

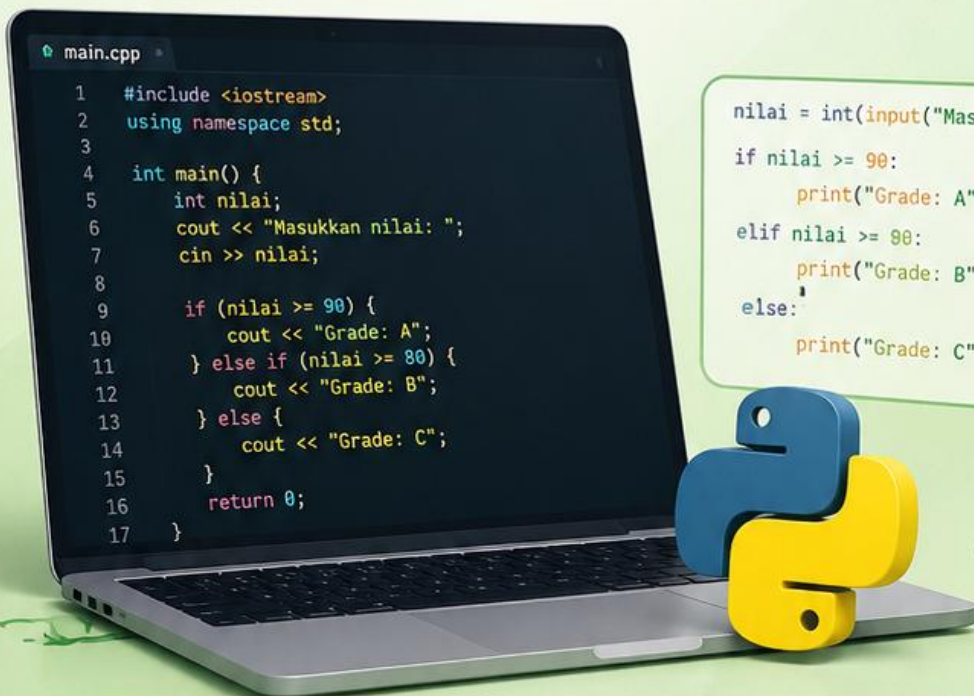
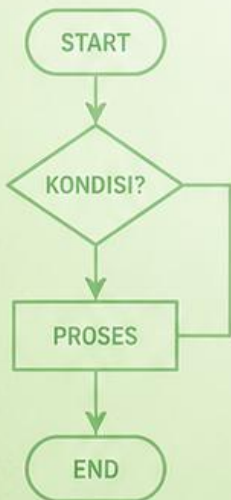


RENCANA PELAKSANAAN PEMBELAJARAN (RPP)

INFORMATIKA

SMA / MA

KELAS X | FASE E



```
nilai = int(input("Masukkan nilai: "))
if nilai >= 90:
    print("Grade: A")
elif nilai >= 80:
    print("Grade: B")
else:
    print("Grade: C")
```



Nama Penyusun:
Ikhsan Budi Maulana, S.Kom



NIM:
25427697



MODUL AJAR
FASE E KELAS X
MATA PELAJARAN INFORMATIKA

BAB 7: ALGORITMA dan PEMROGRAMAN

IDENTITAS	
Satuan Pendidikan	: SMAN 2 Padang
Nama Pendidik	: Ikhsan Budi Maulana
Mata Pelajaran	: Informatika
Fase	: E
Kelas / Semester	: X / 2 (Genap)
BAB	: 7 “Algoritma dan Pemrograman”
Materi Pokok	: Struktur Kondisional, Nested If, Perulangan
Alokasi Waktu	: 12 x 45 Menit (6 Pertemuan)

IDENTIFIKASI	
Kesiapan Peserta Didik	A. Pengetahuan Awal Peserta didik diharapkan memiliki pemahaman dasar tentang berpikir komputasional (sebagaimana yang sudah dibahas pada bab sebelumnya). Ini meliputi kemampuan dekomposisi masalah, pengenalan pola, abstraksi, dan merumuskan algoritma sederhana dalam bahasa alami. Mereka mungkin sudah pernah berinteraksi dengan aplikasi atau game yang didasarkan pada logika pemrograman. Beberapa mungkin sudah memiliki pengalaman bermain dengan blok coding atau editor teks sederhana.
	B. Minat Minat terhadap pemrograman sangat bervariasi. Pendekatan yang mengaitkan pemrograman dengan pembuatan aplikasi yang mereka gunakan sehari-hari (game sederhana, kalkulator, chatbot), atau pemecahan masalah yang relevan dengan hobi mereka, akan meningkatkan minat. Penggunaan platform Programmiz

	<p>(programmiz.com) untuk uji coba C++ dan Python online secara langsung dapat membuat pengalaman lebih menyenangkan.</p>
	<p>C. Latar Belakang</p> <p>Peserta didik berasal dari latar belakang yang beragam dalam hal akses dan paparan terhadap teknologi dan pemrograman. Beberapa mungkin sudah pernah mencoba coding secara otodidak, sementara yang lain sama sekali belum. Perlu pendekatan berdiferensiasi untuk menjembatani kesenjangan ini.</p>
	<p>D. Kebutuhan Belajar</p> <ul style="list-style-type: none"> • Visual: Membutuhkan diagram alur (flowchart), contoh kode C++ dan Python dengan highlighting sintaks, video tutorial langkah demi langkah, dan tampilan output di Programmiz. • Auditori: Membutuhkan penjelasan lisan yang jelas tentang konsep pemrograman, diskusi kelompok untuk memecahkan bug. • Kinestetik: Membutuhkan praktik langsung menulis dan menjalankan kode di Programmiz Online IDE, debugging, mencoba berbagai input.
<p>Karakteristik Materi Pembelajaran</p>	<p>1. Jenis Pengetahuan yang Akan Dicapai</p> <p>Pengetahuan konseptual (kondisional if-elseif-else, nested if, perulangan for/while, break/continue, fungsi), pengetahuan prosedural (menulis kode C++ dan Python di Programmiz, debugging, menguji program), dan pengetahuan metakognitif (merencanakan solusi sebelum coding, merefleksikan efisiensi kode).</p> <p>2. Relevansi dengan Kehidupan Nyata Peserta Didik</p> <p>Materi ini sangat relevan dan fundamental di era digital. Pemrograman menggerakkan sebagian besar teknologi yang mereka gunakan. Belajar pemrograman C++ dan Python mengembangkan kemampuan problem-solving, logika, kreativitas, dan ketelitian yang bermanfaat di berbagai bidang.</p> <p>3. Tingkat Kesulitan</p> <p>Moderat hingga tinggi. Kondisional if-elseif-else dan nested if membutuhkan logika bersarang yang cermat. Perulangan</p>

	<p>dengan break/continue, serta fungsi dan modularitas, memerlukan ketelitian dan penalaran logis yang kuat.</p>
	<p>4. Struktur Materi</p> <p>Materi diawali dari kondisional sederhana (if-elseif-else), kondisional bersarang (nested if), struktur perulangan dasar (for/while), perulangan lanjutan (break/continue), fungsi dan modularitas, hingga proyek pemrograman sederhana dengan debugging. Setiap konsep diimplementasikan dalam dua bahasa: C++ dan Python, menggunakan platform Programmiz.</p>
	<p>5. Integrasi Nilai dan Karakter</p> <ul style="list-style-type: none"> • Penalaran Kritis: Menganalisis masalah, merancang solusi logis, dan menemukan serta memperbaiki bug. • Kreativitas: Merancang program inovatif dan menemukan berbagai cara memecahkan masalah. • Kolaborasi: Bekerja sama merancang algoritma, menulis kode, dan debugging. • Kemandirian: Tekun dan sabar dalam menghadapi tantangan pemrograman. • Ketelitian: Sintaks yang benar dan logika presisi dalam dua bahasa pemrograman.
<p>Dimensi Profil Lulusan</p>	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Keimanan dan Ketaqwaan Kepada Tuhan YME <input checked="" type="checkbox"/> Penalaran Kritis <input checked="" type="checkbox"/> Kreativitas <input checked="" type="checkbox"/> Kolaborasi <input checked="" type="checkbox"/> Kemandirian <input checked="" type="checkbox"/> Kesehatan <input checked="" type="checkbox"/> Komunikasi
<p>DESAIN PEMBELAJARAN</p>	
<p>Capaian Pembelajaran</p>	<p>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,</p>

	<p>mempunyai wawasan tentang profesi informatika, serta memahami hak dan kewajiban sebagai warga digital dan aspek hukumnya.</p>
<p>Tujuan Pembelajaran</p>	<p>Pertemuan 1: Struktur Kondisional IF-Elseif-Else</p> <p>Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:</p> <ol style="list-style-type: none"> Menjelaskan konsep percabangan (kondisional) if-elseif-else dalam algoritma dan pemrograman. Menulis dan menjalankan program kondisional menggunakan C++ dan Python di Programmiz. Menyelesaikan masalah yang memerlukan pengambilan keputusan bertingkat (misal: menentukan grade nilai, jenis bilangan, kategori BMI). Membandingkan sintaks kondisional antara C++ (if/else if/else) dan Python (if/elif/else).
	<p>Pertemuan 2: Nested If (Kondisional Bersarang)</p> <p>Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:</p> <ol style="list-style-type: none"> Menjelaskan konsep nested if (kondisional bersarang) dan kapan digunakan. Menulis program dengan nested if dalam C++ dan Python di Programmiz untuk menyelesaikan masalah berlapis. Menganalisis alur eksekusi program dengan kondisional bersarang melalui tracing manual. Membedakan penggunaan nested if versus if-elseif-else dalam konteks masalah yang tepat.
	<p>Pertemuan 3: Struktur Kontrol Perulangan (For dan While)</p> <p>Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:</p> <ol style="list-style-type: none"> Menjelaskan konsep perulangan (for loop dan while loop) dalam algoritma dan pemrograman. Menggunakan for loop dan while loop dalam C++ dan Python di Programmiz. Membandingkan sintaks perulangan antara C++ dan Python. Menyelesaikan masalah yang memerlukan pengulangan tindakan (mencetak angka, menghitung deret, pola bintang).
	<p>Pertemuan 4: Perulangan Lanjutan (Break dan Continue)</p> <p>Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:</p>

	<ul style="list-style-type: none"> a. Menjelaskan fungsi break dan continue dalam perulangan. b. Menggunakan break dan continue dalam program C++ dan Python di Programmiz. c. Menyelesaikan masalah yang memerlukan penghentian atau pelompatan iterasi tertentu dalam perulangan. d. Menganalisis perbedaan perilaku program saat menggunakan break versus continue.
	<p>Pertemuan 5: Fungsi dan Modularitas Program</p> <p>Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:</p> <ul style="list-style-type: none"> a. Mendefinisikan fungsi dan menjelaskan manfaat modularitas dalam pemrograman. b. Membuat dan memanggil fungsi sederhana dalam C++ dan Python di Programmiz. c. Menggunakan parameter dan nilai kembalian (return value) dalam fungsi. d. Melakukan debugging dasar untuk menemukan dan memperbaiki kesalahan dalam kode.
	<p>Pertemuan 6: Proyek Pemrograman Sederhana</p> <p>Setelah kegiatan pembelajaran, peserta didik diharapkan mampu:</p> <ul style="list-style-type: none"> a. Menerapkan konsep kondisional, perulangan, dan fungsi untuk membuat proyek program sederhana menggunakan C++ atau Python di Programmiz. b. Menganalisis dan memperbaiki bug dalam program yang lebih kompleks secara mandiri atau kolaboratif. c. Mempresentasikan hasil proyek program dan menjelaskan logika di baliknya.
Praktis Pedagogis	<p>A. Model Pembelajaran</p> <p>Discovery Learning (untuk menemukan cara kerja sintaks/konsep), Problem-Based Learning (PBL) untuk menyelesaikan masalah dengan kode, dan Project-Based Learning (PjBL) untuk membuat aplikasi sederhana.</p>
	<p>B. Strategi Pembelajaran</p> <p>1. Mindful Learning: Fokus pada setiap baris kode, latihan tracing manual eksekusi kode, sesi debugging yang sistematis, refleksi terhadap bug sebagai peluang belajar.</p>

	<p>2. Meaningful Learning: Mengaitkan setiap konsep pemrograman dengan contoh aplikasi nyata, membandingkan implementasi dalam C++ dan Python menggunakan Programmiz.</p> <p>3. Joyful Learning: Tantangan coding kompetitif, permainan debugging interaktif, game sederhana, sesi pair programming di Programmiz.</p>
	<p>C. Metode Pembelajaran</p> <p>Demonstrasi langsung (live coding di Programmiz), latihan terstruktur, pair programming, code review, diskusi kelompok, debugging bersama, proyek pemrograman.</p>
<p>Kemitraan Pembelajaran</p>	<p>1. Lingkungan Sekolah: Guru TIK/komputer sebagai fasilitator, pengelola lab komputer. Anggota klub pemrograman atau ekstrakurikuler IT sebagai mentor sebaya.</p> <p>2. Lingkungan Luar Sekolah: Narasumber (programmer, software developer) untuk berbagi pengalaman dan inspirasi melalui daring atau kunjungan singkat.</p> <p>3. Masyarakat: Mengidentifikasi masalah sehari-hari di lingkungan sekitar yang berpotensi diselesaikan dengan program sederhana.</p>
<p>Lingkungan Pembelajaran</p>	<p>1. Ruang Fisik: Laboratorium komputer dengan perangkat yang memadai dan koneksi internet stabil untuk mengakses Programmiz. Proyektor untuk demonstrasi guru.</p> <p>2. Ruang Virtual: Platform Classroom (classroom.sman2-padang.sch.id) untuk berbagi materi. Programmiz Online IDE (programmiz.com/c-programming/online-compiler dan programmiz.com/python-programming/online-compiler) sebagai lingkungan coding utama. Portal daring (informatikasma2pdg.com).</p> <p>3. Budaya Belajar: Lingkungan yang mendukung eksperimen dan mencoba hal baru. Mendorong peserta didik untuk tidak takut membuat kesalahan (bug), melainkan melihatnya sebagai bagian alami dari proses belajar.</p>
<p>Pemanfaatan Digital</p>	<p>1. Programmiz Online IDE: Platform utama untuk menulis, menjalankan, dan menguji program C++ dan Python secara langsung tanpa instalasi software.</p>

	<p>2. Perpustakaan Digital: Mengakses tutorial pemrograman online, dokumentasi bahasa pemrograman C++ dan Python, e-book tentang algoritma.</p> <p>3. Forum Diskusi Daring: Classroom untuk memposting bug yang ditemui, meminta bantuan, atau berbagi kode.</p> <p>4. Penilaian Daring: Kuis sintaks melalui Kahoot/Quizizz. Pengumpulan file program/screenshot hasil Programmiz melalui Classroom.</p>
--	--

PENGALAMAN BELAJAR

Pertemuan 1: Struktur Kondisional If-Elseif-Else	√	Berkesadaran	√	Bermakna	√	Menggembirakan
		• Orientasi		• Apersepsi		• Motivasi

	<p>KEGIATAN AWAL (15 MENIT)</p> <p style="background-color: #d9ead3;">Berkesadaran – Orientasi</p> <p>Guru memulai dengan pertanyaan pemantik:</p> <ul style="list-style-type: none"> • "Pernahkah kalian memutuskan sesuatu berdasarkan kondisi? Misalnya: jika nilai ≥ 75 lulus, jika tidak remedi." • "Bagaimana cara komputer membuat keputusan seperti itu?" <p>Guru meminta peserta didik untuk fokus pada ide "pengambilan keputusan berbasis kondisi" dalam kehidupan sehari-hari.</p> <p>Guru menyampaikan tujuan pembelajaran dan memperkenalkan bahwa hari ini akan mengimplementasikan kondisional dalam dua bahasa: C++ dan Python, menggunakan Programmiz.</p> <p style="background-color: #f4cccc;">Bermakna – Apersepsi</p> <p>Guru mengaitkan aktivitas pengambilan keputusan dalam kehidupan nyata dengan konsep "struktur kondisional if-elseif-else" dalam pemrograman.</p> <p>Guru menunjukkan contoh kasus nyata: sistem penilaian otomatis di aplikasi belajar online yang menggunakan kondisional untuk menentukan grade peserta didik.</p> <p style="background-color: #fff2cc;">Menggembirakan – Motivasi</p>
--	---

Guru mengajak peserta didik bermain "Kondisi Saya": guru menyebut sebuah kondisi (misal: "jika kamu suka matematika, angkat tangan"), lalu peserta didik merespons.

Guru kemudian menantang: "Sekarang mari kita terjemahkan permainan ini ke dalam kode C++ dan Python!"

Guru membuka Programmiz di proyektor dan memperlihatkan betapa mudahnya menjalankan program langsung di browser.

KEGIATAN INTI (60 MENIT)

Memahami (20 menit)

a. Penjelasan Konsep

Guru menjelaskan struktur kondisional if-elseif-else:

- Konsep percabangan: program memilih satu jalur dari beberapa kemungkinan berdasarkan kondisi boolean.
- Sintaks if-elseif-else dalam C++ (else if) dan Python (elif).
- Perbedaan utama: Python menggunakan indentasi, C++ menggunakan kurung kurawal {}.

b. Demonstrasi di Programmiz

Guru membuka programmiz.com dan mendemonstrasikan kode paralel:

C++: Program penentu grade nilai (A/B/C/D) menggunakan if-else if-else.

Python: Program penentu grade nilai yang sama menggunakan if-elif-else.

Guru menjalankan kedua program dan membandingkan outputnya.

Mengaplikasi (25 menit)

a. Latihan Terbimbing

Peserta didik secara berpasangan membuka Programmiz dan mengerjakan latihan:

- Program penentu kategori BMI (kurus/normal/gemuk/obesitas) dalam C++ dan Python.
- Program penentu jenis bilangan (positif/negatif/nol) dalam C++ dan Python.

b. Diferensiasi Proses

- Peserta didik reguler: mengerjakan soal dengan panduan template kode.
- Peserta didik cepat: membuat program penentu diskon berdasarkan total belanja (bertingkat 3 kondisi atau lebih).

Guru berkeliling memantau layar Programmiz peserta didik dan memberikan bantuan langsung.

Merefleksi (15 menit)

Guru meminta peserta didik menjawab di Classroom:

- "Apa perbedaan utama sintaks kondisional C++ dan Python yang kamu temukan hari ini?"
- "Program mana yang lebih mudah dibaca? Mengapa?"

Beberapa peserta didik mempresentasikan dan menjalankan kode mereka di Programmiz di depan kelas.

KEGIATAN PENUTUP (15 MENIT)

a. Umpan Balik Konstruktif

Guru memberikan apresiasi atas usaha peserta didik. Guru mengulas kesalahan umum: lupa else, kondisi yang tumpang tindih, perbedaan indentasi Python vs kurung kurawal C++.

b. Menyimpulkan Pembelajaran

Peserta didik secara kolaboratif menyimpulkan: if-elseif-else memungkinkan program mengambil keputusan bertingkat. Sintaks C++ dan Python berbeda namun logikanya sama.

c. Perencanaan Pembelajaran Selanjutnya

Guru memberikan pengantar untuk pertemuan berikutnya (nested if) dan tugas: cari satu contoh kasus di kehidupan sehari-hari yang butuh kondisi bersarang.

Pertemuan 2: Nested If (Kondisional Bersarang)

√	Berkesadaran	√	Bermakna	√	Menggembirakan
	• Orientasi		• Apersepsi		• Motivasi

KEGIATAN AWAL (15 MENIT)

Berkesadaran – Orientasi

Guru memulai dengan pertanyaan pemantik:

- "Bagaimana jika syarat untuk masuk suatu keputusan tidak cukup satu kondisi? Misalnya: untuk mendapatkan beasiswa, harus nilai ≥ 80 dan aktif organisasi."
- "Bagaimana cara kita memeriksa dua kondisi secara bertingkat dalam program?"

Guru meminta peserta didik fokus pada ide "kondisi di dalam kondisi" sebagai pemikiran awal.

Bermakna – Apersepsi

Guru mengaitkan konsep nested if dengan situasi nyata:

- Sistem loket: jika memiliki tiket → jika tiket VIP → masuk pintu A, jika reguler → masuk pintu B. Jika tidak memiliki tiket → beli tiket.

Guru menekankan bahwa nested if adalah kondisional di dalam kondisional, dan ini berbeda dari penggunaan elif/else if biasa.

Menggembirakan – Motivasi

Guru mengajak peserta didik bermain "Pohon Keputusan":

Guru menggambar pohon keputusan di papan tulis: "Apakah kamu punya uang? → Ya → Apakah warung buka? → Ya → Beli makan. Tidak → Masak sendiri. Tidak → Pinjam dulu."

Peserta didik diminta menebak kode nested if untuk pohon tersebut sebelum melihat jawabannya di Programmiz.

KEGIATAN INTI (60 MENIT)

Memahami (20 menit)

a. Penjelasan Konsep Nested If

Guru menjelaskan:

- Nested if: pernyataan if di dalam blok if atau else lainnya.
- Kapan nested if lebih tepat daripada if-elseif-else: ketika kondisi ke-2 hanya relevan jika kondisi ke-1 terpenuhi.
- Perhatian khusus: indentasi Python yang bertingkat vs kurung kurawal bersarang C++.

b. Demonstrasi di Programmiz

Guru mendemonstrasikan program loker digital:

C++: Nested if untuk menentukan akses ruangan berdasarkan jabatan dan shift kerja.

Python: Program yang sama, menekankan pentingnya indentasi yang benar.

Guru sengaja menunjukkan error akibat indentasi yang salah di Python, lalu memperbaikinya bersama.

Mengaplikasi (25 menit)

a. Latihan Kelompok

Peserta didik dalam kelompok kecil (2-3 orang) mengerjakan latihan di Programmiz:

- Program seleksi penerimaan siswa: jika nilai US $\geq 80 \rightarrow$ periksa nilai wawancara; jika nilai wawancara $\geq 70 \rightarrow$ diterima, jika tidak \rightarrow cadangan. Jika nilai US $< 80 \rightarrow$ tidak diterima. (C++ dan Python)
- Program penentu tarif ojek online berbasis jarak dan waktu (siang/malam). (C++ dan Python)

b. Diferensiasi

- Peserta didik cepat: tambahkan kondisi ketiga (misal: bonus diskon jika pengguna premium).
- Peserta didik yang kesulitan: gunakan template dengan beberapa bagian kosong untuk diisi.

Guru mendorong peserta didik untuk menguji berbagai kombinasi input di Programmiz.

Merefleksi (15 menit)

Guru memfasilitasi diskusi:

- "Kapan sebaiknya kita menggunakan nested if, dan kapan lebih baik menggunakan elif/else if?"
- "Apa risiko terbesar dari nested if yang terlalu dalam?"

Peserta didik menuliskan refleksi di Classroom dan saling memberikan komentar pada kode teman.

KEGIATAN PENUTUP (15 MENIT)

a. Umpan Balik Konstruktif

Guru menekankan pentingnya keseimbangan antara nested if dan elif: nested if yang terlalu dalam (deep nesting)

	<p>membuat kode sulit dibaca. Guru menunjukkan refactoring sederhana.</p> <p>b. Menyimpulkan Pembelajaran</p> <p>Nested if memungkinkan pemeriksaan kondisi bertingkat yang lebih spesifik. Harus digunakan dengan bijak agar kode tetap mudah dibaca, baik di C++ maupun Python.</p> <p>c. Perencanaan Pembelajaran Selanjutnya</p> <p>Guru memberikan pengantar untuk pertemuan berikutnya (struktur perulangan for dan while) dan tugas: cari contoh proses berulang dalam kehidupan sehari-hari.</p>					
Pertemuan 3: Struktur Kontrol Perulangan (For dan While)	√	Berkesadaran	√	Bermakna	√	Menggembirakan
		• Orientasi		• Apersepsi		• Motivasi
	KEGIATAN AWAL (15 MENIT)					
	Berkesadaran – Orientasi					
	<p>Guru memulai dengan pertanyaan:</p> <ul style="list-style-type: none"> • "Bayangkan jika kita harus mencetak 'Selamat datang!' 100 kali. Apakah kita tulis 100 baris kode?" • "Bagaimana cara yang jauh lebih efisien untuk melakukan tugas yang berulang?" <p>Guru meminta peserta didik fokus pada ide "efisiensi melalui pengulangan terstruktur".</p> <p>Guru menyampaikan tujuan pembelajaran dan menekankan bahwa for dan while loop adalah dua alat utama perulangan yang akan dipraktikkan di Programmiz.</p>					
	Bermakna – Apersepsi					
<p>Guru mengaitkan konsep perulangan dengan kehidupan nyata:</p> <ul style="list-style-type: none"> • Mesin pabrik yang mencetak produk berulang kali sampai target terpenuhi → for loop. • Alarm yang terus berbunyi sampai tombol snooze ditekan → while loop. <p>Guru menekankan perbedaan: for loop digunakan saat jumlah iterasi sudah diketahui, while loop saat kondisi berhentinya yang diketahui.</p>						
Menggembirakan – Motivasi						
<p>Guru mengajak peserta didik bermain "Robot Berulang":</p>						

Guru bertanya: "Saya perintahkan robot untuk melompat 5 kali – ini for loop. Saya perintahkan robot untuk terus jalan sampai bertemu dinding – ini while loop."

Peserta didik secara fisik bergiliran menjadi "robot" yang menjalankan loop. Kemudian guru menantang: "Sekarang mari terjemahkan gerakan robot ini ke kode C++ dan Python di Programmiz!"

KEGIATAN INTI (60 MENIT)

Memahami (20 menit)

a. Penjelasan For Loop dan While Loop

Guru menjelaskan:

- For loop: sintaks C++ (`for(int i=0; i<n; i++)`) vs Python (`for i in range(n)`).
- While loop: sintaks C++ (`while(kondisi){}`) vs Python (`while kondisi:`).
- Kapan menggunakan for vs while: jumlah iterasi diketahui vs berbasis kondisi.

b. Demonstrasi di Programmiz

C++: For loop mencetak angka 1-10, while loop menebak angka.

Python: Program yang sama untuk perbandingan sintaks langsung.

Guru menjalankan kedua versi dan mendiskusikan persamaan serta perbedaannya.

Mengaplikasi (25 menit)

a. Latihan Berpasangan

Peserta didik berpasangan mengerjakan di Programmiz:

- For loop: mencetak tabel perkalian suatu angka. (C++ dan Python)
- For loop: mencetak pola segitiga bintang. (C++ dan Python)
- While loop: program hitung mundur dari angka yang diinput pengguna. (C++ dan Python)

b. Diferensiasi

- Reguler: selesaikan ketiga soal di atas.

	<ul style="list-style-type: none"> • Cepat: buat program menghitung jumlah digit suatu bilangan menggunakan while loop. <p>Peserta didik diminta untuk mem-screenshot hasil output Programmiz dan mengunggah ke Classroom.</p>					
	<p>Merefleksi (15 menit)</p> <p>Guru memfasilitasi refleksi:</p> <ul style="list-style-type: none"> • "Kapan kamu akan memilih for loop dan kapan while loop? Berikan alasannya." • "Apa perbedaan terbesar sintaks for loop antara C++ dan Python?" <p>Beberapa pasang mempresentasikan hasil kode pola bintang mereka dan menjalankannya di Programmiz di depan kelas.</p>					
<p>KEGIATAN PENUTUP (15 MENIT)</p>						
<p>a. Umpan Balik Konstruktif</p> <p>Guru mengoreksi kesalahan umum: infinite loop akibat kondisi yang salah, off-by-one error pada for loop, perbedaan range() Python yang tidak inklusif pada batas atas.</p>						
<p>b. Menyimpulkan Pembelajaran</p> <p>For dan while loop adalah alat fundamental untuk mengotomatisasi pengulangan. C++ dan Python memiliki sintaks berbeda namun konsepnya identik.</p>						
<p>c. Perencanaan Pembelajaran Selanjutnya</p> <p>Guru memberikan pengantar untuk break dan continue, dan tugas: buat program while loop sederhana yang memiliki kondisi berhenti.</p>						
<p>Pertemuan 4: Looping Lanjutan (Break dan Continue)</p>	√	Berkesadaran	√	Bermakna	√	Menggembirakan
	<ul style="list-style-type: none"> • Orientasi 		<ul style="list-style-type: none"> • Apersepsi 		<ul style="list-style-type: none"> • Motivasi 	
<p>KEGIATAN AWAL (15 MENIT)</p> <p>Berkesadaran – Orientasi</p> <p>Guru memulai dengan skenario:</p> <ul style="list-style-type: none"> • "Bayangkan kamu sedang mencari nama di daftar panjang. Begitu menemukan nama yang dicari, kamu 						

berhenti – tidak perlu membaca sisa daftar. Ini seperti BREAK."

- "Atau kamu sedang memeriksa daftar, dan kamu melewati nama yang diawali huruf 'X' karena ada aturan khusus. Ini seperti CONTINUE."

Guru meminta peserta didik fokus pada ide "kontrol aliran dalam perulangan".

Bermakna – Apersepsi

Guru mengaitkan break dan continue dengan contoh nyata:

- Break: tombol emergency stop pada conveyor belt pabrik – hentikan segera saat ada masalah.
- Continue: sistem filtrasi email spam – lewati email spam, lanjutkan memproses email berikutnya.

Guru menekankan bahwa break dan continue memberi kendali lebih presisi atas jalannya perulangan, baik di C++ maupun Python.

Menggembirakan – Motivasi

Guru mengajak simulasi fisik:

"Kita akan berjalan melingkar (loop). Saya sebut angka 1-10. Jika saya teriak 'BREAK' → semua berhenti. Jika saya teriak 'CONTINUE angka 5' → lewati giliran saat angka 5, lanjutkan ke berikutnya."

Peserta didik melakukan simulasi, lalu guru menantang: "Sekarang kita kodekan permainan ini di Programmiz C++ dan Python!"

KEGIATAN INTI (60 MENIT)

Memahami (20 menit)

a. Penjelasan Break dan Continue

Guru menjelaskan:

- Break: menghentikan perulangan segera, eksekusi berlanjut ke baris setelah loop.
- Continue: melewati iterasi saat ini, langsung ke iterasi berikutnya.
- Sintaks identik di C++ dan Python, namun perbedaannya ada pada struktur blok.

b. Demonstrasi di Programmiz

C++: Program pencarian bilangan prima pertama < 100 (break saat ditemukan). Program mencetak bilangan 1-20 kecuali kelipatan 3 (continue).

Python: Program yang sama untuk perbandingan. Guru menampilkan keduanya side-by-side di Programmiz.

Mengaplikasi (25 menit)

a. Latihan Kelompok

Peserta didik dalam kelompok mengerjakan di Programmiz:

- Program game tebak angka: pengguna input angka terus-menerus (while loop) sampai menebak benar (break). Jika input bukan angka valid, skip (continue). (C++ dan Python)
- Program mencetak bilangan 1-50, lewati bilangan yang habis dibagi 7 (continue), berhenti jika menemukan bilangan yang habis dibagi 13 (break). (C++ dan Python)

b. Diferensiasi

- Cepat: program menu dengan loop – tampilkan pilihan berulang, break saat user pilih 'keluar', continue jika input tidak valid.

Guru mendorong peserta didik untuk bereksperimen: "Apa yang terjadi jika break dihapus? Apa yang berubah jika continue diganti if?"

Merefleksi (15 menit)

Guru memfasilitasi diskusi:

- "Apa perbedaan perilaku program ketika menggunakan break vs menghapus kondisi while saja?"
- "Kapan continue lebih efisien daripada menggunakan if biasa untuk melewati kondisi?"

Peserta didik mengunggah kode game tebak angka beserta screenshot output ke Classroom.

KEGIATAN PENUTUP (15 MENIT)

a. Umpan Balik Konstruktif

	<p>Guru mengoreksi: penggunaan break di luar loop (error), continue yang menyebabkan infinite loop karena variabel counter tidak diperbarui, perbedaan efek break pada nested loop.</p> <p>b. Menyimpulkan Pembelajaran</p> <p>Break menghentikan loop sepenuhnya, continue hanya melewati satu iterasi. Keduanya memberikan kontrol aliran yang lebih presisi dan tersedia di C++ maupun Python.</p> <p>c. Perencanaan Pembelajaran Selanjutnya</p> <p>Guru memberikan pengantar untuk fungsi dan modularitas, dan tugas: amati kode program yang sudah dibuat bagian mana yang bisa dijadikan fungsi tersendiri?</p>					
Pertemuan 5: Fungsi dan Modularitas Program	√	Berkesadaran	√	Bermakna	√	Menggembirakan
		• Orientasi		• Apersepsi		• Motivasi
	KEGIATAN AWAL (15 MENIT)					
	Berkesadaran – Orientasi					
	Guru memulai dengan menampilkan dua versi kode di Programmiz:					
	<ul style="list-style-type: none"> • Versi 1: Program kalkulator panjang tanpa fungsi (semua kode di main/badan utama). • Versi 2: Program kalkulator yang sama namun dengan fungsi (tambah, kurang, kali, bagi) yang terpisah. 					
	Guru bertanya: "Mana yang lebih mudah dibaca, diperbaiki, dan dikembangkan? Mengapa?"					
	Guru meminta peserta didik fokus pada ide "mengorganisir kode menjadi modul yang bisa digunakan kembali".					
	Bermakna – Apersepsi					
	Guru mengaitkan fungsi dengan analogi nyata:					
	<ul style="list-style-type: none"> • Resep masakan: satu resep (fungsi) bisa digunakan berulang kali tanpa menulis ulang langkah-langkahnya. • Remote TV: satu tombol (fungsi) menjalankan serangkaian instruksi kompleks di dalam TV. 					
	Guru menekankan tiga manfaat utama fungsi: reusability (bisa dipakai berulang), readability (kode mudah dibaca), dan maintainability (mudah diperbaiki).					
	Menggembirakan – Motivasi					

Guru mengajak peserta didik bermain "Pabrik Fungsi":
Guru menyebut sebuah tugas (misalnya: "cetak garis pemisah"), dan peserta didik harus "mendefinisikan fungsi"-nya secara lisan (nama, input apa, output apa).
Kelompok tercepat yang benar mendapat poin. Kemudian semua kelompok mengimplementasikan fungsi tersebut di Programmiz.

KEGIATAN INTI (60 MENIT)

Memahami (20 menit)

a. Penjelasan Fungsi dan Modularitas

Guru menjelaskan:

- Struktur fungsi: nama, parameter (input), badan fungsi, return value (output).

C++: `int namaFungsi(int param1) { ... return nilai; }`

Python: `def namaFungsi(param1): ... return nilai`

- Perbedaan fungsi dengan dan tanpa return value (void di C++, tanpa return di Python).
- Cara memanggil fungsi dan meneruskan argumen.

b. Demonstrasi di Programmiz

Guru mendemonstrasikan refactoring program kalkulator dari kode tanpa fungsi menjadi bermodular. Guru juga memperkenalkan teknik debugging: membaca pesan error, menggunakan print/cout di titik tertentu untuk melacak nilai variabel.

Mengaplikasi (25 menit)

a. Latihan

Peserta didik secara individu atau berpasangan mengerjakan di Programmiz:

- Buat fungsi `hitungLuas(panjang, lebar)` dan `hitungKeliling(panjang, lebar)` untuk persegi panjang. Program utama memanggil kedua fungsi tersebut. (C++ dan Python)
- Refactoring: ambil program kondisional (pertemuan 1) dan bungkus logika utamanya ke dalam fungsi `cekGrade(nilai)`. (C++ dan Python)

- Debugging exercise: guru membagikan kode dengan 3 bug tersembunyi. Peserta didik harus menemukan dan memperbaiki semua bug di Programiz.

b. Diferensiasi

- Cepat: buat fungsi rekursif sederhana (misalnya: faktorial).

Guru mendorong diskusi antar peserta didik saat debugging.

Merefleksi (15 menit)

Guru memfasilitasi refleksi:

- "Bagaimana fungsi membuat program kamu lebih mudah dipahami dan diperbaiki?"
- "Apa strategi debugging yang paling efektif yang kamu gunakan hari ini?"

Peserta didik mengunggah kode refactoring beserta screenshot output ke Classroom dan memberikan peer review pada kode teman satu kelompok.

KEGIATAN PENUTUP (15 MENIT)

a. Umpan Balik Konstruktif

Guru mengoreksi: fungsi yang dipanggil sebelum didefinisikan (C++: perlu deklarasi maju/forward declaration), tipe return yang tidak sesuai, parameter yang tidak digunakan.

b. Menyimpulkan Pembelajaran

Fungsi adalah inti dari pemrograman modular. Kode yang terstruktur dengan baik menjadi lebih mudah dibaca, di-debug, dan dikembangkan – berlaku untuk C++ maupun Python.

c. Perencanaan Pembelajaran Selanjutnya

Guru menginformasikan pertemuan berikutnya adalah proyek akhir. Peserta didik diminta memilih topik proyek mereka dan mulai merancang flowchart/algorithm sebagai persiapan.

Pertemuan 6: Proyek Pemrograman Sederhana	√	Berkesadaran	√	Bermakna	√	Menggembirakan
		• Orientasi		• Apersepsi		• Motivasi

KEGIATAN AWAL (15 MENIT)

Berkesadaran – Orientasi

Guru memulai dengan kilas balik perjalanan belajar:

"Selama 5 pertemuan, kita sudah belajar kondisional (if-elseif-else), nested if, for/while loop, break/continue, dan fungsi. Hari ini saatnya kalian menggabungkan semua itu menjadi sebuah karya!"

Guru meminta peserta didik fokus pada ide "menciptakan solusi nyata dengan kode yang telah dipelajari".

Guru menyampaikan instruksi proyek, rubrik penilaian, dan batas waktu secara jelas.

Bermakna – Apersepsi

Guru menekankan bahwa tujuan akhir belajar pemrograman adalah menciptakan solusi bermakna.

Guru menghubungkan setiap fitur proyek dengan konsep yang telah dipelajari:

- Menu program → perulangan while + break
- Validasi input → kondisional + continue
- Kalkulasi → fungsi
- Keputusan hasil → nested if atau if-elseif-else

Menggembirakan – Motivasi

Guru menampilkan 3 contoh proyek sederhana yang sudah berjalan di Programmiz:

- Kalkulator BMI lengkap dengan kategori dan saran kesehatan.
- Game tebak angka dengan scoring dan batas percobaan.
- Program nilai rapor: input nilai mata pelajaran → hitung rata-rata → tentukan grade → tampilkan predikat.

Guru berkata: "Kalian sudah punya semua bahan untuk membuat ini. Sekarang waktunya berkreasi!"

KEGIATAN INTI (60 MENIT)

Memahami (20 menit)

a. Penjelasan Tugas Proyek

Setiap kelompok (2-3 orang) memilih satu dari pilihan proyek berikut (atau mengusulkan sendiri):

- Kalkulator Saintifik Sederhana: menu pilihan operasi, fungsi untuk setiap operasi, validasi input.
- Game Tebak Angka Plus: tebak angka dengan hint (lebih besar/kecil), batas percobaan, skor berdasarkan jumlah tebakan.
- Program Laporan Nilai: input nilai beberapa mata pelajaran, hitung rata-rata, tentukan grade dan predikat kelulusan.
- Konverter Unit: konversi suhu, panjang, atau berat dengan menu pilihan menggunakan fungsi.

b. Perencanaan

Peserta didik membuat flowchart/pseudocode proyek mereka terlebih dahulu sebelum mulai coding di Programmiz. Guru memeriksa rancangan setiap kelompok.

Mengaplikasi (25 menit)

a. Pengembangan Proyek

Peserta didik bekerja secara kolaboratif mengembangkan proyek di Programmiz. Setiap proyek harus:

- Ditulis dalam dua bahasa: C++ dan Python (kode paralel).
- Menggunakan minimal 3 dari 5 konsep yang dipelajari: kondisional, nested if, for/while, break/continue, fungsi.
- Memiliki minimal satu fungsi yang dapat dipanggil berulang kali.
- Menyertakan input dari pengguna dan validasi input dasar.

b. Debugging

Peserta didik menguji program mereka dengan berbagai skenario input (normal, ekstrem, tidak valid). Guru berperan sebagai fasilitator dan debugger ahli, berkeliling memberikan bimbingan sesuai kebutuhan.

c. Diferensiasi Dukungan

- Kelompok yang kesulitan: guru memberikan template struktur program.
- Kelompok cepat: tantangan tambahan – buat program berjalan berulang (loop menu) hingga pengguna memilih keluar.

Merefleksi (15 menit)

a. Presentasi Proyek

Setiap kelompok mempresentasikan proyek mereka (maksimal 3 menit per kelompok):

- Menjelaskan masalah yang diselesaikan.
- Mendemonstrasikan program berjalan di Programmiz (C++ dan Python).
- Menjelaskan konsep-konsep yang digunakan dan tantangan yang diatasi.

b. Diskusi Refleksi

- “Apa tantangan terbesar saat mengerjakan proyek ini dan bagaimana kamu mengatasinya?”
- “Konsep mana yang paling sering kamu gunakan? Mengapa?”
- "Apa yang ingin kamu kembangkan dari proyek ini jika ada waktu lebih?"

KEGIATAN PENUTUP (15 MENIT)

a. Umpan Balik Konstruktif

Guru memberikan umpan balik spesifik pada setiap proyek: keberhasilan logika, kebersihan kode, kreativitas solusi, dan saran peningkatan. Guru menyoroti kode terbaik sebagai contoh.

b. Menyimpulkan Pembelajaran

Guru dan peserta didik bersama-sama merefleksikan perjalanan 6 pertemuan: dari kondisional sederhana hingga proyek nyata. Guru menekankan: "Kemampuan debug dan berpikir sistematis yang kalian latih adalah keterampilan yang sangat berharga di era digital."

c. Perencanaan Ke Depan

Guru mendorong peserta didik untuk terus berlatih di Programmiz, mengeksplorasi materi lanjutan (array, string,

	<p>OOP), dan bergabung dengan komunitas coding online. Guru menginformasikan pengumpulan portofolio digital (semua kode + jurnal refleksi) melalui Classroom.</p>
ASESMEN	
Assesment as Learning (AS)	<ol style="list-style-type: none"> 1. Self-Assessment: Jurnal refleksi setelah setiap sesi coding tentang pemahaman konsep kondisional, perulangan, dan fungsi; bug yang ditemui dan cara mengatasinya; serta perbandingan implementasi C++ vs Python. 2. Peer Assessment: Peserta didik saling melakukan code review terhadap program teman di Programmiz, memberikan masukan konstruktif, dan menilai partisipasi dalam kelompok. 3. Diskusi Kode: Observasi partisipasi aktif dalam diskusi tentang logika kode, debugging, dan berbagi solusi antar bahasa (C++ dan Python).
Assesment for Learning (AFL)	<ol style="list-style-type: none"> 1. Kuis Singkat/Tanya Jawab Lisan: Menguji pemahaman sintaks kondisional (if-elseif-else, nested if), perulangan (for/while, break/continue), dan fungsi dalam C++ dan Python. 2. Latihan Coding Harian: Tugas coding singkat di Programmiz untuk setiap konsep – peserta didik mengunggah screenshot hasil ke Classroom. 3. Umpan Balik Guru: Umpan balik langsung saat demonstrasi live coding di Programmiz, saat debugging bersama, atau saat mengoreksi program. 4. Observasi: Guru mengamati kemampuan peserta didik merancang algoritma, menulis kode C++ dan Python, mengidentifikasi bug, dan menerapkan konsep.
Assesment of Learning (AOL)	<ol style="list-style-type: none"> 1. Tes Tertulis: Menulis pseudocode/flowchart, menganalisis potongan kode C++ dan Python, memprediksi output, mengidentifikasi bug, menjelaskan konsep (kondisional, perulangan, fungsi). 2. Penilaian Kinerja/Proyek Pemrograman: Proyek Akhir di Programmiz penilaian terhadap program C++ dan Python yang dihasilkan (fungsionalitas, kode bersih, penggunaan konsep yang tepat, kreativitas). Rubrik mencakup ketepatan, fungsionalitas, keterbacaan kode, dan kreativitas.

memperbaiki bug dalam waktu tertentu.

4. **Portofolio Digital:** Kumpulan semua program (C++ dan Python), screenshot output Programmiz, laporan proyek, dan jurnal refleksi yang menunjukkan perkembangan kompetensi peserta didik.

Padang, 11 Mei 2026

Mengetahui,

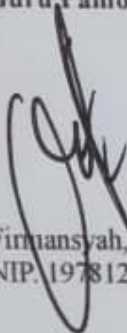
Wakil Kurikulum SMAN 2 Padang



Fitri Sari Ramadhani, M.Pd
NIP. 19831010 200803 2 002

Menyetujui,

Guru Pamong



Firmansyah, M.Kom
NIP. 19781210 202121 1 003

Mahasiswa



Ikhsan Budi Maulana, S.Kom
NIM. 25427697