

**MODUL AJAR DEEP LEARNING**  
**MATA PELAJARAN INFORMATIKA**  
**BAB 7: Algoritma dan Pemrograman**

Struktur Kondisional, Nested If, dan Perulangan

**A. IDENTITAS MODUL**

<b>Satuan Pendidikan</b>	:	SMA Negeri 2 Padang
<b>Mata Pelajaran</b>	:	Informatika
<b>Kelas / Semester</b>	:	X / Ganjil & Genap
<b>Fase</b>	:	E
<b>Materi Pokok</b>	:	Algoritma dan Pemrograman (Struktur Kondisional, Nested If, dan Perulangan)
<b>Alokasi Waktu</b>	:	12 × 45 Menit (6 Pertemuan)
<b>Model Pembelajaran</b>	:	Project Based Learning (PjBL)
<b>Penyusun</b>	:	Firandi Julian (25427691)
<b>Tahun Pelajaran</b>	:	2025/2026

**B. IDENTIFIKASI KESIAPAN PESERTA DIDIK**

**Pengetahuan Awal:**

Peserta didik diharapkan memiliki pemahaman dasar tentang berpikir komputasional, meliputi kemampuan dekomposisi masalah, pengenalan pola, abstraksi, dan merumuskan algoritma sederhana. Pada pertemuan lanjut, peserta didik telah memahami variabel, tipe data, operator, dan struktur kontrol percabangan (if-else).

**Kebutuhan Belajar:**

1. Visual: Diagram alur (flowchart), contoh kode dengan highlighting sintaks, visualisasi iterasi step by step
2. Auditori: Penjelasan lisan yang jelas tentang konsep, diskusi kelompok untuk memecahkan bug
3. Kinestetik: Praktik langsung menulis kode, debugging, mencoba berbagai input

## C. TUJUAN PEMBELAJARAN PER PERTEMUAN

Pertemuan	Tujuan Pembelajaran
<b>Pertemuan 1</b>	Menjelaskan sintaks if/elif/else (C2); Menganalisis bug logika kondisional (C4); Merancang & mempresentasikan program kondisional [Warung Makan + Toko Diskon] (C4, P3)
<b>Pertemuan 2</b>	Menguji program kondisional dengan edge case (C3); Mengidentifikasi & memperbaiki bug kondisional secara sistematis (C4); Mengevaluasi efisiensi if-elif-else vs tiga if terpisah (C5)
<b>Pertemuan 3</b>	Menjelaskan konsep nested if dan membedakannya dari if-elif-else biasa (C2); Menerapkan nested if untuk kondisi berlapis (C3); Menganalisis alur dengan flowchart & tracing (C4)
<b>Pertemuan 4</b>	Mengevaluasi kapan sebaiknya menggunakan nested if vs operator logika (C5); Menulis program nested if terdokumentasi (P3); Melakukan debugging nested if (C4)
<b>Pertemuan 5</b>	Menganalisis konsep & fungsi perulangan for dan while (C4); Membandingkan karakteristik for vs while (C4); Menulis pseudocode perulangan untuk masalah sederhana (P3)
<b>Pertemuan 6</b>	Mengimplementasikan for/while loop dengan output benar (C3); Merancang nested loop & break/continue (C4); Melakukan debugging bug umum pada loop (C4)

## D. MATERI PEMBELAJARAN

### Pertemuan 1 & 2 Struktur Kondisional (if, elif, else):

1. Pengertian dan konsep struktur kondisional dalam pemrograman
2. Sintaks if, elif, dan else dalam Python serta alur eksekusi
3. Operator perbandingan (==, !=, >, <, >=, <=) dan operator logika (and, or, not)
4. Debugging: mengidentifikasi dan memperbaiki kesalahan logika kondisional

### Pertemuan 3 & 4 Nested If (Kondisi Bersarang):

1. Pengertian nested if dan perbedaannya dengan if-elif-else biasa
2. Sintaks nested if dalam Python dan pentingnya indentasi
3. Flowchart nested if visualisasi alur keputusan berlapis
4. Contoh aplikasi: sistem nilai + bonus, tiket bioskop, sistem login
5. Perbandingan nested if vs operator logika (and/or)

### Pertemuan 5 & 6 Struktur Perulangan (Looping):

1. Pengertian dan komponen perulangan: kondisi, counter, variabel kontrol, body loop
2. For loop: sintaks, fungsi range(), iterasi pada list dan string
3. While loop: sintaks, perbedaan dengan for loop, risiko infinite loop
4. Break, continue, dan else pada loop
5. Nested loop dan aplikasinya (pola bintang, tabel perkalian)
6. Debugging bug umum: infinite loop, off-by-one error, logika akumulator salah

## E. DESAIN KEGIATAN PEMBELAJARAN (SEMUA PERTEMUAN)

### PERTEMUAN 1 Struktur Kondisional, Teori & Proyek PjBL

No	Kegiatan	Deskripsi	Waktu
1.	<b>PENDAHULUAN</b>		<b>15'</b>
a.	<b>Orientasi &amp; Ice Breaking</b>	Salam, doa, absensi. Ice Breaking: tampilkan 3 output program, siswa menebak kondisi if-else yang menghasilkannya (PPK: Religius & Disiplin)	3'
b.	<b>Apersepsi</b>	Tanya jawab: 'Pernahkah menggunakan ATM atau kasir digital? Bagaimana mesin bisa membuat keputusan otomatis?'	4'

c.	<b>Motivasi</b>	Tampilkan video 1 menit penggunaan kondisional pada kasir digital dan sistem nilai otomatis	4'
d.	<b>Pemberian Acuan</b>	Sampaikan tujuan (C2-C4, P3, A3), alur PjBL 6 tahap, produk akhir, dan sistem penilaian	4'
<b>2.</b>	<b>KEGIATAN INTI 6</b>		<b>60'</b>
	<b>Tahap PjBL</b>		
Tahap 1	<b>Pertanyaan Mendasar</b>	Penjelasan konsep if/elif/else + demo kode di layar. Pertanyaan: 'Dapatkah kalian membuat program yang secara otomatis menentukan kelulusan siswa, besaran diskon, dan jenis bilangan hanya menggunakan logika kondisional?'	10'
Tahap 2	<b>Desain Perencanaan</b>	Bagi kelompok (3-4 orang): Kelompok A = Program penentu kelulusan & jenis bilangan. Kelompok B = Program diskon dengan nested if ( $\geq 500$ rb: 20%, 200rb-499rb: 10%, $< 200$ rb: 0%). Setiap kelompok merancang fitur, flowchart, dan pembagian peran.	10'
Tahap 3	<b>Menyusun Jadwal</b>	Kelompok menyusun timeline di Jobsheet Pertemuan 1 (Bagian B) dan menandatangani kontrak belajar (PPK: Tanggung Jawab)	5'
Tahap 4	<b>Monitor Kemajuan</b>	Kelompok mengerjakan program sesuai Jobsheet Pertemuan 1 (Bagian C Template Kode). Guru berkeliling memberi scaffolding. Referensi: Bahan Ajar Pertemuan 1 Bagian C & D	20'
Tahap 5	<b>Menguji Hasil</b>	Setiap kelompok presentasi di depan kelas (maks 10 menit), demonstrasikan program dengan 5+ variasi input. Kelompok lain memberi tanggapan konstruktif	10'

Tahap 6	<b>Evaluasi Pengalaman</b>	Refleksi: 'Apa yang paling menantang dari menulis kondisi?' Bersama simpulkan konsep kunci. Siswa isi Jobsheet Bagian D (Refleksi)	5'
<b>3.</b>	<b>PENUTUP</b>		<b>15'</b>
a.	<b>Penguatan &amp; Penugasan</b>	Penguatan konsep, jawab pertanyaan tersisa. Tugas: revisi kode berdasarkan masukan + kumpulkan kode final. Preview: Pertemuan 2 = debugging & asesmen	10'
b.	<b>Refleksi &amp; Doa</b>	Exit Ticket: 'Satu hal yang paling saya pelajari hari ini adalah...' Apresiasi kerja keras, doa penutup.	5'

<b>Jobsheet yang Digunakan:</b> Jobsheet Pertemuan 1 Bagian A, B, C (Template Kode Warung Makan untuk contoh tambahan), D	<b>Bahan Ajar yang Digunakan:</b> Bahan Ajar Pertemuan 1 Bagian A-G (Sintaks, Operator, Contoh Program, Common Mistakes)
--	---

## PERTEMUAN 2 Debugging, Pengujian, dan Asesmen Kondisional

No	Kegiatan	Deskripsi	Waktu
1.	<b>PENDAHULUAN Review &amp; Orientasi</b>	Salam, doa, absensi. Review: tampilkan 3 kode kondisional, siswa tracing output secara manual. Pastikan semua kelompok memiliki kode dari Pertemuan 1.	<b>10'</b>
2.	<b>KEGIATAN INTI</b>		<b>65'</b>
a.	<b>Debugging Terstruktur (Jobsheet P2 Bagian A)</b>	Guru tampilkan 3 program berisi bug: (1) input tidak dikonversi ke int, (2) dua if terpisah saat seharusnya elif, (3) operator => tidak valid. Kelompok identifikasi bug, jelaskan mengapa salah, tulis versi yang diperbaiki.	20'



b.	<b>Asesmen Formatif Kuis (Bagian B)</b>	Kuis 5 soal pilihan ganda (Jobsheet P2 Bagian B): LOTS-HOTS. Mandiri, lalu bahas bersama.	20'
c.	<b>Asesmen Sumatif Esai (Bagian C)</b>	4 soal esai C2-C5: jelaskan analogi if-elif-else (C2), tulis kode kalkulator diskon (C3), analisis bug (C4), evaluasi dua pendekatan BMI (C5). Dikerjakan individu.	25'
<b>3.</b>	<b>PENUTUP</b>	Bahas poin esai yang sering salah. Pengumuman materi berikutnya: Nested If. Lembar refleksi: 3 hal dipelajari, 2 hal menarik, 1 pertanyaan.	<b>15'</b>

Jobsheet P2 Bagian A (Debugging), B (Kuis PG), C (Esai)	Bahan Ajar P2 Bagian A (Testing), B (Debugging), C (Evaluasi if-elif-else vs if terpisah)
---	---

### PERTEMUAN 3 Nested If, Teori & Proyek PjBL

No	Kegiatan	Deskripsi	Waktu
<b>1.</b>	<b>PENDAHULUAN</b>	Salam, doa, absensi. Ice Breaking: skenario keputusan berlapis. Review materi P1-P2 (5 menit tracing bersama). Guru tunjukkan kode login sebagai contoh nested if nyata.	<b>15'</b>
<b>2.</b>	<b>INTI 6 Tahap PjBL</b>		<b>60'</b>
Tahap 1	<b>Pertanyaan Mendasar</b>	Live coding program penentuan nilai + bonus kehadiran menggunakan nested if. Tunjukkan perbedaan dengan if-elif biasa. Pertanyaan: 'Dapatkah kalian membuat sistem tiket wisata yang memeriksa hari DAN usia pengunjung?'	10'
Tahap 2	<b>Desain Perencanaan</b>	Kelompok A: Sistem nilai + remedial berdasarkan kehadiran. Kelompok B: Sistem tiket bioskop (hari + usia + kelas kursi). Buat:	10'

Tahap 3	<b>Menyusun Jadwal</b>	flowchart nested if, tabel kombinasi input-output (Jobsheet P3 Bagian B), pembagian peran. Timeline: Sesi 1 (flowchart + kode dasar)   Sesi 2 (pengujian + presentasi). Tandatangani kontrak belajar.	5'
Tahap 4	<b>Monitor Kemajuan</b>	Setiap kelompok: (a) gambar flowchart nested if, (b) tulis kode Python + komentar di setiap blok, (c) isi tabel 6 kombinasi input-output (Jobsheet P3 Bagian C). Guru beri scaffolding tentang indentasi.	20'
Tahap 5	<b>Menguji Hasil</b>	Presentasi (maks 10 menit/kelompok), demonstrasi 6+ kombinasi input. Kelompok lain tanya tentang logika nested if yang digunakan.	10'
Tahap 6	<b>Evaluasi Pengalaman</b>	Refleksi: 'Apa bedanya nested if vs operator and?' Simpulkan konsep kunci. Siswa isi Jobsheet P3 Bagian D.	5'
<b>3.</b>	<b>PENUTUP</b>	Penguatan: nested if untuk kondisi dalam yang bergantung kondisi luar; and/or untuk kondisi independen. Tugas: program pembelian online dengan nested if (cek stok → ongkos kirim per kota). Doa penutup.	<b>15'</b>

 Jobsheet P3 Bagian A (Tracing), B (Perencanaan + Flowchart), C (Kode + Test), D (Refleksi)	 Bahan Ajar P3 Bagian A-E (Konsep, Sintaks, Flowchart, Nested if vs and/or)
--	--

**PERTEMUAN 4 Nested If: Evaluasi, Debugging, dan Asesmen**

No	Kegiatan	Deskripsi	Waktu
----	----------	-----------	-------

1.	<b>PENDAHULUAN</b> <b>Review</b>	Salam, doa, absensi. Review: tracing kode nested if dari tugas P3 bersama. Tanya: 'Kapan sebaiknya nested if daripada operator and?'	<b>10'</b>
2.	<b>KEGIATAN INTI</b>		<b>65'</b>
a.	<b>Evaluasi &amp; Komparasi (Bagian A)</b>	Guru tampilkan dua pendekatan untuk masalah BMI yang sama: nested if berlapis vs if-elif-else linear. Kelompok diskusi: mana yang tepat untuk kasus BMI, login, tiket? Presentasi argumen → guru klarifikasi. (Jobsheet P4 Bagian A, Bahan Ajar P4 Bagian A-B)	20'
b.	<b>Debugging Nested If (Bagian B)</b>	Guru tampilkan kode nested if berisi 2 bug: (1) indentasi tidak konsisten, (2) else terhubung ke if yang salah. Kelompok identifikasi semua kesalahan, jelaskan output untuk input tertentu, diskusi cara mencegahnya. (Jobsheet P4 Bagian B)	20'
c.	<b>Asesmen Sumatif (Bagian C)</b>	5 soal PG + 2 esai: C2 (flowchart login), C3 (tulis kode nested if sistem login). Individu. Bahas kunci bersama setelah selesai. (Jobsheet P4 Bagian C)	25'
3.	<b>PENUTUP</b>	Rangkum perbedaan if-elif-else, nested if, dan operator and/or. Pengumuman: Pertemuan berikutnya = Struktur Perulangan (Looping). Lembar refleksi singkat.	<b>15'</b>

## PERTEMUAN 5 Perulangan: For Loop & While Loop (Teori)

No	Kegiatan	Deskripsi	Waktu
1.	<b>PENDAHULUAN</b>	Salam, doa, absensi, kesepakatan belajar. Apersepsi: tampilkan kode 100 print() vs 2 baris with for loop. Motivasi: 'Hari ini kalian memahami mekanisme yang menggerakkan game dan otomasi data!'	<b>15'</b>

<b>2.</b>	<b>KEGIATAN INTI</b>		<b>60'</b>
a.	<b>Memahami Konsep Looping</b>	Masalah pemantik: 'Bagaimana mencetak tabel perkalian 1-10 tanpa print() berulang?' Kelompok (4-5 orang) diskusi hipotesis awal. Guru jelaskan konsep looping: pengertian, jenis (for, while), flowchart, demo live coding range(). Siswa catat dan bertanya aktif. (Bahan Ajar P5 Bagian B-D)	20'
b.	<b>Tracing &amp; Pseudocode (Jobsheet P5 Bagian B-C)</b>	Kelompok kerjakan tracing kode: tulis output setiap iterasi secara manual. Diskusi for vs while untuk masalah yang sama. Guru keliling beri umpan balik formatif. Susun pseudocode: 'Hitung jumlah bilangan ganjil 1-100'. Perwakilan presentasi pseudocode → kelompok lain tanggapi.	25'
c.	<b>Merefleksi</b>	Individu: tulis satu konsep yang dipahami, satu yang masih membingungkan, satu pertanyaan untuk guru. Guru tampung dan luruskan miskonsepsi.	15'
<b>3.</b>	<b>PENUTUP</b>	Rangkum: for loop, while loop, perbedaan keduanya. Kuis singkat 5 soal lisan. Preview P6: nested loop, break/continue, asesmen sumatif looping.	<b>15'</b>

Jobsheet P5 Bagian A (Tabel Perbandingan), B (Tracing 1 & 2), C (Pseudocode), D (Refleksi)	Bahan Ajar P5 Bagian A-E (Konsep, For Loop, While Loop, Perbandingan)
--	---

**PERTEMUAN 6 Perulangan Lanjutan: Nested Loop, Break, Continue, Asesmen**

No	Kegiatan	Deskripsi	Waktu
<b>1.</b>	<b>PENDAHULUAN</b>	Salam, doa, absensi, pastikan semua siswa dapat mengakses IDE (Google Colab/Replit/IDLE). Paparkan tujuan pembelajaran P6.	<b>10'</b>

<b>2.</b>	<b>KEGIATAN INTI Praktik Coding</b>		<b>65'</b>
a.	<b>Demo Break, Continue, Nested Loop</b>	Guru demo: break (hentikan loop seketika), continue (lewati iterasi), else pada while. Analogi jam untuk nested loop: 'Jarum menit berputar penuh (loop dalam) sebelum jarum jam berpindah (loop luar)'. Siswa ikuti langsung di perangkat. (Bahan Ajar P6 Bagian A-B)	10'
b.	<b>Jobsheet Perulangan (Kegiatan 1-4)</b>	Siswa kerjakan Jobsheet P6 Kegiatan 1-4 secara mandiri di Google Colab: (1) Pola segitiga for loop, (2) Tebak angka while loop, (3) Break & continue, (4) Nested loop tabel perkalian. Bonus: pola berlian untuk yang selesai lebih awal. Dokumentasikan kode + simpan tautan.	25'
c.	<b>Debugging Loop Bersama (Kegiatan 5)</b>	Guru tampilkan 3 program berisi bug: (1) infinite loop lupa update, (2) off-by-one error, (3) logika akumulator salah. Kelompok identifikasi & perbaiki (Jobsheet P6 Kegiatan 5). Diskusi kelas: mengapa bug terjadi & cara mencegah.	15'
d.	<b>Peer Review &amp; Mini Presentasi</b>	2-3 kelompok tampilkan solusi & jelaskan logika kode. Peer review 3 pertanyaan: 'Apakah kode berjalan? Apakah logika tepat? Apa yang bisa diperbaiki?' Siswa catat satu hal yang akan diperbaiki.	15'
<b>3.</b>	<b>PENUTUP &amp; Asesmen</b>	Rangkum seluruh materi looping. Kumpulkan link Google Colab / file .py ke Google Classroom (termasuk Jobsheet P6 Asesmen Sumatif PG + Esai berlian). Lembar refleksi: 3 dipelajari, 2 menarik, 1 pertanyaan. Apresiasi & doa.	<b>15'</b>

Jobsheet P6 Kegiatan 1-5 (Praktik), Asesmen Sumatif (PG + Esai)

Bahan Ajar P6 Bagian A-D (Break/Continue, Nested Loop, Debugging, Latihan)

## F. ASESMEN

Jenis Asesmen	Pertemuan	Bentuk	Instrumen
<b>Diagnostik</b>	Awal P1	Tanya jawab lisan	Pertanyaan pemantik ATM & kasir digital
<b>Formatif Sikap</b>	P1, P3, P5	Observasi	Lembar observasi kolaborasi & kemandirian
<b>Formatif Pengetahuan</b>	P1, P3, P5	Exit ticket & kuis lisan	Exit ticket + 5 soal kuis cepat
<b>Formatif Keterampilan</b>	P1, P3	Rubrik proyek PjBL	Rubrik 4 kriteria @ 25 poin (Jobsheet Bagian Penilaian)
<b>Sumatif P2</b>	P2	Kuis PG + Esai Kondisional	Jobsheet P2: 5 PG + 4 soal esai C2-C5
<b>Sumatif P4</b>	P4	Kuis PG + Esai Nested If	Jobsheet P4: 5 PG + 2 soal esai (flowchart + kode)
<b>Sumatif P6</b>	P6	Kuis PG + Esai Looping	Jobsheet P6: 5 PG + 1 esai pola berlian nested loop

## G. MEDIA, ALAT, DAN SUMBER BELAJAR

### Media & Alat:

1. Laptop/komputer dengan Python terinstal (IDLE / VS Code / Thonny / Google Colab/Programiz)
2. Proyektor / layar presentasi untuk live coding guru
3. Akses internet (opsional untuk Google Colab)
4. Jobsheet Pertemuan 1-6 (6 dokumen jobsheet yang telah disiapkan)
5. Bahan Ajar Pertemuan 1-6 (6 dokumen bahan ajar yang telah disiapkan)

### Sumber Belajar:

1. Bahan Ajar Pertemuan 1-6 (dokumen terlampir)

2. Python Official Documentation: <https://docs.python.org/3/tutorial/controlflow.html>
3. W3Schools Python: <https://www.w3schools.com/python/>
4. Google Colab: <https://colab.research.google.com>

## H. RENCANA DIFERENSIASI PEMBELAJARAN

Kelompok	Strategi	Contoh Penerapan
<b>Reguler</b>	Scaffolding bertahap, template kode dengan bagian kosong	Kelompok A: Program penentu kelulusan & jenis bilangan
<b>Pengayaan</b>	Proyek lebih kompleks, tantangan bonus	Kelompok B: Diskon nested if kompleks, pola berlian
<b>Kesulitan Belajar</b>	Peer tutoring, bimbingan individual guru	Penugasan partner + waktu ekstra + kode template lebih lengkap

## I. PEMAHAMAN BERMAKNA

Struktur kondisional adalah fondasi pengambilan keputusan dalam pemrograman. Hampir semua sistem digital ATM, aplikasi belanja, sistem rapor otomatis menggunakan logika if-else. Nested if memungkinkan program mengambil keputusan berlapis seperti sistem login nyata. Perulangan adalah mekanisme yang menggerakkan game, otomasi data, dan simulasi di balik layar setiap aplikasi yang digunakan. Memahami ketiganya berarti mampu mengajarkan komputer untuk 'berpikir', 'memutuskan', dan 'mengulang' secara efisien.

**BAHAN****AJAR****Pertemuan 1****BAHAN AJAR INFORMATIKA KELAS X SMA****Struktur Kondisional (if, elif, else)**

Teori, Contoh, dan Proyek PjBL

<b>Mata Pelajaran / Fase</b>	Informatika / Fase E
<b>Kelas / Semester</b>	X (Sepuluh) / Ganjil
<b>Alokasi Waktu</b>	2 × 45 Menit
<b>Penyusun</b>	Firandi Julian
<b>Sekolah</b>	SMA Negeri 2 Padang

**TUJUAN PEMBELAJARAN**

Setelah mempelajari bahan ajar ini, peserta didik mampu:

1. Menjelaskan konsep dan sintaks if, elif, else di Python (C2)
2. Menganalisis kesalahan logika kondisional (C4)
3. Merancang dan membuat program kondisional untuk kasus nyata (C4, P3)

**A. PERTANYAAN PEMANTIK**

Pernahkah kalian menggunakan sistem ATM? Bagaimana mesin ATM bisa secara otomatis memutuskan apakah PIN yang dimasukkan benar atau salah? Atau bagaimana aplikasi kasir menentukan besaran kembalian?

**Semua keputusan otomatis dalam program komputer dibuat menggunakan STRUKTUR KONDISIONAL.**

**B. PENGERTIAN STRUKTUR KONDISIONAL**

Struktur kondisional adalah mekanisme dalam pemrograman yang memungkinkan program mengeksekusi blok kode yang berbeda berdasarkan hasil evaluasi suatu kondisi (benar/salah). Ini adalah fondasi dari pengambilan keputusan otomatis dalam komputer.

## Analogi kehidupan nyata:

Bayangkan seorang petugas tiket bioskop: 'JIKA usia di bawah 17 tahun, MAKA tidak boleh masuk film dewasa. JIKA tidak, MAKA diperbolehkan masuk.' Itulah logika ifelse!

## C. SINTAKS DAN ALUR EKSEKUSI

### C.1 Struktur if Tunggal

```
if kondisi:
    # Blok ini dijalankan HANYA jika kondisi bernilai True
    pernyataan

# Contoh:
nilai = 80
if nilai >= 75:
    print('Selamat! Kamu lulus.')
```

### C.2 Struktur if, else

```
if kondisi:
    # Dijalankan jika kondisi True
else:
    # Dijalankan jika kondisi False

# Contoh:
nilai = 60
if nilai >= 75:
    print('Lulus')
else:
    print('Tidak lulus perlu remedial')
```

### C.3 Struktur if, elif, else (Kondisi Berlapis)

```
if kondisi_1:
```

```

    # Jika kondisi_1 True
elif kondisi_2:
    # Jika kondisi_1 False, tapi kondisi_2 True
elif kondisi_3:
    # Jika kondisi_1 & 2 False, tapi kondisi_3 True
else:
    # Jika semua kondisi False

# Contoh Sistem Nilai:
nilai = float(input('Masukkan nilai: '))
if nilai >= 90:
    grade = 'A'
elif nilai >= 80:
    grade = 'B'
elif nilai >= 70:
    grade = 'C'
elif nilai >= 60:
    grade = 'D'
else:
    grade = 'E'
print(f'Grade: {grade}')

```

### ATURAN PENTING:

1. Python WAJIB menggunakan indentasi (spasi/tab) konsisten 4 spasi per level
2. Setiap blok if/elif/else diakhiri tanda titik dua (:)
3. Kondisi dievaluasi dari atas ke bawah begitu satu kondisi True, blok lainnya DILEWATI

## D. OPERATOR PERBANDINGAN & LOGIKA

Operator	Arti	Contoh
==	Sama dengan	5 == 5 → True
!=	Tidak sama dengan	5 != 3 → True

<code>&gt; / &lt;</code>	Lebih besar / kecil	<code>7 &gt; 3 → True</code>
<code>&gt;= / &lt;=</code>	Lebih besar/kecil sama dengan	<code>5 &gt;= 5 → True</code>
<b>and</b>	Kedua kondisi harus True	<code>x &gt; 0 and x &lt; 10</code>
<b>or</b>	Salah satu kondisi True	<code>x &lt; 0 or x &gt; 100</code>
<b>not</b>	Membalik kondisi	<code>not (x == 0)</code>

## E. CONTOH PROGRAM LENGKAP

### Contoh 1 Kalkulator Diskon Toko:

```
# Program Kalkulator Diskon
nama = input('Nama pembeli: ')
total = float(input('Total belanja (Rp): '))

if total >= 500000:
    diskon = 0.20 # 20%
    kategori = 'VIP'
elif total >= 200000:
    diskon = 0.10 # 10%
    kategori = 'Reguler'
else:
    diskon = 0 # 0%
    kategori = 'Biasa'

hemat = total * diskon
bayar = total - hemat

print('==== Struk Belanja ====')
print(f'Pembeli : {nama}')
print(f'Kategori : {kategori}')
print(f'Total : Rp {total:,.0f}')
```

```
print(f'Diskon : Rp {hemat:,.0f} ({{int(diskon*100)}}%)')
print(f'Total Bayar: Rp {bayar:,.0f}')
```

## Contoh 2 Program Penentu Jenis Bilangan:

```
angka = int(input('Masukkan bilangan: '))

if angka > 0:
    jenis = 'positif'
elif angka < 0:
    jenis = 'negatif'
else:
    jenis = 'nol'

if angka % 2 == 0 and angka != 0:
    sifat = 'genap'
elif angka % 2 != 0:
    sifat = 'ganjil'
else:
    sifat = 'nol (bukan genap/ganjil)'

print(f'{angka} adalah bilangan {jenis} dan {sifat}')
```

## F. KESALAHAN UMUM (COMMON MISTAKES)

### ✗ Menggunakan = bukan ==

#### ✗ Salah:

```
if x = 10:
```

#### ✓ Benar:

```
if x == 10: # == untuk perbandingan
```

### ✗ Menggunakan if bertumpuk saat seharusnya elif

#### ✗ Salah:

```
if nilai >= 90:
```

#### ✓ Benar:

```
if nilai >= 90:
```

```
grade = 'A'  
if nilai >= 80:  
    grade = 'B'
```

```
grade = 'A'  
elif nilai >= 80:  
    grade = 'B'
```

### ✗ Input tidak dikonversi ke int/float

#### ✗ Salah:

```
umur = input('Usia: ')  
if umur >= 17:
```

#### ✓ Benar:

```
umur = int(input('Usia: '))  
if umur >= 17:
```

## G. LATIHAN MANDIRI

Kerjakan latihan berikut secara mandiri atau bersama kelompok:

1. Latihan 1: Buat program yang meminta input nilai rapor 5 mata pelajaran dan menentukan apakah siswa naik kelas ( $\text{rata-rata} \geq 75$  dan tidak ada nilai di bawah 60)
2. Latihan 2: Buat program kalkulator BMI yang menentukan kategori berat badan (Kurus/Normal/Gemuk/Obesitas)
3. Latihan 3 (Pengayaan): Buat program simulasi mesin ATM cek saldo, tarik tunai, cek PIN

Ingat: Selalu uji program Anda dengan minimal 3 variasi input berbeda!

**BAHAN****AJAR****Pertemuan 2****BAHAN AJAR INFORMATIKA KELAS X SMA****Pengujian & Debugging Kondisional**

Strategi Testing, Bug Hunting, dan Asesmen Formatif

<b>Mata Pelajaran / Fase</b>	Informatika / Fase E
<b>Kelas / Semester</b>	X (Sepuluh) / Ganjil
<b>Alokasi Waktu</b>	2 × 45 Menit
<b>Penyusun</b>	Firandi Julian & Ikhsan Budi Maulana
<b>Sekolah</b>	SMA Negeri 2 Padang

**TUJUAN PEMBELAJARAN**

- Menguji program kondisional dengan berbagai variasi input termasuk edge case (C3)
- Mengidentifikasi dan memperbaiki bug logika kondisional secara sistematis (C4)
- Mengevaluasi efisiensi penggunaan if-else vs tiga if terpisah (C5)

**A. STRATEGI PENGUJIAN PROGRAM**

Menulis kode yang benar saja tidak cukup program harus diuji secara menyeluruh sebelum digunakan. Ada 3 jenis test case yang wajib dilakukan:

<b>Jenis Test</b>	<b>Penjelasan</b>	<b>Contoh (nilai ≥ 75 lulus)</b>
<b>Normal Case</b>	Input yang seharusnya masuk ke setiap cabang	nilai = 80 (lulus), nilai = 50 (tidak lulus)
<b>Boundary Case (Edge Case)</b>	Nilai tepat di batas kondisi	nilai = 75 (tepat lulus), nilai = 74 (tidak lulus)
<b>Invalid Input</b>	Input di luar rentang yang diharapkan	nilai = 10, nilai = 200, nilai = 'abc'

## B. TEKNIK DEBUGGING SISTEMATIS

Debugging adalah seni menemukan dan memperbaiki bug (kesalahan) dalam program. Langkah sistematis:

- Langkah 1 Baca pesan error dengan teliti: Python selalu memberitahu baris dan jenis error
- Langkah 2 Lakukan tracing manual: ikuti alur program langkah demi langkah di atas kertas
- Langkah 3 Gunakan print() sebagai debugging tool: tambahkan print(variabel) untuk melihat nilai variabel
- Langkah 4 Periksa kondisi: apakah kondisi if sudah benar? Apakah menggunakan == bukan =?
- Langkah 5 Uji dengan input sederhana dahulu, baru ke kasus kompleks

### 3 Bug Paling Umum pada Struktur Kondisional:

```
# BUG 1: = digunakan untuk perbandingan (seharusnya ==)
if x = 10: # SyntaxError!
# Perbaiki:
if x == 10:

# BUG 2: Input string dibandingkan dengan angka
umur = input('Usia: ') # Bertipe string!
if umur >= 17:      # TypeError!
# Perbaiki:
umur = int(input('Usia: '))

# BUG 3: Dua if terpisah saat seharusnya elif
skor = 85
if skor >= 90: grade = 'A' # Tidak dijalankan
if skor >= 75: grade = 'B' # DIJALANKAN (grade = B)
if skor >= 60: grade = 'C' # JUGA DIJALANKAN (grade = C, menimpa B!)
# Perbaiki:
```

```

if skor >= 90: grade = 'A'
elif skor >= 75: grade = 'B'
elif skor >= 60: grade = 'C'

```

### C. EVALUASI: if, elif, else vs TIGA if TERPISAH

Kapan menggunakan ifelifelse? Kapan menggunakan beberapa if terpisah?

Aspek	ifelifelse	3 if Terpisah
Kondisi saling eksklusif	✓ Tepat hanya satu blok yang dijalankan	✗ Berpotensi beberapa blok berjalan
Efisiensi	✓ Lebih efisien berhenti setelah kondisi True	✗ Semua kondisi selalu diperiksa
Keterbacaan	✓ Lebih jelas strukturnya	✗ Bisa membingungkan
Kondisi independen	✗ Kurang tepat jika kondisi tidak berkaitan	✓ Lebih tepat

**KESIMPULAN:** Gunakan ifelifelse saat kondisi saling eksklusif (nilai rapor, kategori harga).  
Gunakan if terpisah saat setiap kondisi bisa bernilai true secara bersamaan (misalnya: cek apakah memenuhi kriteria 1, kriteria 2, dan kriteria 3 secara terpisah).

Referensi: Python Official Documentation <https://docs.python.org/3/tutorial/controlflow.html>

SMA Negeri 2 Padang Bahan Ajar Informatika Kelas X Pertemuan 2

**BAHAN****AJAR****Pertemuan 3****BAHAN AJAR INFORMATIKA KELAS X SMA****Nested If (Kondisi Bersarang)**

Teori, Flowchart, dan Proyek PjBL

<b>Mata Pelajaran / Fase</b>	Informatika / Fase E
<b>Kelas / Semester</b>	X (Sepuluh) / Ganjil
<b>Alokasi Waktu</b>	2 × 45 Menit
<b>Penyusun</b>	Firandi Julian & Ikhsan Budi Maulana
<b>Sekolah</b>	SMA Negeri 2 Padang

**TUJUAN PEMBELAJARAN**

1. Menjelaskan konsep nested if dan perbedaannya dari ifelse biasa (C2)
2. Menerapkan nested if untuk masalah yang membutuhkan kondisi berlapis (C3)
3. Menganalisis alur eksekusi nested if menggunakan flowchart dan tracing (C4)

**A. PERTANYAAN PEMANTIK**

Bayangkan kamu masuk wahana di taman hiburan. Syaratnya: tinggi badan minimal 120 cm ATAU berusia di atas 12 tahun. Jika lolos syarat itu, barulah diperiksa: apakah membawa tiket yang valid?

**Ini adalah kondisi BERLAPIS di dalam kondisi ada kondisi lagi. Inilah nested if!**

**B. APA ITU NESTED IF?**

Nested if (kondisi bersarang) adalah struktur di mana terdapat satu atau lebih blok if di dalam blok if lainnya. Kondisi dalam HANYA diperiksa jika kondisi luar sudah terpenuhi (True).

**Perbedaan utama:**

ifelifelse Biasa	Nested If
<p>Kondisi diperiksa secara berurutan satu level</p> <pre> if A:     ... elif B:     ... else:     ... </pre>	<p>Kondisi bertingkat di dalam kondisi ada kondisi lagi</p> <pre> if A:     if B:         ...     else:         ... else:     ... </pre>

### C. SINTAKS NESTED IF

```

# Sintaks dasar nested if
if kondisi_luar:
    # Masuk ke sini jika kondisi_luar True
    if kondisi_dalam:
        # Masuk ke sini jika kondisi_dalam JUGA True
        pernyataan_A
    else:
        # kondisi_luar True, tapi kondisi_dalam False
        pernyataan_B
else:
    # kondisi_luar False kondisi_dalam TIDAK DIPERIKSA
    pernyataan_C

```

**⚠️ KUNCI: Indentasi menentukan tingkat nested if. Python sangat ketat soal indentasi!**

```

# Contoh: Sistem Tiket Bioskop
hari = input('Hari (weekend/weekday): ')
usia = int(input('Usia Anda: '))
kursi = input('Kelas kursi (reguler/premium): ')

```

```

if hari == 'weekend':
    if usia < 5:
        harga = 0
        ket = 'Gratis (anak balita)'
    elif usia < 12:
        harga = 25000
        ket = 'Anak'
    else:
        if kursi == 'premium':
            harga = 80000
        else:
            harga = 50000
        ket = 'Dewasa Weekend'
else: # weekday
    if usia < 12:
        harga = 15000
        ket = 'Anak Weekday'
    else:
        harga = 35000
        ket = 'Dewasa Weekday'

print(f'Tiket {ket}: Rp {harga:,}')

```

#### D. MEMBACA FLOWCHART NESTED IF

Flowchart nested if memiliki ciri khas: setelah satu keputusan (berlian/diamond), ada keputusan lagi.  
 Cara membacanya:

- Setiap berlian (◇) = satu kondisi if
- Garis 'Ya' = kondisi True → masuk ke blok if
- Garis 'Tidak' = kondisi False → masuk ke blok else
- Berlian bertingkat = nested if

[ Gambarlah flowchart sistem tiket bioskop di atas di sini ]

## E. NESTED IF vs OPERATOR and PERBANDINGAN

```
# SITUASI: Masuk wahana jika tinggi >= 120 cm DAN membawa tiket valid
```

```
# Cara 1 Nested If:
```

```
if tinggi >= 120:
```

```
    if punya_tiket:
```

```
        print('Boleh masuk')
```

```
    else:
```

```
        print('Tidak punya tiket')
```

```
else:
```

```
    print('Tinggi kurang')
```

```
# Cara 2 Operator and:
```

```
if tinggi >= 120 and punya_tiket:
```

```
    print('Boleh masuk')
```

```
else:
```

```
    print('Tidak bisa masuk')
```

```
# PERHATIKAN perbedaan output elsenya!
```

```
# Nested if lebih detail: bisa memberi pesan berbeda untuk setiap kondisi
```

```
# Operator and lebih ringkas: tapi pesan error lebih generik
```

1. Gunakan nested if saat kondisi dalam BERGANTUNG pada kondisi luar, atau saat membutuhkan pesan/tindakan berbeda untuk setiap kombinasi kondisi.
2. Gunakan operator and/or saat kondisi independen dan hanya butuh satu respons untuk kondisi terpenuhi/tidak.

**BAHAN****AJAR****Pertemuan 4****BAHAN AJAR INFORMATIKA KELAS X SMA****Evaluasi & Asesmen Nested If**

Komparasi Pendekatan, Debugging Lanjutan, dan Kuis

<b>Mata Pelajaran / Fase</b>	Informatika / Fase E
<b>Kelas / Semester</b>	X (Sepuluh) / Ganjil
<b>Alokasi Waktu</b>	2 × 45 Menit
<b>Penyusun</b>	Firandi Julian & Ikhsan Budi Maulana
<b>Sekolah</b>	SMA Negeri 2 Padang

**TUJUAN PEMBELAJARAN**

- Mengevaluasi kapan sebaiknya menggunakan nested if dibanding pendekatan lain (C5)
- Menulis program nested if yang berfungsi benar, rapi, dan terdokumentasi (P3)
- Melakukan debugging nested if secara sistematis (C4)

**A. RANGKUMAN MATERI NESTED IF**

Poin-poin kunci yang perlu dikuasai sebelum asesmen:

1. Nested if = if di dalam if kondisi dalam hanya diperiksa jika kondisi luar terpenuhi
2. Indentasi Python WAJIB konsisten (4 spasi per level) indentasi yang salah adalah sumber bug #1 pada nested if
3. else terhubung ke if pada level indentasi yang sama bukan if terdekat secara fisik
4. Nested if cocok untuk kondisi bertingkat (login, tiket, grade + bonus)
5. Gunakan komentar (# ...) di setiap blok if/else untuk memudahkan pembacaan kode

**# TEMPLATE DOKUMENTASI NESTED IF YANG BAIK**

```
username = input('Username: ')
```

```

password = input('Password: ')

# Tingkat 1: Cek username
if username == 'admin':
    # Tingkat 2: Cek password (hanya jika username benar)
    if password == 'admin123':
        level = 'superadmin'
        print(f'Login berhasil! Level: {level}')
    else:
        print('Password salah!')
elif username == 'guru':
    if password == 'guru2025':
        level = 'teacher'
        print(f'Login berhasil! Level: {level}')
    else:
        print('Password salah!')
else:
    print('Username tidak ditemukan!')

```

## B. DEBUGGING NESTED IF PANDUAN LENGKAP

### 3 Jenis Bug yang Paling Sering Muncul pada Nested If:

#### **Bug #1 Else terhubung ke if yang salah karena indentasi tidak tepat:**

```

# BUG Indentasi else tidak konsisten:
if A:
    if B:
        print('A dan B benar')
else:      # Ini terhubung ke 'if B', bukan 'if A!'
    print('???')

# BENAR else di level yang tepat:
if A:
    if B:

```

```

    print('A dan B benar')
else:    # Terhubung ke 'if B'
    print('A benar, B salah')
else:    # Terhubung ke 'if A'
    print('A salah')

```

## Bug #2 Variabel tidak terdefinisi untuk semua jalur eksekusi:

```

# BUG 'status' tidak terdefinisi jika kondisi tidak terpenuhi:
if nilai >= 75:
    if kehadiran >= 80:
        status = 'Naik Kelas'
print(status) # NameError jika nilai < 75!

# BENAR selalu definisikan nilai default:
status = 'Tidak Naik' # Nilai default
if nilai >= 75:
    if kehadiran >= 80:
        status = 'Naik Kelas'
    else:
        status = 'Kehadiran Kurang'
print(status) # Selalu aman

```

## C. TEKNIK MEMBACA KODE NESTED IF

Cara tracing kode nested if: Buat tabel dengan kolom untuk setiap variabel input, dan ikuti alur eksekusi:

```

# Program Penentuan Grade + Bonus Kehadiran
nilai = float(input('Nilai: '))
kehadiran = float(input('Kehadiran (%): '))

if nilai >= 80:
    if kehadiran >= 90:

```

```

grade = 'A+'
bonus = 'Dapat bonus beasiswa'
elif kehadiran >= 75:
    grade = 'A'
    bonus = 'Tidak ada bonus'
else:
    grade = 'A'
    bonus = 'Perlu perbaikan kehadiran'
elif nilai >= 70:
    if kehadiran >= 90:
        grade = 'B+'
        bonus = 'Dapat nilai tambah'
    else:
        grade = 'B'
        bonus = 'Standar'
else:
    grade = 'C atau lebih rendah'
    bonus = 'Remedial wajib'

print(f'Grade: {grade} | {bonus}')

```

Latihan: Lakukan tracing untuk input (nilai=85, kehadiran=92), (nilai=75, kehadiran=80), (nilai=65, kehadiran=95). Tulis jalur eksekusi dan output masing-masing!

**BAHAN****AJAR****Pertemuan 5****BAHAN AJAR INFORMATIKA KELAS X SMA****Struktur Perulangan For Loop & While Loop**

Konsep, Komponen, Sintaks, dan Pseudocode

<b>Mata Pelajaran / Fase</b>	Informatika / Fase E
<b>Kelas / Semester</b>	X (Sepuluh) / Ganjil
<b>Alokasi Waktu</b>	2 × 45 Menit
<b>Penyusun</b>	Firandi Julian & Ikhsan Budi Maulana
<b>Sekolah</b>	SMA Negeri 2 Padang

**TUJUAN PEMBELAJARAN**

- Menganalisis konsep dan fungsi struktur perulangan dalam pemrograman (C4)
- Membandingkan karakteristik, kelebihan, dan kasus penggunaan for loop dan while loop (C4)
- Menulis pseudocode perulangan untuk menyelesaikan masalah sederhana (P3)

**A. PERTANYAAN PEMANTIK**

Bayangkan kamu diminta mencetak 'Halo' sebanyak 100 kali. Tanpa loop, kamu harus menulis `print('Halo')` sebanyak 100 baris. Dengan loop, cukup 2 baris kode!

**Perulangan adalah mekanisme yang menggerakkan game, otomatisasi data, animasi, dan hampir semua program nyata.**

**B. KONSEP PERULANGAN (LOOPING)**

Perulangan adalah struktur kontrol yang memungkinkan sekelompok pernyataan dieksekusi berulang kali selama kondisi tertentu terpenuhi. Ada 4 komponen utama:

Komponen	Fungsi	Contoh
<b>Kondisi / Counter</b>	Menentukan kapan loop berhenti	$i < 10$ , $jumlah > 0$
<b>Variabel Kontrol</b>	Variabel yang mengontrol loop	$i$ , $n$ , $counter$ , $total$
<b>Body Loop</b>	Kode yang dieksekusi setiap iterasi	$print(i)$ , $total += i$
<b>Update</b>	Perubahan variabel setelah tiap iterasi	$i += 1$ , $n = 1$

### C. FOR LOOP Sintaks dan Penggunaan

For loop digunakan saat JUMLAH ITERASI SUDAH DIKETAHUI sebelumnya.

# Sintaks dasar for loop:

```
for variabel in range(berhenti):          # 0, 1, 2, ..., berhenti1
for variabel in range(mulai, berhenti):   # mulai, mulai+1, ..., berhenti1
for variabel in range(mulai, berhenti, step): # mulai, mulai+step, ...
```

# Contoh 1 Cetak angka 110:

```
for i in range(1, 11):
    print(i)
```

# Contoh 2 Hitung jumlah  $1+2+\dots+100$ :

```
total = 0
for i in range(1, 101):
    total += i
print(f'Jumlah 1100: {total}')
```

# Contoh 3 Iterasi pada list:

```
buah = ['apel', 'mangga', 'jeruk']
for b in buah:
    print(f'Saya suka {b}')
```

```
# Contoh 4 Iterasi pada string:
kata = 'PYTHON'
for huruf in kata:
    print(huruf, end=") # Output: PYTHON
```

## D. WHILE LOOP Sintaks dan Penggunaan

While loop digunakan saat KONDISI BERHENTI BELUM PASTI sebelum loop dimulai.

```
# Sintaks dasar while loop:
while kondisi:
    # Blok ini dijalankan selama kondisi True
    # WAJIB ada update variabel agar loop tidak infinite!

# Contoh 1 Hitung mundur dari N:
n = int(input('Mulai dari: '))
while n > 0:
    print(n)
    n = 1 # UPDATE WAJIB!
print('Selesai!')

# Contoh 2 Validasi input pengguna:
password = ""
while password != 'rahasia123':
    password = input('Masukkan password: ')
    if password != 'rahasia123':
        print('Password salah, coba lagi!')
print('Akses diterima!')
```

## E. PERBANDINGAN FOR vs WHILE

Aspek	For Loop	While Loop
<b>Jumlah iterasi</b>	Diketahui sebelumnya	Tidak selalu diketahui
<b>Kontrol utama</b>	range(), list, string	Kondisi boolean
<b>Risiko</b>	Offbyone error	Infinite loop
<b>Kasus ideal</b>	Tabel perkalian, iterasi data	Validasi input, game loop
<b>Perlu counter manual?</b>	Tidak (range otomatis)	Ya (harus update manual)

**INGAT:** Kedua jenis loop bisa saling menggantikan dalam banyak kasus. Pilih yang lebih natural untuk masalah yang sedang diselesaikan.

**BAHAN****AJAR****Pertemuan 6****BAHAN AJAR INFORMATIKA KELAS X SMA****Perulangan Lanjutan: Nested Loop, Break & Continue**

Praktik Coding, Debugging Loop, dan Asesmen Sumatif

<b>Mata Pelajaran / Fase</b>	Informatika / Fase E
<b>Kelas / Semester</b>	X (Sepuluh) / Ganjil
<b>Alokasi Waktu</b>	2 × 45 Menit
<b>Penyusun</b>	Firandi Julian & Ikhsan Budi Maulana
<b>Sekolah</b>	SMA Negeri 2 Padang

**TUJUAN PEMBELAJARAN**

- Mengimplementasikan for loop dan while loop dengan output yang benar (C3)
- Merancang nested loop serta pernyataan break/continue (C4)
- Melakukan debugging terhadap bug umum pada loop (C4)

**A. BREAK, CONTINUE, DAN ELSE PADA LOOP**

# BREAK Menghentikan loop seketika:

```
for i in range(1, 11):  
    if i == 5:  
        print('Berhenti di angka 5!')  
        break # Loop langsung berhenti  
    print(i)
```

# Output: 1 2 3 4 Berhenti di angka 5!

# CONTINUE Melompati iterasi tertentu:

```
for i in range(1, 11):
```

```

if i % 2 == 0:
    continue # Lewati angka genap
print(i, end=' ')
# Output: 1 3 5 7 9

# ELSE pada LOOP Dijalankan jika loop SELESAI TANPA break:
for i in range(2, 10):
    if 15 % i == 0:
        print(f'15 habis dibagi {i}')
        break
    else:
        print('Tidak ditemukan pembagi') # Hanya jika tidak ada break

```

## B. NESTED LOOP (PERULANGAN BERSARANG)

Nested loop = loop di dalam loop. Loop dalam akan menyelesaikan SELURUH iterasinya sebelum loop luar berpindah ke iterasi berikutnya.

Analogi: Jarum menit berputar penuh 60 kali (loop dalam) sebelum jarum jam berpindah 1 angka (loop luar).

```

# Contoh 1 Tabel Perkalian 3x3:
for baris in range(1, 4): # Loop luar: 3 iterasi
    for kolom in range(1, 4): # Loop dalam: 3 iterasi per baris
        print(f'{baris} x {kolom} = {baris * kolom}', end=' ')
    print() # Pindah baris setelah setiap baris selesai

# Contoh 2 Pola Segitiga Bintang:
n = 5
for i in range(1, n + 1): # Loop luar: kontrol baris
    for j in range(i): # Loop dalam: kontrol jumlah bintang
        print('*', end='')
    print() # Pindah baris

```

```

# Output:
# *
# **
# ***
# ****
# *****

# Contoh 3 Mencari bilangan prima (nested loop + break):
for angka in range(2, 20):
    adalah_prima = True
    for i in range(2, angka):
        if angka % i == 0:
            adalah_prima = False
            break # Tidak perlu cek faktor lain
    if adalah_prima:
        print(angka, end=' ')
# Output: 2 3 5 7 11 13 17 19

```

## C. DEBUGGING BUG UMUM PADA LOOP

### Bug #1 Infinite Loop (lupa update variabel kontrol)

```

# BUG Loop tidak pernah berhenti:
i = 0
while i < 10:
    print(i)
    # LUPA: i += 1

# PERBAIKAN:
i = 0
while i < 10:
    print(i)
    i += 1 # UPDATE WAJIB!

```

## Bug #2 Offbyone Error (batas range salah)

```
# BUG Harusnya cetak 110, tapi range(1, 10) hanya 19:
for i in range(1, 10):
    print(i) # Output: 1 2 3 4 5 6 7 8 9 ← 10 tidak tercetak!

# PERBAIKAN range(1, 11) untuk mendapat 110:
for i in range(1, 11):
    print(i)
```

## Bug #3 Logika akumulator salah

```
# BUG total selalu = nilai terakhir, bukan jumlah:
total = 0
for i in range(1, 6):
    total = i # BUG! Harusnya total += i
print(total) # Output: 5 (bukan 15)

# PERBAIKAN:
total = 0
for i in range(1, 6):
    total += i # Akumulasi yang benar
print(total) # Output: 15
```

## D. LATIHAN MANDIRI

- Latihan 1: Buat program FizzBuzz (1100): cetak 'Fizz' untuk kelipatan 3, 'Buzz' untuk kelipatan 5, 'FizzBuzz' untuk kelipatan keduanya
- Latihan 2: Buat program yang membaca 10 angka dari pengguna dan menampilkan: jumlah, rata-rata, nilai terbesar, dan nilai terkecil
- Latihan 3 (Pengayaan): Buat pola berlian (\*) menggunakan nested loop

## Referensi Tambahan:

- PythonOfficial Docs Loops: <https://docs.python.org/3/tutorial/controlflow.html#forstatements>
- W3Schools Python For Loops: [https://www.w3schools.com/python/python\\_for\\_loops.asp](https://www.w3schools.com/python/python_for_loops.asp)

<b>Nama Kelompok</b>	: _____	<b>Kelompok</b>	: A / B
<b>Anggota</b>	: 1. _____ 2. _____	<b>Kelas / Tgl</b>	: _____ _____

## BAGIAN A — Pemahaman Konsep

Petunjuk: Jawab pertanyaan berikut sebelum memulai pengerjaan proyek!

**1. Jelaskan perbedaan antara if tunggal dan if-elif-else! Berikan contoh situasi yang membutuhkan masing-masing.**

---



---



---

**2. Perhatikan kode berikut. Identifikasi minimal 2 kesalahan logika di dalamnya!**

```
nilai = 75
if nilai > 90:
    print('A')
if nilai > 75:
    print('B')
if nilai > 60:
    print('C')
if nilai <= 60:
    print('D')
```

Kode dengan Kesalahan (di atas)	Kode yang Diperbaiki:
Kesalahan 1: _____	_____
Kesalahan 2: _____	_____
_____	_____

## BAGIAN B — Perencanaan Proyek

**Kelompok A: Program Penentu Kelulusan + Jenis Bilangan**

**Kelompok B: Program Penentu Diskon dengan Percabangan (≥500rb: 20%, 200rb-499rb: 10%, <200rb: 0%)**

### B.1 Rancangan Fitur Program

No	Fitur Program	Deskripsi Logika
----	---------------	------------------

1.		
2.		
3.		

## B.2 Diagram Alir (Flowchart) Program

[ Gambar flowchart program di sini ]

## B.3 Pembagian Tugas Kelompok

Nama Anggota	Peran	Tugas Spesifik
_____	Perancang Logika	_____
_____	Penulis Kode	_____
_____	Penguji Program	_____
_____	Presenter	_____

## BAGIAN C — Pengerjaan Proyek

### C.1 Template Kode Program (Kelompok A — Penentu Kelulusan & Jenis Bilangan)

Salin dan lengkapi kode berikut:

```
# =====
# Kelompok A: Program Penentu Kelulusan & Jenis Bilangan
# Nama Kelompok : _____
# Kelas          : _____
# =====

# --- BAGIAN 1: Penentu Kelulusan ---
nama = input('Masukkan nama siswa : ')
nilai = float(input('Masukkan nilai          : '))

if nilai >= 75:
    status = _____ # Isi: teks status lulus
    keterangan = 'Selamat! Kamu berhasil.'
elif nilai >= 60:
    status = _____ # Isi: teks status remedial
    keterangan = 'Kamu perlu mengikuti remedial.'
else:
    status = _____ # Isi: teks status tidak lulus
    keterangan = 'Kamu belum mencapai KKM.'

print(f'Siswa {nama} - Nilai: {nilai} - Status: {status}')
print(keterangan)

# --- BAGIAN 2: Penentu Jenis Bilangan ---
angka = int(input('Masukkan sebuah bilangan: '))

if angka > 0:
    jenis = _____ # Isi sendiri
```

```

elif angka < 0:
    jenis = _____ # Isi sendiri
else:
    jenis = _____ # Isi sendiri

print(f'{angka} adalah bilangan {jenis}')

```

### C.1 Template Kode Program (Kelompok B — Penentu Diskon)

```

# =====
# Kelompok B: Program Penentu Diskon Belanja
# Nama Kelompok : _____
# Kelas         : _____
# =====

print('='*50)
print('    TOKO ELEKTRONIK CERIA')
print('='*50)

nama_pembeli = input('Nama Pembeli    : ')
total_belanja = float(input('Total Belanja    : Rp '))

if total_belanja >= 500000:
    diskon_persen = _____ # Isi: 20
    kategori = 'Pelanggan VIP'
elif total_belanja >= 200000:
    diskon_persen = _____ # Isi: 10
    kategori = 'Pelanggan Reguler'
else:
    diskon_persen = _____ # Isi: 0
    kategori = 'Pelanggan Biasa'

diskon_nominal = total_belanja * diskon_persen / 100
bayar          = total_belanja - diskon_nominal

print(f'Kategori      : {kategori}')
print(f'Diskon        : {diskon_persen}% = Rp {diskon_nominal:,.0f}')
print(f'Total Bayar   : Rp {bayar:,.0f}')

```

### C.2 Tabel Pengujian Program (Test Case) — Uji dengan minimal 5 variasi input

No	Input	Output Diharapkan	Output Aktual	Status
1.				Benar / Salah
2.				Benar / Salah
3.				Benar / Salah
4.				Benar / Salah
5.				Benar / Salah

## BAGIAN D — Refleksi

### 1. Apa yang paling menantang dari menulis kondisi dalam program?

---



---

2. Berikan satu contoh masalah sehari-hari yang akan lebih mudah dipecahkan dengan if-else!

---

---

3. Apa yang akan kelompok Anda perbaiki jika mengerjakan proyek ini lagi?

---

---

Kriteria	Indikator	Skor
Pemahaman Konsep (Bagian A)	Jawaban tepat dan lengkap	25
Perencanaan Proyek (Bagian B)	Flowchart dan pembagian tugas jelas	25
Kode Program Berjalan (Bagian C)	Program berjalan sesuai semua test case	30
Refleksi (Bagian D)	Jawaban reflektif dan konstruktif	20

Nama Siswa	_____
Kelas / Tanggal	_____ / _____
Alokasi Waktu	90 Menit

## BAGIAN A — Sesi Debugging Terstruktur

Petunjuk: Temukan semua bug pada kode berikut, jelaskan mengapa salah, dan tulis kode yang benar!

### Bug #1 — Input tidak dikonversi ke tipe data yang benar

```
umur = input('Masukkan umur Anda: ')
if umur >= 17:
    print('Anda boleh membuat KTP')
else:
    print('Anda belum cukup umur')
```

Jelaskan bug yang ditemukan:

---

---

Tulis kode yang sudah diperbaiki:

```
# Kode perbaikan di sini:
```

### Bug #2 — Penggunaan dua if terpisah yang seharusnya if-elif

```
skor = 85
if skor >= 90:
    grade = 'A'
if skor >= 75:
    grade = 'B'
if skor >= 60:
    grade = 'C'
print(f'Grade Anda: {grade}')
```

Jelaskan apa yang terjadi saat skor = 85 dengan kode di atas:

---

---

Tulis kode yang sudah diperbaiki:

```
# Kode perbaikan:
```

### Bug #3 — Operator perbandingan yang tidak valid

```
harga = 150000
if harga => 100000:
    print('Harga di atas 100 ribu')
elif harga =< 50000:
    print('Harga di bawah 50 ribu')
else:
    print('Harga antara 50 ribu - 100 ribu')
```

Identifikasi semua operator yang salah dan tuliskan perbaikannya:

---

---

---

### BAGIAN B — Kuis Tertulis (Pilihan Ganda)

Lingkari jawaban yang paling tepat!

1. Fungsi `int()` pada program Python digunakan untuk...

- A. Mengubah string menjadi bilangan bulat
- B. Mengubah bilangan menjadi huruf
- C. Mencetak teks ke layar
- D. Membuat variabel baru

2. Manakah sintaks percabangan yang benar di Python?

- A. `if x = 10:`
- B. `if x == 10:`
- C. `if (x == 10) then:`
- D. `if x equal 10:`

3. Jika nilai = 72, blok kode mana yang dijalankan: `if nilai >= 80: ... elif nilai >= 70: ... else: ...`

- A. Blok `if`
- B. Blok `elif`
- C. Blok `else`
- D. Semua blok dijalankan

4. Perbedaan utama antara `elif` dan `else` adalah...

- A. `elif` tidak membutuhkan kondisi, `else` membutuhkan
- B. `elif` memeriksa kondisi tertentu, `else` dijalankan jika semua kondisi salah
- C. `elif` dan `else` adalah sama
- D. `else` lebih cepat dari `elif`

5. Apa output program berikut: `x=5; if x>3: print('A'); if x>7: print('B')`

- A. A
- B. B

- C. AB
- D. Tidak ada output

 **BAGIAN C — Soal Esai**

Jawablah pertanyaan berikut dengan lengkap dan jelas!

**1. [C2] Jelaskan analogi kehidupan nyata untuk struktur if-elif-else! Gunakan contoh selain dari pembelajaran ini.**

---

---

---

**2. [C3] Tulis program Python kalkulator diskon: belanja  $\geq$  Rp1.000.000  $\rightarrow$  diskon 25%,  $\geq$  Rp500.000  $\rightarrow$  15%,  $\geq$  Rp200.000  $\rightarrow$  10%, sisanya  $\rightarrow$  0%.**

```
# Tulis kode Python di sini:
```

**3. [C4] Analisis bug: kode berikut salah ketika harga = 300000, mengapa?  $\rightarrow$  if harga < 200000: ... if harga < 500000: ... if harga  $\geq$  500000: ...**

---

---

---

**4. [C5] Evaluasi dua pendekatan untuk cek berat badan ideal (if-elif-else berlapis vs operator and/or). Kapan masing-masing lebih efisien? Jelaskan dengan contoh!**

---

---

---

Kriteria	Indikator	Skor
Debugging (Bagian A)	Semua bug teridentifikasi & kode diperbaiki benar	30
Kuis Pilihan Ganda (Bagian B)	Jawaban tepat (6 poin/soal)	30
Esai C2 & C3 (No.1 & 2)	Penjelasan akurat, kode berjalan benar	20
Esai C4 & C5 (No.3 & 4)	Analisis dan evaluasi logis dan mendalam	20

**Nama Kelompok**

: \_\_\_\_\_

**Kelompok**

: A / B

**BAGIAN A — Pemahaman Konsep Nested If**

1. Jelaskan perbedaan antara nested if dan if-elif-else biasa! Kapan sebaiknya menggunakan nested if?

---

---

---

2. Perhatikan kode nested if berikut. Lakukan tracing manual: tulis output untuk setiap kombinasi input!

```
hari = input('Hari (weekend/weekday): ')
usia = int(input('Usia: '))

if hari == 'weekend':
    if usia < 12:
        harga = 25000
        kategori = 'Anak - Weekend'
    else:
        harga = 50000
        kategori = 'Dewasa - Weekend'
else:
    if usia < 12:
        harga = 15000
        kategori = 'Anak - Weekday'
    else:
        harga = 35000
        kategori = 'Dewasa - Weekday'

print(f'{kategori}: Rp {harga:,}')
```

Input: hari	Input: usia	Output (kategori)	Output (harga)
weekend	8	_____	_____
weekend	25	_____	_____
weekday	10	_____	_____
weekday	30	_____	_____

**BAGIAN B — Perencanaan Proyek**

**Kelompok A: Sistem Nilai Siswa (grade A/B/C/D + status remedial berdasarkan kehadiran)**

**Kelompok B: Sistem Tiket Bioskop (berdasarkan hari, usia, dan kelas kursi)**

### B.1 Tabel Kombinasi Input-Output (buat minimal 6 kombinasi)

No	Input 1	Input 2	Input 3 (jika ada)	Output yang Diharapkan
1				
2				
3				
4				
5				
6				

### B.2 Gambar Flowchart Nested If Program Anda

[ Gambar flowchart nested if di sini ]

## BAGIAN C — Pengerjaan Proyek

Tulis kode Python lengkap program Anda di bawah ini:

```
# =====  
# PROYEK PERTEMUAN 3 - NESTED IF  
# Nama Kelompok : _____  
# Kelas          : _____  
# Deskripsi      : _____  
# =====  
  
# Kelompok A: nilai + kehadiran → grade + status remedial  
# Kelompok B: hari + usia + kelas kursi → harga tiket  
  
# --- Tulis kode di sini ---
```

### Tabel Pengujian (uji minimal 6 kombinasi input):

No	Input (semua variabel)	Output Diharapkan	Output Aktual & Status
1			

2			
3			
4			
5			
6			

## BAGIAN D — Refleksi

1. Apa bedanya nested if dibanding operator and? Kapan masing-masing lebih tepat digunakan?

---



---



---

2. Berikan satu contoh situasi nyata yang membutuhkan nested if minimal 2 tingkat!

---




---

3. Tuliskan 3 baris kode nested if berdasarkan kasus yang kelompok Anda kerjakan hari ini!

Kriteria	Indikator	Skor
Pemahaman & Tracing (Bagian A)	Tracing manual benar untuk semua 4 kombinasi	<b>25</b>
Perencanaan (Bagian B)	Tabel kombinasi & flowchart nested if lengkap	<b>25</b>
Program Berjalan (Bagian C)	Semua 6 test case menghasilkan output benar	<b>30</b>
Refleksi (Bagian D)	Jawaban reflektif dan kode contoh benar	<b>20</b>

Nama Siswa	_____
Kelas / Tanggal	_____ / _____

 **BAGIAN A — Evaluasi dan Komparasi Pendekatan**

Dua pendekatan berikut menyelesaikan masalah yang sama. Analisis dan bandingkan!

Pendekatan 1 — Nested If	Pendekatan 2 — Operator and/or
<pre>bmi = float(input('BMI: ')) jk = input('Jenis Kelamin (P/L): ') if bmi &lt; 18.5:     if jk == 'P':         status = 'Wanita Kurus'     else:         status = 'Pria Kurus' elif bmi &lt; 25:     if jk == 'P':         status = 'Wanita Normal'     else:         status = 'Pria Normal' else:     status = 'Kelebihan Berat' print(status)</pre>	<pre>bmi = float(input('BMI: ')) jk = input('Jenis Kelamin (P/L): ') if bmi &lt; 18.5 and jk == 'P':     status = 'Wanita Kurus' elif bmi &lt; 18.5 and jk == 'L':     status = 'Pria Kurus' elif bmi &lt; 25 and jk == 'P':     status = 'Wanita Normal' elif bmi &lt; 25 and jk == 'L':     status = 'Pria Normal' else:     status = 'Kelebihan Berat' print(status)</pre>

**Diskusikan dan jawab: Pendekatan mana yang lebih mudah dibaca? Mana yang lebih mudah dikembangkan? Untuk kasus mana pendekatan 1 lebih baik? Kasus mana pendekatan 2 lebih baik?**

---



---



---



---

 **BAGIAN B — Debugging Nested If**

Temukan semua kesalahan pada kode berikut (ada 2 bug: indentasi dan else yang salah):

```
kategori_usia = input('Kategori (anak/remaja/dewasa): ')
penghasilan = int(input('Penghasilan per bulan: Rp '))

if kategori_usia == 'dewasa':
    if penghasilan >= 5000000:
        pajak = penghasilan * 0.15
        status = 'Wajib Pajak'
    else:
        pajak = 0
        status = 'Tidak Kena Pajak'
else:
    if penghasilan >= 3000000:
```

```
pajak = penghasilan * 0.10
else:
    pajak = 0
print(f'Status: {status}, Pajak: Rp {pajak:,.0f}')
```

**Bug #1 — Identifikasi dan jelaskan:**

---

---

**Bug #2 — Identifikasi dan jelaskan:**

---

---

**Tuliskan kode yang sudah diperbaiki:**

```
# Kode perbaikan:
```

## BAGIAN C — Asesmen Sumatif Nested If

Soal Pilihan Ganda (lingkari jawaban yang benar):

**1. Pada nested if, blok else paling dalam akan terhubung dengan...**

- A. if paling luar
- B. if paling dalam (if terdekat)
- C. semua if
- D. tergantung kondisi

**2. Apa yang terjadi jika indentasi nested if tidak konsisten di Python?**

- A. Program berjalan lebih cepat
- B. IndentationError / logika salah
- C. Tidak ada pengaruh
- D. Variabel terhapus

**3. Kapan nested if lebih tepat dibanding operator and?**

- A. Saat kondisi dalam bergantung pada kondisi luar
- B. Saat semua kondisi independen
- C. Saat menggunakan loop
- D. Selalu lebih baik

**4. Berapa banyak kombinasi input yang minimal harus diuji pada nested if 2 tingkat dengan 2 cabang masing-masing?**

- A. 2
- B. 4

- C. 6
- D. 8

**5. Pada program login (cek username dulu, baru password), struktur yang paling tepat adalah...**

- A. if username and password:
- B. Nested if: cek username, jika benar baru cek password
- C. Dua if terpisah
- D. While loop

**Soal Esai:**

**1. [C2] Gambarkan flowchart nested if untuk sistem login: cek username → jika benar, cek password → tentukan level akses (admin/user).**

[ Gambar flowchart di sini ]

**2. [C3] Tulis program Python sistem login sederhana menggunakan nested if (username='admin', password='12345', level='superadmin' jika username='root', password='root123').**

```
# Kode Python:
```

Kriteria	Indikator	Skor
Analisis Komparasi (Bagian A)	Argumen logis dan tepat untuk kedua pendekatan	25
Debugging (Bagian B)	Semua bug teridentifikasi & kode diperbaiki	25
Pilihan Ganda (Bagian C.1)	Jawaban benar (5 poin/soal)	25
Esai (Bagian C.2)	Flowchart tepat, kode berjalan dengan benar	25

Nama Kelompok

: \_\_\_\_\_

Kelas /  
Tgl

: \_\_\_\_\_

## BAGIAN A — Pemahaman Konsep Looping

1. Lengkapi tabel perbandingan For Loop vs While Loop di bawah ini!

Aspek	For Loop	While Loop
Digunakan saat	Jumlah iterasi diketahui	_____
Risiko utama	_____	Infinite loop
Sintaks dasar	for i in range(n):	_____
Kapan dihentikan	Setelah n iterasi	_____
Contoh penggunaan	Cetak 1-100	_____

## BAGIAN B — Latihan Tracing Kode

Tracing 1 — Tulis output yang dihasilkan untuk setiap langkah iterasi!

```
total = 0
for i in range(1, 6):
    total = total + i
    print(f'Iterasi {i}: total = {total}')
print(f'Total akhir: {total}')
```

Iterasi (i)	Nilai total sebelum	total = total + i	Output print
1	_____	_____	_____
2	_____	_____	_____
3	_____	_____	_____
4	_____	_____	_____
5	_____	_____	_____

Tracing 2 — Tulis output program while loop berikut langkah demi langkah:

```
n = 10
while n > 0:
    if n % 2 == 0:
        print(f'{n} genap')
    n -= 3
print('Selesai')
```

Output:

---

---

---

---

---

 **BAGIAN C — Menyusun Pseudocode Perulangan**

Susun pseudocode untuk masalah berikut, lalu tulis kode Python-nya!

**Masalah 1: Hitung jumlah semua bilangan ganjil antara 1 sampai 100**

Pseudocode	Kode Python
<hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	

**Masalah 2: Cetak tabel perkalian untuk angka yang diinput pengguna ( $1 \times n$  sampai  $10 \times n$ )**

Pseudocode	Kode Python
<hr/> <hr/> <hr/> <hr/> <hr/> <hr/>	

 **BAGIAN D — Refleksi**

1. Satu konsep loop yang sudah kamu pahami dengan baik:

---

---

2. Satu hal yang masih membingungkan tentang loop:

---

---

3. Satu pertanyaan yang ingin kamu tanyakan kepada guru:

---

Kriteria	Indikator	Skor
Tabel Perbandingan (Bagian A)	Semua sel terisi tepat dan lengkap	25
Tracing Kode (Bagian B)	Semua iterasi dan output ditulis benar	30

Pseudocode & Kode Python (Bagian C)	Pseudocode logis, kode Python berjalan benar	<b>30</b>
Refleksi (Bagian D)	Refleksi jujur dan pertanyaan bermakna	<b>15</b>

Nama Siswa \_\_\_\_\_

Kelas / Tanggal \_\_\_\_\_ / \_\_\_\_\_

**KEGIATAN 1 — For Loop Dasar**

Tulis program yang mencetak pola segitiga bintang menggunakan for loop:

```
# Output yang diharapkan:
# *
# **
# ***
# ****
# *****

n = int(input('Berapa baris segitiga? '))

# Tulis kode for loop di sini:
```

**KEGIATAN 2 — While Loop**

Buat program menebak angka: program memiliki angka rahasia, pengguna terus menebak sampai benar!

```
# Program Tebak Angka menggunakan While Loop
angka_rahasia = 42
tebakan = 0
percobaan = 0

# Tulis while loop di sini:
# - Minta input tebak dari pengguna
# - Tambahkan counter percobaan
# - Jika benar: cetak pesan selamat + jumlah percobaan, hentikan loop
# - Jika terlalu kecil: beri petunjuk 'Terlalu kecil!'
# - Jika terlalu besar: beri petunjuk 'Terlalu besar!'
```

**KEGIATAN 3 — Break dan Continue**

Lengkapi kode berikut dengan break dan continue yang tepat:

```

# Program mencari bilangan prima pertama setelah angka N
import math

N = int(input('Masukkan angka N: '))

for angka in range(N + 1, N + 100):
    adalah_prima = True
    for i in range(2, int(math.sqrt(angka)) + 1):
        if angka % i == 0:
            adalah_prima = False
            # Tulis: break (hentikan loop dalam)
    if adalah_prima:
        print(f'Bilangan prima pertama setelah {N} adalah: {angka}')
        # Tulis: break (hentikan loop luar)

# Tambahan: cetak semua bilangan 1-20, SKIP bilangan kelipatan 3
for i in range(1, 21):
    if i % 3 == 0:
        # Tulis: continue
    print(i, end=' ')

```

## KEGIATAN 4 — Nested Loop

Buat program tabel perkalian menggunakan nested loop:

```

# Program Tabel Perkalian 1-5 x 1-5
# Output yang diharapkan:
# 1 2 3 4 5
# 2 4 6 8 10
# 3 6 9 12 15
# 4 8 12 16 20
# 5 10 15 20 25

# Tulis nested loop di sini:

```

Tabel Pengujian Kegiatan 1-4:

No	Input	Output Diharapkan	Output Aktual	Status
1.				Benar / Salah
2.				Benar / Salah
3.				Benar / Salah
4.				Benar / Salah
5.				Benar / Salah
6.				Benar / Salah

## KEGIATAN 5 — Debugging Loop

Temukan dan perbaiki semua bug pada 3 program berikut!

### Bug #1 — Infinite Loop (lupa update variabel kontrol):

```
total = 0
i = 1
while i <= 10:
    total += i
    # BUG: Apa yang hilang di sini?
print(f'Total: {total}')
```

#### Penyebab bug:

---

Kode perbaikan: tulis baris yang perlu ditambahkan

---

### Bug #2 — Off-by-one Error (batas range salah):

```
# Harusnya mencetak 1, 2, 3, 4, 5
for i in range(1, 5): # BUG: range salah
    print(i)
```

#### Perbaikan range yang benar:

### Bug #3 — Logika akumulator salah:

```
# Harusnya menghitung rata-rata 5 angka input pengguna
total = 0
for i in range(5):
    angka = int(input(f'Angka ke-{i+1}: '))
    total = angka # BUG: harusnya total += angka
rata_rata = total / 5
print(f'Rata-rata: {rata_rata}')
```

#### Jelaskan dampak bug dan tulis perbaikannya:



## ASESMEN SUMATIF — Perulangan

Pilihan Ganda (lingkari jawaban yang benar):

### 1. Fungsi `range(2, 10, 2)` akan menghasilkan angka...

- A. 2, 4, 6, 8, 10
- B. 2, 4, 6, 8
- C. 2, 3, 4, ..., 9
- D. 0, 2, 4, 6, 8

### 2. Pernyataan `break` pada loop berguna untuk...

- A. Melompati satu iterasi
- B. Menghentikan loop seketika
- C. Memulai ulang loop
- D. Menambah counter

### 3. Perbedaan utama `for` loop dan `while` loop adalah...

- A. For lebih cepat dari while
- B. For untuk iterasi terhitung, while untuk kondisi tidak pasti

- C. While tidak bisa menggunakan break
- D. Tidak ada perbedaan

**4. Pada nested loop dengan outer 3 iterasi dan inner 4 iterasi, body loop dalam dieksekusi sebanyak...**

- A. 3 kali
- B. 4 kali
- C. 7 kali
- D. 12 kali

**5. Pernyataan continue pada loop berguna untuk...**

- A. Menghentikan semua loop
- B. Melanjutkan ke iterasi berikutnya dan melewati sisa kode di iterasi ini
- C. Mengulang dari awal
- D. Mendeklarasikan variabel

**Soal Esai Akhir:**

**[C4] Rancang program Python menggunakan nested loop yang mencetak pola berlian berikut dan jelaskan logikanya: \* \*\*\* \*\*\*\*\* \*\*\* \***

```
# Kode Python:
```

**Penjelasan logika kode:**

---



---



---

Kriteria	Indikator	Skor
Kegiatan 1-4 (Praktik Coding)	Semua program berjalan dan test case benar	40
Debugging (Kegiatan 5)	Semua bug teridentifikasi & diperbaiki	20
Pilihan Ganda Asesmen	Jawaban tepat (4 poin/soal)	20
Esai Asesmen	Kode berjalan & penjelasan logika tepat	20