: 25 **Waktu anjuran:** 60 menit

Kerjakan sendiri dan berwaktu. Setelah selesai, cocokkan dengan kunci di Bagian D, isi tabel skor, lalu lihat anjuran di bawahnya.

Kerjakan Kuis Bab 5 di LMS

Skor dan progresmu tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-KB5

1. Perhatikan potongan program Python berikut.

```
class Playlist:
    def __init__(self):
        self.lagu = []
    def tambah(self, judul):
        self.lagu.append(judul)
        return len(self.lagu)
    def hapus_pertama(self):
        if self.lagu:
            return self.lagu.pop(0)
        return None

p = Playlist()
n = p.tambah("A")
p.tambah("B")
x = p.hapus_pertama()
y = Playlist().hapus_pertama()
print(n, x, y, p.lagu)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Nilai n adalah 1 karena tambah() mengembalikan jumlah lagu saat itu

☐ B. Nilai y adalah "B" karena lagu kedua dihapus oleh pemanggilan terakhir

☐ C. Playlist() pada baris y adalah objek baru yang kosong, sehingga y bernilai None

☐ D. Pada akhir program, p.lagu berisi ["B"]

☐ E. Nilai x adalah "B" karena pop(0) mengambil elemen terakhir

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Perhatikan potongan program Python berikut.

```
class Meja:
    jumlah = 0
    def __init__(self, kode):
        self.kode = kode
        Meja.jumlah += 1

a = Meja("M1")
b = Meja("M2")
c = a
c.kode = "M3"
print(Meja.jumlah, a.kode, b.kode)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 2 M1 M2
B. 2 M3 M2
C. 3 M3 M2
D. 2 M3 M3
E. 3 M1 M2

3. Tim pengembang aplikasi pembelajaran bahasa (2027) merancang class `Penerjemah`. Programmer lain cukup memanggil `terjemah("halo")` tanpa perlu tahu bagaimana kamus dimuat dan bagaimana kalimat diolah. Class itu juga memiliki atribut `kuota` yang hanya dapat diubah lewat method yang memeriksa nilainya lebih dulu.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Rancangan itu menyembunyikan rincian rumit di balik antarmuka sederhana, yaitu penerapan abstraksi.	○	○
Rancangan itu adalah contoh pewarisan karena banyak bagian aplikasi memakai class Penerjemah.	○	○
Atribut kuota yang hanya dapat diubah lewat method yang memeriksa nilainya adalah penerapan enkapsulasi.	○	○

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. Perhatikan potongan program Python berikut.

```
class Speaker:
    def __init__(self):
        self.__volume = 30
    def set_volume(self, v):
        if 0 <= v <= 100:
            self.__volume = v
            return True
        return False
    def get_volume(self):
        return self.__volume

s = Speaker()
a = s.set_volume(120)
b = s.set_volume(100)
c = s.set_volume(-1)
print(a, b, c, s.get_volume())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. True True False 100
B. False True False 100
C. False True False 30
D. True True True 100
E. False False False 30

5. Perhatikan program Python berikut.

```
class Perabot:
    def __init__(self):
        self.__kode = "P-01"
    def info(self):
        return "Perabot " + self.__kode

class Lemari(Perabot):
    def info(self):
        return super().info() + " (lemari)"

for m in [Perabot(), Lemari()]:
    print(m.info())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

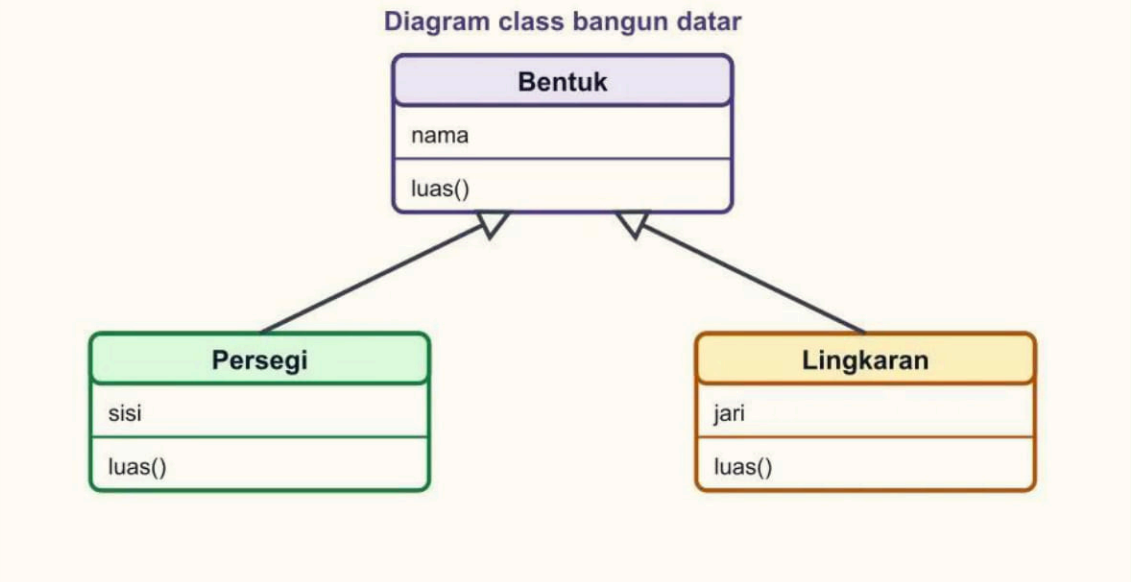
Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Atribut __kode tidak dapat diubah langsung dari luar class, ini penerapan enkapsulasi
☐ B. Keluaran baris kedua hanya Perabot P-01 karena info() milik Perabot tidak boleh diubah
☐ C. Lemari adalah subclass yang mewarisi atribut dan method Perabot
☐ D. Pemanggilan m.info() dalam perulangan menghasilkan hasil berbeda menurut jenis objek, ciri polimorfisme
☐ E. Pewarisan terjadi karena objek Perabot dan Lemari dimasukkan ke dalam satu list

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

6. Seorang siswa menggambar diagram class untuk program pengolah bangun datar (2027), seperti pada gambar. Panah dengan kepala segitiga kosong menunjuk dari subclass ke superclass. Ketiga class memiliki method `luas()` yang ditulis sendiri-sendiri pada tiap class.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Dua prinsip OOP yang paling jelas tampak pada gambar adalah ....



A. enkapsulasi dan abstraksi
B. abstraksi dan pewarisan
C. enkapsulasi dan polimorfisme
D. abstraksi dan polimorfisme
E. pewarisan dan polimorfisme

7. Perhatikan potongan program Python berikut.

```
class Kupon:
    def __init__(self, kode, sisa):
        self.kode = kode
        self.__sisa = sisa
    def pakai(self, n):
        if n > 0 and n <= self.__sisa:
            self.__sisa -= n
            return True
        return False
    def lihat_sisa(self):
        return self.__sisa

k = Kupon("DISK10", 3)
print(k.pakai(2), k.pakai(2), k.pakai(0), k.lihat_sisa())
k.kode = "BARU"
k.__sisa = 99
print(k.lihat_sisa(), k.kode)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Baris pertama yang dicetak adalah True False False 1
☐ B. Pernyataan k.__sisa = 99 mengubah sisa kupon menjadi 99
☐ C. Baris kedua yang dicetak adalah 1 BARU
☐ D. k.kode dapat diubah langsung dari luar class karena atribut itu publik
☐ E. Pemanggilan k.pakai(0) menurunkan sisa kupon karena 0 dikurangkan

8. Perhatikan potongan program Python berikut.

```
class Sepatu:
    def __init__(self, merek, ukuran):
        self.merek = merek
        self.ukuran = ukuran

s1 = Sepatu("Alfa", 40)
s2 = Sepatu("Beta", 42)
s2.ukuran -= 1
print(s1.ukuran, s2.ukuran)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 40 41
B. 40 42
C. 39 41
D. 39 42
E. 41 41

9. Seorang siswa menulis class untuk perangkat pencetak di lab (2027).

```
class Printer:
    def __init__(self, merek):
        merek = merek
        self.aktif = False
    def hidupkan(self):
        self.aktif = True

p = Printer("Zeta")
p.hidupkan()
print(p.aktif, p.merek)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Hasil dari menjalankan program tersebut adalah ....

A. True Zeta, karena merek dikirim saat objek dibuat
B. True None, karena atribut yang belum diisi bernilai None
C. AttributeError karena merek tidak disimpan sebagai atribut
D. TypeError karena hidupkan() tidak mengembalikan nilai
E. False Zeta, karena hidupkan() tidak mengubah apa pun

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

10. Perhatikan class berikut yang menyimpan suhu sebuah kulkas.

```
class Kulkas:
    def __init__(self):
        self.__suhu = 4
    def lihat_suhu(self):
        return self.__suhu

k = Kulkas()
print(k.lihat_suhu())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Prinsip OOP yang paling jelas diterapkan pada class tersebut adalah ....

A. abstraksi, karena suhu dihitung dengan rumus rumit di dalam class tersebut
B. pewarisan, karena method lihat_suhu diwarisi dari class lain yang lebih umum
C. polimorfisme, karena satu method menghasilkan beberapa bentuk keluaran berbeda
D. enkapsulasi, karena atribut dilindungi dan dibaca lewat method
E. bukan prinsip OOP, hanya cara menyimpan sebuah angka dalam variabel

11. Perhatikan potongan program Python berikut.

```
class Rak:
    def __init__(self):
        self.__isi = 5

r = Rak()
print(r.__isi)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Hasil dari menjalankan program tersebut adalah ....

A. AttributeError
B. 5
C. None
D. 0
E. TypeError

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

12. Perhatikan potongan program Python berikut.

```
class Pengirim:
    def __init__(self):
        self.terkirim = 0
    def __siapkan(self, pesan):
        return pesan.strip().upper()
    def kirim(self, pesan):
        data = self.__siapkan(pesan)
        self.terkirim += 1
        return data

p = Pengirim()
print(p.kirim("  halo "))
print(p.__siapkan("x"))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Hasil dari menjalankan program tersebut adalah ....

A. HALO lalu X
B. HALO lalu NameError
C. HALO lalu AttributeError
D. AttributeError tanpa mencetak apa pun
E. HALO lalu None

13. Perhatikan potongan program Python berikut.

```
class Barang:
    def __init__(self, nama, harga):
        self.nama = nama
        self.harga = harga

class BarangImpor(Barang):
    def __init__(self, nama, harga, bea):
        self.bea = bea

b = BarangImpor("Kamera", 500, 50)
print(b.bea)
print(b.nama)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Baris pertama yang dicetak adalah 50
☐ B. Baris kedua mencetak Kamera karena nama diwarisi secara otomatis
☐ C. Baris kedua menimbulkan AttributeError karena __init__ BarangImpor tidak memanggil super().__init__
☐ D. Perbaikannya dengan menambahkan super().__init__(nama, harga) di dalam __init__ BarangImpor
☐ E. Perbaikannya dengan menghapus parameter bea dari __init__ BarangImpor

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

14. Sebuah jasa pengiriman (2027) menghitung ongkos kirim dengan program berikut.

```
class Paket:
    def __init__(self, berat):
        self.berat = berat
    def ongkos(self):
        return self.berat * 1000

class Kilat(Paket):
    def ongkos(self):
        return super().ongkos() + 5000

class Gratis(Paket):
    def ongkos(self):
        return 0

class Fragil(Kilat):
    def ongkos(self):
        return super().ongkos() * 2

total = 0
for p in [Paket(2), Kilat(3), Gratis(4), Fragil(1)]:
    total += p.ongkos()
print(total)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. 16000
B. 11000
C. 15000
D. 22000
E. 26000

15. Perhatikan potongan program Python berikut.

```
class Baterai:
    def __init__(self, daya):
        self.daya = daya

class Senter:
    def __init__(self):
        self.baterai = Baterai(1200)

s = Senter()
print(s.baterai.daya)
print(isinstance(s, Baterai))
```

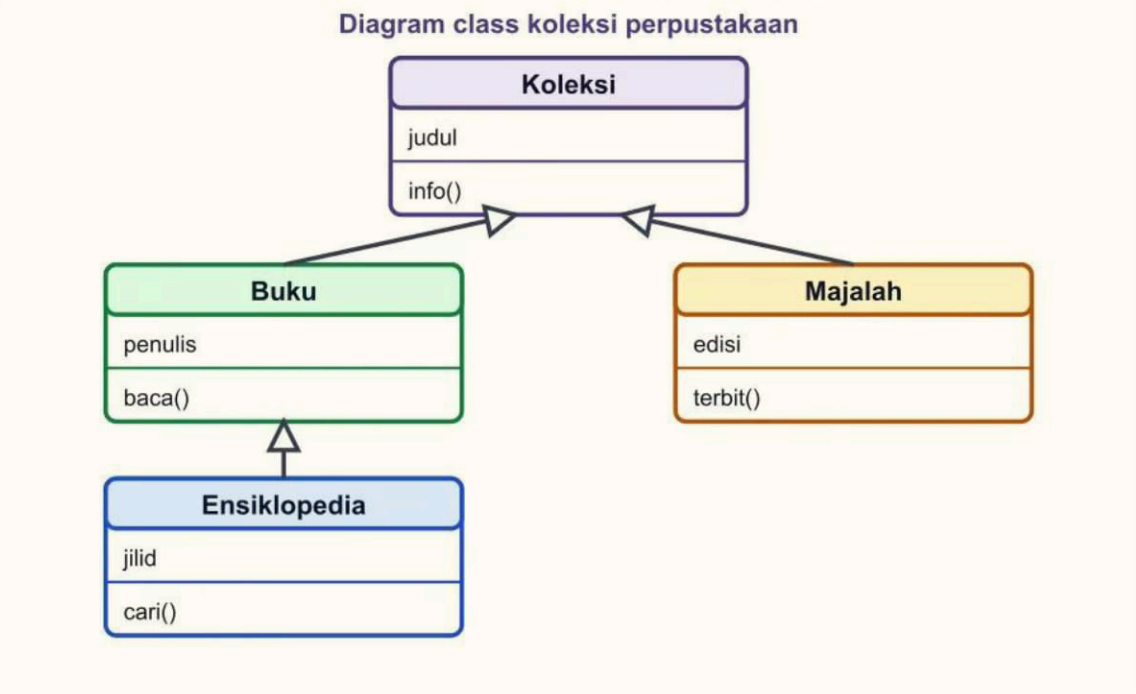

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Senter memiliki Baterai (hubungan has-a), sehingga baterai dipakai sebagai atribut, bukan lewat pewarisan.	○	○
Baris isinstance(s, Baterai) mencetak True karena Senter memakai Baterai.	○	○
Menulis class Senter(Baterai) tepat karena senter adalah sejenis baterai.	○	○

16. Perpustakaan sekolah (2027) menggambar hierarki class koleksinya seperti pada gambar; panah menunjuk dari subclass ke superclass. Sebuah objek dibuat dari class `Ensiklopedia`.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Atribut dan method yang dimiliki objek tersebut, termasuk yang diwarisi, adalah ....



A. atribut judul, penulis, dan jilid; method info(), baca(), dan cari()
B. atribut jilid saja; method cari() saja
C. atribut penulis dan jilid; method baca() dan cari()
D. atribut judul, penulis, edisi, dan jilid; method info(), baca(), terbit(), dan cari()
E. atribut judul dan jilid; method info() dan cari()

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

17. Perhatikan potongan program Python berikut.

```
class Kurir:
    def antar(self):
        return "reguler"

class KurirMotor(Kurir):
    def antar(self):
        return "motor"

class KurirSepeda(Kurir):
    pass

class KurirDrone(Kurir):
    def antar(self):
        return "udara"

daftar = [KurirMotor(), KurirSepeda(), KurirDrone(), Kurir()]
hasil = [k.antar() for k in daftar]
print(hasil)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. `['motor', 'sepeda', 'udara', 'reguler']`
B. `['motor', 'udara', 'udara', 'reguler']`
C. `['motor', 'reguler', 'udara', 'reguler']`
D. `['motor', 'reguler', 'udara']`
E. AttributeError karena KurirSepeda tidak memiliki method antar

18. Perhatikan potongan program Python berikut.

```
class Gedung:
    def __init__(self, nama, lantai):
        self.nama = nama
        self.lantai = lantai

class Asrama(Gedung):
    def __init__(self, nama, lantai, kamar):
        super().__init__(nama, lantai)
        self.kamar = kamar
    def kapasitas(self):
        return self.lantai * self.kamar * 2

a = Asrama("Melati", 3, 10)
print(a.nama, a.kapasitas())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. Melati 30
B. Melati 20
C. Melati 13
D. AttributeError
E. Melati 60

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

19. Perhatikan potongan program Python berikut.

```
class Tanaman:
    def __init__(self, nama):
        self.nama = nama
    def siram(self):
        return self.nama + " disiram"

class Kaktus(Tanaman):
    pass

k = Kaktus("Kaktus mini")
print(k.siram())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. Kaktus mini
B. kaktus disiram
C. siram
D. Kaktus mini disiram
E. AttributeError karena Kaktus tidak memiliki method siram

20. Perhatikan potongan program Python berikut.

```
class Alat:
    def bunyi(self):
        return "..."

class Gitar(Alat):
    def bunyi(self):
        return "petik"

class Drum(Alat):
    def bunyi(self):
        return "tabuh"

for a in [Gitar(), Drum()]:
    print(a.bunyi())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. ... lalu ...
B. petik lalu tabuh
C. tabuh lalu petik
D. petik lalu petik
E. AttributeError

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

21. Pengelola parkir sebuah kampus (2027) menghitung biaya dengan fungsi berikut. Mereka mempertimbangkan mengubah rancangannya menjadi class `Tarif` dengan subclass `Motor`, `Mobil`, dan `Bus`, yang masing-masing menulis ulang method `biaya(jam)`.

```
def biaya(jenis, jam):
    if jenis == "motor":
        return jam * 2000
    elif jenis == "mobil":
        return jam * 5000
    elif jenis == "bus":
        return jam * 10000
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang rancangan tersebut adalah ....

A. rancangan if–elif lebih baik karena seluruh aturan tarif berada di satu fungsi dan tidak pernah perlu diubah lagi

B. polimorfisme mengharuskan setiap subclass memakai nama method yang berbeda agar tidak tertukar satu sama lain

C. menambah jenis truk pada rancangan if–elif tidak memerlukan perubahan pada fungsi biaya yang sudah ada

D. polimorfisme hanya berlaku bila semua objek berasal dari class yang sama persis, bukan dari subclass

E. menambah jenis truk pada rancangan polimorfik cukup menambah subclass baru yang menulis ulang biaya()

22. Perhatikan potongan program Python berikut.

```
class Catatan:
    def __init__(self, isi):
        self.isi = isi
    def baca(self):
        return "[" + self.isi + "]"

class CatatanPenting(Catatan):
    def baca(self):
        return "!" + super().baca() + "!"

class CatatanPribadi(CatatanPenting):
    def baca(self):
        return "*" * len(self.isi)

for p in [Catatan("hai"), CatatanPenting("rapat"), CatatanPribadi("kode")]:
    print(p.baca())
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang keluaran program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Baris pertama adalah [hai]
☐ B. Baris kedua adalah ![rapat]! karena CatatanPenting memanggil versi induk lewat super()
☐ C. Baris ketiga adalah ![kode]! karena CatatanPribadi mewarisi dari CatatanPenting
☐ D. Baris ketiga adalah **** karena CatatanPribadi menulis ulang method tanpa memanggil super()
☐ E. super().baca() pada CatatanPenting memanggil method milik CatatanPribadi

23. Perhatikan potongan program Python berikut.

```
class Nilai:
    def __init__(self):
        self.__angka = 0
    def set_angka(self, a):
        if 0 <= a <= 100:
            self.__angka = a
    def get_angka(self):
        return self.__angka

n = Nilai()
n.set_angka(85)
n.set_angka(110)
n.set_angka(100)
n.set_angka(-5)
print(n.get_angka())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Nilai yang dicetak adalah 100
☐ B. set_angka(110) mengubah nilai menjadi 110 karena method sudah dipanggil
☐ C. set_angka(-5) tidak mengubah nilai karena syarat 0 <= a <= 100 tidak terpenuhi
☐ D. set_angka(100) diterima karena batas atas ikut diterima oleh syarat
☐ E. Atribut __angka sebaiknya dijadikan publik supaya tidak perlu diperiksa di dalam method

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

24. Perhatikan potongan program Python berikut.

```
class Server:
    def __init__(self):
        self.nama = "alfa"
        self._log = []
        self.__kunci = "rahasia"

s = Server()
print(s.nama)
print(s._log)
print(s.__kunci)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tentukan setiap pernyataan Benar atau Salah.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Baris print(s.nama) menampilkan alfa karena atribut itu publik.	○	○
Baris print(s.__kunci) menampilkan rahasia karena Python tidak melarang akses atribut.	○	○
Garis bawah tunggal pada _log hanyalah kesepakatan sehingga s._log tetap dapat dibaca dari luar.	○	○

25. Perhatikan potongan program Python berikut.

```
class Kue:
    def rasa(self):
        return "manis"

class Brownies(Kue):
    def rasa(self):
        return "cokelat"
    def hias(self):
        return "topping"

daftar = [Kue(), Brownies()]
for k in daftar:
    print(k.rasa())
daftar[1].hias()
daftar[0].hias()
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Pernyataan yang tepat tentang program tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Dua baris pertama yang dicetak adalah manis lalu cokelat

☐ B. daftar[1].hias() berhasil karena objek di indeks 1 adalah Brownies yang memiliki method hias

☐ C. daftar[0].hias() berhasil karena semua elemen daftar adalah Kue

☐ D. daftar[0].hias() menimbulkan AttributeError karena objek di indeks 0 adalah Kue biasa

☐ E. Method hias dapat dipanggil pada semua objek di daftar karena daftar bertipe Kue

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 5.1 · Class, object, dan prinsip dasar OOP	1, 2, 3, 5, 6, 8, 9, 10	___ / 8	Ulangi hlm. 243
Sub bab 5.2 · Enkapsulasi, access modifier, dan getter–setter	4, 7, 11, 12, 23, 24	___ / 6	Ulangi hlm. 259
Sub bab 5.3 · Pewarisan	13, 15, 16, 18, 19	___ / 5	Ulangi hlm. 276
Sub bab 5.4 · Polymorphism	14, 17, 20, 21, 22, 25	___ / 6	Ulangi hlm. 293

Skor 70 % atau lebih: lanjut ke bab berikutnya. Skor 50–69 %: ulangi sub bab pada tabel yang kurang dari separuh benar. Skor di bawah 50 %: baca ulang bab ini dari awal. (Angka ini anjuran belajar, bukan prediksi nilai TKA.)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bagian C · Simulasi TKA

Kerjakan setelah semua bab selesai. Kerjakan tiap simulasi dalam sekali duduk sesuai waktunya, tanpa membuka materi.

Daftar simulasi

Simulasi	Jumlah soal	Waktu	Kode LMS
Simulasi 1 — Prediksi TKA 2026	25 soal	60 menit	SMKPPLG-FIN1
Simulasi 2 — Prediksi TKA 2026	25 soal	60 menit	SMKPPLG-FIN2
Simulasi 3 — Prediksi TKA 2026	25 soal	60 menit	SMKPPLG-FIN3

Saran: beri jeda beberapa hari antarsimulasi. Setelah tiap simulasi, cocokkan jawaban di Bagian D lalu ulangi sub bab yang ditunjuk tabel anjuran sebelum mengerjakan simulasi berikutnya.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Simulasi 1 — Prediksi TKA 2026

SMKPPLG-FIN1

Petunjuk

Jumlah soal: 25. Waktu: 60 menit.

Kerjakan sekaligus tanpa membuka materi. Tandai jawaban pada lembar jawaban. Kunci dan pembahasan ada di Bagian D; hitung skor dan lihat tabel anjuran di akhir paket.

Kerjakan Simulasi 1 di LMS

Hasil dan analisis per sub bab tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-FIN1

1. Hendra, programmer yang selama ini mengurus akun-akun studio gim Pelita, mengundurkan diri pada Mei 2027 dan akan berhenti dua minggu lagi. Akun pemilik repositori GitHub, akun pengembang di toko aplikasi, dan akun penyedia domain studio masih terdaftar dengan email pribadi Hendra.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat dilakukan pimpinan studio adalah ....

A. Membiarkan akun tetap atas nama Hendra karena ia masih dapat dihubungi
B. Meminta kata sandi semua akun Hendra, lalu memakainya bersama-sama
C. Membuat akun baru dan meninggalkan akun lama beserta seluruh isinya
D. Memindahkan kepemilikan akun ke email resmi studio sebelum ia berhenti
E. Menunggu Hendra berhenti dulu, baru menghubungi penyedia layanan akun

2. Aplikasi PPDB online sebuah SMK akan ditambah kolom baru pada tabel pendaftar yang sudah berisi 3.500 data (2027). Programmer sudah menulis skrip perubahan struktur tabel dan mengujinya di laptopnya dengan 10 data contoh. Pendaftaran masih berjalan setiap hari, paling ramai pukul 08.00–16.00.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat sebelum dan saat menjalankan skrip di server produksi adalah ....

A. Langsung menjalankan skrip di produksi siang ini karena sudah lolos uji di laptop
B. Mencadangkan basis data, menguji skrip pada salinannya, lalu menjalankannya malam hari
C. Menghapus tabel pendaftar, lalu membuat ulang tabel itu dengan kolom yang baru
D. Menjalankan skrip di produksi lebih dulu, lalu mencadangkan data bila ada masalah
E. Meminta semua pendaftar mengisi ulang formulir setelah kolom baru ditambahkan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Setelah merilis aplikasi pemesanan kantin pada April 2027, tim mengadakan rapat retrospeksi. Datanya: 9 dari 12 bug yang dilaporkan pengguna berasal dari layar yang tidak pernah diuji di ponsel berlayar kecil, dan dua anggota sering lembur karena pembagian tugas tidak merata. Tim akan memakai cara kerja yang sama untuk proyek aplikasi koperasi bulan depan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Retrospeksi terutama bertujuan menemukan anggota yang paling banyak membuat bug agar diberi sanksi.	○	○
Menambahkan uji di ponsel berlayar kecil ke daftar periksa sebelum rilis adalah tindak lanjut yang sesuai data.	○	○
Karena aplikasi kantin sudah dirilis, hasil retrospeksi tidak perlu dibawa ke proyek aplikasi koperasi.	○	○

4. Studio gim lulusan SMK akan merilis gim *Lintas Selat* di toko aplikasi (2027). Toko aplikasi memotong 30 % dari setiap penjualan, sehingga studio menerima 70 %. Hasil uji pasar memperkirakan jumlah pembeli dalam 3 bulan pertama untuk setiap pilihan harga seperti pada tabel.

Pilihan harga	Perkiraan pembeli
Rp15.000	4.000 orang
Rp30.000	2.200 orang
Rp45.000	1.300 orang

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pilihan harga yang memberi pendapatan bersih terbesar bagi studio beserta nilainya adalah ....

A. Rp30.000 dengan pendapatan bersih Rp46.200.000
B. Rp15.000 dengan pendapatan bersih Rp60.000.000
C. Rp45.000 dengan pendapatan bersih Rp40.950.000
D. Rp30.000 dengan pendapatan bersih Rp66.000.000
E. Rp15.000 dengan pendapatan bersih Rp42.000.000

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

5. Saat praktik di lab RPL (2027), Lina melihat kabel charger laptopnya terkelupas di dekat kepala colokan sehingga kawat tembaganya terlihat. Laptop itu sedang dipakai mengerjakan tugas yang harus dikumpulkan satu jam lagi, dan baterainya tinggal 40 %. *(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)*
Tindakan Lina yang paling tepat adalah ....

A. Membalut bagian terkelupas dengan selotip kertas lalu tetap memakainya
B. Mencabut charger dari stopkontak, lalu melapor untuk mengganti charger
C. Tetap memakai charger asalkan bagian yang terkelupas tidak tersentuh
D. Menekuk kabel agar bagian yang terkelupas tersembunyi di balik meja
E. Memakai charger itu sampai tugas selesai, baru kemudian melapor

6. Tim pengembang aplikasi Toko Daring Sinar terdiri atas tiga programmer (Andi, Sari, Budi) dan satu penguji. Tim memantau pekerjaan dengan papan Kanban pada gambar dan memiliki dua aturan: (1) kolom "Sedang dikerjakan" berisi paling banyak 3 kartu; (2) setiap programmer hanya memegang satu kartu di kolom itu agar pekerjaan cepat tuntas. *(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)*
Tindakan ketua tim yang paling tepat pada rapat harian berdasarkan papan tersebut adalah ....



A. Menaikkan batas kolom Sedang dikerjakan menjadi 5 kartu
B. Memindahkan Fitur notifikasi ke Sedang dikerjakan sekarang
C. Meminta Andi menuntaskan satu kartu dan menunda kartu lainnya
D. Meminta Budi melepas Pencarian karena ia bekerja sendirian
E. Mengembalikan kartu Keranjang ke kolom Akan dikerjakan

7. Aplikasi rapor digital yang dipakai 30 sekolah mulai lambat saat musim pengisian nilai (2027). Perusahaan pembuatnya membuka lowongan untuk orang yang bertugas merapikan struktur tabel, mengatur hak akses setiap akun pengguna basis data,

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

mempercepat kueri yang lambat, serta memastikan cadangan basis data berjalan dan dapat dipulihkan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Profesi yang dicari perusahaan tersebut adalah ....

A. Frontend developer yang membuat tampilan rapor di peramban
B. Network administrator yang mengatur router jaringan sekolah
C. UI/UX designer yang merancang alur pengisian nilai oleh guru
D. Technical support yang membantu guru yang lupa kata sandi
E. Database administrator yang mengelola basis data aplikasi

8. Proyek aplikasi absensi SMK Pelita akan diserahkan kepada tim PKL angkatan berikutnya (2027). Saat ini berkas README di repositori hanya berisi satu baris: "Aplikasi absensi SMK". Ketua tim menyusun rencana isi README yang baru.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
README perlu memuat langkah memasang dan menjalankan aplikasi, termasuk versi Python yang dipakai.	○	○
Daftar variabel konfigurasi perlu dijelaskan kegunaannya tanpa menuliskan nilai rahasianya.	○	○
Kata sandi basis data produksi sebaiknya ditulis di README agar tim baru tidak perlu bertanya.	○	○

9. Website portofolio milik Sinta berjalan normal di laptop Windows-nya (2027). Setelah semua berkas diunggah ke server Linux, logo tidak tampil, sedangkan gambar lain di folder yang sama tampil normal. Di kode HTML tertulis `<img src="gambar/logo.png">`, sedangkan di folder proyek berkasnya tersimpan sebagai `Gambar/Logo.PNG`. Gambar lain tertulis di HTML persis sama dengan nama berkasnya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perbaikan yang paling tepat adalah ....

A. Mengunggah ulang semua berkas karena unggahan pertama tidak lengkap
B. Mengecilkan ukuran berkas logo agar server mampu menampilkannya
C. Memindahkan logo ke folder utama tanpa mengubah penulisan di HTML
D. Mengubah format logo dari PNG ke JPG agar dikenali server Linux
E. Menyamakan huruf besar-kecil nama folder dan berkas dengan HTML

10. Bima, pengembang gim lepas, dikontrak studio Awan Biru untuk membuat sistem pertarungan sebuah gim yang belum diumumkan (2027). Kontraknya memuat perjanjian kerahasiaan (NDA) yang berlaku sampai gim dirilis. Di tengah pekerjaan, Bima memperkirakan pekerjaannya akan selesai dua minggu lebih lambat dari tenggat.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Bima yang sesuai dengan etika kerja profesional adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Memberi tahu studio sejak dini bahwa pekerjaan akan terlambat, beserta alasan dan perkiraan waktunya.
☐ B. Mengunggah cuplikan video sistem pertarungan ke media sosial sebagai portofolio sebelum gim diumumkan.
☐ C. Menyerahkan pekerjaan tepat waktu dengan bagian yang belum diuji, tanpa memberi tahu studio.
☐ D. Mencantumkan proyek ini di portofolio setelah gim dirilis, sesuai izin yang diberikan studio.
☐ E. Mencatat jam kerja dan pekerjaan yang diselesaikan dengan jujur dalam laporan mingguan.

11. Saat siswa mengetik `https://belajar.smkmaju.sch.id/kuis` di peramban (2027), terjadi beberapa proses berikut (belum berurutan).
(1) Server mengirim respons berisi halaman kuis dengan kode status 200.
(2) Peramban menanyakan alamat IP belajar.smkmaju.sch.id ke server DNS.
(3) Peramban mengirim permintaan HTTP GET /kuis.
(4) Peramban membuka koneksi TCP ke alamat IP server di port 443 dan menyiapkan enkripsi.
(5) Peramban menampilkan halaman kuis.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Urutan proses yang benar adalah ....

A. (2) – (3) – (4) – (1) – (5)
B. (4) – (2) – (3) – (1) – (5)
C. (2) – (4) – (3) – (1) – (5)
D. (3) – (2) – (4) – (1) – (5)
E. (2) – (4) – (1) – (3) – (5)

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

12. Aplikasi PPDB sebuah SMK (2027) mengirim email konfirmasi otomatis ke setiap pendaftar. Petugas membaca email balasan pendaftar di aplikasi email kantor. Laptop petugas mendapat alamat IP secara otomatis setiap kali tersambung ke Wi-Fi kantor. *(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)*
Pernyataan yang tepat tentang protokol yang bekerja dalam kegiatan tersebut adalah .... (Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Email konfirmasi dikirim oleh server aplikasi menggunakan protokol IMAP.
☐ B. Laptop petugas mendapat alamat IP secara otomatis dari layanan DHCP.
☐ C. Petugas membaca email balasan pendaftar menggunakan protokol SMTP.
☐ D. DHCP bekerja di lapisan internet karena tugasnya membagikan alamat IP.
☐ E. SMTP, IMAP, dan DHCP termasuk protokol pada lapisan aplikasi TCP/IP.

13. Aplikasi sewa sepeda sekolah (2027) membedakan tarif umum dan tarif pelajar. Programmer menulis kode berikut untuk menghitung pendapatan dari dua penyewa yang masing-masing menyewa 2 jam.

```
class Sewa:
    def tarif(self):
        return 5000

    def total(self, jam):
        return self.tarif() * jam + 2000

class SewaPelajar(Sewa):
    def tarif(self):
        return 3000

hasil = 0
for s in [Sewa(), SewaPelajar()]:
    hasil += s.total(2)
print(hasil)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 24000
B. 16000
C. 12000
D. 20000
E. 18000

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

14. Aplikasi sekolah (2027) mengatur hak akses akun dengan class berikut.

```
class Akun:
    def __init__(self, nama):
        self.nama = nama
    def hak(self):
        return ["lihat"]

class Guru(Akun):
    def hak(self):
        return super().hak() + ["nilai"]

class WaliKelas(Guru):
    def hak(self):
        return super().hak() + ["rapor"]

w = WaliKelas("Bu Sri")
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang kode tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `w.hak()` menghasilkan `['lihat', 'nilai', 'rapor']`.
☐ B. `w.nama` berisi "Bu Sri" karena konstruktor Akun diwarisi lewat Guru.
☐ C. `Guru("Pak Ali").hak()` menghasilkan `['lihat', 'nilai', 'rapor']`.
☐ D. WaliKelas adalah subclass Guru, dan Guru adalah subclass Akun.
☐ E. Bila hak() di class Guru dihapus, `w.hak()` menghasilkan `['lihat', 'rapor']`.

15. Aplikasi kasir koperasi (2027) akan mendukung tiga cara bayar: tunai, QRIS, dan transfer bank; tahun depan mungkin ditambah dompet digital. Tim merancang superclass `Pembayaran` dengan method `bayar(jumlah)`, lalu subclass `Tunai`, `Qris`, dan `Transfer` yang masing-masing menulis ulang bayar(). Bagian kasir hanya memanggil `metode.bayar(total)` tanpa perlu tahu cara kerja setiap pembayaran.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang rancangan tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Bagian kasir harus diubah setiap kali ada cara bayar baru yang ditambahkan.
☐ B. Bagian kasir memanggil bayar() tanpa mengetahui detail setiap cara bayar; ini menunjukkan abstraksi.
☐ C. Tunai, Qris, dan Transfer yang menulis ulang bayar() dengan caranya masing-masing menunjukkan polimorfisme.
☐ D. Agar polimorfisme berjalan, ketiga subclass harus memiliki nama method yang berbeda-beda.
☐ E. Dompet digital kelak dapat ditambahkan sebagai subclass baru dari Pembayaran.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

16. Aplikasi kuis kelas (2027) memberi 4 poin untuk setiap jawaban benar, mengurangi 1 poin untuk setiap jawaban salah, dan memberi bonus 5 poin bagi peserta yang tidak salah sama sekali. Programmer menulis kode berikut.

```
def poin(benar, salah, bonus=0):
    return benar * 4 - salah + bonus

def nilai_akhir(benar, salah):
    if salah == 0:
        return poin(benar, salah, 5)
    return poin(benar, salah)

print(nilai_akhir(6, 3), nilai_akhir(5, 0))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `21 25`
B. `21 20`
C. `26 25`
D. `27 25`
E. Error, karena poin() dipanggil dengan dua argumen

17. Aplikasi kantin (2027) memakai class berikut agar harga produk tidak pernah nol atau negatif.

```
class Produk:
    def __init__(self, nama, harga):
        self.nama = nama
        self.__harga = 0
        self.set_harga(harga)

    def get_harga(self):
        return self.__harga

    def set_harga(self, h):
        if h > 0:
            self.__harga = h

p = Produk("Kopi", -5000)
print(p.get_harga())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Keluaran program adalah -5000 karena harga dikirim saat objek dibuat.	○	○
Konstruktor memanggil set_harga() sehingga aturan harga positif juga berlaku saat objek dibuat.	○	○
Bila setelah itu dijalankan p.set_harga(12000) lalu p.set_harga(0), get_harga() menghasilkan 12000.	○	○

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

18. Aplikasi pemilihan ketua OSIS (2027) menghitung perolehan suara tiap calon. Programmer menguji dengan enam suara berikut.

```
suara = ["A", "B", "A", "C", "A", "B"]
hasil = {}
for s in suara:
    if s in hasil:
        hasil[s] += 1
    else:
        hasil[s] = 1
print(hasil)
```

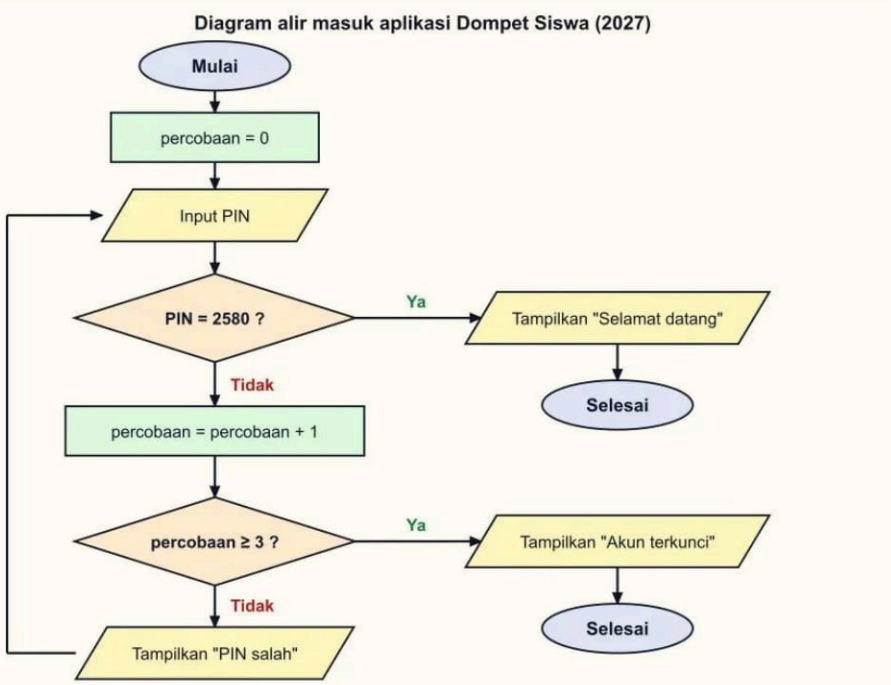
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. `{'A': 1, 'B': 1, 'C': 1}`
B. `{'A': 2, 'B': 1, 'C': 0}`
C. `{'A': 4, 'B': 3, 'C': 2}`
D. `['A', 'B', 'A', 'C', 'A', 'B']`
E. `{'A': 3, 'B': 2, 'C': 1}`

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

19. Aplikasi Dompet Siswa memeriksa PIN saat pengguna masuk dengan diagram alir pada gambar. PIN yang benar adalah 2580. Seorang pengguna mengetik PIN berturut-turut: 1111, lalu 0000, lalu 9999, lalu 2580 (2027).
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pesan yang tampil di layar secara berurutan adalah ....



A. PIN salah, PIN salah, PIN salah, Selamat datang
B. PIN salah, PIN salah, Akun terkunci
C. PIN salah, PIN salah, PIN salah, Akun terkunci
D. PIN salah, Akun terkunci
E. PIN salah, PIN salah, Selamat datang

20. Pusat rental komputer milik koperasi SMK (2027) menghitung biaya sewa dengan fungsi berikut. Member mendapat potongan bila biayanya lebih dari Rp5.000.

```
def tarif(jam, member):
    if jam <= 1:
        biaya = 3000
    elif jam <= 3:
        biaya = 3000 + (jam - 1) * 2000
    else:
        biaya = 10000
    if member and biaya > 5000:
        biaya = biaya - 2000
    return biaya
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang fungsi tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `tarif(2, True)` menghasilkan 3000 karena member mendapat potongan.
☐ B. `tarif(5, True)` menghasilkan 10000 karena lebih dari 3 jam.
☐ C. `tarif(3, True)` menghasilkan 5000.
☐ D. `tarif(1, False)` dan `tarif(1, True)` menghasilkan nilai berbeda.
☐ E. `tarif(4, False)` menghasilkan 10000.

21. Tim PPLG SMK Bakti membuat tiga aplikasi sekolah: absensi, perpustakaan, dan koperasi (2027). Ketiganya memiliki kode untuk memeriksa format NIS dan mengubah format tanggal yang sama persis, tetapi ditulis ulang di setiap aplikasi. Tim berencana membuat modul bersama `utilitas.py` berisi `validasi_nis()` dan `format_tanggal()` yang diimpor oleh ketiga aplikasi.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pendapat yang tepat tentang rencana tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Bila aturan NIS berubah, perbaikan dilakukan sekali di utilitas.py dan berlaku di ketiga aplikasi.
☐ B. Fungsi di utilitas.py sebaiknya punya satu tugas dan tidak bergantung pada data khusus satu aplikasi.
☐ C. Setiap perubahan utilitas.py perlu diuji pada ketiga aplikasi karena semuanya memakai modul itu.
☐ D. validasi_nis() sebaiknya langsung menampilkan pesan galat di layar aplikasi absensi agar praktis.
☐ E. Nama, parameter, dan nilai kembalian fungsi di utilitas.py perlu didokumentasikan untuk pemakainya.

22. Aplikasi perpustakaan daerah (2027) mengizinkan pengunjung meminjam buku bila berumur minimal 17 tahun atau memiliki kartu pelajar. Programmer menulis kode berikut untuk seorang pengunjung.

```
umur = 16
punya_kartu = True
boleh = umur >= 17 or punya_kartu
print(boleh, type(boleh))
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `False <class 'bool'>`
B. `True <class 'str'>`
C. `16 <class 'int'>`
D. `True <class 'bool'>`
E. `False <class 'int'>`

23. Di komputer lab sekolah (2027), Sinta mengkloning repositori tim lalu membuat commit. Di GitHub, commit itu tercatat atas nama "Admin Lab" dengan email admin.lab@contoh.sch.id, padahal Sinta sudah masuk ke akun GitHub-nya sendiri di peramban. Komputer itu dipakai bergantian oleh banyak siswa yang juga membuat commit ke repositori masing-masing.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat agar commit Sinta berikutnya tercatat atas namanya adalah ....

A. Mengatur `user.name` dan `user.email` miliknya hanya untuk repositori itu
B. Mengubah `user.name` dan `user.email` global komputer menjadi miliknya
C. Keluar dari akun GitHub di peramban, lalu masuk kembali dengan akunnya
D. Menghapus repositori, lalu mengkloningnya ulang dengan akun GitHubnya
E. Menambahkan akun Sinta sebagai kolaborator di pengaturan repositori

24. Laptop praktik di meja 7 lab RPL tidak dapat membuka server aplikasi (2027). Konfigurasi IP-nya statis 192.168.10.34, mask 255.255.255.0, gateway 192.168.10.1, sama seperti dokumen lab. Laptop di meja 6 yang tersambung ke switch yang sama dapat membuka server itu. Teknisi menjalankan beberapa uji di laptop meja 7.

No	Perintah	Hasil
1	ping 127.0.0.1	balasan diterima
2	ping 192.168.10.1 (gateway)	Request timed out
3	ping 192.168.10.20 (server aplikasi)	Request timed out

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Kesimpulan dan langkah yang tepat adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Protokol TCP/IP di laptop meja 7 berjalan karena ping ke 127.0.0.1 berhasil.
☐ B. Server aplikasi sedang mati karena ping ke 192.168.10.20 tidak dibalas.
☐ C. Masalah kemungkinan berada pada sambungan fisik, misalnya kabel LAN atau port switch.
☐ D. Langkah awal yang tepat adalah memeriksa dan mencoba mengganti kabel LAN laptop itu.
☐ E. Alamat DNS laptop salah, sehingga semua ping ke alamat IP gagal.

25. Aplikasi absensi sekolah (2027) memakai class berikut untuk mencatat siswa di setiap kelas.

```
class Kelas:
    def __init__(self, nama):
        self.nama = nama
        self.siswa = []

    def tambah(self, s):
        self.siswa.append(s)

x = Kelas("XI PPLG 1")
y = Kelas("XI PPLG 2")
x.tambah("Ani")
x.tambah("Budi")
y.tambah("Cici")
print(len(x.siswa), len(y.siswa))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `3 3`
B. `2 2`
C. `2 1`
D. `1 1`
E. `3 0`

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Lembar jawaban

No	A	B	C	D	E
1	○	○	○	○	○
2	○	○	○	○	○
3					
4	○	○	○	○	○
5	○	○	○	○	○
6	○	○	○	○	○
7	○	○	○	○	○
8					
9	○	○	○	○	○

No	A	B	C	D	E
10					
11	○	○	○	○	○
12					
13	○	○	○	○	○
14					
15					
16	○	○	○	○	○
17					
18	○	○	○	○	○

No	A	B	C	D	E
19	○	○	○	○	○
20					
21					
22	○	○	○	○	○
23	○	○	○	○	○
24					
25	○	○	○	○	○

Soal PG kompleks / Benar–Salah (nomor 3, 8, 10, 12, 14, 15, 17, 20, 21, 24): jawab langsung pada soalnya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 1.1 · Profesi dan peluang usaha PPLG	4, 7	___ / 2	Ulangi hlm. 9
Sub bab 1.2 · Manajemen proyek dan budaya mutu	3, 6	___ / 2	Ulangi hlm. 23
Sub bab 2.1 · K3LH dan budaya kerja profesional	5, 10	___ / 2	Ulangi hlm. 52
Sub bab 2.2 · Pengelolaan aset fisik dan digital	1	___ / 1	Ulangi hlm. 66
Sub bab 2.3 · Pengelolaan file proyek, source code, dan dokumentasi	2, 8	___ / 2	Ulangi hlm. 79
Sub bab 3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi	9, 23	___ / 2	Ulangi hlm. 104
Sub bab 3.2 · Konfigurasi jaringan dasar	24	___ / 1	Ulangi hlm. 118
Sub bab 3.3 · Layer TCP/IP dan fungsi protokol	12	___ / 1	Ulangi hlm. 133
Sub bab 3.4 · Mekanisme komunikasi jaringan	11	___ / 1	Ulangi hlm. 145
Sub bab 4.1 · Tipe data dan struktur data	18, 22	___ / 2	Ulangi hlm. 172
Sub bab 4.2 · Struktur kontrol percabangan	20	___ / 1	Ulangi hlm. 186
Sub bab 4.3 · Struktur kontrol perulangan	19	___ / 1	Ulangi hlm. 201
Sub bab 4.4 · Fungsi, prosedur, dan modularisasi	16, 21	___ / 2	Ulangi hlm. 215
Sub bab 5.1 · Class, object, dan prinsip dasar OOP	15, 25	___ / 2	Ulangi hlm. 243
Sub bab 5.2 · Enkapsulasi, access modifier, dan getter–setter	17	___ / 1	Ulangi hlm. 259
Sub bab 5.3 · Pewarisan	14	___ / 1	Ulangi hlm. 276
Sub bab 5.4 · Polymorphism	13	___ / 1	Ulangi hlm. 293

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Simulasi 2 — Prediksi TKA 2026

SMKPPLG-FIN2

Petunjuk

Jumlah soal: 25. Waktu: 60 menit.

Kerjakan sekaligus tanpa membuka materi. Tandai jawaban pada lembar jawaban. Kunci dan pembahasan ada di Bagian D; hitung skor dan lihat tabel anjuran di akhir paket.

Kerjakan Simulasi 2 di LMS

Hasil dan analisis per sub bab tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-FIN2

1. Studio aplikasi Cahaya memeriksa biaya langganan perangkat lunak per bulan pada Juli 2027. Pimpinan ingin berhemat tanpa mengganggu pekerjaan tim.

Layanan	Biaya per bulan	Pemakaian 3 bulan terakhir
Aplikasi desain UI (5 akun × Rp150.000)	Rp750.000	5 desainer memakai setiap hari
Manajemen tugas (10 akun × Rp50.000)	Rp500.000	hanya 4 akun yang pernah masuk
Perekam layar	Rp200.000	tidak dipakai; fitur sama ada di aplikasi rapat gratis tim
Server uji	Rp1.000.000	dipakai tim backend setiap hari

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Langkah penghematan yang paling tepat beserta besarnya adalah ....

A. Menghentikan aplikasi desain UI, hemat Rp750.000 per bulan
B. Menghentikan perekam layar saja, hemat Rp200.000 per bulan
C. Hentikan perekam layar, akun tugas jadi 4: hemat Rp500.000
D. Hentikan manajemen tugas dan perekam layar: hemat Rp700.000
E. Mengurangi akun manajemen tugas jadi 4, hemat Rp300.000

2. Dua hari sebelum rilis aplikasi pembayaran SPP (2027), penguji melaporkan tiga bug: (1) aplikasi menutup sendiri saat siswa menekan tombol "Bayar" dengan QRIS; (2) salah ketik "Pembayran" di halaman bantuan; (3) warna tombol "Riwayat" sedikit berbeda dari rancangan. Waktu tim hanya cukup untuk memperbaiki sebagian. Aturan mutu tim: bug yang menghalangi fungsi utama tidak boleh ada saat rilis.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Bug (1) harus diperbaiki sebelum rilis karena menghalangi fungsi utama aplikasi.	○	○
Bug (2) dan (3) didahulukan karena jumlahnya lebih banyak dan cepat diperbaiki agar aplikasi tampak rapi.	○	○
Bug (2) dan (3) boleh ditunda ke versi berikutnya asalkan tetap tercatat di daftar bug.	○	○

3. Tim membuat aplikasi kasir untuk koperasi SMK Karya (2027). Kebutuhan dikumpulkan lewat wawancara dengan ketua koperasi, rancangan disetujui ketua koperasi, kode ditinjau rekan, dan semua uji otomatis lolos. Aplikasi lalu langsung dipakai. Pada hari pertama, para kasir kesulitan: dalam praktik mereka memasukkan diskon anggota setelah semua barang dipindai, sedangkan aplikasi meminta status anggota di awal transaksi. Antrean pun memanjang.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tahap yang paling mungkin mencegah masalah tersebut sebelum aplikasi dipakai adalah ....

A. Menambah uji otomatis untuk fungsi penghitung total belanja
B. Uji penerimaan oleh kasir sungguhan dengan alur kerja nyata
C. Peninjauan kode oleh dua programmer sekaligus sebelum digabung
D. Menyusun dokumentasi teknis yang lebih lengkap untuk programmer
E. Pemeliharaan rutin server setiap minggu setelah aplikasi dirilis

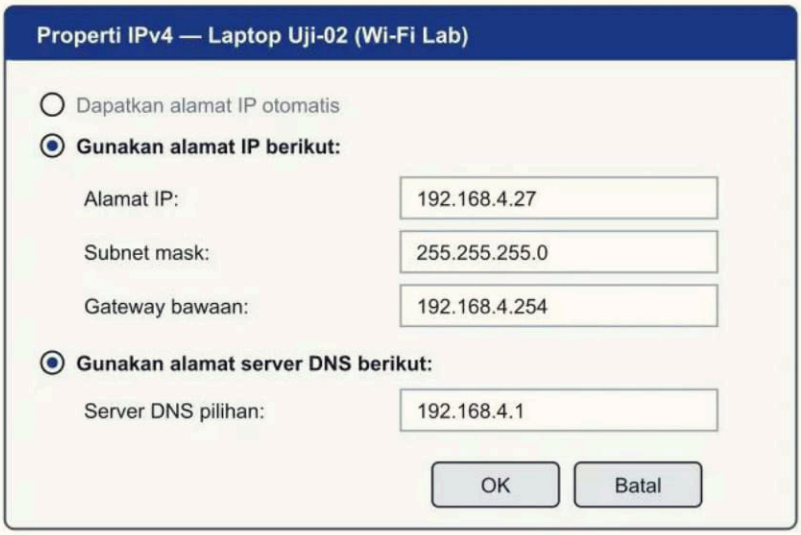
4. Aplikasi web tim membaca alamat basis data dari variabel `DB_HOST` di berkas `.env` (2027). Berkas .env tercantum di .gitignore, sehingga repositori hanya berisi `.env.example` yang memuat nama-nama variabel tanpa nilai. Gilang mengkloning repositori di laptop barunya, lalu aplikasi gagal dengan pesan "DB_HOST belum diatur".
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang paling tepat adalah ....

A. Menghapus .env dari .gitignore agar berkas itu ikut ke repositori
B. Menulis alamat dan kata sandi basis data langsung di dalam kode
C. Menyalin .env.example jadi .env, lalu mengisi nilainya dari ketua
D. Mengganti nama .env.example menjadi .env tanpa mengisi nilainya
E. Mengkloning ulang repositori karena unduhannya diduga tidak lengkap

5. Router lab RPL beralamat 192.168.4.1 dan sekaligus menjadi server DNS lab (2027). Laptop Uji-02 harus memakai IP statis karena dijadikan server uji. Pengaturan IP laptop itu ditunjukkan pada gambar. Hasil uji dari laptop tersebut:
• membuka server lokal http://192.168.4.10:8000 → berhasil

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• ping 8.8.8.8 (alamat IP sebuah server di internet) → gagal
• membuka situs internet mana pun → gagal
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perbaikan yang paling tepat adalah ....



A. Mengubah gateway bawaan menjadi 192.168.4.1
B. Mengubah server DNS pilihan menjadi 8.8.8.8
C. Mengubah subnet mask menjadi 255.255.0.0
D. Mengubah alamat IP menjadi 192.168.4.254
E. Menyalakan ulang server lokal 192.168.4.10

6. Gim teka-teki kata *Balik Kata* (2027) membalik urutan huruf sebuah kata. Programmer menulis kode berikut.

```
kata = "GIM"
hasil = ""
for h in kata:
    hasil = h + hasil
print(hasil)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Keluaran program adalah MIG.	○	○
Setelah putaran kedua, isi variabel hasil adalah "GI".	○	○
Bila baris di dalam perulangan diganti menjadi hasil = hasil + h, keluarannya tetap MIG.	○	○

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

7. Rina membuat gim dengan Python di laptopnya (2027). Kodenya memuat gambar latar dengan path `C:/Users/Rina/Desktop/gim/aset/latar.png`. Folder gim lalu disalin utuh ke laptop Doni di `D:/proyek/gim`, dan gim berhenti dengan `FileNotFoundError`. Gim selalu dijalankan dari dalam folder gim, dan ke depan proyek ini akan dikerjakan bergantian di beberapa laptop.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perbaikan yang paling tepat adalah ....

A. Doni membuat folder C:/Users/Rina/Desktop/gim di laptopnya
B. Memakai path relatif dari folder proyek, yaitu aset/latar.png
C. Mengganti nama berkas latar.png menjadi Latar.png agar dikenali
D. Memindahkan berkas latar.png ke folder Desktop di laptop Doni
E. Mengubah path menjadi D:/proyek/gim/aset/latar.png di dalam kode

8. Aplikasi kasir koperasi (2027) mulai salah menghitung total struk untuk pembeli BUKAN anggota sejak Selasa siang. Uji otomatis yang dijalankan Selasa pukul 11.00 masih lolos semua, termasuk uji total untuk pembeli bukan anggota. Riwayat commit minggu itu seperti pada tabel.

Commit	Waktu	Pesan commit	Berkas yang diubah
a1	Senin 09.00	Tambah halaman profil	profil.py
b2	Senin 15.00	Ubah warna tombol	style.css
c3	Selasa 10.00	Ganti rumus diskon anggota	kasir.py
d4	Selasa 13.00	update	kasir.py, laporan.py
e5	Rabu 08.00	Perbaiki ejaan menu	menu.py

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Commit yang paling mungkin menjadi sumber bug dan perlu diperiksa lebih dulu adalah ....

A. Commit a1, karena halaman profil menyimpan data pembeli
B. Commit b2, karena tombol bayar ikut berubah tampilannya
C. Commit c3, karena pesannya menyebut perubahan rumus diskon
D. Commit e5, karena commit terbaru paling mungkin bermasalah
E. Commit d4, setelah uji lolos dan ikut mengubah kasir.py

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

9. Startup lulusan SMK membuat gim edukasi matematika untuk siswa SD *Hitung Seru* (2027). Calon pembelinya adalah sekolah dan orang tua. Hasil survei: sekolah bersedia membayar langganan tahunan bila ada laporan kemajuan siswa untuk guru, sedangkan banyak orang tua ingin mencoba dulu sebelum membayar.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Model pendapatan yang sesuai untuk startup tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Lisensi tahunan per sekolah yang menyertakan dasbor laporan kemajuan siswa bagi guru.
☐ B. Iklan yang muncul di tengah permainan dan mengarahkan anak ke toko daring.
☐ C. Versi gratis berisi beberapa level, sedangkan seluruh level tersedia di versi berbayar.
☐ D. Pembelian dalam gim yang mendorong anak membeli koin agar tidak tertinggal teman.
☐ E. Langganan keluarga bulanan yang dapat dihentikan kapan saja oleh orang tua.

10. Aplikasi e-learning sekolah (2027) ingin menampilkan jumlah siswa berbeda yang masuk ke aplikasi hari ini. Seorang siswa dapat masuk berkali-kali dalam sehari, dan setiap kali masuk aplikasi menerima nama pengguna (username) siswa itu. Urutan masuk tidak perlu disimpan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Struktur data Python yang paling tepat untuk menampung username tersebut adalah ....

A. List, karena setiap username ditambahkan di akhir sesuai urutan masuk
B. Tuple, karena data username siswa tidak boleh diubah selama sehari
C. String panjang, karena username dapat digabung dengan tanda koma
D. Dictionary, dengan jam masuk sebagai kunci dan username sebagai nilai
E. Set, karena username yang sama otomatis hanya tersimpan sekali

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

11. Studio animasi kecil menata ulang ruang kerja pada Januari 2027 karena beberapa animator mengeluh mata cepat lelah dan pinggang pegal setelah bekerja seharian di depan komputer. Ruangan itu memiliki satu jendela besar yang terang di siang hari. Pemilik studio menerima beberapa usulan dari karyawan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Usulan yang sesuai dengan prinsip K3 adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Meletakkan monitor tepat di depan jendela sehingga pemandangan luar terlihat di belakang layar.
☐ B. Mengatur tinggi kursi sehingga kaki menapak lantai dan lutut kira-kira membentuk sudut siku.
☐ C. Bekerja empat jam tanpa jeda agar fokus mengerjakan animasi tidak terputus.
☐ D. Merapikan kabel di bawah meja dengan pengikat agar tidak melintang di jalan.
☐ E. Mematikan lampu ruangan agar layar terlihat lebih kontras saat bekerja.

12. Aplikasi gudang koperasi sekolah (2027) mensimulasikan aturan isi ulang: rak yang kosong diisi dua kali jumlah barang di rak sebelah kirinya. Programmer menulis kode berikut.

```
stok = [5, 0, 0, 3, 8]
i = 0
kosong = 0
while i < len(stok):
    if stok[i] == 0:
        kosong += 1
        stok[i] = stok[i - 1] * 2
    i += 1
print(kosong, sum(stok))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. 2 46
B. 2 26
C. 2 16
D. 1 26
E. 2 36

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

13. Aplikasi pemesanan katering sekolah memanggil API di server (2027). Programmer mencatat permintaan dan respons berikut.

No	Permintaan dari aplikasi	Kode status	Pesan server
1	POST /api/login (kata sandi salah)	401	Unauthorized
2	POST /api/login (kata sandi benar)	200	token diberikan
3	POST /api/pesanan tanpa token	401	Unauthorized
4	POST /api/pesanan dengan token, jumlah porsi −3	400	jumlah tidak valid
5	POST /api/pesanan dengan token, jumlah porsi 3	201	Created

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Permintaan nomor 3 gagal karena server sedang mengalami galat.	○	○
Kode 400 pada nomor 4 menunjukkan bahwa server tidak dapat dijangkau oleh aplikasi.	○	○
Permintaan nomor 5 berhasil membuat data pesanan baru di server.	○	○

14. Gim lomba antar-SMK *Pixel Cup* (2027) memakai kode voucher. Empat huruf pertama kode menunjukkan jurusan, dan dua karakter terakhir adalah nomor urut. Programmer menguji kode berikut.

```
kode = "PPLG27-AB12"
print(kode[:4], kode[-2:], len(kode))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)

Keluaran program tersebut adalah ....

A. `PPLG2 12 11`

B. `PPLG 21 11`

C. `PPLG 12 10`

D. `PPLG 12 11`

E. `PPL 12 11`

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

15. Aplikasi kuis kelas (2027) memakai prosedur berikut untuk mencatat skor setelah bonus. Programmer menguji hasilnya.

```
def tambah_bonus(skor, daftar):
    skor = skor + 10
    daftar.append(skor)

s = 50
d = []
tambah_bonus(s, d)
print(s, d)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `60 [60]`
B. `50 []`
C. `50 [60]`
D. `60 []`
E. Error, karena daftar tidak dikembalikan dengan return

16. Gim *Jelajah Kota* (2027) menampilkan warna ikon baterai di pojok layar. Programmer menulis fungsi berikut.

```
def ikon(persen, mengisi):
    if mengisi:
        return "petir"
    elif persen <= 15:
        return "merah"
    elif persen < 50:
        return "kuning"
    return "hijau"
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang fungsi tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `ikon(15, False)` menghasilkan "kuning".
☐ B. `ikon(50, False)` menghasilkan "kuning".
☐ C. `ikon(10, True)` menghasilkan "petir".
☐ D. `ikon(49, False)` menghasilkan "kuning".
☐ E. Bila pemeriksaan mengisi dipindah ke urutan terakhir sebelum hijau, ikon(10, True) tetap "petir".

17. Klinik Sehat Sentosa ingin beralih dari kartu pasien kertas ke aplikasi (2027). Perusahaan pengembang menugasi seseorang untuk mempelajari alur pendaftaran pasien, pemeriksaan, dan pembayaran di klinik; mewawancarai petugas; menggambar diagram alur proses yang sekarang dan yang diusulkan; lalu menyusun dokumen kebutuhan sistem yang menjadi pegangan programmer.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Profesi yang paling sesuai dengan tugas tersebut adalah ....

A. Project manager yang mengatur jadwal, anggaran, dan pembagian tugas
B. Programmer yang menulis kode fitur pendaftaran dan pembayaran
C. UI designer yang menggambar tampilan layar aplikasi klinik
D. System analyst yang menganalisis proses dan kebutuhan sistem
E. QA tester yang menguji aplikasi berdasarkan skenario uji

18. Aplikasi kelas daring *RuangBelajar* (2027) memiliki fitur siaran video guru secara langsung, obrolan teks, unggah berkas tugas, dan halaman masuk dengan kata sandi. Tim sedang memilih protokol untuk setiap fitur dan menuliskan alasannya dalam dokumen rancangan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat dalam dokumen rancangan tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Video langsung cocok dikirim lewat UDP karena gambar yang terlambat lebih baik dilewati.
☐ B. Unggah berkas tugas memakai TCP agar berkas sampai utuh tanpa bagian yang hilang.
☐ C. UDP lebih cocok untuk unggah berkas tugas karena kiriman UDP lebih cepat sampai.
☐ D. Halaman masuk memakai HTTPS yang berjalan di atas koneksi TCP.
☐ E. TCP dan UDP bekerja di lapisan aplikasi karena dipilih sesuai fitur aplikasi.

19. Saat mencoba situs pendaftaran lomba milik sekolah lain (2027), Ilham, siswa PPLG, tanpa sengaja menemukan bahwa daftar nama dan nomor telepon peserta dapat dilihat siapa saja hanya dengan mengubah angka di alamat halaman. Ilham tidak bekerja untuk sekolah itu, dan situs tersebut mencantumkan alamat email pengelolanya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan Ilham yang paling sesuai dengan etika profesi adalah ....

A. Melaporkan temuan secara pribadi kepada pengelola tanpa menyebarkan data
B. Mengunduh seluruh data peserta sebagai bukti, lalu mengirimkannya ke pengelola
C. Membagikan tangkapan layar data peserta ke grup kelas sebagai peringatan
D. Mencoba mengubah data peserta untuk memastikan celahnya memang berbahaya
E. Mengunggah cara memakai celah itu ke media sosial agar cepat diperbaiki

20. Tim merancang class `Rekening` untuk aplikasi tabungan siswa (2027) dengan notasi diagram class: + publik, – privat (hanya class itu), # terproteksi (class itu dan subclass-nya). Kebutuhannya:
• nama pemilik boleh dibaca oleh class mana pun;
• saldo tidak boleh diakses langsung oleh class lain mana pun, termasuk subclass;

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• biaya_admin dipakai juga oleh subclass RekeningPelajar untuk menghitung potongan khusus, tetapi tidak boleh diakses class lain.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Rancangan access modifier yang tepat adalah ....

A. + nama, + saldo, + biaya_admin
B. − nama, − saldo, − biaya_admin
C. + nama, # saldo, − biaya_admin
D. + nama, − saldo, # biaya_admin
E. # nama, − saldo, + biaya_admin

21. Aplikasi toko daring sekolah (2027) memiliki beberapa jenis potongan harga. Programmer menguji ketiganya untuk barang seharga Rp40.000.

```
class Diskon:
    def potong(self, h):
        return h
class DiskonPersen(Diskon):
    def __init__(self, p):
        self.p = p
    def potong(self, h):
        return h - h * self.p // 100
class DiskonTetap(Diskon):
    def __init__(self, n):
        self.n = n
    def potong(self, h):
        return max(h - self.n, 0)
for d in [DiskonPersen(10), DiskonTetap(50000), Diskon()]:
    print(d.potong(40000))
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program secara berurutan adalah ....

A. 36000, lalu -10000, lalu 40000
B. 36000, lalu 0, lalu 40000
C. 40000, lalu 40000, lalu 40000
D. 4000, lalu 0, lalu 40000
E. 36000, lalu 0, lalu 0

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

22. Aplikasi Sekolah Pintar (2027) mengatur perangkat listrik dengan class berikut. Programmer lupa satu baris pada konstruktor subclass.

```
class Perangkat:
    def __init__(self, nama):
        self.nama = nama
        self.daya = 10

    def info(self):
        return self.nama + " " + str(self.daya) + " W"

class Kipas(Perangkat):
    def __init__(self, nama, kecepatan):
        self.kecepatan = kecepatan

k = Kipas("Kipas lab", 3)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang kode tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `k.daya` bernilai 10 karena Kipas adalah subclass Perangkat.
☐ B. `k.kecepatan` bernilai 3.
☐ C. Pemanggilan `k.info()` menimbulkan error karena atribut nama belum dibuat.
☐ D. Kipas tidak mewarisi method info() karena menulis konstruktornya sendiri.
☐ E. Menambah `super().__init__(nama)` sebagai baris pertama konstruktor Kipas memperbaiki masalah.

23. Aplikasi Sekolah Pintar (2027) mengendalikan lampu di setiap ruang. Programmer menulis class berikut.

```
class Lampu:
    def __init__(self, ruang, nyala=False):
        self.ruang = ruang
        self.nyala = nyala

    def tekan(self):
        self.nyala = not self.nyala

a = Lampu("kelas")
b = Lampu("lab", True)
a.tekan()
b.tekan()
b.tekan()
print(a.nyala, b.nyala)
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `False True`
B. `True False`
C. `False False`
D. Error, karena nyala tidak dikirim saat membuat a
E. `True True`

24. Tim PKL merancang class `Buku` untuk aplikasi peminjaman buku perpustakaan sekolah (2027). Rapat menghasilkan daftar calon atribut: judul, kode_buku, penulis, status_pinjam, tanggal_kembali, warna_sampul, dan berat_gram. Ketua tim mengingatkan agar class dirancang dengan prinsip abstraksi.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang rancangan class Buku adalah ....
(Jawaban benar lebih dari satu.)

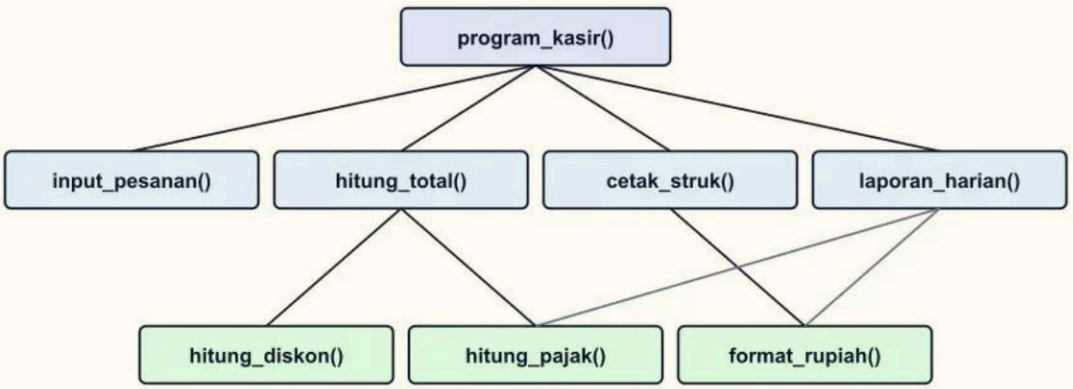
Pilih semua jawaban yang benar.

☐ A. Semakin banyak atribut, semakin baik class Buku karena data buku makin lengkap.
☐ B. warna_sampul dan berat_gram tidak perlu karena tidak dipakai dalam proses peminjaman.
☐ C. kode_buku perlu dimasukkan karena dipakai untuk mencari dan mencatat peminjaman.
☐ D. Abstraksi berarti semua atribut class harus disembunyikan dari class lain.
☐ E. status_pinjam perlu dimasukkan agar aplikasi tahu buku sedang tersedia atau dipinjam.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

25. Gambar menunjukkan diagram struktur program kasir kantin. Pada tahun 2027 tarif pajak berubah, dan sekolah juga meminta penulisan uang diubah dari "Rp 10.000" menjadi "Rp10.000" di semua tampilan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat berdasarkan diagram adalah ....
(Jawaban benar lebih dari satu.)

Diagram struktur program kasir kantin SMK Wiyata (2027)



Garis dari atas ke bawah = fungsi di atas memanggil fungsi di bawahnya.

Pilih semua jawaban yang benar.

☐ A. Perubahan tarif pajak dilakukan di hitung_pajak() dan otomatis berlaku untuk total belanja dan laporan harian.

☐ B. format_rupiah() dipakai cetak_struk() dan laporan_harian(), sehingga perubahannya perlu diuji pada keduanya.

☐ C. input_pesanan() perlu diubah karena pajak dihitung dari pesanan yang dimasukkan kasir.

☐ D. hitung_diskon() dapat diuji tersendiri tanpa menjalankan cetak_struk().

☐ E. Diagram menunjukkan hitung_pajak() dipakai ulang oleh dua bagian program tanpa ditulis dua kali.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Lembar jawaban

No	A	B	C	D	E
1	○	○	○	○	○
2					
3	○	○	○	○	○
4	○	○	○	○	○
5	○	○	○	○	○
6					
7	○	○	○	○	○
8	○	○	○	○	○
9					

No	A	B	C	D	E
10	○	○	○	○	○
11					
12	○	○	○	○	○
13					
14	○	○	○	○	○
15	○	○	○	○	○
16					
17	○	○	○	○	○
18					

No	A	B	C	D	E
19	○	○	○	○	○
20	○	○	○	○	○
21	○	○	○	○	○
22					
23	○	○	○	○	○
24					
25					

Soal PG kompleks / Benar–Salah (nomor 2, 6, 9, 11, 13, 16, 18, 22, 24, 25): jawab langsung pada soalnya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 1.1 · Profesi dan peluang usaha PPLG	9, 17	___ / 2	Ulangi hlm. 9
Sub bab 1.2 · Manajemen proyek dan budaya mutu	2, 3	___ / 2	Ulangi hlm. 23
Sub bab 2.1 · K3LH dan budaya kerja profesional	11, 19	___ / 2	Ulangi hlm. 52
Sub bab 2.2 · Pengelolaan aset fisik dan digital	1	___ / 1	Ulangi hlm. 66
Sub bab 2.3 · Pengelolaan file proyek, source code, dan dokumentasi	8	___ / 1	Ulangi hlm. 79
Sub bab 3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi	4, 7	___ / 2	Ulangi hlm. 104
Sub bab 3.2 · Konfigurasi jaringan dasar	5	___ / 1	Ulangi hlm. 118
Sub bab 3.3 · Layer TCP/IP dan fungsi protokol	18	___ / 1	Ulangi hlm. 133
Sub bab 3.4 · Mekanisme komunikasi jaringan	13	___ / 1	Ulangi hlm. 145
Sub bab 4.1 · Tipe data dan struktur data	10, 14	___ / 2	Ulangi hlm. 172
Sub bab 4.2 · Struktur kontrol percabangan	16	___ / 1	Ulangi hlm. 186
Sub bab 4.3 · Struktur kontrol perulangan	6, 12	___ / 2	Ulangi hlm. 201
Sub bab 4.4 · Fungsi, prosedur, dan modularisasi	15, 25	___ / 2	Ulangi hlm. 215
Sub bab 5.1 · Class, object, dan prinsip dasar OOP	23, 24	___ / 2	Ulangi hlm. 243
Sub bab 5.2 · Enkapsulasi, access modifier, dan getter–setter	20	___ / 1	Ulangi hlm. 259
Sub bab 5.3 · Pewarisan	22	___ / 1	Ulangi hlm. 276
Sub bab 5.4 · Polymorphism	21	___ / 1	Ulangi hlm. 293

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Simulasi 3 — Prediksi TKA 2026

SMKPPLG-FIN3

Petunjuk

Jumlah soal: 25. Waktu: 60 menit.

Kerjakan sekaligus tanpa membuka materi. Tandai jawaban pada lembar jawaban. Kunci dan pembahasan ada di Bagian D; hitung skor dan lihat tabel anjuran di akhir paket.

Kerjakan Simulasi 3 di LMS

Hasil dan analisis per sub bab tersimpan di LMS.

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg&node=SMKPPLG-FIN3

1. Studio gim seluler *Petani Cilik* (2027) melihat banyak pemain berhenti bermain di level 4. Studio membutuhkan orang yang mengolah data rekaman permainan ribuan pemain, mencari titik yang paling sering membuat pemain berhenti, membandingkan dua versi level 4 dengan uji A/B, lalu menyajikan temuannya dalam laporan untuk tim perancang gim.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Profesi yang paling sesuai dengan tugas tersebut adalah ....

A. Game designer yang merancang ulang aturan dan rintangan level 4
B. QA tester yang menguji level 4 untuk menemukan kesalahan program
C. Data analyst yang mengolah data permainan dan menyajikan temuan
D. Backend developer yang membangun server penyimpan data pemain
E. Community manager yang menjawab keluhan pemain di media sosial

2. Raka, lulusan SMK PPLG, menerima pesanan aplikasi katalog untuk toko oleh-oleh (2027). Rincian biaya dan laba yang ia inginkan seperti pada tabel.

Komponen	Rincian
Upah kerja	40 jam × Rp50.000
Sewa server dan domain 1 tahun	Rp1.200.000
Laba yang diinginkan	20 % dari total biaya

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Harga penawaran yang tepat untuk pesanan tersebut adalah ....

A. Rp3.200.000
B. Rp3.600.000
C. Rp3.840.000
D. Rp4.000.000
E. Rp4.320.000

3. Rapat harian tim aplikasi pemesanan katering pada Juni 2027 menghasilkan laporan berikut.
• Andi: fitur masuk (login) selesai, hari ini lanjut halaman profil.
• Sari: fitur pembayaran belum dapat diuji sejak 3 hari lalu karena kunci akses mode uji dari penyedia pembayaran milik klien belum dikirim; selama itu ia menunggu.
• Budi: memperbaiki tampilan menu, perkiraan selesai besok.
Demo ke klien dijadwalkan 5 hari lagi, dan pembayaran adalah fitur utama demo.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan ketua tim yang paling tepat adalah ....

A. Meminta kunci uji ke klien hari ini; sementara itu Sari memakai data tiruan
B. Menunggu klien mengirim kunci sendiri, sementara Sari dipindah ke halaman profil
C. Menghapus fitur pembayaran dari demo agar jadwal tetap terpenuhi tanpa kunci
D. Meminta Sari memakai kunci akses produksi koperasi lain yang pernah ia pegang
E. Meminta Andi membantu Sari menulis ulang fitur pembayaran dari awal

4. Saat siswa membuka aplikasi web di server lab dari laptopnya (2027), data mengalir lewat Wi-Fi ke access point, lalu lewat kabel UTP ke switch lab, ke router sekolah, dan akhirnya ke server di jaringan server. Guru meminta siswa mengaitkan perjalanan data itu dengan model TCP/IP.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Router memilih jalur paket antarjaringan berdasarkan alamat IP tujuan, sehingga bekerja di lapisan internet.	○	○
Router membaca nomor port tujuan untuk memilih jalur paket menuju server.	○	○
Sinyal Wi-Fi dan kabel UTP merupakan media pada lapisan akses jaringan.	○	○

5. Gambar menunjukkan laporan bug yang dikirim penguji kepada tim programmer aplikasi rapor digital. Programmer mengembalikan laporan itu karena tidak dapat memunculkan ulang kesalahannya. Ketua tim lalu meminta laporan diperbaiki sesuai standar mutu tim (2027).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Perbaikan yang tepat untuk laporan tersebut adalah ....
(Jawaban benar lebih dari satu.)

Laporan Bug #57 — Aplikasi Rapor Digital

Judul	Aplikasi error
Tanggal	12 Juni 2027
Pelapor	Dina (penguji)
Perangkat	HP
Langkah-langkah	Buka aplikasi, lalu error
Hasil yang diharapkan	(kosong)
Hasil yang terjadi	tidak bisa
Lampiran	tidak ada
Tingkat keparahan	Tinggi

Pilih semua jawaban yang benar.

☐ A. Judul diperjelas dengan fitur dan gejalanya, misalnya "Tombol Simpan Nilai tidak merespons".
☐ B. Langkah-langkah ditulis rinci sampai bug muncul, termasuk menu yang dibuka dan data yang diisi.
☐ C. Kolom tanggal diganti karena laporan bug seharusnya dibuat pada hari rilis aplikasi.
☐ D. Jenis ponsel dan versi Android serta versi aplikasi dicantumkan, bukan hanya "HP".
☐ E. Tingkat keparahan dihapus agar programmer bebas menentukan prioritasnya sendiri.

6. Tim TI sekolah harus memindahkan UPS seberat 30 kg dari gudang ke ruang server di lantai yang sama (2027). UPS itu sudah dilepas dari listrik. Di gudang tersedia troli barang, dan ada dua teknisi yang sedang bertugas.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Cara memindahkan UPS yang paling tepat adalah ....

A. Mengangkat sendiri dengan punggung membungkuk agar cepat selesai
B. Menyeret UPS di lantai menuju ruang server sambil menarik kabelnya
C. Memanggul UPS di satu bahu agar satu tangan bebas membuka pintu
D. Memakai troli, atau mengangkat berdua dengan lutut ditekuk
E. Mengangkat sendiri sambil menahan napas agar tenaga lebih kuat

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

7. Router lab RPL membagikan alamat lewat DHCP dengan rentang 192.168.7.20 sampai 192.168.7.100 (2027). Sejak pagi printer lab kadang tidak dapat dipakai, dan salah satu laptop menampilkan peringatan "konflik alamat IP". Daftar perangkat yang tersambung:

Perangkat	Cara mendapat alamat	Alamat IP
Router	statis	192.168.7.1
Server uji	statis	192.168.7.10
Printer lab	statis	192.168.7.50
Laptop-03	DHCP	192.168.7.50
Laptop-04	DHCP	192.168.7.51

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat berdasarkan tabel adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Laptop-03 dan printer memakai alamat yang sama sehingga terjadi konflik.
☐ B. Server uji juga bermasalah karena alamatnya berada di luar rentang DHCP.
☐ C. Alamat statis printer berada di dalam rentang DHCP sehingga router dapat membagikannya.
☐ D. Memindahkan alamat printer ke luar rentang DHCP, misalnya 192.168.7.15, mencegah konflik berulang.
☐ E. Mengubah alamat Laptop-04 menjadi 192.168.7.50 menyelesaikan masalah.

8. Koperasi sekolah akan menjual lima laptop lama inventaris lab RPL (2027). Laptop itu pernah dipakai menyimpan data pembeli koperasi dan masih terhubung ke akun email beberapa guru. Petugas aset menyusun langkah sebelum laptop dijual.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah yang tepat dilakukan adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Menghapus folder data lalu mengosongkan Recycle Bin agar data tidak dapat dipulihkan.
☐ B. Mencadangkan data yang masih diperlukan ke penyimpanan resmi sebelum laptop dibersihkan.
☐ C. Menghapus isi media penyimpanan dengan cara ditimpa (wipe), lalu memasang ulang sistem.
☐ D. Keluar dari semua akun dan menghapus laptop dari daftar perangkat akun guru.
☐ E. Menjual laptop apa adanya karena pembeli biasanya akan memformat ulang sendiri.

9. Tim aplikasi perpustakaan memakai penomoran versi MAYOR.MINOR.PERBAIKAN dan berkas catatan perubahan (changelog) (2027). Aturannya: perbaikan bug menaikkan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

angka ketiga; fitur baru yang tidak mengubah cara pakai lama menaikkan angka kedua dan angka ketiga kembali 0; perubahan besar yang membuat cara pakai lama tidak berlaku menaikkan angka pertama. Versi saat ini 2.3.1.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Rilis yang hanya memperbaiki bug perhitungan denda keterlambatan diberi nomor 2.4.0.	○	○
Rilis yang menambah fitur pencarian berdasarkan penerbit diberi nomor 2.4.0.	○	○
Setiap rilis dicatat dalam changelog beserta perubahan yang dibawanya.	○	○

10. Tim pengembang aplikasi yang bekerja jarak jauh (2027) menyepakati tiga aturan: diskusi pekerjaan dilakukan di kanal tim, jam kerja pukul 08.00–17.00, dan setiap komentar tinjauan kode harus sopan serta menyebut alasannya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan anggota tim yang sesuai dengan budaya kerja profesional adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Menulis komentar tinjauan "Kode ini jelek, ulangi saja" agar rekan cepat memperbaikinya.
☐ B. Menulis komentar "Pajak dihitung dua kali di baris 40 dan 52; bisakah digabung?"
☐ C. Mengirim tugas mendesak ke nomor pribadi rekan pukul 23.00 hampir setiap malam.
☐ D. Menyebutkan di kanal tim bahwa ide perbaikan itu berasal dari rekan bernama Tari.
☐ E. Membahas kesalahan rekan di grup lain tanpa melibatkan rekan tersebut.

11. Laptop Rafi kehabisan ruang saat membuat build gim (2027). Ia membutuhkan ruang kosong minimal 8 GB. Isi folder proyek gimnya seperti pada tabel.

Folder	Ukuran	Keterangan
src/	40 MB	kode sumber; sudah di-push ke repositori
aset/	3,2 GB	gambar dan suara final; satu-satunya salinan ada di laptop ini
build/	6,5 GB	hasil kompilasi; dapat dibuat ulang dengan perintah build
cache/	2,1 GB	berkas sementara mesin gim; dibuat ulang otomatis
.git/	900 MB	riwayat repositori lokal

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tindakan yang paling tepat adalah ....

A. Menghapus aset/ dan cache/, membebaskan 5,3 GB
B. Menghapus .git/ dan src/, membebaskan sekitar 0,94 GB
C. Menghapus build/ saja karena ukurannya paling besar
D. Menghapus build/ dan cache/, membebaskan 8,6 GB
E. Menghapus aset/ dan build/, membebaskan 9,7 GB

12. Proyek tim memakai sintaks `match–case` yang baru tersedia sejak Python 3.10, dan README proyek menulis "Membutuhkan Python 3.11" (2027). Di laptop Yoga terpasang Python 3.8. Saat program dijalankan, muncul `SyntaxError` tepat pada baris yang memuat match.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Langkah Yoga yang paling tepat adalah ....

A. Mengganti semua match–case menjadi if–elif tanpa memberi tahu tim
B. Memasang ulang VS Code dengan versi terbaru agar sintaks dikenali
C. Menyalin folder Python 3.11 dari laptop teman ke laptopnya
D. Menjalankan program sebagai administrator agar sintaks dikenali
E. Memasang Python 3.11 sesuai README lalu menjalankan dengannya

13. Tim gim *Kuda Lumping Run* (2027) menyimpan semua gambar di satu folder cloud bersama. Desainer sering menyimpan revisi dengan menimpa berkas lama bernama sama, misalnya `pemain.png`. Akibatnya, programmer yang sedang menguji tiba-tiba melihat ukuran gambar berubah sehingga tampilan gim rusak, dan tim tidak dapat kembali ke versi sebelumnya.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Prosedur yang paling tepat untuk mengatasi masalah tersebut adalah ....

A. Melarang desainer merevisi gambar setelah programmer mulai menguji gim
B. Memisahkan folder kerja dan folder final, memberi nomor versi, dan mencatatnya
C. Meminta programmer menyalin seluruh folder gambar ke laptopnya setiap pagi
D. Menyimpan setiap revisi dengan nama acak agar tidak ada berkas yang tertimpa
E. Mengubah izin folder sehingga hanya programmer yang dapat menyimpan gambar

14. Aplikasi pendaftaran karyawisata SMK (2027) menentukan siapa yang boleh ikut. Programmer menulis fungsi berikut.

```
def boleh_ikut(umur, izin_ortu, sehat):
    if not sehat:
        return "Tidak"
    if umur >= 17 or izin_ortu:
        return "Ya"
    return "Tidak"
```

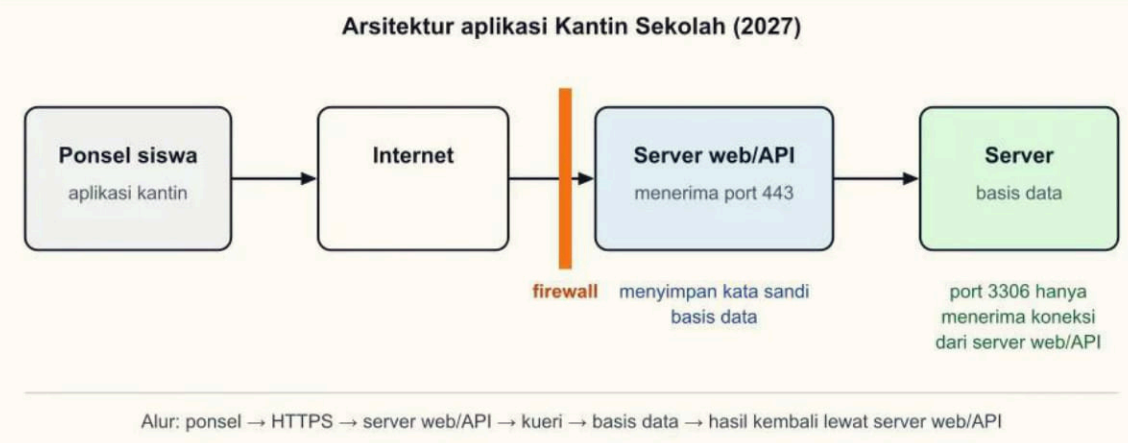

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang fungsi tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. `boleh_ikut(16, True, True)` menghasilkan "Ya".
☐ B. `boleh_ikut(18, True, False)` menghasilkan "Ya" karena umur dan izin terpenuhi.
☐ C. `boleh_ikut(16, False, True)` menghasilkan "Ya" karena siswa itu sehat.
☐ D. `boleh_ikut(17, False, True)` menghasilkan "Tidak" karena umurnya belum lebih dari 17.
☐ E. Siswa yang tidak sehat ditolak apa pun umur dan izinnya karena sehat diperiksa lebih dulu.

15. Gambar menunjukkan arsitektur aplikasi pemesanan Kantin Sekolah. Seorang programmer baru mengusulkan agar aplikasi di ponsel siswa langsung tersambung ke server basis data supaya data tampil lebih cepat (2027).
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tanggapan yang paling tepat beserta alasannya berdasarkan gambar adalah ....



A. Tidak tepat; port 3306 hanya menerima server web/API, jadi ponsel lewat port 443
B. Tepat; port 3306 terbuka untuk internet, sehingga ponsel dapat langsung tersambung
C. Tepat, asalkan ponsel memakai kata sandi basis data yang disimpan di aplikasinya
D. Tidak tepat; firewall terpasang di antara server web/API dan server basis data
E. Tidak tepat; basis data memakai port 443 yang sudah dipakai server web/API

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

16. Dalam gim *Kerajaan Nusantara* (2027), raja seharusnya memiliki 500 hp. Programmer menulis kode berikut, tetapi hasilnya tidak seperti yang diharapkan.

```
class Tokoh:
    def __init__(self, nama):
        self.nama = nama
        self.hp = 100

class Raja(Tokoh):
    def __init__(self, nama):
        self.hp = 500
        super().__init__(nama)
        self.mahkota = True

r = Raja("Arya")
print(r.nama, r.hp, r.mahkota)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `Arya 500 True`
B. Error, karena hp diisi sebelum super().__init__()
C. `Arya 100 True`
D. `Arya 600 True`
E. `None 500 True`

17. Startup aplikasi sekolah memperkirakan biaya server naik 10 % per tahun, dibulatkan ke bawah dalam ribuan rupiah (2027). Programmer ingin mengetahui berapa tahun biaya per tahunnya masih di bawah 150 ribu rupiah dan total biaya yang dibayar selama tahun-tahun itu.

```
harga = 100
tahun = 0
total = 0
while harga < 150:
    total += harga
    harga = harga + harga // 10
    tahun += 1
print(tahun, total)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `4 464`
B. `5 670`
C. `6 770`
D. `5 160`
E. `5 610`

18. Di gim *Pasar Malam* (2027), class `Pemain` memiliki atribut koin. Di 14 tempat berbeda dalam program, bagian lain langsung menulis `pemain.koin = pemain.koin - harga`. Beberapa kali koin pemain menjadi negatif karena ada bagian yang lupa memeriksa saldo

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

sebelum mengurangi koin.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Prinsip PBO yang diabaikan beserta perbaikannya adalah ....

A. Pewarisan; buat subclass PemainKaya yang koinnya tidak dapat negatif
B. Polimorfisme; buat method beli() yang berbeda untuk setiap barang
C. Abstraksi; hapus atribut koin dan simpan koin di variabel global
D. Enkapsulasi; jadikan koin privat dan ubah lewat method berpemeriksa
E. Modularisasi; salin pemeriksaan saldo ke 14 tempat yang mengubah koin

19. Aplikasi laporan koperasi (2027) mencetak beberapa jenis laporan. Programmer menulis kode berikut.

```
class Laporan:
    def judul(self):
        return "Laporan"
    def cetak(self):
        return "[" + self.judul() + "]"
class LaporanHarian(Laporan):
    def judul(self):
        return "Harian"
class LaporanBulanan(Laporan):
    def cetak(self):
        return "<<" + self.judul() + ">>"
for r in [Laporan(), LaporanHarian(), LaporanBulanan()]:
    print(r.cetak())
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang keluaran dan kode tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Baris ketiga keluaran adalah `<<Bulanan>>`.
☐ B. Baris kedua keluaran adalah `[Harian]`.
☐ C. LaporanHarian memakai cetak() milik Laporan, tetapi judul() miliknya sendiri.
☐ D. Semua baris keluaran diawali tanda [karena cetak() ditulis di class Laporan.
☐ E. Baris ketiga keluaran adalah `<<Laporan>>`.

20. Gim musik *Irama Desa* (2027) menyimpan daftar lagu latar dalam list. Programmer mengubah daftar itu dengan kode berikut.

```
lagu = ["Hujan", "Pelangi", "Senja"]
lagu.insert(1, "Fajar")
lagu.remove("Senja")
lagu.append("Pelangi")
print(lagu, lagu.index("Pelangi"))
```


Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `['Hujan', 'Fajar', 'Pelangi', 'Pelangi'] 3`
B. `['Fajar', 'Hujan', 'Pelangi', 'Pelangi'] 2`
C. `['Hujan', 'Fajar', 'Pelangi'] 2`
D. `['Hujan', 'Pelangi', 'Fajar', 'Pelangi'] 1`
E. `['Hujan', 'Fajar', 'Pelangi', 'Pelangi'] 2`

21. Fungsi `hitung_nilai()` di aplikasi rapor (2027) mengerjakan semuanya: membaca nilai dengan input(), menghitung rata-rata, menentukan predikat, lalu mencetak hasil. Tim kini ingin memakai perhitungan yang sama untuk halaman web dan menguji perhitungan secara otomatis, tetapi fungsi itu sulit dipakai. Tim menyusun rencana pemecahan.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Usulan yang tepat dalam rencana tersebut adalah ....
(Jawaban benar lebih dari satu.)

Pilih semua jawaban yang benar.

☐ A. Fungsi rata_rata(daftar) hanya menghitung dan mengembalikan rata-rata, tanpa input() dan print().
☐ B. Fungsi predikat(rata) menerima angka dan mengembalikan predikat, sehingga dapat diuji dengan contoh angka.
☐ C. Bagian membaca masukan dan mencetak hasil dipisahkan agar halaman web dapat memakai perhitungannya.
☐ D. Setiap fungsi baru tetap memanggil input() sendiri agar dapat berjalan mandiri.
☐ E. Perubahan aturan predikat kelak hanya perlu dilakukan di fungsi predikat().

22. Siswa PKL di aplikasi kasir (2027) membaca jumlah barang dari kolom isian yang selalu berupa teks. Ia mencoba kode berikut untuk memahami perbedaan tipe data.

```
jumlah = "12"
print(int(jumlah) + 3, jumlah + "3")
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `123 15`
B. `15 123`
C. `15 15`
D. `123 123`
E. Error, karena "12" bertipe teks

23. Tim merancang class `Termometer` untuk aplikasi UKS sekolah (2027). Rancangannya: atribut __suhu dibuat privat; method get_suhu() mengembalikan suhu; method catat(nilai) hanya menerima nilai 34–42 °C dan menolak nilai lain; tidak ada method lain yang mengubah suhu.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Tentukan **Benar** atau **Salah** untuk setiap pernyataan berikut.

Tentukan setiap pernyataan Benar atau Salah.

Pernyataan	Benar	Salah
Setelah catat(36.5) lalu catat(30) dijalankan, get_suhu() menghasilkan 30 karena itulah nilai terakhir yang dicatat.	○	○
Halaman laporan dapat membaca suhu lewat get_suhu() tanpa dapat mengubahnya.	○	○
Menambah method set_suhu(nilai) tanpa pemeriksaan rentang tetap menjaga enkapsulasi karena atributnya masih privat.	○	○

24. Aplikasi absensi SMK (2027) memiliki class `Siswa` dengan atribut nama dan nis, serta class `Kelas` dengan atribut nama_kelas dan wali_kelas. Untuk kelas XI PPLG 1, program membuat satu data kelas dan data untuk 32 siswanya, masing-masing dengan nama dan nis berbeda.
(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Pernyataan yang tepat tentang program tersebut adalah ....

A. Program memiliki 2 class dan membuat 33 objek
B. Program memiliki 33 class dan membuat 2 objek
C. Program memiliki 2 class dan membuat 2 objek
D. Program memiliki 34 class dan membuat 33 objek
E. Program memiliki 1 class dan membuat 33 objek

25. Aplikasi koperasi (2027) memberi diskon biasa 10 %, sedangkan pada hari ulang tahun sekolah diskonnya 20 %. Programmer menulis dua fungsi berikut dan mengujinya untuk barang seharga Rp50.000.

```
diskon = 10

def harga_biasa(harga):
    return harga - harga * diskon // 100

def harga_ultah(harga):
    diskon = 20
    return harga - harga * diskon // 100

print(harga_ultah(50000), harga_biasa(50000), diskon)
```

(Nama orang, lembaga, dan data disusun untuk latihan soal; fiktif.)
Keluaran program tersebut adalah ....

A. `40000 40000 20`
B. `40000 45000 10`
C. `40000 45000 20`
D. `45000 45000 10`
E. Error, karena harga_biasa() tidak punya variabel diskon

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Lembar jawaban

No	A	B	C	D	E
1	○	○	○	○	○
2	○	○	○	○	○
3	○	○	○	○	○
4					
5					
6	○	○	○	○	○
7					
8					
9					

No	A	B	C	D	E
10					
11	○	○	○	○	○
12	○	○	○	○	○
13	○	○	○	○	○
14					
15	○	○	○	○	○
16	○	○	○	○	○
17	○	○	○	○	○
18	○	○	○	○	○

No	A	B	C	D	E
19					
20	○	○	○	○	○
21					
22	○	○	○	○	○
23					
24	○	○	○	○	○
25	○	○	○	○	○

Soal PG kompleks / Benar–Salah (nomor 4, 5, 7, 8, 9, 10, 14, 19, 21, 23): jawab langsung pada soalnya.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Tabel skor dan anjuran

Sub bab / subtopik	Nomor soal	Benar	Jika kurang separuh benar
Sub bab 1.1 · Profesi dan peluang usaha PPLG	1, 2	___ / 2	Ulangi hlm. 9
Sub bab 1.2 · Manajemen proyek dan budaya mutu	3, 5	___ / 2	Ulangi hlm. 23
Sub bab 2.1 · K3LH dan budaya kerja profesional	6, 10	___ / 2	Ulangi hlm. 52
Sub bab 2.2 · Pengelolaan aset fisik dan digital	8	___ / 1	Ulangi hlm. 66
Sub bab 2.3 · Pengelolaan file proyek, source code, dan dokumentasi	9, 13	___ / 2	Ulangi hlm. 79
Sub bab 3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi	11, 12	___ / 2	Ulangi hlm. 104
Sub bab 3.2 · Konfigurasi jaringan dasar	7	___ / 1	Ulangi hlm. 118
Sub bab 3.3 · Layer TCP/IP dan fungsi protokol	4	___ / 1	Ulangi hlm. 133
Sub bab 3.4 · Mekanisme komunikasi jaringan	15	___ / 1	Ulangi hlm. 145
Sub bab 4.1 · Tipe data dan struktur data	20, 22	___ / 2	Ulangi hlm. 172
Sub bab 4.2 · Struktur kontrol percabangan	14	___ / 1	Ulangi hlm. 186
Sub bab 4.3 · Struktur kontrol perulangan	17	___ / 1	Ulangi hlm. 201
Sub bab 4.4 · Fungsi, prosedur, dan modularisasi	21, 25	___ / 2	Ulangi hlm. 215
Sub bab 5.1 · Class, object, dan prinsip dasar OOP	18, 24	___ / 2	Ulangi hlm. 243
Sub bab 5.2 · Enkapsulasi, access modifier, dan getter–setter	23	___ / 1	Ulangi hlm. 259
Sub bab 5.3 · Pewarisan	16	___ / 1	Ulangi hlm. 276
Sub bab 5.4 · Polymorphism	19	___ / 1	Ulangi hlm. 293

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Bagian D · Kunci dan Pembahasan

Urutan sama dengan buku. Setiap pembahasan memuat konsep, langkah, dan alasan tiap pilihan. Baca pembahasan soal yang kamu jawab salah — dan juga yang kamu jawab benar tetapi menebak.

Latihan sub bab 1.1

No. 1 Kunci: C

Konsep: Profesi di tim pengembang gim dibedakan menurut tugas utamanya: merancang permainan (game designer), merancang antarmuka (UI/UX), menulis kode (programmer/developer), menguji mutu (QA tester), dan merilis (release engineer).

Langkah:

1. Kata kunci tugas: memainkan sesuai skenario uji, mencatat kesalahan dan langkah memunculkannya, melapor ke pelacak bug, menguji ulang.
2. Rangkaian itu adalah pekerjaan penjaminan mutu (quality assurance), yaitu QA tester.

Analisis setiap opsi:

- **A.** ✗ Jebakan: game designer juga memainkan prototipe, tetapi untuk merancang aturan dan keseruan, bukan mencatat bug sesuai skenario uji.
- **B.** ✗ UI/UX designer merancang tampilan dan alur pemakaian; menguji ulang perbaikan bug bukan tugas utamanya.
- **C.** ✓ QA tester menjalankan skenario uji, melaporkan bug dengan langkah reproduksi, dan menguji ulang perbaikannya.
- **D.** ✗ Backend developer menulis kode server; ia yang memperbaiki bug, bukan yang mencari dan menguji ulang.
- **E.** ✗ Jebakan: release engineer bekerja di akhir menjelang rilis, tetapi tugasnya mengemas dan mengunggah, bukan menguji isi gim.

Jawaban: C.

No. 2 Kunci: D

Konsep: Dalam tim gim, level designer merancang isi dan alur setiap level (letak rintangan, musuh, hadiah, kenaikan kesulitan), game artist membuat aset visual, game programmer menulis kode, sound designer membuat audio, dan QA tester menguji mutu.

Langkah:

1. Tugas inti: menentukan letak pijakan, rintangan, musuh, dan koin per level, serta menaikkan kesulitan bertahap.
2. Prototipe kotak-kotak dipakai untuk menguji rancangan level sebelum aset gambar jadi.
3. Pekerjaan ini milik level designer.

Analisis setiap opsi:

- **A.** ✗ Game artist membuat gambar karakter dan latar, padahal di kasus pekerjaan dilakukan sebelum gambar selesai.
- **B.** ✗ Jebakan: prototipe memang diprogram, tetapi yang dicari adalah orang yang merancang isi level, bukan yang menulis kode gerak.
- **C.** ✗ Sound designer mengurus audio, tidak menyusun rintangan dan kesulitan level.
- **D.** ✓ Level designer merancang letak rintangan, musuh, hadiah, dan kenaikan kesulitan tiap level.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **E.** ✗ Jebakan: QA juga mencoba gim, tetapi tugasnya mencari dan melaporkan bug, bukan merancang level.

Jawaban: D.

No. 3 Kunci: B

Konsep: UI/UX designer meneliti kebutuhan pengguna, merancang tampilan (wireframe, prototipe), dan menguji kemudahan pemakaian (usability test) sebelum tampilan diprogram.

Langkah:
1. Kegiatan: wawancara pengguna, rancangan layar, prototipe yang dapat diklik, pengamatan uji coba.
2. Semuanya dilakukan sebelum pemrograman dan berfokus pada pengalaman pengguna → UI/UX designer.

Analisis setiap opsi:
• **A.** ✗ Jebakan: frontend developer juga mengurus tampilan, tetapi ia memprogram rancangan yang sudah jadi, bukan meneliti pengguna.
• **B.** ✓ UI/UX designer meneliti pengguna, membuat prototipe, dan menguji kemudahan pakai.
• **C.** ✗ Database administrator mengurus data di balik aplikasi, bukan tampilan.
• **D.** ✗ Technical writer menulis panduan setelah aplikasi jadi, bukan merancang layar.
• **E.** ✗ Mobile developer menulis kode aplikasi; rancangan layarnya dibuat lebih dulu oleh UI/UX designer.

Jawaban: B.

No. 4 Kunci: A, C, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Usaha rintisan perangkat lunak dimulai dengan memvalidasi masalah calon pelanggan, membuat produk minimum (MVP), menguji dan memperbaikinya dari masukan pengguna, serta menetapkan harga yang menutup biaya. Semua aset harus legal.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Memvalidasi masalah ke calon pelanggan adalah langkah awal agar modal tidak habis untuk produk yang tidak dibutuhkan.
• **Pernyataan 2** ✗ Jebakan: aplikasi tampak lengkap, tetapi modal dan waktu habis sebelum terbukti ada yang mau memakai dan membayar.
• **Pernyataan 3** ✓ MVP menguji gagasan dengan biaya kecil dan menghasilkan masukan nyata dari pengguna.
• **Pernyataan 4** ✓ Harga yang menutup biaya dan sesuai daya beli pelanggan membuat usaha dapat bertahan.
• **Pernyataan 5** ✗ Memakai aset milik pihak lain tanpa izin melanggar hak cipta dan dapat berujung tuntutan.

Jawaban: pernyataan 1, 3, 4.

No. 5 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Peluang usaha dianalisis dari kebutuhan pasar, kemampuan bayar calon pelanggan, dan pesaing. Produk, harga, dan cara pembayaran disusun dari data itu dengan tetap legal dan menutup biaya.

Analisis setiap pernyataan:
• **Pernyataan 1** ✗ Jebakan: adanya pesaing tidak menutup peluang; harga pesaing jauh di atas kemampuan sebagian besar UMKM, sehingga ada celah pasar.
• **Pernyataan 2** ✓ Produk dirancang dari kebutuhan nyata calon pelanggan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 3** ✓ Skema ini menyesuaikan kemampuan bayar UMKM dan memberi pemasukan rutin.
• **Pernyataan 4** ✗ Template bajakan melanggar hak cipta dan merusak kepercayaan pelanggan.
• **Pernyataan 5** ✓ Harga bersaing tetap harus menutup biaya agar usaha bertahan.

Jawaban: pernyataan 2, 3, 5.

Latihan sub bab 1.2

No. 1 Kunci: E

Konsep: Dalam manajemen proyek, perubahan ruang lingkup di tengah pelaksanaan dikendalikan: dicatat, dianalisis dampaknya pada waktu, biaya, dan mutu, lalu diputuskan bersama klien (misalnya jadwal diundur atau fitur masuk tahap berikutnya).

Langkah:
1. QRIS berada di luar ruang lingkup yang disepakati → ini permintaan perubahan.
2. Menambah fitur tanpa mengubah jadwal mengancam mutu; menolak mentah-mentah merusak hubungan dengan klien.
3. Langkah yang benar: catat, hitung dampaknya, lalu ajukan pilihan agar klien yang memutuskan.

Analisis setiap opsi:
• **A.** ✗ Jebakan: tampak melayani klien, tetapi menambah beban tanpa analisis dampak berisiko membuat jadwal dan mutu sama-sama gagal.
• **B.** ✗ Jebakan: dokumen ruang lingkup memang tidak memuat QRIS, tetapi menolak tanpa menawarkan pilihan mengabaikan kebutuhan klien.
• **C.** ✗ Mengorbankan pengujian menurunkan mutu; bug pembayaran justru paling merugikan.
• **D.** ✗ Bu Rina tetap berharap QRIS ada saat rilis; menunda pembicaraan membuat harapan klien tidak terkelola.
• **E.** ✓ Permintaan dicatat dan dampaknya dihitung, lalu Bu Rina memilih: jadwal diundur atau QRIS masuk tahap berikutnya.

Jawaban: E.

No. 2 Kunci: D

Konsep: Pemantauan dan pengendalian proyek membandingkan realisasi dengan rencana. Bila sisa pekerjaan melebihi kemampuan tim dan tenggat tidak dapat digeser, ruang lingkup diprioritaskan bersama klien (fitur wajib lebih dulu).

Langkah:
1. Realisasi dua sprint terakhir: 22 dan 21 → sekitar 21–22 poin per sprint.
2. Sisa 2 sprint × ±21,5 = ±43 poin, padahal sisa pekerjaan 77 poin → tidak akan selesai semuanya.
3. Fitur wajib 40 poin masih muat dalam ±43 poin → dahulukan fitur wajib, fitur tambahan dirilis berikutnya, disepakati dengan klien.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:
• **A.** ✗ Jebakan: menambah orang di akhir proyek justru memperlambat karena mereka perlu dikenalkan pada kode dan proyek.
• **B.** ✗ Mengorbankan pengujian berisiko meloloskan bug pada fitur pendaftaran dan pembayaran yang paling penting.
• **C.** ✗ Data dua sprint terakhir menunjukkan realisasi stabil sekitar 21–22 poin; berharap tanpa tindakan tidak mengendalikan proyek.
• **D.** ✓ Kemampuan ±43 poin cukup untuk 40 poin fitur wajib; fitur tambahan dijadwalkan ulang bersama klien.
• **E.** ✗ Tanggal rilis terikat penerimaan siswa baru dan perubahan jadwal harus dibicarakan dengan klien.

Jawaban: D.

No. 3 Kunci: 1B 2S 3B

Konsep: Budaya mutu dalam pengembangan perangkat lunak berarti mematuhi standar yang disepakati (definisi selesai, code review, pengujian) dan belajar dari cacat agar tidak berulang (perbaikan berkelanjutan).

Analisis setiap pernyataan:
• **Pernyataan 1 → Benar.** Uji di ponsel Android lama termasuk definisi selesai, tetapi dilewati. Cacat yang seharusnya tertangkap lolos ke pengguna.
• **Pernyataan 2 → Salah.** Jebakan: rilis memang lebih cepat, tetapi cacat yang sama akan lolos lagi. Budaya mutu memperbaiki proses, bukan menghapus pemeriksaan.
• **Pernyataan 3 → Benar.** Menambah kasus uji dari cacat yang pernah lolos adalah bagian dari perbaikan berkelanjutan.

Jawaban: Benar, Salah, Benar.

No. 4 Kunci: A, C, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Kriteria penerimaan (acceptance criteria) adalah syarat terukur yang disepakati dengan klien. Pemeriksaan mutu membandingkan hasil kerja dengan setiap kriteria; pekerjaan baru dinyatakan selesai bila semua kriteria terpenuhi.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Kolom kedua menampilkan titik-titik dan ada ikon mata di ujung kanannya.
• **Pernyataan 2** ✗ Jebakan: tautan yang ada adalah "Belum punya akun? Daftar", bukan "Lupa kata sandi?".
• **Pernyataan 3** ✓ Kolom NIS punya label di atasnya, tetapi kolom kata sandi tidak memiliki label.
• **Pernyataan 4** ✓ Tombol Masuk berada di bawah kedua kolom dengan lebar yang sama.
• **Pernyataan 5** ✗ Budaya mutu: hasil kerja dinyatakan selesai bila semua kriteria penerimaan terpenuhi; dua kriteria belum terpenuhi.

Jawaban: pernyataan 1, 3, 4.

No. 5 Kunci: 1S 2B 3B

Konsep: Budaya mutu dalam pemrograman meliputi mematuhi pedoman penulisan kode (penamaan, keterangan), peninjauan rekan, dan pengujian, sehingga kode mudah dibaca, dirawat, dan dikembangkan tim.

Analisis setiap pernyataan:
• **Pernyataan 1 → Salah.** Jebakan: hasil memang sama, tetapi kode yang sulit dibaca menyulitkan anggota lain memperbaiki dan mengembangkannya.
• **Pernyataan 2 → Benar.** Nama deskriptif menjelaskan maksud kode tanpa harus menelusuri seluruh

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

isinya.

• **Pernyataan 3 → Benar.** Code review adalah bagian dari budaya mutu: kesalahan tertangkap lebih awal dan gaya kode seragam.

Jawaban: Salah, Benar, Benar.

Kuis Akhir Bab 1

No. 1 Kunci: D

Konsep: Profesi di bidang PPLG dibedakan oleh objek kerja utamanya. Administrator basis data (DBA) bertanggung jawab atas keamanan, ketersediaan, dan kecepatan basis data, bukan atas fitur aplikasi.

Langkah:

1. Kumpulkan kata kunci tugas: cadangan basis data, kueri lambat, hak akses tabel, uji pemulihan.
2. Semua tugas itu berpusat pada satu objek: basis data. Tidak ada tugas menulis fitur, mengelola sistem operasi, menganalisis angka bisnis, atau mencari celah keamanan.

Analisis setiap opsi:

• **A.** ✗ Jebakan: pengembang sisi server memang berhubungan dengan basis data, tetapi tugas utamanya menulis logika aplikasi, sedangkan staf ini tidak menulis fitur.

• **B.** ✗ Jebakan: analis data memakai isi basis data untuk laporan dan keputusan bisnis, bukan menjaga cadangan dan hak aksesnya.

• **C.** ✗ Jebakan: administrator sistem operasi mengurus server (pembaruan, layanan, pengguna sistem), bukan isi dan kinerja basis data.

• **D.** ✓ Cadangan, kecepatan kueri, hak akses, dan uji pemulihan adalah tugas inti administrator basis data.

• **E.** ✗ Penguji keamanan mencari celah, bukan menjadwalkan cadangan dan mengatur hak akses harian.

Jawaban: D.

No. 2 Kunci: B

Konsep: Lima fase manajemen proyek: inisiasi (menilai kelayakan, mengenali pemangku kepentingan, piagam proyek) → perencanaan (jadwal, anggaran, pembagian tugas) → pelaksanaan → pemantauan dan pengendalian → penutupan.

Langkah:

1. Kegiatan tim: menilai kelayakan, mencatat pihak berkepentingan, menyepakati piagam proyek.
2. Ketiganya khas fase awal, sebelum jadwal dan anggaran disusun. Itulah inisiasi.

Analisis setiap opsi:

• **A.** ✗ Pelaksanaan = mengerjakan produk; tim belum menulis kode.

• **B.** ✓ Kelayakan, pemangku kepentingan, dan piagam proyek adalah isi fase inisiasi.

• **C.** ✗ Jebakan: perencanaan juga memakai tujuan proyek, tetapi isinya jadwal, anggaran, dan pembagian tugas, yang belum disusun tim.

• **D.** ✗ Pemantauan dan pengendalian membandingkan kemajuan dengan rencana; belum ada rencana.

• **E.** ✗ Penutupan terjadi di akhir proyek.

Jawaban: B.

No. 3 Kunci: A, C, E

Konsep: Profesi dikenali dari kata kerja dan objek kerjanya: analis sistem menggali kebutuhan, front-end atau mobile developer menulis kode tampilan, administrator basis data mengelola tabel dan hak akses, dan QA tester menguji lalu melapor. Penguji tidak memperbaiki kode; pembuat kodenya yang memperbaiki.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah: Cocokkan kata kerja tiap orang dengan profesi: menggali dan menuliskan kebutuhan (Alya), menulis kode tampilan (Bram), mengelola tabel dan hak akses (Citra), menjalankan dan melapor (Dani). Lalu nilai tiap pernyataan.

Analisis setiap opsi:

- **A.** ✓ Menggali dan menuliskan kebutuhan fitur adalah tugas analis sistem.
- **B.** ✗ Jebakan: kode memang memakai data, tetapi menulis kode tampilan bukan tugas administrator basis data; yang mengelola tabel dan hak akses adalah Citra.
- **C.** ✓ Mengelola tabel dan hak akses data adalah tugas administrator basis data.
- **D.** ✗ Jebakan: QA tester menguji dan melaporkan; memperbaiki kode adalah tugas pembuat kodenya.
- **E.** ✓ Laporan bug diteruskan kepada pembuat bagian itu (di sini Bram) untuk diperbaiki.

Jawaban: A, C, dan E.

No. 4 Kunci: A, C, D

Konsep: Business Model Canvas memetakan model bisnis dalam sembilan blok, antara lain segmen pelanggan (siapa yang dilayani), proposisi nilai (manfaat yang dijanjikan), saluran (cara menjangkau pelanggan), sumber pendapatan (dari mana uang masuk), dan struktur biaya (ke mana uang keluar).

Langkah: Tanyakan tiap catatan: siapa, manfaat apa, lewat apa, uang masuk, atau uang keluar?

Analisis setiap opsi:

- **A.** ✓ Kelompok orang yang dilayani adalah segmen pelanggan.
- **B.** ✗ Jebakan: sewa server dan honor adalah uang keluar, jadi masuk blok struktur biaya.
- **C.** ✓ Komisi adalah uang yang masuk, sehingga termasuk sumber pendapatan.
- **D.** ✓ Manfaat yang dijanjikan kepada pelanggan adalah proposisi nilai.
- **E.** ✗ Jebakan: iklan di media sosial adalah cara menjangkau pelanggan, jadi masuk blok saluran, bukan kegiatan utama.

Jawaban: A, C, dan D.

No. 5 Kunci: A, B, D

Konsep: Profesi dikenali dari kata kerja dan objek kerja: cadangan dan hak akses data (administrator basis data), logika dan data di sisi server (back-end developer), menggali kebutuhan (analis sistem), server dan rilis (DevOps), rancangan layar dan uji pemakaian (UI/UX designer).

Langkah: Untuk tiap baris tabel, tulis kata kerja dan objeknya, lalu cocokkan dengan profesi pada pilihan.

Analisis setiap opsi:

- **A.** ✓ Cadangan basis data dan hak akses tabel = administrator basis data.
- **B.** ✓ Kode server yang menyediakan data = back-end developer.
- **C.** ✗ Jebakan: mewawancarai pemilik toko dan menuliskan kebutuhan adalah tugas analis sistem, bukan penguji.
- **D.** ✓ Menyiapkan server awan dan mengemas versi rilis = DevOps engineer.
- **E.** ✗ Jebakan: merancang alur layar dan mengamati pengguna adalah UI/UX designer, bukan analis sistem.

Jawaban: A, B, dan D.

No. 6 Kunci: 1B 2B 3S

Konsep: Di tim kecil atau pada pekerja lepas, satu orang sering memegang beberapa peran. Meskipun begitu, tiap tugas tetap berbeda dan dikenali dari kata kerjanya. Kemampuan mengerjakan tampilan dan server disebut full-stack, bukan administrator basis data.

Langkah:

1. Daftar kegiatan Rudi: merancang tampilan, menulis kode tampilan dan server, memasang di hosting,

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

menguji, mencatat kesalahan.
2. Kegiatan itu mencakup beberapa profesi, sehingga pernyataan 1 benar.
3. Menguji (mencari dan mencatat kesalahan) berbeda tujuannya dari menulis kode; pernyataan 2 benar.
4. Administrator basis data bertugas atas cadangan, kecepatan, dan hak akses data; tidak ada kegiatan itu pada kasus. Pernyataan 3 salah.

Analisis:
• **1.** ✓ Satu orang memegang beberapa peran adalah hal wajar pada pekerja lepas.
• **2.** ✓ Tugas menguji tetap berbeda walaupun orangnya sama.
• **3.** ✗ Jebakan: kemampuan tampilan dan server disebut full-stack developer, bukan administrator basis data.

Jawaban: Benar, Benar, Salah.

No. 7 Kunci: B

Konsep: Hasil usaha = jumlah sumbangan semua produk − biaya tetap. Sumbangan satu produk = (harga − biaya variabel) × jumlah terjual.

Langkah:
1. Paket situs: (1.500.000 − 400.000) × 4 = 1.100.000 × 4 = 4.400.000.
2. Lisensi kasir: (300.000 − 50.000) × 20 = 250.000 × 20 = 5.000.000.
3. Jumlah sumbangan = 4.400.000 + 5.000.000 = 9.400.000.
4. Hasil = 9.400.000 − 8.200.000 = laba Rp1.200.000.

Analisis setiap opsi:
• **A.** ✗ Jebakan: Rp3.800.000 dihitung dari harga jual tanpa biaya variabel (12.000.000 − 8.200.000).
• **B.** ✓ Sumbangan 9.400.000 dikurangi biaya tetap 8.200.000.
• **C.** ✗ Jebakan: rugi Rp3.800.000 muncul bila hanya paket situs yang dihitung (4.400.000 − 8.200.000).
• **D.** ✗ Jebakan: 9.400.000 adalah jumlah sumbangan, biaya tetap belum dikurangkan.
• **E.** ✗ Jebakan: tandanya terbalik; hasil positif berarti laba, bukan rugi.

Jawaban: B.

No. 8 Kunci: A, D, E

Konsep: Pada fase pemantauan dan pengendalian, tim membandingkan kemajuan dengan rencana. Bila ada penyimpangan, tim menilai dampaknya, membicarakannya dengan klien, dan memperbarui rencana, bukan menyembunyikannya atau memotong mutu.

Langkah: Nilai tiap pilihan: apakah terbuka, menjaga mutu, dan memperbarui rencana?

Analisis setiap opsi:
• **A.** ✓ Menilai dampak pada jalur kritis dan melapor kepada klien adalah langkah pertama yang benar.
• **B.** ✗ Jebakan: lembur diam-diam menyembunyikan masalah dan membuat anggota kelelahan.
• **C.** ✗ Jebakan: memangkas pengujian akhir menurunkan mutu produk.
• **D.** ✓ Memindahkan tugas atau menunda fitur kurang penting, setelah disepakati klien, mengelola ruang lingkup dengan sehat.
• **E.** ✓ Rencana dan jadwal diperbarui agar mencerminkan keadaan nyata.

Jawaban: A, D, dan E.

No. 9 Kunci: C

Konsep: Titik impas = biaya tetap ÷ (harga jual − biaya variabel per unit). Bila harga atau biaya variabel berubah, hitung ulang selisihnya.

Langkah:
1. Selisih (sumbangan) per peserta pada harga baru = 70.000 − 20.000 = Rp50.000.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Titik impas = 3.600.000 ÷ 50.000 = 72 peserta.
3. Cek: 72 × 70.000 = 5.040.000; biaya total = 3.600.000 + 72 × 20.000 = 5.040.000 ✓.

Analisis setiap opsi:
- **A.** ✗ Jebakan: 3.600.000 ÷ 70.000 ≈ 51,4 → 52; biaya variabel per peserta terlupa.
- **B.** ✗ Jebakan: 60 adalah titik impas pada harga lama (3.600.000 ÷ 60.000).
- **C.** ✓ Dihitung dengan harga baru dan biaya variabel.
- **D.** ✗ Jebakan: 3.600.000 ÷ 40.000; selisih yang dipakai keliru.
- **E.** ✗ Jebakan: 3.600.000 ÷ 20.000; yang dibagi hanya biaya variabel.

Jawaban: C.

No. 10 Kunci: C

Konsep: Pemilihan model mengikuti kepastian kebutuhan: kebutuhan yang pasti dan tetap cocok dikerjakan berurutan (waterfall), sedangkan yang belum jelas dan perlu tanggapan pengguna cocok dikerjakan lewat putaran pendek (agile).

Langkah:
1. Pisahkan bagian proyek menurut kepastian: format surat dan penomoran (pasti), tampilan dan alur layar (belum pasti).
2. Bagian yang pasti dikerjakan terencana dan berurutan.
3. Bagian yang belum pasti dikerjakan dalam putaran pendek, diuji warga, lalu diperbaiki.

Analisis setiap opsi:
- **A.** ✗ Jebakan: waterfall untuk semua membuat tampilan baru terlihat di akhir, padahal perlu dicoba warga lebih dulu.
- **B.** ✗ Jebakan: agile bukan berarti tanpa pencatatan; aturan baku tetap harus dicatat dan dipatuhi.
- **C.** ✓ Menyesuaikan pendekatan dengan kepastian tiap bagian.
- **D.** ✗ Jebakan: menunda sampai segalanya pasti menghentikan proyek dan menghilangkan manfaat uji coba.
- **E.** ✗ Jebakan: tampilan yang tidak diuji pengguna berisiko tidak dipakai.

Jawaban: C.

No. 11 Kunci: A

Konsep: DevOps (atau administrator sistem) menjembatani pembuatan dan pengoperasian: menyiapkan server, mengemas dan merilis versi baru, serta memantau agar layanan tetap hidup.

Langkah:
1. Kata kunci: jalur otomatis, kode dikemas dan dipasang ke server, pemantauan server, pemberitahuan layanan mati.
2. Semuanya berpusat pada rilis dan pengoperasian server, bukan pada tampilan, kebutuhan klien, atau dokumentasi.

Analisis setiap opsi:
- **A.** ✓ Jalur rilis otomatis dan pemantauan server adalah tugas inti DevOps.
- **B.** ✗ Jebakan: front-end developer menulis kode tampilan; ia tidak mengelola server dan jalur rilis.
- **C.** ✗ Jebakan: UI/UX designer merancang layar dan menguji kemudahan pakai, bukan menyiapkan server.
- **D.** ✗ Jebakan: analis sistem bekerja di awal proyek menggali kebutuhan, jauh sebelum rilis.
- **E.** ✗ Jebakan: technical writer menulis dokumentasi; pekerjaannya tidak mencakup server dan rilis.

Jawaban: A.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 12 Kunci: D

Konsep: Harga jual = biaya pokok + laba. Bila laba dinyatakan dalam persen dari biaya, labanya = persen × biaya pokok.

Langkah:

1. Biaya pokok = 700.000 + 1.500.000 + 300.000 + 500.000 = Rp3.000.000.
2. Laba 30 % dari biaya = 0,30 × 3.000.000 = Rp900.000.
3. Harga jual = 3.000.000 + 900.000 = Rp3.900.000.
4. Cek: 900.000 ÷ 3.000.000 = 0,30 ✓.

Analisis setiap opsi:

- **A.** ✗ Jebakan: Rp4.285.714 adalah 3.000.000 ÷ 0,7, yaitu laba 30 % dari harga jual, bukan dari biaya.
- **B.** ✗ Jebakan: Rp3.600.000 menambah laba 20 %, bukan 30 %.
- **C.** ✗ Jebakan: Rp3.030.000 menambah laba 1 %; tanda persen terbaca sebagai 0,01.
- **D.** ✓ Biaya Rp3.000.000 ditambah laba Rp900.000.
- **E.** ✗ Jebakan: Rp3.300.000 menambah laba 10 %.

Jawaban: D.

No. 13 Kunci: C

Konsep: Pada jaringan kerja, kegiatan yang tidak saling menunggu berjalan paralel. Lama proyek ditentukan oleh **jalur terpanjang** (jalur kritis). Mempercepat kegiatan di luar jalur kritis tidak memajukan rilis.

Langkah:

1. Daftar jalur dari awal ke akhir: A–B–D–F = 3 + 4 + 5 + 2 = 14 hari; A–C–E–F = 3 + 2 + 3 + 2 = 10 hari.
2. Jalur kritis = A–B–D–F (14 hari). Rilis baru dapat maju bila kegiatan di jalur ini dipercepat.
3. E ada di jalur 10 hari, yang punya kelonggaran 4 hari. Mempercepat E 1 hari membuat jalur itu menjadi 9 hari, sedangkan proyek tetap 14 hari.
4. Pengujian F baru dapat mulai setelah D dan E selesai; D selesai paling akhir, jadi D yang ditunggu.

Analisis setiap opsi:

- **A.** ✗ Jebakan: E memang mendahului F, tetapi F menunggu yang paling lambat di antara D dan E, yaitu D.
- **B.** ✗ Jebakan: durasi proyek bukan jumlah semua kegiatan (19 hari) karena ada kegiatan paralel.
- **C.** ✓ Perbandingan dua jalur menunjukkan E tidak kritis.
- **D.** ✗ Jebakan: kesimpulan "tidak efektif" benar, tetapi alasannya keliru. Yang menentukan adalah jalur terpanjang, bukan satu kegiatan terpanjang.
- **E.** ✗ Jebakan: A memang kritis, tetapi pertanyaannya tentang E, dan memilih kegiatan "paling awal" bukan dasar menilai usulan.

Jawaban: C.

No. 14 Kunci: C

Konsep: Pada tiap tahap, profesinya ditentukan oleh tugas yang tertulis: menguji dan melaporkan bug adalah QA tester; mengemas versi dan memantau server adalah DevOps engineer.

Langkah:

1. Periksa tiap tahap: tugas dan profesinya cocok? Tahap 1 (menggali kebutuhan, analis sistem), tahap 2 (menggambar layar, UI/UX designer), tahap 3 (logika server, back-end developer) sudah tepat.
2. Tahap 4 bertugas memainkan versi uji dan melaporkan bug, tetapi profesinya tertulis DevOps; tahap 5 bertugas mengemas versi dan memantau server, tetapi tertulis QA tester.
3. Hanya dua profesi yang salah, dan keduanya tertukar. Perbaikannya: tukar profesi tahap 4 dan tahap 5.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

• **A.** ✗ Jebakan: tahap Analisis dan Rancangan sudah cocok dengan tugasnya.
• **B.** ✗ Jebakan: kedua tahap itu juga sudah tepat; menukarnya menimbulkan kesalahan baru.
• **C.** ✓ Pengujian seharusnya QA tester dan Rilis seharusnya DevOps engineer.
• **D.** ✗ Jebakan: menukar tahap 3 dan 4 memindahkan penguji ke pembuatan server, padahal tahap 3 sudah benar.
• **E.** ✗ Jebakan: tukaran ini menghasilkan analis sistem di rilis dan DevOps di analisis, jelas tidak cocok.

Jawaban: C.

No. 15 Kunci: 1B 2S 3B

Konsep: Titik impas (jumlah terjual agar rugi = laba = 0) = biaya tetap ÷ (harga jual − biaya variabel per unit). Laba = (harga − biaya variabel) × jumlah − biaya tetap.

Langkah:

1. Selisih per undangan = 75.000 − 15.000 = 60.000.
2. Titik impas = 1.200.000 ÷ 60.000 = 20 undangan → pernyataan 1 **Benar**.
3. Laba pada 30 undangan = 30 × 60.000 − 1.200.000 = 1.800.000 − 1.200.000 = Rp600.000. Jadi Rp1.800.000 **Salah**.
4. Biaya variabel baru 25.000 → selisih 50.000 → titik impas = 1.200.000 ÷ 50.000 = 24 undangan → pernyataan 3 **Benar**.

Analisis:

• **1.** ✓ Hitungan 1.200.000 ÷ 60.000 = 20 tepat.
• **2.** ✗ Jebakan: Rp1.800.000 adalah hasil 30 × 60.000 tanpa mengurangi biaya tetap; itu bukan laba.
• **3.** ✓ Kenaikan biaya variabel memperkecil selisih per undangan sehingga titik impas naik dari 20 menjadi 24.

Jawaban: Benar, Salah, Benar.

No. 16 Kunci: A

Konsep: Produk digital punya beberapa model pendapatan: jual putus atau lisensi, langganan, freemium, pembelian dalam aplikasi, iklan, dan jasa pembuatan.

Langkah:

1. Gim gratis diunduh, jadi bukan jual putus.
2. Uang masuk dari pembelian kostum dan koin di dalam gim.
3. Tidak ada pembayaran berkala maupun iklan.
4. Itulah pembelian dalam aplikasi.

Analisis setiap opsi:

• **A.** ✓ Kostum dan paket koin dibeli di dalam gim.
• **B.** ✗ Jebakan: langganan menuntut pembayaran tiap bulan; pemain di sini membayar hanya ketika membeli barang.
• **C.** ✗ Jebakan: gim dapat diunduh gratis, bukan dijual per salinan.
• **D.** ✗ Jebakan: soal menyebut studio tidak menampilkan iklan.
• **E.** ✗ Jebakan: jasa pembuatan melayani pesanan klien, bukan menjual barang kepada pemain.

Jawaban: A.

No. 17 Kunci: 1S 2B 3B

Konsep: Definisi selesai adalah daftar syarat yang harus terpenuhi seluruhnya sebelum sebuah pekerjaan disebut selesai. Kriteria penerimaan yang baik jelas dan terukur sehingga dapat diperiksa dengan jawaban ya atau tidak.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah:

1. Definisi selesai memuat tiga syarat; fitur yang baru memenuhi dua belum selesai.
2. "Cepat" dan "bagus" adalah penilaian subjektif; tidak ada angka atau bukti yang dapat diperiksa.
3. Menyebut batas waktu mengubahnya menjadi pernyataan terukur.

Analisis:

- **1.** ✗ Jebakan: tinjauan rekan juga bagian definisi selesai; satu syarat yang belum terpenuhi sudah cukup untuk menyatakan belum selesai.
- **2.** ✓ Kriteria yang baik terukur dan dapat dijawab ya atau tidak.
- **3.** ✓ "Paling lama dua detik" membuat kriteria dapat diuji.

Jawaban: Salah, Benar, Benar.

No. 18 Kunci: B, C, E

Konsep: Budaya mutu memperbaiki proses secara bersama: pedoman tertulis yang disepakati, pemeriksaan dalam tinjauan kode, dan alat bantu otomatis agar kepatuhan tidak bergantung pada ingatan.

Langkah: Nilai tiap pilihan: apakah memperbaiki proses yang melibatkan seluruh tim?

Analisis setiap opsi:

- **A.** ✗ Jebakan: kode yang berjalan tetapi sulit dibaca merugikan perawatan; mutu tidak hanya soal berjalan.
- **B.** ✓ Pedoman tertulis menyamakan gaya semua anggota.
- **C.** ✓ Tinjauan kode memastikan pedoman dipatuhi sebelum penggabungan.
- **D.** ✗ Jebakan: sanksi tidak memperbaiki proses dan membuat anggota enggan terbuka.
- **E.** ✓ Alat pemeriksa otomatis menjaga kepatuhan dengan konsisten.

Jawaban: B, C, dan E.

No. 19 Kunci: B

Konsep: Lama proyek ditentukan oleh jalur terpanjang. Mempercepat kegiatan pada jalur kritis hanya memperpendek proyek sampai jalur lain menjadi yang terpanjang.

Langkah:

1. Rencana awal: P–Q–S–U = 3 + 4 + 5 + 2 = 14 hari; P–R–T–U = 3 + 2 + 6 + 2 = 13 hari. Jalur kritis awal = 14 hari.
2. S dipercepat menjadi 2 hari: P–Q–S–U = 3 + 4 + 2 + 2 = 11 hari.
3. Jalur P–R–T–U tetap 13 hari, sekarang menjadi yang terpanjang.
4. Lama proyek paling cepat 13 hari (hanya berkurang 1 hari dari 14, bukan 3 hari).

Analisis setiap opsi:

- **A.** ✗ Jebakan: 11 hari hanya panjang jalur P–Q–S–U setelah dipercepat; jalur P–R–T–U masih 13 hari.
- **B.** ✓ Jalur P–R–T–U (13 hari) menjadi jalur terpanjang.
- **C.** ✗ Jebakan: 14 hari adalah lama sebelum S dipercepat.
- **D.** ✗ Jebakan: 12 hari tidak sesuai panjang jalur mana pun setelah perubahan.
- **E.** ✗ Jebakan: 10 hari tidak sesuai panjang jalur mana pun.

Jawaban: B.

No. 20 Kunci: A, C, D

Konsep: Pada scrum, product owner mengurutkan backlog menurut prioritas, dan tim memilih isi sprint sesuai kecepatan (poin yang biasanya selesai per sprint) dari urutan teratas.

Langkah:

1. Ambil dari prioritas tertinggi: login (8) + keranjang (13) = 21 poin, sama dengan kecepatan tim.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Menambah pembayaran (13) membuat 34 poin, jauh di atas kecepatan.
3. Urutan ditentukan product owner, bukan menurut ukuran poin.

Analisis setiap opsi:
• **A.** ✓ 8 + 13 = 21 poin, pas dengan kecepatan tim dan sesuai urutan prioritas.
• **B.** ✗ Jebakan: urutan ditentukan prioritas, bukan ukuran poin terkecil.
• **C.** ✓ 8 + 13 + 13 = 34 poin melebihi kecepatan 21.
• **D.** ✓ Mengurutkan prioritas backlog adalah tanggung jawab product owner.
• **E.** ✗ Jebakan: menambah isi sprint harus dibicarakan dengan product owner dan tidak otomatis wajib.

Jawaban: A, C, dan D.

No. 21 Kunci: A

Konsep: Pedoman tim dan definisi selesai berlaku untuk semua fungsi. Fungsi baru boleh digabung bila semua syarat terpenuhi; satu syarat yang belum terpenuhi cukup untuk menahannya.

Langkah:
1. Periksa tiap fungsi terhadap tiga syarat: nama deskriptif, ada keterangan, uji lolos.
2. hitung_gaji: nama deskriptif, ada keterangan, uji lolos → memenuhi semua.
3. cek: nama variabel x dan y tidak deskriptif → tidak memenuhi.
4. simpan_data: tanpa keterangan → tidak memenuhi.
5. kirim_slip: uji otomatis belum dibuat → tidak memenuhi.

Analisis setiap opsi:
• **A.** ✓ Hanya hitung_gaji yang memenuhi ketiga syarat.
• **B.** ✗ Jebakan: cek lolos uji, tetapi x dan y melanggar syarat nama deskriptif.
• **C.** ✗ Jebakan: simpan_data tidak memiliki keterangan fungsi.
• **D.** ✗ Jebakan: kirim_slip belum memiliki uji otomatis yang lolos.
• **E.** ✗ Jebakan: "sudah berjalan" tidak menggantikan pedoman tim.

Jawaban: A.

No. 22 Kunci: C

Konsep: Kriteria penerimaan dan definisi selesai tidak dilewati demi jadwal. Bila waktu mepet, tim menguji yang belum diuji dan membicarakan penyesuaian rilis bersama klien.

Langkah:
1. Uji layar kecil adalah kriteria yang sudah disepakati dan relevan bagi mayoritas pengguna.
2. Pilih tindakan yang menjalankan uji itu dan terbuka kepada klien.
3. Bila hasil gagal, bicarakan opsi: menunda fitur pembayaran atau seluruh rilis.

Analisis setiap opsi:
• **A.** ✗ Jebakan: pengguna menjadi penguji; kerusakan pada fitur pembayaran merugikan nyata.
• **B.** ✗ Jebakan: kriteria yang disepakati tidak boleh diturunkan sepihak demi jadwal.
• **C.** ✓ Menguji lebih dulu dan membicarakan penundaan bila perlu menjaga mutu dan kepercayaan klien.
• **D.** ✗ Jebakan: mengalihkan tanggung jawab uji kepada klien setelah rilis tidak profesional.
• **E.** ✗ Jebakan: mengurangi uji fitur lain menurunkan mutu di tempat lain.

Jawaban: C.

No. 23 Kunci: A, C, E

Konsep: Budaya mutu = mutu dibangun ke dalam proses kerja oleh seluruh tim: kriteria penerimaan yang jelas, pengujian, tinjauan rekan, dan perbaikan berkelanjutan (rencanakan–kerjakan–periksa–tindak lanjuti), bukan mengandalkan satu orang atau hukuman.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah: Nilai setiap pilihan: apakah memperbaiki **proses** agar masalah tercegah sejak awal dan melibatkan seluruh tim?

Analisis setiap opsi:

• **A.** ✓ Kriteria penerimaan yang memuat uji ukuran layar mencegah fitur lolos tanpa uji itu.
• **B.** ✗ Jebakan: pemeriksaan oleh satu orang menjelang rilis membuat mutu bergantung pada satu titik dan terlambat menemukan masalah.
• **C.** ✓ Mencatat penyebab, memperbarui pedoman, dan memeriksa hasilnya adalah siklus perbaikan berkelanjutan.
• **D.** ✗ Jebakan: menyalahkan individu membuat anggota menyembunyikan kesalahan; budaya mutu memperbaiki proses.
• **E.** ✓ Tinjauan kode oleh rekan menangkap kesalahan sebelum masuk cabang utama dan menyebarkan tanggung jawab mutu.

Jawaban: A, C, dan E.

No. 24 Kunci: B

Konsep: Kriteria penerimaan yang baik terukur: ada angka, batas, atau perilaku yang dapat diamati, sehingga penguji dapat menjawab ya atau tidak.

Langkah:

1. Periksa tiap pilihan: apakah ada ukuran yang bisa diperiksa?
2. Hanya satu pilihan yang menyebut perilaku jelas (kata kunci → daftar barang) dan batas waktu (dua detik).

Analisis setiap opsi:

• **A.** ✗ Jebakan: "berfungsi dengan baik" dan "terasa cepat" tidak terukur.
• **B.** ✓ Perilaku dan batas waktunya jelas; dapat diuji dengan jawaban ya atau tidak.
• **C.** ✗ Jebakan: kepuasan adalah perasaan, bukan ukuran yang dapat dibuktikan penguji.
• **D.** ✗ Jebakan: harapan semua pengguna pada setiap kesempatan tidak mungkin diperiksa.
• **E.** ✗ Jebakan: tampilan indah adalah penilaian selera dan bukan kriteria fitur pencarian.

Jawaban: B.

No. 25 Kunci: E

Konsep: Proses bisnis jasa pembuatan aplikasi mengalir: permintaan klien → penawaran dan kesepakatan (kontrak) → pengerjaan → pengujian → serah terima dan pembayaran → dukungan setelah rilis.

Langkah:

1. Pesanan dimulai dari permintaan klien.
2. Sebelum bekerja, studio dan klien menyepakati penawaran dan kontrak.
3. Pengerjaan lalu pengujian, kemudian produk diserahkan dan dibayar.
4. Dukungan layanan datang setelah produk dipakai.

Analisis setiap opsi:

• **A.** ✗ Jebakan: penawaran tidak mungkin mendahului permintaan klien.
• **B.** ✗ Jebakan: pengerjaan dimulai sebelum ada kesepakatan membuat pekerjaan tanpa kepastian ruang lingkup dan biaya.
• **C.** ✗ Jebakan: produk diuji sebelum dikerjakan.
• **D.** ✗ Jebakan: dukungan terjadi setelah serah terima, bukan sebelumnya.
• **E.** ✓ Urutan yang mengikuti alur nyata dari permintaan sampai dukungan.

Jawaban: E.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Latihan sub bab 2.1

No. 1 Kunci: B, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: K3LH di lingkungan TI meliputi ergonomi (posisi duduk dan monitor, istirahat mata), keselamatan listrik (tidak membebani stopkontak), kebersihan area perangkat, dan pengelolaan limbah elektronik/B3.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Stopkontak bertumpuk membebani satu titik listrik sehingga dapat panas dan memicu kebakaran.

• **Pernyataan 2** ✓ Posisi monitor sejajar mata mencegah leher terus menunduk (ergonomi).

• **Pernyataan 3** ✗ Jebakan: terasa nyaman, tetapi ventilasi laptop tertutup sehingga panas berlebih, dan posisi tubuh membungkuk tidak ergonomis.

• **Pernyataan 4** ✗ Jebakan: tampak memanfaatkan ulang, tetapi baterai bekas termasuk limbah B3 yang harus dipisah dan diserahkan ke pengelola berizin, bukan pengepul biasa.

• **Pernyataan 5** ✓ Istirahat mata berkala mengurangi kelelahan mata akibat menatap layar terlalu lama.

Jawaban: pernyataan 2, 5.

No. 2 Kunci: E

Konsep: Limbah elektronik (e-waste) dan baterai/aki bekas mengandung bahan berbahaya (B3) seperti timbal dan asam. Limbah ini dipisahkan menurut jenisnya dan diserahkan ke pengelola berizin, tidak dibuang, dibakar, atau ditimbun sembarangan.

Langkah:

1. Monitor dan baterai UPS termasuk limbah elektronik/B3.
2. Penanganan aman: pisahkan jenisnya, simpan sementara dengan aman, serahkan ke pengelola berizin.

Analisis setiap opsi:

• **A.** ✗ Limbah B3 tidak boleh dibuang bersama sampah umum karena mencemari tanah dan air.

• **B.** ✗ Baterai yang soak dapat bocor atau panas; menyimpannya di area kerja membahayakan pegawai, dan angkutan sampah biasa tidak boleh menerima limbah B3.

• **C.** ✗ Jebakan: monitor sudah ditangani benar, tetapi baterai UPS juga limbah B3 dan tidak boleh dibuang ke TPS.

• **D.** ✗ Jebakan: barang memang keluar dari kantor, tetapi pengolahan tanpa izin berisiko mencemari lingkungan.

• **E.** ✓ Pengelola berizin mengolah limbah elektronik dengan aman; baterai dipisah karena mengandung asam dan timbal.

Jawaban: E.

No. 3 Kunci: B

Konsep: Kode sumber adalah ciptaan yang dilindungi hak cipta. Bila kontrak menyerahkan kepemilikannya kepada klien, programmer tidak boleh memakai ulang kode itu tanpa izin pemiliknya; yang boleh dibawa adalah pengetahuan dan keterampilannya.

Langkah:

1. Menurut kontrak, pemilik kode adalah Toko Maju.
2. Memakai kode itu untuk klien lain tanpa izin melanggar kontrak dan hak cipta.
3. Pilihan yang etis: menulis kode baru dengan pengetahuan yang dimiliki.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

• **A.** ✗ Mengganti nama variabel dan warna tidak mengubah fakta bahwa kodenya disalin dari milik Toko Maju.

• **B.** ✓ Pengetahuan Rina tetap boleh dipakai; yang tidak boleh adalah menyalin kode milik klien lama.

• **C.** ✗ Jebakan: Rina memang penulisnya, tetapi kontrak sudah menyerahkan kepemilikan kode kepada Toko Maju.

• **D.** ✗ Jebakan: izin harus diminta sebelum kode dipakai; meminta izin setelahnya berarti sudah melanggar lebih dulu.

• **E.** ✗ Menjual kode milik orang lain jelas melanggar kontrak dan hak cipta.

Jawaban: B.

No. 4 Kunci: B, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Etika kerja profesional di industri perangkat lunak meliputi menjaga kerahasiaan data pengguna, jujur dan bertanggung jawab atas kesalahan, serta berkoordinasi dengan anggota tim.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: niatnya rajin, tetapi data pribadi siswa keluar dari sistem yang aman dan mudah hilang atau bocor.

• **Pernyataan 2** ✓ Data uji yang disamarkan melindungi data pribadi siswa tanpa menghambat pengujian.

• **Pernyataan 3** ✗ Jebakan: bertanya di forum boleh, tetapi berkas konfigurasi memuat alamat dan kata sandi basis data sehingga data siswa terancam.

• **Pernyataan 4** ✗ Jebakan: tampak membantu, tetapi perubahan tanpa koordinasi dapat bertabrakan dengan pekerjaan Galih dan sulit ditelusuri.

• **Pernyataan 5** ✓ Kejujuran dan pelaporan cepat memungkinkan kesalahan diperbaiki sebelum merugikan siswa.

Jawaban: pernyataan 2, 5.

No. 5 Kunci: C

Konsep: Etika kerja profesional menuntut kejujuran dan tanggung jawab: kesalahan dilaporkan segera beserta dampaknya agar kerugian klien dapat dihentikan, lalu diperbaiki bersama tim.

Langkah:

1. Kesalahan sedang merugikan klien (stok salah, toko masih buka).
2. Ketua tim masih dapat dihubungi → laporkan sekarang dan bantu perbaikan.

Analisis setiap opsi:

• **A.** ✗ Toko tetap memakai stok yang salah malam ini dan sepanjang akhir pekan, dan tim tidak tahu ada masalah.

• **B.** ✗ Jebakan: tampak cepat dan bertanggung jawab, tetapi perbaikan tergesa-gesa tanpa peninjauan dapat menambah kesalahan baru, dan ketua tim tidak tahu.

• **C.** ✓ Laporan cepat dan jujur menghentikan kerugian dan memungkinkan perbaikan terkoordinasi.

• **D.** ✗ Jebakan: kesalahan berikutnya mungkin berhenti, tetapi data stok yang sudah salah tidak dibetulkan dan tim tidak tahu apa yang terjadi.

• **E.** ✗ Jebakan: kerugian stok memang berhenti, tetapi mematikan layanan klien tanpa pemberitahuan menimbulkan kerugian baru dan melanggar kepercayaan.

Jawaban: C.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Latihan sub bab 2.2

No. 1 Kunci: C

Konsep: Aset digital (musik, gambar, font, kode) memiliki lisensi. Lisensi BY-NC mewajibkan pencantuman nama pencipta (BY) dan melarang penggunaan komersial (NC). Produk yang dijual harus memakai aset berlisensi komersial atau mendapat izin khusus dari pencipta.

Langkah:

1. Syarat lisensi: (1) cantumkan nama pencipta, (2) tidak untuk tujuan komersial.
2. Gim dijual Rp25.000 → penggunaan komersial → syarat (2) dilanggar.
3. Jalan keluar: izin komersial dari pencipta atau ganti musik yang lisensinya membolehkan.

Analisis setiap opsi:

- **A.** ✗ Jebakan: syarat pencantuman nama terpenuhi, tetapi syarat "tidak komersial" tetap dilanggar karena gim dijual.
- **B.** ✗ Musik yang diubah tetap turunan karya pencipta dan tetap terikat lisensi NC.
- **C.** ✓ Izin komersial atau musik pengganti yang sah memenuhi hak pencipta.
- **D.** ✗ Gratis diunduh tidak sama dengan bebas dipakai untuk produk yang dijual.
- **E.** ✗ Keterangan sumber tidak mengubah larangan penggunaan komersial dalam lisensi NC.

Jawaban: C.

No. 2 Kunci: A, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pengelolaan aset fisik meliputi pencatatan peminjaman dan batas kembali, pemeriksaan kondisi, penarikan perangkat yang membahayakan, serta pencatatan kerusakan di kartu aset.

Analisis setiap pernyataan:

- **Pernyataan 1** ✓ Batas kembali 8 Maret sudah lewat pada 10 Maret.
- **Pernyataan 2** ✗ Jebakan: ponsel memang menyala, tetapi baterai menggembung berisiko terbakar, apalagi saat diisi daya.
- **Pernyataan 3** ✗ Batas kembali HP-03 adalah 16 Maret, belum terlewati.
- **Pernyataan 4** ✓ Baterai menggembung harus dikeluarkan dari pemakaian dan dibuang lewat pengelola limbah B3.
- **Pernyataan 5** ✓ Pencatatan kondisi membantu memantau kerusakan dan merencanakan perbaikan.

Jawaban: pernyataan 1, 4, 5.

Latihan sub bab 2.3

No. 1 Kunci: 1B 2B 3S

Konsep: Pengelolaan berkas proyek memakai aturan penamaan yang konsisten (nama_proyek, nomor versi, tanggal), struktur folder yang jelas, dan arsip versi lama, sehingga versi terbaru mudah ditemukan dan riwayat tetap tersimpan.

Analisis setiap pernyataan:

- **Pernyataan 1 → Benar.** Kata "final", "revisi", dan "BENAR" tidak menunjukkan urutan; tidak jelas mana yang terbaru.
- **Pernyataan 2 → Benar.** Nomor versi dan tanggal membuat berkas terurut dan versi terbaru mudah dikenali.
- **Pernyataan 3 → Salah.** Jebakan: folder memang lebih bersih, tetapi riwayat hilang. Versi lama sebaiknya dipindah ke folder arsip agar dapat dibuka bila klien meminta kembali ke rancangan sebelumnya.

Jawaban: Benar, Benar, Salah.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 2 Kunci: A

Konsep: Konflik terjadi bila dua cabang mengubah bagian berkas yang sama. Cara yang benar: gabungkan perubahan terbaru main ke cabang fitur, selesaikan konflik dengan memilih/menggabungkan isi yang benar, uji ulang, lalu perbarui pull request untuk ditinjau.

Langkah:

1. Konflik muncul karena main sudah berubah sejak cabang fitur-skor dibuat.
2. Konflik diselesaikan di cabang fitur, bukan di main, agar main tetap siap rilis.
3. Setelah diuji, pull request diperbarui dan ditinjau rekan seperti biasa.

Analisis setiap opsi:

- **A.** ✓ Konflik diselesaikan di cabang fitur, perubahan rekan tetap ada, dan main tetap aman.
- **B.** ✗ Jebakan: push --force ke main menimpa tiga commit rekan dan melewati peninjauan.
- **C.** ✗ Berkas config.py dibutuhkan; menghapusnya merusak program dan tidak menyelesaikan konflik.
- **D.** ✗ Menimpa folder secara manual menghapus perubahan rekan dan melewati peninjauan.
- **E.** ✗ Jebakan: konflik memang hilang, tetapi pekerjaan rekan dibuang dan main mundur dari versi terbarunya.

Jawaban: A.

No. 3 Kunci: E

Konsep: Prinsip cadangan 3-2-1: minimal 3 salinan data, di 2 jenis media, dan 1 salinan di lokasi lain. Salinan di lokasi lain melindungi dari bencana di satu tempat seperti kebakaran atau pencurian.

Langkah:

1. Risiko yang dikhawatirkan: kebakaran dan pencurian → mengenai seluruh ruangan sekaligus.
2. Saat ini komputer dan hard disk cadangan berada di tempat yang sama → keduanya bisa hilang bersamaan.
3. Perbaikan: tambah salinan di lokasi berbeda (cloud atau kantor lain).

Analisis setiap opsi:

- **A.** ✗ Dua hard disk di laci yang sama tetap hilang bersama bila terjadi kebakaran atau pencurian.
- **B.** ✗ Mirror melindungi dari kerusakan satu hard disk, tetapi keduanya berada di komputer yang sama.
- **C.** ✗ Jebakan: cadangan lebih sering memang baik, tetapi lokasinya tetap satu ruangan dengan komputer.
- **D.** ✗ Memindahkan berkas tidak menambah salinan; data justru hanya ada di satu tempat.
- **E.** ✓ Salinan di lokasi lain tetap aman walaupun ruangan studio terbakar atau dibobol.

Jawaban: E.

Kuis Akhir Bab 2

No. 1 Kunci: B

Konsep: Rambu K3 dikenali dari warna dan bentuk: larangan (lingkaran merah dicoret), wajib (lingkaran biru), peringatan (segitiga kuning), dan kondisi aman atau informasi (persegi hijau, misalnya jalur evakuasi, titik kumpul, P3K).

Langkah:

1. Titik kumpul termasuk informasi tentang kondisi aman, jadi rambunya berupa persegi hijau.
2. Pada gambar, persegi hijau dengan gambar orang yang bergerak menuju panah adalah rambu Q.

Analisis setiap opsi:

- **A.** ✗ Jebakan: segitiga kuning adalah peringatan bahaya, bukan penanda tempat aman.
- **B.** ✓ Persegi hijau dengan orang dan panah menandai arah ke titik kumpul.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **C.** ✗ Jebakan: lingkaran biru berarti wajib (di sini wajib memakai helm), bukan titik kumpul.
• **D.** ✗ Jebakan: lingkaran merah dengan coretan berarti larangan (di sini dilarang membawa minuman).
• **E.** ✗ Salah: rambu Q sudah sesuai.

Jawaban: B.

No. 2 Kunci: A, C, E

Konsep: Ergonomi menata cara dan alat kerja agar sesuai dengan tubuh. Postur yang dianjurkan: layar setinggi mata, jarak sekitar 50–70 cm, punggung bersandar, siku sekitar 90°, telapak kaki di lantai, dan istirahat mata dengan aturan 20-20-20.

Langkah: Nilai tiap saran terhadap penyebab keluhan: leher (tinggi layar dan postur) dan mata (jarak pandang yang terus sama).

Analisis setiap opsi:
• **A.** ✓ Layar setinggi mata dan berjarak cukup mengurangi beban leher dan mata.
• **B.** ✗ Jebakan: layar yang rendah memaksa leher menunduk lama dan justru menimbulkan nyeri.
• **C.** ✓ Aturan 20-20-20 mengistirahatkan otot mata.
• **D.** ✗ Jebakan: membungkuk membebani tulang belakang; punggung seharusnya bersandar pada sandaran kursi.
• **E.** ✓ Siku sekitar 90° dan kaki menapak mengurangi tegangan bahu, lengan, dan pinggang.

Jawaban: A, C, dan E.

No. 3 Kunci: 1S 2B 3B

Konsep: Hak atas kode hasil pekerjaan ditentukan oleh perjanjian kerja. Bila kontrak menyebut kode milik klien, kode itu tidak boleh disalin untuk klien lain tanpa izin.

Langkah:
1. Kontrak klien A menetapkan kode menjadi milik klien A, bukan milik penulisnya.
2. Jalan yang benar: menulis kode baru atau memperoleh izin tertulis.
3. Memberi tahu ketua tim membuat keputusan terbuka dan dapat dipertanggungjawabkan.

Analisis:
• **1.** ✗ Jebakan: yang menulis kode belum tentu pemilik haknya.
• **2.** ✓ Dua jalan yang sah dan jujur.
• **3.** ✓ Terbuka kepada tim adalah kebiasaan profesional.

Jawaban: Salah, Benar, Benar.

No. 4 Kunci: A

Konsep: Aset fisik berwujud (perangkat, perabot), sedangkan aset digital tidak berwujud tetapi bernilai: lisensi, data, kode sumber, akun, dan nama domain.

Langkah: Tanyakan: dapatkah benda itu disentuh? Yang tidak dapat disentuh tetapi bernilai adalah aset digital.

Analisis setiap opsi:
• **A.** ✓ Lisensi adalah hak pakai yang bernilai dan tidak berwujud.
• **B.** ✗ Jebakan: meja kerja berwujud, aset fisik.
• **C.** ✗ Jebakan: monitor berwujud, aset fisik.
• **D.** ✗ Jebakan: router adalah perangkat keras, aset fisik.
• **E.** ✗ Jebakan: proyektor adalah perangkat keras, aset fisik.

Jawaban: A.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 5 Kunci: E

Konsep: Profesional bekerja dalam kewenangannya: tidak memakai akun orang lain, tidak membuat akses sendiri, dan mengikuti prosedur permintaan akses ketika darurat. Hak akses diberikan menurut tugas, dan semua tindakan harus dapat ditelusuri.

Langkah:

1. Eko tidak berwenang atas server, jadi ia perlu akses resmi.
2. Pada keadaan darurat, jalurnya: lapor ketua tim, lalu pihak berwenang memberi akses sementara.

Analisis setiap opsi:

- **A.** ✗ Jebakan: memakai akun orang lain membuat jejak tindakan salah nama dan melanggar prinsip akses.
- **B.** ✗ Jebakan: membagikan kata sandi lewat pesan menyebarkan rahasia.
- **C.** ✗ Jebakan: membuat akun tanpa persetujuan melewati kontrol keamanan.
- **D.** ✗ Jebakan: diam tidak menolong klien dan tidak melaporkan gangguan layanan.
- **E.** ✓ Menjaga kewenangan, tetap menolong klien, dan terbuka kepada tim.

Jawaban: E.

No. 6 Kunci: C

Konsep: Penyusutan garis lurus per tahun = (harga perolehan − nilai sisa) ÷ umur manfaat. Nilai buku = harga perolehan − akumulasi penyusutan.

Langkah:

1. Laptop: (15.000.000 − 3.000.000) ÷ 4 = 3.000.000 per tahun; akumulasi 2 tahun = 6.000.000; nilai buku = 15.000.000 − 6.000.000 = 9.000.000.
2. Server: (20.000.000 − 2.000.000) ÷ 6 = 3.000.000 per tahun; akumulasi 3 tahun = 9.000.000; nilai buku = 20.000.000 − 9.000.000 = 11.000.000.
3. Jumlah nilai buku = 9.000.000 + 11.000.000 = Rp20.000.000.

Analisis setiap opsi:

- **A.** ✗ Jebakan: Rp15.000.000 adalah jumlah akumulasi penyusutan (6.000.000 + 9.000.000), bukan nilai buku.
- **B.** ✗ Jebakan: Rp17.500.000 muncul bila nilai sisa diabaikan (laptop 7.500.000 dan server 10.000.000).
- **C.** ✓ Nilai buku laptop dan server dijumlahkan.
- **D.** ✗ Jebakan: Rp35.000.000 adalah jumlah harga perolehan.
- **E.** ✗ Jebakan: Rp5.000.000 adalah jumlah nilai sisa.

Jawaban: C.

No. 7 Kunci: A, B, D

Konsep: Budaya kerja profesional: jujur, komunikasi terbuka, terbuka terhadap tinjauan dan kritik, tepat waktu dengan perkiraan yang jujur, dan tidak memakai sumber daya kantor untuk kepentingan pribadi tanpa izin.

Langkah: Nilai tiap kebiasaan: apakah jujur, terbuka, dan menjaga kepercayaan?

Analisis setiap opsi:

- **A.** ✓ Hambatan yang disampaikan lebih awal masih dapat diatur ulang oleh tim.
- **B.** ✓ Tinjauan dilakukan untuk menyempurnakan produk, bukan menilai pribadi.
- **C.** ✗ Jebakan: menyembunyikan kesalahan memperbesar kerugian bila akhirnya terungkap.
- **D.** ✓ Perkiraan yang jujur lebih berharga daripada janji yang tidak dapat ditepati.
- **E.** ✗ Jebakan: lisensi kantor tidak boleh dipakai untuk kepentingan pribadi tanpa izin.

Jawaban: A, B, dan D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 8 Kunci: B, C, E

Konsep: Pada insiden listrik, keselamatan penolong dan korban diutamakan: jangan menyentuh korban yang masih menempel pada sumber listrik sebelum aliran listrik diputus, putus sumber listrik bila dapat dijangkau dengan aman, minta bantuan, lalu beri pertolongan pertama dan laporkan kejadian walau lukanya ringan.

Langkah: Urutkan prioritas: putus listrik → minta bantuan → periksa dan tolong korban → laporkan. Tolak tindakan yang menambah bahaya.

Analisis setiap opsi:

- **A.** ✗ Jebakan: menyentuh korban yang masih menempel pada sumber listrik membuat penolong ikut tersengat.
- **B.** ✓ Memutus aliran listrik menghilangkan sumber bahaya sebelum korban disentuh.
- **C.** ✓ Meminta bantuan dan melapor kepada petugas penanggung jawab.
- **D.** ✗ Jebakan: air menghantarkan listrik dan menambah risiko tersengat.
- **E.** ✓ Pertolongan pertama dan laporan tetap dilakukan walau lukanya ringan.

Jawaban: B, C, dan E.

No. 9 Kunci: 1B 2S 3S

Konsep: Lisensi adalah izin pakai dengan syarat tertentu. Syarat itu harus dipenuhi: atribusi pada CC BY, pemberitahuan hak cipta pada MIT, dan batas jumlah perangkat pada lisensi berbayar.

Langkah: Cocokkan tiap pernyataan dengan syarat lisensinya.

Analisis:

- **1.** ✓ MIT sangat longgar: boleh dipakai untuk produk berbayar dengan mempertahankan pemberitahuan hak cipta.
- **2.** ✗ Jebakan: BY artinya pembuat wajib disebut.
- **3.** ✗ Jebakan: lisensi satu perangkat tidak boleh dipasang pada banyak komputer.

Jawaban: Benar, Salah, Salah.

No. 10 Kunci: D

Konsep: Hierarki pengendalian risiko dari yang paling efektif: eliminasi, substitusi, rekayasa teknik, administratif, dan alat pelindung diri (APD).

Langkah:

1. Q memindahkan meja sehingga sumber bahaya (cahaya langsung) tidak lagi mengenai layar: eliminasi.
2. P memasang tirai, yaitu perlindungan fisik pada sumber cahaya: rekayasa teknik.
3. R menetapkan jadwal dan pengumuman: administratif.
4. S kacamata penyaring: APD, pilihan terakhir karena bahayanya tetap ada.
5. Urutan: Q – P – R – S.

Analisis setiap opsi:

- **A.** ✗ Jebakan: tirai (rekayasa) ditempatkan lebih tinggi daripada pemindahan meja (eliminasi).
- **B.** ✗ Jebakan: jadwal istirahat (administratif) didahulukan dari tirai (rekayasa).
- **C.** ✗ Jebakan: urutannya terbalik; APD bukan yang paling efektif.
- **D.** ✓ Eliminasi, rekayasa teknik, administratif, lalu APD.
- **E.** ✗ Jebakan: jadwal istirahat (administratif) ditempatkan paling efektif, padahal ia tidak menghilangkan bahaya.

Jawaban: D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 11 Kunci: B, C, E

Konsep: Pengelolaan aset digital menerapkan hak akses minimum (hanya sebesar yang diperlukan), peninjauan dan pencabutan akses secara berkala, serta perlindungan akun yang kuat, misalnya verifikasi dua langkah.

Langkah: Nilai tiap langkah: apakah mengurangi akses yang tidak perlu dan melindungi akun?

Analisis setiap opsi:

- **A.** ✗ Jebakan: akses administrator berlebihan bagi magang menambah risiko salah hapus dan penyalahgunaan.
- **B.** ✓ Akses sesuai tugas adalah prinsip hak akses minimum.
- **C.** ✓ Akses orang yang sudah keluar harus dicabut agar tidak disalahgunakan.
- **D.** ✗ Jebakan: satu sandi bersama membuat semua akun jatuh bila satu bocor.
- **E.** ✓ Verifikasi dua langkah menambah lapisan perlindungan akun.

Jawaban: B, C, dan E.

No. 12 Kunci: B

Konsep: Creative Commons: BY = wajib menyebut pembuat; NC = tidak untuk tujuan komersial; ND = tanpa turunan (tidak boleh diubah); SA = bagi serupa (karya turunan dibagikan dengan lisensi yang sama). Syarat-syarat itu berlaku bersamaan.

Langkah:

1. BY: nama pembuat asli wajib dicantumkan.
2. SA: ikon hasil ubahan (karya turunan) boleh dibuat, tetapi harus dibagikan dengan lisensi yang sama (BY-SA).
3. Tidak ada syarat NC atau ND pada lisensi ini, sehingga perubahan dan pemakaian dalam aplikasi diperbolehkan.

Analisis setiap opsi:

- **A.** ✗ Jebakan: syarat SA mewajibkan hasil ubahan dibagikan dengan lisensi yang sama, bukan bebas.
- **B.** ✓ Atribusi (BY) dan bagi serupa (SA).
- **C.** ✗ Jebakan: yang melarang perubahan adalah ND, bukan SA.
- **D.** ✗ Jebakan: gratis atau berbayar tidak menghapus syarat lisensi.
- **E.** ✗ Jebakan: SA bukan berarti berbayar; lisensi CC memberi izin tanpa izin tertulis tambahan selama syaratnya dipenuhi.

Jawaban: B.

No. 13 Kunci: C

Konsep: Pekerja profesional menjaga kerahasiaan data pelanggan dan klien. NDA melarang membocorkan informasi proyek, dan data pribadi tidak boleh dipindahkan ke pihak lain tanpa izin.

Langkah:

1. Data pasien asli adalah informasi rahasia yang tercakup NDA.
2. Alasan apa pun (teman, sebagian kecil, janji menghapus, layanan aman) tidak menghapus larangan itu.

Analisis setiap opsi:

- **A.** ✗ Jebakan: kepercayaan pada teman tidak menggantikan izin klien.
- **B.** ✗ Jebakan: memindahkan data ke layanan pribadi tetap membocorkan data.
- **C.** ✓ Menolak adalah sikap profesional yang menjaga kerahasiaan.
- **D.** ✗ Jebakan: sebagian kecil data asli tetap termasuk data rahasia.
- **E.** ✗ Jebakan: janji menghapus tidak dapat dipastikan dan pelanggaran sudah terjadi.

Jawaban: C.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 14 Kunci: E

Konsep: Nama berkas yang baik deskriptif, konsisten (huruf kecil, tanpa spasi, dipisah garis bawah), tidak memakai kata seperti "final", dan bila tanpa Git memuat nomor versi serta tanggal agar mudah diurutkan.

Langkah: Periksa tiap nama: deskriptif? konsisten? ada versi dan tanggal?

Analisis setiap opsi:

- **A.** ✗ Jebakan: memakai spasi dan "final", tidak jelas mana yang terbaru.
- **B.** ✗ Jebakan: tanda kurung dan "baru sekali" tidak menunjukkan versi.
- **C.** ✗ Jebakan: tidak deskriptif.
- **D.** ✗ Jebakan: memakai "FINAL BENAR"; kata semacam itu cepat tidak berlaku bila ada versi lagi.
- **E.** ✓ Deskriptif, huruf kecil, tanpa spasi, dengan nomor versi dan tanggal.

Jawaban: E.

No. 15 Kunci: A

Konsep: Baterai bekas termasuk limbah elektronik berbahaya: masih menyimpan bahan kimia dan dapat bocor atau terbakar. Perangkat bekas disimpan di tempat kering, terpisah dari bahan mudah terbakar, lalu diserahkan kepada pengelola yang berizin.

Langkah:

1. Hindari semua pilihan yang melepaskan bahan berbahaya ke lingkungan: dibuang, dibakar, dibongkar, atau dikubur.
2. Pilih jalur resmi: simpan dengan aman, lalu serahkan ke pengelola berizin.

Analisis setiap opsi:

- **A.** ✓ Jalur yang benar untuk limbah elektronik.
- **B.** ✗ Jebakan: bahan kimia baterai mencemari tempat pembuangan akhir.
- **C.** ✗ Jebakan: pembakaran melepaskan gas beracun dan dapat memicu ledakan.
- **D.** ✗ Jebakan: membongkar sel baterai berisiko tersengat dan terpapar bahan kimia.
- **E.** ✗ Jebakan: menguburnya mencemari tanah dan air tanah.

Jawaban: A.

No. 16 Kunci: D

Konsep: Penghapusan aset secara aman: cadangkan yang masih diperlukan, cabut lisensi dan akun yang terhubung, hapus data dengan cara yang tidak dapat dipulihkan, lalu perbarui inventaris.

Langkah:

1. Selamatkan data yang masih diperlukan.
2. Cabut akun dan lisensi agar tidak dipakai pihak lain.
3. Hapus data dengan aman sebelum perangkat dilepas.

Analisis setiap opsi:

- **A.** ✗ Jebakan: berkas yang dikosongkan dari tempat sampah masih dapat dipulihkan.
- **B.** ✗ Jebakan: format cepat tidak menghapus data secara tuntas.
- **C.** ✗ Jebakan: data proyek klien termasuk rahasia dan tidak boleh tertinggal.
- **D.** ✓ Urutan penghapusan aman yang lengkap.
- **E.** ✗ Jebakan: menyerahkan data kepada pembeli adalah kebocoran.

Jawaban: D.

No. 17 Kunci: A, B, D

Konsep: Inventaris mencatat setiap aset dengan kode yang unik, kondisi, lokasi, dan penanggung jawab. Aset digital seperti lisensi juga dicatat, termasuk masa berlakunya. Penghapusan aset mengikuti prosedur resmi.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah: Periksa tiap baris tabel terhadap prinsip inventaris: kode unik, penanggung jawab jelas, semua jenis aset tercatat, penghapusan sesuai prosedur.

Analisis setiap opsi:

- **A.** ✓ Kode seharusnya unik; dua proyektor tidak boleh berkode sama.
- **B.** ✓ LAP-02 tidak punya penanggung jawab; tanpa itu perbaikannya tidak jelas.
- **C.** ✗ Jebakan: penghapusan aset memerlukan keputusan dan persetujuan sesuai prosedur.
- **D.** ✓ Lisensi adalah aset digital; masa berlakunya perlu dipantau agar tidak lapse.
- **E.** ✗ Jebakan: aset digital juga bernilai dan wajib dicatat.

Jawaban: A, B, dan D.

No. 18 Kunci: B

Konsep: Data produksi yang berisi data pribadi tidak dimasukkan ke repositori. Pengujian memakai data contoh atau salinan yang disamarkan. Riwayat Git menyimpan isi berkas yang pernah dicommit, sehingga menghapusnya lewat commit baru tidak cukup.

Langkah:

1. Tolak penggabungan agar data pribadi tidak masuk ke main.
2. Bersihkan berkas dari cabang dan riwayatnya.
3. Sediakan data contoh yang disamarkan untuk pengujian.

Analisis setiap opsi:

- **A.** ✗ Jebakan: repositori privat tetap dapat diakses banyak orang dan salinannya menyebar ke setiap komputer anggota.
- **B.** ✓ Menjaga data pribadi dan tetap memenuhi kebutuhan pengujian.
- **C.** ✗ Jebakan: commit penghapusan tidak menghapus isi berkas dari riwayat.
- **D.** ✗ Jebakan: .gitignore tidak berlaku untuk berkas yang sudah di-commit.
- **E.** ✗ Jebakan: mengganti nama tidak mengubah isi data pribadinya.

Jawaban: B.

No. 19 Kunci: E

Konsep: Alur kerja Git yang sehat: buat cabang dari main, kerjakan dengan commit berkala, tarik perubahan terbaru dari main dan uji, ajukan pull request, ditinjau rekan, lalu merge ke main.

Langkah:

1. Cabang fitur dibuat lebih dulu (1).
2. Fitur dikerjakan dengan commit kecil (3).
3. Sebelum meminta tinjauan, tarik main terbaru dan uji (5) agar konflik kecil terselesaikan dan fitur tetap lolos uji.
4. Ajukan pull request (2), lalu ditinjau (6), lalu merge ke main (4).
5. Urutan: 1 – 3 – 5 – 2 – 6 – 4.

Analisis setiap opsi:

- **A.** ✗ Jebakan: mengerjakan fitur (3) sebelum membuat cabangnya (1).
- **B.** ✗ Jebakan: pull request diajukan (2) sebelum menarik main terbaru dan menguji (5).
- **C.** ✗ Jebakan: tinjauan (6) dilakukan sebelum pull request diajukan (2).
- **D.** ✗ Jebakan: menarik main (5) sebelum fitur dikerjakan, dan merge (4) sebelum ditinjau (6).
- **E.** ✓ Urutan yang sesuai alur.

Jawaban: E.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 20 Kunci: D

Konsep: Aturan 3-2-1: tiga salinan (data asli ditambah dua cadangan), dua jenis media berbeda, dan satu salinan di lokasi lain. Sinkronisasi otomatis bukan cadangan sungguhan: berkas yang terhapus di satu tempat ikut terhapus di tempat lain.

Langkah:
1. Jumlah salinan: hanya dua (data asli dan satu salinan awan) ✗; aturan meminta tiga.
2. Salinan awan berada di luar studio, tetapi sifatnya sinkronisasi, sehingga berkas yang terhapus atau rusak ikut tersinkron ✗.
3. Perbaikan: tambahkan satu cadangan sungguhan pada media lain, misalnya disk eksternal yang diperbarui terjadwal, selain salinan awan.

Analisis setiap opsi:
- **A.** ✗ Jebakan: jenis media dan lokasi tampak terpenuhi, tetapi jumlah baru dua dan sinkronisasi bukan cadangan sungguhan.
- **B.** ✗ Jebakan: penyimpanan awan di luar studio termasuk lokasi lain.
- **C.** ✗ Jebakan: menambah disk di studio tidak memperbaiki kekurangan jumlah salinan dan kurang lokasi lain.
- **D.** ✓ Dua salinan kurang dari tiga, dan sinkronisasi tidak melindungi dari berkas yang terhapus.
- **E.** ✗ Jebakan: tiga salinan justru dianjurkan; satu salinan terlalu berisiko.

Jawaban: D.

No. 21 Kunci: A, B, D

Konsep: README adalah berkas pertama yang dibaca orang: memuat nama dan tujuan proyek, cara memasang, cara menjalankan, cara mengisi pengaturan (tanpa rahasia sungguhan), cara menjalankan uji, dan nama pengelola.

Langkah: Nilai tiap isi: membantu anggota baru memulai dan tidak membocorkan rahasia?

Analisis setiap opsi:
- **A.** ✓ Identitas proyek dan kontak pengelola membantu orang baru.
- **B.** ✓ Cara memasang dan menjalankan adalah fungsi utama README.
- **C.** ✗ Jebakan: kunci asli tidak boleh dituliskan di berkas apa pun dalam repositori.
- **D.** ✓ Contoh nilai palsu menunjukkan format tanpa membocorkan rahasia, dan cara uji memastikan pemasangan benar.
- **E.** ✗ Jebakan: menyalin kode ke README menggandakan isi repositori dan cepat usang.

Jawaban: A, B, dan D.

No. 22 Kunci: 1S 2B 3B

Konsep: Aturan 3-2-1: tiga salinan, dua jenis media, satu salinan di lokasi lain. Cadangan yang belum pernah diuji pulih belum tentu dapat dipakai.

Langkah: Nilai tiap pernyataan terhadap tujuan cadangan: dapat memulihkan data saat perangkat atau tempat terkena musibah.

Analisis:
- **1.** ✗ Jebakan: bila komputer rusak, terbakar, atau dicuri, cadangan di tempat yang sama ikut hilang.
- **2.** ✓ Uji pemulihan membuktikan cadangan benar-benar dapat dipakai.
- **3.** ✓ Lokasi lain melindungi dari kebakaran atau banjir di satu gedung.

Jawaban: Salah, Benar, Benar.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 23 Kunci: A, C, D

Konsep: .gitignore memuat berkas yang tidak boleh dilacak Git: rahasia, hasil kompilasi sementara, dan data pribadi. Yang dilacak adalah kode, dokumentasi, dan skrip yang diperlukan untuk membangun ulang proyek.

Langkah: Tanyakan pada tiap berkas: rahasia? dapat dibuat ulang? data pribadi? Bila ya, abaikan. Bila diperlukan pihak lain, lacak.

Analisis setiap opsi:

- **A.** ✓ Kunci API tidak boleh masuk repositori.
- **B.** ✗ Jebakan: README adalah dokumentasi yang harus dibagikan.
- **C.** ✓ Hasil kompilasi dapat dibuat ulang dan tidak perlu dilacak.
- **D.** ✓ Data asli pelanggan tidak dimasukkan ke repositori.
- **E.** ✗ Jebakan: skrip migrasi adalah rancangan basis data yang perlu dilacak bersama kode.

Jawaban: A, C, dan D.

No. 24 Kunci: A

Konsep: Pada MAJOR.MINOR.PATCH, MAJOR naik bila ada perubahan yang tidak kompatibel, MINOR naik bila menambah fitur yang kompatibel, dan PATCH naik bila hanya memperbaiki bug. Bila satu rilis memuat beberapa jenis perubahan, yang dinaikkan adalah tingkat paling besar, dan angka di kanannya kembali ke 0.

Langkah:

1. Perubahan nomor 3 tidak kompatibel, sehingga MAJOR naik: 1 → 2.
2. MINOR dan PATCH kembali ke 0.
3. Versi baru: 2.0.0.

Analisis setiap opsi:

- **A.** ✓ MAJOR naik menjadi 2; MINOR dan PATCH menjadi 0.
- **B.** ✗ Jebakan: 1.10.4 menaikkan MINOR dan PATCH sekaligus tanpa menaikkan MAJOR.
- **C.** ✗ Jebakan: 2.10.4 menaikkan MAJOR tetapi tidak mengembalikan MINOR dan PATCH ke 0.
- **D.** ✗ Jebakan: 1.10.0 hanya memperhitungkan fitur baru dan mengabaikan perubahan yang merusak.
- **E.** ✗ Jebakan: 2.0.1 menaikkan PATCH setelah MAJOR; PATCH harus kembali ke 0.

Jawaban: A.

No. 25 Kunci: C

Konsep: Sebuah perubahan sudah berada di main bila commit-nya dapat ditelusuri dari cabang main melalui garis induknya. Commit hasil penggabungan (merge) memiliki dua induk.

Langkah:

1. M4 memiliki dua induk: M3 dari main dan S2 dari fitur-skor. Jadi S1 dan S2 sudah masuk main.
2. P1 bercabang dari M3, tetapi tidak ada garis dari P1 menuju commit mana pun di main.
3. Jadi hanya P1 yang belum ada di main.

Analisis setiap opsi:

- **A.** ✗ Jebakan: S1 dan S2 sudah tergabung lewat commit M4.
- **B.** ✗ Jebakan: S2 sudah tergabung; hanya P1 yang belum.
- **C.** ✓ P1 belum digabung karena tidak ada commit penggabungan yang menerimanya.
- **D.** ✗ Jebakan: M3 berada di cabang main sendiri.
- **E.** ✗ Jebakan: cabang perbaikan-pajak belum digabung.

Jawaban: C.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Latihan sub bab 3.1

No. 1 Kunci: D

Konsep: Pada path relatif, ".." berarti naik satu folder dan "." berarti folder saat ini. Perintah cp menyalin (berkas asal tetap ada), sedangkan mv memindahkan. Tanda ~ berarti folder home pengguna (/home/dina).

Langkah:

1. Folder saat ini: /home/dina/proyek/kasir.
2. ".." → /home/dina/proyek; "../.." → /home/dina.
3. "../../template/config.json" → /home/dina/template/config.json (benar), lalu "." = folder saat ini sebagai tujuan.
4. Karena berkas asal harus tetap ada, perintahnya cp, bukan mv.

Analisis setiap opsi:

- **A.** ✗ Jebakan: path-nya benar, tetapi mv memindahkan berkas sehingga berkas di template hilang.
- **B.** ✗ "../template" menunjuk /home/dina/proyek/template, folder yang tidak ada.
- **C.** ✗ Arah salinan terbalik: menyalin dari folder kasir ke folder template.
- **D.** ✓ Path relatif menunjuk /home/dina/template/config.json dan cp tidak menghapus berkas asal.
- **E.** ✗ Jebakan: ~ berarti /home/dina, sehingga path ini menunjuk /home/dina/proyek/template yang tidak ada.

Jawaban: D.

No. 2 Kunci: A

Konsep: cd mengubah folder aktif; ".." berarti folder induk. mkdir membuat folder di dalam folder aktif. mv memindahkan berkas (berkas asal tidak tersisa).

Langkah:

1. cd /home/raka/gim → aktif: /home/raka/gim.
2. mkdir aset → dibuat /home/raka/gim/aset; cd aset → aktif: /home/raka/gim/aset.
3. mkdir suara → /home/raka/gim/aset/suara.
4. mv ../musik.mp3 suara/ → /home/raka/gim/musik.mp3 dipindah ke /home/raka/gim/aset/suara/.
5. cd .. → aktif kembali /home/raka/gim.

Analisis setiap opsi:

- **A.** ✓ Berkas pindah ke aset/suara dan cd .. kembali ke /home/raka/gim.
- **B.** ✗ Jebakan: mv dianggap menyalin, padahal mv memindahkan sehingga berkas asal tidak tersisa.
- **C.** ✗ Jebakan: cd .. dihitung dari /home/raka/gim, padahal saat itu folder aktif adalah aset.
- **D.** ✗ Folder suara dibuat di dalam aset, bukan langsung di dalam gim.
- **E.** ✗ "../musik.mp3" dibaca dari folder aset sehingga menunjuk /home/raka/gim/musik.mp3, dan tujuannya aset/suara di dalam gim.

Jawaban: A.

No. 3 Kunci: B

Konsep: Variabel lingkungan seperti PATH dibaca saat sebuah program dibuka. Program yang sudah berjalan sebelum PATH diubah (termasuk VS Code dan terminalnya) masih memakai PATH lama sampai dibuka ulang.

Langkah:

1. Penyebab awal: folder Python belum ada di PATH → sudah diperbaiki.
2. Terminal VS Code dibuka sejak pagi, sebelum PATH diubah → masih memakai PATH lama.
3. Tutup dan buka ulang VS Code agar PATH baru terbaca.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

• **A.** ✗ Pesannya muncul karena PATH, bukan karena berkas rusak; memasang ulang tidak perlu.
• **B.** ✓ VS Code yang dibuka ulang membaca PATH baru, sehingga perintah python dikenali.
• **C.** ✗ Jebakan: PATH sudah benar; menambah lagi hanya membuat isian ganda dan terminal lama tetap memakai PATH lama.
• **D.** ✗ Hak administrator tidak berkaitan dengan PATH yang belum terbaca.
• **E.** ✗ Jebakan: nama python3 lazim di Linux, tetapi masalahnya terminal belum membaca PATH baru, bukan nama berkasnya.

Jawaban: B.

No. 4 Kunci: C

Konsep: Virtual environment adalah lingkungan Python terpisah untuk satu proyek. Pustaka yang dipasang di luar venv (Python global) tidak terlihat dari dalam venv, sehingga pustaka harus dipasang saat venv aktif.

Langkah:

1. flask dipasang SEBELUM venv aktif → masuk ke Python global.
2. Program dijalankan di dalam venv → hanya melihat pustaka di venv → flask tidak ditemukan.
3. Perbaikan: aktifkan venv, jalankan pip install flask, lalu jalankan program lagi.

Analisis setiap opsi:

• **A.** ✗ Python dan venv tidak rusak; yang kurang hanya pustaka di dalam venv.
• **B.** ✗ Menyalin folder pustaka secara manual tidak lengkap dan tidak tercatat; cara yang benar memakai pip di dalam venv.
• **C.** ✓ flask terpasang di venv yang dipakai program, sesuai aturan tim.
• **D.** ✗ Jebakan: program memang berjalan, tetapi melanggar aturan tim dan versi pustaka antarproyek dapat saling mengganggu.
• **E.** ✗ Venv tidak diaktifkan lewat PATH permanen; masalahnya flask belum terpasang di venv.

Jawaban: C.

No. 5 Kunci: D

Konsep: Program server "mendengarkan" pada nomor port tertentu. Satu port di satu alamat hanya dapat dipakai satu program; bila port masih dipakai program lama, program baru gagal dengan pesan "Address already in use".

Langkah:

1. Pesan "Address already in use" → port 8000 sudah dipakai program lain.
2. Program itu adalah server kemarin yang masih berjalan.
3. Hentikan server lama (Ctrl+C di terminalnya) atau jalankan di port lain, misalnya 8001.

Analisis setiap opsi:

• **A.** ✗ Python tidak rusak; perintahnya kemarin berjalan normal.
• **B.** ✗ Port dipakai program, bukan terikat ke alamat IP yang perlu diganti.
• **C.** ✗ Jebakan: firewall mengatur akses dari luar, bukan penyebab port sudah dipakai program lain.
• **D.** ✓ Port 8000 dipegang server lama; menghentikannya atau memakai port lain menyelesaikan masalah.
• **E.** ✗ Folder proyek tidak berhubungan dengan port yang sedang terpakai.

Jawaban: D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Latihan sub bab 3.2

No. 1 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Dengan subnet mask 255.255.255.0, perangkat satu jaringan memiliki tiga angka depan alamat IP yang sama. Perangkat satu jaringan berkomunikasi langsung lewat switch; gateway (router) hanya dipakai untuk menjangkau jaringan lain, misalnya internet.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: masalah Laptop 3 adalah alamat IP-nya; lagi pula tidak ada router 192.168.6.1 di lab itu.

• **Pernyataan 2** ✓ 192.168.6.x berbeda jaringan dengan 192.168.5.x (mask 255.255.255.0).

• **Pernyataan 3** ✓ Komunikasi dalam satu jaringan tidak melewati gateway, sehingga gateway yang salah tidak berpengaruh ke server uji.

• **Pernyataan 4** ✗ Laptop 1 dan 2 justru menjadi berbeda jaringan dan gagal membuka server.

• **Pernyataan 5** ✓ Akses ke jaringan lain (internet) harus melewati gateway; 192.168.15.1 tidak ada di jaringan itu.

Jawaban: pernyataan 2, 3, 5.

No. 2 Kunci: B

Konsep: Perangkat dapat menjangkau server lokal bila berada di jaringan yang sama atau ada jalur antarjaringan. Jaringan tamu yang terpisah dan data seluler tidak dapat menjangkau alamat 192.168.1.x di lab.

Langkah:

1. Laptop Andi dan ponsel A di jaringan 192.168.1.x (Lab-RPL) → berhasil.
2. Ponsel B di 192.168.50.x (Tamu-Sekolah) yang tidak terhubung ke jaringan lab → gagal.
3. Ponsel C memakai data seluler, di luar jaringan sekolah → gagal.
4. Solusi: sambungkan B dan C ke Wi-Fi Lab-RPL.

Analisis setiap opsi:

• **A.** ✗ Masalahnya jaringan, bukan nomor port; port 5000 sudah dapat dibuka ponsel A.

• **B.** ✓ Di jaringan Lab-RPL, ponsel B dan C satu jaringan dengan laptop Andi.

• **C.** ✗ Mengganti alamat tanpa pindah jaringan membuat laptop tidak cocok dengan jaringannya sendiri, dan ponsel A ikut gagal.

• **D.** ✗ Jebakan: localhost di ponsel menunjuk ke ponsel itu sendiri, bukan ke laptop Andi.

• **E.** ✗ Jebakan: tanpa data seluler, ponsel C tidak otomatis tersambung ke Lab-RPL, dan ponsel B tetap di jaringan tamu.

Jawaban: B.

Latihan sub bab 3.3

No. 1 Kunci: 1B 2S 3B

Konsep: Model TCP/IP: lapisan aplikasi (HTTP, HTTPS, DNS) menyusun data; transport (TCP/UDP) menambahkan nomor port; internet (IP) menambahkan alamat IP asal dan tujuan; akses jaringan (Wi-Fi/Ethernet) mengirim data secara fisik.

Analisis setiap pernyataan:

• **Pernyataan 1 → Benar.** HTTP/HTTPS adalah protokol lapisan aplikasi yang menentukan format permintaan dan respons.

• **Pernyataan 2 → Salah.** Jebakan: alamat IP memang di lapisan internet, tetapi nomor port ditentukan di lapisan transport (TCP/UDP).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 3 → Benar.** Lapisan akses jaringan mengirim data secara fisik (Wi-Fi/kabel); bila putus, seluruh lapisan di atasnya ikut tidak dapat mengirim.

Jawaban: Benar, Salah, Benar.

No. 2 Kunci: A, C, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Protokol lapisan aplikasi memiliki fungsi berbeda: DNS (nama domain → alamat IP), HTTP/HTTPS (halaman web; HTTPS terenkripsi), SSH (akses perintah jarak jauh terenkripsi), FTP (unggah berkas tanpa enkripsi), SFTP (unggah berkas lewat SSH, terenkripsi).

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Setelah IP berganti, catatan DNS harus menunjuk alamat baru agar nama domain mengarah ke server yang benar.
• **Pernyataan 2** ✗ Jebakan: FTP biasa mengirim kata sandi dan berkas tanpa enkripsi.
• **Pernyataan 3** ✓ SSH memberikan akses baris perintah jarak jauh yang terenkripsi.
• **Pernyataan 4** ✓ SFTP berjalan di atas SSH sehingga data unggahan terenkripsi.
• **Pernyataan 5** ✓ HTTPS mengenkripsi lalu lintas antara peramban dan server.

Jawaban: pernyataan 1, 3, 4, 5.

Latihan sub bab 3.4

No. 1 Kunci: A, B, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Dalam komunikasi client-server berbasis TCP/IP, client mengirim permintaan HTTP ke server dan server membalas dengan respons berkode status (2xx berhasil, 4xx kesalahan permintaan dari client, 5xx galat di server). HTTPS mengenkripsi data selama perjalanan.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Pada arsitektur client-server, client meminta layanan dan server memproses lalu membalas.
• **Pernyataan 2** ✓ HTTPS mengenkripsi data selama perjalanan sehingga tidak dapat dibaca pihak lain.
• **Pernyataan 3** ✗ Jebakan: pengguna lain mungkin melihat ada lalu lintas data, tetapi isinya terenkripsi sehingga kata sandi tidak terbaca.
• **Pernyataan 4** ✗ Menu mungkin tampil dari ponsel, tetapi pesanan harus diproses dan disimpan oleh server.
• **Pernyataan 5** ✓ Kode 5xx berarti server gagal memproses permintaan, misalnya karena beban berlebih saat jam istirahat.

Jawaban: pernyataan 1, 2, 5.

No. 2 Kunci: 1B 2S 3S

Konsep: Dalam komunikasi TCP/IP, alamat IP menentukan komputer tujuan dan nomor port menentukan program/layanan di komputer itu (HTTP 80, HTTPS 443, MySQL 3306). Protokol aplikasi seperti HTTP dikirim melalui koneksi TCP.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap pernyataan:

• **Pernyataan 1 → Benar.** Alamat IP menunjuk server; nomor port (80 atau 3306) menunjuk program yang dituju.

• **Pernyataan 2 → Salah.** Jebakan: urutan menyalakan program tidak berpengaruh. http:// tanpa nomor port memakai port 80, sehingga yang dituju adalah aplikasi web.

• **Pernyataan 3 → Salah.** HTTP adalah protokol lapisan aplikasi yang berjalan di atas TCP; koneksi TCP dibuka lebih dulu sebelum permintaan HTTP dikirim.

Jawaban: Benar, Salah, Salah.

Kuis Akhir Bab 3

No. 1 Kunci: C

Konsep: `cp` membuat salinan (berkas asal tetap ada), `mv` memindahkan atau mengganti nama (berkas asal hilang), `rm` menghapus permanen berkas yang disebut, dan tanda bintang * mewakili banyak karakter sehingga `*.png` berarti semua berkas berakhiran `.png`.

Langkah (isi folder aktif dan `arsip`):

1. Awal: folder aktif berisi a.png, b.png, c.txt, d.png, arsip; `arsip` kosong.
2. `cp *.png arsip/`: a.png, b.png, d.png disalin ke `arsip`; berkas asal tetap ada.
3. `mv c.txt arsip/catatan.txt`: c.txt berpindah ke `arsip` dengan nama catatan.txt; di folder aktif c.txt hilang.
4. `rm b.png`: b.png di folder aktif terhapus; salinannya di `arsip` tetap ada.
5. `ls`: a.png, arsip, d.png. `ls arsip`: a.png, b.png, catatan.txt, d.png.

Analisis setiap opsi:

• **A.** ✗ Jebakan: b.png dikira ikut terhapus dari `arsip`, padahal `rm` hanya menghapus berkas pada folder aktif.

• **B.** ✗ Jebakan: c.txt dikira masih ada di folder aktif, padahal sudah dipindah.

• **C.** ✓ Hasil sesuai langkah di atas.

• **D.** ✗ Jebakan: a.png di folder aktif terhapus, padahal yang dihapus hanya b.png.

• **E.** ✗ Jebakan: a.png dan d.png dikira tidak disalin ke `arsip`, padahal `cp *.png` menyalin semua berkas .png.

Jawaban: C.

No. 2 Kunci: A, B, D

Konsep: "Permission denied" berarti pengguna tidak memiliki izin yang dibutuhkan (baca, tulis, atau jalankan). Tindakan yang tepat: periksa izin dengan `ls -l`, lalu ubah izin atau kepemilikan seperlunya, bukan membuka akses untuk semua orang.

Langkah:

1. Baca izin: `-rw-r--r--`. Pemilik: `rw-`, grup: `r--`, lainnya: `r--`. Tidak ada huruf `x`.
2. Berkas bukan program yang dapat dijalankan karena belum ada izin execute; pemilik berkas adalah tomi (yang menjalankan).
3. Perbaikan: `chmod u+x cadangan.sh`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:
• **A.** ✓ Tanpa `x`, berkas tidak dapat dijalankan.
• **B.** ✓ `u+x` menambah izin jalankan bagi pemilik.
• **C.** ✗ Jebakan: izin 777 membuka berkas bagi siapa saja, berbahaya dan tidak perlu.
• **D.** ✓ `ls -l` menunjukkan pemiliknya tomi; yang kurang adalah izin `x`.
• **E.** ✗ Jebakan: pesan itu berasal dari izin berkas, bukan kerusakan bash.

Jawaban: A, B, dan D.

No. 3 Kunci: A

Konsep: `requirements.txt` mencatat pustaka proyek. Rekan memasangnya dengan `pip install -r requirements.txt`; sebaliknya, `pip freeze > requirements.txt` adalah cara menulis daftar itu.

Langkah: Tentukan arah: berkas sudah ada dan akan dibaca untuk memasang pustaka, sehingga yang diperlukan adalah pemasangan dari berkas, yaitu opsi `-r`.

Analisis setiap opsi:
• **A.** ✓ Memasang semua pustaka yang tercatat di berkas.
• **B.** ✗ Jebakan: perintah ini menimpa berkas dengan daftar pustaka yang terpasang di komputer ini, bukan memasang.
• **C.** ✗ Jebakan: `python -m venv` membuat lingkungan virtual; argumen yang diberikan adalah nama folder, bukan daftar pustaka.
• **D.** ✗ Jebakan: tanpa `-r`, pip menganggap `requirements.txt` sebagai nama pustaka.
• **E.** ✗ Jebakan: berkas teks itu bukan program Python yang dapat dijalankan.

Jawaban: A.

No. 4 Kunci: B

Konsep: Path relatif dihitung dari folder aktif. Simbol `..` berarti folder induk (naik satu tingkat).

Langkah:
1. Folder aktif: `/home/budi/proyek/src`.
2. Segmen pertama `..` naik satu tingkat ke `/home/budi/proyek`.
3. Segmen berikutnya masuk ke `docs`, lalu berkas `panduan.txt`.
4. Hasil: `/home/budi/proyek/docs/panduan.txt`. Pada gambar, berkas itu memang berada di bawah `docs`.

Analisis setiap opsi:
• **A.** ✗ Jebakan: ini mencari `docs` di dalam `src`, seolah-olah `..` tidak naik tingkat.
• **B.** ✓ Naik satu tingkat dari `src`, lalu masuk `docs`.
• **C.** ✗ Jebakan: naik dua tingkat, bukan satu.
• **D.** ✗ Jebakan: folder `budi` terlewat; bukan susunan yang ada pada gambar.
• **E.** ✗ Jebakan: `docs` tidak berada di dalam `aset`.

Jawaban: B.

No. 5 Kunci: E

Konsep: Nilai izin satu golongan = jumlah r (4), w (2), x (1) yang diberikan. Tiga angka berurutan menunjuk pemilik, grup, dan lainnya.

Langkah:
1. backup.sh: pemilik rwx = 4 + 2 + 1 = 7; grup r-x = 4 + 1 = 5; lainnya --- = 0 → 750.
2. catatan.txt: pemilik rw- = 6; grup rw- = 6; lainnya r-- = 4 → 664.
3. kunci.pem: pemilik r-- = 4; grup 0; lainnya 0 → 400.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:
• **A.** ✗ Jebakan: 600 untuk kunci.pem memberi pemilik hak tulis yang tidak diminta.
• **B.** ✗ Jebakan: 755 memberi golongan lainnya hak baca dan jalankan padahal tidak boleh ada.
• **C.** ✗ Jebakan: 646 menukar angka grup dan lainnya pada catatan.txt (grup hanya baca, lainnya baca dan tulis).
• **D.** ✗ Jebakan: 740 membuat grup hanya dapat membaca, padahal grup perlu menjalankan skrip.
• **E.** ✓ 750, 664, dan 400.

Jawaban: E.

No. 6 Kunci: B

Konsep: `pip install` memasang pustaka ke lingkungan yang sedang aktif. Bila dijalankan sebelum venv diaktifkan, pustaka masuk ke Python global dan tidak tampak di dalam venv.

Langkah:
1. Baris pertama (`pip install requests`) berjalan sebelum ada venv yang aktif, sehingga `requests` terpasang di Python global.
2. Setelah `source .venv/bin/activate`, Python yang dipakai adalah milik venv yang masih kosong.
3. Maka `import requests` gagal. Perbaikan: dengan venv aktif, jalankan `pip install requests`.

Analisis setiap opsi:
• **A.** ✗ Jebakan: pustakanya tidak rusak; ia hanya tidak ada di lingkungan venv.
• **B.** ✓ Memasang ke lingkungan yang salah; ulangi saat venv aktif.
• **C.** ✗ Jebakan: Python tidak perlu dipasang ulang untuk mengenali venv.
• **D.** ✗ Jebakan: nama folder tidak berpengaruh.
• **E.** ✗ Jebakan: venv harus dibuat lebih dulu; baru diaktifkan dengan source.

Jawaban: B.

No. 7 Kunci: E

Konsep: Switch menghubungkan banyak perangkat dalam satu jaringan; router menghubungkan jaringan yang berbeda, termasuk jaringan lab dengan internet. Alamat router yang dipakai untuk keluar disebut gateway.

Langkah: Tentukan: dua jaringan berbeda (lab dan penyedia layanan) membutuhkan perangkat yang meneruskan data antarjaringan.

Analisis setiap opsi:
• **A.** ✗ Jebakan: switch menyatukan perangkat dalam satu jaringan, bukan antarjaringan.
• **B.** ✗ Jebakan: hub meneruskan ke semua port dalam satu jaringan.
• **C.** ✗ Jebakan: access point memancarkan Wi-Fi ke jaringan yang sudah ada.
• **D.** ✗ Jebakan: kabel hanya media penyalur data.
• **E.** ✓ Router menghubungkan jaringan lab dengan jaringan lain.

Jawaban: E.

No. 8 Kunci: D

Konsep: `cp` membuat salinan (berkas asal tetap ada), `mv` memindahkan atau mengganti nama (berkas asal hilang), dan setiap `cd` mengubah folder aktif sehingga path relatif berikutnya dihitung dari folder yang baru.

Langkah (tulis ulang folder aktif tiap langkah):
1. `mkdir arsip` membuat folder `/home/sari/tugas/arsip`.
2. `cp laporan.txt arsip/` membuat salinan: sekarang ada `tugas/laporan.txt` dan `arsip/laporan.txt`.
3. `cd gambar` membuat folder aktif menjadi `/home/sari/tugas/gambar`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

4. `mv ../laporan.txt ../arsip/laporan_lama.txt`: `..` adalah `/home/sari/tugas`, jadi yang dipindah adalah `tugas/laporan.txt` dan tujuannya `arsip/laporan_lama.txt`.
5. Hasil: `arsip` berisi `laporan.txt` (salinan dari langkah 2) dan `laporan_lama.txt`; `laporan.txt` di `tugas` sudah tidak ada.

Analisis setiap opsi:

- **A.** ✗ Jebakan: mengabaikan langkah `mv` sehingga `laporan_lama.txt` dan hilangnya berkas di `tugas` terlewat.
- **B.** ✗ Jebakan: salinan `arsip/laporan.txt` dari `cp` terlupa.
- **C.** ✗ Jebakan: `mv` memindahkan berkas, bukan menyalinnya.
- **D.** ✓ Salinan dari `cp` tetap ada, `mv` memindahkan berkas asli.
- **E.** ✗ Jebakan: `mv` membawa berkas ke `arsip`, bukan ke `gambar` (folder tempat Sari berdiri).

Jawaban: D.

No. 9 Kunci: A, B, D

Konsep: `localhost` (`127.0.0.1`) selalu berarti komputer itu sendiri. Perangkat lain menjangkau server lewat alamat IP server pada jaringan yang sama.

Langkah:

1. Di laptop, localhost = laptop; di ponsel, localhost = ponsel. Aplikasinya tidak berjalan di ponsel.
2. Agar ponsel dapat membukanya, pakai alamat IP laptop, mis. `http://192.168.x.y:5000`.
3. Ponsel dan laptop harus berada pada jaringan yang sama; data seluler berada di jaringan lain.

Analisis setiap opsi:

- **A.** ✓ localhost di ponsel menunjuk ponsel.
- **B.** ✓ Alamat IP laptop di jaringan bersama diperlukan.
- **C.** ✗ Jebakan: port bukan penyebab; localhost tetap menunjuk ponsel.
- **D.** ✓ Data seluler berada di jaringan yang berbeda.
- **E.** ✗ Jebakan: aplikasi berjalan di laptop; ponsel hanya perlu peramban.

Jawaban: A, B, dan D.

No. 10 Kunci: D

Konsep: Menelusuri gangguan dari layer bawah ke atas: akses jaringan (kabel menyala), internet (ping ke IP berhasil), transport (port dapat dijangkau), lalu aplikasi (protokol seperti DNS).

Langkah:

1. Lampu port menyala: layer akses jaringan baik.
2. Ping ke alamat IP berhasil: layer internet baik.
3. Membuka alamat IP dengan port 8080 berhasil: layer transport dan layanan baik.
4. Yang gagal hanya penerjemahan nama: DNS, protokol layer aplikasi.

Analisis setiap opsi:

- **A.** ✗ Jebakan: lampu menyala dan ping berhasil.
- **B.** ✗ Jebakan: ping ke IP server berhasil.
- **C.** ✗ Jebakan: port 8080 berhasil dibuka lewat alamat IP.
- **D.** ✓ Hanya penerjemahan nama yang gagal; DNS adalah protokol layer aplikasi.
- **E.** ✗ Jebakan: satu uji gagal menunjukkan ada gangguan.

Jawaban: D.

No. 11 Kunci: A, B, D

Konsep: TCP untuk data yang tidak boleh kurang atau tertukar (login, berkas, obrolan). UDP untuk data yang lebih baik dilewati daripada ditunggu (posisi waktu nyata, siaran langsung).

Langkah: Tentukan untuk tiap fitur: boleh hilang atau tidak?

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

- **A.** ✓ Login tidak boleh rusak atau tertukar.
- **B.** ✓ Posisi terbaru menggantikan posisi lama.
- **C.** ✗ Jebakan: UDP tidak mengulang bagian yang hilang sehingga tidak andal untuk berkas.
- **D.** ✓ Pesan obrolan harus lengkap dan berurutan.
- **E.** ✗ Jebakan: TCP lebih andal, bukan lebih cepat.

Jawaban: A, B, dan D.

No. 12 Kunci: B, C, E

Konsep: Prinsip firewall: buka hanya port yang diperlukan dan tutup sisanya. Klien memanggil server aplikasi (443); basis data tidak dijangkau dari internet; SSH dibatasi untuk administrator.

Langkah: Nilai tiap aturan: apakah perlu, dan apakah membuka akses seminimal mungkin?

Analisis setiap opsi:

- **A.** ✗ Jebakan: port basis data yang terbuka ke publik adalah celah keamanan; klien tidak berbicara langsung ke basis data.
- **B.** ✓ Port 443 dibutuhkan untuk HTTPS ke server aplikasi.
- **C.** ✓ Hanya server aplikasi yang membutuhkan basis data.
- **D.** ✗ Jebakan: membuka semua port melanggar prinsip akses minimum.
- **E.** ✓ SSH sebaiknya dibatasi untuk administrator.

Jawaban: B, C, dan E.

No. 13 Kunci: A

Konsep: Metode HTTP menyatakan maksud permintaan: GET membaca data, POST membuat atau mengirim data baru, PUT memperbarui data, DELETE menghapus data.

Langkah:

1. Baris pertama permintaan memuat metode: POST.
2. Body berisi data baru yang dikirim: menu dan jumlah.
3. POST ke `/pesanan` dengan body data: membuat pesanan baru.

Analisis setiap opsi:

- **A.** ✓ POST dengan data pesanan membuat pesanan baru.
- **B.** ✗ Jebakan: GET tidak membawa body dan hanya membaca data.
- **C.** ✗ Jebakan: metodenya POST, bukan DELETE.
- **D.** ✗ Jebakan: metode pada permintaan itu POST, bukan PUT; PUT dipakai untuk memperbarui.
- **E.** ✗ Jebakan: POST mengirim data, bukan membaca isi pesanan.

Jawaban: A.

No. 14 Kunci: A, C, E

Konsep: Enkapsulasi: tiap layer menambahkan header. Transport menambah port, internet menambah alamat IP, akses jaringan menambah alamat MAC. Penerima membuka bungkus dengan urutan terbalik.

Langkah:

1. Header R berisi port → ditambahkan di layer transport.
2. Header Q berisi alamat IP → layer internet.
3. Header P berisi alamat MAC → layer akses jaringan; ditambahkan terakhir di pengirim, dibuka pertama di penerima.
4. Port 443 adalah port HTTPS.

Analisis setiap opsi:

- **A.** ✓ Port ditentukan di layer transport.
- **B.** ✗ Jebakan: alamat IP ditambahkan di layer internet, bukan akses jaringan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **C.** ✓ Bungkus paling luar ditambahkan terakhir dan dibuka pertama.
- **D.** ✗ Jebakan: alamat IP ada pada header Q, bukan R.
- **E.** ✓ 443 adalah port bawaan HTTPS.

Jawaban: A, C, dan E.

No. 15 Kunci: D

Konsep: Alamat berawalan 169.254 adalah alamat darurat yang diberikan sistem bila perangkat yang diatur otomatis gagal memperoleh alamat dari DHCP. Gateway kosong menegaskan hal itu.

Langkah:

1. Alamat 169.254.31.7 bukan alamat jaringan lab; gateway kosong.
2. Laptop diatur otomatis, tetapi tidak mendapat jawaban DHCP.
3. Periksa sambungan Wi-Fi ke jaringan yang tepat dan layanan DHCP, bukan aplikasi.

Analisis setiap opsi:

- **A.** ✗ Jebakan: laptop diatur otomatis; alamat itu bukan hasil pengaturan manual.
- **B.** ✗ Jebakan: DNS belum menjadi persoalan karena alamat saja belum diperoleh.
- **C.** ✗ Jebakan: Default Gateway terlihat kosong.
- **D.** ✓ Alamat 169.254 menandakan gagal memperoleh alamat dari DHCP.
- **E.** ✗ Jebakan: masalahnya pada jaringan, bukan aplikasi.

Jawaban: D.

No. 16 Kunci: 1S 2B 3B

Konsep: TCP berorientasi sambungan dan andal (konfirmasi, pengulangan), sedangkan UDP tanpa sambungan, ringan, dan tidak menjamin. TCP lebih andal, bukan lebih cepat.

Langkah: Nilai tiap pernyataan terhadap sifat TCP dan UDP.

Analisis:

- **1.** ✗ Jebakan: menjamin sampai membuat TCP lebih andal, tetapi lebih berat; UDP lebih ringan.
- **2.** ✓ Posisi terbaru lebih berguna daripada menunggu posisi lama yang tertunda.
- **3.** ✓ Jabat tangan tiga langkah adalah ciri TCP.

Jawaban: Salah, Benar, Benar.

No. 17 Kunci: A

Konsep: Dengan mask 255.255.255.0, dua perangkat berada pada satu jaringan bila tiga oktet pertama alamatnya sama persis.

Langkah:

1. Server: 192.168.**40**.10, bagian jaringan 192.168.40.
2. PC-1: 192.168.40.25 → sama. PC-3: 192.168.40.77 → sama.
3. PC-2: 192.168.**4**.25 → bagian jaringan 192.168.4, berbeda dengan 192.168.40. Oktet ketiga 4 dan 40 mudah terbaca sama.
4. PC-2 harus diberi alamat 192.168.40.x yang valid dan belum dipakai.

Analisis setiap opsi:

- **A.** ✓ PC-2 berada di jaringan 192.168.4.
- **B.** ✗ Jebakan: PC-1 sudah satu jaringan dengan server.
- **C.** ✗ Jebakan: PC-3 juga sudah satu jaringan dengan server.
- **D.** ✗ Jebakan: switch tidak memiliki alamat IP untuk menentukan jaringan pada soal ini.
- **E.** ✗ Jebakan: server adalah pembanding, bukan yang harus diubah.

Jawaban: A.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 18 Kunci: C

Konsep: Alamat statis yang benar harus: satu jaringan dengan perangkat lain, berada dalam rentang host (bukan ID jaringan dan bukan broadcast), belum dipakai, berada di luar rentang DHCP, dan valid.

Langkah: Periksa tiap calon terhadap lima syarat itu.

Analisis setiap opsi:
- **A.** ✗ Jebakan: 10.20.8.255 adalah alamat broadcast jaringan /24, tidak boleh dipakai perangkat.
- **B.** ✗ Jebakan: 10.20.8.12 sudah dipakai printer (konflik IP).
- **C.** ✓ Satu jaringan, dalam rentang host, belum dipakai, dan di luar rentang DHCP 50–150.
- **D.** ✗ Jebakan: 10.20.8.100 berada di dalam rentang DHCP.
- **E.** ✗ Jebakan: 10.20.9.30 berada di jaringan 10.20.9, berbeda dari 10.20.8.

Jawaban: C.

No. 19 Kunci: E

Konsep: Kode 4xx menandakan kesalahan di sisi klien, dan 403 Forbidden berarti permintaan dimengerti tetapi ditolak karena tidak ada hak akses. Kode 404 berarti alamat tidak ditemukan, dan 5xx berarti server gagal memproses.

Langkah: Server menjawab, alamat benar, dan siswa sudah login; yang kurang adalah hak akses ke halaman guru.

Analisis setiap opsi:
- **A.** ✗ Jebakan: server menjawab, jadi tidak mati.
- **B.** ✗ Jebakan: alamat yang salah menghasilkan 404, bukan 403, dan soal menyebut alamatnya benar.
- **C.** ✗ Jebakan: jaringan yang terputus tidak menghasilkan jawaban dari server.
- **D.** ✗ Jebakan: kegagalan memproses permintaan yang sah adalah kelompok 5xx.
- **E.** ✓ 403 = tidak memiliki hak akses.

Jawaban: E.

No. 20 Kunci: C

Konsep: Telusuri dari bawah ke atas. Bila IP dapat di-ping dan nama diterjemahkan, tetapi sambungan ke satu port ditolak, layanan di port itu tidak berjalan atau firewall menutupnya.

Langkah:
1. Nama diterjemahkan dengan benar: DNS baik.
2. Ping berhasil: jaringan dan alamat IP baik.
3. Port 8080 ditolak dari luar, tetapi dari dalam server layanan menjawab normal: layanan berjalan.
4. Sisa kemungkinan: firewall menutup port itu dari luar.

Analisis setiap opsi:
- **A.** ✗ Jebakan: nslookup berhasil.
- **B.** ✗ Jebakan: ping berhasil.
- **C.** ✓ Layanan berjalan, tetapi port tertutup untuk sambungan luar.
- **D.** ✗ Jebakan: ping dan nslookup berhasil, jadi sambungan ada.
- **E.** ✗ Jebakan: konflik IP akan mengganggu ping dan layanan, bukan hanya satu port.

Jawaban: C.

No. 21 Kunci: 1B 2S 3S

Konsep: Gateway dipakai untuk keluar dari jaringan; DNS menerjemahkan nama menjadi alamat IP; DHCP memberi alamat IP otomatis; ID jaringan dan broadcast tidak diberikan kepada perangkat.

Langkah: Cocokkan tiap pernyataan dengan peran perangkat atau layanannya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis:

- **1.** ✓ Dalam satu jaringan, gateway tidak dipakai; hanya akses keluar yang terganggu.
- **2.** ✗ Jebakan: menerjemahkan nama adalah tugas DNS, bukan DHCP.
- **3.** ✗ Jebakan: alamat dengan bagian host semuanya 0 (ID jaringan) atau semuanya 1 (broadcast) tidak diberikan kepada perangkat.

Jawaban: Benar, Salah, Salah.

No. 22 Kunci: 1S 2B 3B

Konsep: Klien tidak berbicara langsung dengan basis data; server aplikasi menjadi pintu tunggal yang memeriksa permintaan. Skema https memakai port bawaan 443. Kode 4xx adalah kesalahan sisi klien.

Langkah: Nilai tiap pernyataan terhadap model client–server dan aturan URL.

Analisis:

- **1.** ✗ Jebakan: akses langsung ke basis data melewati pemeriksaan server dan memperluas risiko; server aplikasi menjadi pintu tunggal.
- **2.** ✓ Skema https tanpa port memakai 443.
- **3.** ✓ 4xx = kesalahan sisi klien (404: alamat tidak ada).

Jawaban: Salah, Benar, Benar.

No. 23 Kunci: B

Konsep: Layer transport (TCP atau UDP) menentukan nomor port asal dan tujuan sehingga data sampai ke program yang tepat. Alamat IP ditentukan di layer internet, dan alamat MAC di layer akses jaringan.

Langkah: Tanyakan pada tiap layer: apa informasi yang ditambahkannya? Port = untuk program yang mana → transport.

Analisis setiap opsi:

- **A.** ✗ Jebakan: layer aplikasi menyusun isi pesan sesuai protokol, bukan memberi nomor port.
- **B.** ✓ Header transport memuat port asal dan port tujuan.
- **C.** ✗ Jebakan: layer internet memuat alamat IP.
- **D.** ✗ Jebakan: layer akses jaringan memuat alamat MAC.
- **E.** ✗ Jebakan: tiap layer mengurus tugasnya masing-masing.

Jawaban: B.

No. 24 Kunci: A, B, E

Konsep: Dengan mask /24, tiga oktet pertama harus sama agar dua perangkat satu jaringan. Gateway hanya dipakai untuk keluar jaringan dan harus berada di jaringan perangkat itu sendiri.

Langkah:

1. Bagian jaringan: A = 192.168.12, B = 192.168.12, C = 192.168.21.
2. A dan B satu jaringan; C berbeda.
3. Gateway B (192.168.1.1) bukan alamat di jaringan 192.168.12, sehingga keliru; dampaknya hanya akses keluar jaringan.

Analisis setiap opsi:

- **A.** ✓ Satu jaringan; gateway tidak dipakai di dalam jaringan.
- **B.** ✓ Gateway B tidak berada di jaringan B.
- **C.** ✗ Jebakan: dengan mask /24, tiga oktet pertama harus sama; A dan C berbeda pada oktet ketiga.
- **D.** ✗ Jebakan: mask yang sama tidak membuat jaringannya sama; bagian jaringan B dan C berbeda.
- **E.** ✓ Oktet ketiga 21 dan 12 membuat C berada di jaringan lain.

Jawaban: A, B, dan E.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 25 Kunci: A, C, D

Konsep: 2xx berhasil (200 OK, 201 Created), 4xx kesalahan di sisi klien (401 belum masuk, 404 alamat tidak ada), 5xx kesalahan di sisi server (503 layanan tidak tersedia).

Langkah: Baca setiap baris: angka pertama kode status, lalu artinya.

Analisis setiap opsi:
- **A.** ✓ 404: server menjawab, tetapi alamat yang diminta tidak ada.
- **B.** ✗ Jebakan: 201 Created berarti data baru berhasil dibuat.
- **C.** ✓ 401 berarti belum masuk atau belum terautentikasi.
- **D.** ✓ 503 = layanan sedang tidak tersedia; kesalahan di sisi server.
- **E.** ✗ Jebakan: 200 berarti berhasil.

Jawaban: A, C, dan D.

Latihan sub bab 4.1

No. 1 Kunci: E

Konsep: Operator / selalu menghasilkan float; // menghasilkan hasil bagi bulat (int bila kedua bilangan int); % menghasilkan sisa bagi.

Langkah:
1. 7 / 2 = 3.5 (float).
2. 7 // 2 = 3 (bagian bulat).
3. 7 % 2 = 1 (sisa; 2 × 3 = 6, sisa 1).
4. Artinya: tiap kelompok mendapat 3 voucher dan tersisa 1 voucher.

Analisis setiap opsi:
- **A.** ✗ // membuang bagian pecahan, jadi hasilnya 3, bukan 3.5.
- **B.** ✗ Jebakan: / di Python 3 tidak membulatkan; 7 / 2 = 3.5.
- **C.** ✗ Jebakan: % adalah sisa bagi bilangan bulat (1 voucher), bukan pecahan 0.5.
- **D.** ✗ Hasil operator / dan // tertukar.
- **E.** ✓ 7 / 2 = 3.5, 7 // 2 = 3, dan 7 % 2 = 1.

Jawaban: E.

No. 2 Kunci: A

Konsep: Operator + pada str hanya dapat menggabungkan str dengan str. Bilangan harus diubah dulu dengan `str()`, atau dipisah koma di print(), atau ditaruh di f-string. Operasi int dengan float menghasilkan float.

Langkah:
1. nama bertipe str, nyawa int, kecepatan float.
2. "Kancil nyawa " + 3 → str + int → TypeError.
3. Pilihan lain tidak menggabungkan str dengan int memakai +.

Analisis setiap opsi:
- **A.** ✓ str + int menimbulkan TypeError.
- **B.** ✗ Koma di print() mencetak beberapa nilai berbeda tipe tanpa error: Kancil nyawa 3.
- **C.** ✗ int × float sah: 3 × 2.5 = 7.5.
- **D.** ✗ Jebakan: bentuknya mirip pilihan A, tetapi nyawa sudah diubah menjadi str sehingga sah.
- **E.** ✗ f-string mengubah nilai di dalam kurung kurawal menjadi teks: Kancil nyawa 3.

Jawaban: A.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 3 Kunci: C

Konsep: Pada list dua dimensi peta[b][k], indeks b memilih baris (list di dalamnya) dan indeks k memilih kolom pada baris itu. Indeks dimulai dari 0.

Langkah:

1. peta[1] = baris kedua: [".", ".", "K", "#"]; elemen indeks 2 = "K".
2. peta[2] = baris ketiga: ["#", ".", ".", "."]; elemen indeks 0 = "#".
3. Keluaran: K #.

Analisis setiap opsi:

- **A.** ✗ Jebakan: indeks baris dan kolom tertukar: peta[2][1] = "." dan peta[0][2] = "#".
- **B.** ✗ peta[2][0] adalah elemen pertama baris ketiga, yaitu "#".
- **C.** ✓ peta[1][2] = "K" dan peta[2][0] = "#".
- **D.** ✗ Urutan keluaran terbalik; print mencetak sesuai urutan argumen.
- **E.** ✗ Jebakan: indeks dianggap mulai dari 1, sehingga peta[1][2] dibaca sebagai baris pertama kolom kedua (".").

Jawaban: C.

No. 4 Kunci: B

Konsep: Dictionary menyimpan pasangan kunci–nilai. Dengan kode unik sebagai kunci, data dapat diambil langsung, misalnya `barang["8991234"]`, dan diubah dengan mudah. List cocok untuk data berurutan, set untuk himpunan nilai unik tanpa pasangan, dan tuple untuk data yang tidak diubah.

Langkah:

1. Kebutuhan: cari berdasarkan kode unik, ambil nama dan harga, harga dapat diubah.
2. Pasangan "kode → (nama, harga)" paling cocok disimpan sebagai dictionary.

Analisis setiap opsi:

- **A.** ✗ List nama saja tidak menyimpan kode dan harga, dan harus dicari satu per satu.
- **B.** ✓ Kode unik sebagai kunci memasangkan langsung ke nama dan harga, serta mudah diperbarui.
- **C.** ✗ Jebakan: set memang menjamin kode tidak ganda, tetapi tidak menyimpan nama dan harga.
- **D.** ✗ Jebakan: dapat bekerja, tetapi urutan dua list mudah bergeser saat barang ditambah atau dihapus sehingga nama dan harga tertukar.
- **E.** ✗ Tuple tidak dapat diubah, padahal harga sesekali berubah, dan tidak memuat kode barang.

Jawaban: B.

No. 5 Kunci: E

Konsep: Tumpukan (stack) bekerja LIFO: yang terakhir masuk, pertama keluar. Pada list Python, `append()` menaruh di akhir dan `pop()` tanpa indeks mengambil elemen terakhir. Antrean (queue) bekerja FIFO, misalnya `pop(0)`.

Langkah:

1. Undo membatalkan aksi TERAKHIR lebih dulu → pola LIFO (stack).
2. append() menaruh aksi di akhir list, jadi aksi terakhir ada di ujung list.
3. pop() tanpa indeks mengambil sekaligus menghapus elemen terakhir.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

• **A.** ✗ Jebakan: pop(0) mengambil aksi PERTAMA (pola antrean), sehingga yang dibatalkan justru aksi paling awal.
• **B.** ✗ Mengurutkan menurut abjad merusak urutan waktu aksi.
• **C.** ✗ clear() menghapus seluruh riwayat, bukan satu aksi terakhir.
• **D.** ✗ riwayat[0] hanya membaca aksi pertama dan tidak menghapusnya.
• **E.** ✓ pop() mengambil dan menghapus aksi terakhir, sesuai pola tumpukan.

Jawaban: E.

Latihan sub bab 4.2

No. 1 Kunci: C

Konsep: Diagram alir percabangan diperiksa dari kondisi paling atas. Bila kondisi bernilai Ya, cabang Ya dijalankan dan kondisi di bawahnya tidak diperiksa lagi. Tanda ≥ dan ≤ berarti nilai batas termasuk.

Langkah:

1. P1: 180.000 ≥ 200.000? Tidak → 8 ≤ 5? Tidak → ongkir = 2.000 × 8 = Rp16.000.
2. P2: 200.000 ≥ 200.000? Ya (batas termasuk) → ongkir = Rp0.
3. P3: 150.000 ≥ 200.000? Tidak → 5 ≤ 5? Ya (batas termasuk) → ongkir = Rp8.000.
4. Jumlah: 16.000 + 0 + 8.000 = Rp24.000.

Analisis setiap opsi:

• **A.** ✗ Jebakan: P3 dianggap jarak 5 tidak memenuhi "≤ 5" sehingga dihitung 2.000 × 5 = Rp10.000.
• **B.** ✗ Jebakan: P2 dianggap tidak gratis karena 200.000 tidak "lebih dari" 200.000, sehingga dihitung 2.000 × 12 = Rp24.000.
• **C.** ✓ P1 Rp16.000, P2 Rp0, P3 Rp8.000.
• **D.** ✗ Hanya ongkir P1; ongkir P3 sebesar Rp8.000 terlupa.
• **E.** ✗ Dua kesalahan sekaligus: P2 dihitung Rp24.000 dan P3 dihitung Rp10.000.

Jawaban: C.

No. 2 Kunci: A, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Aturan bercabang diterapkan berurutan, dan setiap cabang memiliki syarat batas yang tegas ("di bawah 5" berarti 5 tidak termasuk). Kasus uji yang baik mencakup setiap cabang dan nilai batasnya.

Langkah:

1. Cabang 1: umur < 5 → Rp0 (tanpa potongan).
2. Cabang 2: punya kartu pelajar → Rp15.000; selain itu Rp25.000.
3. Bila Senin dan tiket berbayar → kurangi Rp5.000.

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ Umur di bawah 5 → gratis. Potongan Senin hanya untuk tiket berbayar, jadi hasilnya Rp0, bukan negatif.
• **Pernyataan 2** ✗ Jebakan: potongan Senin terlupa. Rp15.000 − Rp5.000 = Rp10.000.
• **Pernyataan 3** ✗ Potongan hanya hari Senin; hari Selasa membayar Rp25.000.
• **Pernyataan 4** ✗ Jebakan nilai batas: 5 tahun tidak "di bawah 5", sehingga membayar Rp25.000.
• **Pernyataan 5** ✓ Rp25.000 − Rp5.000 = Rp20.000.

Jawaban: pernyataan 1, 5.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Latihan sub bab 4.3

No. 1 Kunci: 1B 2B 3S

Konsep: Syarat `while` diperiksa di awal setiap putaran; bila syarat gabungan dengan `and` salah salah satu bagiannya, perulangan berhenti. Operator `%` menghasilkan sisa bagi; level % 2 == 0 berarti level genap.

Langkah:

1. Level 1: ganjil → level 2 (nyawa 3).
2. Level 2: genap → nyawa 2 → level 3.
3. Level 3: ganjil → level 4.
4. Level 4: genap → nyawa 1 → level 5.
5. Level 5: ganjil → level 6. Cek syarat: 6 <= 5 salah → berhenti. Keluaran: 6 1 (5 putaran).

Analisis setiap pernyataan:

• **Pernyataan 1 → Benar.** Putaran berjalan untuk level 1, 2, 3, 4, 5. Setelah level menjadi 6, syarat level <= 5 salah sehingga perulangan berhenti.

• **Pernyataan 2 → Benar.** Jejak: level 2 → nyawa 1; level 4 → nyawa 0, lalu level menjadi 5. Syarat nyawa > 0 salah, perulangan berhenti dengan level 5 dan nyawa 0.

• **Pernyataan 3 → Salah.** Jebakan: level masih bertambah 1 pada putaran terakhir. Keluaran sebenarnya 6 1.

Jawaban: Benar, Benar, Salah.

No. 2 Kunci: B

Konsep: `continue` melewati sisa perintah pada putaran itu lalu lanjut ke elemen berikutnya; `break` menghentikan perulangan seluruhnya.

Langkah:

1. nilai 3 → skor = 0 + 3 × 2 = 6.
2. nilai 0 → continue (dilewati), skor tetap 6.
3. nilai 5 → skor = 6 + 10 = 16.
4. nilai −1 → break, perulangan berhenti; 4 dan 2 tidak diproses.
5. print(skor) → 16.

Analisis setiap opsi:

• **A.** ✗ Jebakan: break dianggap seperti continue sehingga 4 dan 2 ikut dihitung: 6 + 10 + 8 + 4 = 28.

• **B.** ✓ Skor 6 + 10 = 16, lalu berhenti di −1.

• **C.** ✗ Nilai dijumlahkan tanpa dikali 2: 3 + 5 = 8.

• **D.** ✗ Jebakan: −1 ikut dikali 2 lalu perulangan lanjut: 6 + 10 − 2 + 8 + 4 = 26.

• **E.** ✗ continue dianggap menghentikan perulangan sehingga hanya nilai 3 yang dihitung.

Jawaban: B.

No. 3 Kunci: E

Konsep: Syarat while diperiksa sebelum setiap putaran; perulangan berhenti segera setelah syarat bernilai salah. Operator % menghasilkan sisa bagi; minggu % 4 == 0 benar pada minggu ke-4, ke-8, dan seterusnya.

Langkah:

1. Minggu 1: 20.000; minggu 2: 40.000; minggu 3: 60.000.
2. Minggu 4: bonus 5.000 + setoran 20.000 → 85.000.
3. Minggu 5: 105.000.
4. Cek syarat: 105.000 < 100.000 salah → berhenti. Keluaran: 5 105000.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

- **A.** ✗ Perulangan belum berhenti di minggu 4 karena 85.000 masih kurang dari 100.000.
- **B.** ✗ Jebakan: bonus minggu ke-4 terlupa sehingga minggu 5 dianggap tepat 100.000.
- **C.** ✗ Jebakan: dianggap masih ada satu putaran lagi; padahal 105.000 sudah tidak memenuhi syarat < 100.000.
- **D.** ✗ Jebakan: pemeriksaan bonus dianggap terjadi sebelum minggu ditambah, sehingga bonus jatuh di putaran ke-1 dan ke-5.
- **E.** ✓ Minggu 4 mendapat bonus (85.000), minggu 5 menjadi 105.000, lalu berhenti.

Jawaban: E.

No. 4 Kunci: A

Konsep: `range(1, 4)` menghasilkan 1, 2, 3 (angka 4 tidak termasuk). `range(baris)` menghasilkan 0 sampai baris − 1, sehingga perulangan dalam berjalan sebanyak baris kali.

Langkah:

1. baris = 1 → perulangan dalam 1 kali → poin = 1.
2. baris = 2 → 2 kali menambah 2 → poin = 1 + 4 = 5.
3. baris = 3 → 3 kali menambah 3 → poin = 5 + 9 = 14.

Analisis setiap opsi:

- **A.** ✓ 1 × 1 + 2 × 2 + 3 × 3 = 14.
- **B.** ✗ Jebakan: perulangan dalam dianggap hanya sekali: 1 + 2 + 3 = 6.
- **C.** ✗ Menara dianggap 3 × 3 balok bernilai 1 poin.
- **D.** ✗ Jebakan: range(1, 4) dianggap ikut memuat 4: 1 + 4 + 9 + 16 = 30.
- **E.** ✗ Jebakan: range(baris) dianggap berjalan baris + 1 kali: 1 × 2 + 2 × 3 + 3 × 4 = 20.

Jawaban: A.

Latihan sub bab 4.4

No. 1 Kunci: D

Konsep: Parameter bawaan (default), misalnya `diskon=0`, dipakai bila argumennya tidak dikirim. Fungsi dapat memanggil fungsi lain, dan `return` di luar perulangan dijalankan setelah perulangan selesai.

Langkah:

1. Di dalam perulangan, harga_akhir(h) memakai diskon = 0 → harga tetap.
2. jumlah = 20.000 + 15.000 + 10.000 = 45.000.
3. return harga_akhir(45000, 5000) = 40.000.

Analisis setiap opsi:

- **A.** ✗ Jebakan: potongan 5.000 dianggap dikenakan pada setiap barang: 45.000 − 3 × 5.000.
- **B.** ✗ Jebakan: potongan voucher pada baris return terlupa.
- **C.** ✗ Jebakan: return dianggap berada di dalam perulangan sehingga hanya barang pertama dihitung: 20.000 − 5.000.
- **D.** ✓ Total 45.000 dikurangi voucher 5.000 = 40.000.
- **E.** ✗ diskon memiliki nilai bawaan 0, sehingga pemanggilan dengan satu argumen sah.

Jawaban: D.

No. 2 Kunci: D

Konsep: Fungsi Python dapat mengembalikan beberapa nilai sekaligus (dikemas sebagai tuple), lalu dibagikan ke beberapa variabel: `t, r = ringkas(...)`. Operator / selalu menghasilkan float.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah:
1. tertinggi = max([70, 80, 90]) = 90.
2. rata = 240 / 3 = 80.0 (float karena operator /).
3. return 90, 80.0 → t = 90, r = 80.0.
4. print(t, r) → 90 80.0.

Analisis setiap opsi:
• **A.** ✗ Jebakan: hasil / dianggap int, padahal selalu float (80.0).
• **B.** ✗ Tuple tercetak dengan kurung bila dicetak utuh; di sini isinya sudah dibagikan ke t dan r.
• **C.** ✗ Urutan tertukar: t menerima nilai pertama (tertinggi).
• **D.** ✓ t = 90 dan r = 80.0 dicetak dipisah spasi.
• **E.** ✗ Jebakan: Python mengizinkan return beberapa nilai sekaligus.

Jawaban: D.

No. 3 Kunci: A, C, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Desain modular memecah program menjadi fungsi atau modul yang masing-masing punya satu tugas jelas. Manfaatnya: tidak ada duplikasi kode, mudah diubah, mudah diuji, dapat dipakai ulang, dan dapat dikerjakan paralel oleh tim.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Satu fungsi untuk satu rumus menghilangkan duplikasi yang menyebabkan laporan dan struk berbeda.
• **Pernyataan 2** ✗ Jebakan: kodenya memang terkumpul, tetapi fungsi yang mengerjakan banyak tugas sulit diuji dan diubah; setiap fungsi sebaiknya punya satu tugas.
• **Pernyataan 3** ✓ Modul kecil dapat diuji sendiri sehingga kesalahan cepat ditemukan.
• **Pernyataan 4** ✓ Modul yang terpisah dapat dikerjakan paralel oleh anggota tim yang berbeda.
• **Pernyataan 5** ✓ Karena keduanya memakai satu fungsi yang sama, struk dan laporan menghitung diskon dengan cara yang sama.

Jawaban: pernyataan 1, 3, 4, 5.

No. 4 Kunci: A, B, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Desain modular yang efektif: setiap logika ditulis sekali di modul yang tepat lalu dipakai ulang (impor), setiap modul punya tugas jelas, dan antarmukanya didokumentasikan. Duplikasi logika antarmodul membuat hasil mudah tidak konsisten.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Dua salinan logika yang sama mudah berbeda saat salah satunya diubah; inilah akar selisih.
• **Pernyataan 2** ✓ Satu sumber logika menjamin pesanan dan laporan menghitung dengan cara yang sama.
• **Pernyataan 3** ✗ Jebakan: duplikasi memang hilang, tetapi berkas besar sulit dirawat, diuji, dan dikerjakan bersama.
• **Pernyataan 4** ✓ Karena kedua fitur memakai fungsi yang sama, satu kumpulan uji memeriksa keduanya.
• **Pernyataan 5** ✓ Dokumentasi antarmuka modul membantu anggota tim memakai fungsi sesuai aturannya.

Jawaban: pernyataan 1, 2, 4, 5.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 5 Kunci: B, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Desain modular yang efektif: setiap modul memiliki satu tugas, modul berkomunikasi lewat fungsi yang jelas (antarmuka), dan isi modul dapat diubah atau diganti tanpa mengubah modul lain selama antarmukanya tetap.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: tampak praktis, tetapi modul fisika jadi punya dua tugas dan terikat pada cara menggambar.

• **Pernyataan 2** ✓ Modul lain hanya memanggil simpan_skor(); isi di baliknya boleh berubah tanpa mengganggu mereka.

• **Pernyataan 3** ✗ Variabel global membuat modul saling bergantung dan kesalahan sulit dilacak.

• **Pernyataan 4** ✓ Modul yang terpisah dapat diuji sendiri.

• **Pernyataan 5** ✓ Pemisahan masukan dari logika membuat satu modul dapat diganti tanpa mengubah modul lain.

Jawaban: pernyataan 2, 4, 5.

Kuis Akhir Bab 4

No. 1 Kunci: A, C, E

Konsep: / menghasilkan bilangan pecahan (float), // pembagian bulat yang membulatkan ke bawah, % sisa bagi, dan ** pangkat.

Langkah:

1. `17 / 5` = 3.4.
2. `17 // 5` = 3 (3 × 5 = 15, sisa 2; hasil bagi dibulatkan ke bawah).
3. `17 % 5` = 2.
4. `17 ** 2` = 289.

Analisis setiap opsi:

• **A.** ✓ 17 ÷ 5 = 3.4.

• **B.** ✗ Jebakan: // membulatkan ke bawah, bukan ke terdekat (3.4 tidak menjadi 4).

• **C.** ✓ 17 = 3 × 5 + 2.

• **D.** ✗ Jebakan: ** adalah pangkat, bukan penyambungan; hasilnya 289.

• **E.** ✓ // menghasilkan 3.

Jawaban: A, C, dan E.

No. 2 Kunci: B

Konsep: Indeks string dimulai dari 0 dan irisan `[a:b]` mengambil indeks a sampai b − 1 (batas akhir tidak ikut). `lower()` mengubah ke huruf kecil dan `len()` menghitung banyak karakter.

Langkah:

1. Indeks: S=0, M=1, K=2, "-"=3, P=4, P=5, L=6, G=7, "-"=8, 2=9, 0=10, 2=11, 7=12.
2. `kode[4:8]` mengambil indeks 4, 5, 6, 7 = `"PPLG"`.
3. `bagian.lower()` = `"pplg"`; `len(bagian)` = 4; `str(4)` = `"4"`.
4. Keduanya disambung sebagai teks: `"pplg4"`.

Analisis setiap opsi:

• **A.** ✗ Jebakan: operator + menyambung teks tanpa spasi; spasi baru muncul bila memakai koma pada `print()`.

• **B.** ✓ Irisan `"PPLG"`, diubah kecil, lalu disambung angka 4 sebagai teks.

• **C.** ✗ Jebakan: ini hasil bila batas akhir dianggap ikut (indeks 4 sampai 8), yaitu `"PPLG-"` dengan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

panjang 5.

- **D. ✗** Jebakan: ini hasil irisan `[5:9]`, yaitu bila indeks dihitung mulai dari 1.
- **E. ✗** Jebakan: `len(bagian)` sudah diubah dengan `str()`, sehingga penyambungan sah.

Jawaban: B.

No. 3 Kunci: 1B 2B 3S

Konsep: `append` menambah di akhir, `pop(0)` mengambil elemen pertama, `pop()` mengambil elemen terakhir, `insert(i, v)` menyisipkan `v` pada indeks `i`.

Langkah (isi `antrean`):

1. Awal: A B C D.
2. `append("E")`: A B C D E.
3. `pop(0)`: x = A; sisa B C D E.
4. `pop()`: y = E; sisa B C D.
5. `insert(1, "Z")`: Z masuk pada indeks 1: B Z C D.
6. Keluaran: `A E ['B', 'Z', 'C', 'D']`.

Analisis:

- **1. ✓** x = "A".
- **2. ✓** y = "E".
- **3. ✗** Jebakan: `insert(1, "Z")` menyisipkan di indeks 1 (setelah B), bukan indeks 2; hasilnya B Z C D.

Jawaban: Benar, Benar, Salah.

No. 4 Kunci: A

Konsep: Tipe data menentukan arti operator. `int("12500")` mengubah teks menjadi bilangan bulat, sedangkan * pada teks (str) berarti mengulang teks tersebut.

Langkah:

1. `harga` berisi teks `"12500"`, `jumlah` berisi bilangan bulat 3.
2. Baris pertama: `int(harga)` menjadi 12500, dikali 3 = 37500.
3. Baris kedua: `harga * jumlah` mengulang teks "12500" sebanyak tiga kali menjadi `125001250012500`. Ini sah dan tidak menimbulkan galat.

Analisis setiap opsi:

- **A. ✓** Baris pertama menghitung, baris kedua mengulang teks.
- **B. ✗** Jebakan: teks yang dikalikan tidak otomatis menjadi angka.
- **C. ✗** Jebakan: `str * int` adalah operasi yang sah (pengulangan), bukan galat.
- **D. ✗** Jebakan: urutan keluaran terbalik.
- **E. ✗** Jebakan: baris pertama sudah mengonversi tipe sehingga tidak ada galat.

Jawaban: A.

No. 5 Kunci: C

Konsep: Indeks list dimulai dari 0. Indeks negatif menghitung dari belakang: `-1` adalah elemen terakhir.

Langkah:

1. Urutan: indeks 0 = apel, 1 = jeruk, 2 = mangga, 3 = pisang.
2. `buah[2]` = mangga.
3. `buah[-1]` = elemen terakhir = pisang.

Analisis setiap opsi:

- **A. ✗** Jebakan: jeruk adalah indeks 1 (hasil menghitung indeks dari 1), dan `-1` bukan elemen pertama.
- **B. ✗** Jebakan: `-1` adalah elemen terakhir, bukan pertama.
- **C. ✓** Mangga lalu pisang.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **D.** ✗ Jebakan: `buah[2]` adalah mangga, bukan jeruk.
- **E.** ✗ Jebakan: urutan terbalik.

Jawaban: C.

No. 6 Kunci: D

Konsep: Pada list dua dimensi, `papan[baris][kolom]` dengan indeks yang dimulai dari 0. Nomor baris dan kolom pada gambar dihitung mulai dari 1, jadi harus dikurangi 1.

Langkah:
1. Sel kuning berada pada baris ke-2 dan kolom ke-3 menurut gambar (mulai dari 1).
2. Indeks list: baris ke-2 → 1, kolom ke-3 → 2.
3. Ekspresinya `papan[1][2]` yang berisi `"O"`.

Analisis setiap opsi:
- **A.** ✗ Jebakan: memakai nomor gambar langsung (mulai dari 1); indeks 3 berada di luar jangkauan (IndexError).
- **B.** ✗ Jebakan: baris dan kolom tertukar dan memakai angka mulai dari 1; indeks 3 di luar jangkauan.
- **C.** ✗ Jebakan: `papan[2][1]` adalah baris ke-3, kolom ke-2, yaitu `"."`; urutan baris dan kolom dibalik.
- **D.** ✓ Baris indeks 1 dan kolom indeks 2.
- **E.** ✗ Jebakan: `papan[1][1]` adalah `"X"` di tengah papan.

Jawaban: D.

No. 7 Kunci: A, C, E

Konsep: Dictionary menyimpan pasangan kunci–nilai. Operator `in` memeriksa kunci. `del` menghapus pasangan; `d[kunci] = nilai` menambah atau memperbarui.

Langkah (keadaan `stok`):
1. Awal: kopi 12, teh 7, gula 20.
2. `stok["teh"] -= 3`: teh = 4.
3. `stok["susu"] = 5`: kunci baru susu = 5.
4. `del stok["gula"]`: gula dihapus. Sisa: kopi, teh, susu → `len` = 3.
5. `"gula" in stok` = False (sudah dihapus); `5 in stok` = False (5 adalah nilai, bukan kunci).

Analisis setiap opsi:
- **A.** ✓ Baris pertama mencetak `3 4`.
- **B.** ✗ Jebakan: `in` tidak memeriksa nilai.
- **C.** ✓ Keduanya False.
- **D.** ✗ Jebakan: kunci gula sudah dihapus, jadi tidak terhitung.
- **E.** ✓ Pernyataan itu menambah kunci baru.

Jawaban: A, C, dan E.

No. 8 Kunci: 1B 2S 3B

Konsep: Modularisasi yang efektif: tugas tunggal, kopling longgar (hubungan lewat parameter dan nilai kembalian), dan modul yang dapat diuji sendiri. Variabel global bersama mengikat modul satu dengan yang lain.

Langkah: Nilai tiap pernyataan terhadap prinsip modularisasi.

Analisis:
- **1.** ✓ Tugas tunggal dan data lewat parameter adalah ciri fungsi yang baik.
- **2.** ✗ Jebakan: variabel global bersama membuat perubahan di satu tempat memengaruhi tempat lain dan menyulitkan pelacakan.
- **3.** ✓ Modul yang diuji sendiri membantu melokalisasi kesalahan.

Jawaban: Benar, Salah, Benar.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 9 Kunci: B

Konsep: Fungsi dengan `return` mengembalikan nilai ke tempat pemanggilannya sehingga dapat dipakai dalam perhitungan.

Langkah:

1. `kuadrat(6)` = 36.
2. `kuadrat(2)` = 4.
3. `hasil` = 36 + 4 = 40.

Analisis setiap opsi:

- **A.** ✗ Jebakan: 36 hanya hasil `kuadrat(6)`.
- **B.** ✓ 36 + 4.
- **C.** ✗ Jebakan: 64 = $(6 + 2)^2$, yaitu menjumlahkan argumen sebelum dikuadratkan.
- **D.** ✗ Jebakan: 8 = 6 + 2, menjumlahkan argumen tanpa memanggil rumus.
- **E.** ✗ Jebakan: 38 = 36 + 2; `kuadrat(2)` bukan 2.

Jawaban: B.

No. 10 Kunci: E

Konsep: Pada rangkaian `if–elif–else`, program berhenti pada kondisi pertama yang benar dan melewati cabang sisanya.

Langkah:

1. `30 >= 35` salah.
2. `30 >= 28` benar → mencetak "hangat". Cabang `else` tidak dijalankan.

Analisis setiap opsi:

- **A.** ✗ Jebakan: 30 tidak memenuhi syarat 35.
- **B.** ✗ Jebakan: `else` hanya berjalan bila semua syarat di atasnya salah.
- **C.** ✗ Jebakan: hanya ada satu cabang yang berjalan, dan "panas sekali" tidak.
- **D.** ✗ Jebakan: `else` tidak berjalan setelah `elif` yang benar.
- **E.** ✓ Kondisi kedua benar.

Jawaban: E.

No. 11 Kunci: A, B, E

Konsep: Syarat `and` pada `while` harus benar seluruhnya agar perulangan berlanjut; cukup satu yang salah untuk berhenti. Telusuri dengan tabel jejak.

Langkah (tabel jejak):

putaran	genap?	nyawa	skor
awal		3	0
1	tidak	3	10
2	ya	2	10
3	tidak	2	20
4	ya	1	20
5	tidak	1	30
6	ya	0	30

Setelah putaran 6, `nyawa > 0` salah dan `putaran < 6` salah, sehingga perulangan berhenti. Keluaran: `6 0 30`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:
- **A.** ✓ Berhenti setelah putaran 6.
- **B.** ✓ Skor bertambah 10 pada putaran 1, 3, dan 5, sehingga akhirnya 30.
- **C.** ✗ Jebakan: nyawa berkurang pada putaran 2, 4, dan 6, sehingga akhirnya 0, bukan 1.
- **D.** ✗ Jebakan: skor tidak menjadi syarat perulangan.
- **E.** ✓ Nyawa 3 berkurang tiga kali (putaran 2, 4, 6) menjadi 0.

Jawaban: A, B, dan E.

No. 12 Kunci: A, C, D

Konsep: Variabel yang diberi nilai di dalam fungsi bersifat lokal, kecuali dinyatakan `global`. Variabel lokal dengan nama yang sama tidak mengubah variabel global.

Langkah:
1. `tambah(10)`: `total = n + 5` membuat variabel lokal bernama `total` (15) dan mengembalikannya ke `a`; `total` global tetap 100.
2. `reset()`: karena ada `global total`, penugasan `total = 0` mengubah variabel global menjadi 0.
3. `print(a, total)`: 15 dan 0.

Analisis setiap opsi:
- **A.** ✓ Keluaran 15 0.
- **B.** ✗ Jebakan: variabel di dalam `tambah` lokal; global tidak berubah.
- **C.** ✓ Penugasan di dalam fungsi tanpa `global` membuat variabel lokal.
- **D.** ✓ `global` membuat penugasan memengaruhi variabel global.
- **E.** ✗ Jebakan: tanpa `global`, `total = 0` hanya membuat variabel lokal; total global tetap 100.

Jawaban: A, C, dan D.

No. 13 Kunci: B

Konsep: Pada `if–elif`, hanya cabang pertama yang syaratnya benar yang dijalankan. Syarat harus disusun dari yang paling ketat ke yang paling longgar; bila tidak, cabang yang lebih spesifik tidak terjangkau.

Langkah:
1. `15 >= 6` benar → `kategori = "anak"`.
2. Cabang `elif` sisanya dilewati, sehingga keluaran "anak".
3. Syarat `>= 6` lebih longgar daripada `>= 13` dan `>= 18`, sehingga remaja dan dewasa tidak pernah tercapai. Perbaikan: periksa `>= 18` dulu, lalu `>= 13`, lalu `>= 6`.

Analisis setiap opsi:
- **A.** ✗ Jebakan: syarat pertama sudah benar untuk 15.
- **B.** ✓ Keluaran "anak", dan urutan syarat harus diperbaiki.
- **C.** ✗ Jebakan: cabang kedua tidak dijalankan karena cabang pertama sudah benar.
- **D.** ✗ Jebakan: pemeriksaan berhenti pada kondisi pertama yang benar.
- **E.** ✗ Jebakan: hanya satu cabang yang dijalankan pada rangkaian `elif`.

Jawaban: B.

No. 14 Kunci: A

Konsep: Pada perulangan bersarang, perulangan dalam berjalan penuh untuk setiap putaran perulangan luar. Jumlah putaran total dihitung dengan menjumlahkan banyak putaran dalam pada tiap putaran luar.

Langkah:
1. `i` bernilai 1, 2, 3, 4 (`range(1, 5)`).
2. Putaran dalam `range(i)`: untuk i = 1 sebanyak 1 kali, i = 2 sebanyak 2 kali, i = 3 sebanyak 3 kali, i = 4

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

sebanyak 4 kali.
3. Jumlah: 1 + 2 + 3 + 4 = 10.

Analisis setiap opsi:
- **A.** ✓ 1 + 2 + 3 + 4.
- **B.** ✗ Jebakan: 16 muncul bila perulangan dalam dianggap selalu 4 kali.
- **C.** ✗ Jebakan: 4 hanya banyak putaran luar.
- **D.** ✗ Jebakan: 6 = 1 + 2 + 3; putaran luar ke-4 terlupa.
- **E.** ✗ Jebakan: 15 = 1 + 2 + 3 + 4 + 5; `range(1, 5)` tidak mencakup 5.

Jawaban: A.

No. 15 Kunci: D

Konsep: `range(mulai, akhir, langkah)` menghasilkan bilangan dari `mulai` bertambah `langkah` sampai sebelum `akhir` (angka akhir tidak ikut).

Langkah:
1. Mulai 2, langkah 3: 2, 5, 8, 11, ...
2. Berhenti sebelum 11 (batas akhir 11 tidak ikut): 2, 5, 8.

Analisis setiap opsi:
- **A.** ✗ Jebakan: 11 sama dengan batas akhir sehingga tidak ikut.
- **B.** ✗ Jebakan: bilangan awal 2 ikut tercetak.
- **C.** ✗ Jebakan: ini `range(2, 11)` tanpa langkah 3.
- **D.** ✓ 2, 5, 8.
- **E.** ✗ Jebakan: ini deret kelipatan 3 yang mulai dari 3, bukan dari 2.

Jawaban: D.

No. 16 Kunci: E

Konsep: Pada `while`, syarat diperiksa di awal setiap putaran; perulangan berhenti ketika syarat menjadi salah. Telusuri dengan tabel jejak.

Langkah (tabel jejak):

Putaran	n	total
awal	0	0
1	1	2
2	2	6
3	3	12
4	4	20

Setelah putaran 4, `total < 20` salah (20 tidak kurang dari 20), sehingga perulangan berhenti. Keluaran: `4 20`.

Analisis setiap opsi:
- **A.** ✗ Jebakan: berhenti terlalu awal; total 12 masih kurang dari 20.
- **B.** ✗ Jebakan: satu putaran terlalu banyak; perulangan berhenti saat total mencapai 20.
- **C.** ✗ Jebakan: 10 adalah jumlah 1 + 2 + 3 + 4 tanpa dikalikan 2, padahal tiap putaran menambah `n * 2`.
- **D.** ✗ Jebakan: ini hasil bila `total += n`: total mencapai 21 pada n = 6.
- **E.** ✓ Perulangan berhenti saat total = 20.

Jawaban: E.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 17 Kunci: A, B, D

Konsep: Modul adalah berkas Python yang dapat dipakai berkas lain lewat `import`. Pemisahan membuat satu aturan ditulis di satu tempat dan setiap modul dapat diuji sendiri.

Langkah:

1. `hitung_pajak(50000)` = `round(50000 * 0.11)` = `round(5500.0)` = 5500.
2. `total_bayar(50000)` = 50000 + 5500 = 55500.
3. Karena tarif hanya ada di `pajak.py`, perubahan tarif cukup di sana.

Analisis setiap opsi:

- **A.** ✓ 55500.
- **B.** ✓ Satu aturan di satu tempat.
- **C.** ✗ Jebakan: `from pajak import hitung_pajak` sudah memungkinkan pemakaian tanpa menyalin.
- **D.** ✓ Fungsi pada `pajak.py` dapat dipanggil dan diuji tersendiri.
- **E.** ✗ Jebakan: `import` memakai isi modul lain; ia tidak menjadikan variabel berkas pemanggil global di modul itu.

Jawaban: A, B, dan D.

No. 18 Kunci: A

Konsep: Pada diagram alir, belah ketupat adalah percabangan: jawaban "ya" mengikuti satu jalur dan "tidak" jalur lain. Nilai batas harus diuji sesuai tanda perbandingannya (>= menyertakan nilai batas).

Langkah:

1. 75: `75 >= 75` ya → "Lulus".
2. 74: tidak lulus (74 < 75), lanjut `74 >= 60` ya → "Remedial".
3. 60: `60 >= 75` tidak, `60 >= 60` ya → "Remedial" (batas ikut).
4. 59: kedua syarat tidak terpenuhi → "Ulang".

Analisis setiap opsi:

- **A.** ✓ Lulus, Remedial, Remedial, Ulang.
- **B.** ✗ Jebakan: 74 belum memenuhi `>= 75`; hasilnya Remedial, bukan Lulus.
- **C.** ✗ Jebakan: 60 memenuhi `>= 60` karena batasnya ikut.
- **D.** ✗ Jebakan: 75 memenuhi syarat pertama, jadi hasilnya Lulus.
- **E.** ✗ Jebakan: 59 tidak memenuhi `>= 60`; hasilnya Ulang.

Jawaban: A.

No. 19 Kunci: C

Konsep: `print` menampilkan teks (efek samping), sedangkan `return` mengirim nilai keluar fungsi. Fungsi tanpa `return` menghasilkan `None`.

Langkah:

1. `sapa("Dewi")` mencetak `Halo Dewi` (baris pertama keluaran).
2. Fungsi tidak memiliki `return`, sehingga `h` berisi `None`.
3. `print(h)` mencetak `None`.

Analisis setiap opsi:

- **A.** ✗ Jebakan: `h` tidak berisi nama.
- **B.** ✗ Jebakan: pesan tidak tersimpan di `h`; hanya dicetak sekali.
- **C.** ✓ Cetakan di dalam fungsi, lalu None.
- **D.** ✗ Jebakan: baris "Halo Dewi" tetap tercetak saat fungsi dipanggil.
- **E.** ✗ Jebakan: nilai bawaan fungsi tanpa return bukan 0, melainkan None.

Jawaban: C.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 20 Kunci: A, C, D

Konsep: `continue` melompat ke putaran berikutnya; `break` menghentikan perulangan sepenuhnya.

Langkah (tabel jejak):

x	x % 3 == 0	x > 7	hasil
1	tidak	tidak	1
2	tidak	tidak	3
3	ya → continue		3
4	tidak	tidak	7
5	tidak	tidak	12
6	ya → continue		12
7	tidak	tidak	19
8	tidak	ya → break	19

Perulangan berhenti pada x = 8; x = 9 tidak diproses. Keluaran 19.

Analisis setiap opsi:

- **A.** ✓ Hasil akhir 19.
- **B.** ✗ Jebakan: 28 adalah 1 + 2 + ... + 7 tanpa memperhitungkan `continue` pada 3 dan 6.
- **C.** ✓ 3 dan 6 dilewati.
- **D.** ✓ Pada x = 8, syarat `x > 7` benar.
- **E.** ✗ Jebakan: pada x = 3 yang dijalankan adalah `continue`, bukan `break`.

Jawaban: A, C, dan D.

No. 21 Kunci: C

Konsep: Kasus uji dihitung dari aturan bisnis, bukan dari kode. Kesalahan yang paling sering terjadi ada pada nilai batas: `>` dan `>=` yang tertukar.

Langkah:

1. Bandingkan kolom "diharapkan" dengan "hasil program" tiap baris: hanya kasus 3 (belanja 200.000) yang berbeda (diharapkan 10, hasil 5).
2. Pada 200.000, `200000 > 200000` salah, sehingga program jatuh ke `elif` dan memberi 5.
3. Aturan meminta "200.000 atau lebih", jadi syaratnya harus `>=`.

Analisis setiap opsi:

- **A.** ✗ Jebakan: satu kasus yang gagal sudah membuktikan ada kesalahan.
- **B.** ✗ Jebakan: kasus 2 (100.000) sudah sesuai; syarat kedua benar.
- **C.** ✓ Nilai batas 200.000 terlewat karena memakai `>`.
- **D.** ✗ Jebakan: aturan bisnis menyebut 200.000, bukan 250.000.
- **E.** ✗ Jebakan: kasus 250.000 berguna untuk memastikan cabang pertama bekerja pada nilai di atas batas.

Jawaban: C.

No. 22 Kunci: D

Konsep: Argumen yang tidak diberikan memakai nilai bawaan. Fungsi yang memanggil fungsi lain meneruskan hanya argumen yang ditulis pada pemanggilan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah:

1. `total(1000, 2000)`: `persen` memakai nilai bawaan 5 (untuk `total`).
2. `diskon(a, persen)` = `diskon(1000, 5)` = 1000 − 1000 × 5 // 100 = 1000 − 50 = 950.
3. `diskon(b)` = `diskon(2000)` tidak meneruskan `persen`, sehingga memakai nilai bawaan `diskon` yaitu 10: 2000 − 2000 × 10 // 100 = 2000 − 200 = 1800.
4. Jumlah: 950 + 1800 = 2750.

Analisis setiap opsi:

- **A.** ✗ Jebakan: 2850 = 950 + 1900, yaitu dengan anggapan keduanya memakai persen 5.
- **B.** ✗ Jebakan: 2700 = 900 + 1800, yaitu dengan anggapan keduanya memakai persen 10.
- **C.** ✗ Jebakan: 3000 adalah jumlah harga tanpa diskon.
- **D.** ✓ 950 + 1800.
- **E.** ✗ Jebakan: hasil tebakan yang tidak dihitung dari fungsi.

Jawaban: D.

No. 23 Kunci: 1B 2S 3B

Konsep: `range(20, 0, -6)` dimulai dari 20, berkurang 6 setiap kali, dan berhenti sebelum mencapai batas akhir (0), yang tidak ikut. Pada langkah negatif, batas akhir juga tidak ikut.

Langkah:

1. Deret: 20, 14, 8, 2; berikutnya −4 sudah di bawah batas 0, sehingga berhenti.
2. Pada `range(20, 2, -6)` deretnya 20, 14, 8; angka 2 sama dengan batas akhir sehingga tidak ikut.
3. Jumlah putaran = banyak angka: 4.

Analisis:

- **1.** ✓ Keluaran 20 14 8 2.
- **2.** ✗ Jebakan: batas akhir tidak ikut; angka 2 tidak tercetak.
- **3.** ✓ Ada empat nilai (20, 14, 8, 2).

Jawaban: Benar, Salah, Benar.

No. 24 Kunci: E

Konsep: Desain modular yang efektif: tugas tunggal per modul, tidak ada duplikasi (satu aturan ditulis di satu tempat), hubungan antarmodul lewat parameter dan `return`, dan modul dapat diuji sendiri.

Langkah:

1. Masalahnya: rumus yang sama ada di empat tempat (duplikasi).
2. Pilih rancangan yang menempatkan rumus di satu tempat dan membuat bagian lain memanggilnya.

Analisis setiap opsi:

- **A.** ✗ Jebakan: menyalin rumus mempertahankan duplikasi; perubahan harus dilakukan di empat tempat.
- **B.** ✗ Jebakan: satu berkas besar tidak menghilangkan pengulangan rumus dan menyulitkan perawatan.
- **C.** ✗ Jebakan: variabel global yang diubah dari mana saja menyulitkan pelacakan kesalahan.
- **D.** ✗ Jebakan: menyalin manual rawan lupa dan menimbulkan kesalahan.
- **E.** ✓ Satu fungsi pada modul tersendiri; perubahan aturan cukup di satu tempat.

Jawaban: E.

No. 25 Kunci: A, C, E

Konsep: `and` benar bila kedua sisi benar; `or` benar bila salah satu sisi benar. Nilai batas ditentukan tanda >= (ikut) dan < (tidak ikut).

Langkah (nilai syarat):

1. `(jam >= 8 and jam < 17)` benar untuk jam 8 sampai 16; salah untuk 17 dan seterusnya, dan

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

salah untuk jam di bawah 8.
2. Bila syarat itu salah, `petugas` yang bernilai True tetap membuat seluruh kondisi benar (karena `or`).
3. Hasil: `boleh_masuk(8, False)` = True; `boleh_masuk(17, False)` = False; `boleh_masuk(20, True)` = True; `boleh_masuk(7, False)` = False; `boleh_masuk(16, False)` = True.

Analisis setiap opsi:
- **A.** ✓ `8 >= 8` benar (batas bawah ikut) dan `8 < 17` benar.
- **B.** ✗ Jebakan: `jam < 17` salah untuk 17 (batas atas tidak ikut) sehingga hasilnya False.
- **C.** ✓ Dengan `or petugas`, petugas selalu boleh masuk.
- **D.** ✗ Jebakan: `7 >= 8` salah, dan pengunjung bukan petugas, sehingga hasilnya False.
- **E.** ✓ 16 masih kurang dari 17.

Jawaban: A, C, dan E.

Latihan sub bab 5.1

No. 1 Kunci: A

Konsep: Class adalah rancangan (cetakan) yang menentukan atribut (data) dan method (tindakan). Objek adalah wujud nyata class; satu class dapat menghasilkan banyak objek dengan nilai atribut berbeda.

Langkah:
1. Mobil = class (cetakan).
2. warna, kecepatan_maks, bensin = atribut; gas(), rem(), isi_bensin() = method.
3. mobil_dina dan mobil_raka = dua objek dari class yang sama dengan nilai atribut berbeda.

Analisis setiap opsi:
- **A.** ✓ Dua objek dibuat dari satu class Mobil dengan nilai warna dan kecepatan berbeda.
- **B.** ✗ warna dan kecepatan_maks adalah data, jadi atribut, bukan method.
- **C.** ✗ Jebakan: gas() memang mengubah nilai kecepatan, tetapi gas() sendiri adalah tindakan, jadi method.
- **D.** ✗ Kebalikannya: Mobil adalah class, sedangkan mobil_dina dan mobil_raka adalah objeknya.
- **E.** ✗ Perbedaan nilai atribut tidak memerlukan class baru; cukup objek yang berbeda.

Jawaban: A.

No. 2 Kunci: C

Konsep: Setiap objek memiliki salinan atributnya sendiri. Method yang dipanggil pada satu objek hanya mengubah atribut objek itu.

Langkah:
1. b1.stok = 2, b2.stok = 1.
2. b1.pinjam() → b1.stok = 1.
3. b2.pinjam() → b2.stok = 0.
4. b2.pinjam() lagi → 0 > 0 salah → stok tetap 0.
5. Keluaran: 1 0.

Analisis setiap opsi:
- **A.** ✗ Jebakan: syarat if diabaikan sehingga stok b2 menjadi −1.
- **B.** ✗ b1 hanya dipinjam sekali, jadi stoknya 1, bukan 0.
- **C.** ✓ b1 berkurang satu kali; b2 berkurang sekali lalu ditahan if.
- **D.** ✗ Method pinjam() memang mengubah atribut stok objeknya.
- **E.** ✗ Jebakan: pemanggilan b2 dianggap juga mengurangi b1, dan if diabaikan.

Jawaban: C.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 3 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Prinsip dasar PBO: enkapsulasi (menyembunyikan data dan mengubahnya lewat method), pewarisan (subclass mendapat atribut dan method superclass), polimorfisme (satu nama method dengan perilaku berbeda), dan abstraksi (menampilkan yang penting saja).

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: membuat banyak objek dari satu class adalah pembuatan objek (instansiasi), bukan pewarisan antar-class.

• **Pernyataan 2** ✓ Data disembunyikan dan hanya diubah lewat method yang memeriksa aturan → enkapsulasi.

• **Pernyataan 3** ✓ Subclass mendapat atribut dari superclass → pewarisan.

• **Pernyataan 4** ✗ Pemeriksaan aturan di dalam method adalah bagian dari enkapsulasi, bukan polimorfisme.

• **Pernyataan 5** ✓ Satu nama method dengan perilaku berbeda menurut class objeknya → polimorfisme.

Jawaban: pernyataan 2, 3, 5.

No. 4 Kunci: A

Konsep: Pewarisan dan polimorfisme: superclass menetapkan method bersama, setiap subclass menulis ulang (override) perilakunya. Kode utama cukup memanggil hewan.gerak() tanpa if–elif, sehingga jenis baru ditambah sebagai class baru tanpa mengubah kode lama.

Langkah:

1. Akar masalah: setiap jenis baru memaksa mengubah fungsi yang dipakai semua hewan.
2. Dengan subclass per jenis, perilaku setiap hewan tersimpan di class-nya sendiri.
3. Menambah jenis baru = menambah subclass baru; class hewan lama tidak tersentuh.

Analisis setiap opsi:

• **A.** ✓ Jenis baru cukup ditambah sebagai subclass; kode hewan lama tidak diubah.

• **B.** ✗ Ini cara yang sedang dipakai dan justru penyebab masalah.

• **C.** ✗ Jebakan: sudah memakai class, tetapi if–elif tetap ada dan tetap harus diubah setiap ada jenis baru.

• **D.** ✗ Pemanggil harus memilih fungsi yang tepat dengan if–elif lagi, dan kode yang mirip tersalin berkali-kali.

• **E.** ✗ Variabel global membuat semua bagian program saling bergantung dan bertentangan dengan enkapsulasi.

Jawaban: A.

No. 5 Kunci: B

Konsep: Abstraksi menampilkan hanya bagian penting (method sederhana seperti putar()) dan menyembunyikan detail rumit di dalam class. Pemakai class cukup tahu apa yang dilakukan method, bukan bagaimana caranya.

Langkah:

1. Detail rumit (membaca berkas, volume, kanal) tersembunyi di dalam class.
2. Yang ditampilkan hanya putar(nama) → abstraksi.

Analisis setiap opsi:

• **A.** ✗ Pewarisan adalah hubungan antar-class (superclass–subclass); di kasus ini tidak ada class turunan.

• **B.** ✓ Detail disembunyikan dan pemakai cukup memanggil putar().

• **C.** ✗ Jebakan: argumen yang berbeda pada satu method bukan polimorfisme; polimorfisme berarti

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

method bernama sama berperilaku berbeda menurut class objeknya.

• **D.** ✗ Instansiasi memang terjadi, tetapi itu proses membuat objek, bukan prinsip yang digambarkan kasus.

• **E.** ✗ Memakai satu class di banyak tempat bukan pewarisan.

Jawaban: B.

Latihan sub bab 5.2

No. 1 Kunci: B, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Enkapsulasi: atribut privat (awalan __) hanya diubah lewat setter yang memeriksa aturan. Setter dapat mengembalikan True/False untuk memberi tahu apakah nilai diterima. Getter hanya membaca nilai.

Langkah:

1. s dibuat → __suhu = 25.
2. set_suhu(75): 0 <= 75 <= 60 salah → return False; __suhu tetap 25.
3. hasil = False; get_suhu() = 25.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: 75 tidak memenuhi 0 <= nilai <= 60, jadi atribut tidak diubah.

• **Pernyataan 2** ✓ set_suhu(75) menolak nilai dan mengembalikan False; suhu tetap nilai awal 25.

• **Pernyataan 3** ✗ Jebakan: syaratnya dua sisi; 0 <= −5 salah, sehingga nilai ditolak dan hasilnya False.

• **Pernyataan 4** ✗ Method yang hanya mengembalikan nilai adalah getter.

• **Pernyataan 5** ✓ Syarat memakai <=, sehingga 60 memenuhi dan suhu menjadi 60.

Jawaban: pernyataan 2, 5.

No. 2 Kunci: 1S 2B 3B

Konsep: Enkapsulasi: atribut penting dibuat privat (di Python ditandai awalan __), dibaca lewat getter, dan diubah lewat setter/method pengubah yang memeriksa aturan. Koreksi pun sebaiknya melalui method khusus yang tercatat.

Analisis setiap pernyataan:

• **Pernyataan 1 → Salah.** Jebakan: koreksi memang lebih cepat, tetapi atribut publik dapat diubah dari mana saja tanpa pemeriksaan, sehingga aturan "tidak negatif" dan "tercatat" mudah dilanggar.

• **Pernyataan 2 → Benar.** Method pengubah dengan validasi adalah satu-satunya jalan mengubah atribut privat, sehingga aturan selalu diperiksa.

• **Pernyataan 3 → Benar.** Getter hanya mengembalikan nilai, sesuai kebutuhan laporan yang hanya membaca.

Jawaban: Salah, Benar, Benar.

Latihan sub bab 5.3

No. 1 Kunci: B

Konsep: Subclass mewarisi konstruktor dan method superclass. Bila subclass menulis ulang method, versinya yang dipakai; di dalamnya `super().bonus()` tetap dapat memanggil versi superclass.

Langkah:

1. Manajer tidak menulis __init__ → memakai milik Pegawai: nama "Sinta", gaji 6.000.000.
2. m.bonus() → super().bonus() = 6.000.000 // 10 = 600.000, ditambah 500.000 = 1.100.000.
3. p.bonus() → 4.000.000 // 10 = 400.000.
4. Jumlah: 1.100.000 + 400.000 = 1.500.000.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

- **A.** ✗ Jebakan: bonus Manajer dianggap hanya 500.000 karena method induk diganti seluruhnya: 500.000 + 400.000.
- **B.** ✓ Bonus Manajer 1.100.000 ditambah bonus Doni 400.000.
- **C.** ✗ Tunjangan 500.000 terlupa: 600.000 + 400.000.
- **D.** ✗ Jebakan: tunjangan dianggap ikut diterima Doni: 1.100.000 + 900.000.
- **E.** ✗ Manajer mewarisi __init__ dari Pegawai, jadi objek dapat dibuat.

Jawaban: B.

No. 2 Kunci: A

Konsep: Pada pewarisan bertingkat, subclass mewarisi semua atribut dan method superclass beserta superclass di atasnya. Method yang ditulis ulang (override) di subclass menggantikan versi superclass untuk objek subclass itu. Superclass tidak mewarisi milik subclass.

Langkah:

1. SepedaListrik mewarisi Sepeda, dan Sepeda mewarisi Kendaraan.
2. Atribut sl: nama, kecepatan (Kendaraan) + gigi (Sepeda) + baterai (sendiri).
3. Method sl: gerak() (Kendaraan), ganti_gigi() (Sepeda), isi_daya() dan info() versi sendiri.

Analisis setiap opsi:

- **A.** ✓ Atribut diwarisi dari Kendaraan dan Sepeda, ditambah baterai miliknya.
- **B.** ✗ Jebakan: SepedaListrik adalah subclass Sepeda, jadi ganti_gigi() ikut diwarisi.
- **C.** ✗ Jebakan: SepedaListrik menulis ulang info(), sehingga versi SepedaListrik yang dijalankan.
- **D.** ✗ Pewarisan berjalan dari atas ke bawah; Sepeda tidak memiliki isi_daya() milik subclass-nya.
- **E.** ✗ gerak() diwarisi dari Kendaraan walaupun tidak tertulis di kotak SepedaListrik.

Jawaban: A.

Latihan sub bab 5.4

No. 1 Kunci: B, C, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Polimorfisme: objek-objek dari subclass berbeda dapat diperlakukan sama (dipanggil n.kirim()), tetapi masing-masing menjalankan versi method miliknya. Subclass yang tidak menulis ulang method memakai versi superclass.

Langkah:

1. Email() → "[email] lunas".
2. WhatsApp() → "[wa] LUNAS".
3. Notifikasi() → "[umum] lunas".

Analisis setiap pernyataan:

- **Pernyataan 1** ✗ Jebakan: Email dan WhatsApp menulis ulang kirim(), sehingga versi milik subclass yang dijalankan.
- **Pernyataan 2** ✓ Objek kedua WhatsApp: "[wa] " + "lunas".upper() = "[wa] LUNAS".
- **Pernyataan 3** ✓ Objek pertama Email memakai kirim() miliknya sendiri.
- **Pernyataan 4** ✓ Perulangan hanya memanggil n.kirim(); objek baru cukup memiliki method dengan nama yang sama.
- **Pernyataan 5** ✓ Tanpa overriding, Email memakai kirim() warisan dari Notifikasi.

Jawaban: pernyataan 2, 3, 4, 5.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 2 Kunci: C

Konsep: Polimorfisme: pemanggilan u.luas() menjalankan versi method milik class objeknya. Segitiga mewarisi __init__ dari Ubin tetapi menulis ulang luas(). Operator // membulatkan ke bawah.

Langkah:

1. Ubin(4).luas() = 4 × 4 = 16.
2. Segitiga(4).luas() = 16 // 2 = 8.
3. Segitiga(3).luas() = 9 // 2 = 4.
4. Total = 16 + 8 + 4 = 28.

Analisis setiap opsi:

- **A.** ✗ Jebakan: semua objek dianggap memakai luas() milik Ubin: 16 + 16 + 9.
- **B.** ✗ Jebakan: // dianggap pembagian biasa sehingga 9 / 2 = 4,5.
- **C.** ✓ 16 + 8 + 4 = 28.
- **D.** ✗ Jebakan: 4,5 dibulatkan ke atas menjadi 5.
- **E.** ✗ Segitiga mewarisi __init__ dari Ubin sehingga objek dapat dibuat.

Jawaban: C.

Kuis Akhir Bab 5

No. 1 Kunci: A, C, D

Konsep: Method dapat mengubah keadaan (atribut) objek dan/atau mengembalikan nilai. Tiap pemanggilan `Playlist()` membuat objek baru dengan keadaan awal sendiri.

Langkah (isi `p.lagu`):

1. `p.tambah("A")`: lagu = ["A"], mengembalikan 1 → `n` = 1.
2. `p.tambah("B")`: lagu = ["A", "B"] (nilai kembalian 2 tidak disimpan).
3. `p.hapus_pertama()`: `pop(0)` mengambil "A" → `x` = "A"; lagu = ["B"].
4. `Playlist().hapus_pertama()`: objek baru kosong → `None`.
5. Keluaran: `1 A None ['B']`.

Analisis setiap opsi:

- **A.** ✓ Nilai kembalian tambah() pertama adalah 1.
- **B.** ✗ Jebakan: `y` berasal dari objek baru yang kosong, bukan dari `p`.
- **C.** ✓ Objek baru kosong → `None`.
- **D.** ✓ Setelah "A" dihapus, tersisa "B".
- **E.** ✗ Jebakan: `pop(0)` mengambil elemen pertama, "A".

Jawaban: A, C, dan D.

No. 2 Kunci: B

Konsep: Atribut class (`jumlah`) dibagi semua objek dan bertambah setiap objek dibuat. Penugasan `c = a` tidak membuat objek baru: `c` dan `a` menunjuk objek yang sama.

Langkah:

1. `Meja("M1")` → jumlah = 1; `Meja("M2")` → jumlah = 2.
2. `c = a`: tidak ada objek baru (jumlah tetap 2).
3. `c.kode = "M3"` mengubah objek yang sama dengan `a`: `a.kode` = "M3".
4. `b.kode` tetap "M2". Keluaran: `2 M3 M2`.

Analisis setiap opsi:

- **A.** ✗ Jebakan: `c = a` dikira menyalin, sehingga `a.kode` tidak berubah.
- **B.** ✓ Dua objek dibuat; `a` dan `c` satu objek.
- **C.** ✗ Jebakan: `c = a` dikira membuat objek ketiga.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **D.** ✗ Jebakan: `b` tidak ikut berubah.
- **E.** ✗ Jebakan: kedua kesalahan pada A dan C terjadi sekaligus.

Jawaban: B.

No. 3 Kunci: 1B 2S 3B

Konsep: Abstraksi menyembunyikan rincian rumit di balik antarmuka sederhana. Enkapsulasi melindungi data dari perubahan langsung. Pewarisan terjadi bila sebuah class mengambil anggota class lain.

Langkah: Tanyakan pada tiap pernyataan: apa yang dilakukan rancangannya (menyembunyikan, melindungi, mewarisi, atau menjawab berbeda)?

Analisis:

- **1.** ✓ Memanggil `terjemah` tanpa tahu rincian = abstraksi.
- **2.** ✗ Jebakan: "banyak bagian memakai class" bukan pewarisan; pewarisan membutuhkan class yang mewarisi class lain.
- **3.** ✓ Data dilindungi dan diubah lewat method yang memeriksa = enkapsulasi.

Jawaban: Benar, Salah, Benar.

No. 4 Kunci: B

Konsep: Setter yang baik memeriksa nilai sebelum mengubah atribut privat. Nilai yang ditolak tidak mengubah atribut. Perhatikan nilai batas: pada `0 <= v <= 100`, 0 dan 100 diterima.

Langkah (nilai `__volume`):

1. `set_volume(120)`: `0 <= 120 <= 100` salah → False; volume tetap 30.
2. `set_volume(100)`: batas atas ikut → volume 100, True.
3. `set_volume(-1)`: salah → False; volume tetap 100.
4. Keluaran: `False True False 100`.

Analisis setiap opsi:

- **A.** ✗ Jebakan: 120 melewati batas dan ditolak, sehingga bukan True.
- **B.** ✓ 120 ditolak, 100 diterima, -1 ditolak, volume akhir 100.
- **C.** ✗ Jebakan: volume akhir 30 muncul bila 100 dikira ditolak.
- **D.** ✗ Jebakan: 120 dan -1 dikira diterima.
- **E.** ✗ Jebakan: nilai batas 100 dikira ditolak.

Jawaban: B.

No. 5 Kunci: A, C, D

Konsep: Empat prinsip dasar OOP: abstraksi, enkapsulasi (atribut privat), pewarisan (subclass), dan polimorfisme (satu pemanggilan, perilaku menurut jenis objek).

Langkah:

1. `__kode` privat dan dipakai lewat `info()`: enkapsulasi.
2. `class Lemari(Perabot)`: pewarisan.
3. `info()` pada `Lemari` ditulis ulang (overriding), sehingga `m.info()` memberi hasil berbeda untuk tiap objek: polimorfisme.
4. Keluaran: `Perabot P-01` lalu `Perabot P-01 (lemari)`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:
- **A.** ✓ Atribut privat dengan akses lewat method.
- **B.** ✗ Jebakan: subclass boleh menulis ulang method; hasil baris kedua memuat tambahan "(lemari)".
- **C.** ✓ Tanda `(Perabot)` menyatakan pewarisan.
- **D.** ✓ Hasil berbeda untuk objek berbeda.
- **E.** ✗ Jebakan: pewarisan ditentukan oleh definisi class, bukan oleh isi list.

Jawaban: A, C, dan D.

No. 6 Kunci: E

Konsep: Pewarisan: subclass mengambil atribut dan method superclass (panah dengan segitiga kosong). Polimorfisme: satu nama method (`luas()`) menghasilkan perilaku berbeda menurut jenis objek.

Langkah:
1. Panah dari `Persegi` dan `Lingkaran` ke `Bentuk` menandakan pewarisan.
2. `luas()` ada di ketiga class dan masing-masing menghitung dengan caranya sendiri (overriding), ciri polimorfisme.
3. Gambar tidak memperlihatkan atribut privat atau rincian yang disembunyikan, sehingga enkapsulasi dan abstraksi tidak tampak jelas.

Analisis setiap opsi:
- **A.** ✗ Jebakan: tidak ada tanda atribut privat maupun penyembunyian rincian.
- **B.** ✗ Jebakan: pewarisan benar, tetapi tidak ada penyembunyian rincian rumit.
- **C.** ✗ Jebakan: polimorfisme benar, tetapi tidak tampak perlindungan atribut.
- **D.** ✗ Jebakan: polimorfisme benar, tetapi abstraksi tidak tampak.
- **E.** ✓ Panah pewarisan dan `luas()` pada ketiga class.

Jawaban: E.

No. 7 Kunci: A, C, D

Konsep: Atribut publik dapat diubah langsung dari luar class. Atribut privat dilindungi: penugasan `k.__sisa = 99` dari luar tidak mengubah atribut privat yang dipakai method; ia hanya membuat atribut lain bernama sama pada objek.

Langkah (nilai `__sisa`):
1. `pakai(2)`: 2 ≤ 3 → sisa 1, True.
2. `pakai(2)`: 2 ≤ 1 salah → False.
3. `pakai(0)`: `n > 0` salah → False.
4. `lihat_sisa()` = 1. Baris pertama: `True False False 1`.
5. `k.kode = "BARU"` mengubah atribut publik. `k.__sisa = 99` tidak memengaruhi sisa sebenarnya, sehingga `lihat_sisa()` tetap 1. Baris kedua: `1 BARU`.

Analisis setiap opsi:
- **A.** ✓ True False False 1.
- **B.** ✗ Jebakan: sisa privat tidak berubah; penugasan dari luar hanya membuat atribut lain.
- **C.** ✓ Sisa tetap 1 dan kode menjadi BARU.
- **D.** ✓ Atribut publik tanpa perlindungan.
- **E.** ✗ Jebakan: syarat `n > 0` salah untuk 0, sehingga tidak ada pengurangan.

Jawaban: A, C, dan D.

No. 8 Kunci: A

Konsep: Setiap objek memiliki atribut instansnya sendiri. Mengubah atribut satu objek tidak memengaruhi objek lain dari class yang sama.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah:
1. `s1.ukuran` = 40; `s2.ukuran` = 42.
2. `s2.ukuran -= 1` hanya mengubah milik `s2`: 41.
3. Keluaran: `40 41`.

Analisis setiap opsi:
- **A.** ✓ Hanya `s2` yang berubah.
- **B.** ✗ Jebakan: perubahan `s2` terlewat.
- **C.** ✗ Jebakan: `s1` ikut berkurang, padahal atributnya terpisah.
- **D.** ✗ Jebakan: `s1` berubah dan `s2` tidak, kebalikan dari yang terjadi.
- **E.** ✗ Jebakan: dua objek dikira berbagi satu atribut.

Jawaban: A.

No. 9 Kunci: C

Konsep: Agar sebuah nilai menjadi atribut objek, ia harus ditugaskan ke `self.nama_atribut`. Penugasan ke variabel biasa di dalam `__init__` hanya membuat variabel lokal yang hilang setelah method selesai.

Langkah:
1. Di `__init__`, `merek = merek` hanya menugaskan parameter ke dirinya sendiri; tidak ada `self.merek`.
2. `p.aktif` = True setelah `hidupkan()`.
3. `p.merek` tidak ada → `AttributeError` muncul sebelum `print` sempat mencetak.

Analisis setiap opsi:
- **A.** ✗ Jebakan: parameter tidak otomatis menjadi atribut.
- **B.** ✗ Jebakan: atribut yang tidak pernah dibuat tidak bernilai `None`, melainkan tidak ada.
- **C.** ✓ Atribut `merek` tidak pernah dibuat.
- **D.** ✗ Jebakan: method tanpa `return` sah dan tidak menimbulkan TypeError.
- **E.** ✗ Jebakan: `hidupkan()` mengubah `aktif` menjadi True.

Jawaban: C.

No. 10 Kunci: D

Konsep: Enkapsulasi membungkus data dan method dalam satu class serta melindungi data dari perubahan langsung (atribut privat berawalan `__`, akses lewat method).

Langkah:
1. `__suhu` adalah atribut privat.
2. Pemakai hanya dapat membacanya lewat `lihat_suhu()`.
3. Itu ciri enkapsulasi. Keluaran program: 4.

Analisis setiap opsi:
- **A.** ✗ Jebakan: abstraksi menyembunyikan rincian rumit, sedangkan di sini tidak ada perhitungan rumit.
- **B.** ✗ Jebakan: tidak ada tanda `(ClassInduk)`; tidak ada pewarisan.
- **C.** ✗ Jebakan: hanya ada satu method dengan satu bentuk keluaran.
- **D.** ✓ Data dilindungi dan diakses lewat method.
- **E.** ✗ Jebakan: perlindungan atribut adalah penerapan prinsip OOP.

Jawaban: D.

No. 11 Kunci: A

Konsep: Atribut berawalan dua garis bawah (`__isi`) bersifat privat: tidak dapat diakses langsung dari luar class dengan nama itu; percobaan akses menimbulkan `AttributeError`.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah: `print(r.__isi)` berada di luar class, sehingga atribut privat tidak ditemukan dengan nama itu → `AttributeError`.

Analisis setiap opsi:
- **A.** ✓ Atribut privat tidak dapat dibaca langsung dari luar.
- **B.** ✗ Jebakan: atribut privat dikira dapat dibaca langsung.
- **C.** ✗ Jebakan: tidak ada nilai `None`; programnya galat.
- **D.** ✗ Jebakan: tidak ada nilai bawaan 0.
- **E.** ✗ Jebakan: jenis galatnya AttributeError, bukan TypeError.

Jawaban: A.

No. 12 Kunci: C

Konsep: Method berawalan dua garis bawah bersifat privat: hanya dapat dipanggil dari dalam class. Method publik boleh memanggilnya.

Langkah:
1. `p.kirim(" halo ")`: di dalam class, `self.__siapkan` dipanggil → `strip()` menghapus spasi, `upper()` mengubah ke huruf besar → "HALO". Baris pertama: `HALO`.
2. `p.__siapkan("x")` dipanggil dari luar class → `AttributeError`.

Analisis setiap opsi:
- **A.** ✗ Jebakan: method privat dikira dapat dipanggil dari luar.
- **B.** ✗ Jebakan: memanggil method privat lewat objek dari luar class menghasilkan `AttributeError`, bukan `NameError` (yang muncul bila nama variabel belum didefinisikan).
- **C.** ✓ Baris pertama berhasil; baris kedua galat.
- **D.** ✗ Jebakan: baris pertama sudah tercetak sebelum galat muncul.
- **E.** ✗ Jebakan: panggilan privat dari luar tidak mengembalikan None, melainkan galat.

Jawaban: C.

No. 13 Kunci: A, C, D

Konsep: Bila subclass menulis `__init__` sendiri, `__init__` induk tidak berjalan otomatis. Atribut milik induk hanya terisi bila `super().__init__(...)` dipanggil dengan argumen yang sesuai.

Langkah:
1. `__init__` milik `BarangImpor` hanya menugaskan `self.bea`. `print(b.bea)` → 50.
2. `nama` dan `harga` tidak pernah diisi, sehingga `print(b.nama)` → `AttributeError`.
3. Perbaikan: tambahkan `super().__init__(nama, harga)`.

Analisis setiap opsi:
- **A.** ✓ `bea` terisi 50.
- **B.** ✗ Jebakan: atribut induk tidak diisi otomatis bila subclass punya `__init__` sendiri.
- **C.** ✓ `nama` tidak pernah dibuat.
- **D.** ✓ Memanggil konstruktor induk dengan argumen yang cocok.
- **E.** ✗ Jebakan: menghapus `bea` tidak mengisi `nama` dan `harga`.

Jawaban: A, C, dan D.

No. 14 Kunci: D

Konsep: Pada perulangan polimorfik, tiap objek memakai versi method milik class-nya. `super()` naik satu tingkat ke class induk langsung dan dapat berantai.

Langkah (tabel jejak):

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

objek	versi yang dipakai	ongkos	total
Paket(2)	Paket	2 × 1000 = 2000	2000
Kilat(3)	Kilat → super(): Paket	3000 + 5000 = 8000	10000
Gratis(4)	Gratis	0	10000
Fragil(1)	Fragil → super(): Kilat → super(): Paket	(1000 + 5000) × 2 = 12000	22000

Analisis setiap opsi:

- **A.** ✗ Jebakan: 16000 muncul bila pengali 2 pada Fragil terlupa (6000).
- **B.** ✗ Jebakan: 11000 muncul bila Fragil dikira memakai versi Paket saja (1000).
- **C.** ✗ Jebakan: 15000 muncul bila versi Kilat dan Gratis dikira sama dengan Paket.
- **D.** ✓ 2000 + 8000 + 0 + 12000.
- **E.** ✗ Jebakan: 26000 muncul bila Gratis dikira memakai versi Paket (4000).

Jawaban: D.

No. 15 Kunci: 1B 2S 3S

Konsep: Pewarisan dipakai untuk hubungan "adalah" (is-a). Hubungan "memiliki" (has-a) dibuat dengan atribut yang berisi objek lain.

Langkah:

1. Senter memiliki baterai (bagian), bukan sejenis baterai.
2. `self.baterai` adalah atribut; `s.baterai.daya` → 1200. `isinstance(s, Baterai)` → False karena Senter bukan subclass Baterai.

Analisis:

- **1.** ✓ Has-a diwujudkan dengan atribut.
- **2.** ✗ Jebakan: memakai objek tidak membuat objek itu menjadi bagian dari jenisnya; hasilnya False.
- **3.** ✗ Jebakan: senter bukan sejenis baterai; hubungannya "memiliki", bukan "adalah".

Jawaban: Benar, Salah, Salah.

No. 16 Kunci: A

Konsep: Pewarisan bertingkat: sebuah class memiliki anggotanya sendiri ditambah seluruh anggota dari semua class di atasnya (induk, induk dari induk, dan seterusnya).

Langkah:

1. `Ensiklopedia` memiliki `jilid` dan `cari()` sendiri.
2. Induk langsungnya `Buku`: `penulis` dan `baca()`.
3. Induk dari `Buku` yaitu `Koleksi`: `judul` dan `info()`.
4. `Majalah` adalah cabang lain; `edisi` dan `terbit()` bukan milik `Ensiklopedia`.
5. Total: atribut judul, penulis, jilid; method info(), baca(), cari().

Analisis setiap opsi:

- **A.** ✓ Tiga tingkat digabung, tanpa Majalah.
- **B.** ✗ Jebakan: hanya anggota yang tertulis pada kotak sendiri.
- **C.** ✗ Jebakan: anggota dari Koleksi (kakek) terlewat.
- **D.** ✗ Jebakan: anggota Majalah ikut ditambahkan padahal berada pada cabang lain.
- **E.** ✗ Jebakan: anggota dari Buku terlewat.

Jawaban: A.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 17 Kunci: C

Konsep: Bila subclass tidak menulis ulang sebuah method, objeknya memakai versi milik induk. Daftar yang berisi objek berbeda jenis dapat diproses dengan satu pemanggilan.

Langkah:

1. `KurirMotor()`: versi KurirMotor → "motor".
2. `KurirSepeda()`: tidak menulis ulang `antar`, memakai versi `Kurir` → "reguler".
3. `KurirDrone()`: versi KurirDrone → "udara".
4. `Kurir()`: versi Kurir → "reguler".

Analisis setiap opsi:

- **A.** ✗ Jebakan: nama class dikira menjadi hasil; KurirSepeda tidak punya versi sendiri.
- **B.** ✗ Jebakan: hasil KurirDrone dipakai untuk KurirSepeda.
- **C.** ✓ Versi sendiri untuk motor dan drone, versi induk untuk KurirSepeda dan Kurir.
- **D.** ✗ Jebakan: empat elemen daftar menghasilkan empat nilai.
- **E.** ✗ Jebakan: method diwarisi dari induk.

Jawaban: C.

No. 18 Kunci: E

Konsep: `super().__init__(...)` menjalankan konstruktor class induk untuk mengisi atribut milik induk. Subclass kemudian menambah atributnya sendiri.

Langkah:

1. `Asrama("Melati", 3, 10)` menjalankan `__init__` milik `Asrama`.
2. `super().__init__(nama, lantai)` mengisi `nama` = "Melati" dan `lantai` = 3.
3. `self.kamar` = 10.
4. `kapasitas()` = 3 × 10 × 2 = 60. Keluaran: `Melati 60`.

Analisis setiap opsi:

- **A.** ✗ Jebakan: 30 muncul bila faktor 2 terlupa.
- **B.** ✗ Jebakan: 20 muncul bila lantai terlupa.
- **C.** ✗ Jebakan: 13 menjumlahkan lantai dan kamar, bukan mengalikan.
- **D.** ✗ Jebakan: `super().__init__` dipanggil sehingga atribut induk ada.
- **E.** ✓ 3 × 10 × 2.

Jawaban: E.

No. 19 Kunci: D

Konsep: Subclass mewarisi seluruh atribut dan method superclass. `pass` berarti subclass tidak menambah apa pun, tetapi tetap memiliki semua milik induknya.

Langkah:

1. `Kaktus("Kaktus mini")` memakai `__init__` milik `Tanaman` (diwarisi): `nama` = "Kaktus mini".
2. `k.siram()` memakai method `siram` milik `Tanaman`: "Kaktus mini disiram".

Analisis setiap opsi:

- **A.** ✗ Jebakan: `siram()` mengembalikan teks yang ditambah " disiram".
- **B.** ✗ Jebakan: nama tidak diubah menjadi huruf kecil.
- **C.** ✗ Jebakan: method tidak mengembalikan namanya sendiri.
- **D.** ✓ Atribut dan method diwarisi.
- **E.** ✗ Jebakan: method yang tidak tertulis di subclass dikira tidak dimiliki.

Jawaban: D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 20 Kunci: B

Konsep: Polimorfisme: satu pemanggilan (`a.bunyi()`) menghasilkan perilaku menurut jenis objek yang sedang dipegang. Versi yang dipakai adalah milik class objek tersebut.

Langkah:

1. Objek pertama `Gitar`: `bunyi()` versi Gitar → "petik".
2. Objek kedua `Drum`: versi Drum → "tabuh".

Analisis setiap opsi:

- **A.** ✗ Jebakan: versi milik induk dipakai bila subclass tidak menulis ulang; di sini kedua subclass menulis ulang.
- **B.** ✓ Gitar lalu Drum.
- **C.** ✗ Jebakan: urutan list `[Gitar(), Drum()]` dibalik.
- **D.** ✗ Jebakan: kedua objek dikira memakai versi yang sama.
- **E.** ✗ Jebakan: `bunyi` ada pada kedua class.

Jawaban: B.

No. 21 Kunci: E

Konsep: Polimorfisme dapat menggantikan percabangan panjang menurut jenis: tiap jenis menjadi subclass yang menulis ulang method yang sama. Menambah jenis baru cukup menambah class baru tanpa mengubah kode lama yang sudah berjalan.

Langkah:

1. Pada if–elif, jenis baru (truk) berarti menambah cabang pada fungsi yang sudah ada.
2. Pada rancangan polimorfik, jenis baru berarti menambah subclass `Truk(Tarif)` dengan `biaya()` sendiri.
3. Nama method harus sama agar satu pemanggilan dapat menangani semua jenis.

Analisis setiap opsi:

- **A.** ✗ Jebakan: tiap jenis baru mengharuskan fungsi diubah.
- **B.** ✗ Jebakan: polimorfisme justru memakai nama method yang sama.
- **C.** ✗ Jebakan: jenis baru pada if–elif menambah cabang pada fungsi yang sama.
- **D.** ✗ Jebakan: polimorfisme bekerja dengan objek dari berbagai subclass yang berbagi superclass.
- **E.** ✓ Cukup menambah subclass baru.

Jawaban: E.

No. 22 Kunci: A, B, D

Konsep: Overriding mengganti method induk pada subclass. `super().method()` memanggil versi induk dan dapat menambahinya. Bila subclass menulis ulang tanpa `super()`, versi induk tidak dipakai.

Langkah:

1. `Catatan("hai")`: versi `Catatan` → `[hai]`.
2. `CatatanPenting("rapat")`: `"!" + super().baca() + "!"` dengan `super().baca()` = `[rapat]` → `![rapat]!`.
3. `CatatanPribadi("kode")`: versi sendiri yang tidak memanggil `super()`: `"*" * 4` = `****`.

Analisis setiap opsi:

- **A.** ✓ Versi induk.
- **B.** ✓ Versi CatatanPenting memakai `super()`.
- **C.** ✗ Jebakan: CatatanPribadi menulis ulang dan tidak memanggil versi CatatanPenting.
- **D.** ✓ Menulis ulang tanpa `super()`.
- **E.** ✗ Jebakan: `super()` memanggil class di atasnya (Catatan), bukan subclass di bawahnya.

Jawaban: A, B, dan D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 23 Kunci: A, C, D

Konsep: Setter menolak nilai di luar aturan; penolakan berarti atribut tidak berubah. Pemanggilan method hanyalah permintaan, bukan perubahan yang pasti terjadi.

Langkah (nilai __angka):

1. Awal 0.
2. `set_angka(85)`: diterima → 85.
3. `set_angka(110)`: ditolak → tetap 85.
4. `set_angka(100)`: diterima (batas atas ikut) → 100.
5. `set_angka(-5)`: ditolak → tetap 100. Keluaran: 100.

Analisis setiap opsi:

- **A.** ✓ Nilai akhir 100.
- **B.** ✗ Jebakan: 110 melewati batas, sehingga ditolak.
- **C.** ✓ -5 berada di bawah batas bawah.
- **D.** ✓ <= menyertakan batas atas 100.
- **E.** ✗ Jebakan: membuka atribut membuat aturan tidak terjaga; validasi justru harus berada di dalam class.

Jawaban: A, C, dan D.

No. 24 Kunci: 1B 2S 3B

Konsep: Public (tanpa awalan) dapat diakses bebas; protected (_) hanyalah kesepakatan dan tetap dapat diakses; private (__) tidak dapat diakses dengan nama itu dari luar class.

Langkah:

1. `s.nama` → alfa.
2. `s._log` → `[]` (dapat dibaca walau berawalan satu garis bawah).
3. `s.__kunci` → `AttributeError`.

Analisis:

- **1.** ✓ Publik dapat dibaca.
- **2.** ✗ Jebakan: atribut privat menimbulkan AttributeError saat diakses dari luar.
- **3.** ✓ Satu garis bawah hanya kesepakatan.

Jawaban: Benar, Salah, Benar.

No. 25 Kunci: A, B, D

Konsep: Sebuah daftar dapat berisi objek subclass dan induk. Method yang hanya ada di subclass tidak dimiliki objek induk, sehingga memanggilnya pada objek induk menimbulkan `AttributeError`.

Langkah:

1. Perulangan: `Kue().rasa()` → manis; `Brownies().rasa()` → cokelat.
2. `daftar[1].hias()`: objek Brownies memiliki `hias`; berhasil (nilai kembalian tidak dicetak).
3. `daftar[0].hias()`: objek Kue tidak memiliki `hias` → `AttributeError`.

Analisis setiap opsi:

- **A.** ✓ manis lalu cokelat.
- **B.** ✓ Brownies memiliki hias.
- **C.** ✗ Jebakan: Kue biasa tidak memiliki hias, walau Brownies adalah Kue.
- **D.** ✓ Objek Kue tidak memiliki method hias.
- **E.** ✗ Jebakan: method khusus subclass tidak dimiliki objek superclass.

Jawaban: A, B, dan D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Simulasi 1 — Prediksi TKA 2026

No. 1 Kunci: D

Konsep: Akun dan aset digital perusahaan (repositori, akun toko aplikasi, domain) sebaiknya dimiliki atas nama resmi perusahaan. Saat pegawai berhenti, kepemilikan dialihkan dan akses pribadinya dicabut sebelum ia pergi.

Langkah:

1. Akun penting studio masih milik email pribadi → risiko kehilangan akses.
2. Selagi Hendra masih bekerja, kepemilikan dapat dialihkan dengan mudah.
3. Alihkan ke email resmi studio, lalu cabut akses pribadi Hendra.

Analisis setiap opsi:

- **A.** ✗ Jebakan: Hendra mungkin bisa dihubungi sekarang, tetapi studio tetap bergantung pada orang yang sudah keluar.
- **B.** ✗ Berbagi kata sandi akun pribadi tidak aman, dan akun tetap milik Hendra.
- **C.** ✗ Akun lama berisi repositori, gim yang sudah terbit, dan domain; meninggalkannya berarti kehilangan aset.
- **D.** ✓ Kepemilikan pindah ke studio selagi Hendra masih dapat membantu prosesnya.
- **E.** ✗ Setelah Hendra pergi, pemindahan akun jauh lebih sulit dan memakan waktu.

Jawaban: D.

No. 2 Kunci: B

Konsep: Perubahan pada basis data produksi didahului cadangan yang dapat dipulihkan dan uji coba pada salinan data nyata, lalu dijalankan saat pemakaian sepi agar pengguna tidak terganggu.

Langkah:

1. Uji di laptop hanya dengan 10 data → belum mewakili 3.500 data nyata.
2. Cadangkan dulu agar data dapat dipulihkan bila skrip gagal.
3. Uji pada salinan data produksi, lalu jalankan di luar jam ramai.

Analisis setiap opsi:

- **A.** ✗ Jebakan: lolos uji kecil di laptop belum menjamin aman untuk 3.500 data, dan siang hari pendaftaran sedang ramai.
- **B.** ✓ Ada cadangan, uji di salinan data nyata, dan dijalankan saat sepi.
- **C.** ✗ Menghapus tabel menghilangkan seluruh data pendaftar.
- **D.** ✗ Cadangan yang dibuat setelah masalah terjadi sudah berisi data yang rusak.
- **E.** ✗ Membebani pendaftar dan tidak perlu; kolom baru dapat ditambahkan tanpa menghapus data.

Jawaban: B.

No. 3 Kunci: 1S 2B 3S

Konsep: Budaya mutu menekankan perbaikan berkelanjutan: tim meninjau proses kerja berdasarkan data (retrospeksi), lalu menerapkan perbaikan pada proyek berikutnya tanpa menyalahkan individu.

Analisis setiap pernyataan:

- **Pernyataan 1 → Salah.** Jebakan: data bug memang dibahas, tetapi retrospeksi mencari perbaikan proses, bukan menyalahkan orang.
- **Pernyataan 2 → Benar.** Sebagian besar bug berasal dari layar yang tidak diuji di ponsel kecil; menambah langkah uji itu mencegah bug serupa.
- **Pernyataan 3 → Salah.** Perbaikan proses justru diterapkan pada proyek berikutnya; tim akan memakai cara kerja yang sama.

Jawaban: Salah, Benar, Salah.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 4 Kunci: A

Konsep: Pendapatan kotor = harga × jumlah pembeli. Pendapatan bersih setelah potongan toko 30 % = 70 % × pendapatan kotor. Harga terbaik adalah yang menghasilkan pendapatan bersih terbesar, bukan yang pembelinya paling banyak atau harganya paling tinggi.

Langkah:

1. Rp15.000 × 4.000 = Rp60.000.000 → 70 % = Rp42.000.000.
2. Rp30.000 × 2.200 = Rp66.000.000 → 70 % = Rp46.200.000.
3. Rp45.000 × 1.300 = Rp58.500.000 → 70 % = Rp40.950.000.
4. Terbesar: Rp30.000 dengan Rp46.200.000.

Analisis setiap opsi:

- **A.** ✓ 70 % × Rp66.000.000 = Rp46.200.000, paling besar.
- **B.** ✗ Rp60.000.000 adalah pendapatan kotor dan bukan yang terbesar.
- **C.** ✗ Jebakan: harga tertinggi tidak memberi pendapatan terbesar karena pembelinya jauh lebih sedikit.
- **D.** ✗ Jebakan: pilihan harganya benar, tetapi Rp66.000.000 belum dipotong 30 %.
- **E.** ✗ Jebakan: pembeli terbanyak tidak berarti pendapatan bersih terbesar.

Jawaban: A.

No. 5 Kunci: B

Konsep: Kabel listrik yang terkelupas dapat menyebabkan sengatan listrik dan korsleting. Perangkat harus segera diputus dari sumber listrik, dihentikan pemakaiannya, dan dilaporkan untuk diganti dengan yang standar.

Langkah:

1. Kawat terbuka → bahaya sengatan listrik dan korsleting.
2. Baterai 40 % masih cukup untuk bekerja sementara tanpa charger.
3. Cabut charger dan laporkan agar diganti.

Analisis setiap opsi:

- **A.** ✗ Selotip kertas bukan isolator listrik yang layak dan mudah lepas.
- **B.** ✓ Bahaya dihentikan segera, dan charger rusak diganti lewat laporan.
- **C.** ✗ Bahaya tetap ada; kawat dapat tersentuh tanpa sengaja atau memicu korsleting.
- **D.** ✗ Menekuk kabel justru dapat memperparah kerusakan dan bahaya tetap tersembunyi.
- **E.** ✗ Jebakan: tenggat tugas tidak sebanding dengan bahaya; baterai 40 % masih dapat dipakai tanpa charger.

Jawaban: B.

No. 6 Kunci: C

Konsep: Papan Kanban memperlihatkan alur kerja proyek. Batas jumlah pekerjaan yang sedang berjalan (WIP limit) mencegah pekerjaan menumpuk setengah jadi; pelanggaran batas diperbaiki dengan menuntaskan pekerjaan, bukan menaikkan batasnya.

Langkah:

1. Kolom Sedang dikerjakan berisi 4 kartu → melanggar aturan (1).
2. Andi memegang 2 kartu (Login Google dan Laporan bulanan) → melanggar aturan (2).
3. Bila salah satu kartu Andi dikembalikan ke Akan dikerjakan, kolom berisi 3 kartu dan setiap programmer memegang satu kartu → kedua aturan terpenuhi.

Analisis setiap opsi:

- **A.** ✗ Jebakan: papan tampak patuh, tetapi masalahnya tetap ada; menaikkan batas meniadakan fungsi aturan.
- **B.** ✗ Menambah kartu ke kolom yang sudah melebihi batas memperparah pelanggaran.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **C.** ✓ Kedua aturan kembali terpenuhi: kolom berisi 3 kartu, satu kartu per programmer.
• **D.** ✗ Budi hanya memegang satu kartu, sesuai aturan; bekerja sendiri pada satu kartu justru dianjurkan.
• **E.** ✗ Jebakan: Keranjang sudah selesai diprogram dan hanya menunggu uji 2 hari; mengembalikannya membuang pekerjaan yang sudah jadi.

Jawaban: C.

No. 7 Kunci: E

Konsep: Database administrator (DBA) mengelola basis data: struktur tabel, hak akses pengguna basis data, kinerja kueri, serta cadangan dan pemulihan data.

Langkah:
1. Tabel, hak akses akun basis data, kueri, cadangan basis data → semuanya tentang basis data.
2. Profesi yang mengelola basis data adalah database administrator.

Analisis setiap opsi:
• **A.** ✗ Frontend developer mengurus tampilan, bukan struktur tabel dan kueri.
• **B.** ✗ Jebakan: aplikasi lambat bisa karena jaringan, tetapi tugas dalam lowongan semuanya tentang basis data, bukan router.
• **C.** ✗ UI/UX designer merancang alur dan tampilan, bukan kinerja kueri.
• **D.** ✗ Jebakan: hak akses disebut, tetapi yang dimaksud hak akses akun basis data, bukan membantu guru yang lupa kata sandi aplikasi.
• **E.** ✓ DBA mengelola struktur, hak akses, kinerja, dan cadangan basis data.

Jawaban: E.

No. 8 Kunci: 1B 2B 3S

Konsep: Dokumentasi proyek (README) memuat tujuan aplikasi, cara memasang dan menjalankan, kebutuhan lingkungan, struktur folder, dan penjelasan konfigurasi. Nilai rahasia (kata sandi, kunci API) tidak ditulis di dokumentasi.

Analisis setiap pernyataan:
• **Pernyataan 1 → Benar.** Tim baru dapat menjalankan proyek tanpa menebak-nebak lingkungan yang dibutuhkan.
• **Pernyataan 2 → Benar.** Tim baru tahu apa yang harus diisi, sedangkan nilai rahasia diserahkan lewat jalur yang aman.
• **Pernyataan 3 → Salah.** Jebakan: memang praktis, tetapi README dapat dibaca siapa saja yang membuka repositori sehingga kata sandi bocor.

Jawaban: Benar, Benar, Salah.

No. 9 Kunci: E

Konsep: Sistem berkas Linux membedakan huruf besar dan kecil (case-sensitive), sedangkan Windows umumnya tidak. "gambar/logo.png" dan "Gambar/Logo.PNG" dianggap sama di Windows, tetapi berbeda di Linux.

Langkah:
1. Di Windows: penulisan berbeda huruf tetap ditemukan → logo tampil.
2. Di Linux: "gambar/logo.png" ≠ "Gambar/Logo.PNG" → berkas tidak ditemukan.
3. Gambar lain tampil karena penulisannya sama persis → masalah hanya pada logo.
4. Perbaikan: samakan penulisan, misalnya ubah nama menjadi gambar/logo.png.

Analisis setiap opsi:
• **A.** ✗ Gambar lain di folder yang sama tampil, jadi unggahan sudah lengkap.
• **B.** ✗ Ukuran berkas tidak membuat berkas "tidak ditemukan".

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **C.** ✗ Jebakan: letak berubah, tetapi HTML masih menunjuk gambar/logo.png sehingga tetap tidak ditemukan.
• **D.** ✗ Jebakan: format PNG didukung; mengganti format juga mengubah nama berkas sehingga HTML tetap tidak cocok.
• **E.** ✓ Penulisan yang sama persis membuat server Linux menemukan berkasnya.

Jawaban: E.

No. 10 Kunci: A, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Etika kerja profesional meliputi kejujuran (melaporkan kemajuan dan hambatan apa adanya), menjaga kerahasiaan sesuai perjanjian, dan menghormati hak klien atas karya yang dipesan.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Komunikasi jujur dan dini memberi studio kesempatan menyesuaikan rencana.
• **Pernyataan 2** ✗ Jebakan: portofolio memang penting, tetapi menyebarkan isi gim sebelum diumumkan melanggar NDA.
• **Pernyataan 3** ✗ Tenggat terlihat terpenuhi, tetapi menyembunyikan mutu yang belum teruji tidak jujur dan merugikan studio.
• **Pernyataan 4** ✓ Setelah NDA berakhir dan dengan izin, proyek boleh ditampilkan.
• **Pernyataan 5** ✓ Laporan yang jujur menjaga kepercayaan klien.

Jawaban: pernyataan 1, 4, 5.

No. 11 Kunci: C

Konsep: Komunikasi web berbasis TCP/IP berjalan berurutan: nama domain diterjemahkan menjadi alamat IP (DNS), koneksi TCP dibuka ke alamat dan port server (443 untuk HTTPS, disertai pembentukan enkripsi), permintaan HTTP dikirim, server membalas, lalu peramban menampilkan hasilnya.

Langkah:
1. Koneksi tidak dapat dibuka tanpa alamat IP → (2) paling awal.
2. Permintaan HTTP dikirim lewat koneksi TCP → (4) sebelum (3).
3. Server membalas setelah menerima permintaan → (3) sebelum (1).
4. Halaman ditampilkan setelah respons diterima → (5) terakhir.

Analisis setiap opsi:
• **A.** ✗ Jebakan: permintaan HTTP dikirim sebelum koneksi TCP dibuka, padahal HTTP membutuhkan koneksi itu.
• **B.** ✗ Koneksi TCP membutuhkan alamat IP yang baru diperoleh dari DNS.
• **C.** ✓ DNS → koneksi TCP 443 → permintaan HTTP → respons 200 → tampil.
• **D.** ✗ Permintaan HTTP tidak dapat dikirim sebelum alamat server diketahui.
• **E.** ✗ Server tidak membalas sebelum menerima permintaan.

Jawaban: C.

No. 12 Kunci: B, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Protokol lapisan aplikasi TCP/IP antara lain SMTP (mengirim email), IMAP/POP3 (mengambil email), DHCP (membagikan konfigurasi IP otomatis), DNS, dan HTTP/HTTPS. Lapisan internet berisi protokol IP yang mengantar paket.

Analisis setiap pernyataan:
• **Pernyataan 1** ✗ Jebakan: IMAP dipakai untuk membaca/mengambil email; pengiriman email memakai SMTP.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 2** ✓ DHCP membagikan alamat IP, mask, gateway, dan DNS secara otomatis.
• **Pernyataan 3** ✗ SMTP untuk mengirim email; membaca email memakai IMAP atau POP3.
• **Pernyataan 4** ✗ Jebakan: DHCP mengurus alamat IP, tetapi DHCP sendiri adalah protokol lapisan aplikasi.
• **Pernyataan 5** ✓ Ketiganya adalah layanan yang dipakai aplikasi, sehingga berada di lapisan aplikasi.

Jawaban: pernyataan 2, 5.

No. 13 Kunci: D

Konsep: Polimorfisme: walaupun total() hanya ditulis di Sewa, pemanggilan self.tarif() di dalamnya menjalankan versi tarif() milik class objek yang sebenarnya. SewaPelajar mewarisi total() tetapi memakai tarif() miliknya sendiri.

Langkah:
1. Sewa().total(2) = 5000 × 2 + 2000 = 12000.
2. SewaPelajar().total(2) = 3000 × 2 + 2000 = 8000 (tarif milik SewaPelajar).
3. hasil = 12000 + 8000 = 20000.

Analisis setiap opsi:
• **A.** ✗ Jebakan: total() dianggap selalu memakai tarif() milik Sewa karena ditulis di class Sewa: 12000 + 12000.
• **B.** ✗ Biaya tetap 2000 terlupa: 10000 + 6000.
• **C.** ✗ Hanya penyewa pertama yang dihitung.
• **D.** ✓ 12000 + 8000 = 20000.
• **E.** ✗ Jebakan: biaya 2000 dianggap hanya dikenakan sekali untuk semua penyewa.

Jawaban: D.

No. 14 Kunci: A, B, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pada pewarisan bertingkat, subclass mewarisi dari superclass beserta seluruh superclass di atasnya. super() memanggil method milik superclass terdekat yang memilikinya. Superclass tidak mewarisi tambahan milik subclass.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ WaliKelas → super() ke Guru → super() ke Akun: ["lihat"] + ["nilai"] + ["rapor"].
• **Pernyataan 2** ✓ Guru dan WaliKelas tidak menulis __init__, sehingga memakai milik Akun.
• **Pernyataan 3** ✗ Jebakan: Guru tidak mewarisi milik subclass-nya; hasilnya ["lihat", "nilai"].
• **Pernyataan 4** ✓ Pewarisan bertingkat: Akun → Guru → WaliKelas.
• **Pernyataan 5** ✓ super() dari WaliKelas mencari hak() ke atas; Guru tidak punya, sehingga memakai milik Akun.

Jawaban: pernyataan 1, 2, 4, 5.

No. 15 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Abstraksi menyembunyikan detail di balik method sederhana; pewarisan membuat subclass dari superclass; polimorfisme membuat method bernama sama berperilaku berbeda menurut class objeknya. Bersama-sama, rancangan mudah dikembangkan tanpa mengubah kode pemakainya.

Analisis setiap pernyataan:
• **Pernyataan 1** ✗ Jebakan: kasir hanya memanggil bayar(); cara bayar baru cukup menjadi subclass baru tanpa mengubah kode kasir.
• **Pernyataan 2** ✓ Detail disembunyikan di balik satu method yang sederhana.
• **Pernyataan 3** ✓ Satu nama method, perilaku berbeda sesuai class objeknya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 4 ✗** Justru nama method harus sama (bayar) agar dapat dipanggil dengan cara yang sama.
• **Pernyataan 5 ✓** Pewarisan memudahkan penambahan jenis baru dengan antarmuka yang sama.

Jawaban: pernyataan 2, 3, 5.

No. 16 Kunci: A

Konsep: Parameter bawaan (`bonus=0`) dipakai bila argumennya tidak dikirim. Fungsi dapat memanggil fungsi lain dan mengembalikan hasilnya lewat return.

Langkah:
1. nilai_akhir(6, 3): salah ≠ 0 → poin(6, 3) dengan bonus 0 → 24 − 3 + 0 = 21.
2. nilai_akhir(5, 0): salah = 0 → poin(5, 0, 5) → 20 − 0 + 5 = 25.
3. Keluaran: 21 25.

Analisis setiap opsi:
• **A. ✓** 21 untuk peserta pertama dan 25 untuk peserta kedua.
• **B. ✗** Jebakan: bonus 5 yang dikirim ke poin() terlupa.
• **C. ✗** Jebakan: bonus dianggap diberikan kepada kedua peserta.
• **D. ✗** Jebakan: jawaban salah ditambahkan, bukan dikurangkan: 24 + 3.
• **E. ✗** bonus memiliki nilai bawaan 0, jadi pemanggilan dengan dua argumen sah.

Jawaban: A.

No. 17 Kunci: 1S 2B 3B

Konsep: Enkapsulasi: atribut privat diubah hanya lewat setter yang memeriksa aturan. Konstruktor yang memakai setter yang sama memastikan aturan berlaku sejak objek dibuat.

Langkah:
1. Produk("Kopi", −5000): __harga = 0, lalu set_harga(−5000) → −5000 > 0 salah → tetap 0.
2. print(p.get_harga()) → 0.

Analisis setiap pernyataan:
• **Pernyataan 1 → Salah.** Jebakan: harga dikirim lewat set_harga(); −5000 tidak memenuhi h > 0 sehingga harga tetap 0. Keluaran: 0.
• **Pernyataan 2 → Benar.** Validasi di setter dipakai juga oleh konstruktor, jadi tidak ada jalan masuk nilai yang salah.
• **Pernyataan 3 → Benar.** 12000 diterima; 0 ditolak karena 0 > 0 salah. Harga tetap 12000.

Jawaban: Salah, Benar, Benar.

No. 18 Kunci: E

Konsep: Dictionary menyimpan pasangan kunci–nilai. Pola penghitung: bila kunci sudah ada, nilainya ditambah 1; bila belum ada, kunci dibuat dengan nilai 1.

Langkah:
1. "A" → belum ada → A: 1.
2. "B" → B: 1.
3. "A" → A: 2.
4. "C" → C: 1.
5. "A" → A: 3.
6. "B" → B: 2. Hasil: {'A': 3, 'B': 2, 'C': 1}.

Analisis setiap opsi:
• **A. ✗** Jebakan: dictionary dianggap hanya mencatat calon yang muncul, tanpa menambah hitungan.
• **B. ✗** Jebakan: suara pertama setiap calon dianggap belum dihitung (dimulai dari 0).
• **C. ✗** Jebakan: suara pertama dianggap masuk ke dua cabang sekaligus sehingga setiap calon

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

kelebihan 1.

• **D.** ✗ List suara tidak berubah; yang dicetak adalah dictionary hasil.

• **E.** ✓ A 3 suara, B 2 suara, C 1 suara.

Jawaban: E.

No. 19 Kunci: B

Konsep: Perulangan pada diagram alir ditandai garis yang kembali ke langkah sebelumnya. Perulangan berhenti saat salah satu syarat keluar terpenuhi; urutan pemeriksaan di dalam perulangan menentukan pesan yang tampil.

Langkah:

1. 1111: salah → percobaan 1 → 1 ≥ 3? Tidak → "PIN salah" → kembali input.
2. 0000: salah → percobaan 2 → Tidak → "PIN salah" → kembali input.
3. 9999: salah → percobaan 3 → 3 ≥ 3? Ya → "Akun terkunci" → selesai.
4. PIN 2580 tidak pernah dibaca karena program sudah selesai.

Analisis setiap opsi:

• **A.** ✗ Jebakan: kesempatan dianggap tiga kali salah lalu masih boleh mencoba lagi; padahal salah ketiga langsung mengunci.

• **B.** ✓ Dua kali "PIN salah", lalu pada salah ketiga syarat ≥ 3 terpenuhi sehingga akun terkunci.

• **C.** ✗ Jebakan: "PIN salah" dianggap tampil sebelum pemeriksaan jumlah percobaan; menurut diagram, pemeriksaan ≥ 3 lebih dulu.

• **D.** ✗ Akun dianggap terkunci pada salah kedua; padahal saat itu percobaan baru 2.

• **E.** ✗ Pada salah ketiga program sudah berhenti, sehingga 2580 tidak sempat diperiksa.

Jawaban: B.

No. 20 Kunci: C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pada if–elif–else, hanya satu cabang yang dijalankan. Syarat gabungan dengan and bernilai benar bila kedua bagiannya benar. Tanda > tidak memasukkan nilai batas.

Langkah:

1. jam ≤ 1 → 3000; 2–3 jam → 3000 + (jam − 1) × 2000; lebih dari 3 jam → 10000.
2. Lalu bila member DAN biaya > 5000 → dikurangi 2000.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: biaya 2 jam = 3000 + 2000 = 5000; syarat biaya > 5000 salah, jadi tidak ada potongan. Hasilnya 5000.

• **Pernyataan 2** ✗ Biaya 10000, lalu member dan 10000 > 5000 → dipotong 2000 → 8000.

• **Pernyataan 3** ✓ 3000 + 2 × 2000 = 7000; member dan 7000 > 5000 → 5000.

• **Pernyataan 4** ✗ Keduanya 3000; 3000 tidak lebih dari 5000 sehingga member tidak mendapat potongan.

• **Pernyataan 5** ✓ 4 jam masuk cabang else → 10000; bukan member, tidak dipotong.

Jawaban: pernyataan 3, 5.

No. 21 Kunci: A, B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Desain modular yang efektif membuat modul yang dapat dipakai ulang: tugasnya tunggal dan umum, tidak terikat pada satu aplikasi, antarmukanya terdokumentasi, dan perubahannya diuji pada semua pemakai.

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ Satu sumber logika menghilangkan duplikasi dan perubahan cukup sekali.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 2** ✓ Modul bersama harus umum agar dapat dipakai ulang oleh aplikasi mana pun.
• **Pernyataan 3** ✓ Perubahan modul bersama berdampak ke semua pemakainya.
• **Pernyataan 4** ✗ Jebakan: praktis untuk absensi, tetapi modul jadi terikat pada tampilan satu aplikasi; sebaiknya fungsi mengembalikan hasil dan tiap aplikasi menampilkan pesannya sendiri.
• **Pernyataan 5** ✓ Dokumentasi antarmuka modul membuat pemakai menggunakannya dengan benar.

Jawaban: pernyataan 1, 2, 3, 5.

No. 22 Kunci: D

Konsep: Operator perbandingan (>=) menghasilkan nilai bool (True/False). Operator `or` bernilai True bila salah satu bagiannya True. Hasilnya bertipe bool.

Langkah:
1. umur >= 17 → 16 >= 17 → False.
2. False or True → True.
3. type(True) → <class 'bool'>.

Analisis setiap opsi:
• **A.** ✗ Jebakan: hanya syarat umur yang diperhatikan; padahal or cukup satu syarat benar.
• **B.** ✗ True dan False adalah nilai bool, bukan teks.
• **C.** ✗ boleh berisi hasil ekspresi logika, bukan nilai umur.
• **D.** ✓ False or True = True, bertipe bool.
• **E.** ✗ Hasil or di sini adalah True dan bertipe bool.

Jawaban: D.

No. 23 Kunci: A

Konsep: Git mencatat nama dan email penulis commit dari konfigurasi `user.name` dan `user.email` di komputer, bukan dari akun yang sedang masuk di peramban. Konfigurasi dapat berlaku global (seluruh komputer) atau lokal (satu repositori).

Langkah:
1. Penulis commit diambil dari konfigurasi Git komputer → saat ini milik "Admin Lab".
2. Komputer dipakai banyak siswa → mengubah konfigurasi global akan memengaruhi commit siswa lain.
3. Atur konfigurasi lokal di repositori itu saja, misalnya git config user.name "Sinta".

Analisis setiap opsi:
• **A.** ✓ Konfigurasi lokal membuat commit Sinta benar tanpa mengubah identitas siswa lain.
• **B.** ✗ Jebakan: commit Sinta menjadi benar, tetapi commit siswa lain di komputer itu ikut tercatat atas nama Sinta.
• **C.** ✗ Akun di peramban tidak menentukan identitas penulis commit.
• **D.** ✗ Kloning ulang tetap memakai konfigurasi global "Admin Lab".
• **E.** ✗ Jebakan: hak menulis ke repositori tidak mengubah nama dan email penulis yang dicatat Git di komputer.

Jawaban: A.

No. 24 Kunci: A, C, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Diagnosis jaringan dilakukan bertahap: ping 127.0.0.1 menguji TCP/IP di komputer sendiri, ping gateway menguji sambungan ke jaringan lokal, ping server menguji jangkauan ke tujuan. Uji pertama yang gagal menunjukkan letak masalah.

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ 127.0.0.1 (loopback) menguji tumpukan TCP/IP di laptop itu sendiri.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 2** ✗ Jebakan: laptop meja 6 dapat membuka server, jadi server berjalan; masalahnya ada di sisi laptop meja 7.
• **Pernyataan 3** ✓ Konfigurasi sudah benar dan semua tujuan di jaringan gagal dijangkau → sambungan fisik patut dicurigai.
• **Pernyataan 4** ✓ Pemeriksaan dimulai dari bagian yang paling mungkin dan paling mudah diuji.
• **Pernyataan 5** ✗ Ping memakai alamat IP langsung, sehingga tidak bergantung pada DNS.

Jawaban: pernyataan 1, 3, 4.

No. 25 Kunci: C

Konsep: Atribut yang dibuat di __init__ dengan self dimiliki masing-masing objek. Objek x dan y memiliki list siswa sendiri-sendiri.

Langkah:
1. x.siswa = [] dan y.siswa = [] (dua list berbeda).
2. x.tambah dua kali → x.siswa = ["Ani", "Budi"].
3. y.tambah sekali → y.siswa = ["Cici"].
4. Keluaran: 2 1.

Analisis setiap opsi:
• **A.** ✗ Jebakan: kedua objek dianggap memakai satu list yang sama.
• **B.** ✗ Jumlah siswa y dianggap sama dengan x.
• **C.** ✓ x berisi 2 siswa, y berisi 1 siswa.
• **D.** ✗ Setiap pemanggilan tambah() dianggap menimpa isi sebelumnya.
• **E.** ✗ Jebakan: semua siswa dianggap masuk ke objek pertama.

Jawaban: C.

Simulasi 2 — Prediksi TKA 2026

No. 1 Kunci: C

Konsep: Aset digital berupa lisensi dan langganan perlu ditinjau pemakaiannya: layanan yang tidak dipakai dihentikan, jumlah akun disesuaikan dengan pemakai aktif, dan layanan yang dipakai setiap hari dipertahankan.

Langkah:
1. Perekam layar tidak dipakai → hentikan: hemat Rp200.000.
2. Manajemen tugas: 10 akun, aktif 4 → kurangi 6 akun × Rp50.000 = Rp300.000.
3. Desain UI dan server uji dipakai setiap hari → pertahankan.
4. Total hemat: 200.000 + 300.000 = Rp500.000 per bulan.

Analisis setiap opsi:
• **A.** ✗ Aplikasi desain dipakai 5 desainer setiap hari; menghentikannya mengganggu pekerjaan.
• **B.** ✗ Jebakan: benar, tetapi masih ada 6 akun manajemen tugas yang dibayar tanpa dipakai.
• **C.** ✓ Kedua pemborosan dihapus tanpa mengganggu pekerjaan: hemat Rp500.000.
• **D.** ✗ Jebakan: 4 orang masih memakai manajemen tugas, jadi menghentikan seluruhnya mengganggu kerja mereka.
• **E.** ✗ Benar sebagian; perekam layar yang tidak dipakai masih dibayar.

Jawaban: C.

No. 2 Kunci: 1B 2S 3B

Konsep: Budaya mutu menetapkan kriteria rilis yang jelas. Bug diprioritaskan menurut tingkat keparahan dan dampaknya pada fungsi utama; bug ringan dicatat dan dijadwalkan agar tidak terlupa.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap pernyataan:

• **Pernyataan 1 → Benar.** Pembayaran adalah fungsi utama; aplikasi yang menutup sendiri saat membayar melanggar aturan mutu.

• **Pernyataan 2 → Salah.** Jebakan: cepat diperbaiki tidak sama dengan penting; prioritas ditentukan dampaknya pada fungsi utama.

• **Pernyataan 3 → Benar.** Bug kecil yang tidak menghalangi fungsi utama boleh dijadwalkan ulang, asalkan terdokumentasi.

Jawaban: Benar, Salah, Benar.

No. 3 Kunci: B

Konsep: Uji penerimaan pengguna (user acceptance test, UAT) dilakukan oleh pemakai sebenarnya dengan alur kerja nyata sebelum aplikasi dipakai. Tahap ini menangkap ketidaksesuaian alur yang tidak terlihat oleh uji teknis.

Langkah:

1. Kode benar secara teknis (uji otomatis dan peninjauan lolos).
2. Masalahnya adalah alur aplikasi tidak sesuai kebiasaan kerja kasir — kasir tidak pernah mencoba sebelum rilis.
3. Tahap yang menangkap hal itu: uji penerimaan oleh kasir sungguhan.

Analisis setiap opsi:

• **A.** ✗ Jebakan: total belanja tidak salah; uji otomatis tidak menilai kecocokan alur kerja kasir.

• **B.** ✓ Kasir yang mencoba lebih dulu akan langsung menemukan urutan diskon yang tidak sesuai.

• **C.** ✗ Peninjauan kode memeriksa mutu kode, bukan kecocokan dengan cara kerja pengguna.

• **D.** ✗ Dokumentasi teknis tidak mengubah alur yang sudah salah sejak awal.

• **E.** ✗ Pemeliharaan server berlangsung setelah rilis dan tidak berkaitan dengan alur kerja kasir.

Jawaban: B.

No. 4 Kunci: C

Konsep: Pengaturan yang berbeda per komputer atau bersifat rahasia disimpan di variabel lingkungan (berkas .env) yang tidak ikut ke repositori. Berkas contoh (.env.example) memberi tahu variabel apa saja yang harus diisi.

Langkah:

1. .env tidak ikut terkloning karena memang dikecualikan → wajar bila belum ada.
2. .env.example berisi daftar variabel → salin menjadi .env.
3. Isi nilainya (alamat, kata sandi) yang diberikan lewat jalur aman oleh ketua tim.

Analisis setiap opsi:

• **A.** ✗ Isi .env yang rahasia akan tersebar ke semua yang dapat membuka repositori.

• **B.** ✗ Kata sandi di kode ikut terunggah dan sulit diganti per komputer.

• **C.** ✓ Variabel tersedia dan nilainya diisi tanpa membocorkan rahasia.

• **D.** ✗ Jebakan: berkas .env ada, tetapi isinya kosong sehingga DB_HOST tetap belum diatur.

• **E.** ✗ Kloning ulang tetap tidak membawa .env karena berkas itu memang dikecualikan.

Jawaban: C.

No. 5 Kunci: A

Konsep: Gateway bawaan adalah alamat router yang dipakai untuk keluar ke jaringan lain (internet). Komunikasi di dalam jaringan lokal tidak memakai gateway, sehingga gateway yang salah hanya mengganggu akses ke luar jaringan.

Langkah:

1. Server lokal dapat dibuka → IP dan subnet mask sudah benar.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. Ping ke 8.8.8.8 (alamat IP, tanpa DNS) gagal → masalah ada pada jalan keluar ke internet, bukan DNS.
3. Di gambar gateway 192.168.4.254, padahal router 192.168.4.1 → gateway salah.

Analisis setiap opsi:
• **A.** ✓ Gateway diarahkan ke router yang benar sehingga laptop dapat keluar ke internet.
• **B.** ✗ Jebakan: DNS tidak dipakai saat ping ke alamat IP; ping 8.8.8.8 tetap gagal, jadi masalahnya bukan DNS.
• **C.** ✗ Mask yang sekarang sudah benar untuk jaringan 192.168.4.x; server lokal dapat dibuka.
• **D.** ✗ Mengubah IP laptop tidak memperbaiki gateway yang tetap menunjuk alamat yang salah.
• **E.** ✗ Server lokal berjalan normal; masalahnya akses ke internet.

Jawaban: A.

No. 6 Kunci: 1B 2S 3S

Konsep: Perulangan for pada string mengambil karakter satu per satu dari kiri. Operator + menggabungkan string; urutan operand menentukan letak karakter baru.

Langkah:
1. h = "G": hasil = "G" + "" = "G".
2. h = "I": hasil = "I" + "G" = "IG".
3. h = "M": hasil = "M" + "IG" = "MIG".

Analisis setiap pernyataan:
• **Pernyataan 1 → Benar.** Setiap huruf baru ditaruh di depan: "G" → "IG" → "MIG".
• **Pernyataan 2 → Salah.** Jebakan: huruf baru ditaruh di DEPAN hasil sebelumnya, sehingga isinya "IG".
• **Pernyataan 3 → Salah.** Huruf baru lalu ditaruh di belakang, sehingga keluaran sama dengan kata asal: GIM.

Jawaban: Benar, Salah, Salah.

No. 7 Kunci: B

Konsep: Path mutlak (absolut) menunjuk lokasi lengkap dari akar drive, sehingga berbeda di setiap komputer. Path relatif dihitung dari folder kerja, sehingga tetap benar selama struktur folder proyek sama.

Langkah:
1. Path di kode menunjuk folder milik laptop Rina → tidak ada di laptop Doni.
2. Struktur di dalam folder gim sama di kedua laptop (aset/latar.png).
3. Gim dijalankan dari folder gim → path relatif aset/latar.png benar di semua laptop.

Analisis setiap opsi:
• **A.** ✗ Akal-akalan ini hanya menyelesaikan masalah di laptop Doni dan menyulitkan anggota lain.
• **B.** ✓ Path relatif berlaku di semua laptop selama gim dijalankan dari folder gim.
• **C.** ✗ Masalahnya letak folder, bukan huruf besar-kecil nama berkas.
• **D.** ✗ Path di kode tetap menunjuk folder Rina, jadi berkas tetap tidak ditemukan.
• **E.** ✗ Jebakan: berjalan di laptop Doni, tetapi rusak lagi di laptop Rina dan laptop lain.

Jawaban: B.

No. 8 Kunci: E

Konsep: Riwayat commit (waktu, pesan, dan berkas yang diubah) dipakai untuk menelusuri sumber bug. Pesan commit yang jelas mempercepat penelusuran; pesan seperti "update" menyulitkan tim.

Langkah:
1. Uji pukul 11.00 masih lolos → kode sampai pukul 11.00 masih benar; sumber bug ada pada commit SETELAH pukul 11.00.
2. Bug sudah muncul Selasa siang → commit yang dicari dibuat antara pukul 11.00 dan Selasa siang.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. Commit pada rentang itu yang mengubah kode perhitungan (kasir.py): d4 pukul 13.00.
4. Pelajaran: pesan "update" tidak menjelaskan apa yang diubah; pesan commit sebaiknya jelas.

Analisis setiap opsi:

• **A.** ✗ profil.py tidak menghitung total struk, dan commit ini sudah ada sebelum uji yang lolos.
• **B.** ✗ style.css hanya mengatur tampilan, tidak menghitung total.
• **C.** ✗ Jebakan: pesannya paling mencurigakan, tetapi c3 dibuat sebelum uji pukul 11.00 yang masih lolos, dan bug terjadi pada pembeli bukan anggota.
• **D.** ✗ Jebakan: commit terbaru tidak otomatis penyebabnya; e5 dibuat Rabu, padahal bug sudah muncul Selasa siang, dan hanya mengubah menu.py.
• **E.** ✓ Satu-satunya commit setelah uji lolos yang mengubah kasir.py sebelum bug muncul.

Jawaban: E.

No. 9 Kunci: A, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Model pendapatan (lisensi, langganan, freemium, iklan, pembelian dalam aplikasi) dipilih sesuai siapa pembelinya, nilai yang mereka butuhkan, dan etika terhadap pengguna, terutama bila pengguna adalah anak-anak.

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ Sesuai temuan survei: sekolah mau membayar langganan bila ada laporan kemajuan.
• **Pernyataan 2** ✗ Jebakan: iklan memang sumber pendapatan, tetapi tidak etis untuk anak-anak dan mengganggu tujuan belajar.
• **Pernyataan 3** ✓ Orang tua dapat mencoba dulu sebelum membayar.
• **Pernyataan 4** ✗ Tekanan membeli pada anak-anak tidak etis dan dapat merugikan orang tua.
• **Pernyataan 5** ✓ Pembayaran dikendalikan orang tua dan berulang setiap bulan.

Jawaban: pernyataan 1, 3, 5.

No. 10 Kunci: E

Konsep: Set adalah kumpulan nilai unik tanpa urutan. Menambahkan nilai yang sudah ada tidak membuat salinan, sehingga `len(set)` langsung memberi jumlah anggota berbeda.

Langkah:

1. Kebutuhan: jumlah siswa BERBEDA; username dapat datang berkali-kali; urutan tidak penting.
2. Set otomatis membuang data ganda → len() = jumlah siswa berbeda.

Analisis setiap opsi:

• **A.** ✗ List menyimpan username ganda, sehingga jumlahnya bukan jumlah siswa berbeda.
• **B.** ✗ Tuple tidak dapat ditambah setelah dibuat, padahal username terus berdatangan.
• **C.** ✗ String gabungan sulit dihitung dan tetap memuat data ganda.
• **D.** ✗ Jebakan: username yang sama tersimpan berkali-kali sebagai nilai pada jam berbeda, sehingga jumlah isinya bukan jumlah siswa berbeda.
• **E.** ✓ Set menyimpan setiap username sekali saja.

Jawaban: E.

No. 11 Kunci: B, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: K3 di ruang kerja komputer meliputi ergonomi (kursi, meja, posisi monitor), pencahayaan yang cukup tanpa silau, istirahat berkala, dan keselamatan lingkungan (kabel tertata).

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: tampak nyaman, tetapi mata terus menatap cahaya terang di belakang layar sehingga cepat lelah karena silau.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

• **Pernyataan 2** ✓ Posisi duduk yang benar mengurangi beban pada pinggang dan punggung.
• **Pernyataan 3** ✗ Duduk dan menatap layar terlalu lama tanpa istirahat memicu kelelahan mata dan otot.
• **Pernyataan 4** ✓ Kabel yang rapi mencegah orang tersandung dan kabel tertarik.
• **Pernyataan 5** ✗ Perbedaan terang layar dan ruangan gelap membuat mata cepat lelah.

Jawaban: pernyataan 2, 4.

No. 12 Kunci: A

Konsep: Perulangan while dengan indeks memeriksa setiap elemen list satu per satu; elemen dapat diubah lewat `stok[i] = ...`. `sum()` menjumlahkan isi list setelah diubah.

Langkah:
1. i = 0: 5 → tidak berubah.
2. i = 1: 0 → kosong 1, stok[1] = stok[0] × 2 = 10 → [5, 10, 0, 3, 8].
3. i = 2: 0 → kosong 2, stok[2] = stok[1] × 2 = 20 (memakai nilai 10 yang baru diisi) → [5, 10, 20, 3, 8].
4. i = 3 dan i = 4: 3 dan 8 → tidak berubah.
5. sum = 5 + 10 + 20 + 3 + 8 = 46. Keluaran: 2 46.

Analisis setiap opsi:
• **A.** ✓ Dua rak kosong; rak ketiga diisi dari nilai rak kedua yang sudah menjadi 10.
• **B.** ✗ Jebakan: rak kedua dianggap masih 0 saat mengisi rak ketiga, sehingga rak ketiga tetap 0: 5 + 10 + 0 + 3 + 8.
• **C.** ✗ Jebakan: jumlah dihitung dari isi awal, padahal list sudah diubah sebelum print.
• **D.** ✗ Perulangan dianggap berhenti setelah menemukan rak kosong pertama.
• **E.** ✗ Kedua rak kosong dianggap diisi dari rak pertama (10 dan 10).

Jawaban: A.

No. 13 Kunci: 1S 2S 3B

Konsep: Dalam komunikasi client-server, setiap permintaan dijawab dengan kode status: 200 OK, 201 Created (data baru dibuat), 400 Bad Request (data permintaan salah), 401 Unauthorized (belum terautentikasi), 404 Not Found, 500 galat server.

Langkah:
1. No. 1 dan 3 (401): identitas tidak sah atau tidak ada token.
2. No. 4 (400): server terjangkau, tetapi isi permintaan tidak valid.
3. No. 5 (201): pesanan baru berhasil dibuat.

Analisis setiap pernyataan:
• **Pernyataan 1 → Salah.** Jebakan: 401 bukan galat server (5xx); permintaan ditolak karena tidak menyertakan token tanda sudah masuk.
• **Pernyataan 2 → Salah.** Server menjawab dengan kode 400, berarti server terjangkau; data yang dikirim (porsi −3) tidak valid.
• **Pernyataan 3 → Benar.** Kode 201 Created berarti permintaan berhasil dan data baru dibuat.

Jawaban: Salah, Salah, Benar.

No. 14 Kunci: D

Konsep: String adalah deretan karakter berindeks mulai 0. `s[:4]` mengambil indeks 0–3 (indeks 4 tidak ikut); `s[-2:]` mengambil dua karakter terakhir; `len(s)` menghitung semua karakter termasuk tanda hubung.

Langkah:
1. kode[:4] → indeks 0, 1, 2, 3 → "PPLG".
2. kode[-2:] → dua karakter terakhir → "12".

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

3. len: P,P,L,G,2,7,-,A,B,1,2 → 11 karakter.
4. Keluaran: PPLG 12 11.

Analisis setiap opsi:

- **A.** ✗ Jebakan: batas akhir 4 dianggap ikut sehingga terambil 5 karakter.
- **B.** ✗ Jebakan: indeks negatif dianggap membalik urutan karakter.
- **C.** ✗ Jebakan: tanda hubung tidak dihitung, padahal termasuk karakter.
- **D.** ✓ Empat karakter pertama, dua karakter terakhir, dan panjang 11.
- **E.** ✗ Potongan [:4] dianggap berakhir di indeks 2.

Jawaban: D.

No. 15 Kunci: C

Konsep: Memberi nilai baru ke parameter (skor = ...) hanya mengubah variabel lokal di dalam fungsi; variabel di luar tidak ikut berubah. Namun, mengubah ISI list yang dikirim (append) terlihat dari luar karena list itu objek yang sama.

Langkah:

1. Di dalam fungsi: skor lokal = 50 + 10 = 60; s di luar tetap 50.
2. daftar.append(60) → list d yang sama kini berisi [60].
3. print(s, d) → 50 [60].

Analisis setiap opsi:

- **A.** ✗ Jebakan: s dianggap ikut berubah, padahal yang berubah hanya skor lokal di dalam fungsi.
- **B.** ✗ Jebakan: list dianggap tidak berubah karena tidak ada return; append mengubah list yang sama.
- **C.** ✓ s tetap 50 dan list d berisi 60.
- **D.** ✗ Keduanya terbalik: s tidak berubah, sedangkan d berubah.
- **E.** ✗ Prosedur tanpa return sah; perubahan isi list tetap terlihat dari luar.

Jawaban: C.

No. 16 Kunci: C, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Pada if–elif, syarat diperiksa dari atas dan berhenti di syarat pertama yang benar, sehingga urutan syarat menentukan hasil. Tanda <= memasukkan nilai batas, sedangkan < tidak.

Langkah:

1. Urutan: mengisi → petir; ≤ 15 → merah; < 50 → kuning; selain itu → hijau.

Analisis setiap pernyataan:

- **Pernyataan 1** ✗ Jebakan nilai batas: 15 <= 15 benar, sehingga hasilnya "merah".
- **Pernyataan 2** ✗ 50 < 50 salah, sehingga tidak masuk cabang kuning; hasilnya "hijau".
- **Pernyataan 3** ✓ Syarat pertama (mengisi) sudah benar, jadi langsung "petir" walaupun baterai 10 %.
- **Pernyataan 4** ✓ 49 <= 15 salah; 49 < 50 benar → "kuning".
- **Pernyataan 5** ✗ Jebakan: 10 <= 15 akan diperiksa lebih dulu sehingga hasilnya "merah".

Jawaban: pernyataan 3, 4.

No. 17 Kunci: D

Konsep: System analyst menganalisis proses bisnis klien dan menerjemahkannya menjadi kebutuhan sistem (dokumen kebutuhan, diagram alur) yang menjadi dasar kerja perancang dan programmer.

Langkah:

1. Kegiatan: mempelajari alur kerja, wawancara, diagram alur proses, dokumen kebutuhan.
2. Hasil kerjanya adalah analisis kebutuhan, bukan kode, jadwal, tampilan, atau pengujian → system analyst.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

• **A.** ✗ Jebakan: project manager juga berhubungan dengan klien, tetapi tugasnya mengelola jadwal, biaya, dan tim, bukan menganalisis proses klinik.
• **B.** ✗ Programmer bekerja SETELAH dokumen kebutuhan tersedia.
• **C.** ✗ UI designer merancang tampilan, bukan alur proses dan dokumen kebutuhan.
• **D.** ✓ System analyst menganalisis proses kerja dan menyusun kebutuhan sistem.
• **E.** ✗ QA tester menguji aplikasi yang sudah dibuat, bukan menyusun kebutuhan.

Jawaban: D.

No. 18 Kunci: A, B, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Di lapisan transport, TCP menjamin data utuh dan berurutan (berkas, halaman web, obrolan), sedangkan UDP mengutamakan kecepatan tanpa kirim ulang (video/suara langsung). Protokol aplikasi seperti HTTPS berjalan di atas TCP.

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ Pada siaran langsung, menunggu kiriman ulang membuat video tersendat; UDP tidak menunggu.
• **Pernyataan 2** ✓ TCP mengirim ulang bagian yang hilang dan menjaga urutan, sehingga berkas tidak rusak.
• **Pernyataan 3** ✗ Jebakan: UDP memang lebih ringan, tetapi bagian berkas yang hilang tidak dikirim ulang sehingga berkas bisa rusak.
• **Pernyataan 4** ✓ HTTP/HTTPS berjalan di atas TCP; HTTPS mengenkripsi data masuk.
• **Pernyataan 5** ✗ TCP dan UDP adalah protokol lapisan transport, walaupun pemilihannya disesuaikan dengan kebutuhan aplikasi.

Jawaban: pernyataan 1, 2, 4.

No. 19 Kunci: A

Konsep: Etika profesi TI menuntut menghormati data pribadi orang lain dan tidak menyalahgunakan akses. Celah keamanan yang ditemukan dilaporkan secara bertanggung jawab kepada pemiliknya tanpa disebarkan atau dimanfaatkan.

Langkah:

1. Data peserta adalah data pribadi → tidak boleh diambil, disebarkan, atau diubah.
2. Pemilik situs perlu tahu agar dapat memperbaiki → laporkan secara pribadi lewat email pengelola.

Analisis setiap opsi:

• **A.** ✓ Pengelola dapat memperbaiki celah dan data peserta tidak menyebar.
• **B.** ✗ Jebakan: tampak membantu pembuktian, tetapi mengunduh semua data pribadi tetap menyalahgunakan celah.
• **C.** ✗ Tangkapan layar menyebarkan data pribadi peserta ke orang lain.
• **D.** ✗ Mengubah data orang lain merusak dan melanggar hukum, apa pun niatnya.
• **E.** ✗ Cara memakai celah yang diumumkan sebelum diperbaiki dapat dipakai orang jahat.

Jawaban: A.

No. 20 Kunci: D

Konsep: Access modifier mengatur siapa yang boleh mengakses anggota class: publik (+) semua class, terproteksi (#) class itu dan subclass-nya, privat (−) class itu saja. Pilihan disesuaikan dengan kebutuhan akses setiap atribut.

Langkah:

1. nama dibaca semua class → publik (+).

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

2. saldo tidak boleh diakses class lain termasuk subclass → privat (−).
3. biaya_admin dipakai subclass tetapi bukan class lain → terproteksi (#).

Analisis setiap opsi:
• **A.** ✗ Semua publik: saldo dan biaya_admin dapat diubah class mana pun.
• **B.** ✗ Jebakan: paling aman, tetapi nama tidak dapat dibaca class lain dan subclass tidak dapat memakai biaya_admin.
• **C.** ✗ Tertukar: saldo menjadi terbuka bagi subclass, sedangkan biaya_admin tertutup bagi RekeningPelajar.
• **D.** ✓ Sesuai ketiga kebutuhan.
• **E.** ✗ nama tertutup bagi class lain dan biaya_admin terbuka bagi semua class.

Jawaban: D.

No. 21 Kunci: B

Konsep: Polimorfisme: d.potong(40000) menjalankan versi potong() milik class setiap objek. Objek Diskon memakai versi dasar yang mengembalikan harga tanpa potongan. max(a, b) mengembalikan nilai yang lebih besar.

Langkah:
1. DiskonPersen(10): 40000 − 40000 × 10 // 100 = 40000 − 4000 = 36000.
2. DiskonTetap(50000): max(40000 − 50000, 0) = max(−10000, 0) = 0.
3. Diskon(): 40000.

Analisis setiap opsi:
• **A.** ✗ Jebakan: max(..., 0) terlupa sehingga harga menjadi negatif.
• **B.** ✓ 36000, 0, dan 40000.
• **C.** ✗ Jebakan: semua objek dianggap memakai potong() milik Diskon.
• **D.** ✗ Yang dikembalikan adalah harga setelah dipotong, bukan besar potongannya.
• **E.** ✗ Objek Diskon dasar tidak memotong harga sama sekali.

Jawaban: B.

No. 22 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Bila subclass menulis konstruktor sendiri, konstruktor superclass tidak dijalankan otomatis; perlu dipanggil dengan super().__init__(...). Method superclass tetap diwarisi.

Langkah:
1. Kipas("Kipas lab", 3) → hanya konstruktor Kipas yang berjalan → hanya ada atribut kecepatan.
2. nama dan daya tidak pernah dibuat.

Analisis setiap pernyataan:
• **Pernyataan 1** ✗ Jebakan: atribut daya dibuat di konstruktor Perangkat, yang tidak pernah dijalankan; k tidak memiliki daya.
• **Pernyataan 2** ✓ Konstruktor Kipas mengisi kecepatan = 3.
• **Pernyataan 3** ✓ info() diwarisi, tetapi di dalamnya self.nama dipakai, padahal nama tidak pernah dibuat (AttributeError).
• **Pernyataan 4** ✗ Method tetap diwarisi; yang tidak dijalankan hanya konstruktor Perangkat.
• **Pernyataan 5** ✓ Konstruktor Perangkat lalu membuat nama dan daya, sehingga info() berjalan: "Kipas lab 10 W".

Jawaban: pernyataan 2, 3, 5.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 23 Kunci: E

Konsep: Parameter konstruktor dapat memiliki nilai bawaan (`nyala=False`). Method dapat mengubah atribut objeknya; `not` membalik nilai bool.

Langkah:
1. a: nyala bawaan False → tekan() → True.
2. b: nyala True → tekan() → False → tekan() → True.
3. Keluaran: True True.

Analisis setiap opsi:
- **A.** ✗ Nilai a sudah dibalik sekali menjadi True.
- **B.** ✗ b dibalik dua kali sehingga kembali True.
- **C.** ✗ Kedua lampu berakhir dalam keadaan menyala.
- **D.** ✗ Jebakan: nyala memiliki nilai bawaan False, jadi boleh tidak dikirim.
- **E.** ✓ a dari False menjadi True; b dari True ke False lalu kembali True.

Jawaban: E.

No. 24 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Abstraksi dalam PBO berarti memodelkan objek nyata hanya dengan atribut dan method yang relevan dengan tujuan program, serta menyembunyikan detail yang tidak dibutuhkan pemakainya.

Analisis setiap pernyataan:
- **Pernyataan 1** ✗ Jebakan: data tambahan yang tidak dipakai justru membebani program dan menyulitkan perawatan.
- **Pernyataan 2** ✓ Abstraksi mengambil ciri yang relevan dengan tujuan aplikasi saja.
- **Pernyataan 3** ✓ Kode unik dibutuhkan untuk mengenali setiap buku.
- **Pernyataan 4** ✗ Menyembunyikan atribut adalah enkapsulasi; abstraksi adalah memilih dan menampilkan hal yang penting.
- **Pernyataan 5** ✓ Status dibutuhkan untuk proses meminjam dan mengembalikan.

Jawaban: pernyataan 2, 3, 5.

No. 25 Kunci: A, B, D, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Diagram struktur menunjukkan pembagian program menjadi fungsi-fungsi dan siapa memanggil siapa. Fungsi yang dipakai bersama membuat perubahan cukup dilakukan di satu tempat, tetapi perlu diuji pada semua pemanggilnya.

Analisis setiap pernyataan:
- **Pernyataan 1** ✓ hitung_pajak() dipanggil oleh hitung_total() dan laporan_harian(); satu perubahan berlaku di keduanya.
- **Pernyataan 2** ✓ Perubahan fungsi yang dipakai bersama memengaruhi semua pemanggilnya.
- **Pernyataan 3** ✗ Jebakan: input_pesanan() hanya menerima pesanan dan tidak memanggil hitung_pajak(); aturan pajak tidak ada di sana.
- **Pernyataan 4** ✓ Fungsi kecil yang terpisah dapat diuji sendiri.
- **Pernyataan 5** ✓ Satu fungsi dipanggil dari dua tempat → tidak ada duplikasi kode.

Jawaban: pernyataan 1, 2, 4, 5.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Simulasi 3 — Prediksi TKA 2026

No. 1 Kunci: C

Konsep: Data analyst mengolah data (misalnya rekaman perilaku pemain), menguji perbandingan (uji A/B), dan menyajikan temuan untuk mendukung keputusan tim lain.

Langkah:

1. Tugas: mengolah data rekaman, mencari pola berhenti, uji A/B, laporan temuan.
2. Hasilnya diberikan kepada tim perancang → orangnya bukan perancang itu sendiri, melainkan data analyst.

Analisis setiap opsi:

- **A.** ✗ Jebakan: game designer memakai temuan itu untuk merancang ulang level, tetapi bukan dia yang mengolah data pemain.
- **B.** ✗ QA mencari kesalahan program, bukan pola perilaku pemain.
- **C.** ✓ Data analyst mengolah data, menguji perbandingan, dan melaporkan temuan.
- **D.** ✗ Jebakan: backend developer menyiapkan tempat penyimpanan data, bukan menganalisis isinya.
- **E.** ✗ Community manager berinteraksi dengan pemain, bukan mengolah data rekaman permainan.

Jawaban: C.

No. 2 Kunci: C

Konsep: Harga jual jasa = total biaya + laba. Bila laba ditetapkan sebagai persentase dari total biaya, laba = persen × (semua biaya dijumlahkan).

Langkah:

1. Upah = 40 × 50.000 = Rp2.000.000.
2. Total biaya = 2.000.000 + 1.200.000 = Rp3.200.000.
3. Laba = 20 % × 3.200.000 = Rp640.000.
4. Harga = 3.200.000 + 640.000 = Rp3.840.000.

Analisis setiap opsi:

- **A.** ✗ Laba belum ditambahkan; ini baru total biaya.
- **B.** ✗ Jebakan: laba 20 % dihitung dari upah saja (2.400.000 + 1.200.000).
- **C.** ✓ Total biaya Rp3.200.000 ditambah laba Rp640.000.
- **D.** ✗ Jebakan: laba dianggap 20 % dari harga jual (3.200.000 ÷ 0,8), bukan dari biaya.
- **E.** ✗ Laba dihitung dua kali: dari upah, lalu sekali lagi dari total.

Jawaban: C.

No. 3 Kunci: A

Konsep: Rapat harian dipakai untuk menemukan hambatan (blocker). Tugas ketua tim adalah segera menghilangkan hambatan, misalnya menghubungi pihak yang ditunggu, sambil menjaga pekerjaan tetap berjalan dengan cara sementara yang aman.

Langkah:

1. Hambatan: kunci uji dari klien belum ada → Sari tertahan 3 hari.
2. Fitur pembayaran adalah inti demo 5 hari lagi → hambatan harus segera dibuka.
3. Ketua tim menghubungi klien; Sari tetap maju dengan data tiruan (mock) sampai kunci datang.

Analisis setiap opsi:

- **A.** ✓ Hambatan langsung ditangani dan pekerjaan Sari tetap berjalan.
- **B.** ✗ Jebakan: Sari tidak menganggur, tetapi hambatan dibiarkan dan fitur utama demo terancam tidak siap.
- **C.** ✗ Fitur utama demo justru dihilangkan.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

- **D.** ✗ Memakai kunci produksi milik pihak lain melanggar keamanan dan etika.
- **E.** ✗ Kode pembayaran tidak bermasalah; yang kurang adalah kunci akses untuk menguji.

Jawaban: A.

No. 4 Kunci: 1B 2S 3B

Konsep: Model TCP/IP: lapisan akses jaringan (kabel, Wi-Fi, switch), internet (alamat IP dan perutean oleh router), transport (TCP/UDP dan nomor port), aplikasi (HTTP, DNS, dll.).

Analisis setiap pernyataan:

- **Pernyataan 1 → Benar.** Penentuan rute antarjaringan memakai alamat IP, fungsi lapisan internet.
- **Pernyataan 2 → Salah.** Jebakan: router memilih jalur dari alamat IP; nomor port dipakai komputer tujuan untuk menyerahkan data ke program yang tepat.
- **Pernyataan 3 → Benar.** Lapisan akses jaringan mengurus pengiriman fisik lewat kabel atau gelombang radio.

Jawaban: Benar, Salah, Benar.

No. 5 Kunci: A, B, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Budaya mutu dalam pengujian menuntut laporan bug yang dapat ditindaklanjuti: judul jelas, langkah memunculkan ulang, hasil yang diharapkan dan yang terjadi, lingkungan (perangkat, versi), bukti (tangkapan layar), dan tingkat keparahan.

Analisis setiap pernyataan:

- **Pernyataan 1** ✓ Judul yang jelas langsung menunjukkan bagian aplikasi dan gejalanya.
- **Pernyataan 2** ✓ Langkah rinci memungkinkan programmer memunculkan ulang bug.
- **Pernyataan 3** ✗ Laporan dibuat saat bug ditemukan; tanggal yang ada sudah benar.
- **Pernyataan 4** ✓ Bug sering hanya muncul pada perangkat atau versi tertentu.
- **Pernyataan 5** ✗ Jebakan: programmer memang ikut menilai, tetapi tingkat keparahan dari penguji membantu tim menentukan prioritas.

Jawaban: pernyataan 1, 2, 4.

No. 6 Kunci: D

Konsep: Mengangkat beban berat dengan aman: gunakan alat bantu (troli) bila ada, angkat berdua untuk beban berat, tekuk lutut dan jaga punggung lurus, dan jangan menarik perangkat lewat kabelnya.

Langkah:

1. Beban 30 kg berisiko mencederai punggung bila diangkat sendiri.
2. Troli tersedia dan ada dua teknisi → pakai troli atau angkat berdua dengan teknik yang benar.

Analisis setiap opsi:

- **A.** ✗ Membungkuk saat mengangkat beban berat dapat mencederai tulang belakang.
- **B.** ✗ Kabel dapat putus dan UPS dapat rusak atau melukai kaki.
- **C.** ✗ Beban berat di satu bahu membuat tubuh tidak seimbang dan mudah cedera.
- **D.** ✓ Alat bantu atau dua orang dengan teknik angkat yang benar mengurangi risiko cedera.
- **E.** ✗ Jebakan: kebiasaan menahan napas tidak membuat aman dan justru membebani tubuh.

Jawaban: D.

No. 7 Kunci: A, C, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Router DHCP membagikan alamat dari rentangnya kepada perangkat yang meminta. Alamat statis yang berada di dalam rentang itu dapat terbagi juga ke perangkat lain sehingga terjadi konflik. Alamat statis sebaiknya di luar rentang DHCP.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ Keduanya 192.168.7.50.

• **Pernyataan 2** ✗ Jebakan: alamat statis justru sebaiknya di luar rentang DHCP; .10 tidak bentrok dengan apa pun.

• **Pernyataan 3** ✓ .50 berada di rentang .20–.100, sehingga router tidak tahu alamat itu sudah dipakai printer.

• **Pernyataan 4** ✓ Alamat di luar rentang tidak akan dibagikan ke perangkat lain.

• **Pernyataan 5** ✗ Laptop-04 justru ikut bentrok dengan printer.

Jawaban: pernyataan 1, 3, 4.

No. 8 Kunci: B, C, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Aset fisik yang menyimpan data (laptop, hard disk, ponsel) diperlakukan juga sebagai pembawa aset digital. Sebelum dilepas: cadangkan data penting, keluarkan dari semua akun, lalu hapus data secara aman (ditimpa) agar tidak dapat dipulihkan.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: berkas yang dihapus biasa masih dapat dipulihkan dengan aplikasi pemulih data.

• **Pernyataan 2** ✓ Data penting diselamatkan dulu sebelum media dibersihkan.

• **Pernyataan 3** ✓ Penimpaan membuat data lama tidak dapat dipulihkan.

• **Pernyataan 4** ✓ Mencegah pembeli masuk ke akun guru lewat perangkat yang masih tercatat.

• **Pernyataan 5** ✗ Data pembeli dan akun guru dapat jatuh ke tangan orang lain.

Jawaban: pernyataan 2, 3, 4.

No. 9 Kunci: 1S 2B 3B

Konsep: Penomoran versi yang konsisten dan catatan perubahan (changelog) adalah bagian dari pengelolaan dokumentasi proyek, sehingga setiap versi dapat dikenali dan ditelusuri perubahannya.

Analisis setiap pernyataan:

• **Pernyataan 1 → Salah.** Perbaikan bug menaikkan angka ketiga: 2.3.2.

• **Pernyataan 2 → Benar.** Fitur baru tanpa mengubah cara pakai lama → angka kedua naik, angka ketiga kembali 0.

• **Pernyataan 3 → Benar.** Changelog membuat tim dan pengguna tahu apa yang berubah di setiap versi.

Jawaban: Salah, Benar, Benar.

No. 10 Kunci: B, D

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Budaya kerja profesional di industri perangkat lunak meliputi komunikasi lewat saluran yang disepakati, menghormati jam kerja, memberi masukan yang sopan dan berdasar, serta menghargai kontribusi rekan.

Analisis setiap pernyataan:

• **Pernyataan 1** ✗ Jebakan: tujuannya perbaikan, tetapi komentar tidak sopan dan tidak menjelaskan apa yang salah.

• **Pernyataan 2** ✓ Komentar sopan, menunjuk lokasi, dan menjelaskan alasannya.

• **Pernyataan 3** ✗ Melanggar kesepakatan jam kerja dan kanal kerja tim.

• **Pernyataan 4** ✓ Mengakui kontribusi orang lain adalah sikap jujur dan menghargai tim.

• **Pernyataan 5** ✗ Tidak etis dan tidak menyelesaikan masalah; pembahasan dilakukan di kanal tim bersama yang bersangkutan.

Jawaban: pernyataan 2, 4.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 11 Kunci: D

Konsep: Folder proyek berisi berkas sumber (harus dijaga), berkas yang dapat dibuat ulang (build, cache), dan riwayat. Saat ruang penuh, yang aman dihapus adalah berkas yang dapat dibuat ulang.

Langkah:

1. Folder yang dapat dibuat ulang: build/ (6,5 GB) dan cache/ (2,1 GB).
2. Jumlah: 6,5 + 2,1 = 8,6 GB ≥ 8 GB → cukup.
3. aset/ tidak boleh dihapus karena tidak punya salinan lain.

Analisis setiap opsi:

- **A.** ✗ aset/ adalah satu-satunya salinan, dan 5,3 GB belum cukup.
- **B.** ✗ Ruang yang didapat terlalu kecil, dan riwayat lokal ikut hilang.
- **C.** ✗ Jebakan: build/ memang paling besar, tetapi 6,5 GB belum memenuhi kebutuhan 8 GB.
- **D.** ✓ Keduanya dapat dibuat ulang dan jumlahnya memenuhi 8 GB.
- **E.** ✗ Jebakan: ruangnya cukup, tetapi aset/ yang tidak punya salinan ikut hilang.

Jawaban: D.

No. 12 Kunci: E

Konsep: Setiap proyek menetapkan versi bahasa dan pustaka yang dibutuhkan (biasanya di README). Lingkungan pengembangan harus disesuaikan dengan versi itu; sintaks baru tidak dikenali oleh versi lama.

Langkah:

1. match–case baru ada sejak Python 3.10; laptop Yoga memakai 3.8 → SyntaxError.
2. Kode tim benar; yang perlu disesuaikan adalah lingkungan Yoga.
3. Pasang Python 3.11 sesuai README dan jalankan program dengan versi itu.

Analisis setiap opsi:

- **A.** ✗ Jebakan: program bisa jalan di laptop Yoga, tetapi kode tim berubah sepihak dan tidak sesuai kesepakatan.
- **B.** ✗ Editor tidak menentukan versi Python yang menjalankan program.
- **C.** ✗ Menyalin folder instalasi sering tidak lengkap dan tidak terdaftar dengan benar di sistem.
- **D.** ✗ Hak administrator tidak mengubah kemampuan Python 3.8 membaca sintaks baru.
- **E.** ✓ Versi sesuai README sehingga sintaks match–case dikenali.

Jawaban: E.

No. 13 Kunci: B

Konsep: Pengelolaan berkas media proyek memisahkan berkas yang sedang dikerjakan dari berkas final yang dipakai program, memberi nomor versi, dan mencatat perubahan, sehingga pemakai tidak terganggu dan versi lama dapat dipulihkan.

Langkah:

1. Masalah 1: berkas yang dipakai program berubah tanpa pemberitahuan.
2. Masalah 2: versi lama hilang karena ditimpa.
3. Folder kerja vs final + nomor versi + catatan perubahan menyelesaikan keduanya.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap opsi:

• **A.** ✗ Revisi tetap dibutuhkan; melarangnya menghambat proyek.
• **B.** ✓ Programmer hanya memakai berkas final, dan setiap versi tersimpan serta tercatat.
• **C.** ✗ Jebakan: salinan pribadi melindungi sebentar, tetapi versi menjadi berbeda-beda di setiap laptop dan riwayat tetap tidak tercatat.
• **D.** ✗ Jebakan: tidak ada yang tertimpa, tetapi nama acak membuat versi tidak dapat diurutkan atau dikenali.
• **E.** ✗ Desainer yang membuat gambar justru tidak dapat menyimpan hasil kerjanya.

Jawaban: B.

No. 14 Kunci: A, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Operator not membalik nilai bool; or bernilai benar bila salah satu bagiannya benar. return menghentikan fungsi, sehingga urutan pemeriksaan menentukan hasil.

Analisis setiap pernyataan:

• **Pernyataan 1** ✓ Sehat → lolos syarat pertama; 16 >= 17 salah, tetapi izin_ortu True → "Ya".
• **Pernyataan 2** ✗ Jebakan: tidak sehat → syarat pertama langsung mengembalikan "Tidak".
• **Pernyataan 3** ✗ Umur kurang dari 17 dan tanpa izin → kedua bagian or salah → "Tidak".
• **Pernyataan 4** ✗ 17 >= 17 benar → "Ya".
• **Pernyataan 5** ✓ return di syarat pertama menghentikan fungsi sebelum syarat lain diperiksa.

Jawaban: pernyataan 1, 5.

No. 15 Kunci: A

Konsep: Pada arsitektur client–server berlapis, client (ponsel) hanya berkomunikasi dengan server web/API lewat HTTPS. Server API yang mengambil data dari basis data, sehingga basis data dan kata sandinya tidak terbuka ke internet.

Langkah:

1. Gambar: ponsel → internet → server web/API (port 443) → basis data.
2. Basis data (port 3306) hanya menerima koneksi dari server web/API.
3. Jadi ponsel harus lewat server web/API; usulan menyambung langsung melanggar aturan itu dan berbahaya.

Analisis setiap opsi:

• **A.** ✓ Menurut gambar, port 3306 hanya menerima koneksi dari server web/API; ponsel berkomunikasi lewat HTTPS port 443 ke server web/API.
• **B.** ✗ Jebakan: gambar menyatakan port 3306 hanya menerima koneksi dari server web/API, bukan dari internet.
• **C.** ✗ Gambar menunjukkan kata sandi basis data disimpan di server web/API; menaruhnya di aplikasi ponsel membuatnya mudah dibongkar.
• **D.** ✗ Jebakan: kesimpulannya benar, tetapi alasannya salah baca gambar; firewall berada di antara internet dan server web/API.
• **E.** ✗ Salah baca gambar: port 443 milik server web/API, sedangkan basis data memakai port 3306.

Jawaban: A.

No. 16 Kunci: C

Konsep: super().__init__() menjalankan konstruktor superclass pada objek yang sama. Atribut yang diisi superclass akan menimpa nilai yang diisi subclass sebelumnya, sehingga pengisian nilai khusus subclass sebaiknya dilakukan SETELAH super().__init__().

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Langkah:

1. self.hp = 500.
2. super().__init__("Arya") → self.nama = "Arya", self.hp = 100 (menimpa 500).
3. self.mahkota = True.
4. Keluaran: Arya 100 True.

Analisis setiap opsi:

- **A.** ✗ Jebakan: nilai 500 tertimpa oleh konstruktor Tokoh yang dijalankan sesudahnya.
- **B.** ✗ Mengisi atribut sebelum memanggil super() tidak menimbulkan error.
- **C.** ✓ hp tertimpa menjadi 100; nama dan mahkota terisi.
- **D.** ✗ Nilai hp tidak dijumlahkan, melainkan ditimpa.
- **E.** ✗ nama diisi oleh konstruktor Tokoh melalui super().

Jawaban: C.

No. 17 Kunci: E

Konsep: Syarat while diperiksa sebelum setiap putaran; perulangan berhenti segera setelah syarat salah. Operator // menghasilkan hasil bagi bulat (dibulatkan ke bawah).

Langkah:

1. Putaran 1: total = 100, harga = 110, tahun = 1.
2. Putaran 2: total = 210, harga = 121, tahun = 2.
3. Putaran 3: total = 331, harga = 133, tahun = 3.
4. Putaran 4: total = 464, harga = 146, tahun = 4.
5. Putaran 5: total = 610, harga = 160, tahun = 5 → 160 < 150 salah → berhenti. Keluaran: 5 610.

Analisis setiap opsi:

- **A.** ✗ Perulangan dianggap berhenti saat harga 146, padahal 146 masih kurang dari 150.
- **B.** ✗ Jebakan: total dianggap ditambah SETELAH harga naik: 110 + 121 + 133 + 146 + 160.
- **C.** ✗ Jebakan: dianggap ada satu putaran lagi, sehingga harga 160 ikut dijumlahkan.
- **D.** ✗ Yang dicetak adalah total, bukan harga terakhir.
- **E.** ✓ Lima putaran; total 100 + 110 + 121 + 133 + 146 = 610.

Jawaban: E.

No. 18 Kunci: D

Konsep: Enkapsulasi menyembunyikan data penting dan mengizinkan perubahan hanya lewat method yang memeriksa aturan. Dengan begitu, aturan "koin tidak boleh negatif" ditulis sekali dan selalu dijalankan.

Langkah:

1. Akar masalah: koin dapat diubah langsung dari mana saja tanpa pemeriksaan.
2. Perbaikan: koin dibuat privat, perubahan hanya lewat method seperti bayar(harga) yang menolak bila saldo kurang.

Analisis setiap opsi:

- **A.** ✗ Subclass tidak mencegah 14 tempat itu mengubah koin langsung.
- **B.** ✗ Masalahnya bukan perilaku per jenis barang, melainkan perubahan koin tanpa pemeriksaan.
- **C.** ✗ Variabel global memperparah masalah karena koin makin mudah diubah dari mana saja.
- **D.** ✓ Aturan saldo berada di satu method dan tidak dapat dilewati.
- **E.** ✗ Jebakan: masalah dapat berkurang, tetapi aturan tersalin 14 kali dan mudah terlupa lagi saat ada tempat baru.

Jawaban: D.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 19 Kunci: B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Polimorfisme: pemanggilan method selalu memakai versi milik class objek yang sebenarnya, termasuk pemanggilan self.method() dari dalam method warisan. Method yang tidak ditulis ulang memakai versi superclass.

Langkah:
1. Laporan() → [Laporan].
2. LaporanHarian() → cetak warisan, judul sendiri → [Harian].
3. LaporanBulanan() → cetak sendiri, judul warisan → <<Laporan>>.

Analisis setiap pernyataan:
• **Pernyataan 1 ✗** Jebakan: LaporanBulanan tidak menulis ulang judul(), sehingga memakai milik Laporan: <<Laporan>>.
• **Pernyataan 2 ✓** LaporanHarian memakai cetak() milik Laporan, dan self.judul() memanggil judul() miliknya sendiri.
• **Pernyataan 3 ✓** cetak() diwarisi; judul() ditulis ulang.
• **Pernyataan 4 ✗** LaporanBulanan menulis ulang cetak(), sehingga barisnya diawali <<.
• **Pernyataan 5 ✓** cetak() milik LaporanBulanan dengan judul() warisan dari Laporan.

Jawaban: pernyataan 2, 3, 5.

No. 20 Kunci: E

Konsep: `insert(i, x)` menyisipkan x di indeks i; `remove(x)` menghapus x pertama yang ditemukan; `append(x)` menambah di akhir; `index(x)` memberi indeks kemunculan pertama x.

Langkah:
1. Awal: [Hujan, Pelangi, Senja].
2. insert(1, "Fajar") → [Hujan, Fajar, Pelangi, Senja].
3. remove("Senja") → [Hujan, Fajar, Pelangi].
4. append("Pelangi") → [Hujan, Fajar, Pelangi, Pelangi].
5. index("Pelangi") → 2 (kemunculan pertama).

Analisis setiap opsi:
• **A. ✗** Jebakan: index() dianggap memberi posisi kemunculan terakhir.
• **B. ✗** insert(1, ...) menyisipkan di indeks 1, bukan di depan.
• **C. ✗** Lagu "Pelangi" yang ditambahkan dengan append terlupa.
• **D. ✗** Jebakan: insert(1, ...) dianggap menyisipkan sesudah elemen indeks 1.
• **E. ✓** Isi list dan indeks kemunculan pertama "Pelangi" sudah benar.

Jawaban: E.

No. 21 Kunci: A, B, C, E

Angka pilihan di pembahasan: 1 = A, 2 = B, 3 = C, 4 = D, 5 = E.

Konsep: Desain modular memisahkan logika (perhitungan) dari masukan dan keluaran. Fungsi yang menerima parameter dan mengembalikan nilai mudah dipakai ulang, diuji, dan diubah.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Analisis setiap pernyataan:
• **Pernyataan 1** ✓ Fungsi perhitungan murni dapat dipakai di mana saja dan mudah diuji.
• **Pernyataan 2** ✓ Masukan dan keluaran jelas sehingga mudah diuji otomatis.
• **Pernyataan 3** ✓ Halaman web memakai fungsi perhitungan dengan cara masukan dan tampilannya sendiri.
• **Pernyataan 4** ✗ Jebakan: tampak mandiri, tetapi fungsi yang meminta input() tidak dapat dipakai halaman web atau uji otomatis.
• **Pernyataan 5** ✓ Satu aturan berada di satu fungsi, sehingga perubahannya terpusat.

Jawaban: pernyataan 1, 2, 3, 5.

No. 22 Kunci: B

Konsep: `int()` mengubah teks angka menjadi bilangan bulat. Operator + pada bilangan menjumlahkan, sedangkan pada teks menggabungkan.

Langkah:
1. int("12") + 3 = 12 + 3 = 15.
2. "12" + "3" = "123" (penggabungan teks).
3. Keluaran: 15 123.

Analisis setiap opsi:
• **A.** ✗ Urutannya terbalik: bagian pertama adalah penjumlahan bilangan.
• **B.** ✓ Penjumlahan bilangan 15, lalu gabungan teks 123.
• **C.** ✗ Jebakan: "12" + "3" dianggap dijumlahkan, padahal keduanya teks.
• **D.** ✗ Jebakan: int() dianggap tidak mengubah tipe data.
• **E.** ✗ int() sudah mengubah "12" menjadi bilangan, dan teks + teks juga sah.

Jawaban: B.

No. 23 Kunci: 1S 2B 3S

Konsep: Enkapsulasi bukan hanya membuat atribut privat, tetapi memastikan setiap jalan untuk mengubahnya menjalankan aturan yang sama. Getter membuka akses baca tanpa akses ubah.

Analisis setiap pernyataan:
• **Pernyataan 1 → Salah.** Jebakan: 30 berada di luar rentang 34–42 sehingga ditolak; suhu tetap 36.5.
• **Pernyataan 2 → Benar.** Getter hanya mengembalikan nilai.
• **Pernyataan 3 → Salah.** Method itu menjadi jalan pintas yang melewati aturan, sehingga tujuan enkapsulasi hilang.

Jawaban: Salah, Benar, Salah.

No. 24 Kunci: A

Konsep: Class adalah rancangan; objek adalah wujud yang dibuat dari class. Satu class dapat dipakai untuk membuat banyak objek dengan nilai atribut berbeda.

Langkah:
1. Class: Siswa dan Kelas → 2.
2. Objek: 1 objek Kelas + 32 objek Siswa → 33.

Analisis setiap opsi:
• **A.** ✓ 2 class (Siswa, Kelas) dan 33 objek (1 kelas + 32 siswa).
• **B.** ✗ Jebakan: setiap siswa dianggap class tersendiri.
• **C.** ✗ Jebakan: setiap class dianggap hanya menghasilkan satu objek.
• **D.** ✗ Class dan objek dihitung bersama-sama secara keliru.
• **E.** ✗ Siswa dan Kelas adalah dua class berbeda.

Jawaban: A.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

No. 25 Kunci: B

Konsep: Variabel yang diberi nilai di dalam fungsi adalah variabel lokal; ia tidak mengubah variabel bernama sama di luar fungsi (global). Fungsi yang hanya membaca sebuah nama tanpa memberinya nilai memakai variabel global.

Langkah:

1. harga_ultah: diskon lokal = 20 → 50000 − 10000 = 40000.
2. harga_biasa: tidak memberi nilai pada diskon → memakai diskon global 10 → 50000 − 5000 = 45000.
3. diskon global tetap 10. Keluaran: 40000 45000 10.

Analisis setiap opsi:

- **A.** ✗ Jebakan: harga_ultah dianggap mengubah diskon global menjadi 20, sehingga harga_biasa ikut memakai 20 %.
- **B.** ✓ harga_ultah memakai diskon lokal 20, harga_biasa memakai diskon global 10, dan diskon global tetap 10.
- **C.** ✗ Jebakan: kedua harga benar, tetapi diskon global dianggap berubah menjadi 20.
- **D.** ✗ harga_ultah dianggap memakai diskon global 10.
- **E.** ✗ Fungsi boleh membaca variabel global yang tidak diberinya nilai.

Jawaban: B.

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Penutup

Kamu sudah sampai di akhir buku ini. Itu sendiri adalah bukti bahwa kamu mampu menepati janji belajarmu.

Lima strategi yang paling penting untuk TKA PPLG:

1. **Baca pertanyaannya dulu, baru ceritanya.** Kasus dunia kerja sering panjang; kalimat pertanyaan memberi tahu petunjuk mana yang perlu kamu cari.
2. **Telusuri kode dengan tabel jejak.** Tulis nilai awal, perbarui nilai variabel setiap baris atau putaran, dan jangan menebak keluaran dari "maksud" programnya.
3. **Cari penyebabnya, bukan gejalanya.** Pilihan yang benar menyasar akar masalah: izin yang kurang, port yang dipakai, kunci yang bocor, atau urutan syarat yang terbalik.
4. **Periksa nilai batas dan arah.** `>=` atau `>`, angka akhir `range` yang tidak ikut, indeks yang dimulai dari 0, dan `[baris][kolom]` adalah jebakan yang berulang di semua bab.
5. **Pilih tindakan yang jujur, terbuka, dan melibatkan tim.** Melapor kesalahan, meminta izin, menulis satu kali dan memakainya berulang adalah kebiasaan profesional yang selalu menjadi jawaban terbaik.

Sepekan sebelum ujian:

- Kerjakan satu Simulasi TKA pada jam yang sama dengan jadwal ujianmu, lengkap dengan waktunya.
- Ulangi soal yang pernah kamu jawab salah, lalu baca lagi "Strategi dan jebakan" di sub babnya. Pada soal pilih lebih dari satu, nilai setiap pilihan sendiri-sendiri.
- Ketik ulang tanpa melihat buku contoh-contoh kode penting: perulangan dengan `range`, class dengan `__init__`, pewarisan dengan `super()`, dan perulangan polimorfik.
- Jangan memaksakan materi baru pada malam terakhir. Siapkan perlengkapan ujian sehari sebelumnya, dan beristirahatlah dengan cukup.

Teruslah belajar lewat LMS Next Level Study; pindai kode QR di setiap bab.

Program yang baik dibangun dari ribuan keputusan kecil yang dikerjakan dengan jujur, bahkan saat tidak ada yang melihat. Bawalah kejujuran dan ketelitian yang sama ke ruang ujian dan ke dunia kerja. Ilmu yang kamu kumpulkan hari ini adalah bekal yang kelak kamu persembahkan untuk keluarga, untuk orang-orang di sekitarmu, dan untuk negeri ini. Bekerjalah dengan sungguh-sungguh dan jujur, sepenuh hati dan segenap tenaga. Apa pun hasil ujianmu nanti, kesungguhan yang kamu bangun selama belajar akan tetap menjadi milikmu.

Selamat berjuang.

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Handaka Lumu

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Lampiran

Lampiran 1 · Glosarium

Istilah	Arti singkat	Dibahas di
Technopreneurship	kewirausahaan yang memakai teknologi, misalnya membuat aplikasi atau gim	1.1
QA tester	penguji perangkat lunak yang melaporkan bug dengan langkah memunculkannya	1.1
MVP	versi awal produk dengan fitur inti saja, untuk diuji pada sedikit pengguna	1.1
Business Model Canvas	peta sembilan blok model bisnis	1.1
Biaya tetap / biaya variabel	biaya yang sama berapa pun yang terjual / biaya yang bertambah bersama jumlah terjual	1.1
Titik impas	jumlah penjualan ketika pendapatan sama dengan biaya total	1.1
Jalur kritis	jalur kegiatan terpanjang yang menentukan lama proyek	1.2
Waterfall / agile	model pengembangan berurutan / berputaran pendek	1.2
Scrum, sprint, backlog	kerangka agile; putaran kerja tetap; daftar kebutuhan berprioritas	1.2
Scope creep	pekerjaan yang bertambah tanpa kesepakatan ruang lingkup baru	1.2
Kriteria penerimaan	pernyataan ya/tidak tentang kapan sebuah fitur diterima klien	1.2
Definisi selesai	syarat yang disepakati tim agar sebuah pekerjaan disebut selesai	1.2
Code review	tinjauan kode oleh rekan sebelum digabung	1.2
PDCA	siklus plan–do–check–act untuk perbaikan berkelanjutan	1.2
K3LH	keselamatan, kesehatan kerja, dan lingkungan hidup	2.1
Ergonomi	penataan kerja agar sesuai tubuh sehingga nyaman dan tidak cedera	2.1
APAR	alat pemadam api ringan	2.1
Hierarki pengendalian	urutan pengendalian risiko: eliminasi, substitusi, rekayasa, administratif, APD	2.1
E-waste	limbah perangkat elektronik	2.1
5R	ringkas, rapi, resik, rawat, rajin	2.1
Aset fisik / aset digital	benda berwujud yang bernilai / data atau hak pakai yang bernilai	2.2
Inventaris	daftar aset beserta kode, kondisi, dan penanggung jawab	2.2
Penyusutan garis lurus	(harga perolehan − nilai sisa) ÷ umur manfaat	2.2
Lisensi	izin pemakaian suatu karya menurut syarat tertentu	2.2

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Creative Commons (BY, NC, ND, SA)	keluarga lisensi karya kreatif: atribusi, non-komersial, tanpa turunan, bagi serupa	2.2
Open source	perangkat lunak dengan kode sumber terbuka, mis. lisensi MIT	2.2
Hak akses minimum	memberi akses hanya sebesar yang dibutuhkan	2.2
Git	sistem kontrol versi untuk mencatat dan menggabungkan perubahan kode	2.3
Commit, branch, merge	catatan perubahan; jalur kerja terpisah; penggabungan jalur	2.3
Pull request	permintaan menggabungkan cabang yang ditinjau rekan	2.3
.gitignore	daftar berkas yang tidak dilacak Git	2.3
Semantic versioning	nomor versi MAJOR.MINOR.PATCH	2.3
Cadangan 3-2-1	3 salinan, 2 jenis media, 1 salinan di lokasi lain	2.3
Path absolut / relatif	alamat berkas dari root / dari folder aktif	3.1
chmod	perintah mengubah izin berkas	3.1
PATH	variabel lingkungan berisi daftar folder pencarian program	3.1
Virtual environment (venv)	lingkungan Python terpisah untuk satu proyek	3.1
Port	nomor yang memilih program di sebuah alamat IP	3.1, 3.4
Switch / router / access point	menyatukan satu jaringan / menghubungkan jaringan berbeda / memancarkan Wi-Fi	3.2
Alamat IP, subnet mask	alamat perangkat; penanda bagian jaringan dan bagian host	3.2
ID jaringan, broadcast	alamat pertama dan terakhir sebuah jaringan, tidak dipakai host	3.2
Gateway	alamat router untuk keluar dari jaringan	3.2
DHCP / DNS	pemberi alamat IP otomatis / penerjemah nama menjadi alamat IP	3.2, 3.4
TCP/IP	keluarga protokol empat layer yang dipakai internet	3.3
Enkapsulasi (jaringan)	pembungkusan data dengan header di tiap layer	3.3
TCP / UDP	protokol transport andal / ringan	3.3
HTTPS, SSH, SFTP	protokol terenkripsi untuk web, akses jarak jauh, dan unggah berkas	3.3
Client–server	pola komunikasi: klien meminta, server menjawab	3.4
Kode status HTTP	angka tiga digit hasil permintaan (2xx, 4xx, 5xx)	3.4
REST API, JSON	layanan berbasis endpoint HTTP; format data teks terstruktur	3.4
Firewall	penyaring lalu lintas menurut aturan alamat dan port	3.4
Variabel, tipe data	nama untuk nilai; jenis nilai (int, float, str, bool)	4.1
Konversi tipe	mengubah tipe nilai, mis. int("15")	4.1
List, tuple, set, dictionary	kumpulan berurutan; tetap; unik; pasangan kunci–nilai	4.1
Indeks, irisan	nomor posisi (mulai 0); potongan [a:b]	4.1

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Diagram alir	gambar langkah program dengan simbol baku	4.2
if, elif, else	percabangan bersyarat	4.2
Operator logika	and, or, not	4.2
Nilai batas	nilai tepat pada ambang kondisi	4.2
for, while	perulangan menurut urutan; perulangan menurut syarat	4.3
range	deretan bilangan; angka akhir tidak ikut	4.3
break / continue	menghentikan perulangan / melompat ke putaran berikutnya	4.3
Tabel jejak	tabel nilai variabel tiap putaran untuk menelusuri program	4.3
Fungsi, parameter, argumen	blok perintah bernama; nama di definisi; nilai saat dipanggil	4.4
return / None	mengirim nilai keluar fungsi; hasil fungsi tanpa return	4.4
Lingkup lokal / global	variabel di dalam fungsi / di luar fungsi	4.4
Modul, import	berkas .py; cara memakai isi modul lain	4.4
Kohesi / kopling	seberapa terfokus satu tugas / seberapa bergantung antarmodul	4.4
DRY	jangan menulis logika yang sama di banyak tempat	4.4
Class, object	cetak biru; hasil cetakannya	5.1
Atribut, method	data milik objek; tindakan objek	5.1
init, self	konstruktor; objek yang sedang dipakai	5.1
Abstraksi	menyembunyikan rincian rumit di balik antarmuka sederhana	5.1
Enkapsulasi (OOP)	membungkus data dan method serta melindungi data	5.1, 5.2
Public, protected, private	tingkat akses: bebas, kesepakatan, tertutup	5.2
Getter / setter	method pembaca nilai / pengubah nilai dengan validasi	5.2
Pewarisan, superclass, subclass	class mengambil milik class lain; induk; turunan	5.3
super()	memanggil method atau konstruktor class induk	5.3
Is-a / has-a	hubungan "adalah" (pewarisan) / "memiliki" (atribut)	5.3
Overriding	subclass menulis ulang method induk	5.3, 5.4
Polymorphism	satu pemanggilan, perilaku berbeda menurut jenis objek	5.4

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Lampiran 2 · Daftar sumber

Buku ini tidak memuat data statistik nyata. Semua tokoh, lembaga, usaha, dan angka dalam soal dan contoh adalah rekaan untuk latihan dan diberi keterangan di soalnya. Rujukan resmi dan acuan teknis buku ini:

Sumber	Dipakai untuk
Pusmendik Kemendikdasmen, halaman kisi-kisi TKA Pengembangan Perangkat Lunak dan Gim SMA/MA/SMK/MAK (`pusmendik.kemendikdasmen.go.id/tka/tka/view/mata-pelajaran-pilihan/sma/pengembangan-perangkat-lunak-dan-gim-smk-mak`, diakses 16 Oktober 2026)	elemen, sub-elemen, level kognitif, bentuk soal, enam contoh soal resmi
Soal asli TKA SMK 2025 Projek Kreatif dan Kewirausahaan (Pusat Asesmen Pendidikan, Kemendikdasmen)	acuan bentuk dan kesulitan soal; belum ada soal asli untuk PPLG
Dokumentasi resmi bahasa Python 3 (python.org)	sintaks dan perilaku bahasa pada contoh kode (semua contoh dijalankan sebelum dimuat)
Standar umum bidang TI: model TCP/IP dan OSI, HTTP, DNS, versi semantik, Business Model Canvas, Scrum	istilah dan konsep pada Bab 1 dan Bab 3
Warna dan bentuk rambu keselamatan (kesepakatan umum, mengacu pada standar ISO 7010)	Gambar 2.4 (rambu digambar sendiri oleh Tim Next Level Study Indonesia)

Semua gambar materi digambar sendiri oleh Tim Next Level Study Indonesia; daftar lengkapnya ada pada "Daftar sumber gambar".

Lampiran 3 · Lembar jawaban kosong

Lembar ini boleh difotokopi. Untuk soal PG, lingkari satu huruf. Untuk pilih lebih dari satu (MCMA), lingkari semua huruf yang benar. Untuk Benar/Salah, tulis B atau S untuk setiap pernyataan.

No	Jawaban	No	Jawaban	No	Jawaban
1	A B C D E	10	A B C D E	19	A B C D E
2	A B C D E	11	A B C D E	20	A B C D E
3	A B C D E	12	A B C D E	21	A B C D E
4	A B C D E	13	A B C D E	22	A B C D E
5	A B C D E	14	A B C D E	23	A B C D E
6	A B C D E	15	A B C D E	24	A B C D E
7	A B C D E	16	A B C D E	25	A B C D E
8	A B C D E	17	A B C D E		
9	A B C D E	18	A B C D E		

Skor: _____ / _____ · Waktu yang dipakai: _____ menit

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Lampiran 4 · Catatan edisi

Edisi 1 (2026/2027), dibagikan gratis untuk pelajar Indonesia sebagai dedikasi Next Level Study Indonesia untuk bangsa; boleh diperbanyak dan dibagikan untuk belajar, tetapi tidak untuk diperjualbelikan. Semua soal latihan, kuis, dan simulasi TKA diambil dari Bank Soal NLS dengan kode yang sama dengan LMS. Belum ada soal asli TKA PPLG; bentuk soal mengikuti soal asli TKA SMK 2025 Projek Kreatif dan Kewirausahaan dan contoh resmi Pusmendik. Bila kamu menemukan kesalahan, sampaikan kepada pengajarmu agar dapat diperbaiki pada edisi berikutnya.

Daftar sumber gambar

Gambar materi yang diambil dari sumber berlisensi terbuka dipakai sesuai lisensinya dengan menyebut pembuatnya; gambar lain digambar oleh tim Next Level Study Indonesia.

Gambar	Judul	Pembuat	Lisensi	Sumber
Gambar 1.1	Peta profesi dalam alur pengembangan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.2	Alur kerja satu tim studio gim	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.3	Business Model Canvas	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.4	Kurva biaya dan pendapatan, titik impas	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.5	Peta konsep profesi dan peluang usaha PPLG	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.6	Lima fase manajemen proyek	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.7	Perbandingan waterfall dan agile	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.8	Satu siklus scrum	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.9	Jaringan kerja dan jalur kritis	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.10	Siklus PDCA	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 1.11	Peta konsep manajemen proyek dan budaya mutu	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.1	Postur kerja ergonomis	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.2	Segitiga api dan pilihan APAR	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.3	Hierarki pengendalian risiko	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.4	Empat jenis rambu K3	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.5	Alur penanganan kesalahan setelah rilis	Tim Next Level Study Indonesia	digambar sendiri	—

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Gambar	Judul	Pembuat	Lisensi	Sumber
Gambar 2.6	Peta konsep K3LH dan budaya kerja profesional	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.7	Klasifikasi aset sebuah studio aplikasi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.8	Daur hidup aset	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.9	Contoh inventaris dan catatan peminjaman	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.10	Perbandingan jenis lisensi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.11	Peta konsep pengelolaan aset	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.12	Contoh struktur folder proyek	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.13	Penomoran versi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.14	Alur kerja Git	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.15	Menangani rahasia yang terlanjur terunggah	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.16	Aturan cadangan 3-2-1	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 2.17	Peta konsep pengelolaan berkas proyek	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.1	Menelusuri path dari folder aktif	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.2	Membaca izin berkas	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.3	Cara sistem mencari program lewat PATH	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.4	Virtual environment	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.5	Pesan galat, penyebab, dan tindakan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.6	Peta konsep berkas, direktori, dan lingkungan pengembangan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.7	Perangkat jaringan dan tugas utamanya	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.8	Topologi star lab sekolah	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.9	Alamat IP dan subnet mask dalam biner	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.10	Rentang host dan syarat alamat statis	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.11	Alur memeriksa konektivitas	Tim Next Level Study Indonesia	digambar sendiri	—

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Gambar	Judul	Pembuat	Lisensi	Sumber
Gambar 3.12	Peta konsep konfigurasi jaringan dasar	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.13	Empat layer TCP/IP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.14	Enkapsulasi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.15	Jabat tangan tiga langkah TCP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.16	Perbandingan TCP dan UDP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.17	Protokol biasa dan padanan terenkripsi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.18	Peta konsep layer TCP/IP dan protokol	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.19	Request dan response HTTP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.20	Keluarga kode status HTTP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.21	Bagian-bagian URL	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.22	Alur DNS	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.23	Aplikasi tiga lapis dan port	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 3.24	Peta konsep mekanisme komunikasi jaringan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.1	Tipe data dasar	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.2	Indeks list	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.3	Dictionary	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.4	List dua dimensi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.5	Memilih struktur data	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.6	Peta konsep tipe data dan struktur data	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.7	Simbol diagram alir	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.8	Diagram alir if–elif–else	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.9	Tabel kebenaran	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.10	Nilai batas pada garis bilangan	Tim Next Level Study Indonesia	digambar sendiri	—

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Gambar	Judul	Pembuat	Lisensi	Sumber
Gambar 4.11	Peta konsep struktur kontrol percabangan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.12	range pada garis bilangan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.13	Alur while	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.14	Tabel jejak	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.15	Perulangan bersarang	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.16	Peta konsep perulangan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.17	Aliran data pada fungsi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.18	Lingkup variabel	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.19	Urutan pemanggilan fungsi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.20	Aplikasi kasir dipecah menjadi modul	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.21	Kohesi dan kopling	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 4.22	Peta konsep fungsi, prosedur, dan modularisasi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.1	Class dan object	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.2	Diagram class	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.3	Keadaan objek	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.4	Empat prinsip OOP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.5	Peta konsep class, object, dan prinsip dasar OOP	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.6	Dinding enkapsulasi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.7	Tiga tingkat akses	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.8	Diagram class dengan tanda akses	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.9	Setter dengan validasi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.10	Peta konsep enkapsulasi	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.11	Hierarki class	Tim Next Level Study Indonesia	digambar sendiri	—

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Gambar	Judul	Pembuat	Lisensi	Sumber
Gambar 5.12	Pencarian method naik ke induk	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.13	Urutan super().__init__	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.14	Is-a dan has-a	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.15	Peta konsep pewarisan	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.16	Satu pemanggilan, banyak perilaku	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.17	Tabel jejak loop polimorfik	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.18	Percabangan dan polimorfisme	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.19	Referensi superclass, objek subclass	Tim Next Level Study Indonesia	digambar sendiri	—
Gambar 5.20	Peta konsep polimorfisme	Tim Next Level Study Indonesia	digambar sendiri	—

Tabel silang sub bab dan soal

Sub bab	Soal
1.1 · Profesi dan peluang usaha PPLG	Bedah Prediksi 08 no. 1; Latihan 1.1 no. 1; Latihan 1.1 no. 2; Latihan 1.1 no. 3; Latihan 1.1 no. 4; Latihan 1.1 no. 5; Kuis Bab 1 no. 1; Kuis Bab 1 no. 3; Kuis Bab 1 no. 4; Kuis Bab 1 no. 5; Kuis Bab 1 no. 6; Kuis Bab 1 no. 7; Kuis Bab 1 no. 9; Kuis Bab 1 no. 11; Kuis Bab 1 no. 12; Kuis Bab 1 no. 14; Kuis Bab 1 no. 15; Kuis Bab 1 no. 16; Simulasi 1 no. 4; Simulasi 1 no. 7; Simulasi 2 no. 9; Simulasi 2 no. 17; Simulasi 3 no. 1; Simulasi 3 no. 2
1.2 · Manajemen proyek dan budaya mutu	Bedah Prediksi 08 no. 3; Latihan 1.2 no. 1; Latihan 1.2 no. 2; Latihan 1.2 no. 3; Latihan 1.2 no. 4; Latihan 1.2 no. 5; Kuis Bab 1 no. 2; Kuis Bab 1 no. 8; Kuis Bab 1 no. 10; Kuis Bab 1 no. 13; Kuis Bab 1 no. 17; Kuis Bab 1 no. 18; Kuis Bab 1 no. 19; Kuis Bab 1 no. 20; Kuis Bab 1 no. 21; Kuis Bab 1 no. 22; Kuis Bab 1 no. 23; Kuis Bab 1 no. 24; Kuis Bab 1 no. 25; Simulasi 1 no. 3; Simulasi 1 no. 6; Simulasi 2 no. 2; Simulasi 2 no. 3; Simulasi 3 no. 3; Simulasi 3 no. 5
2.1 · K3LH dan budaya kerja profesional	Bedah Prediksi 09 no. 2; Latihan 2.1 no. 1; Latihan 2.1 no. 2; Latihan 2.1 no. 3; Latihan 2.1 no. 4; Latihan 2.1 no. 5; Kuis Bab 2 no. 1; Kuis Bab 2 no. 2; Kuis Bab 2 no. 3; Kuis Bab 2 no. 5; Kuis Bab 2 no. 7; Kuis Bab 2 no. 8; Kuis Bab 2 no. 10; Kuis Bab 2 no. 13; Kuis Bab 2 no. 15; Simulasi 1 no. 5; Simulasi 1 no. 10; Simulasi 2 no. 11; Simulasi 2 no. 19; Simulasi 3 no. 6; Simulasi 3 no. 10
2.2 · Pengelolaan aset fisik dan digital	Bedah Prediksi 07 no. 10; Latihan 2.2 no. 1; Latihan 2.2 no. 2; Kuis Bab 2 no. 4; Kuis Bab 2 no. 6; Kuis Bab 2 no. 9; Kuis Bab 2 no. 11; Kuis Bab 2 no. 12; Kuis Bab 2 no. 16; Kuis Bab 2 no. 17; Simulasi 1 no. 1; Simulasi 2 no. 1; Simulasi 3 no. 8

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab	Soal
2.3 · Pengelolaan file proyek, source code, dan dokumentasi	Bedah Prediksi 07 no. 8; Latihan 2.3 no. 1; Latihan 2.3 no. 2; Latihan 2.3 no. 3; Kuis Bab 2 no. 14; Kuis Bab 2 no. 18; Kuis Bab 2 no. 19; Kuis Bab 2 no. 20; Kuis Bab 2 no. 21; Kuis Bab 2 no. 22; Kuis Bab 2 no. 23; Kuis Bab 2 no. 24; Kuis Bab 2 no. 25; Simulasi 1 no. 2; Simulasi 1 no. 8; Simulasi 2 no. 8; Simulasi 3 no. 9; Simulasi 3 no. 13
3.1 · Berkas, direktori, dan lingkungan pengembangan pada sistem operasi	Bedah Prediksi 08 no. 23; Latihan 3.1 no. 1; Latihan 3.1 no. 2; Latihan 3.1 no. 3; Latihan 3.1 no. 4; Latihan 3.1 no. 5; Kuis Bab 3 no. 1; Kuis Bab 3 no. 2; Kuis Bab 3 no. 3; Kuis Bab 3 no. 4; Kuis Bab 3 no. 5; Kuis Bab 3 no. 6; Kuis Bab 3 no. 8; Kuis Bab 3 no. 9; Simulasi 1 no. 9; Simulasi 1 no. 23; Simulasi 2 no. 4; Simulasi 2 no. 7; Simulasi 3 no. 11; Simulasi 3 no. 12
3.2 · Konfigurasi jaringan dasar	Bedah Prediksi 07 no. 15; Latihan 3.2 no. 1; Latihan 3.2 no. 2; Kuis Bab 3 no. 7; Kuis Bab 3 no. 15; Kuis Bab 3 no. 17; Kuis Bab 3 no. 18; Kuis Bab 3 no. 21; Kuis Bab 3 no. 24; Simulasi 1 no. 24; Simulasi 2 no. 5; Simulasi 3 no. 7
3.3 · Layer TCP/IP dan fungsi protokol	Bedah Prediksi 07 no. 2; Latihan 3.3 no. 1; Latihan 3.3 no. 2; Kuis Bab 3 no. 10; Kuis Bab 3 no. 11; Kuis Bab 3 no. 14; Kuis Bab 3 no. 16; Kuis Bab 3 no. 23; Simulasi 1 no. 12; Simulasi 2 no. 18; Simulasi 3 no. 4
3.4 · Mekanisme komunikasi jaringan	Bedah Prediksi 07 no. 7; Latihan 3.4 no. 1; Latihan 3.4 no. 2; Kuis Bab 3 no. 12; Kuis Bab 3 no. 13; Kuis Bab 3 no. 19; Kuis Bab 3 no. 20; Kuis Bab 3 no. 22; Kuis Bab 3 no. 25; Simulasi 1 no. 11; Simulasi 2 no. 13; Simulasi 3 no. 15
4.1 · Tipe data dan struktur data	Bedah Prediksi 07 no. 4; Latihan 4.1 no. 1; Latihan 4.1 no. 2; Latihan 4.1 no. 3; Latihan 4.1 no. 4; Latihan 4.1 no. 5; Kuis Bab 4 no. 1; Kuis Bab 4 no. 2; Kuis Bab 4 no. 3; Kuis Bab 4 no. 4; Kuis Bab 4 no. 5; Kuis Bab 4 no. 6; Kuis Bab 4 no. 7; Simulasi 1 no. 18; Simulasi 1 no. 22; Simulasi 2 no. 10; Simulasi 2 no. 14; Simulasi 3 no. 20; Simulasi 3 no. 22
4.2 · Struktur kontrol percabangan	Bedah Prediksi 08 no. 20; Latihan 4.2 no. 1; Latihan 4.2 no. 2; Kuis Bab 4 no. 10; Kuis Bab 4 no. 13; Kuis Bab 4 no. 18; Kuis Bab 4 no. 21; Kuis Bab 4 no. 25; Simulasi 1 no. 20; Simulasi 2 no. 16; Simulasi 3 no. 14
4.3 · Struktur kontrol perulangan	Bedah Prediksi 09 no. 17; Latihan 4.3 no. 1; Latihan 4.3 no. 2; Latihan 4.3 no. 3; Latihan 4.3 no. 4; Kuis Bab 4 no. 11; Kuis Bab 4 no. 14; Kuis Bab 4 no. 15; Kuis Bab 4 no. 16; Kuis Bab 4 no. 20; Kuis Bab 4 no. 23; Simulasi 1 no. 19; Simulasi 2 no. 6; Simulasi 2 no. 12; Simulasi 3 no. 17
4.4 · Fungsi, prosedur, dan modularisasi	Bedah Prediksi 07 no. 17; Latihan 4.4 no. 1; Latihan 4.4 no. 2; Latihan 4.4 no. 3; Latihan 4.4 no. 4; Latihan 4.4 no. 5; Kuis Bab 4 no. 8; Kuis Bab 4 no. 9; Kuis Bab 4 no. 12; Kuis Bab 4 no. 17; Kuis Bab 4 no. 19; Kuis Bab 4 no. 22; Kuis Bab 4 no. 24; Simulasi 1 no. 16; Simulasi 1 no. 21; Simulasi 2 no. 15; Simulasi 2 no. 25; Simulasi 3 no. 21; Simulasi 3 no. 25
5.1 · Class, object, dan prinsip dasar OOP	Bedah Prediksi 09 no. 21; Latihan 5.1 no. 1; Latihan 5.1 no. 2; Latihan 5.1 no. 3; Latihan 5.1 no. 4; Latihan 5.1 no. 5; Kuis Bab 5 no. 1; Kuis Bab 5 no. 2; Kuis Bab 5 no. 3; Kuis Bab 5 no. 5; Kuis Bab 5 no. 6; Kuis Bab 5 no. 8; Kuis Bab 5 no. 9; Kuis Bab 5 no. 10; Simulasi 1 no. 15; Simulasi 1 no. 25; Simulasi 2 no. 23; Simulasi 2 no. 24; Simulasi 3 no. 18; Simulasi 3 no. 24

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

Sub bab	Soal
5.2 · Enkapsulasi, access modifier, dan getter–setter	Bedah Prediksi 07 no. 21; Latihan 5.2 no. 1; Latihan 5.2 no. 2; Kuis Bab 5 no. 4; Kuis Bab 5 no. 7; Kuis Bab 5 no. 11; Kuis Bab 5 no. 12; Kuis Bab 5 no. 23; Kuis Bab 5 no. 24; Simulasi 1 no. 17; Simulasi 2 no. 20; Simulasi 3 no. 23
5.3 · Pewarisan	Bedah Prediksi 07 no. 19; Latihan 5.3 no. 1; Latihan 5.3 no. 2; Kuis Bab 5 no. 13; Kuis Bab 5 no. 15; Kuis Bab 5 no. 16; Kuis Bab 5 no. 18; Kuis Bab 5 no. 19; Simulasi 1 no. 14; Simulasi 2 no. 22; Simulasi 3 no. 16
5.4 · Polymorphism	Bedah Prediksi 07 no. 22; Latihan 5.4 no. 1; Latihan 5.4 no. 2; Kuis Bab 5 no. 14; Kuis Bab 5 no. 17; Kuis Bab 5 no. 20; Kuis Bab 5 no. 21; Kuis Bab 5 no. 22; Kuis Bab 5 no. 25; Simulasi 1 no. 13; Simulasi 2 no. 21; Simulasi 3 no. 19

NEXT LEVEL STUDY
Educational Institutions

Gratis dari Next Level Study Indonesia · next-level-study.com/library · tidak untuk diperjualbelikan

NEXT LEVEL STUDY INDONESIA

Era Baru Pendidikan

TKA SMK PPLG

Tetaplah menari, sekalipun lagu pengiringmu telah hilang. Sebab di luar sana, selalu ada seseorang yang menemukan bahagianya karena kamu tidak berhenti melangkah.

— HANDAKA LUMU

Buku ini menjelaskan secara rinci apa saja yang perlu kamu persiapkan untuk TKA Pengembangan Perangkat Lunak dan Gim: materi yang diuji, strategi menjawab, dan latihan yang terarah.

Belajar lebih jauh di LMS NLS

next-level-study.com/belajar/lms-player.html?course=buku-smkpplg

GRATIS · TIDAK UNTUK DIPERJUALBELIKAN

SMKPPLG · Edisi 1 · Gratis untuk pelajar Indonesia · 2026/2027