

BAHAN AJAR & JOBSHEET

INFORMATIKA FASE E (KELAS X)

STRUKTUR KONTROL PEMROGRAMAN

C++ & Python

Berbasis Project-Based Learning (PjBL)

Mata Pelajaran Informatika	Fase / Kelas Fase E / Kelas X
Jumlah Pertemuan 6 Pertemuan (2 JP × 6)	Kurikulum Deep Learning
Model Pembelajaran Project-Based Learning (PjBL)	Platform Coding Programmiz (programmiz.com)

A. CAPAIAN & TUJUAN PEMBELAJARAN

Capaian Pembelajaran (CP) Fase E

Pada akhir Fase E, peserta didik mampu menerapkan proses berpikir efektif dan efisien untuk menyelesaikan persoalan secara algoritmik sebagai solusi atas rancangan instruksi dan data yang dapat dijalankan secara efektif dan efisien oleh sistem komputasi, menerapkan berpikir kritis dalam menyikapi beragam data yang tersedia di internet untuk menjadi informasi yang bermanfaat, mempunyai wawasan tentang profesi informatika, serta memahami hak dan kewajiban sebagai warga digital dan aspek hukumnya.

Tujuan Pembelajaran per Pertemuan

Pertemuan	Tujuan Pembelajaran
P-1	Menjelaskan konsep if-elseif-else; menulis dan menjalankan program kondisional di C++ dan Python; menyelesaikan masalah pengambilan keputusan bertingkat; membandingkan sintaks C++ vs Python.
P-2	Menjelaskan nested if; menulis program bersarang di C++ dan Python; menganalisis alur eksekusi; membedakan nested if vs if-elseif-else.
P-3	Menjelaskan for dan while loop; menggunakan keduanya di C++ dan Python; membandingkan sintaks; menyelesaikan masalah berulang.
P-4	Menjelaskan break dan continue; menggunakannya di C++ dan Python; menganalisis perbedaan perilaku program.
P-5	Mendefinisikan fungsi; membuat dan memanggil fungsi; menggunakan parameter dan return value; debugging dasar.
P-6	Menerapkan kondisional, perulangan, dan fungsi dalam proyek; memperbaiki bug; mempresentasikan hasil proyek.

PERTEMUAN 1 STRUKTUR KONDISIONAL IF-ELSEIF-ELSE

📄 Informasi Pertemuan

Alokasi Waktu: 2 × 45 menit | Metode: PjBL Individual

Materi: Struktur kondisional if, else if/elif, else dalam C++ dan Python

Platform: Programmiz Online Compiler (programmiz.com)

BAHAN AJAR

1.1 Konsep Percabangan (Kondisional)

Struktur kondisional adalah mekanisme dalam pemrograman yang memungkinkan program mengambil keputusan berdasarkan kondisi tertentu. Seperti dalam kehidupan nyata: 'Jika hujan, bawa payung; jika tidak, tidak perlu.' Program mengeksekusi blok kode yang berbeda tergantung apakah suatu kondisi bernilai benar (true) atau salah (false).

Diagram Alur Kondisional:

Alur Eksekusi If-Elseif-Else

START → Evaluasi Kondisi 1

├── TRUE → Jalankan Blok-1 → END

└── FALSE → Evaluasi Kondisi 2

 ├── TRUE → Jalankan Blok-2 → END

 └── FALSE → Jalankan Blok-ELSE → END

1.2 Sintaks C++ vs Python

C++ if / else if / else

```
#include <iostream>
using namespace std;
int main() {
    int nilai;
    cout << "Masukkan nilai: ";
    cin >> nilai;
    if (nilai >= 90) {
        cout << "Grade A" << endl;
    } else if (nilai >= 80) {
        cout << "Grade B" << endl;
    } else if (nilai >= 70) {
        cout << "Grade C" << endl;
    } else {
        cout << "Grade D" << endl;
    }
    return 0;
}
```

Python if / elif / else

```
nilai = int(input('Masukkan nilai: '))

if nilai >= 90:
    print('Grade A')
elif nilai >= 80:
    print('Grade B')
elif nilai >= 70:
    print('Grade C')
else:
    print('Grade D')
```

1.3 Perbandingan Sintaks C++ dan Python

Aspek	C++	Python
Keyword percabangan	if / else if / else	if / elif / else
Pembatas blok	Kurung kurawal { }	Indentasi (4 spasi)
Kondisi diapit	Tanda kurung ()	Tidak perlu tanda kurung
Akhir kondisi	Tidak ada tanda khusus	Tanda titik dua (:)
Perbandingan	==, !=, <, >, <=, >=	==, !=, <, >, <=, >=
Operator logika	&& (AND), (OR), ! (NOT)	and, or, not

1.4 Contoh Kasus: Penentuan Kategori BMI

Indeks Massa Tubuh (BMI) = Berat Badan (kg) / Tinggi Badan² (m²). Kategori: <18.5 = Kurus; 18.5–24.9 = Normal; 25–29.9 = Kelebihan berat; ≥30 = Obesitas.

C++ Kalkulator BMI

```
#include <iostream>
using namespace std;
int main() {
    float bb, tb, bmi;
    cout << "Berat badan (kg): ";
    cin >> bb;
    cout << "Tinggi badan (m): ";
    cin >> tb;
    bmi = bb / (tb * tb);
    cout << "BMI: " << bmi << endl;
    if (bmi < 18.5) {
        cout << "Kurus" << endl;
    } else if (bmi < 25) {
        cout << "Normal" << endl;
    } else if (bmi < 30) {
        cout << "Kelebihan BB" << endl;
    } else {
        cout << "Obesitas" << endl;
    }
    return 0;
}
```

Python Kalkulator BMI

```
bb = float(input('Berat badan (kg): '))
tb = float(input('Tinggi badan (m): '))
bmi = bb / (tb ** 2)
print(f'BMI Anda: {bmi:.2f}')

if bmi < 18.5:
    print('Kategori: Kurus')
elif bmi < 25:
    print('Kategori: Normal')
elif bmi < 30:
    print('Kategori: Kelebihan BB')
else:
    print('Kategori: Obesitas')
```

JOBSHEET 1 LEMBAR KERJA PESERTA DIDIK** Petunjuk Pengerjaan JOBSHEET**

1. Buka Programmiz: <https://www.programmiz.com/cpp-programming/online-compiler/> (C++) dan <https://www.programmiz.com/python-programming/online-compiler/> (Python)
2. Ketik program sesuai instruksi, jalankan, lalu SCREENSHOT hasilnya (tampilkan kode + output)
3. Tempelkan screenshot di JOBSHEET digital (Google Classroom) atau cetak dan tempelkan
4. Kerjakan SECARA INDIVIDUAL
5. Simpan semua program sebagai bagian Portofolio Digital

IDENTITAS PESERTA DIDIK

Nama: _____	Kelas: _____
No. Absen: _____	Tanggal: _____

Kegiatan 1 Membuat Program Grade Nilai (C++ dan Python)

Konteks Literasi: Sebuah sekolah menggunakan sistem penilaian berikut: A (90-100), B (80-89), C (70-79), D (60-69), E (<60). Kamu diminta membuat program yang membaca nilai seorang siswa dan menampilkan grade beserta keterangan Lulus/Tidak Lulus (D dan E = Tidak Lulus).

 Langkah Kerja

1. Buka Programmiz C++ Compiler
2. Buat program menentukan grade berdasarkan input nilai (0–100)
3. Jalankan dengan input: 95, 83, 72, 65, 45
4. Screenshot tampilan kode DAN output-nya
5. Ulangi langkah 1–4 menggunakan Programmiz Python Compiler

Screenshot Program C++	Screenshot Program Python
<i>[Tempelkan screenshot program C++ dan output di sini]</i>	<i>[Tempelkan screenshot program Python dan output di sini]</i>

Kegiatan 2 Tantangan Numerasi: Konverter Suhu

Konteks Numerasi: Buat program konverter suhu yang meminta input angka suhu dan jenis konversi (C=Celsius ke Fahrenheit, F=Fahrenheit ke Celsius, K=Kelvin ke Celsius). Rumus: $^{\circ}\text{F} = (^{\circ}\text{C} \times 9/5) + 32$ | $^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 5/9$ | $^{\circ}\text{C} = \text{K} - 273.15$

Screenshot Konverter Suhu C++	Screenshot Konverter Suhu Python
<i>[Tempelkan screenshot C++ di sini]</i>	<i>[Tempelkan screenshot Python di sini]</i>

Pertanyaan Analisis (C4–C6)

Soal Literasi & Numerasi Kognitif Tinggi

1. [C4-Analisis] Bandingkan sintaks if-else if pada C++ dengan if-elif pada Python. Apa perbedaan mendasar dan mana yang lebih mudah dibaca? Jelaskan dengan contoh dari program yang kamu buat!
2. [C5-Evaluasi] Seorang siswa mendapat nilai 75. Teman kamu menulis kondisi: `if(nilai>70) grade=C;` `else if(nilai>80) grade=B.` Apakah program ini benar? Evaluasi dan perbaiki jika ada kesalahan!
3. [C6-Kreasi] Rancang program kalkulator tarif parkir: Motor Rp 2.000/jam pertama + Rp 1.000/jam berikutnya; Mobil Rp 4.000/jam pertama + Rp 2.000/jam berikutnya; Truk Rp 6.000/jam pertama + Rp 3.000/jam berikutnya. Implementasikan dalam C++ DAN Python, jalankan di Programmiz, dan screenshot hasilnya!

Jawaban No. 1:

Jawaban No. 2:

No. 3: [Tempelkan screenshot program kalkulator parkir C++ dan Python di sini]

Refleksi Diri (Assessment AS Learning)

Pernyataan Refleksi	Sudah Paham	Cukup Paham	Belum Paham
Saya memahami konsep percabangan if-elseif-else	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saya dapat menulis program kondisional dalam C++	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saya dapat menulis program kondisional dalam Python	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saya dapat membandingkan sintaks C++ dan Python	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Saya dapat mengidentifikasi dan memperbaiki bug kondisional	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Bug yang saya temui dan cara mengatasinya:

PERTEMUAN 2 NESTED IF (KONDISIONAL BERSARANG)

📄 Informasi Pertemuan

Alokasi Waktu: 2 × 45 menit | Metode: PjBL KELOMPOK (3–4 orang)

Materi: Nested if / kondisional bersarang dalam C++ dan Python

Platform: Programmiz Online Compiler

BAHAN AJAR

2.1 Konsep Nested If

Nested if (kondisional bersarang) adalah penempatan pernyataan if di dalam blok if atau else yang lain. Digunakan ketika keputusan kedua bergantung pada hasil keputusan pertama membentuk hierarki logika berlapis.

Kapan Menggunakan Nested If?

- ✓ Ketika kondisi kedua hanya relevan jika kondisi pertama terpenuhi
- ✓ Validasi berlapis (contoh: cek usia LALU cek kewarganegaraan untuk hak pilih)
- ✓ Klasifikasi multi-dimensi (contoh: cek nilai LALU cek kehadiran untuk kelulusan)
- ✗ HINDARI: Lebih dari 3 tingkat nesting gunakan fungsi atau && untuk kondisi sederhana

2.2 Sintaks Nested If C++ vs Python

C++ Nested If (Tiket Bioskop)

```
#include <iostream>
using namespace std;
int main() {
    int usia; bool punya_tiket;
    char jawab;
    cout << "Usia: "; cin >> usia;
    cout << "Punya tiket? (y/n): ";
    cin >> jawab;
    punya_tiket = (jawab == 'y');
    if (usia >= 17) {
        if (punya_tiket) {
            cout << "Silakan masuk!" << endl;
        } else {
            cout << "Beli tiket dulu!" << endl;
        }
    } else {
        cout << "Anda di bawah umur!" << endl;
    }
    return 0;
}
```

Python Nested If (Tiket Bioskop)

```
usia = int(input("Usia: "))
jawab = input("Punya tiket? (y/n): ")
punya_tiket = (jawab == 'y')

if usia >= 17:
    if punya_tiket:
        print("Silakan masuk!")
    else:
        print("Beli tiket dulu!")
else:
    print("Anda di bawah umur!")
```

2.3 Tracing Manual Analisis Alur

Usia Input	Punya Tiket	Kondisi 1 (>=17)	Output
25	y (true)	TRUE → masuk blok	Silakan masuk!
25	n (false)	TRUE → masuk blok	Beli tiket dulu!

Usia Input	Punya Tiket	Kondisi 1 (≥ 17)	Output
15	y (true)	FALSE \rightarrow else	Anda di bawah umur!
15	n (false)	FALSE \rightarrow else	Anda di bawah umur!

JOBSHEET 2 LEMBAR KERJA PESERTA DIDIK (KELOMPOK)**👥 Ketentuan Kerja Kelompok**

- Kerjakan dalam kelompok 3–4 orang
- Bagi tugas: 1 orang sebagai Driver (yang mengetik kode), 1 orang sebagai Navigator (yang mengarahkan), 1 orang sebagai Dokumenter (screenshot dan laporan), 1 orang sebagai Presenter
- Setiap anggota kelompok harus memahami semua bagian kode
- Lakukan peer assessment (code review) terhadap kelompok lain di akhir sesi

IDENTITAS KELOMPOK

Nama Kelompok: _____	Kelas: _____
Anggota: 1. _____ 2. _____ 3. _____ 4. _____	

Kegiatan 1 Proyek: Sistem Penerimaan Beasiswa

Konteks Literasi Nyata: Yayasan ABC memberikan beasiswa dengan kriteria: (1) Nilai rata-rata minimal 80. (2) Jika nilai ≥ 80 dan ≥ 85 , beasiswa penuh. (3) Jika nilai ≥ 80 tetapi < 85 , cek penghasilan orang tua: jika $< \text{Rp } 3.000.000 \rightarrow$ beasiswa parsial 75%; jika $\geq \text{Rp } 3.000.000 \rightarrow$ beasiswa parsial 50%. (4) Jika nilai $< 80 \rightarrow$ tidak lolos.

🔗 Langkah Kerja Kelompok

1. Diskusikan alur logika menggunakan flowchart di kertas/whiteboard
2. Implementasikan dalam C++ di Programmiz Screenshot kode + output
3. Implementasikan dalam Python di Programmiz Screenshot kode + output
4. Uji dengan minimal 4 test case berbeda
5. Dokumentasikan dalam tabel test case

No.	Nilai	Penghasilan OT	Expected Output	Actual Output
1	87	Rp 2.500.000	Beasiswa Penuh	
2	82	Rp 2.800.000	Beasiswa 75%	
3	82	Rp 3.500.000	Beasiswa 50%	
4	78	-	Tidak Lolos	

Screenshot Program C++	Screenshot Program Python
[Screenshot C++]	[Screenshot Python]

Pertanyaan Analisis Kelompok (C4–C6)

Soal Diskusi Kelompok

- [C4-Analisis] Gambarkan flowchart program beasiswa yang kalian buat! Identifikasi semua titik percabangan (decision node) dan hubungkannya dengan kode nested if yang ditulis.
- [C5-Evaluasi] Apakah program beasiswa kalian bisa disederhanakan menggunakan if-elif (tanpa nested)? Kapan nested if lebih tepat digunakan dibanding if-elif? Berikan justifikasi!
- [C6-Kreasi] Modifikasi program beasiswa dengan menambahkan fitur: jika nilai ≥ 90 DAN penghasilan orang tua $< \text{Rp } 2.000.000$, tambahkan tunjangan hidup Rp 500.000/bulan. Implementasikan dan screenshot hasilnya!

Jawaban Diskusi Kelompok:

Flowchart (No.1) & Screenshot Modifikasi (No.3) di sini:

Peer Assessment Code Review Antar Kelompok

Kelompok yang Direview: _____

Nama Reviewer: _____

Aspek Penilaian	Skor (1-4)	Komentar/Saran
Logika nested if sudah benar		
Program berhasil dijalankan tanpa error		
Kode mudah dibaca (indentasi/nama variabel)		
Diimplementasi dalam C++ dan Python		
Test case lengkap dan tepat		

PERTEMUAN 3 STRUKTUR KONTROL PERULANGAN (FOR & WHILE)

Informasi Pertemuan

Alokasi Waktu: 2 × 45 menit | Metode: PjBL Individual

Materi: For loop dan While loop dalam C++ dan Python

Platform: Programmiz Online Compiler

BAHAN AJAR

3.1 Konsep Perulangan

Perulangan (loop) memungkinkan program mengeksekusi blok kode berulang kali. Ada dua jenis utama: FOR LOOP digunakan ketika jumlah iterasi diketahui; WHILE LOOP digunakan ketika pengulangan bergantung pada kondisi.

3.2 FOR Loop C++ vs Python

C++ For Loop

```
// Mencetak 1 sampai 10
for (int i = 1; i <= 10; i++) {
    cout << i << " ";
}

// Deret bilangan genap
for (int i = 2; i <= 20; i += 2) {
    cout << i << " ";
}

// Pola bintang segitiga
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= i; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

Python For Loop

```
# Mencetak 1 sampai 10
for i in range(1, 11):
    print(i, end=' ')

# Deret bilangan genap
for i in range(2, 21, 2):
    print(i, end=' ')

# Pola bintang segitiga
for i in range(1, 6):
    for j in range(i):
        print('*', end='')
    print()
```

3.3 WHILE Loop C++ vs Python

C++ While Loop

```
// Menghitung faktorial
int n, hasil = 1;
cout << "Masukkan n: ";
cin >> n;
int i = 1;
while (i <= n) {
    hasil *= i;
    i++;
}
cout << "Faktorial: " << hasil;

// Validasi input
int angka;
do {
    cout << "Input (1-10): ";
    cin >> angka;
} while (angka < 1 || angka > 10);
```

Python While Loop

```
# Menghitung faktorial
n = int(input("Masukkan n: "))
hasil = 1
i = 1
while i <= n:
    hasil *= i
    i += 1
print(f'Faktorial: {hasil}')

# Validasi input
angka = 0
while angka < 1 or angka > 10:
    angka = int(input("Input (1-10): "))
    if angka < 1 or angka > 10:
        print("Input tidak valid!")
```

3.4 Perbandingan For vs While

Aspek	FOR Loop	WHILE Loop
Penggunaan ideal	Jumlah iterasi diketahui	Kondisi pengulangan dinamis
Counter	Otomatis (bagian dari sintaks)	Manual (harus diupdate di dalam loop)
Risiko infinite loop	Rendah	Tinggi (jika kondisi tidak berubah)
Contoh kasus	Mencetak tabel perkalian, iterasi array	Input validasi, menu interaktif

JOBSHEET 3 LEMBAR KERJA PESERTA DIDIK** Petunjuk**

Kerjakan INDIVIDUAL. Setiap kegiatan: tulis program di Programmiz C++ DAN Python, jalankan, screenshot, tempelkan di JOBSHEET.

IDENTITAS PESERTA DIDIK

Nama: _____	Kelas/No: _____
-------------	-----------------

Kegiatan 1 Tabel Perkalian Digital

Konteks Numerasi: Buat program yang meminta input angka n (1–12) dan menampilkan tabel perkalian n dari 1 hingga 12 dalam format rapi. Contoh output untuk n=3: $3 \times 1 = 3$ | $3 \times 2 = 6$ | ... | $3 \times 12 = 36$.

Screenshot C++ Tabel Perkalian	Screenshot Python Tabel Perkalian
<i>[Screenshot C++]</i>	<i>[Screenshot Python]</i>

Kegiatan 2 Pola Bintang Kreatif

Buat program menggunakan nested for loop yang menampilkan pola berikut (input: tinggi = 5): Segitiga bintang kanan, Segitiga bintang terbalik, dan Berlian (diamond) bintang.

Screenshot C++ Pola Bintang	Screenshot Python Pola Bintang
<i>[Screenshot C++]</i>	<i>[Screenshot Python]</i>

Pertanyaan Analisis (C4–C6)**Soal Numerasi & Literasi Tingkat Tinggi**

1. [C4-Analisis] Perhatikan dua kode berikut manakah yang lebih efisien untuk mencetak bilangan 1–100? Jelaskan mengapa dengan menganalisis jumlah operasi yang dilakukan.

Kode A: `for (int i=1; i<=100; i++) cout<<i;`

Kode B: `int i=1; while(i<=100) {cout<<i; i++;}`

2. [C5-Evaluasi] Seorang programmer menulis: `while(true) { cin>>input; if(input==0) break; }`. Apakah pendekatan ini baik atau buruk? Kapan penggunaan `while(true)` bisa dibenarkan? Berikan alternatif yang lebih baik!

3. [C6-Kreasi] Buat program kalkulator deret bilangan Fibonacci sampai N suku (input dari pengguna). Program harus: menggunakan while loop, menampilkan setiap suku, menghitung jumlah total seluruh suku. Implementasi dalam C++ DAN Python, screenshot hasilnya!

Jawaban No. 1 & 2:

No. 3: [Screenshot Fibonacci C++ dan Python di sini]

PERTEMUAN 4 LOOPING LANJUTAN: BREAK DAN CONTINUE

Informasi Pertemuan

Alokasi Waktu: 2 × 45 menit | Metode: PjBL KELOMPOK (3–4 orang)

Materi: Pernyataan break dan continue dalam loop C++ dan Python

Platform: Programmiz Online Compiler

BAHAN AJAR

4.1 Pernyataan Break

Break menghentikan eksekusi loop secara langsung, meskipun kondisi loop masih terpenuhi. Eksekusi berlanjut ke pernyataan setelah blok loop.

4.2 Pernyataan Continue

Continue melewati sisa iterasi saat ini dan langsung ke iterasi berikutnya. Loop tidak berhenti hanya satu iterasi yang dilewati.

C++ Break dan Continue

```
// Break: cari bilangan prima pertama
for (int i = 2; i <= 100; i++) {
    bool prima = true;
    for (int j = 2; j < i; j++) {
        if (i % j == 0) {
            prima = false;
            break; // keluar loop dalam
        }
    }
    if (prima) { cout << i; break; }
}

// Continue: cetak ganjil saja
for (int i = 1; i <= 10; i++) {
    if (i % 2 == 0) continue;
    cout << i << " ";
}
```

Python Break dan Continue

```
# Break: cari bilangan prima pertama
for i in range(2, 101):
    prima = True
    for j in range(2, i):
        if i % j == 0:
            prima = False
            break # keluar loop dalam
    if prima:
        print(i)
        break # keluar loop luar

# Continue: cetak ganjil saja
for i in range(1, 11):
    if i % 2 == 0:
        continue
    print(i, end='')
```

4.3 Perbedaan Break vs Continue

Aspek	BREAK	CONTINUE
Efek	Menghentikan loop sepenuhnya	Melompati iterasi saat ini saja
Loop berlanjut?	TIDAK eksekusi keluar dari loop	YA lanjut ke iterasi berikutnya
Contoh kasus	Pencarian (berhenti jika ditemukan), validasi	Filter data (lewati elemen yang tidak sesuai)
Nested loop	Hanya menghentikan loop paling dalam	Hanya melewati iterasi loop paling dalam

JOBSHEET 4 LEMBAR KERJA PESERTA DIDIK (KELOMPOK)**IDENTITAS KELOMPOK**

Nama Kelompok: _____	Kelas: _____
Anggota: 1. _____ 2. _____ 3. _____ 4. _____	

Kegiatan 1 Proyek: Mesin Kasir Sederhana dengan Loop

Konteks Literasi Nyata: Buat program kasir yang: (1) Terus meminta input barang dan harga hingga pengguna memasukkan 'selesai'. (2) Menggunakan continue jika harga yang dimasukkan negatif (harga tidak valid, minta input lagi). (3) Menggunakan break ketika pengguna mengetik 'selesai'. (4) Menampilkan total belanja di akhir.

Screenshot C++ Mesin Kasir	Screenshot Python Mesin Kasir
[Screenshot C++]	[Screenshot Python]

Kegiatan 2 Tracing Manual

Lacak output program berikut TANPA menjalankannya terlebih dahulu (tracing manual), lalu verifikasi dengan Programiz:

```
for i in range(1, 8):
    if i == 3:
        continue
    if i == 6:
        break
    print(i, end='')
```

Iterasi (i)	Kondisi i==3?	Kondisi i==6?	Aksi / Output
i = 1			
i = 2			
i = 3			
i = 4			
i = 5			

Iterasi (i)	Kondisi i==3?	Kondisi i==6?	Aksi / Output
i = 6			
Output final:	_____		

Pertanyaan Analisis Kelompok (C4–C6)

Soal Diskusi

1. [C4-Analisis] Analisis perbedaan output program ini jika 'continue' diganti 'break' dan sebaliknya. Prediksi output SEBELUM menjalankan, lalu verifikasi!
2. [C5-Evaluasi] Kapan lebih baik menggunakan break dibanding mengatur kondisi loop? Berikan 2 contoh skenario nyata untuk masing-masing!
3. [C6-Kreasi] Buat program 'Tebak Angka': komputer memiliki angka rahasia 1–100, user menebak, program memberi petunjuk 'Terlalu besar/kecil/Benar!', dan menggunakan break ketika tebakan benar. Batasi maks 7 percobaan dengan continue jika input tidak valid. Implementasi C++ DAN Python, screenshot!

Jawaban Analisis:

[Screenshot Tebak Angka C++ dan Python di sini]

PERTEMUAN 5 FUNGSI DAN MODULARITAS PROGRAM

Informasi Pertemuan

Alokasi Waktu: 2 × 45 menit | Metode: PjBL Individual

Materi: Fungsi, parameter, return value, dan debugging dasar dalam C++ dan Python

Platform: Programmiz Online Compiler

BAHAN AJAR

5.1 Konsep Fungsi dan Modularitas

Fungsi adalah blok kode yang diberi nama dan dapat dipanggil berulang kali. Modularitas berarti memecah program besar menjadi bagian-bagian kecil (fungsi) yang mudah dipahami, diuji, dan digunakan kembali.

Manfaat Fungsi (Prinsip DRY Don't Repeat Yourself)

- ✓ Reusability: Kode yang sama tidak perlu ditulis ulang
- ✓ Readability: Program lebih mudah dibaca dan dipahami
- ✓ Debugging: Lebih mudah menemukan dan memperbaiki bug
- ✓ Maintenance: Perubahan cukup dilakukan di satu tempat
- ✓ Testing: Setiap fungsi dapat diuji secara terpisah (unit testing)

5.2 Sintaks Fungsi C++ vs Python

C++ Fungsi dengan Parameter & Return

```
// Fungsi menghitung luas lingkaran
#include <iostream>
#include <cmath>
using namespace std;

double luasLingkaran(double r) {
    return M_PI * r * r;
}

// Fungsi tanpa return (void)
void sapa(string nama) {
    cout << "Halo, " << nama << "!" << endl;
}

int main() {
    double radius = 7.0;
    cout << "Luas: " << luasLingkaran(radius) << endl;
    sapa("Budi");
    return 0;
}
```

Python Fungsi dengan Parameter & Return

```
import math

# Fungsi menghitung luas lingkaran
def luas_lingkaran(r):
    return math.pi * r ** 2

# Fungsi tanpa return
def sapa(nama):
    print(f'Halo, {nama}!')

# Program utama
radius = 7.0
print(f'Luas: {luas_lingkaran(radius):.2f}')
sapa('Budi')
```

5.3 Debugging Dasar

Jenis Bug	Deskripsi	Cara Mengatasi
Syntax Error	Kesalahan penulisan kode (typo, missing semicolon)	Perhatikan pesan error, cek baris yang ditunjuk compiler
Logic Error	Program berjalan tapi output salah	Gunakan print/cout untuk melacak nilai variabel (debug print)

Jenis Bug	Deskripsi	Cara Mengatasi
Runtime Error	Program crash saat dijalankan (division by zero, dll)	Tambahkan validasi input dan pengecekan kondisi batas
Semantic Error	Kode valid tapi tidak sesuai maksud programmer	Tinjau ulang logika algoritma dengan tracing manual

JOBSHEET 5 LEMBAR KERJA PESERTA DIDIK**IDENTITAS PESERTA DIDIK**

Nama: _____	Kelas/No: _____
-------------	-----------------

Kegiatan 1 Library Fungsi Matematika

Buat program yang berisi minimal 5 fungsi matematika berikut dan tampilkan menu pilihan operasi: (1) Luas dan keliling lingkaran (parameter: radius), (2) Luas dan keliling persegi panjang (parameter: panjang, lebar), (3) Kalkulator pangkat (parameter: basis, eksponen), (4) Pengecekan bilangan prima (parameter: angka, return: true/false), (5) Konverter satuan panjang (km/m/cm).

Screenshot C++ Library Fungsi	Screenshot Python Library Fungsi
<i>[Screenshot C++]</i>	<i>[Screenshot Python]</i>

Kegiatan 2 Debugging Challenge

Temukan dan perbaiki bug dalam program Python berikut (ada 4 bug!):

```
def hitung_nilai_akhir(uts, uas, tugas):  
    nilai = uts * 0.3 + uas * 0.4 + tugas * 0.2 # Bug 1  
    return nilai
```

```
def kategori_nilai(nilai):  
    if nilai >= 90:  
        return 'A'  
    elif nilai >= 80  
        return 'B'  
    elif nilai >= 70:  
        return 'c'  
    else:  
        return 'D'
```

```
siswa = ['Andi', 'Budi', 'Cici']  
nilai_uts = [85, 70, 92]  
nilai_uas = [80, 75, 88]  
nilai_tugas = [90, 85, 95]
```

```
for i in range(4): # Bug 4: should be range(3)
```

```
na = hitung_nilai_akhir(nilai_uts[i], nilai_uas[i], nilai_tugas[i])
print(f'{{siswa[i]}}: {{na:.1f}} ({{kategori_nilai(na)}}')
```

Bug #	Lokasi (Baris)	Deskripsi Bug	Perbaikan
1			
2			
3			
4			

[Screenshot program setelah diperbaiki dan berhasil dijalankan]

Pertanyaan Analisis (C4–C6)

Soal Tingkat Tinggi

- [C4-Analisis] Apa perbedaan mendasar antara fungsi dengan void/return None dan fungsi dengan return value? Berikan contoh kasus penggunaan masing-masing dalam konteks nyata!
- [C5-Evaluasi] Seorang programmer menulis semua kode dalam fungsi main() tanpa memisahkan ke fungsi-fungsi terpisah. Evaluasi praktik ini apa kelemahan dan bagaimana seharusnya kode diorganisasikan? Gunakan prinsip modularitas!
- [C6-Kreasi] Buat program kalkulator nilai rapor menggunakan MINIMAL 4 fungsi terpisah: input_nilai(), hitung_rata_rata(), tentukan_predikat(), tampilkan_laporan(). Program harus memproses minimal 5 mata pelajaran. Implementasi di C++ DAN Python, screenshot!

Jawaban No. 1 & 2:

[Screenshot Kalkulator Rapor C++ dan Python]

PERTEMUAN 6 PROYEK PEMROGRAMAN SEDERHANA & DEBUGGING

Informasi Pertemuan

Alokasi Waktu: 2 × 45 menit | Metode: PjBL KELOMPOK (3–4 orang)

Kegiatan: Proyek akhir, debugging kolaboratif, dan presentasi

Platform: Programmiz Online Compiler | Output: Portofolio Digital

DESKRIPSI PROYEK AKHIR

Tema Proyek: Membuat Aplikasi Pemrograman Sederhana

Setiap kelompok memilih SATU dari tema proyek berikut dan mengimplementasikannya LENGKAP dalam C++ DAN Python:

TEMA A: Sistem Manajemen Data Siswa

- Input dan tampilkan data minimal 5 siswa (nama, nilai 3 mapel)
- Hitung rata-rata, nilai tertinggi/terendah
- Tampilkan ranking menggunakan loop + kondisional
- Fungsi terpisah: input_data(), hitung_statistik(), tampilkan_ranking()

TEMA B: Kalkulator Keuangan Pribadi

- Input pemasukan dan pengeluaran bulanan
- Kategorikan pengeluaran (makan, transportasi, hiburan, lainnya)
- Hitung sisa anggaran dan persentase tiap kategori
- Rekomendasi hemat berdasarkan kondisi keuangan

TEMA C: Simulator Ujian / Kuis Interaktif

- Minimal 10 soal pilihan ganda yang tersimpan dalam program
- Hitung skor, tampilkan jawaban benar/salah
- Gunakan loop untuk iterasi soal dan kondisional untuk penilaian
- Fungsi: tampilkan_soal(), periksa_jawaban(), hitung_skor()

JOBSHEET 6 LEMBAR KERJA PROYEK (KELOMPOK)

IDENTITAS KELOMPOK

Nama Kelompok: _____	Tema Proyek: A / B / C (lingkari)
Anggota: 1. _____ 2. _____ 3. _____ 4. _____	

Tahap 1 Perencanaan Proyek (15 menit)

Judul Program Kami: _____

Deskripsi singkat apa yang dilakukan program:

Konsep yang akan digunakan (centang):

If-elseif-else Nested if For loop While loop Break/Continue Fungsi Parameter & Return

Pembagian Tugas Kelompok

Nama Anggota	Peran	Bagian Kode yang Dikerjakan
	Driver / Coder	
	Navigator / Debugger	
	Dokumenter / Screenshot	
	Presenter / Reporter	

Tahap 2 Implementasi & Screenshot

Screenshot Program C++	Screenshot Program Python
<i>[Screenshot kode C++ lengkap + output]</i>	<i>[Screenshot kode Python lengkap + output]</i>

Tahap 3 Debugging Log

Dokumentasikan minimal 3 bug yang ditemui selama pengerjaan proyek:

#	Jenis Bug	Deskripsi Error	Penyebab	Solusi
1				
2				
3				

Tahap 4 Presentasi (10 menit per kelompok)

Panduan Presentasi

1. Demo program berjalan (C++ DAN Python) 3 menit
2. Jelaskan logika dan struktur kode yang digunakan 3 menit
3. Ceritakan tantangan/bug terbesar dan cara mengatasinya 2 menit
4. Tanya jawab dari kelompok lain/guru 2 menit

Pertanyaan Evaluasi Proyek (C4–C6)**Refleksi Proyek**

1. [C4-Analisis] Identifikasi bagian kode mana yang paling sering dipanggil ulang (reused) dalam program kalian. Apakah bagian itu sudah dijadikan fungsi? Jika belum, refactor dan screenshot hasilnya!
2. [C5-Evaluasi] Bandingkan implementasi C++ dan Python kalian. Mana yang lebih mudah ditulis? Mana yang output-nya lebih rapi? Berikan justifikasi berdasarkan pengalaman langsung!
3. [C6-Kreasi] Jika diberi waktu tambahan, fitur apa yang akan kalian tambahkan ke proyek? Buat pseudocode atau flowchart untuk fitur tersebut dan jelaskan bagaimana cara mengimplementasikannya!

Jawaban Evaluasi Proyek:

INSTRUMEN ASESMEN

A. ASSESSMENT AS LEARNING (Jurnal Refleksi)

 Jurnal Refleksi Harian Diisi Setelah Setiap Pertemuan

Tujuan: Membantu peserta didik memantau perkembangan pemahaman diri sendiri

Frekuensi: Setiap akhir pertemuan (6 kali total)

Format: Digital (Google Forms/Classroom) atau kertas

JURNAL REFLEKSI Nama: _____ | Pertemuan ke: ____ | Tanggal: _____

1. Konsep yang paling saya pahami hari ini:

2. Konsep yang masih membingungkan:

3. Bug yang saya temui dan cara mengatasinya:

4. Perbedaan C++ vs Python yang saya temukan:

5. Target saya untuk pertemuan berikutnya:

B. ASSESSMENT FOR LEARNING KUIS SINGKAT

 Kuis Formatif Diberikan di Awal / Pertengahan Pertemuan

Waktu: 10–15 menit | Format: Google Forms atau Lembar Terpisah

Tujuan: Mengukur pemahaman awal dan umpan balik untuk penyesuaian pembelajaran

Kuis Pertemuan 1 & 2 Kondisional

No.	Pertanyaan / Soal
1.	Apa output kode berikut? (C++): <code>int x=15; if(x>10){if(x>20){cout<<"A";}else{cout<<"B";}else{cout<<"C";}</code> → Jawab: _____
2.	Seorang siswa lulus jika nilai ≥ 75 DAN hadir $\geq 80\%$. Tulis kondisi Python (if/elif/else) yang tepat!
3.	Manakah yang BENAR tentang nested if: (a) Selalu lebih efisien dari if-elif (b) Kondisi dalam hanya diperiksa jika kondisi luar terpenuhi (c) Tidak bisa digunakan dalam Python
4.	Perbaiki kode C++ berikut: <code>if(nilai=90){cout<<"A";} elseif(nilai=80){cout<<"B";}</code> Ada 3 kesalahan, sebutkan!

Kuis Pertemuan 3 & 4 Perulangan

No.	Pertanyaan / Soal
1.	Tulis output program Python: <code>[for i in range(1,6): if i%2!=0: print(i, end=' ')]</code> → Jawab: _____

No.	Pertanyaan / Soal
2.	Tuliskan program C++ menggunakan while loop untuk menampilkan bilangan yang habis dibagi 3 antara 1-50!
3.	Apa perbedaan break dan continue? Berikan 1 contoh penggunaan masing-masing dalam konteks nyata!
4.	Seorang programmer menulis 'for i in range(10): pass'. Apa efek 'pass'? Kapan digunakan?

Kuis Pertemuan 5 Fungsi

No.	Pertanyaan / Soal
1.	Apa output program berikut? <code>def tambah(a,b=10): return a+b; print(tambah(5)); print(tambah(3,7))</code> → Jawab: _____
2.	Jelaskan konsep 'scope variabel' dalam fungsi apa perbedaan variabel lokal dan global? Berikan contoh!
3.	Mengapa fungsi penting dalam pemrograman? Sebutkan 3 manfaat utama dan berikan contoh konkret!

C. ASSESSMENT OF LEARNING TES TERTULIS

Tes Sumatif Akhir Unit

Waktu: 90 menit | Format: Tertulis + Coding di Programmiz

Bobot: Bagian A (30%) + Bagian B (40%) + Bagian C (30%)

Bagian A Analisis Kode (30 poin)

Petunjuk: Jawab SEMUA pertanyaan dengan tepat dan lengkap.

No.	Soal	Poin
A1.	[C4] Trace program C++ berikut dan tulis outputnya langkah demi langkah: <code>for(int i=1; i<=5; i++) { for(int j=1; j<=i; j++) { if(j%2==0) continue; cout<<"*"; } cout<<endl; }</code>	8
A2.	[C5] Program Python berikut mengandung logic error. Identifikasi masalahnya dan jelaskan perbaikan: <code>def cek_segitiga(a,b,c): if a+b>c or a+c>b or b+c>a: return 'Segitiga' else: return 'Bukan Segitiga'</code>	8
A3.	[C4] Gambarkan flowchart untuk program yang: membaca 10 bilangan, menampilkan hanya bilangan positif, menghentikan input jika ditemukan angka 0, dan menampilkan rata-rata bilangan positif.	8
A4.	[C5] Bandingkan for loop dan while loop: kapan masing-masing lebih tepat digunakan? Berikan 2 contoh kasus nyata untuk masing-masing!	6

Bagian B Coding di Programmiz (40 poin)

Petunjuk: Kerjakan di Programmiz, jalankan program, screenshot kode + output, dan tempelkan di lembar ini.

No.	Soal Coding	Poin
B1.	[C5] Buat program Python LENGKAP (dengan fungsi) yang membaca data 5 produk (nama, harga, stok), menampilkan produk dengan stok < 10 sebagai 'Perlu Restock', menghitung total nilai inventori (harga × stok), dan menampilkan produk termahal. Gunakan minimal 3 fungsi terpisah.	20
B2.	[C6] Buat program C++ yang mengimplementasikan game 'FizzBuzz Lanjutan': cetak angka 1–100, jika habis dibagi 3 cetak 'Fizz', habis dibagi 5 cetak 'Buzz', habis dibagi 7 cetak 'Whiz', habis dibagi 15 cetak 'FizzBuzz', kombinasi lainnya gabungkan (contoh: habis 3 dan 7 = 'FizzWhiz'). Hitung berapa kali masing-masing kata muncul.	20

Screenshot B1 (Python)	Screenshot B2 (C++)
<i>[Screenshot B1]</i>	<i>[Screenshot B2]</i>

Bagian C Debugging Challenge (30 poin)

[C6] Program C++ berikut mengandung 5 bug. Temukan semua bug, jelaskan masalahnya, perbaiki kode, jalankan di Programmiz, dan screenshot hasilnya:

```
#include <iostream>
using namespace std;

int hitungDiskon(int harga, int persen) {
    int diskon = harga * persen / 100
    return diskon;
}

void tampilkanBelanja(int harga[], int n) {
    int total = 0;
    for (int i = 0; i <= n; i++) {
        total += harga[i];
    }
    cout << "Total: " << total << endl;
}

int main() {
    int harga[] = {50000, 75000, 30000, 120000};
    int n = 4;
    tampilkanBelanja(harga, n);

    int diskon = hitungDiskon(75000, 20);
    cout << "Diskon: " << diskon;
    cout << "Harga akhir: " << 75000 + diskon << endl;

    // Cari barang termahal
    int maks = harga[0];
    for (int i = 1; i < n; i++) {
        if (harga[i] = maks) {
            maks = harga[i];
        }
    }
    cout << "Termahal: " << maks << endl;
    return 1;
}
```

Bug #	Lokasi (Baris ~)	Deskripsi & Penyebab	Kode yang Diperbaiki
1			
2			
3			
4			
5			

[Screenshot kode C++ yang sudah diperbaiki dan output yang benar]

RUBRIK PENILAIAN**1. RUBRIK PENILAIAN JOBSHEET INDIVIDUAL (Pertemuan 1, 3, 5)**

Komponen	Sangat Baik (4)	Baik (3)	Cukup (2)	Perlu Perbaikan (1)
1. Kelengkapan Screenshot (25%)	Screenshot C++ dan Python, kode + output lengkap dan jelas	Screenshot ada, kode atau output kurang jelas	Hanya satu bahasa disertai screenshot	Screenshot tidak ada atau tidak relevan
2. Kebenaran Program / Fungsionalitas (35%)	Program berjalan sempurna, output benar untuk semua test case	Program berjalan, 1–2 output tidak sesuai	Program berjalan sebagian, beberapa error	Program tidak bisa dijalankan / error fatal
3. Kualitas Kode (20%)	Indentasi rapi, nama variabel deskriptif, ada komentar	Indentasi rapi, nama variabel cukup jelas	Indentasi kurang konsisten, nama variabel kurang jelas	Kode sulit dibaca, tidak ada indentasi
4. Jawaban Analisis C4–C6 (20%)	Jawaban lengkap, tepat, analisis mendalam dan orisinal	Jawaban tepat tapi kurang mendalam	Jawaban sebagian benar, kurang elaborasi	Jawaban tidak relevan atau kosong

Skor Akhir JOBSHEET Individual = $(\text{Komponen 1} \times 25\% + \text{Komponen 2} \times 35\% + \text{Komponen 3} \times 20\% + \text{Komponen 4} \times 20\%) / 4 \times 100$

2. RUBRIK PENILAIAN JOBSHEET KELOMPOK (Pertemuan 2, 4, 6)

Komponen	Sangat Baik (4)	Baik (3)	Cukup (2)	Perlu Perbaikan (1)
1. Fungsionalitas Program (30%)	Program C++ dan Python berjalan sempurna sesuai spesifikasi	Berjalan dengan sedikit kekurangan fitur	Sebagian fitur berjalan, beberapa error	Program tidak dapat dijalankan
2. Penerapan Konsep (25%)	Semua konsep diterapkan tepat dan efisien	Sebagian besar konsep diterapkan dengan benar	Beberapa konsep diterapkan, ada yang keliru	Konsep tidak diterapkan atau salah semua
3. Kolaborasi & Pembagian Tugas (20%)	Tugas terbagi merata, semua berkontribusi aktif	Sebagian besar anggota berkontribusi	Hanya beberapa yang aktif berkontribusi	Tidak ada pembagian tugas yang jelas
4. Dokumentasi (Screenshot) (15%)	Screenshot lengkap, berlabel, dan informatif	Screenshot ada tapi kurang lengkap	Hanya beberapa screenshot	Tidak ada screenshot
5. Presentasi (10%)	Presentasi jelas, terstruktur, menjawab pertanyaan dengan baik	Presentasi cukup jelas, menjawab sebagian pertanyaan	Presentasi kurang terstruktur	Tidak melakukan presentasi

3. RUBRIK PENILAIAN PROYEK AKHIR (Pertemuan 6 AOL)

Komponen	Sangat Baik (4) 90–100	Baik (3) 75–89	Cukup (2) 60–74	Perlu Perbaikan (1) <60
A. Fungsionalitas Program (25%)	Semua fitur berjalan sempurna tanpa error di C++ dan Python	Sebagian besar fitur berjalan, 1–2 minor bug	Program berjalan dengan beberapa fitur yang tidak bekerja	Program tidak bisa dijalankan / error fatal
B. Penerapan Konsep (25%)	Kondisional, loop, fungsi digunakan secara tepat dan efisien dalam kedua bahasa	Konsep diterapkan benar tapi belum optimal	Beberapa konsep diterapkan, ada yang kurang tepat	Konsep tidak diterapkan atau salah penerapan
C. Kualitas Kode (20%)	Kode bersih: indentasi konsisten, nama variabel deskriptif, ada komentar, DRY principle diterapkan	Kode cukup bersih dengan minor issue	Kode kurang rapi, beberapa nama variabel tidak deskriptif	Kode sulit dibaca, tidak terstruktur
D. Kreativitas & Kompleksitas (15%)	Program menunjukkan kreativitas tinggi, fitur melampaui minimum, penyelesaian masalah nyata	Ada kreativitas, sedikit di atas minimum	Memenuhi requirement minimum	Di bawah requirement minimum
E. Debugging & Dokumentasi (15%)	Log debugging lengkap (min 3 bug), solusi dijelaskan dengan baik, screenshot informatif	Log debugging ada dengan penjelasan cukup	Log debugging ada tapi kurang detail	Tidak ada dokumentasi debugging

Formula Nilai Proyek Akhir

$$\text{Nilai Proyek} = (A \times 25 + B \times 25 + C \times 20 + D \times 15 + E \times 15) / 25$$

$$\text{Konversi: } 4 = 90-100 \mid 3 = 75-89 \mid 2 = 60-74 \mid 1 = <60$$

4. RUBRIK TES TERTULIS / SUMATIF

Bagian	Deskripsi Penilaian	Kriteria Nilai Penuh	Bobot
A Analisis (A1)	Tracing program C++ menggunakan nested loop + continue poin per step	Semua 5 langkah tracing benar	8 poin

Bagian	Deskripsi Penilaian	Kriteria Nilai Penuh	Bobot
A Analisis (A2)	Identifikasi logic error dalam fungsi + penjelasan perbaikan	Identifikasi benar + solusi tepat + kode perbaikan	8 poin
A Analisis (A3)	Flowchart program lengkap: notasi benar, alur logis, semua kondisi ada	Semua elemen flowchart benar dan lengkap	8 poin
A Analisis (A4)	Perbandingan for vs while: definisi + 2 contoh kasus masing-masing	Penjelasan tepat + contoh kasus relevan	6 poin
B Coding (B1)	Program Python dengan fungsi: rubrik = fungsi (5) + fungsionalitas (8) + kode bersih (4) + screenshot (3)	Semua kriteria terpenuhi sempurna	20 poin
B Coding (B2)	Program C++ FizzBuzz Lanjutan: rubrik = logika kondisional (8) + output benar (7) + hitung kata (3) + screenshot (2)	Semua logika benar + output sempurna	20 poin
C Debugging	5 bug C++: identifikasi (10) + penjelasan (10) + kode fixed + screenshot (10)	Semua 5 bug ditemukan + diperbaiki + program berjalan	30 poin
Rentang Nilai	Predikat	Keterangan	
91–100	A Sangat Baik	Melampaui target pembelajaran	
81–90	B Baik	Memenuhi target pembelajaran	
71–80	C Cukup	Mendekati target pembelajaran	
61–70	D Perlu Perbaikan	Di bawah target, perlu remedial	
<61	E Remedi	Harus mengulang, butuh intervensi guru	

5. RUBRIK PEER ASSESSMENT (Code Review)

Aspek yang Dinilai	4 Sangat Baik	3 Baik	2 Cukup	1 Perlu Bimb.
Kebenaran logika program	Logika sempurna, tidak ada kesalahan alur	Logika benar dengan minor issue	Beberapa logika keliru	Logika salah/tidak jelas
Keterbacaan kode	Sangat mudah dibaca, nama var. deskriptif, ada komentar	Mudah dibaca, nama var. cukup jelas	Agak sulit dibaca	Sulit dibaca
Penggunaan konsep yang tepat	Konsep digunakan secara optimal dan efisien	Konsep digunakan dengan benar	Konsep digunakan tapi kurang tepat	Konsep tidak/salah digunakan
Kelengkapan implementasi C++ & Python	Keduanya lengkap dan berjalan sempurna	Keduanya ada, satu kurang sempurna	Hanya satu bahasa diimplementasi	Tidak lengkap
Kreativitas solusi	Solusi kreatif, melampaui requirement	Solusi baik, memenuhi requirement	Solusi minimal, tepat requirement	Solusi tidak memenuhi requirement

Aspek yang Dinilai

4 Sangat Baik

3 Baik

2 Cukup

1 Perlu Bimb.

Panduan Memberikan Feedback Peer Assessment

1. POSITIF: Sebutkan minimal 2 hal yang sudah bagus dari kode teman
2. KONSTRUKTIF: Berikan saran perbaikan yang spesifik dan dapat dilakukan
3. SANTUN: Gunakan bahasa yang mendukung dan tidak merendahkan
4. CONTOH: Sertakan contoh kode yang lebih baik jika memungkinkan

6. REKAP NILAI AKHIR

Komponen Penilaian	Instrumen	Bobot	Nilai
Nilai Formatif (AFL)	Kuis 1–5 + Observasi + Latihan Harian	20%	_____
Nilai JOBSHEET Individual (P1, P3, P5)	JOBSHEET + Screenshot Programmiz	20%	_____
Nilai JOBSHEET Kelompok (P2, P4)	JOBSHEET Kelompok + Peer Assessment	20%	_____
Nilai Proyek Akhir (P6)	Proyek + Presentasi + Portofolio	25%	_____
Nilai Tes Sumatif	Tes Tertulis + Coding + Debugging	15%	_____
NILAI AKHIR	Jumlah Tertimbang	100%	_____

7. PORTOFOLIO DIGITAL** Komponen Portofolio Digital Dikumpulkan di Google Classroom**

Portofolio adalah kumpulan bukti belajar peserta didik sepanjang unit ini.

WAJIB ADA:

- Screenshot program C++ dan Python dari setiap pertemuan (12 screenshot minimum)
- JOBSHEET yang sudah diisi lengkap (P1–P6)
- Jurnal Refleksi (6 entri)
- Log Debugging Proyek Akhir
- Laporan Proyek dengan kode lengkap

NILAI TAMBAH:

- Program bonus / kreasi sendiri
- Perbandingan mendalam C++ vs Python
- Video demo program berjalan
- Catatan belajar / mind map konsep