



KEMENTERIAN PENDIDIKAN DASAR DAN MENENGAH
BADAN, STANDAR, KURIKULUM, DAN ASESMEN PENDIDIKAN
PUSAT KURIKULUM DAN PEMBELAJARAN
REPUBLIK INDONESIA

MODUL AJAR

KODING & KECERDASAN ARTIFISIAL

SMA/SMK

MENERAPKAN PEMROGRAMAN BERBASIS TEKS



KELAS



Modul Ajar SMA/SMK Kelas X
Mata Pelajaran Koding & Kecerdasan Artifisial
Menerapkan Pemrograman Berbasis Teks

Pengarah

Kepala Badan Standar, Kurikulum dan Asesmen Pendidikan
Kementerian Pendidikan Dasar dan Menengah
Prof. Dr. Toni Toharudin, S.Si., M.Sc.

Staf Khusus Menteri Pendidikan Dasar dan Menengah
Bidang Transformasi Digital dan Kecerdasan Buatan
Dr. Muhammad Muchlas Rowi, S.F., S.H., M.M.

Penanggung Jawab

Kepala Pusat Kurikulum dan Pembelajaran
Badan Standar, Kurikulum, dan Asesmen Pendidikan
Kementerian Pendidikan Dasar dan Menengah
Dr. Laksmi Dewi, M.Pd.

Penyusun

Listyanti Dewi Astuti, S.Pd., M.Kom. (SMKN 12 Malang)

Penelaah

Nur Rofika Ayu Shinta Amalia, S.Si. (Pusat Kurikulum dan Pembelajaran)
Prayoga Rendra Vendiktama, S.Pd. (Pusat Kurikulum dan Pembelajaran)
Irwan Nurwiansyah, S.Pd. (Pusat Kurikulum dan Pembelajaran)
Fijar Hafizh, S.E. (Pusat Kurikulum dan Pembelajaran)
Abdallah Sadurdeen Khan, B.Eng., M.Ed. (Koding Next)
Muhammad Rizki Dwi Putra, S.Pd., M.T. (Koding Next)
Permata Nur Miftahur Rizki, PhD. (Prasetiya Mulya University)
Suprihatin, S.Kom., M.Pd. (SMAN 9 Yogyakarta)

Desainer Sampul

Nayla Syifa Efendi

Penerbit

Pusat Kurikulum dan Pembelajaran
Badan Standar, Kurikulum, dan Asesmen Pendidikan
Kementerian Pendidikan Dasar dan Menengah

2025

MODUL AJAR

KODING DAN KECERDASAN ARTIFISIAL

Kelas : X (SMA/SMK/MA/MAK)
Semester : Gasal
Alokasi Waktu : 6 Pertemuan (@90 menit)

DIMENSI PROFIL LULUSAN

☐ DPL1 Keimanan dan Ketakwaan terhadap Tuhan YME	☒ DPL3 Penalaran Kritis	☒ DPL5 Kolaborasi	☐ DPL7 Kesehatan
☐ DPL2 Kewargaan	☒ DPL4 Kreativitas	☒ DPL6 Kemandirian	☐ DPL8 Komunikasi

TUJUAN PEMBELAJARAN

Peserta didik mampu menerapkan pemrograman berbasis teks

KERANGKA PEMBELAJARAN

Praktik Pedagogis: Model Pembelajaran Kontekstual	Lingkungan Pembelajaran - Ruang fisik: ruang laboratorium komputer - Ruang virtual: <i>Learning Management System (LMS)</i>
Kemitraan Pembelajaran Kolaborasi dengan pengajar matematika (menyusun skenario uji coba program menghitung akar kuadrat dan nilai trigonometri)	Pemanfaatan Digital - Video (YouTube) <i>Integrated Development Environment</i> atau <i>online compiler</i> untuk bahasa pemrograman python, C, atau JavaScript, misalnya Visual Studio Code, Google Colab, Programiz, dan lain-lain

Referensi Buku Ajar:

- **Algoritma & Pemrograman Python Series** oleh Irwan A. Kautsar, Ph.D
- **Buku Algoritma dan Pemrograman Bahasa C** oleh Kharisma Monika Dian Pertiwi, dkk
- **Dasar Algoritma dan Pemrograman Javascript** oleh Irwan Lewenusa, M.Kom.

LANGKAH-LANGKAH PEMBELAJARAN

Pertemuan 1: Memahami Konsep Penerapan Algoritma ke Dalam Pemrograman Berbasis Teks, Menerapkan Input, Output, dan Komentar

✓ “MEMAHAMI” dengan prinsip “berkesadaran” dan “menggembirakan”

1. Guru menyapa siswa, memastikan siswa siap untuk belajar, mengkondisikan siswa untuk belajar
2. Guru menjelaskan tujuan pembelajaran dan mengaitkannya dengan situasi nyata
3. Guru menjelaskan teknik asesmen dalam pembelajaran pada topik menerapkan pemrograman berbasis teks
4. Guru melakukan asesmen awal dengan beberapa pertanyaan singkat melalui LMS untuk mengetahui tingkat pemahaman awal siswa.
5. Siswa menyimak simulasi yang dilaksanakan guru dengan menampilkan sebuah video simulasi, seperti "Bagaimana GPS menentukan rute tercepat?" atau "Bagaimana YouTube dapat merekomendasikan video yang sesuai dengan minat kalian?"

Guru bertanya (contoh):

- "Menurut kalian, bagaimana aplikasi seperti Google Maps dapat menentukan rute tercepat?"

Suplemen video: “**How does Google Maps find the shortest path?**”

Melalui kanal “**The Unqualified Tutor**”. Tautan <https://s.id/en20G>



Catatan:

Jika diperlukan, siswa/guru dapat menyalakan *subtitle/auto-translate caption* bahasa Indonesia untuk video berbahasa Inggris, atau gunakan perangkat KA untuk *transcribe* video. Referensi cara penggunaan perangkat KA untuk terjemahan video, seperti *Keevi* dan *Kapwing* bisa dibaca di artikel: <https://s.id/ETfri>

- "Bagaimana YouTube dapat merekomendasikan video yang sesuai dengan minat kalian?"

Suplemen video: "**The YouTube Algorithms in 2025 — Explained!**"

Melalui kanal "**Creator Insider**". Tautan <https://s.id/fEmTJ> :



6. Siswa bertanya tentang isi video, dan guru memberikan jawaban atau referensi ke sumber referensi yang dapat dipelajari lebih lanjut oleh siswa
7. Siswa melakukan eksperimen kecil (mengaitkan dengan materi sebelumnya, yaitu perbandingan algoritma) melalui **LKPD 1.1 - Membandingkan Algoritma Perjalanan Menuju ke Sekolah** (<https://s.id/LKPD1-1>). LKPD dapat dikelola melalui **Learning Management System**

✓ MENGAPLIKASI dengan prinsip "bermakna"

8. Siswa menuliskan algoritma sederhana dalam bentuk *pseudocode* atau *flowchart* untuk salah satu masalah yang mereka hadapi sehari-hari

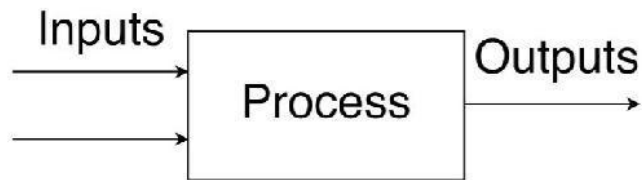
Contoh pembagian skenario:

- Kelompok 1: Membuat algoritma untuk sistem ATM yang menentukan apakah saldo cukup atau tidak untuk transaksi.
- Kelompok 2: Membuat algoritma sederhana untuk sistem CAPTCHA yang menentukan apakah input dari pengguna valid atau tidak.
- Kelompok 3: Membuat algoritma untuk pemrosesan tiket digital yang hanya mengizinkan pemesanan jika kursi masih tersedia.

9. Guru mencontohkan bagaimana pseudocode yang dibuat oleh tersebut diubah menjadi kode dalam bahasa C, Python, **atau** JavaScript, dan **mempertegas perbedaan mendasar antara bahasa pemrograman berbasis blok/visual vs. bahasa pemrograman berbasis teks.**
10. Siswa menjalankan kode yang diberikan guru menggunakan *IDE* yang dipilih. Guru menjelaskan kegunaan setiap baris instruksi, dan siswa mencatat dalam bentuk komentar di dalam kode program. Kegiatan ini dapat dilakukan dengan bantuan **LKPD 1.2: “Mencoba Kode Program”** (<https://s.id/NwU2K>)
 - a. Siswa mencoba kode program persis dengan yang diberikan guru
 - b. Siswa, kemudian, mencoba memodifikasi kode program dan mengamati hasilnya
11. Guru menjelaskan dan mendemonstrasikan cara menangani *syntax error* pada kode program, termasuk bagaimana mencari bantuan di dunia maya dalam menangani *error*.
12. Siswa mendiskusikan hasil percobaan kode programnya dalam kelompok. Diskusi dapat dilakukan dengan panduan yang tercantum di dalam **LKPD 1.2: “Mencoba Kode Program”**.

✓ **MEMAHAMI dengan prinsip “berkesadaran”**

13. Guru menampilkan diagram blok *Input/Output* dan memberi penjelasan tentang *input output*, lalu menunjukkan bagian *input output* pada program yang telah dicoba oleh siswa



14. Siswa mencari lirik lagu favoritnya menggunakan mesin pencari
15. Siswa mengidentifikasi *Input* dan *Output* dari proses pencairan lirik lagu favorit
16. Guru mencontohkan teknik *input/output* dalam bahasa pemrograman
17. Siswa mempraktikkan *input/output* yang dicontohkan guru dengan modifikasi:
 - o Mengganti data yang di-*input*-kan
 - o Mengganti data yang di-*output*-kan
 - o Mencoba membuat kode untuk output *multi-line*
 - o Menandai bagian input dan output dalam kode program yang dibuat dengan memanfaatkan komentar.

Contoh implementasi Input Output:

Output

```
Masukkan nama Anda: Jennie  
Halo, Jennie
```

Output

```
Halo, siapa Nama Anda? : Santi  
Di mana Anda tinggal? : Malang  
Halo Santi  
Kota Malang sangat indah!
```

18. Guru meminta siswa untuk membaca berbagai sumber belajar di internet tentang format *output* (*output formatting*), jenis-jenis *error*, serta cara menanganinya dalam bahasa pemrograman yang dipilih **di rumah**, dan membuat catatan bermakna dari sumber-sumber yang dibaca. Aktivitas

rumah ini dapat difasilitasi dengan **LKPD 1.3: Membuat Catatan Bermakna** (<https://s.id/sbWK3>)

✓ **MEREFLEKSI dengan prinsip “berkesadaran”**

19. Setiap siswa menuliskan refleksi mendalam dalam jurnal belajar di LMS (kegiatan refleksi ini dapat difasilitasi dengan **JURNAL REFLEKSI BELAJAR** terlampir di <https://s.id/jurnal-refleksi>), meliputi
 - o Pemahaman Konsep:
 - o Tantangan & Kesulitan
 - o Strategi Perbaikan
 - o Aplikasi Nyata
 - o Setiap kelompok berbagi wawasan mereka mengenai pemecahan masalah yang telah mereka lakukan.
20. Siswa berdiskusi mengenai solusi terbaik yang mereka temukan, apakah ada cara yang lebih efisien, serta bagaimana konsep yang dipelajari dapat diterapkan dalam dunia nyata.
21. Guru memberikan umpan balik spesifik terkait hasil kerja siswa, menyoroti kekuatan dan area yang perlu diperbaiki.
22. Guru menutup sesi refleksi dengan memberikan wawasan tentang bagaimana materi hari ini akan dihubungkan dengan pelajaran berikutnya, sehingga siswa memahami arah pembelajaran ke depan.

Pertemuan 2: Menerapkan Variabel, Tipe Data, Dan Operator Untuk Pengolahan Data Sederhana

✓ MEMAHAMI dengan prinsip “berkesadaran”

1. Guru menyapa siswa dan mengkondisikan kelas untuk belajar
2. Guru mengajukan pertanyaan untuk memberi stimulasi kepada siswa:
 - o "Saat kalian menggunakan kalkulator di HP, bagaimana menurut kalian angka-angka tersebut disimpan dan diproses?"
 - o "Apa perbedaan antara angka dan teks di komputer?"

Contoh Skenario Kontekstual:

Guru menunjukkan bagaimana aplikasi seperti kalkulator digital dan form input di web menggunakan variabel untuk menyimpan dan memproses data pengguna.

3. Setiap kelompok **berdiskusi untuk membahas materi *variable*, tipe data, dan operator dari berbagai sumber** (w3schools, petanikode, Khan Academy, dan sebagainya) sesuai bahasa pemrograman yang dipilih. Poin-poin diskusi meliputi:
 - Mengulas dan memahami materi mengenai variabel, tipe data, dan operator sesuai bahasa pemrograman yang dipilih.
 - Membandingkan informasi tentang variabel, tipe data, dan operator dari berbagai sumber (misalnya: w3schools, PetaniKode, Khan Academy, dsb).
 - Bahas penerapan dan contoh kode sederhana yang menunjukkan penggunaan variabel, tipe data, dan operator dalam bahasa yang dipilih.
 - Saling mengajukan pertanyaan untuk mengklarifikasi konsep yang masih kurang dipahami.
4. Guru mendampingi siswa dalam berdiskusi, dan menguji pemahaman siswa dengan observasi dan menanyakan terkait konsep-konsep kunci dalam *variable*, tipe data, serta operator kepada masing-masing kelompok.

5. Guru menguatkan pemahaman siswa dan meluruskan kekeliruan pemahaman.
6. Guru memberikan contoh penggunaan tipe data dalam bahasa pemrograman
7. **Setiap kelompok diberi kartu bertuliskan tipe data (*Integer, Float, String, Boolean*).**
8. Siswa diminta **mengidentifikasi data dalam kehidupan nyata yang cocok dengan tipe data pada kartu yang diterima** (misalnya, "Usia = *Integer*", "Nama = *String*").
9. Siswa diminta **menyampaikan pendapat tentang apa yang terjadi bila tipe data yang salah digunakan pada suatu data.**
10. Siswa, dengan bimbingan guru, **melakukan percobaan menggunakan tipe data yang salah dalam pemrograman.** Kegiatan ini merupakan bagian dari asesmen proses dan dapat difasilitasi dengan **LKPD 2.1 “Menerapkan Variabel dan Tipe Data”** (<https://s.id/iTp3T>).
11. **Siswa mendeklarasikan variabel di bahasa pemrograman yang dipilih dalam dua tahap.** Tahap pertama, siswa hanya mendeklarasikan variabel saja. Tahap kedua, siswa memanfaatkan input pengguna untuk memberi nilai ke dalam variabel yang dideklarasikan, dan menampilkan nilainya menggunakan fungsi output. Kegiatan ini dapat difasilitasi dengan **LKPD 2.1 “Menerapkan Variabel dan Tipe Data”**

✓ **MENGAPLIKASI dengan prinsip “bermakna” dan “berkesadaran”**

12. Guru menampilkan **simulasi transaksi di e-commerce** yang menampilkan harga sebelum dan setelah diskon.
13. Guru bertanya kepada siswa:
 - "Bagaimana sistem di e-commerce bisa menghitung total harga belanja secara otomatis?"
 - "Bagaimana kita bisa memastikan bahwa diskon diterapkan dengan benar?"
14. Guru memberikan **contoh penggunaan operator untuk memanipulasi nilai variabel** dalam bahasa pemrograman

15. Siswa bekerja dalam kelompok kecil (3-4 orang) untuk menganalisis dan merancang perhitungan total harga setelah diskon. Kegiatan ini merupakan bagian dari asesmen proses dan dapat difasilitasi melalui **LKPD 2.2: Menghitung Harga Setelah Diskon** (<https://s.id/FDhfk>).

16. Guru membagikan skenario berbeda kepada setiap kelompok

- Skenario 1: Menghitung harga total untuk 1 barang dengan diskon tertentu.
- Skenario 2: Menghitung harga total untuk beberapa barang dengan harga berbeda, tanpa diskon.
- Skenario 3: Menghitung harga total dengan diskon yang dihitung dalam bentuk angka tetap, bukan persentase.
- Skenario 4: Menghitung harga total setelah menambahkan pajak (10%).

17. Setiap kelompok menuliskan *pseudocode* berdasarkan skenario mereka.

18. Setiap kelompok membandingkan solusi mereka dengan kelompok lain.

19. Siswa berdiskusi tentang keakuratan dan efisiensi perhitungan dalam algoritma yang mereka buat.

20. Guru memfasilitasi diskusi, menanyakan:

- o "Apakah hasil yang kalian dapatkan sesuai dengan harga yang dihitung secara manual?"
- o "Bagaimana cara memeriksa apakah operator yang digunakan sudah benar?"

21. Tiap kelompok **menerjemahkan *pseudocode* mereka ke dalam kode program yang dipilih** menggunakan C, Python, atau JavaScript.

22. Setelah selesai, **setiap kelompok bertukar kode untuk melakukan *code review*** dan mencari kesalahan perhitungan.

23. Setiap kelompok **mengomunikasikan solusi** mereka dan menjelaskan bagaimana mereka melakukan perhitungan.

24. Kelompok lain memberikan umpan balik, terutama terkait penggunaan variabel dan operator.

✓ MEREFLAKSI dengan prinsip “berkesadaran” dan “bermakna”

25. Setiap siswa menuliskan refleksi mendalam dalam jurnal belajar di LMS (kegiatan refleksi ini dapat difasilitasi dengan JURNAL REFLEKSI BELAJAR, terlampir di <https://s.id/jurnal-refleksi>), meliputi
- o Pemahaman Konsep:
 - o Tantangan & Kesulitan
 - o Strategi Perbaikan
 - o Aplikasi Nyata
 - o Setiap kelompok berbagi wawasan mereka mengenai pemecahan masalah yang telah mereka lakukan.
26. Siswa berdiskusi mengenai solusi terbaik yang mereka temukan, apakah ada cara yang lebih efisien, serta bagaimana konsep yang dipelajari dapat diterapkan dalam dunia nyata.
27. Guru memberikan umpan balik spesifik terkait hasil kerja siswa, menyoroti kekuatan dan area yang perlu diperbaiki.
28. Guru menutup sesi refleksi dengan memberikan wawasan tentang bagaimana materi hari ini akan dihubungkan dengan pelajaran berikutnya, sehingga siswa memahami arah pembelajaran ke depan.
29. Guru memberi tugas rumah:
- a. Mempelajari dan membuat catatan bermakna (bisa dalam bentuk *Cornell Notes*) tentang penggunaan indentasi (jika menggunakan bahasa pemrograman python)
 - b. Mempelajari dan membuat catatan bermakna tentang blok kode (jika menggunakan bahasa pemrograman C/JavaScript)
 - c. Mempelajari dan membuat catatan bermakna tentang *built-in functions*
 - d. Membuat daftar *built-in functions* yang telah digunakan, baik di kelas maupun di tempat lain, beserta kegunaannya

Pertemuan 3 : Menerapkan Struktur Kontrol (Conditions & Loops) Untuk Kasus Spesifik

✓ MEMAHAMI dengan prinsip “berkesadaran” dan “bermakna

1. Guru menyapa siswa dan mengkondisikan kelas untuk belajar
2. Guru memeriksa tugas rumah yang dikerjakan siswa, dan memberi umpan balik, penguatan pemahaman, dan mengaitkan dengan materi pada pertemuan hari ini.
3. Guru memulai dengan **pertanyaan reflektif**:
 - o "Bagaimana mesin ATM memutuskan apakah saldo cukup untuk transaksi?" (*Konsep percabangan: ATM harus memeriksa apakah saldo cukup sebelum mengizinkan penarikan.*)
 - o "Bagaimana sistem online bisa memastikan *password* yang dimasukkan benar?" (*Konsep percabangan: Sistem hanya mengizinkan akses jika password benar.*)
4. Guru menunjukkan **contoh pemrograman berbasis kondisi di aplikasi kehidupan nyata**:
 - o Sistem ATM: Memeriksa apakah saldo cukup (percabangan).
5. Setiap kelompok siswa membuat **pseudocode** atau **flowchart** keputusan **sederhana** seperti:
 - o Jika saldo mencukupi → transaksi berhasil.
 - o Jika saldo tidak cukup → tampilkan pesan error.
6. Guru mengajak siswa untuk **belajar melalui sumber belajar** tentang struktur kontrol kondisional (percabangan) dalam pemrograman berbasis teks, dengan **membuat jurnal membaca/catatan bermakna “Cornell Notes”** yang difasilitasi melalui *LMS* dengan bantuan **LKPD 3.1: “Cornell Notes Struktur Kontrol Kondisional”** (<https://s.id/bMP96>).
7. Guru mereviu *Cornell Notes* yang dibuat siswa dan memberikan umpan balik untuk menguatkan pemahaman

✓ MENGAPLIKASI dengan prinsip “berkesadaran” dan “bermakna”

8. Guru memberi kesempatan kepada siswa untuk **menuliskan algoritma pengecekan saldo rekening ke dalam bahasa pemrograman berbasis teks** sesuai bahasa pemrograman yang dipilih dengan bantuan LKPD 3.2: “Saldo Rekening” (<https://s.id/WxpTR>)

Keluaran program yang diharapkan:

Output

Masukkan jumlah penarikan: A
Input tidak valid. Harap masukkan angka.

Output

Masukkan jumlah penarikan: 50000
Transaksi Berhasil!

Output

Masukkan jumlah penarikan: -9000
Jumlah penarikan harus positif!

9. Guru memberikan *feedback* terhadap hasil penulisan program siswa, dan **mencontohkan kepada siswa bagaimana melakukan uji coba terhadap program** berdasarkan skenario.

✓ MEREFLAKSI dengan prinsip “berkesadaran” dan “bermakna”

10. Setiap siswa menuliskan refleksi mendalam dalam jurnal belajar di LMS (kegiatan refleksi ini dapat difasilitasi dengan **JURNAL REFLEKSI BELAJAR** terlampir di tautan <https://s.id/jurnal-refleksi>), meliputi
 - a. Pemahaman Konsep:
 - b. Tantangan & Kesulitan
 - c. Strategi Perbaikan
 - d. Aplikasi Nyata
 - e. Setiap kelompok berbagi wawasan mereka mengenai pemecahan masalah yang telah mereka lakukan.
11. Siswa berdiskusi mengenai solusi terbaik yang mereka temukan, apakah ada cara yang lebih efisien, serta bagaimana konsep yang dipelajari dapat diterapkan dalam dunia nyata.
12. Guru memberikan umpan balik spesifik terkait hasil kerja siswa, menyoroti kekuatan dan area yang perlu diperbaiki.
13. Guru menutup sesi refleksi dengan memberikan wawasan tentang bagaimana materi hari ini akan dihubungkan dengan pelajaran berikutnya, sehingga siswa memahami arah pembelajaran ke depan.
14. Guru memberikan tugas rumah:
 - Siswa mempelajari dan mempraktikkan 2 contoh struktur kondisi yang menggunakan *SWITCH...CASE*. Tugas ini dapat difasilitasi dengan menggunakan *LMS*.

Pertemuan 4 : Menerapkan Struktur Kontrol Perulangan (*Loops*) Untuk Kasus Spesifik

✓ MEMAHAMI dengan prinsip “berkesadaran” dan “bermakna

1. Guru menyapa siswa dan mengkondisikan kelas untuk belajar
2. Guru memeriksa tugas rumah yang dikerjakan siswa, dan memberi umpan balik, penguatan pemahaman, dan mengaitkan dengan materi pada pertemuan hari ini.
3. Guru memulai dengan **pertanyaan reflektif**:
 - "Bagaimana sistem *login* bisa memberi kesempatan lebih dari satu kali saat kita salah memasukkan password?" (*Konsep perulangan: Sistem akan meminta ulang input hingga pengguna memasukkan password yang benar atau mencapai batas percobaan.*)
 - "Bagaimana aplikasi ride-sharing seperti Gojek atau Grab dapat terus mencari pengemudi sampai ada yang menerima pesanan?" (*Konsep perulangan: Sistem akan mencari ulang jika pengemudi tidak tersedia.*)
4. Guru **menunjukkan contoh pemrograman berbasis kondisi dan perulangan di aplikasi kehidupan nyata**,
 - Sistem *login*: Mengizinkan pengguna mencoba memasukkan *password* hingga benar atau mencapai batas percobaan (perulangan).
 - Sistem pemesanan tiket: Memeriksa apakah kursi masih tersedia, jika tidak, meminta pengguna memilih tanggal lain (percabangan dan perulangan).
 - Sistem pembayaran digital: Memeriksa apakah transaksi berhasil, jika gagal, mencoba kembali atau meminta metode pembayaran lain (percabangan dan perulangan).
5. Eksperimen Kelompok:
 - Siswa diberikan salah satu dari skenario berikut ini:
 - "Bagaimana cara mengulang proses *login* hingga *password* benar?"

- "Bagaimana cara menghitung total harga beberapa barang tanpa mengetik satu per satu?"
 - Setiap kelompok **menuliskan *pseudocode* atau *flowchart***.
6. **Guru memberikan tanggapan** terhadap *pseudocode* atau *flowchart* dan menggunakannya untuk mengenalkan siswa kepada konsep perulangan (*loops*).
 7. Siswa **menyimpan *pseudocode* atau *flowchart* yang dibuat untuk kegiatan berikutnya**.
 8. Siswa **menyimak video di LMS tentang struktur perulangan**

a. **Kegiatan *pre-watching* (sebelum melihat video):**

i. Guru memberikan **pertanyaan pemantik**:

- "Bagaimana kita bisa membuat program yang dapat menjalankan perintah berulang tanpa menulis kode yang sama berkali-kali?"
- "Bagaimana mesin kasir bisa terus menerima *input* barang sampai kita menekan tombol selesai?"

- ii. Siswa menuliskan jawaban awal mereka di selembar kertas, setelah mendiskusikannya dengan teman sebangku.

b. **Kegiatan saat menonton video:**

Siswa **mencatat hal-hal yang membingungkan atau perlu ditanyakan**. Catatan ini digunakan untuk sesi diskusi setelah menyimak video.

c. **Kegiatan Setelah Menonton Video (*Post-Watching*)**

- i. Siswa **memvalidasi jawaban yang telah ditulis** sebelum menyimak video berdasarkan pengetahuan yang mereka dapat setelah menyimak video
- ii. Siswa **memperbaiki jawabannya**, dengan menyadur dari berbagai sumber belajar lain, selain video yang telah disimak

- iii. Siswa melakukan diskusi dalam kelompok kecil untuk **membahas hal-hal yang menjadi pertanyaan** pada saat menyimak video.
- iv. Guru **mendampingi diskusi** dan memberi **penguatan terhadap pendapat-pendapat** yang dikemukakan siswa.

Rekomendasi video:

- *for and while Loops* melalui kanal Neso Academy <https://s.id/m3QYR> .



- *Intro to Programming: Loops* melalui kanal CodeAcademy <https://s.id/8SAcA> .



- *Coding Basics: While Loops & Do While Loops | Programming for Beginners* melalui kanal Transcode. <https://s.id/LEFwY>



- *Computer Science Basics: Sequences, Selections, and Loops* melalui kanal LearnFree. <https://s.id/OZTPx>



✓ **MENGAPLIKASI** dengan prinsip “berkesadaran” dan “menggembirakan”

9. Guru **memberikan contoh implementasi suatu algoritma** yang menggunakan perulangan ke dalam bahasa pemrograman
Contoh algoritma & implementasi dalam pemrograman: “mengulang proses *login* hingga *password* benar”: <https://s.id/lqe9w> atau QR Code:



10. **Ice breaking:** Senam jari dengan bantuan video *Hands Gymnastics, Fingers Warm Up & Brain Break with Emoji* pada tautan <https://s.id/09Kjs> atau QR:



11. Siswa **menerapkan pemrograman dari pseudocode atau flowchart** yang sebelumnya telah dibuat ke dalam bahasa pemrograman yang dipilih. Kegiatan ini dapat difasilitasi melalui LMS dengan bantuan **LKPD 4.1: Total Harga Barang**: <https://s.id/1rpPbJWD> (asesmen proses).

12. Guru memberikan tugas rumah:

Membuat infografis/poster/*mind map* yang memvisualisasikan *NESTED IF* dan perbedaan antara perulangan *FOR*, *WHILE*, dan *DO..WHILE*. Tugas ini dapat difasilitasi melalui LMS

✓ MEREFLIKSI dengan prinsip “berkesadaran” dan “bermakna”

13. Setiap siswa menuliskan refleksi mendalam dalam jurnal belajar di LMS (kegiatan refleksi ini dapat difasilitasi dengan **JURNAL REFLEKSI BELAJAR** terlampir di tautan <https://s.id/jurnal-refleksi>), meliputi
 - o Pemahaman Konsep:
 - o Tantangan & Kesulitan
 - o Strategi Perbaikan
 - o Aplikasi Nyata
 - o Setiap kelompok berbagi wawasan mereka mengenai pemecahan masalah yang telah mereka lakukan.
14. Siswa berdiskusi mengenai solusi terbaik yang mereka temukan, apakah ada cara yang lebih efisien, serta bagaimana konsep yang dipelajari dapat diterapkan dalam dunia nyata.
15. Guru memberikan umpan balik spesifik terkait hasil kerja siswa, menyoroti kekuatan dan area yang perlu diperbaiki.
16. Guru menutup sesi refleksi dengan memberikan wawasan tentang bagaimana materi hari ini akan dihubungkan dengan pelajaran berikutnya, sehingga siswa memahami arah pembelajaran ke depan.

Pertemuan 5: Menerapkan Fungsi Untuk Meningkatkan Modularitas serta Efisiensi Kode Program

✓ MEMAHAMI dengan prinsip “menggembirakan” dan “bermakna”

1. Guru menyapa siswa dan mengkondisikan kelas untuk belajar
2. Guru memeriksa tugas rumah yang dikerjakan siswa, dan memberi umpan balik, penguatan pemahaman, dan mengaitkan dengan materi pada pertemuan hari ini.
3. Guru memulai diskusi dengan pertanyaan:
 - "Mengapa aplikasi seperti Gojek atau Google Maps bisa memberikan hasil dengan cepat, meskipun banyak data yang harus diproses?"
4. Guru menjelaskan bahwa fungsi dan modul eksternal membantu memecah kode menjadi bagian yang lebih kecil, lebih efisien, dan lebih mudah dipelihara.
 - Google Maps menggunakan berbagai fungsi untuk menghitung rute tercepat.
5. Siswa membuka kembali kode perhitungan harga belanja (dari pertemuan ke-2).
6. Guru kemudian memberikan contoh kode yang untuk kasus perhitungan harga belanja **tanpa menggunakan fungsi vs. menggunakan fungsi** untuk membandingkan modularitasnya.

Kode program menghitung harga total belanja **tanpa fungsi**

<https://s.id/lGJZl>



Kode program menghitung harga total belanja **dengan fungsi**

<https://s.id/X8yp3>



✓ **MENGAPLIKASI** dengan prinsip “menggembirakan” dan “berkesadaran”

7. **Siswa bermain “*Coding Estafet*”**. Kegiatan ini difasilitasi melalui **LKPD 5.1: “*Coding Estafet*”**: <https://s.id/IWGev> (asesmen proses)
8. Siswa dibagi menjadi kelompok dengan anggota 5 orang per kelompok dan menunjuk **beberapa siswa untuk menjadi TA (*teaching assistant*)** untuk masing-masing kelompok
9. Setiap kelompok diberi waktu **30 menit untuk berlatih implementasi fungsi dalam pemrograman**
10. Guru memberikan daftar 5 kode program melalui LMS yang diimplementasikan tanpa menggunakan fungsi untuk siswa pertama dalam setiap kelompok
11. Siswa yang mendapat daftar kode program, memilih salah satu untuk **diubah menjadi versi program yang memanfaatkan fungsi**.
12. Setelah siswa tersebut selesai **menulis dan menguji program sesuai skenario**, maka siswa **menugaskan siswa berikutnya secara estafet**, setiap siswa mendapat jatah 1 set kode program.
13. Guru dan TA membantu dan membimbing siswa yang kesulitan dalam mengerjakan
14. Kelompok yang menjadi pemenang dinilai dari ketepatan dan kecepatan dalam pemrograman,
15. Guru mereviu dan membahas solusi dari kumpulan soal dalam koding estafet
16. Guru mengajak seluruh siswa untuk saling mengucapkan terimakasih atas dukungan dan bantuannya, dan berpelukan/bersalaman (*social-emotional learning*)

✓ **MEREFLEKSI dengan prinsip “berkesadaran” dan “bermakna”**

17. **Guru menginformasikan rencana asesmen akhir** kepada siswa, meliputi bentuk, instrumen, teknik penilaian. dan informasi lain yang mendukung.
18. Guru memberi kesempatan kepada siswa untuk bertanya tentang asesmen akhir yang akan dilakukan.
19. Setiap siswa menuliskan refleksi mendalam dalam jurnal belajar di LMS (**kegiatan refleksi ini dapat difasilitasi dengan JURNAL REFLEKSI BELAJAR** terlampir di tautan <https://s.id/jurnal-refleksi>), meliputi
 - o Pemahaman Konsep:
 - o Tantangan & Kesulitan
 - o Strategi Perbaikan
 - o Aplikasi Nyata
 - o Setiap kelompok berbagi wawasan mereka mengenai pemecahan masalah yang telah mereka lakukan.
20. Siswa berdiskusi mengenai solusi terbaik yang mereka temukan, apakah ada cara yang lebih efisien, serta bagaimana konsep yang dipelajari dapat diterapkan dalam dunia nyata.
21. Guru memberikan umpan balik spesifik terkait hasil kerja siswa, menyoroti kekuatan dan area yang perlu diperbaiki.
22. Guru menutup sesi refleksi dengan memberikan wawasan tentang bagaimana materi hari ini akan dihubungkan dengan pelajaran berikutnya, sehingga siswa memahami arah pembelajaran ke depan.

Pertemuan 6: Menerapkan *Library*/Modul Eksternal Untuk Meningkatkan Modularitas serta Efisiensi Kode Program

✓ MEMAHAMI dengan prinsip “bermakna”

1. Guru menyapa siswa dan mengkondisikan kelas untuk belajar
2. Guru memulai diskusi dengan pertanyaan:
 - "Pernahkah kalian menggunakan aplikasi *e-commerce*? Bagaimana aplikasi tersebut dapat juga digunakan untuk melacak kiriman dari berbagai pilihan perusahaan ekspedisi?"
3. Guru menjelaskan bahwa modul eksternal membantu memecah kode menjadi bagian yang lebih kecil, lebih efisien, dan lebih mudah dipelihara.
 - Aplikasi *e-commerce* menggunakan modul eksternal sehingga dapat terhubung dengan sistem pelacakan kiriman dari perusahaan ekspedisi.
4. Guru **berkolaborasi dengan guru mata pelajaran matematika** untuk menjelaskan teknik perhitungan akar kuadrat dan nilai trigonometri
5. Siswa diberikan **studi kasus penerapan pemrograman: menghitung akar kuadrat dan nilai trigonometri (asesmen proses)**, yang lebih sulit dilakukan tanpa *library*. Studi kasus ini difasilitasi dengan **LKPD 6.1: Akar Kuadrat & Nilai Trigonometri** (<https://s.id/slrFR>).
6. Siswa mengungkapkan kesulitannya dalam mengimplementasikan kode **program** untuk menghitung akar kuadrat atau nilai trigonometri
7. Guru mengenalkan penggunaan ***library*/modul eksternal** sebagai solusi untuk pemrograman yang membutuhkan operasi kompleks
8. Guru menjelaskan mengapa kita menggunakan *library* eksternal, seperti *math.h* di C, *math* di Python, atau *Math* di JavaScript.
9. Guru memberikan solusi pemrograman menghitung akar kuadrat dan nilai trigonometri dengan menggunakan *library* eksternal
10. Guru **berkolaborasi dengan guru matematika** untuk menyusun skenario uji coba program menghitung akar kuadrat dan nilai trigonometri
11. Siswa membuat perbandingan dari contoh program yang menggunakan *library* eksternal dan tidak menggunakan *library* eksternal, dan menyimpulkan kelebihan penggunaan *library* eksternal.

12. **Guru menguatkan pemahaman** siswa dan menjawab pertanyaan siswa
13. **Siswa menyiapkan diri untuk asesmen akhir**

✓ MENGAPLIKASI dengan prinsip “bermakna” dan “berkesadaran”

14. Guru membagikan tautan **asesmen akhir** (test praktik): "*Bubble Tea Shop Simulator*" (<https://s.id/asesmen-akhir-tp6>). Asesmen ini dapat dikelola dengan bantuan LMS.
15. Siswa untuk berdoa sebelum asesmen dilaksanakan
16. Siswa mengerjakan asesmen akhir dengan seksama
17. Selesai asesmen, siswa mengucapkan syukur dan terimakasih kepada diri sendiri
18. Guru mengucapkan terimakasih kepada seluruh siswa yang telah berjuang menyelesaikan topik menerapkan pemrograman berbasis teks.
19. Guru mengajak siswa meneriakkan jargon kelas.

✓ MEREFLIKSI dengan prinsip “berkesadaran” dan “bermakna”

20. Setiap siswa menuliskan refleksi mendalam dalam jurnal belajar di LMS (**kegiatan refleksi ini dapat difasilitasi dengan JURNAL REFLEKSI BELAJAR** terlampir di tautan <https://s.id/jurnal-refleksi>), meliputi
 - o Pemahaman Konsep:
 - o Tantangan & Kesulitan
 - o Strategi Perbaikan
 - o Aplikasi Nyata
 - o Setiap kelompok berbagi wawasan mereka mengenai pemecahan masalah yang telah mereka lakukan.
21. Siswa berdiskusi mengenai solusi terbaik yang mereka temukan, apakah ada cara yang lebih efisien, serta bagaimana konsep yang dipelajari dapat diterapkan dalam dunia nyata.

22. Guru memberikan umpan balik spesifik terkait hasil kerja siswa, menyoroti kekuatan dan area yang perlu diperbaiki.
23. Guru menutup sesi refleksi dengan memberikan wawasan tentang bagaimana materi hari ini akan dihubungkan dengan pelajaran berikutnya, sehingga siswa memahami arah pembelajaran ke depan.

ASESMEN PEMBELAJARAN

PERTEMUAN 1

1. ASESMEN AWAL: *Jurnal Reflektif*

Asesmen ini dilaksanakan di awal pertemuan 1. Siswa menulis jurnal reflektif dengan panduan pertanyaan terbuka yang mengaitkan materi sebelumnya (kompleksitas algoritma) dengan materi hari ini (implementasi algoritma), dan menguji kesiapan siswa untuk mengimplementasikan algoritma ke dalam pemrograman berbasis teks.

Instrumen Jurnal Reflektif

Instruksi untuk Siswa:

Jawablah pertanyaan berikut dengan jujur dan mendalam. Gunakan contoh konkret dari tugas/praktikum yang pernah kamu kerjakan.

1. Kompleksitas Algoritma:

- o Jelaskan dengan kata-katamu sendiri: Apa itu kompleksitas waktu dalam algoritma? Mengapa hal ini penting?

2. Implementasi Algoritma:

- o Apakah kamu pernah mencoba menulis kode program untuk suatu algoritma? Jika pernah, maka:
 1. Ceritakan pengalamanmu saat menulis dan mencoba kode program untuk algoritma tertentu.
 2. Bagaimana cara kamu memastikan kode yang kamu tulis sesuai dengan algoritma yang direncanakan?
 3. Apakah kamu pernah mengalami situasi di mana algoritma yang kamu pahami, ternyata sulit diimplementasikan dalam kode program? Jelaskan!
 4. Bagaimana pemahamanmu tentang kompleksitas algoritma membantumu dalam memahami makna kode program yang pernah kamu coba?

- o Jika belum pernah menulis kode program, maka:
 1. Ceritakan dengan singkat apa saja yang kamu ketahui tentang bahasa pemrograman!
 2. Apakah, menurut Anda, kemampuan menuliskan algoritma dalam bentuk pseudocode atau flowchart akan memudahkan Anda mengimplementasikannya dalam kode program nantinya? Jelaskan alasannya!

Rubrik Penilaian Jurnal Reflektif

Kriteria	Skor 4 (Unggul)	Skor 3 (Baik)	Skor 2 (Cukup)	Skor 1 (Kurang)
Pemahaman Konsep Kompleksitas Algoritma	Menjelaskan konsep kompleksitas waktu dengan sangat jelas, tepat, dan mendalam; mengaitkan dengan konteks praktikum secara konkret.	Menjelaskan konsep dengan cukup jelas dan tepat; mengaitkan dengan konteks praktikum, meskipun refleksi kurang mendalam.	Menjelaskan konsep dengan beberapa kekurangan atau kesalahan; hubungan dengan praktikum masih minim.	Penjelasan kurang jelas, banyak kesalahan, dan tidak relevan dengan konteks praktikum.
Implementasi Algoritma / Pengalaman Praktis	Menyajikan contoh pengalaman nyata saat menulis kode (atau pemahaman dasar tentang bahasa pemrograman) secara rinci,	Menyajikan contoh atau pengalaman yang relevan, namun detail dan refleksi belum optimal; terdapat upaya untuk menguraikan langkah verifikasi.	Menyajikan contoh yang umum atau terbatas, dengan penjelasan minim tentang proses verifikasi atau tantangan implementasi.	Tidak memberikan contoh nyata atau penjelasan pengalaman; refleksi sangat minim atau tidak ada.

	termasuk langkah-langkah verifikasi dan tantangan yang dihadapi.			
Keterpaduan Argumen dan Refleksi	Argumen sangat terintegrasi dengan baik, mencerminkan refleksi mendalam dan kritis terhadap pembelajaran serta menghubungkan teori dengan praktik secara menyeluruh.	Argumen cukup terintegrasi dengan refleksi yang memadai; mengaitkan teori dan praktik meskipun masih terdapat beberapa bagian yang kurang mendalam.	Argumen dan refleksi kurang konsisten; penghubung antara teori dan praktik masih belum jelas atau minim.	Argumen tidak terintegrasi, refleksi dangkal, dan tidak mengaitkan teori dengan praktik.
Struktur, Keterbacaan, dan Penulisan	Tulisan sangat terstruktur, koheren, dan mudah dipahami; penggunaan bahasa sangat tepat dan tata tulis rapi.	Tulisan terstruktur dengan baik dan cukup koheren; bahasa umumnya tepat dengan sedikit kekurangan pada tata tulis.	Tulisan kurang terstruktur, ide-ide tidak tersampaikan dengan jelas, serta terdapat beberapa kesalahan tata tulis.	Tulisan tidak terstruktur, ide tidak jelas, dengan banyak kesalahan dalam penggunaan bahasa dan tata tulis.

Pedoman Penskoran

- a. **Setiap kriteria** diberikan skor antara 1 hingga 4 sesuai dengan deskripsi di atas.
- b. **Total Skor Maksimum:** 16 (4 kriteria x 4 poin).
- c. **Interpretasi Total Skor:**
 - **14 – 16 (Sangat Baik):**
Siswa menunjukkan pemahaman mendalam dan refleksi kritis. Jawaban terintegrasi dengan baik antara konsep, pengalaman, dan argumen.
 - **10 – 13 (Baik):**
Siswa memahami konsep secara baik, namun masih ada ruang untuk pendalaman pada beberapa aspek, terutama dalam integrasi antara teori dan praktik.
 - **6 – 9 (Cukup):**
Siswa menunjukkan pemahaman dasar namun penjelasan dan refleksi masih terbatas. Diperlukan bimbingan untuk memperdalam integrasi konsep dan pengalaman.
 - **1 – 5 (Kurang):**
Siswa belum memahami konsep secara memadai; jawaban minim refleksi dan kurang mengaitkan pengalaman atau teori secara nyata.

2. ASESMEN PROSES 1: LKPD 1.1

Asesmen ini dilakukan pada pertemuan 1, sebagai penilaian terhadap kegiatan siswa melalui **LKPD 1.1 - Membandingkan Algoritma Perjalanan Menuju ke Sekolah**. Asesmen ini menguatkan pemahaman mendasar tentang perbandingan algoritma, agar siswa lebih siap dalam menerapkan pemrograman berbasis teks menggunakan algoritma yang sudah teruji.

Rubrik Perbandingan Algoritma

Kriteria	4 (Sangat Baik)	3 (Baik)	2 (Cukup)	1 (Kurang)
Metode yang Digunakan	Algoritma menggunakan metode yang jelas (misalnya, waktu tempuh, jarak terpendek, jumlah lampu merah) dan sesuai dengan konteks permasalahan.	Algoritma menggunakan metode yang relevan tetapi masih kurang optimal dalam implementasi.	Algoritma menggunakan metode tetapi kurang jelas atau tidak sepenuhnya sesuai dengan permasalahan.	Algoritma tidak menggunakan metode yang jelas atau tidak sesuai dengan tujuan perbandingan.
Faktor yang Dipertimbangkan	Algoritma mempertimbangkan banyak faktor (kemacetan, jumlah persimpangan, kondisi jalan, dll.) dengan penjelasan yang kuat.	Algoritma mempertimbangkan beberapa faktor tetapi ada beberapa faktor penting yang diabaikan.	Algoritma hanya mempertimbangkan satu atau dua faktor dan tidak menjelaskan relevansinya dengan baik.	Tidak ada faktor yang dipertimbangkan secara jelas atau hanya disebutkan tanpa pertimbangan matang.
Kompleksitas Algoritma	Kompleksitas dijelaskan dengan baik, menggunakan konsep yang relevan (misal: $O(n)$, $O(n^2)$) dan sesuai dengan tingkat	Kompleksitas dijelaskan tetapi masih kurang tepat atau tidak mendalam.	Kompleksitas disebutkan tetapi kurang dipahami atau tidak dikaitkan dengan implementasi.	Tidak menjelaskan kompleksitas atau tidak menunjukkan pemahaman terhadap konsep

	pembelajaran siswa.			perbandingan algoritma.
Keakuratan Perkiraan Waktu Tempuh	Algoritma memberikan perkiraan waktu yang realistis, berdasarkan data atau logika yang kuat.	Perkiraan waktu cukup realistis tetapi masih ada beberapa aspek yang kurang diperhitungkan.	Perkiraan waktu tidak cukup akurat dan masih memiliki banyak asumsi yang tidak jelas.	Tidak ada perkiraan waktu atau perkiraan tidak masuk akal berdasarkan faktor yang dipertimbangkan.
Kelebihan Algoritma	Keunggulan algoritma dijelaskan dengan rinci, menunjukkan pemahaman yang kuat tentang kelebihan dan kekuatan metode yang dipilih.	Keunggulan disebutkan dengan baik tetapi tidak semua aspek didukung dengan penjelasan yang kuat.	Keunggulan disebutkan tetapi tidak dijelaskan secara jelas atau tidak relevan dengan metode yang digunakan.	Tidak menyebutkan atau menjelaskan keunggulan algoritma dengan baik.
Kekurangan Algoritma	Kekurangan algoritma diidentifikasi dengan jelas dan disertai refleksi tentang bagaimana mengatasinya.	Kekurangan disebutkan dengan baik tetapi tanpa solusi atau perbaikan yang jelas.	Kekurangan disebutkan tetapi tidak dijelaskan atau tidak relevan dengan konteks masalah.	Tidak menyebutkan atau menganalisis kekurangan algoritma yang dibuat.

Kolaborasi dalam Kelompok	Setiap anggota berkontribusi secara aktif, berdiskusi dengan baik, dan saling mendukung dalam pembuatan algoritma serta pengisian matriks perbandingan.	Anggota kelompok bekerja sama dengan baik, tetapi ada anggota yang kurang aktif dalam diskusi.	Kolaborasi kurang efektif, hanya beberapa anggota yang aktif berpartisipasi.	Kelompok tidak bekerja sama dengan baik atau hanya satu atau dua anggota yang berperan aktif.
---------------------------	---	--	--	---

Pedoman Penskoran

a. Total Skor Maksimum: 28

b. Interpretasi Skor:

- o 24 – 28 → Sangat Baik (Siswa memahami konsep algoritma dengan sangat baik dan mampu menganalisis serta membandingkan solusi secara efektif.)
- o 18 – 23 → Baik (Siswa memiliki pemahaman yang cukup kuat tetapi masih perlu meningkatkan analisis dan justifikasi mereka.)
- o 12 – 17 → Cukup (Siswa memahami konsep dasar tetapi masih kesulitan dalam penerapan dan perbandingan.)
- o <12 → Kurang (Siswa masih memerlukan banyak bimbingan dalam memahami dan membandingkan algoritma.)

3. ASESMEN PROSES 2: LKPD 1.2

Rubrik Asesmen: Kemampuan Menerapkan Input, Output, dan Komentar dalam Pemrograman

Tujuan: Mengukur kemampuan siswa dalam menerapkan dan memodifikasi input/output, membuat output multi-line, dan menggunakan komentar dalam kode program.

Kriteria Penilaian

Berikut adalah kriteria dan skala penilaian (1-4):

4 = Sangat Baik

3 = Baik

2 = Cukup

1 = Perlu Perbaikan

Kriteria	Deskripsi	Skor
Modifikasi Input/Output	Siswa mengganti data input dan output sesuai contoh dengan sintaks yang benar.	1-4
Output Multi-line	Siswa membuat output yang terdiri dari beberapa baris dengan format yang tepat dan logis.	1-4
Penggunaan Komentar	Komentar jelas, relevan, dan menandai bagian input/output dengan tepat.	1-4
Kreativitas Modifikasi	Siswa menunjukkan inisiatif dalam memodifikasi data (misal: menambahkan variasi atau kasus baru).	1-4

Pedoman Penilaian

a. Modifikasi *Input/Output*

- 4: Semua data *input/output* diubah sesuai contoh, sintaks benar, dan program berjalan tanpa *error*.
- 3: Data *input/output* diubah, tetapi terdapat 1 kesalahan sintaks *minor* yang tidak mengganggu eksekusi.

- 2: Data *input/output* diubah sebagian, tetapi terdapat kesalahan sintaks yang menyebabkan *output* tidak sesuai.
- 1: Tidak ada modifikasi *input/output* atau modifikasi menyebabkan program gagal berjalan.

b. Output *Multi-line*

- 4: Output *multi-line* dibuat dengan struktur rapi (contoh: tabel, paragraf) dan logis.
- 3: Output *multi-line* dibuat, tetapi format kurang konsisten (misal: spasi tidak seragam).
- 2: Output hanya terdiri dari 2 baris atau format tidak jelas.
- 1: Tidak ada upaya membuat output *multi-line*.

c. Penggunaan Komentar

- 4: Komentar lengkap, menjelaskan fungsi *input/output*, dan mudah dipahami.
- 3: Komentar ada tetapi kurang detail atau tidak menandai semua bagian *input/output*.
- 2: Komentar hanya ada di satu bagian (*input* atau *output*) atau terlalu ambigu.
- 1: Tidak ada komentar atau komentar tidak relevan.

d. Kreativitas Modifikasi

- 4: Modifikasi data unik (misal: menambahkan validasi tambahan atau kasus khusus).
- 3: Modifikasi data bervariasi tetapi masih dalam batas contoh guru.
- 2: Modifikasi minimal (hanya mengganti angka/huruf tanpa variasi).
- 1: Tidak ada kreativitas dalam modifikasi.

Total Skor & Interpretasi

- 13-16: Sangat Baik (Siswa menguasai konsep dengan kreativitas tinggi).
- 9-12: Baik (Siswa memahami konsep tetapi perlu peningkatan pada detail).
- 5-8: Cukup (Siswa perlu latihan lebih untuk memperbaiki kesalahan).
- 1-4: Perlu Perbaikan (Siswa belum memahami konsep dasar).

Catatan untuk guru:

- Berikan contoh komentar yang baik, misal:

```
# INPUT: Masukkan saldo pengguna
saldo = int(input("Masukkan saldo: "))

# OUTPUT: Tampilkan sisa saldo
print("Saldo tersisa:", saldo)
```
- Diskusikan kesalahan umum (misal: typo, logika percabangan) selama penilaian.

3. ASESMEN PROSES 3: LKPD 1.3

Rubrik Asesmen Proses: Membuat Catatan Bermakna tentang Format *Output*, Jenis *Error*, dan Penanganannya

Tujuan: Mengukur pemahaman siswa dalam mengeksplorasi format *output*, jenis-jenis *error*, dan cara menangani *error* melalui catatan bermakna berbasis *Cornell Notes*.

Kriteria Penilaian

Berikut adalah kriteria dan skala penilaian (1-4):

4 = Sangat Baik

3 = Baik

2 = Cukup

1 = Perlu Perbaikan

Kriteria	Deskripsi	Skor
Kelengkapan Konten	Catatan mencakup format output , jenis error , dan penanganan error secara lengkap.	1-4
Akurasi Informasi	Definisi, contoh, dan penjelasan sesuai dengan sumber belajar (tidak ada kesalahan konsep).	1-4
Struktur Cornell Notes	Catatan mengikuti format Cornell Notes: kolom pertanyaan/kata kunci, catatan utama, dan ringkasan.	1-4
Kedalaman Pemahaman	Ringkasan mencerminkan pemahaman pribadi siswa, bukan sekadar salinan sumber.	1-4
Contoh dan Aplikasi	Siswa menyertakan contoh konkret (kode program/ilustrasi) untuk setiap topik.	1-4

Pedoman Penilaian

a. Kelengkapan Konten

- 4: Semua topik (format output, jenis error, penanganan error) dibahas secara rinci.
- 3: Topik dibahas, tetapi salah satu sub-topik kurang lengkap.
- 2: Dua topik utama terlewat atau sangat minim penjelasan.

- 1: Hanya satu topik yang tercakup atau informasi sangat tidak lengkap.

b. Akurasi Informasi

- 4: Semua definisi, contoh, dan penanganan error akurat sesuai referensi.
- 3: Terdapat 1-2 kesalahan minor (misal: contoh kurang spesifik).
- 2: Kesalahan konsep pada 1-2 poin (misal: jenis error tertukar).
- 1: Banyak kesalahan konsep atau contoh tidak relevan.

c. Struktur Cornell Notes

- 4: Format Cornell Notes diikuti dengan sempurna: kolom kiri berisi pertanyaan/kata kunci, kolom kanan berisi catatan, dan ringkasan di bawah.
- 3: Format diikuti, tetapi salah satu bagian kurang rapi (misal: ringkasan terlalu singkat).
- 2: Hanya dua bagian yang terisi atau format tidak konsisten.
- 1: Tidak mengikuti format Cornell Notes sama sekali.

d. Kedalaman Pemahaman

- 4: Ringkasan mencakup refleksi pribadi, aplikasi praktis, atau pertanyaan lanjutan.
- 3: Ringkasan menjelaskan poin utama tetapi kurang mencerminkan pemikiran siswa.
- 2: Ringkasan hanya menyalin ulang catatan utama tanpa analisis.
- 1: Tidak ada ringkasan atau rangkuman tidak bermakna.

e. Contoh dan Aplikasi

- 4: Contoh diberikan untuk semua topik dan relevan dengan bahasa pemrograman yang dipilih.

- 3: Contoh ada tetapi kurang detail atau hanya untuk 2 topik.
- 2: Contoh hanya untuk satu topik atau terlalu umum.
- 1: Tidak ada contoh atau contoh tidak sesuai.

Total Skor & Interpretasi

- 17-20: Sangat Baik (Pemahaman mendalam, catatan terstruktur, dan contoh konkret).
- 13-16: Baik (Pemahaman baik, tetapi perlu peningkatan pada kedalaman atau contoh).
- 9-12: Cukup (Pemahaman dasar ada, tetapi banyak aspek perlu diperbaiki).
- 5-8: Perlu Perbaikan (Catatan tidak lengkap/akurat dan struktur tidak sesuai).

PERTEMUAN 2

1. ASESMEN PROSES 1: PEMAHAMAN KONSEP VARIABEL, TIPE DATA, DAN OPERATOR

Asesmen ini dilaksanakan oleh guru dengan teknik observasi dan tanya jawab saat siswa melakukan diskusi kelompok di pertemuan ke-2, yang mendiskusikan tentang variabel, tipe data, dan operator

Lembar Observasi Guru

Mata Pelajaran: Koding dan Kecerdasan Artifisial

Materi: Variabel, Tipe Data

Nama Kelompok: _____

Tanggal: _____

Nama Pengamat: _____

Instruksi Penggunaan Lembar Observasi:

1. Beri tanda (✓) pada kolom "Ya" jika kelompok menunjukkan indikator yang sesuai.
2. Beri tanda (✓) pada kolom "Perlu Bimbingan" jika kelompok masih membutuhkan arahan dalam indikator tertentu.
3. Berikan catatan tambahan untuk aspek yang memerlukan penguatan atau umpan balik lebih lanjut.

Lembar Observasi

Indikator	Ya (✓)	Perlu Bimbingan (✓)	Catatan Guru
1. Pemahaman Konsep Variabel, Tipe Data,			

Operator – Kelompok dapat menjelaskan konsep variabel dan tipe data dengan jelas serta memberikan contoh konkret dari kehidupan nyata.			
2. Partisipasi dalam Diskusi – Semua anggota aktif berdiskusi, mengajukan pertanyaan, dan memberikan pendapat terkait materi.			
3. Kemampuan Menghubungkan dengan Kehidupan Nyata – Kelompok dapat mengidentifikasi contoh nyata yang sesuai dengan tipe data yang diberikan.			
4. Analisis Kesalahan Tipe Data – Kelompok dapat menjelaskan akibat kesalahan tipe data, memberikan contoh, serta mengusulkan solusi yang tepat.			
5. Eksperimen dan Implementasi Kode – Kelompok mencoba menjalankan kode, mengamati hasilnya, serta menjelaskan perbedaan antara kode yang benar dan yang salah.			

Catatan Observasi & Rekomendasi Guru:

(Guru mencatat temuan khusus, kesulitan yang dihadapi siswa, serta langkah tindak lanjut yang perlu dilakukan pada pertemuan berikutnya.)

Pedoman Penggunaan Hasil Observasi:

- ✓ Jika semua indikator dicentang “Ya”, berarti kelompok memiliki pemahaman yang kuat dan mampu bekerja secara mandiri.
- ✓ Jika ada indikator di kolom "Perlu Bimbingan", maka materi tersebut perlu diperjelas kembali dalam sesi tanya jawab atau diskusi tambahan.
- ✓ Catatan guru dapat digunakan untuk menyusun umpan balik spesifik kepada siswa dan menentukan strategi pembelajaran lanjutan.

2. ASESMEN PROSES 2: PENERAPAN VARIABEL DAN TIPE DATA (LKPD 2.1)

Asesmen ini menguji kemampuan siswa dalam menerapkan variabel dan tipe data melalui dua tahap:

- Tahap 1: menguji kode program dengan skenario *test case* untuk memastikan program yang dibuat siswa berjalan dengan benar
- Tahap 2: menguji kemampuan siswa dalam menerapkan variabel dan tipe data dengan menggunakan rubrik penilaian

Skenario Test Case:

Deskripsi	Input	Output yang Diharapkan	Tujuan
Input Valid	Usia: 20 Tinggi: 1.75 Nama: Budi	Semua Bahasa: Usia: 20 tahun Tinggi: 1.75 meter Nama: Budi	Memastikan program membaca input sesuai tipe data dan menampilkan output.
Input Desimal untuk Usia	Usia: 20.5 Tinggi: 1.68 Nama: Rina	Python: Error ValueError JS: Usia = 20 C: Error input	Mengevaluasi penanganan tipe data tidak sesuai.
Nama dengan Spasi	Usia: 25 Tinggi: 1.80 Nama: Budi Santoso	C: Nama = Budi Python/JS: Nama = Budi Santoso	Memeriksa pembacaan string dengan spasi.
Input Nilai Negatif	Usia: -5 Tinggi: -1.60 Nama: Ali	Semua Bahasa: Output sesuai input tanpa validasi (misal: Usia: -5 tahun).	Memastikan program tidak melakukan validasi nilai negatif.

Instrumen Asesmen: Rubrik Penilaian Penerapan Variabel dan Tipe Data

Kriteria Penilaian

Kriteria	4 (Sangat Baik)	3 (Baik)	2 (Cukup)	1 (Perlu Perbaikan)
Deklarasi Variabel	Semua variabel dideklarasikan dengan tipe data yang tepat sesuai bahasa pemrograman.	Ada 1 kesalahan tipe data.	Ada 2 kesalahan tipe data.	Deklarasi tidak sesuai.
Penanganan Input	Input di-convert ke tipe data yang benar (misal: <code>int(input(...))</code> di Python).	Ada 1 kesalahan konversi (misal: lupa <code>parseInt</code>).	<i>Input</i> tidak di-convert sesuai tipe data.	<i>Input</i> tidak relevan atau tidak terbaca.
Format Output	<i>Output</i> menampilkan semua data dengan format jelas dan sesuai ekspektasi.	Ada 1 kesalahan <i>minor</i> pada format (misal: desimal).	<i>Output</i> tidak lengkap atau format kacau.	<i>Output</i> tidak sesuai <i>input</i> .
Kesesuaian Sintaks	Kode bebas <i>error</i> dan mengikuti praktik terbaik bahasa pemrograman.	Ada 1 kesalahan sintaks <i>minor</i> (misal: <i>warning</i> di C).	Beberapa kesalahan sintaks.	Kode tidak dapat dijalankan.

Pedoman Penilaian

- a. Deklarasi Variabel:
 - 4: Contoh di C: `int usia; float tinggi; char nama[50];`.
 - 1: Deklarasi tidak sesuai (misal: `var usia;` di C).
- b. Penanganan Input:
 - 4: Menggunakan `scanf("%d", &usia)` di C, `int(input(...))` di Python.
 - 2: Tidak ada konversi tipe data (misal: `usia = input(...)` di Python tanpa `int`).
- c. Format Output:
 - 4: Output lengkap dengan format spesifik (contoh: `printf("Usia: %d tahun", usia);`).
 - 1: Hanya 1 variabel yang ditampilkan.
- d. Kesesuaian Sintaks:
 - 4: Kode berjalan tanpa error.
 - 3: Ada warning di C tetapi program tetap berjalan.

Total Skor & Interpretasi

Total Skor	Kategori	Deskripsi
13-16	Sangat Baik	Kode fungsional, sintaks rapi, dan output sesuai semua test case.
9-12	Baik	Ada minor error (misal: format desimal), tetapi logika benar.
5-8	Cukup	Kode berjalan dengan kesalahan konsep (misal: tipe data tidak sesuai).
1-4	Perlu Perbaikan	Kode tidak selesai/tidak sesuai instruksi.

Contoh Umpan Balik

- Positif:
"Deklarasi variabel tepat, dan konversi input sesuai tipe data. Output rapi!"
- Perbaikan:
"Perbaiki penanganan input string dengan spasi di C menggunakan fgets()."

Catatan: Siswa dengan skor rendah disarankan mempelajari konversi tipe data dan praktik input/output dasar.

3. ASESMEN PROSES 3: PENERAPAN OPERATOR (LKPD 2.2)

Asesmen ini digunakan untuk mengetahui kemampuan siswa dalam menerapkan operator aritmatika untuk kasus perhitungan harga setelah diskon yang terdapat pada LKPD 2.2: Menerapkan Operator. Kemampuan siswa dalam menerapkan jenis operator yang lain, akan dinilai pada pertemuan-pertemuan berikutnya, terintegrasi dalam struktur kontrol, fungsi, dan modul eksternal.

Rubrik Penilaian LKPD 2.2

Kriteria	4 (Sangat Baik)	3 (Baik)	2 (Cukup)	1 (Perlu Perbaikan)
Deklarasi Variabel	Semua variabel dideklarasikan dengan tipe data yang tepat.	Ada 1 kesalahan tipe data.	Ada 2 kesalahan tipe data.	Deklarasi tidak sesuai.

Operator Aritmetika	Perhitungan total, diskon, dan pajak benar dengan operator yang tepat.	Ada 1 kesalahan <i>minor</i> .	Ada kesalahan logika perhitungan.	Perhitungan salah total.
Format Output	Output menampilkan hasil sesuai format (termasuk desimal jika diperlukan).	Ada 1 kesalahan format (misal: desimal).	Format tidak lengkap/kacau.	Output tidak sesuai.
Kesesuaian Sintaks	Kode bebas error dan sesuai dengan sintaks bahasa pemrograman.	Ada 1 <i>warning</i> (C) atau kesalahan <i>minor</i> .	Beberapa kesalahan sintaks.	Kode tidak dapat dijalankan.

Interpretasi Skor:

- 13-16: Sangat Baik (Kode rapi, logika benar, dan output sesuai).
- 9-12: Baik (Ada *minor error*, tetapi logika benar).
- 5-8: Cukup (Kesalahan konsep atau sintaks).
- 1-4: Perlu Perbaikan (Kode tidak selesai/tidak sesuai instruksi).

Catatan:

- Pastikan diskon 5% dan pajak 10% menggunakan tipe data *float* (C/Python) atau *number* (JavaScript).
- Output untuk kasus desimal harus menampilkan dua angka di belakang koma.

PERTEMUAN 3

1. ASESMEN PROSES 1 - Penilaian Terhadap Catatan Bermakna/Cornell Notes Struktur Kondisional

Asesmen ini membantu guru untuk menilai progres belajar siswa yang dilakukan dengan membuat catatan bermakna dalam bentuk *Cornell Notes*

Rubrik Penilaian Cornell Notes

Aspek Penilaian	4 (Sangat Baik)	3 (Baik)	2 (Cukup)	1 (Kurang)
Struktur dan Format	Catatan mengikuti format <i>Cornell</i> dengan jelas (kolom <i>Cue</i> , Area Catatan, dan Ringkasan).- Tampilan tertata, mudah dibaca, dan rapi.	Struktur <i>Cornell</i> sudah ada, namun ada sedikit ketidakteraturan di salah satu bagian (misalnya, ringkasan kurang lengkap atau kolom <i>cue</i> kurang jelas).	Format <i>Cornell</i> sudah diterapkan, tetapi beberapa bagian tidak terpisah dengan baik atau tidak konsisten.	Tidak mengikuti format <i>Cornell</i> atau tampilan catatan sangat berantakan, sehingga sulit dipahami.
Kelengkapan Materi	Semua aspek penting <i>if..else</i> tercatat: definisi, sintaks, alur kerja, contoh implementasi, dan aplikasi nyata.- Mengidentifikasi kelebihan dan	Sebagian besar aspek tercakup, tetapi ada 1-2 poin penting yang kurang dijelaskan secara mendetail.	Hanya mencakup sebagian konsep (misalnya, hanya definisi dan contoh singkat) tanpa penjelasan mendalam.	Materi yang dicatat tidak lengkap atau hanya menyalin sedikit definisi tanpa penjelasan tambahan.

	kekurangan secara komprehensif.			
Kedalaman Analisis	Menyertakan analisis mendalam tentang penggunaan <i>if..else</i> , termasuk contoh situasi nyata dan refleksi tentang relevansi dalam pemrograman.	Menyediakan analisis, namun tidak terlalu mendalam; beberapa contoh situasi nyata sudah ada namun kurang didukung refleksi pribadi.	Menyebutkan analisis secara singkat tetapi kurang menghubungkan dengan aplikasi nyata atau tidak ada refleksi pribadi.	Hanya mencatat definisi tanpa analisis atau refleksi yang menunjukkan pemahaman mendalam.
Integrasi Sumber dan Referensi	Catatan mencantumkan referensi (misalnya, <i>Cornell Notes</i> dan sumber terkait <i>if..else</i>) dengan jelas, serta mengintegrasikan informasi dari sumber secara efektif.	Mencantumkan referensi, namun integrasinya kurang konsisten atau kurang menekankan poin-poin kunci dari sumber.	Referensi disebutkan, tetapi tidak jelas bagaimana sumber tersebut memengaruhi isi catatan.	Tidak mencantumkan referensi atau catatan tidak menunjukkan keterkaitan dengan sumber belajar yang telah diberikan.
Refleksi Pribadi dan Pertanyaan	Menyediakan ruang refleksi yang mendalam, dengan pertanyaan reflektif terkait	Ada ruang refleksi, namun jawabannya cenderung umum atau tidak disertai contoh konkrit.	Refleksi pribadi terbatas; siswa hanya menuliskan pernyataan singkat tanpa	Tidak ada bagian refleksi atau pertanyaan yang mengaitkan dengan

	pengalaman pribadi dan aplikasi <i>if..else</i> dalam situasi nyata.- Menyertakan contoh konkret dari pengalaman atau tantangan.		analisis mendalam.	pengalaman pribadi.
--	--	--	--------------------	---------------------

Pedoman Penskoran

- a. Total Skor Maksimum: 20 poin (dari 5 aspek, masing-masing maksimal 4 poin).
- b. Konversi Skor ke Skala 0-100:
 - o Nilai Akhir = $(\text{Skor Perolehan} / 20) \times 100$
 - o Misalnya, jika total skor siswa adalah 16, maka nilai akhir = $(16/20) \times 100 = 80$.
- c. Interpretasi Hasil:
 - o 17 – 20 poin (85-100): Siswa memiliki pemahaman sangat mendalam dan mampu mengintegrasikan informasi dengan baik; catatan sangat terstruktur dan reflektif.
 - o 13 – 16 poin (65-84): Siswa menunjukkan pemahaman yang baik tetapi masih perlu pendalaman di beberapa aspek; catatan cukup rapi dan analisis cukup mendalam.
 - o 9 – 12 poin (45-64): Siswa memiliki pemahaman dasar, tetapi catatan kurang mendalam atau terstruktur; memerlukan bimbingan tambahan.
 - o < 9 poin (<45): Siswa masih kesulitan memahami konsep *if..else*; catatan tidak lengkap atau sangat tidak terstruktur.

Catatan untuk Guru:

- Observasi: Saat siswa menyusun *Cornell Notes*, guru dapat menggunakan lembar rubrik ini untuk mencatat performa masing-masing kelompok atau individu.
- Umpan Balik: Berikan umpan balik tertulis dan lisan berdasarkan skor, serta diskusikan area perbaikan secara spesifik.

2. ASESMEN PROSES 2 - Menerapkan Struktur Kontrol Kondisional (LKPD 3.2)

Asesmen ini dilakukan untuk menguji kemampuan siswa dalam menerapkan struktur kondisional melalui LKPD 3.2: “Saldo Rekening”

Kriteria Penilaian

Skala:

4 = Sangat Baik

3 = Baik

2 = Cukup

1 = Perlu Perbaikan

Kriteria	4	3	2	1
Validasi <i>Input</i>	Memvalidasi <i>input</i> dengan benar menggunakan <i>scanf</i> dan menangani kasus <i>input</i> tidak valid (non-angka).	Validasi <i>input</i> ada tetapi tidak menangani semua kasus (misal: tidak keluar program).	Validasi <i>input</i> diimplementasikan dengan kesalahan logika.	Tidak ada validasi <i>input</i> .

Pengecekan Kondisi	Semua kondisi (negatif, saldo cukup, saldo tidak cukup) diimplementasikan dengan tepat menggunakan <i>if</i> , <i>else if</i> , dan <i>else</i> .	Salah satu kondisi kurang tepat (misal: urutan logika salah).	Hanya dua kondisi yang ditangani.	Hanya satu kondisi atau logika tidak jelas.
Urutan Logika	Urutan pengecekan logis: <i>input</i> tidak valid → nilai negatif → saldo cukup → saldo tidak cukup.	Urutan logika benar tetapi ada redundansi.	Urutan logika salah (misal: cek saldo sebelum nilai negatif).	Tidak ada urutan logika yang jelas.
Penanganan Error	Menampilkan pesan <i>error</i> spesifik untuk setiap kasus dan keluar program saat input tidak valid.	Pesan <i>error</i> kurang spesifik atau tidak keluar program saat input <i>invalid</i> .	Pesan <i>error</i> ambigu atau tidak sesuai kondisi.	Tidak ada penanganan <i>error</i> .
Keterbacaan Kode	Kode rapi, indentasi konsisten, dan komentar jelas (jika ada).	Indentasi kurang konsisten atau komentar minim.	Kode sulit dipahami karena format berantakan.	Kode tidak terstruktur.

Pedoman Penilaian

a. Validasi Input:

o Contoh 4:

```
if (scanf("%d", &tarik) != 1) {  
    printf("Input tidak valid. Harap masukkan  
    angka.\n");  
    return 1;  
}
```

o Contoh 1: Tidak ada blok kode untuk validasi input.

b. Pengecekan Kondisi:

o 4:

```
if (tarik < 0) { ... }  
else if (tarik <= saldo) { ... }  
else { ... }
```

o 1: Hanya menggunakan satu kondisi if tanpa else.

c. Urutan Logika:

o 4: Prioritas pengecekan: input *invalid* → nilai negatif → saldo cukup.

o 2: Contoh kesalahan: mengecek tarik <= saldo sebelum tarik < 0.

d. Penanganan *Error*:

o 4: Pesan *error* jelas dan program keluar dengan kode 1 saat input *invalid*.

o 3: Pesan *error* ada tetapi tidak keluar program.

e. Keterbacaan Kode:

o 4: Indentasi rapi, spasi konsisten, dan kode mudah diikuti.

o 1: Kode berantakan tanpa struktur.

Total Skor & Interpretasi

- 18-20: Sangat Baik
Siswa menguasai struktur percabangan dengan logika jelas, validasi lengkap, dan kode rapi.
- 13-16: Baik
Ada minor error (misal: urutan logika kurang optimal), tetapi konsep dasar benar.
- 8-12: Cukup
Terdapat kesalahan logika atau penanganan error yang tidak lengkap.
- 5-7: Perlu Perbaikan
Implementasi percabangan tidak sesuai dengan tujuan program.

Contoh Umpan Balik

- Kekuatan:
"Validasi *input* dan pengecekan nilai negatif diimplementasikan dengan tepat."
- Area Perbaikan:
"Urutan pengecekan seharusnya: input *invalid* → nilai negatif → saldo. Perbaiki penggunaan *else if*."

Catatan:

- Siswa dengan skor rendah disarankan berlatih membuat *pseudocode* sebelum menulis kode.
- Diskusikan kasus tepi (misal: input 0, nilai melebihi saldo).

Rubrik ini fokus pada penerapan struktur kontrol percabangan, bukan aspek lain seperti deklarasi variabel.

PERTEMUAN 4

ASESMEN PROSES - Menerapkan Struktur Kontrol Perulangan Untuk Kasus Spesifik (LKPD 4.1)

Asesmen ini dilakukan untuk menguji kemampuan siswa dalam menerapkan struktur kontrol perulangan melalui LKPD 4.1: “Total Harga Barang”

Kriteria Penilaian (Skala: 1-4):

Kriteria	4 (Sangat Baik)	3 (Baik)	2 (Cukup)	1 (Perlu Perbaikan)
Pemilihan Struktur Loop	Menggunakan <i>for</i> untuk kasus 1 dan <i>while</i> untuk kasus 2 dengan tepat.	Salah satu struktur <i>loop</i> kurang tepat.	Kedua struktur <i>loop</i> tidak sesuai.	Tidak menggunakan <i>loop</i> .
Penanganan Input	Input di-convert ke tipe data numerik dengan benar.	Ada 1 kesalahan konversi.	Input tidak di-convert.	Program <i>error</i> saat input.
Akurasi Perhitungan	Total harga dihitung dengan benar.	Ada 1 kesalahan <i>minor</i> dalam penjumlahan.	Total tidak akurat.	Perhitungan salah total.
Penanganan Kondisi Berhenti	Kasus 2: Program berhenti saat pengguna memasukkan sinyal berhenti.	Sinyal berhenti tidak ditangani dengan sempurna.	Program berhenti secara paksa.	Tidak ada mekanisme berhenti.
Output	Menampilkan total dengan format desimal (2 angka di belakang koma).	Format desimal tidak konsisten.	<i>Output</i> tidak jelas.	Tidak menampilkan total.

Pedoman Penskoran

- Total Skor:
 - 16-20: Sangat Baik (Program berjalan sempurna, logika jelas).
 - 12-15: Baik (Ada *minor error*, tetapi konsep dasar benar).
 - 8-11: Cukup (Kesalahan logika atau sintaks yang mempengaruhi hasil).
 - 4-7: Perlu Perbaikan (Program tidak menyelesaikan masalah).
- Contoh Umpan Balik:
 - Positif: "Penggunaan *for* dan *while* tepat. Total dihitung dengan benar!"
 - Perbaikan: "Pastikan konversi *input* ke *float* di JavaScript menggunakan `parseFloat`, bukan `parseInt`."

Catatan:

- Siswa yang menggunakan `do-while` untuk kasus 2 di C tetap dinilai benar.

PERTEMUAN 5

ASESMEN PROSES - Menerapkan Fungsi Untuk Meningkatkan Modularitas serta Efisiensi Kode Program (LKPD 5.1)

Asesmen ini dilakukan untuk menguji kemampuan siswa dalam penggunaan fungsi melalui aktivitas dalam LKPD 5.1: “*Coding Estafet*” yang mampu meningkatkan modularitas dan efisiensi kode program.

Total Skor Maksimal: 16

Kriteria	4 (Sangat Baik)	3 (Baik)	2 (Cukup)	1 (Perlu Perbaikan)
Ketepatan Logika	Semua program berjalan sesuai skenario uji.	1 program memiliki kesalahan logika.	2-3 program tidak berjalan.	Lebih dari 3 program error.
Penggunaan Fungsi	Fungsi diimplementasikan dengan benar dan modular.	Fungsi kurang modular (misal: parameter tidak jelas).	Fungsi tidak lengkap.	Tidak menggunakan fungsi.
Waktu Penyelesaian	Selesai dalam ≤ 25 menit.	Selesai dalam 26-30 menit.	Selesai dalam 31-35 menit.	>35 menit atau tidak selesai.
Kolaborasi	Anggota aktif berdiskusi dan saling membantu.	1 anggota kurang aktif.	2 anggota pasif.	Tidak ada kolaborasi.

PERTEMUAN 6

ASESMEN AKHIR : Menerapkan Pemrograman Berbasis Teks

Bentuk asesmen : Tes Praktik

Asesmen akhir ini dilakukan untuk menilai kemampuan siswa dalam menerapkan pemrograman berbasis teks.

"Bubble Tea Shop Simulator"

Pseudocode

```
PROGRAM BubbleTeaShop
DEKLARASI
    MENU: {"Original Milk Tea": 15000, ...}
    TOPPINGS: {"Boba": 3000, ...}
    harga_dasar, total_topping, total: integer
    pilihan_rasa, ukuran: string
    ID_pesanan: integer

MULAI
    TAMPILKAN "✨ SELAMAT DATANG DI BUBBLE TEA SHOP! ✨"
    TAMPILKAN_MENU()

    ULANGI:
        BACA pilihan_rasa
        SAMPAI pilihan_rasa ADA DI MENU

        ukuran = PILIH_UKURAN()
        total_topping = PILIH_TOPPING()

        JIKA ukuran = "Large" MAKA
            harga_dasar = MENU[pilihan_rasa] + 5000
        LAINNYA
            harga_dasar = MENU[pilihan_rasa]

        total = harga_dasar + total_topping
        ID_pesanan = GENERATE_ANGKA_ACAK(1000, 9999)

        TAMPILKAN_STRUK(pilihan_rasa, ukuran, total_topping, total, ID_pesanan)
    SELESAI
```

Contoh Output

```
✦ SELAMAT DATANG DI BUBBLE TEA SHOP! ✦

=== MENU BUBBLE TEA ===
Original Milk Tea: Rp15,000
Taro Milk Tea: Rp18,000
Matcha Milk Tea: Rp20,000
Chocolate Milk Tea: Rp17,000
=====
Pilih rasa: matcha
Pilih ukuran (R/L): L

=== TOPPING ===
Boba: Rp3,000
Pudding: Rp4,000
Jelly: Rp2,500
Cheese Foam: Rp5,000
Pilih topping (atau 'selesai'): boba
Ditambahkan: Boba
Pilih topping (atau 'selesai'): cheese foam
Ditambahkan: Cheese Foam
Pilih topping (atau 'selesai'): selesai

=== STRUK PESANAN ===
Rasa      : Matcha Milk Tea (Large)
Topping   : Rp8,000
TOTAL     : Rp33,000
ID Pesanan: #7284
Terima kasih! 😊
```

Instruksi Implementasi untuk Siswa

Langkah 1: *Setup Variabel*

- Buat *dictionary* untuk *menu* dan *topping*

Langkah 2: *Fungsi Tampilkan Menu*

- Gunakan *loop* untuk menampilkan *item menu*

Langkah 3: Validasi Input Ukuran

- Gunakan while loop sampai input valid

Langkah 4: Hitung Total Topping

- Gunakan *accumulator pattern*

Catatan untuk Siswa

a. Python:

- Gunakan title() untuk format input
- Manfaatkan random.randint() untuk ID pesanan

b. JavaScript:

- Gunakan toLocaleString() untuk format mata uang

c. C:

- Hindari *string* kompleks dengan menggunakan *array index*
- Gunakan toupper() untuk normalisasi input

Tantangan: Tambahkan fitur diskon 10% untuk pembelian di atas Rp50.000!

Tabel Skenario Uji Coba

ID	Deskripsi	Input	Output yang Diharapkan	Tujuan
TC-1	Pesan valid dengan ukuran <i>Large</i> dan <i>topping</i>	Rasa: Matcha Milk Tea Ukuran: L <i>Topping: Boba, Pudding</i>	Total: Rp20.000 (harga dasar) + Rp5.000 (<i>Large</i>) + Rp7.000 (<i>topping</i>) = Rp32.000 ID Pesanan: 4 digit angka	Verifikasi perhitungan harga dan format <i>output</i> .

		Durasi: -		
TC-2	Pesan tanpa <i>topping</i>	Rasa: <i>Taro Milk Tea</i> Ukuran: <i>R</i> <i>Topping</i> : Selesai	Total: Rp18.000 (harga dasar) + Rp0 = Rp18.000	Memastikan program menangani <i>input</i> tanpa <i>topping</i> .
TC-3	Input rasa tidak valid	Rasa: "Coklat"	Pesan error: " <i>Rasa tidak tersedia!</i> "	Validasi input rasa.
TC-4	Input ukuran tidak valid	Ukuran: "XL"	Pesan error: " <i>Pilihan tidak valid!</i> "	Validasi input ukuran.
TC-5	Input topping tidak valid	<i>Topping</i> : "Keju"	<i>Topping</i> diabaikan/tidak menambah harga	Validasi input topping.
TC-6	Total harga $\geq$ Rp50.000	Rasa: <i>Matcha Milk Tea (Large)</i> + 4 topping (misal: <i>Boba, Pudding, Jelly, Cheese Foam</i>)	Total: Rp25.000 (harga dasar) + Rp14.000 = Rp39.000 (jika tanpa diskon)	Uji kasus tepi untuk fitur diskon (opsional).

Rubrik Penilaian

Kriteria Penilaian (Skala: 1-4, 4 = Sangat Baik):

Aspek	4	3	2	1
Fungsionalitas	Semua skenario uji berjalan sempurna.	1-2 skenario <i>error</i> .	3-4 skenario <i>error</i> .	Program tidak berjalan.

Variabel & Tipe Data	Variabel deskriptif, tipe data tepat, dan konstanta digunakan.	Ada 1-2 kesalahan penamaan/tipe data.	Variabel ambigu/tipe data salah.	Variabel tidak relevan.
Struktur Kontrol	<i>Loop</i> dan percabangan digunakan optimal untuk validasi <i>input</i> .	Ada 1-2 redundansi logika.	Logika tidak efisien.	Tidak ada validasi <i>input</i> .
Fungsi	Program modular dengan ≥ 4 fungsi yang jelas.	Menggunakan 3 fungsi.	Menggunakan 2 fungsi.	Tidak menggunakan fungsi.
Library Eksternal	Menggunakan random untuk ID pesanan dan format mata uang (jika ada).	Hanya menggunakan 1 fitur <i>library</i> .	Tidak menggunakan <i>library</i> .	-
Keterbacaan Kode	Kode rapi, indentasi konsisten, komentar jelas.	Kode rapi tetapi komentar kurang.	Indentasi tidak konsisten.	Kode berantakan.
Penanganan Error	Menangani semua input tidak valid dengan pesan jelas.	Menangani 75% kasus <i>error</i> .	Menangani 50% kasus <i>error</i> .	Tidak ada penanganan <i>error</i> .
Kreativitas	Menambahkan fitur unik (diskon, loyalitas, dll).	Menambahkan 1 fitur tambahan.	Hanya mengikuti contoh.	Tidak ada inovasi.

Pedoman Penilaian

a. Skor Total:

- Setiap aspek dinilai 1-4.
- Total skor = Jumlah skor semua aspek.
- Maksimal: 32 poin (8 aspek × 4).

b. Konversi Nilai Akhir:

- A (Sangat Baik): 28-32 poin
- B (Baik): 21-27 poin
- C (Cukup): 14-20 poin
- D (Perlu Perbaikan): ≤13 poin

c. Ketentuan Tambahan:

- Bonus 5 poin untuk fitur kreatif yang relevan (misal: sistem diskon).
- Pengurangan 2 poin untuk keterlambatan pengumpulan.
- Plagiarisme: Langsung mendapat nilai E.

d. Umpan Balik:

- Berikan komentar spesifik untuk setiap aspek (contoh: "Validasi input ukuran sudah baik, tetapi penanganan topping tidak valid perlu diperbaiki").
- Soroti bagian kode yang inovatif.

Contoh Penilaian

Nama Siswa: Budi

- Fungsionalitas: 4
- Variabel & Tipe Data: 3
- Struktur Kontrol: 4

- Fungsi: 4
 - *Library*: 3
 - Keterbacaan: 4
 - *Error Handling*: 3
 - Kreativitas: 4 (+5 bonus)
- Total: 29 + 5 = 34 (Nilai A)

Catatan: "Program Budi sangat kreatif dengan tambahan fitur diskon, tetapi perlu memperbaiki penanganan *input topping*!"

Rangkuman Materi

Rangkuman Materi dapat diakses melalui tautan <https://s.id/KodingKA-materi-tp6>



KEMENTERIAN PENDIDIKAN DASAR DAN MENENGAH
BADAN, STANDAR, KURIKULUM, DAN ASESMEN PENDIDIKAN
PUSAT KURIKULUM DAN PEMBELAJARAN
REPUBLIK INDONESIA