KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa yang telah melimpahkan rahmat, taufik, dan hidayah-Nya sehingga kami dapat menhyelesaikan penyusunan Bahan Ajar Informatika Kelas IX Semester Gasal Tahun Ajaran 2026/2027 ini dengan baik tanpa suatu halangan yang berarti.

Bahan Ajar ini berisi ringkasan materi, latihan, petunjuk praktikum, dan soal-soal evaluasi yang diharapkan dapat membantu pelaksanaan kegiatan pembelajaran. Kami menyadari bahwa dalam penyusunan Bahan Ajar ini tidak lepas dari adanya kerja sama dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini kami ingin menyampaikan banyak terima kasih kepada semua pihak yang telah membantu dalam penyusunan Bahan Ajar ini.

Akhirnya kami menyadari bahwa Bahan Ajar ini masih jauh dari kata sempurna. Oleh karena itu dengan penuh kerendahan hati, kami mengharap kritik dan saran yang membangun guna perbaikan diwaktu yang akan datang.

Tim Penyusun
Bahan Ajar Informatika Kelas IX Semester Gasal
Tahun Ajaran 2026/2027
MGMP Informatika SMP Kabupaten Kudus
Penyusun : 1. Achmad Romadlon, S.Kom.
2. Dwi Susilo, S.T.
Editor : Peni Retnowati, S.Pd., S.Kom., M.Pd
Koordinator : Yusro, S.Pd., M.Pd.

DAFTAR ISI

BAB I
BERPIKIR KOMPUTASIONAL DAN PENERAPAN LINTAS BIDANG

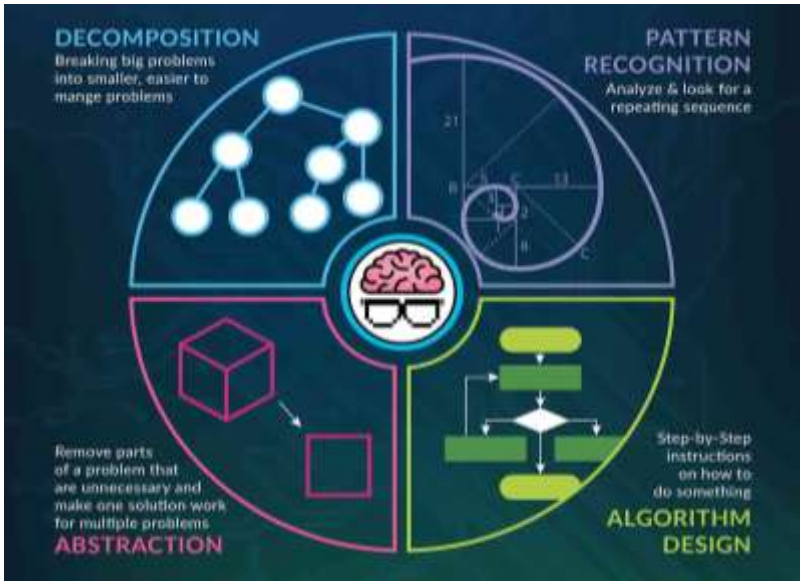
TUJUAN PEMBELAJARAN:
Setelah mempelajari materi ”Berpikir Komputasional dan Penerapan Lintas Bidang” ini, peserta didik diharapkan dapat:
1. Menerapkan konsep berpikir komputasional untuk menyelesaikan masalah
2. Mengembangkan dan menumbuhkan budaya komputasi melalui berpikir komputasional
3. Merumuskan tujuan analisis data yang spesifik, terukur, dan memiliki batasan waktu (*SMART principles*) berdasarkan studi kasus
4. Merancang kuesioner digital (menggunakan *Google Forms*) untuk menghasilkan himpunan data
5. Memanipulasi tampilan data menggunakan teknik *sorting* (pengurutan), *filtering* (penyaringan)
6. Membuat visualisasi data yang informatif dan menarik menggunakan alat bantu *spreadsheet*.

A. Pengertian Berpikir Komputasional

Berpikir Komputasional atau *Computational thinking* adalah proses berpikir yang digunakan untuk memecahkan masalah secara sistematis dan logis. Kemampuan pemecahan masalah ini memungkinkan kita untuk menganalisis masalah yang kompleks, memahami apa masalahnya, dan menentukan solusi yang tepat.

Dengan Berpikir Komputasional, kita dapat menguraikan masalah menjadi beberapa bagian atau tahapan yang efektif. Sehingga menghasilkan solusi yang tepat dan dapat dipahami baik oleh komputer, manusia, atau keduanya. Dalam konsepnya, terdapat empat pilar utama dalam berpikir komputasi, yang masing-masing memiliki tujuannya sendiri.

Berpikir Komputasional memiliki empat pilar atau tahapan penting yang perlu diketahui, diantaranya yaitu sebagai berikut:



Sumber: https://www.hostnic.id
Gambar 1.1. Empat pilar berpikir komputasional

Proses ini terdiri dari beberapa langkah, yaitu:
a. **Decomposing (Dekomposisi):** Memecah masalah yang kompleks menjadi masalah-masalah yang lebih kecil dan mudah dipecahkan.
b. **Pattern recognition (Pengenalan pola):** Mencari pola dalam data atau informasi yang tersedia.
c. **Abstraction (Abstraksi):** Mengidentifikasi bagian-bagian penting dari suatu masalah dan mengabaikan detail yang tidak penting.
d. **Algorithms (Algoritma):** Menyusun langkah-langkah yang terstruktur dan logis untuk menyelesaikan suatu masalah.

Manfaat Berpikir Komputasional:

a. **Meningkatkan kemampuan memecahkan masalah:** Berpikir komputasional dapat membantu siswa untuk memecahkan masalah secara sistematis dan logis, sehingga mereka dapat menemukan solusi yang lebih efektif dan efisien.
b. **Meningkatkan kemampuan berpikir kritis:** Berpikir komputasional dapat membantu siswa untuk berpikir kritis dan analitis, sehingga mereka dapat mengevaluasi informasi dan membuat keputusan yang tepat.
c. **Meningkatkan kemampuan kreativitas:** Berpikir komputasional dapat membantu siswa untuk berpikir kreatif dan inovatif, sehingga mereka dapat menemukan solusi yang baru dan unik untuk suatu masalah.
d. **Meningkatkan kemampuan kolaborasi:** Berpikir komputasional dapat membantu siswa untuk bekerja sama dengan orang lain untuk menyelesaikan masalah secara efektif.

TUGAS 1

1. Bayangkan Anda adalah seorang pengembang game. Jelaskan bagaimana Anda akan menerapkan pilar-pilar berpikir komputasional dalam proses pengembangan game Anda, mulai dari ide awal hingga peluncuran game.

2. Dalam era digital ini, informasi dengan mudah tersebar dan diakses oleh siapa saja. Bagaimana berpikir komputasional dapat membantu kita dalam mengevaluasi informasi yang kita temukan di internet, sehingga terhindar dari informasi yang salah atau menyesatkan?

3. Berpikir komputasional tidak hanya digunakan dalam bidang informatika, tetapi juga dapat diterapkan dalam berbagai disiplin ilmu lainnya. Berikan contoh penerapan berpikir komputasional dalam dua disiplin ilmu yang berbeda, dan jelaskan bagaimana hal tersebut membantu dalam menyelesaikan masalah di bidang tersebut.

B. Berpikir Komputasional Dalam Kehidupan Sehari-hari

1. Dekomposisi (*Decomposition)*

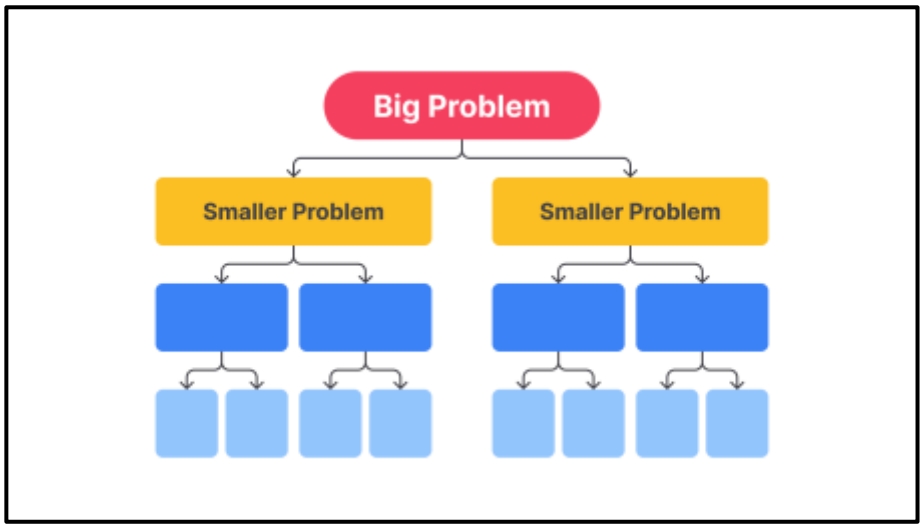
Dekomposisi (*Decomposition*): merupakan tahapan di mana kamu harus menguraikan masalah yang kompleks menjadi bagian yang lebih kecil, sehingga lebih mudah menyelesaikannya satu persatu.

Dekomposisi, atau pemecahan masalah, adalah salah satu langkah penting dalam berpikir komputasional (Computational Thinking). Teknik ini merupakan proses memecah masalah yang kompleks menjadi masalah-masalah yang lebih kecil dan mudah dipecahkan. Dengan mendekomposisi masalah, kita dapat membuatnya lebih mudah dipahami, dianalisis, dan diselesaikan.

Langkah-langkah Dekomposisi:

1. **Identifikasi masalah:** Langkah pertama adalah mengidentifikasi masalah yang ingin dipecahkan dengan jelas dan rinci.
2. **Definisikan tujuan:** Tetapkan tujuan yang ingin dicapai dengan menyelesaikan masalah.
3. **Bagi masalah menjadi sub-masalah:** Pecahkan masalah besar menjadi sub-masalah yang lebih kecil dan lebih terdefinisi dengan baik.
4. **Analisis sub-masalah:** Analisis setiap sub-masalah untuk memahami sifat dan kompleksitasnya.
5. **Ulangi langkah 3 dan 4:** Jika sub-masalah masih kompleks, ulangi langkah 3 dan 4 untuk memecahnya menjadi sub-sub-masalah yang lebih kecil.

6. **Selesaikan sub-masalah:** Selesaikan setiap sub-masalah satu per satu dengan menggunakan metode yang sesuai.
7. **Gabungkan solusi:** Gabungkan solusi dari sub-masalah untuk mendapatkan solusi keseluruhan untuk masalah utama.



Sumber: https://www.dicoding.com
Gambar 1.2. Dekomposisi (*Decomposition*)

Contoh penerapan decomposisi dalam kehidupan sehari-hari:

1. Merencanakan Liburan:

- **Masalah:** Merencanakan liburan keluarga ke pantai.
- **Tujuan:** Menghabiskan waktu berkualitas bersama keluarga di pantai dengan menikmati berbagai aktivitas yang menyenangkan.
- **Sub-masalah:**
  - Memilih destinasi pantai yang sesuai dengan keinginan keluarga.
  - Menentukan tanggal dan durasi perjalanan.
  - Memilih akomodasi yang nyaman dan sesuai budget.
  - Memesan tiket transportasi dan akomodasi.
  - Menyusun *itinerary* perjalanan, termasuk aktivitas dan tempat wisata yang ingin dikunjungi.
  - Membawa perlengkapan yang dibutuhkan selama perjalanan.
  - Mempersiapkan anggaran perjalanan.
- **Solusi:** Selesaikan setiap sub-masalah satu per satu, kemudian gabungkan solusinya untuk mendapatkan rencana liburan yang lengkap dan terorganisir.

2. Belajar Memasak:

- **Masalah:** Belajar membuat hidangan baru, seperti rendang.
- **Tujuan:** Mampu memasak rendang dengan rasa yang lezat dan otentik.
- **Sub-masalah:**
  - Mempelajari bahan-bahan dan bumbu yang dibutuhkan untuk membuat rendang.
  - Memahami langkah-langkah dan teknik memasak rendang yang benar.
  - Berlatih memasak rendang dengan mengikuti resep dan petunjuk yang tersedia.
  - Mengevaluasi rasa dan tekstur rendang yang telah dimasak.
  - Memperbaiki dan menyempurnakan teknik memasak rendang berdasarkan evaluasi.
- **Solusi:** Selesaikan setiap sub-masalah satu per satu, kemudian gabungkan solusinya untuk mendapatkan kemampuan memasak rendang yang handal.

Manfaat Decomposisi:

- **Membuat masalah lebih mudah dipahami:** Dengan memecah masalah besar menjadi sub-masalah yang lebih kecil, kita dapat lebih mudah memahami kompleksitas dan interaksi antar komponen masalah.
- **Memudahkan analisis masalah:** Sub-masalah yang lebih kecil dan terdefinisi dengan baik lebih mudah dianalisis untuk menemukan akar penyebab masalah dan solusi yang tepat.
- **Meningkatkan efisiensi penyelesaian masalah:** Dengan fokus pada satu sub-masalah pada satu waktu, kita dapat menyelesaikan masalah dengan lebih efisien dan menghindari kebingungan.
- **Meningkatkan kreativitas:** Saat memecah masalah menjadi sub-masalah, kita terdorong untuk mencari solusi yang kreatif dan inovatif untuk setiap sub-masalah.
- **Meningkatkan kemampuan kolaborasi:** Dekomposisi dapat membantu tim untuk bekerja sama secara efektif dalam menyelesaikan masalah yang kompleks dengan membagi tugas dan tanggung jawab.

2. Pengenalan Pola (*Pattern recognition*)

Pengenalan pola (Pattern Recognition) adalah salah satu pilar penting dalam berpikir komputasional (*Computational Thinking*). Teknik ini merupakan proses menemukan pola dalam data atau informasi yang tersedia. Pola ini dapat berupa bentuk, struktur, urutan, atau hubungan yang berulang. Dengan mengenali pola, kita dapat memahami data dengan lebih baik, membuat prediksi, dan mengambil keputusan yang lebih efektif.

Sumber: https://www.dicoding.com
Gambar 1.3. Pengenalan pola (Pattern Recognition)

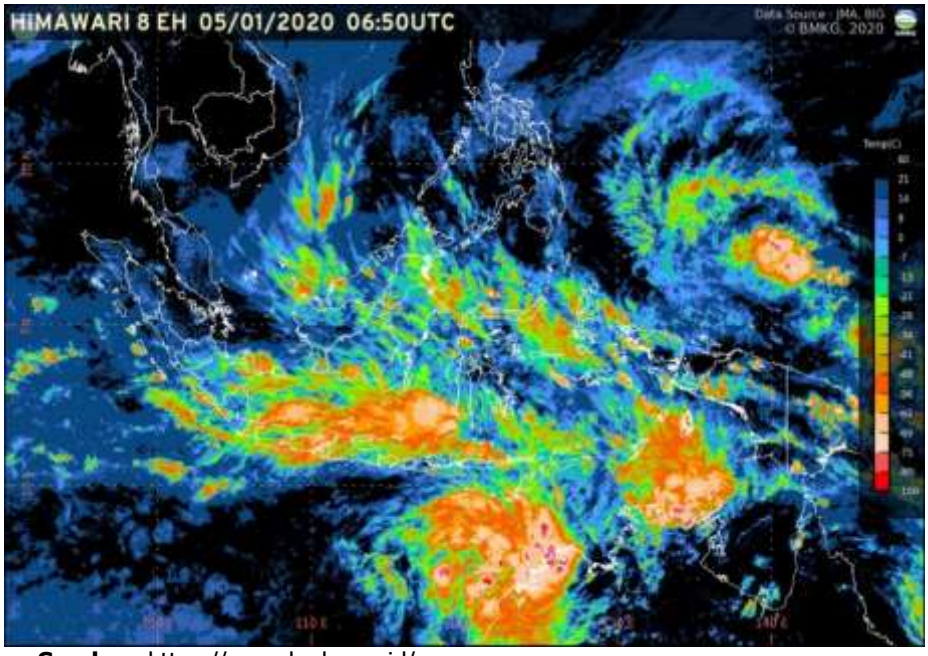
Langkah-langkah Pengenalan Pola:

1. **Pengumpulan data:** Kumpulkan data yang relevan dengan masalah atau pertanyaan yang ingin dijawab.
2. **Pembersihan data:** Pastikan data yang dikumpulkan bersih dan bebas dari kesalahan atau redundansi.
3. **Analisis data:** Jelajahi data untuk mencari pola yang terlihat secara kasat mata.
4. **Visualisasi data:** Gunakan grafik, tabel, atau diagram untuk memvisualisasikan data dan mempermudah identifikasi pola.
5. **Abstraksi:** Identifikasi bagian-bagian penting dari pola dan abaikan detail yang tidak relevan.
6. **Formulasi hipotesis:** Buatlah hipotesis tentang makna di balik pola yang ditemukan.
7. **Pengujian hipotesis:** Uji hipotesis dengan menggunakan data tambahan atau eksperimen.
8. **Penyesuaian hipotesis:** Perbaiki hipotesis berdasarkan hasil pengujian.

Contoh Penerapan Pengenalan Pola dalam Kehidupan Sehari-hari:

1. Prediksi Cuaca:

Ahli meteorologi menggunakan pola data cuaca di masa lampau untuk memprediksi cuaca di masa depan. Mereka mencari pola dalam suhu, tekanan udara, kelembaban, dan faktor lainnya untuk membuat model prediksi cuaca.



Sumber: https://www.bmkg.go.id/

Gambar 1.4. Pengenalan pola (Pattern Recognition) pada cuaca

2. Rekomendasi Produk:

Situs web *e-commerce* menggunakan pola pembelian dan perilaku pengguna untuk merekomendasikan produk yang mungkin mereka sukai. Mereka menganalisis data pembelian, riwayat penelusuran, dan interaksi pengguna untuk mengidentifikasi pola yang menunjukkan minat dan kebutuhan pengguna.

3. Deteksi Penipuan:

Lembaga keuangan menggunakan pola transaksi keuangan untuk mendeteksi aktivitas penipuan. Mereka mencari pola yang tidak biasa dalam jumlah transaksi, lokasi transaksi, dan waktu transaksi untuk mengidentifikasi potensi penipuan.

4. Diagnosa Medis:

Dokter menggunakan pola gejala dan hasil tes medis untuk mendiagnosis penyakit. Mereka mencari pola yang menunjukkan penyakit tertentu dan kemudian melakukan tes lebih lanjut untuk mengkonfirmasi diagnosis.

5. Pengenalan Wajah:

Sistem pengenalan wajah menggunakan pola fitur wajah untuk mengidentifikasi seseorang. Mereka menganalisis jarak antar mata, bentuk hidung, dan fitur lainnya untuk membuat model wajah seseorang dan kemudian mencocokkannya dengan gambar atau video baru.

Manfaat Pengenalan Pola:

a. **Meningkatkan pemahaman data:** Pengenalan pola membantu kita memahami data dengan lebih baik dan menemukan makna yang tersembunyi di dalamnya.
b. **Membuat prediksi:** Dengan mengenali pola dalam data di masa lampau, kita dapat membuat prediksi tentang apa yang akan terjadi di masa depan.
c. **Mengambil keputusan:** Pengenalan pola membantu kita mengambil keputusan yang lebih efektif dengan memberikan informasi yang lebih akurat dan relevan.
d. **Meningkatkan efisiensi:** Pengenalan pola dapat mengotomatiskan tugas-tugas yang berulang dan meningkatkan efisiensi dalam berbagai bidang.

e. **Mendorong inovasi:** Pengenalan pola dapat memicu ide-ide baru dan inovasi dalam berbagai bidang, seperti ilmu pengetahuan, teknologi, dan bisnis.

3. Abstraksi (*Abstraction*)

Abstraksi adalah salah satu pilar penting dalam berpikir komputasional (*Computational Thinking*). Teknik ini merupakan proses mengidentifikasi bagian-bagian penting dari suatu masalah dan mengabaikan detail yang tidak penting. Dengan melakukan abstraksi, kita dapat menyederhanakan masalah yang kompleks menjadi lebih mudah dipahami dan dipecahkan.

Langkah-langkah Abstraksi:

a. **Identifikasi masalah:** Definisikan masalah yang ingin dipecahkan dengan jelas dan ringkas.
b. **Analisis masalah:** Uraikan masalah menjadi komponen-komponen yang lebih kecil dan identifikasi bagian-bagian yang penting.
c. **Identifikasi detail yang relevan:** Tentukan detail mana yang penting untuk menyelesaikan masalah dan detail mana yang dapat diabaikan.
d. **Membuat model abstrak:** Buatlah model yang mewakili bagian-bagian penting dari masalah dan abaikan detail yang tidak relevan.
e. **Verifikasi model abstrak:** Pastikan model abstrak secara akurat mewakili bagian-bagian penting dari masalah dan dapat digunakan untuk memecahkannya.

Manfaat Abstraksi:

a. **Meningkatkan pemahaman:** Abstraksi membantu kita memahami masalah dengan lebih baik dengan fokus pada bagian-bagian yang penting dan mengabaikan detail yang tidak relevan.
b. **Memudahkan penyelesaian masalah:** Dengan menyederhanakan masalah, abstraksi membantu kita menemukan solusi yang lebih mudah dan efektif.
c. **Meningkatkan komunikasi:** Abstraksi membantu kita mengkomunikasikan masalah dan solusi dengan lebih jelas kepada orang lain.
d. **Mendorong kreativitas:** Abstraksi memungkinkan kita untuk melihat masalah dari sudut pandang yang berbeda dan menemukan solusi yang inovatif.

Contoh Abstraksi dalam Kehidupan Sehari-hari:

1. Peta:

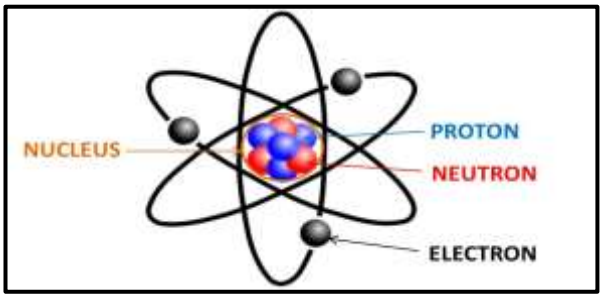
Peta adalah contoh abstraksi yang umum digunakan dalam kehidupan sehari-hari. Peta menyederhanakan dunia yang kompleks menjadi gambar dua dimensi yang mudah dipahami. Peta hanya menunjukkan informasi yang penting untuk navigasi, seperti jalan, sungai, dan gunung, dan mengabaikan detail yang tidak relevan, seperti pohon, mobil, dan orang.

Sumber: https://betanews.id
Gambar 1.5. Abstraksi pada peta Jawa Tengah

2. Resep:

Resep adalah contoh lain dari abstraksi. Resep menyederhanakan proses memasak yang kompleks menjadi langkah-langkah yang mudah diikuti. Resep hanya menunjukkan bahan-bahan dan instruksi yang penting untuk membuat hidangan, dan mengabaikan detail yang tidak relevan, seperti ukuran dapur, jenis peralatan masak, dan preferensi pribadi.

3. Model Atom:



Model atom adalah contoh abstraksi yang digunakan dalam ilmu pengetahuan. Model atom menyederhanakan struktur atom yang kompleks menjadi representasi visual yang mudah dipahami. Model atom hanya menunjukkan bagian-bagian penting dari atom, seperti proton, neutron, dan elektron, dan mengabaikan detail yang tidak relevan, seperti massa dan spin partikel subatomik.

Sumber: https://fatek.umsu.ac.id/
Gambar 1.6. Abstraksi pada struktur atom

4. Algoritma Komputer:

Algoritma komputer adalah contoh abstraksi yang digunakan dalam ilmu komputer. Algoritma menyederhanakan proses menyelesaikan tugas menjadi langkah-langkah yang jelas dan terdefinisi. Algoritma hanya menunjukkan instruksi yang penting untuk menyelesaikan tugas, dan mengabaikan detail yang tidak relevan, seperti jenis komputer, bahasa pemrograman, dan antarmuka pengguna.

4. Algoritma (Algorithms)

Algoritma adalah urutan langkah-langkah yang terstruktur dan logis untuk menyelesaikan suatu masalah. Algoritma seperti resep masakan yang memandu kita dalam menghasilkan hidangan yang lezat.

Ciri-ciri Algoritma:

a. **Terstruktur:** Algoritma memiliki urutan langkah yang jelas dan tidak ambigu.
b. **Logis:** Setiap langkah dalam algoritma didasarkan pada logika dan alasan yang benar.
c. **Terbatas:** Algoritma memiliki jumlah langkah yang terbatas dan akan selalu berhenti pada suatu titik.
d. **Umum:** Algoritma dapat diterapkan untuk menyelesaikan berbagai masalah dengan jenis yang sama.

Manfaat Algoritma:

a. **Memudahkan penyelesaian masalah:** Algoritma membantu kita memecahkan masalah yang kompleks menjadi langkah-langkah yang lebih kecil dan lebih mudah dipecahkan.
b. **Meningkatkan efisiensi:** Algoritma yang optimal dapat menyelesaikan masalah dengan lebih cepat dan dengan sumber daya yang lebih sedikit.
c. **Meningkatkan akurasi:** Algoritma yang dirancang dengan baik dapat membantu kita menghindari kesalahan dan menghasilkan solusi yang akurat.
d. **Mempermudah komunikasi:** Algoritma dapat didokumentasikan dan dibagikan dengan orang lain, sehingga memudahkan kolaborasi dan pemahaman.

Contoh Algoritma dalam Kehidupan Sehari-hari:

1. **Membuat Teh:**
   - **Langkah 1:** Rebus air dalam panci.
   - **Langkah 2:** Masukkan kantong teh atau teh celup ke dalam air mendidih.
   - **Langkah 3:** Biarkan teh diseduh selama beberapa menit.
   - **Langkah 4:** Angkat kantong teh atau celup.
   - **Langkah 5:** Tambahkan gula, susu, atau lemon sesuai selera.
   - **Langkah 6:** Sajikan teh dan nikmati.

2. **Mencari Akar Kuadrat:**
   - **Langkah 1:** Bagi bilangan yang ingin dicari akar kuadratnya dengan 2.
   - **Langkah 2:** Hitung rata-rata dari hasil langkah 1 dan bilangan yang ingin dicari akar kuadratnya.
   - **Langkah 3:** Gunakan hasil langkah 2 sebagai perkiraan akar kuadrat.
   - **Langkah 4:** Ulangi langkah 2 dan 3 dengan menggunakan perkiraan akar kuadrat dari langkah sebelumnya sebagai nilai baru untuk dibagi dan dihitung rata-ratanya.
   - **Langkah 5:** Teruskan mengulangi langkah 2 dan 3 sampai perkiraan akar kuadrat mencapai tingkat akurasi yang diinginkan.

TUGAS 2

1. Anda adalah seorang seniman yang ingin menciptakan karya seni baru yang unik dan inspiratif. Bagaimana Anda dapat menggunakan pengenalan pola untuk membantu Anda dalam proses kreatif Anda? Jelaskan bagaimana Anda dapat menemukan pola dalam alam, budaya, atau pengalaman pribadi Anda dan menggunakan pola tersebut untuk menginspirasi karya seni Anda
2. Anda adalah seorang peneliti yang ingin mempelajari perilaku hewan liar di habitat alaminya. Bagaimana Anda dapat menerapkan abstraksi untuk membantu Anda dalam penelitian Anda? Jelaskan bagaimana Anda dapat mengabstraksi data pengamatan, pola perilaku, dan faktor lingkungan untuk memahami dengan lebih baik bagaimana hewan liar beradaptasi dengan lingkungannya

C. Struktur Data

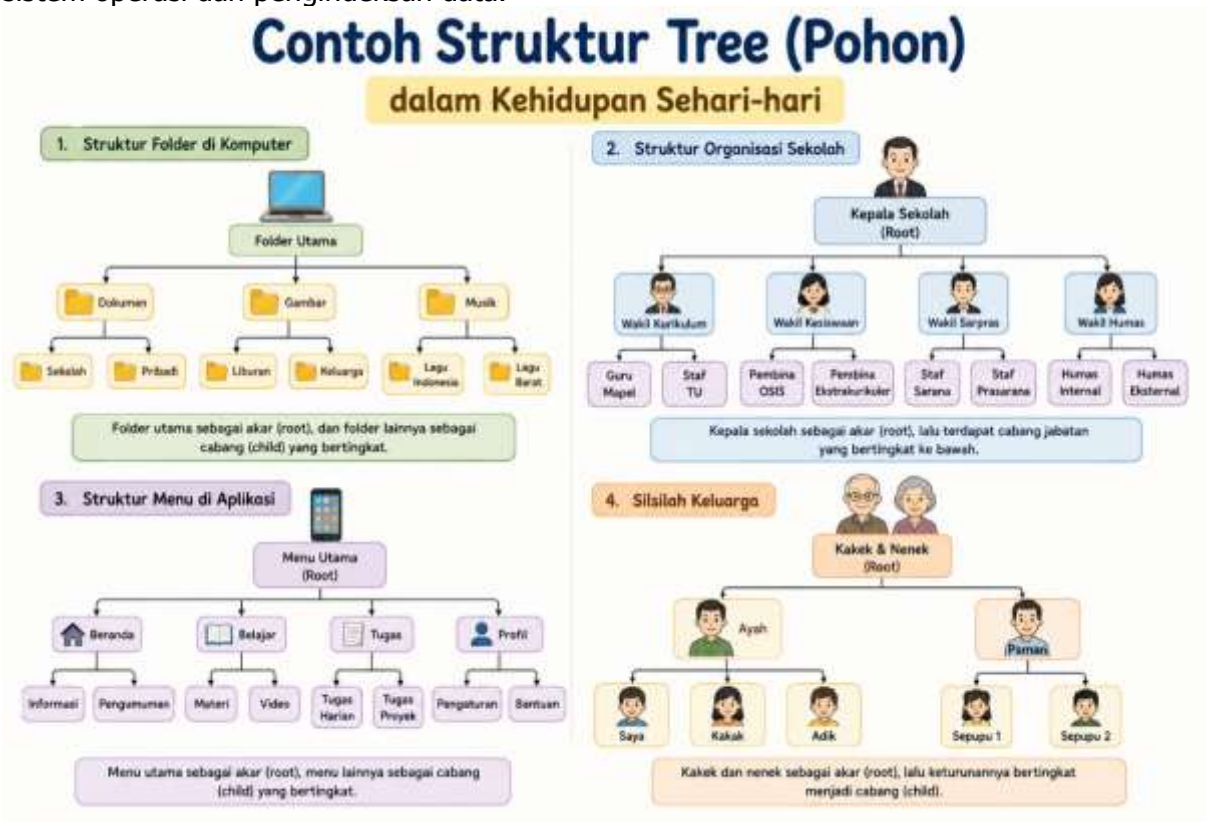
Struktur data adalah cara mengorganisasi, mengatur, dan menyimpan data dalam sistem komputer atau *database* agar dapat diakses, dikelola, dan dimodifikasi secara efisien. Ibarat sebuah wadah, struktur data memastikan data tertata rapi sehingga memori komputer digunakan secara optimal dan proses pencarian data berjalan cepat.

1. **Struktur Data Tree (Pohon)**

   Tree merupakan bentuk struktur data yang membentuk hierarki melalui hubungan antar simpul (node) yang terhubung oleh jalur (edge). Dimulai dari simpul utama yang disebut root, setiap node bisa memiliki satu atau lebih node turunan (child), sedangkan node yang tidak memiliki turunan disebut sebagai daun (leaf).

   Karena sifatnya yang tidak memiliki urutan linear tetap, tree memungkinkan representasi data yang lebih fleksibel dan cocok untuk berbagai aplikasi kompleks, seperti **struktur direktori sistem file**, **parser compiler**, dan **pohon keputusan dalam AI** (Artificial Intelligence).

   Struktur data tree dalam Informatika digunakan untuk menyimpan data yang memiliki hubungan bertingkat atau disebut juga struktur hierarki. Misalnya, dalam sebuah keluarga, ada kakek-nenek di tingkat paling atas, lalu orang tua, dan terakhir anak-anak. Nah, struktur seperti ini cocok disimpan dalam bentuk pohon karena setiap data bisa dihubungkan parent dan child.
   Setiap simpul dalam tree memiliki satu simpul induk, kecuali simpul akar yang tidak memiliki simpul induk. Tree digunakan dalam representasi struktur hirarkis, seperti direktori file dalam sistem operasi dan pengindeksan data.

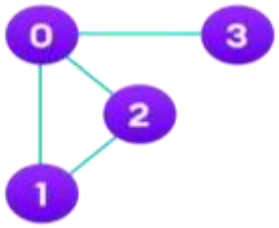


Gambar 1.7. Contoh Pohon

2. **Struktur Data Graph**

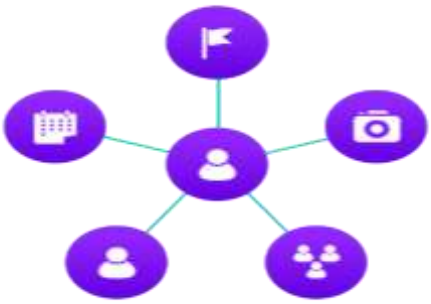
   Graph adalah struktur data yang terdiri dari simpul (vertex) dan sisi (edge) yang merepresentasikan hubungan antara objek. Graph adalah jenis struktur data umum yang susunan datanya tidak berdekatan satu sama lain (non-linier). Graph terdiri dari kumpulan simpul berhingga untuk menyimpan data dan antara dua buah simpul terdapat hubungan saling keterkaitan.
   Graph digunakan dalam analisis jaringan sosial, algoritma pencarian jalur terpendek, dan permasalahan yang melibatkan relasi antara entitas.

Simpul pada graph disebut dengan **verteks (V)**, sedangkan sisi yang menghubungkan antar verteks disebut **edge (E)**. Pasangan (x,y) disebut sebagai edge, yang menyatakan bahwa simpul x terhubung ke simpul y.
Sebagai contoh, terdapat graph seperti berikut:



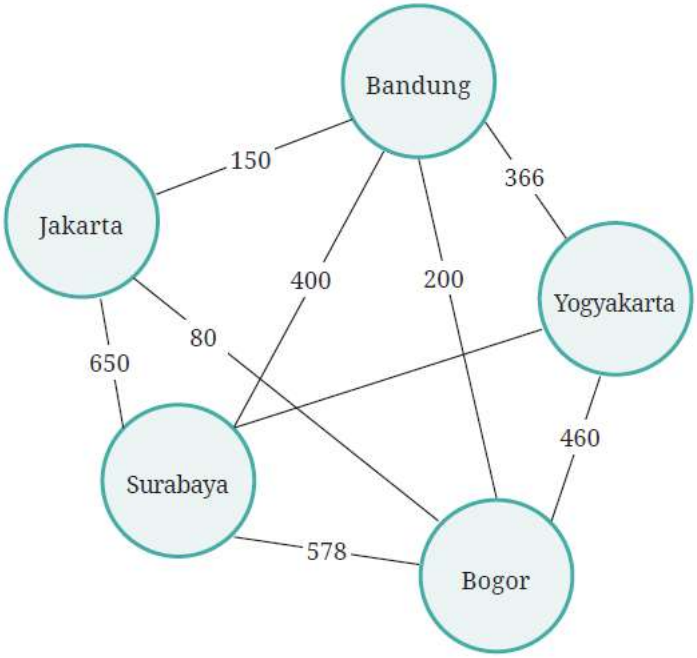
Sumber : programiz.com
Gambar 1.8 Contoh Graph

Graph di atas terdiri atas 4 buah verteks dan 4 pasang sisi atau edge. Dengan verteks disimbolkan sebagai V, edge dilambangkan E, dan graph disimbolkan G, ilustrasi di atas dapat ditulis dalam notasi berikut:
V = {0, 1, 2, 3}
E = {(0,1), (0,2), (0,3), (1,2)}
G = {V, E}
Graph banyak dimanfaatkan untuk menyelesaikan masalah dalam kehidupan nyata, dimana masalah tersebut perlu direpresentasikan atau diimajinasikan seperti sebuah jaringan.



Sumber: programiz.com
Gambar 1.9 Gambar Relasi Graph

Contohnya adalah jejaring sosial (seperti Facebook, Instagram, LinkedIn, dkk). Pengguna di Facebook dapat dimisalkan sebagai sebuah simpul atau verteks, sementara hubungan pertemanan antara pengguna tersebut dengan pengguna lain direpresentasikan sebagai edge. Tiap tiap verteks dapat berupa struktur yang mengandung informasi seperti id user, nama, gender, dll. Tidak hanya data pengguna, data apapun yang ada di Facebook adalah sebuah simpul atau verteks.Termasuk foto, album, komentar, event, group, story, dan lain sebagainya.
Pengguna dapat mengunggah foto. Ketika telah diunggah, foto akan menjadi bagian dari album. Foto juga dapat dikomentari oleh pengguna lain dan mereka dapat saling berbalas komentar. Semuanya terhubung satu sama lain, baik dalam bentuk relasi one-to-many, many-to-one, atau many-to-many.
Misalnya, jarak transportasi dari Jakarta ke Bandung adalah 150 km.



Gambar 1.20 Contoh Representasi Jalur Transportasi

TUGAS 3

Dari contoh diatas, kamu bisa melihat bahwa struktur data *graph* berbeda dengan struktur data *tree*. Menurutmu, apa perbedaan yang paling jelas di antara keduanya?

3. **Analisis Himpunan Data Terstruktur dengan Teknik Visualisasi**
   Analisis data tidak pernah dimulai dari langsung membuka komputer dan mengetik angka. Proses ini selalu diawali dengan sebuah tujuan yang jelas. Menentukan tujuan di awal berfungsi sebagai kompas penunjuk arah, agar kita tahu data apa saja yang perlu dikumpulkan dan pola apa yang harus dicari.

   Metode S.M.A.R.T dalam Menentukan Tujuan

   Agar tujuan analisis data menjadi tajam dan efektif, kita dapat menggunakan pendekatan SMART:
   - **Specific (Spesifik):** Tujuan harus terfokus. Jangan katakan 'Saya ingin menganalisis data kantin', tetapi gunakan 'Saya ingin mengetahui makanan yang paling laris dikonsumsi siswa kelas 9 pada jam istirahat pertama'.
   - **Measurable (Terukur):** Tujuan harus dapat dinilai dengan angka numerik, seperti persentase peningkatan atau jumlah kuantitas tertentu.
   - **Achievable (Dapat Dicapai):** Tujuan analisis harus realistis sesuai dengan kemampuan dan batas waktu yang kita miliki sebagai siswa.
   - **Relevant (Relevan):** Analisis yang dilakukan harus memberikan manfaat nyata bagi penyelesaian masalah yang dihadapi.
   - **Time-bound (Batas Waktu):** Memiliki batasan waktu pengamatan yang jelas (misal: data selama bulan Agustus).

   Contoh Kasus Sekolah:
   Pengurus OSIS ingin mengadakan turnamen e-sports pasca-ujian semester. Tujuannya adalah menentukan jenis game apa yang paling diminati oleh siswa agar partisipasi lomba maksimal. Pertanyaan pemandunya adalah: 'Apa genre game terpopuler di antara siswa kelas 7, 8, dan 9?.

Jenis Sumber Data

a. **Data Primer:** Data yang diperoleh secara langsung dari objek penelitian oleh analis. Contohnya: menyebarkan kuesioner digital (Google Forms) kepada teman sekelas, melakukan wawancara terstruktur, atau melakukan penghitungan fisik langsung di lapangan.
b. **Data Sekunder:** Data yang dikumpulkan oleh pihak lain dan kita gunakan kembali. Contohnya: mengambil data absensi siswa dari staf tata usaha, mengunduh dataset publik dari Badan Pusat Statistik (BPS), atau dari repositori online seperti Kaggle.

Memanfaatkan Formulir Digital (Google Forms)

Untuk mendapatkan data terstruktur secara cepat, kuesioner digital adalah pilihan terbaik. Agar data langsung rapi dalam bentuk kolom, gunakan jenis pertanyaan yang membatasi input pengguna, seperti:

a. **Pilihan Ganda (Multiple Choice):** Untuk pilihan mutlak yang hanya boleh memilih satu jawaban.
b. **Kotak Centang (Checkboxes):** Jika responden boleh memilih lebih dari satu jawaban.
c. **Skala Linier (Linear Scale):** Untuk mengukur tingkat kepuasan atau intensitas (misal skala 1-5).

Hindari menggunakan tipe jawaban singkat/paragraf terbuka yang terlalu banyak, karena jawaban yang ditulis bebas oleh manusia cenderung sulit dibaca langsung oleh sistem komputer.

Data mentah yang baru saja didapatkan dari kuesioner sering kali 'kotor' atau berantakan. Dalam ilmu informatika, ada prinsip populer bernama GIGO: Garbage In, Garbage Out. Jika kita memasukkan data yang rusak, maka keputusan yang dihasilkan pun akan salah. Oleh karena itu, kita harus melakukan pembersihan data

Masalah Utama pada Data Mentah & Solusinya:

a. **Data Kosong (Missing Values):** Terdapat baris yang kosong karena responden terlewat menjawab.
   Solusi: Menghapus baris tersebut jika tidak krusial, atau mengisinya dengan nilai rata-rata (mean).
b. **Data Ganda (Duplicates):** Satu responden tidak sengaja menekan tombol kirim dua kali sehingga datanya sama persis.
   Solusi: Menggunakan fitur 'Hapus Duplikat' pada aplikasi spreadsheet.
c. Format Tidak Konsisten: Penulisan teks atau angka yang berbeda-beda. Misal ada yang menulis 'Rp 10.000', '10000', atau '10 ribu'.
   Solusi: Menyeragamkannya menjadi tipe data angka murni (Number format).
d. **Pencilan (Outliers):** Data yang nilainya tidak masuk akal atau terlalu jauh dari rata-rata karena salah ketik. Contoh: Umur siswa kelas 9 tertulis '140 tahun' padahal seharusnya '14 tahun'.
   Solusi: Memperbaiki atau menghapus baris tersebut.

Mengeksplorasi data adalah proses 'berkenalan' lebih dalam dengan data yang sudah bersih guna mencari tahu karakteristik dasar, pola menarik, atau keanehan di dalam kumpulan data tersebut menggunakan bantuan statistik sederhana.

Fungsi Statistik Dasar pada Spreadsheet:

a. **Mean (Rata-rata):** Digunakan untuk mencari nilai rata-rata dari himpunan data angka. Rumus di Excel/Google Sheets: =AVERAGE(range_data)
b. **Median (Nilai Tengah):** Nilai tengah yang membagi data menjadi dua bagian sama besar setelah diurutkan. Rumus: =MEDIAN(range_data)
c. **Modus (Mode):** Nilai atau kategori yang paling sering muncul dalam tabel data. Rumus: =MODE(range_data)
d. **Min dan Max:** Masing-masing digunakan untuk mendeteksi nilai terendah dan tertinggi dalam rentang data. Rumus: =MIN(range_data) dan =MAX(range_data)

Teknik Manipulasi Data Tambahan:

a. **Sorting (Pengurutan):** Mengurutkan data dari nilai terkecil ke terbesar (Ascending) atau terbesar ke terkecil (Descending) agar memudahkan pencarian peringkat.
b. **Filtering (Penyaringan):** Menyembunyikan baris data yang tidak sesuai kriteria dan hanya memunculkan data yang diinginkan (Misal: memfilter tabel agar hanya menampilkan siswa perempuan saja).

Memvisualisasikan dan Memublikasikan Hasil Analisis Data

Data yang menumpuk di spreadsheet berupa deretan angka akan sulit dipahami secara cepat oleh mata manusia. Visualisasi data bertugas mengubah kumpulan angka tersebut menjadi representasi grafis (gambar/diagram) agar pesan utama data bisa ditangkap seketika.

Jenis Grafik Utama dan Fungsinya:

a. **Grafik Batang (Bar Chart):** Sangat ideal untuk membandingkan nilai atau kuantitas antar-kategori yang berbeda. Contoh: Perbandingan jumlah koleksi buku per genre di perpustakaan.
b. **Grafik Garis (Line Chart):** Tepat digunakan untuk menampilkan tren, fluktuasi, atau perkembangan data kontinu dari waktu ke waktu secara kronologis. Contoh: Tren nilai rata-rata ujian kelas dari bulan Januari hingga Mei.
c. **Grafik Lingkaran (Pie Chart):** Digunakan untuk memperlihatkan proporsi atau persentase kontribusi bagian terhadap keseluruhan nilai total (total wajib 100%). Contoh: Komposisi persentase moda transportasi siswa pergi ke sekolah.
d. **Diagram Pencar (Scatter Plot):** Berguna untuk melihat pola hubungan atau korelasi antara dua variabel numerik yang berbeda. Contoh: Hubungan antara durasi bermain gadget (jam) dengan pencapaian nilai rapor siswa.

Etika Publikasi dan Komunikasi Data

Setelah visualisasi selesai dikemas menjadi laporan, infografis, atau slide presentasi, kita harus memperhatikan aspek legalitas dan moralitas sebelum membagikannya ke publik:

a. **Anonimisasi Data (Privacy):** Wajib menghapus atau menyamarkan informasi pribadi sensitif seperti Nama Lengkap, Nomor HP, atau Alamat Rumah demi melindungi hak privasi narasumber sesuai dengan Undang-Undang Pelindungan Data Pribadi (UU PDP).
b. **Integritas Skala Grafik:** Seorang analis data tidak boleh memanipulasi atau memotong skala sumbu diagram (misal memulai sumbu Y dari angka acak bukan nol) secara sengaja untuk melebih-lebihkan perbedaan kecil. Hal ini disebut visualisasi yang menyesatkan (misleading charts).

TUGAS 4

Buatkan visualisasi data menjadi representasi grafik, "Data kehadiran siswa dikelas bulan Juli, Agustus, September, Oktober".

UJI KOMPETENSI BAB. I
BERPIKIR KOMPUTASIONAL DAN PENERAPAN LINTAS BIDANG

I. Berilah tanda silang (X) pada huruf A, B, C, atau D pada jawaban yang paling tepat!

1. Manakah langkah pertama dalam proses dekomposisi?
   A. Menganalisis sub-masalah
   B. Mengidentifikasi masalah
   C. Menentukan tujuan
   D. Menggabungkan solusi
2. Apa manfaat dekomposisi dalam menyelesaikan masalah?
   A. Membuat masalah lebih sulit dipahami
   B. Memudahkan analisis masalah
   C. Meningkatkan kebingungan dalam penyelesaian masalah
   D. Mengurangi kreativitas dalam mencari solusi
3. Dalam dekomposisi, masalah besar dipecah menjadi...
   A. Masalah yang lebih besar
   B. Sub-masalah yang lebih kecil
   C. Masalah yang sama
   D. Solusi yang berbeda
4. Berikut adalah yang bukan contoh penerapan dekomposisi dalam kehidupan sehari-hari adalah...
   A. Merencanakan liburan keluarga
   B. Belajar memasak hidangan baru
   C. Menulis novel
   D. Menonton film
5. Manakah pernyataan yang paling tepat tentang dekomposisi?
   A. Dekomposisi adalah teknik yang membingungkan dan tidak efektif dalam menyelesaikan masalah.
   B. Dekomposisi hanya bermanfaat untuk menyelesaikan masalah matematika.
   C. Dekomposisi adalah teknik yang membantu memecah masalah kompleks menjadi bagian-bagian kecil yang mudah dipecahkan.
   D. Dekomposisi hanya dapat digunakan oleh orang-orang yang ahli dalam ilmu komputer.
6. Manakah langkah pertama dalam proses pengenalan pola?
   A. Formulasi hipotesis
   B. Pengumpulan data
   C. Abstraksi
   D. Visualisasi data
7. Apa manfaat pengenalan pola dalam pengambilan keputusan?
   A. Memperlambat proses pengambilan keputusan
   B. Meningkatkan akurasi dan efektivitas pengambilan keputusan
   C. Mengurangi informasi yang tersedia untuk pengambilan keputusan
   D. Membuat proses pengambilan keputusan lebih kompleks
8. Berikut adalah contoh penerapan pengenalan pola dalam kehidupan sehari-hari, kecuali...
   A. Prediksi cuaca
   B. Deteksi penipuan
   C. Pengenalan wajah
   D. Menulis puisi
9. Dalam pengenalan pola, data yang relevan dikumpulkan untuk...
   A. Menemukan pola dalam data
   B. Membuat prediksi tentang masa depan
   C. Mengambil keputusan yang efektif
   D. Semua jawaban di atas benar
10. Manakah pernyataan yang paling tepat tentang pengenalan pola?
    A. Pengenalan pola hanya dapat digunakan untuk menganalisis data numerik.
    B. Pengenalan pola adalah teknik yang tidak efektif untuk memahami data yang kompleks.
    C. Pengenalan pola adalah teknik yang membantu menemukan makna dalam data dan membuat prediksi.
    D. Pengenalan pola hanya dapat digunakan oleh orang-orang yang ahli dalam ilmu data.
11. Abstraksi adalah proses...
    A. Menambahkan detail yang tidak relevan pada suatu masalah.
    B. Mengidentifikasi bagian-bagian penting dari suatu masalah dan mengabaikan detail yang tidak penting.
    C. Membuat masalah menjadi lebih kompleks dan sulit dipahami.
    D. Menghilangkan semua informasi yang terkait dengan suatu masalah.
12. Manakah contoh abstraksi yang tepat?
    A. Peta yang menunjukkan semua detail jalan, bangunan, dan pohon di suatu kota.

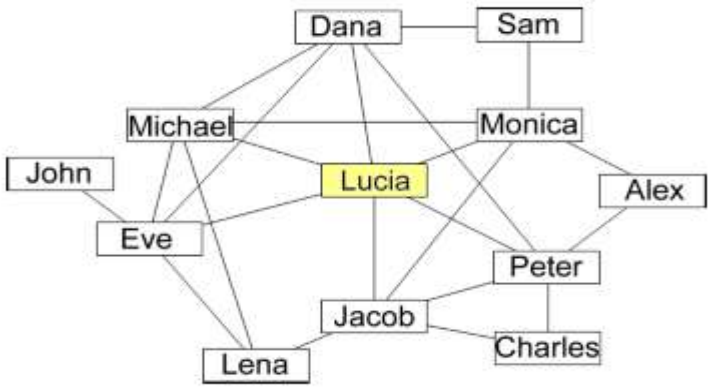
B. Model atom yang hanya menunjukkan proton, neutron, dan elektron.
C. Resep masakan yang menyertakan informasi tentang jenis panci dan spatula yang digunakan.
D. Definisi matematika yang rumit dengan banyak simbol dan rumus.

13. Abstraksi membantu kita dalam...
A. Membuat masalah menjadi lebih membingungkan.
B. Memahami masalah dengan lebih baik dan menemukan solusi yang lebih mudah.
C. Menghabiskan lebih banyak waktu untuk menyelesaikan masalah.
D. Menghindari mempelajari detail penting dari suatu masalah.

14. Abstraksi penting dalam...
A. Bermain game online.
B. Memasak makanan yang lezat.
C. Berlatih olahraga favorit.
D. Memecahkan masalah dan membuat keputusan.

15. Dalam abstraksi, kita harus fokus pada...
A. Semua detail, sekecil apa pun.
B. Bagian-bagian penting yang esensial untuk menyelesaikan masalah.
C. Informasi yang menarik dan menghibur.
D. Hal-hal yang tidak relevan dengan masalah.

16. Lucia dan teman-temannya terdaftar di sebuah jaringan media sosial, yang digambarkan sebagai "jaringan" sebagai berikut:



Sebuah garis berarti pertemanan antara dua orang. Contohnya Monica adalah teman Lucia tetapi Alex bukan teman Lucia. Aturan yang berlaku adalah:
Jika seseorang berbagi foto dengan dari temannya, maka temannya itu dapat mengomentarinya. Jika seseorang memberi komentar pada sebuah foto, maka semua teman-temannya dapat melihat komentar dan foto tersebut, tetapi tidak dapat mengomentarinya sampai mereka bisa. Lucia mengunggah sebuah foto. Dengan siapa dia harus berbagi agar Jacob tidak dapat melihatnya?
A. Dana, Michael, Eve
B. Dana, Eve, Monica
C. Michael, Eve, Jacob
D. Micheal, Peter, Monica

17. Seorang siswa mengumpulkan data dengan menyebarkan Google Forms kepada teman seangkatan. Data yang didapatkan siswa tersebut termasuk jenis data...
A. Sekunder
B. Primer
C. Kualitatif murni
D. Metadata

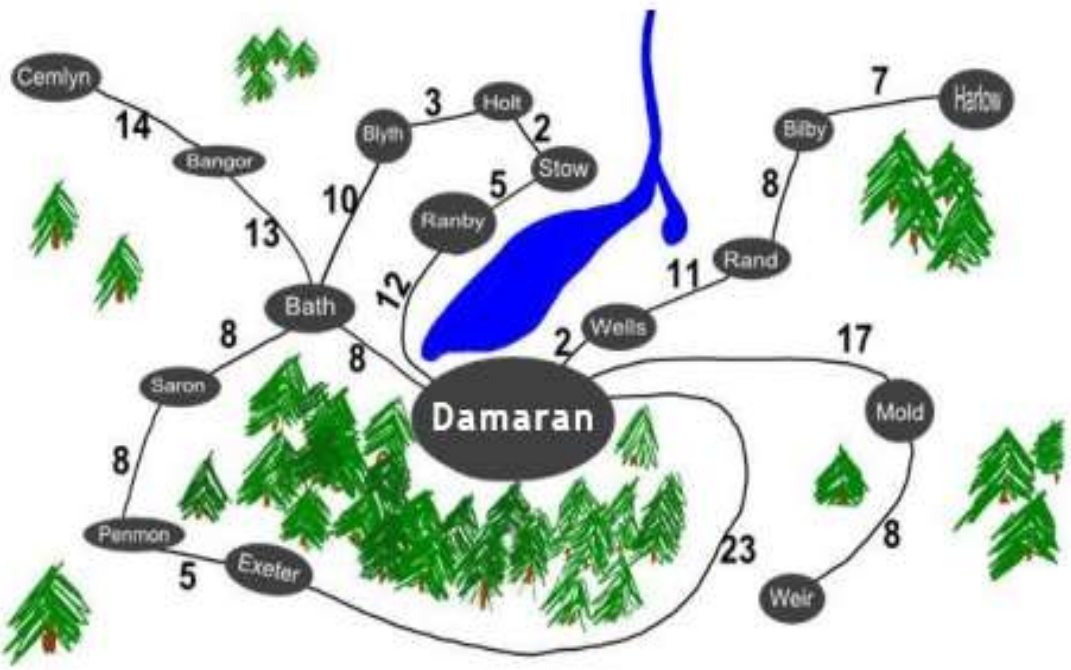
18. Jika ditemukan dua baris data yang memiliki nilai identik untuk seluruh kolom karena responden menekan tombol kirim formulir dua kali, tindakan pembersihan data yang tepat adalah...
A. Menghapus salah satu baris data ganda (Remove Duplicates)
B. Mengalikan kedua nilai data tersebut
C. Mengubah jenis datanya menjadi format mata uang
D. Mengisi bagian yang kosong dengan nilai rata-rata

19. Fitur pada spreadsheet yang digunakan untuk menyembunyikan data yang tidak memenuhi kriteria tertentu dinamakan...
A. Sorting
B. Filtering
C. Formatting
D. Shading

20. Jenis grafik yang paling tepat untuk menggambarkan tren naik-turunnya suhu udara di laboratorium sepanjang hari adalah...
    A. Pie Chart
    B. Line Chart
    C. Bar Chart
    D. Scatter Plot
21. Berdasarkan UU Pelindungan Data Pribadi, sebelum mempublikasikan hasil infografis survei sekolah, tindakan yang wajib dilakukan adalah...
    A. Menampilkan seluruh data nomor HP siswa agar transparan
    B. Melakukan anonimisasi dengan menghapus identitas pribadi sensitif
    C. Menjual data hasil survei ke pihak luar
    D. Mengubah seluruh data fiktif menjadi karangan
22. Sebuah sekolah ingin mengetahui penyebab menurunnya prestasi belajar siswa. Data yang dikumpulkan meliputi nilai ujian, kehadiran, dan lama penggunaan media sosial setiap hari. Langkah pertama yang paling tepat dilakukan adalah …
    A. Membuat diagram batang
    B. Menghapus data siswa
    C. Menentukan tujuan analisis data
    D. Membandingkan semua data
23. Seorang siswa menemukan beberapa data kosong pada tabel nilai kelas. Agar hasil analisis tetap akurat, tindakan terbaik adalah …
    A. Menghapus seluruh table
    B. Membersihkan dan melengkapi data
    C. Mengabaikan data kosong
    D. Langsung membuat grafik
24. Sebuah koperasi sekolah ingin mengetahui barang yang paling banyak dibeli siswa selama satu bulan. Jenis visualisasi yang paling sesuai untuk menunjukkan perbandingan jumlah barang adalah …
    A. Diagram batang
    B. Diagram pohon
    C. Diagram garis
    D. Flowchart
25. Sebuah tim OSIS membuat survei kegiatan ekstrakurikuler favorit siswa. Setelah data terkumpul, mereka langsung membuat diagram tanpa memeriksa data terlebih dahulu. Dampak yang mungkin terjadi adalah …
    A. Hasil analisis menjadi lebih cepat dan akurat
    B. Diagram menjadi lebih menarik
    C. Hasil visualisasi dapat mengandung kesalahan
    D. Data menjadi lebih lengkap

II.Jawablah pertanyaan-pertanyaan di bawah ini dengan tepat!

1. Analisis dan jelaskan bagaimana konsep berpikir komputasional dapat diterapkan dalam kehidupan sehari-hari untuk menyelesaikan masalah-masalah yang kompleks!
2. Jika Anda adalah seorang data scientist yang bekerja untuk sebuah perusahaan teknologi. Anda diminta untuk menganalisis data pengguna untuk meningkatkan pengalaman pengguna produk perusahaan. Bagaimana Anda dapat menerapkan pengenalan pola untuk membantu Anda dalam tugas ini? Jelaskan bagaimana Anda dapat menemukan pola dalam perilaku pengguna dan menggunakan pola tersebut untuk memberikan rekomendasi dan personalisasi yang lebih baik kepada pengguna!
3. Gambarkan graph sederhana rute perjalanan dari rumah ke sekolah !
4. Venaz tinggal di desa Damaran. Ia ingin mengundang teman- temannya untuk merayakan ulangtahunnya. Namun ia hanya mengundang teman-teman yang tinggal tidak lebih dari 20 km dari rumahnya agar tak kemalaman pulang. Jarak rumah teman- temannya dimunculkan dalam angka pada peta sebagai berikut.



Teman-teman yang diundang, dan rumahnya berjarak lebih dari 17 akan disediakan jemputan. Tentukan teman-teman Venaz yang diundang dan akan dijemput!

5. Uraikan apa yang dimaksud dengan istilah 'misleading charts' dalam visualisasi data dan mengapa tindakan tersebut melanggar etika publikasi data!

BAB II

BERPIKIR KOMPUTASIONAL DALAM BIDANG KEHIDUPAN

Tujuan Pembelajaran

Setelah mempelajari materi ”Berpikir Komputasional dalam Algoritma dan Pemrograman” ini, peserta didik diharapkan dapat:

1. Mengidentifikasi dan memecahkan masalah kompleks di lingkungan sekitar atau bidang kehidupan tertentu
2. Menerapkan algoritma optimasi untuk menyelesaikan masalah efisiensi dalam kehidupan sehari-hari.
3. Menguji dan mengevaluasi efektivitas berbagai alternatif solusi berbasis komputasional terhadap suatu isu sosial atau global sebelum mengambil keputusan.

Seiring berkembangnya teknologi di era globalisasi, kita sebagai manusia juga harus berkembang agar tidak tertinggal. Seperti komputer yang dapat menerima tugas dan menyelesaikannya dengan cepat, kita juga harus berpikir cepat dan mengembangkan hal agar dapat maju ke depan

Pola berpikir komputasional dapat dipraktikkan untuk memecahkan masalah yang biasa ditemukan dalam keseharian. Ada beberapa contoh berpikir komputasional dalam kehidupan sehari-hari. Sebelum itu, mari pahami dulu apa itu cara berpikir komputasional.

Cara berpikir komputasional adalah metode pemecahan masalah dengan menerapkan teknik ilmu komputer atau informatika. Dapat disimpulkan bahwa berpikir komputasional mendorong seseorang untuk mendekati masalah secara sistematis, serta perlu mengembangkan dan mengartikulasikan solusi dalam istilah yang cukup sederhana untuk dilakukan komputer atau orang lain.

Beberapa orang menganggap bahwa berpikir komputasional memerlukan penggunaan aplikasi komputer. Pada kenyataannya, setiap orang bisa berpikir komputasional tanpa menggunakan komputer.

Seiring berkembangnya dunia digital yang semakin rumit, konsep berpikir komputasional dapat membantu mengatasi berbagai tantangan dengan cara yang efektif dan mudah diterapkan. Berikut beberapa manfaat berpikir komputasional yang bisa dirasakan dalam kehidupan sehari-hari:

a. Meningkatkan kemampuan pemecahan masalah yang rumit dengan menciptakan solusi yang memungkinkan
b. Bermanfaat jika diterapkan dalam bidang sains dan teknik, serta pemasaran, penjualan, ilmu sosial, dan lainnya
c. Melatih kemampuan berpikir otak agar terbiasa berpikir secara matematis, kreatif, logis, dan terstruktur
d. Meningkatkan kemampuan berinovasi karena berpikir komputasional mendorong seseorang untuk mengembangkan berbagai solusi.

A. Komponen cara berpikir komputasional

Cara berpikir komputasional membantu seseorang untuk mendekati, memahami, menganalisis, dan menyelesaikan masalah secara efektif. Berikut empat komponen yang merupakan proses berpikir komputasional:

1. Dekomposisi

Dalam tahap dekomposisi, masalah diurai menjadi bagian-bagian yang lebih kecil sehingga lebih mudah dikelola. Hal ini membantu seseorang untuk lebih memahami masalah dan membuang informasi yang tidak relevan.

Contoh:

Masalah: Mengadakan pentas seni sekolah.

Dapat dipecah menjadi:

- Menentukan panitia

- Menentukan tempat
- Menyusun jadwal
- Menyiapkan perlengkapan
- Mengatur konsumsi
- Menentukan pengisi acara

Manfaat Dekomposisi

- Masalah menjadi lebih sederhana
- Mempermudah pembagian tugas
- Mempercepat penyelesaian masalah

2. Pengenalan pola

Pada tahap ini, dilakukan identifikasi pola atau hubungan antara berbagai bagian masalah. Tujuannya adalah untuk lebih menyederhanakan masalah, serta untuk mulai mengidentifikasi bagian masalah yang dapat dipecahkan dengan solusi yang sama.

Contoh:

- Lampu lalu lintas memiliki pola merah → kuning → hijau.
- Jadwal pelajaran sekolah memiliki pola mingguan.

Manfaat:

- Mempermudah prediksi
- Membantu menemukan solusi lebih cepat
- Mengurangi pekerjaan berulang

3. Abstraksi

Tahap abstraksi digunakan untuk mengidentifikasi informasi yang paling penting dalam suatu masalah, serta mengabaikan informasi yang tidak relevan. Hal ini berguna untuk menyederhanakan masalah yang rumit dan menciptakan lingkungan yang lebih efisien bagi individu untuk mengidentifikasi masalah yang dapat dipecahkan.

Contoh:

Dalam membuat peta sekolah, hanya ditampilkan:

- Gedung kelas
- Kantor
- Lapangan
- Perpustakaan

Sedangkan detail kecil seperti warna cat tembok tidak perlu ditampilkan.

Tujuan Abstraksi:

- Fokus pada inti masalah
- Mengurangi kompleksitas
- Mempermudah analisis

4. Desain algoritma

Desain algoritma merupakan tahapan terakhir di mana seseorang menghasilkan solusi langkah demi langkah guna memecahkan masalah. Berpikir algoritmik harus menghasilkan solusi yang dapat direplikasi untuk mendapatkan hasil yang dapat diprediksi dan diandalkan.

Contoh Algoritma Membuat Teh:

1. Siapkan gelas
2. Masukkan teh
3. Tambahkan gula
4. Tuangkan air panas
5. Aduk hingga rata
6. Sajikan

Ciri-ciri Algoritma

a. Memiliki langkah jelas
b. Sistematis
c. Logis

d. Memiliki tujuan akhir
e. Dapat diulang

B. Karakteristik cara berpikir komputasional

Meskipun berpikir komputasional artinya berpikir dengan meniru proses pemikiran komputer, perlu ditekankan bahwa manusia harus menerapkan cara berpikirnya sendiri. Oleh karena itu, manusia harus dapat memanfaatkan kemampuannya sendiri untuk memecahkan suatu masalah.

Adapun beberapa karakteristik dari berpikir komputasional meliputi hal-hal berikut:

a. Mampu menerapkan konsep-konsep berpikir komputasional agar proses pemecahan masalah dapat menghasilkan solusi yang efektif
b. Mampu mengelompokkan dan menganalisis data untuk menyelesaikan masalah secara tersusun dan terstruktur
c. Dapat mengabstraksikan data untuk mendapat gambaran singkat dan padat mengenai masalah yang sedang dihadapi
d. Mampu memahami permasalahan yang ada dan menghasilkan solusi yang efisien dan efektif
e. Dapat menggeneralisasi penyelesaian untuk berbagai masalah berbeda dengan menganalisis tiap-tiap bagian kecil permasalahan.

C. Contoh berpikir komputasiona dalam kehidupan sehari hari

Berpikir komputasional dapat diterapkan untuk memecahkan masalah yang biasa ditemukan di keseharian, baik yang sederhana maupun rumit. Berikut beberapa contoh cara berpikir komputasional dalam kehidupan sehari-hari:

1. Mencuci sepatu

Mencuci sepatu merupakan kegiatan yang sepertinya mudah, tetapi dapat menantang dalam kasus tertentu. Seperti mencuci sepatu di tengah musim hujan.

Untuk memecahkan masalah tersebut, dapat diterapkan pola berpikir komputasional. Pertama-tama, pahami dulu dampak apa saja yang timbul jika mencuci sepatu ketika musim hujan. Misalnya, hujan akan turun deras di sore hari. Untuk itu, sepatu harus dicuci di pagi hari dan sudah selesai sebelum siang hari agar bisa dijemur sebelum hujan turun.

2. Merencanakan perjalanan

Contoh berpikir komputasional berikutnya adalah merencanakan perjalanan. Seseorang yang hendak liburan memerlukan perencanaan yang matang, mulai dari waktu yang tepat, biaya atau tiket transportasi, hingga kegiatan apa saja yang akan dilakukan.

Mula-mula, pisahkan dulu berbagai hal yang akan dihadapi. Misalnya, tentukan kapan akan berlibur. Setelah itu, pikirkan biayanya, jika ingin mengendarai mobil maka harus menyiapkan uang bensin. Jika ingin naik transportasi publik, maka pertimbangkan harga tiket yang paling sesuai. Dilanjutkan dengan destinasi apa saja yang ingin dikunjungi dan kapan waktu yang tepat untuk itu. Prediksikan juga keramaian tempat dan cuaca untuk liburan tertentu.

3. Memperhitungkan jam berangkat kerja

Di Indonesia, khususnya Kudus, semua orang pasti merasakan sesaknya lalu lintas dengan kepadatan di mana-mana apalagi sekita pukul 06.30. Hal ini menyebabkan para pekerja kantoran dan karyawan pabrik yang masuk pukul 07.00 harus memperhitungkan dengan tepat waktu perjalanan mereka dari rumah ke kantor atau pabrik.

Pertama-tama, uraikan masalah yang mungkin dihadapi. Misalnya, jam berapa harus hadir di tempat kerja, di mana lokasi tempat kerja, dan kendaraan apa yang digunakan. Jika seseorang harus hadir di tempat kerja pukul 07.00 pagi, tempat kerja berlokasi di daerah macet atau melewati daerah padat lalu lintas, dan mengendarai mobil atau motor ke tempat kerja, maka solusinya adalah mencari tahu berapa lama perjalanan dengan mobil atau motor lalu tentukan jika ingin berangkat beberapa jam lebih awal.

4. Melukis

Merencanakan proses melukis juga merupakan contoh berpikir komputasional. Misalnya, seseorang pertama-tama harus memikirkan terlebih dahulu genre lukisan apa yang akan ia pilih. Kemudian, rencanakan desain dan tata letak seninya. Lalu, putuskan warna yang akan digunakan untuk melukis.

5. **Memecahkan soal matematika**

   Berpikir komputasional sebenarnya sudah sering kita terapkan tanpa sadar. Salah satu contohnya ialah memecahkan soal matematika. Menjawab soal matematika khususnya dalam bentuk cerita mengharuskan seseorang berpikir secara komputasional untuk mengidentifikasi masalah, membagi beberapa langkah pengerjaan, lalu menerapkan metode untuk menjawab soal tersebut.

6. **Menyiapkan presentasi**

   Contoh berpikir komputasional yang sering ditemukan dalam kehidupan sehari-hari adalah menyiapkan presentasi. Agar presentasi berjalan lancar, seseorang harus menyusun rencana langkah demi langkah.

   Ada banyak yang perlu diperhatikan, mulai dari materi, bagaimana cara menjelaskannya agar mudah dimengerti, desain PPT seperti apa yang menarik, berapa lama waktu yang dibutuhkan untuk pemaparan, hingga penampilan yang tepat untuk presentasi.

7. **Belanja bulanan**

   Bahkan, belanja bulanan pun harus dilakukan dengan menerapkan cara berpikir komputasional. Persiapan matang penting dilakukan sebelum belanja bulanan, jika tidak, seseorang yang impulsif akan membeli terlalu banyak produk tidak penting dan menjadi boros.

   Untuk itu, cari tahu dulu keperluan apa saja yang sudah habis lalu buat daftarnya di sebuah catatan. Pertimbangkan merek apa yang sesuai, baik dari segi harga maupun kualitas. Setelah itu, sesuaikan belanja bulanan dengan biaya yang ada.

8. **Memilih menu di restoran**

   Terkadang, Bunda mungkin bingung ingin memilih menu apa di restoran. Untuk memecahkan masalah tersebut, pola berpikir komputasional dapat diterapkan.

   Pertama-tama, identifikasi dulu masalah apa yang sedang dihadapi. Misalnya, seseorang merasa bingung dengan banyaknya menu yang tersedia. Lalu, bisa dipikirkan solusinya dengan mempertimbangkan menu apa yang paling baik untuk saat ini.

   Sebut saja menu A memiliki porsi besar yang dapat membuat kekenyangan. Untuk itu, pilihlah menu yang memiliki porsi lebih kecil. Selain itu, pertimbangkan juga lama waktu memasaknya, jika sedang terburu-buru maka pilihlah menu yang sekiranya dapat dengan cepat disajikan.

9. **Menulis makalah**

   Berikutnya, contoh berpikir komputasional sering diterapkan oleh pelajar yakni menulis *essay* atau makalah. Untuk menulis makalah, seseorang harus menemukan rumusan masalah, mengumpulkan data, menyajikan pembahasan, dan menghasilkan kesimpulan.

   Berpikir komputasional sangat digunakan dalam hal ini. Prosesnya mungkin tidak terlalu disadari, namun teknik berpikir komputasional membantu seseorang untuk memecahkan masalah, mengeksplorasi ide-ide baru, dan menghasilkan penemuan.

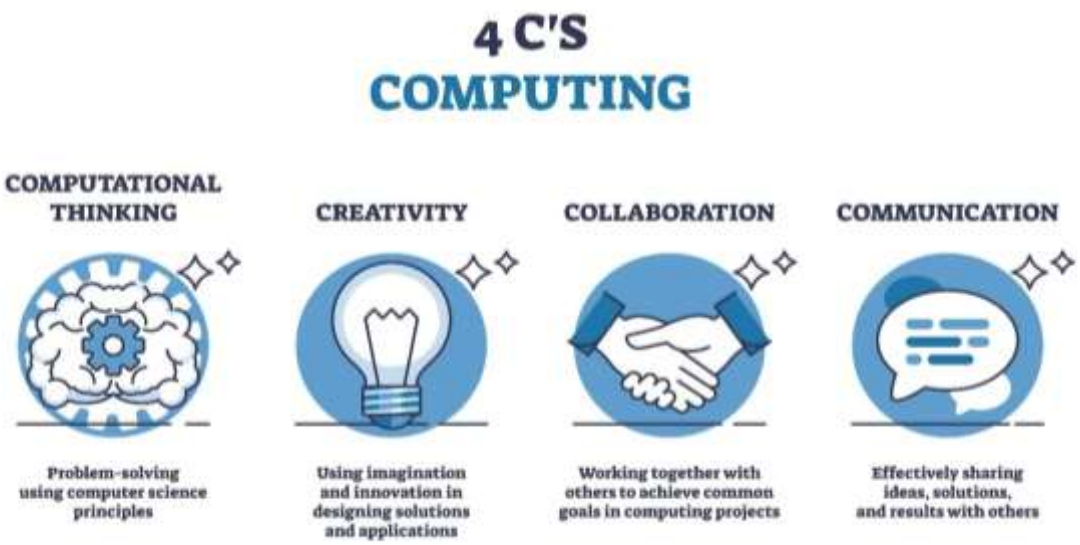
10. **Membuat Cheesecake**

    Contoh berpikir komputasional yang terakhir adalah membuat cheesecake. Hidangan pencuci mulut ini juga memerlukan perencanaan matang untuk membuatnya.

    Pertama-tama, uraikan dulu proses membuat cheesecake, yakni bahan dan peralatan apa yang diperlukan, lama waktu memasak, hingga kapan ingin menyajikannya. Misalnya, jika ingin memakan cheesecake di malam hari, maka usahakan untuk menyiapkannya di pagi atau siang hari karena cheesecake harus dibekukan dulu beberapa jam di dalam kulkas.

TUGAS 1

1. Dengan berpikir komputasional, denganmemperhitungkan waktu berangkat sekolah agar sebelum pukul 06.10 sudah sampai ke sekolah. Dan mempertimbangkan kepadatan lalu lintas dan jarak tempuh dari rumah ke sekolah.
   Buat solusi apabila menggunakan kendaraan sepeda dan diantar menggunakan kendaraan sepeda motor atau mobil.
2. Penumpukan sampah di lingkungan sekolah yang tidak kunjung selesai
   Identifikasi masalah dan siapa saja yang terlibat serta buat solusinya !



Sumber : *Shutterstock*
Gambar 2.1 : 4 pilar berpikir komputasional

D. Berpikir Komputasional Dalam Pemrograman

1. Penerapan berpikir komputasional dalam pemrograman membawa manfaat yang signifikan, seperti:
   a. **Meningkatkan Kemampuan Pemecahan Masalah:** Kita dapat memecahkan masalah secara sistematis dan terstruktur, sehingga lebih mudah menemukan solusi yang tepat.
   b. **Membuat Program yang Lebih Efisien:** Dengan memahami cara kerja algoritma, kita dapat merancang program yang lebih efisien dan efektif dalam menyelesaikan tugas.
   c. **Meningkatkan Keterampilan Koding:** Berpikir komputasional membantu kita memahami konsep-konsep dasar pemrograman, sehingga lebih mudah untuk belajar bahasa pemrograman baru.
   d. **Meningkatkan Kemampuan Berkomunikasi:** Berpikir komputasional membantu kita untuk menjelaskan ide dan solusi dengan cara yang terstruktur dan logis, baik dalam konteks pemrograman maupun dalam situasi lain.
   e. **Meningkatkan Keterampilan Kolaborasi:** Dalam proyek pemrograman yang kompleks, kemampuan untuk berpikir komputasional membantu tim untuk bekerja sama secara efektif dan menyelesaikan masalah bersama-sama

2. **Notasi Algoritma dalam Pemrograman**
   Notasi algoritma adalah cara untuk mencatat dan mendeskripsikan urutan langkah-langkah yang diperlukan untuk menyelesaikan suatu masalah. Dalam pemrograman, terdapat beberapa notasi algoritma yang umum digunakan, antara lain:

a. Bahasa Deskriptif

Bahasa deskriptif menggunakan kalimat dan kata-kata alami untuk menjelaskan langkah-langkah dalam algoritma. Bahasa ini mudah dipahami oleh manusia, namun tidak dapat langsung diinterpretasikan oleh komputer. Contohnya:

1. Mulai
2. Masukkan bilangan pertama
3. Masukkan bilangan kedua
4. Hitung jumlah bilangan pertama dan bilangan kedua
5. Tampilkan hasil penjumlahan
6. Selesai

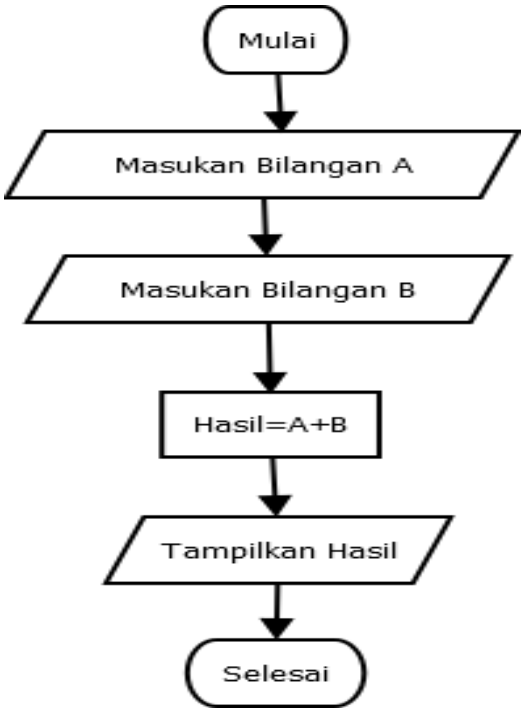
b. Pseudocode

Pseudocode adalah bahasa yang mirip dengan bahasa pemrograman, namun tidak terikat pada sintaks dan aturan tata bahasa yang ketat. Pseudocode lebih mudah dibaca dan dipahami oleh manusia dibandingkan dengan kode program, namun tetap dapat diinterpretasikan oleh programmer untuk diterjemahkan ke dalam bahasa pemrograman yang sebenarnya. Contohnya:

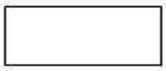
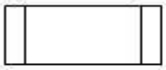
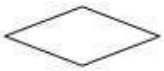
```
// Algoritma Penjumlahan Dua Bilangan
INPUT bilangan1, bilangan2
OUTPUT jumlah
// Hitung jumlah bilangan1 dan bilangan2
jumlah = bilangan1 + bilangan2
// Tampilkan hasil penjumlahan
TAMPILKAN jumlah
// Akhir algoritma
```

c. Flowchart

Flowchart adalah diagram yang menggunakan simbol-simbol untuk menggambarkan urutan langkah-langkah dalam algoritma. Flowchart mudah dipahami dan divisualisasikan, sehingga membantu programmer untuk memahami alur logika algoritma. Contohnya sebagai berikut:



Tabel simbol Flowchart beserta fungsinya

SIMBOL	NAMA	FUNGSI
	**TERMINATOR**	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	**PREPARATION**	Proses inisialisasi/pemberian harga awal
	**PROCESS**	Proses perhitungan/proses pengolahan data
	**INPUT/OUTPUT DATA**	Proses input/output data, parameter, informasi
	**PREDEFINED PROCESS (SUB PROGRAM)**	Permulaan sub program/proses menjalankan sub program
	**DECISION**	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	**ON PAGE CONNECTOR**	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	**OFF PAGE CONNECTOR**	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

TUGAS 2

Buat algoritma untuk menyelesaikan permalahan berikut dengan memilih notasi yang sesuai :

- Meminjam buku perpustakaan
- Menyalakan komputer
- Membuat akun email

2. **Contoh Penerapan Berpikir Komputasional di Scratch:**
    1. **Membuat Game Menembak Bola:**
        - **Abstraksi:** Mengidentifikasi elemen-elemen game (bola, target, skor).
        - **Algoritma:** Menyusun langkah-langkah untuk menggerakkan bola, mendeteksi tabrakan, dan menghitung skor.
        - **Dekomposisi:** Memecah game menjadi bagian-bagian kecil, seperti animasi bola, deteksi tabrakan, dan pembaruan skor.
        - **Pola:** Mengidentifikasi pola pengulangan untuk menggerakkan bola dan memperbarui skor.
    2. **Membuat Simulator Cuaca:**
        - **Algoritma:** Menyusun langkah-langkah untuk menampilkan elemen cuaca, mengubah kondisi cuaca, dan merespons interaksi pengguna.
        - **Dekomposisi:** Memecah simulator menjadi bagian-bagian kecil, seperti menampilkan elemen cuaca, mengubah kondisi cuaca, dan menangani interaksi pengguna.

- **Pola:** Mengidentifikasi pola dalam data cuaca untuk menghasilkan simulasi yang realistis.
- **Abstraksi:** Mengidentifikasi elemen-elemen cuaca (suhu, angin, hujan).

3. **Membuat Kalkulator Sederhana:**
   - Abstraksi: Input (angka, operasi matematika), Output (hasil perhitungan).
   - Algoritma: Menerima input, melakukan operasi matematika, menampilkan hasil.
   - Dekomposisi: Input, operasi, output.
   - Pola: Pengulangan untuk input dan operasi, kondisi untuk operasi.
4. **Membuat Simulator Antrian:**
   - Abstraksi: Input (jumlah pelanggan), Output (waktu tunggu rata-rata).
   - Algoritma: Simulasi antrian, hitung waktu tunggu, tampilkan hasil.
   - Dekomposisi: Simulasi antrian, hitung waktu tunggu, tampilkan hasil.
   - Pola: Pengulangan untuk simulasi antrian, kondisi untuk menghitung waktu tunggu.

TUGAS 2

1. Buatlah notasi algoritma dengan Bahasa Deskriptif, perkalian dua bilangan!
2. Buatlah notasi algoritma dengan Pseudocode menghitung luas segitiga!
3. Buatlah notasi algoritma dengan Flowchart memasak telur!
4. Buatlah aplikasi sederhana menghitung luas segitiga dengan **Scratch!**

E. Pengenalan Modularisasi Program

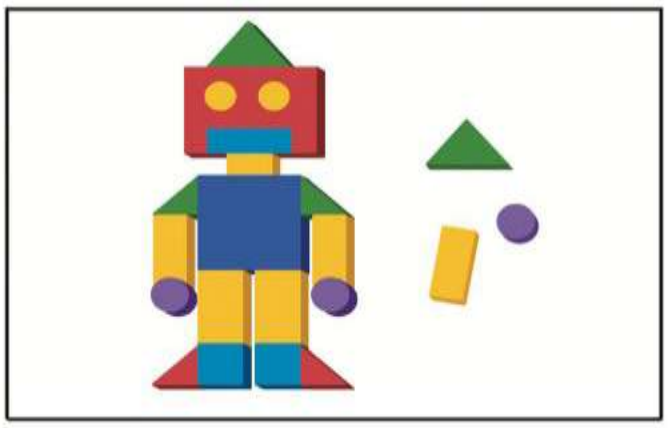
Modularisasi adalah teknik memecah sebuah program besar yang kompleks menjadi beberapa bagian kecil yang mandiri dan memiliki fungsi spesifik. Bagian kecil ini disebut sebagai 'modul'. Dalam pemrograman visual blok, modul ini diimplementasikan dalam bentuk Custom Blocks (Blok Buatan Sendiri) atau prosedur khusus (misalnya fitur 'Make a Block' pada Scratch).

1. **Mengapa Modularisasi Sangat Penting?**
   a. **Meningkatkan Keterbacaan (Readability):** Kode program tidak menumpuk dalam satu deretan panjang ke bawah. Program utama hanya memanggil nama modulnya saja, sehingga alur logika berpikir menjadi jauh lebih jernih.
   b. **Menghindari Duplikasi Kode (Reusability):** Jika ada perintah yang harus dijalankan berulang kali (misalnya animasi karakter melompat atau menghitung rumus matematika), kita tidak perlu menyusun ulang blok-blok tersebut. Cukup buat satu modul, lalu panggil modul tersebut kapan pun dibutuhkan.
   c. **Memudahkan Debugging (Pelacakan Eror):** Ketika program mengalami kesalahan atau 'bug', kita bisa langsung memeriksa modul spesifik yang bertanggung jawab, tanpa perlu mengaduk-aduk seluruh baris kode program.

2. **Parameter dalam Modularisasi**
   Modul yang baik sering kali membutuhkan input agar bisa bekerja secara dinamis. Input ini dinamakan parameter atau argumen. Sebagai contoh, jika kita membuat modul bernama [GambarPersegi], kita bisa menambahkan parameter bernama [UkuranSisi]. Ketika modul dipanggil, kita bisa menentukan apakah ingin menggambar persegi ukuran 50, 100, atau 200 tanpa perlu mengubah isi dalam modul dasar.

   Analogi Dunia Nyata: Bayangkan sebuah sepeda motor. Sepeda motor dibentuk dari modul-modul terpisah: modul mesin, modul roda, dan modul rem. Jika rem rusak, montir hanya perlu memperbaiki atau mengganti modul rem, tanpa harus membongkar silinder mesin. Itulah kekuatan modularisasi!

   Pernahkan kamu membuat robot-robotan dari sekumpulan blok kayu seperti ilustrasi pada gambar di bawah ini?



Gambar 2.3

Pada Gambar 2.3, kamu dapat melihat secara keseluruhan, bahwa tubuh robot tersusun dari beberapa blok kayu yang berbeda-beda. Namun, ada juga blok yang sama dan dapat digunakan kembali. Seperti blok kayu untuk tangan kanan dan kiri memiliki bentuk yang mirip, sehingga blok kayu tangan kanan dapat digunakan kembali untuk membentuk tangan kiri. Begitu juga blok kayu bagian kaki. Blok kayu pada kaki kanan dapat digunakan kembali untuk kaki kiri. Blok kayu yang dapat digunakan berulang kali disebut modul, yaitu komponen yang dapat digunakan kembali lebih dari satu kali.

Program komputer umumnya terdiri atas beberapa modul yang dapat digunakan kembali. Menurut Kamus Besar Bahasa Indonesia (KBBI), modul adalah komponen dari suatu sistem yang berdiri sendiri, tetapi menunjang program dari sistem tersebut. Seperti pada penggunaan blok kayu dalam penyusunan robot-robotan, modul program dapat digunakan berulang kali meskipun hanya ditulis sekali. Penggunaan modul secara berulang dapat diibaratkan seperti mencantumkan referensi saat menulis tugas esai sekolah. Cukup ditulis sekali, tetapi dapat dirujuk berkali-kali dalam teks. Agar dapat memahami modul program, ayo kerjakan aktivitas berikut ini. Ingat, perhatikan penamaan file aktivitas latihanmu pada program Scratch, agar
mudah digunakan dan dipelajari kembali.

F. Library (Pustaka)

Jika modul adalah komponen kecil di dalam satu berkas program, maka Library (Pustaka) adalah kumpulan dari modul-modul atau fungsi-fungsi yang dikelompokkan bersama dan disimpan agar bisa digunakan oleh program-program lain di luar program asalnya.

Dalam konteks pemrograman visual blok, Library sering disebut sebagai 'Ekstensi' (Extensions) atau 'Paket Blok Pustaka'. Library bertindak sebagai jembatan untuk memperluas kemampuan bawaan dari aplikasi pemrograman blok tersebut.

Perbedaan Blok Standar vs Blok Library:

Karakteristik	Blok Standar (Bawaan)	Blok Library (Pustaka/Ekstensi)
Ketersediaan	Langsung ada saat aplikasi pertama kali dibuka.	Harus dimuat atau diimpor terlebih dahulu sesuai kebutuhan.
Fungsi	Umum dan mendasar (logika perulangan, variabel, gerakan dasar).	Spesifik dan canggih (pengenalan suara, kecerdasan buatan, sensor IoT).
Beban Memori	Ringan karena merupakan fungsi inti.	Bisa menambah beban ukuran program jika terlalu banyak dimuat.

Contoh Blok	Move 10 steps, If-Else, Repeat, Set Variable.	Translate Text, Speech to Text, Draw Pen, Read Sensor.

Mengembangkan Library Sendiri

Pada tingkat lanjut, programmer tidak hanya menggunakan library buatan orang lain, tetapi juga mengeksplorasi cara membungkus custom blocks yang telah mereka buat menjadi sebuah pustaka (.json atau format ekstensi khusus) yang bisa dibagikan ke teman sekelas. Hal ini memupuk budaya kolaborasi dalam pengembangan perangkat lunak (open-source culture).

G. Penggunaan Library dalam Pemrograman Visual Blok

Menggunakan library dalam lingkungan blok visual umumnya melibatkan proses penambahan ekstensi melalui menu yang tersedia. Kita akan mengambil contoh generik yang biasa ditemukan pada platform seperti Scratch atau MakeCode.

Langkah-Langkah Menggunakan Library / Ekstensi:

a. **Langkah 1:** Buka lembar kerja pemrograman visual blok (misalnya Scratch). Di pojok kiri bawah, klik tombol 'Add Extension' (Tambah Ekstensi).
b. **Langkah 2:** Pilih kategori pustaka yang diinginkan. Contohnya, jika ingin membuat program menggambar otomatis, pilih library 'Pen' (Pena). Jika ingin membuat aplikasi multibahasa, pilih library 'Translate'.
c. **Langkah 3:** Setelah dipilih, kategori baru akan muncul di panel kode sebelah kiri dengan warna ikon yang khas. Blok-blok baru di dalam library tersebut kini siap digunakan.

Studi Kasus Praktis: Program Menggambar Geometri Otomatis

Tanpa library 'Pen', karakter sprite di Scratch hanya bergerak tanpa meninggalkan jejak. Namun, dengan mengimpor library 'Pen', kita mendapatkan akses ke perintah khusus seperti [erase all], [pen down], [set pen color], dan [pen up]. Kita dapat mengombinasikannya dengan modul buatan kita [GambarPersegi] untuk menggambar pola batik geometri yang sangat indah secara otomatis dan cepat.

Saat memanggil fungsi dari sebuah library, pastikan Anda memahami tipe data masukan yang diminta. Misalnya, library 'Translate' meminta masukan berupa teks kalimat dan target bahasa. Mengisi parameter dengan jenis data yang salah (misal memasukkan angka acak pada pilihan bahasa) akan menyebabkan program mengalami error runtime

UJI KOMPETENSI BAB. II
BERPIKIR KOMPUTASIONAL DALAM BIDANG KEHIDUPAN

I. Berilah tanda silang (X) pada huruf A, B, C, atau D pada jawaban yang paling tepat!

1. Cara berpikir untuk menyelesaikan masalah secara logis dan sistematis disebut ...
   A. Berpikir abstrak
   B. Berpikir sosial
   C. Berpikir komputasional
   D. Berpikir kreatif
2. Berikut yang termasuk komponen berpikir komputasional adalah ...
   A. Analisis pasar
   B. Dekomposisi
   C. Dekorasi
   D. Kolaborasi
3. Manfaat berpikir komputasional adalah ...
   A. Membuat masalah semakin rumit
   B. Mengurangi kemampuan berpikir
   C. Membantu menyelesaikan masalah secara efektif
   D. Menghilangkan kebutuhan analisis
4. Dalam membuat peta sekolah, hanya gedung penting yang ditampilkan. Hal tersebut merupakan contoh ...
   A. Algoritma
   B. Abstraksi
   C. Dekomposisi
   D. Debugging
5. Apakah seseorang harus selalu menggunakan komputer atau aplikasi tertentu untuk menerapkan berpikir komputasional?
   A. Ya, karena namanya meniru cara kerja komputer
   B. Ya, jika tidak menggunakan komputer maka bukan komputasional
   C. Tidak, setiap orang bisa berpikir komputasional tanpa menggunakan komputer
   D. Tidak, kecuali jika sedang menyelesaikan soal matematika yang sulit

6. Pilar berpikir komputasional yang mengurai masalah besar menjadi bagian-bagian yang lebih kecil agar lebih mudah dikelola dinamakan...
   A. Pengenalan pola
   B. Desain algoritma
   C. Abstraksi
   D. Dekomposisi
7. Tahap berpikir komputasional yang digunakan untuk mengidentifikasi informasi paling penting dan mengabaikan detail yang tidak relevan disebut...
   A. Abstraksi
   B. Pengenalan pola
   C. Dekomposisi
   D. Desain algoritma
8. Panitia OSIS ingin mengadakan acara pentas seni sekolah. Mereka membagi rencana kerja menjadi: menentukan panitia, menentukan tempat, menyusun jadwal, dan mengatur konsumsi. Tindakan ini merupakan manfaat dari dekomposisi, yaitu...
   A. Mempermudah prediksi kejadian di masa depan
   B. Mengurangi pekerjaan yang dilakukan secara berulang-ulang
   C. Masalah menjadi lebih sederhana dan mempermudah pembagian tugas
   D. Menghilangkan informasi yang tidak penting dari inti masalah
9. Ketika seseorang memilih berangkat lebih awal agar tidak terlambat akibat macet, maka ia sedang menerapkan ...
   A. Pengenalan suara
   B. Pengolahan gambar
   C. Berpikir komputasional
   D. Sistem operasi
10. SMP 1 Jekulo mengalami penumpukan sampah setiap hari. Langkah berpikir komputasional yang paling tepat dilakukan pertama kali adalah ...
    A. Membeli tempat sampah baru
    B. Mengidentifikasi penyebab penumpukan sampah
    C. Menghukum siswa yang membuang sampah
    D. Menutup kantin sekolah
11. Rina ingin membuat aplikasi penghitung luas segitiga di Scratch. Agar program mudah diperbaiki jika terjadi kesalahan, langkah terbaik adalah ..
    A. Menulis seluruh kode dalam satu blok

B. Menghapus variable
C. Membagi program menjadi beberapa modul
D. Menggunakan warna berbeda

12. Perhatikan permasalahan berikut!

"Seorang siswa sering terlambat sekolah karena jalan yang dilalui macet."

Solusi paling efektif berdasarkan berpikir komputasional adalah ...

A. Tetap berangkat pada jam yang sama
B. Tidak masuk sekolah
C. Mencari rute alternatif dan memperkirakan waktu tempuh
D. Menyalahkan pengguna jalan lain

13. Mengapa penerapan modularisasi dalam pemrograman dapat menghindari duplikasi kode (Reusability)?
A. Karena program utama tidak perlu memanggil modul sama sekali
B. Karena jika ada perintah berulang, kita hanya perlu membuat satu modul lalu memanggilnya kapan pun dibutuhkan
C. Karena modularisasi otomatis menghapus kode program yang salah
D. Karena memori yang dibutuhkan modul jauh lebih kecil daripada blok biasa

14. **Perhatikan cuplikan notasi berikut:**

```
INPUT bilangan1, bilangan2
jumlah = bilangan1 + bilangan2

TAMPILKAN jumlah
```

Kelebihan utama dari notasi di atas dibandingkan dengan bahasa deskriptif…
A. Menggunakan gambar simbol visual sehingga lebih menarik
B. Sangat kaku karena terikat pada aturan sintaks bahasa pemrograman tertentu
C. Lebih mudah dibaca manusia tetapi tidak dapat dipahami oleh programmer
D. Lebih mudah diinterpretasikan oleh programmer untuk diterjemahkan ke bahasa pemrograman yang sebenarnya

15. Seorang programmer menggunakan satu modul yang sama untuk menghitung luas persegi dengan ukuran berbeda. Hal tersebut menunjukkan manfaat ..
A. Debugging
B. Flowline
C. Reusability
D. Dekorasi

16. Jika sebuah program mengalami error runtime karena parameter salah, maka penyebabnya adalah ..
A. Input tidak sesuai tipe data
B. Flowchart terlalu kecil
C. Komputer terlalu cepat
D. Program memiliki warna berbeda

17. Mengapa modularisasi penting dalam pemrograman kompleks?
A. Agar kode semakin Panjang
B. Agar program sulit dipahami
C. Agar program lebih mudah dikembangkan dan diperbaiki
D. Agar tidak dapat digunakan ulang

18. Dalam menghadapi isu penyebaran hoaks di media sosial, solusi berbasis komputasional yang paling tepat adalah ...
A. Menyebarkan ulang informasi
B. Membiarkan informasi tanpa verifikasi
C. Menggunakan aplikasi pemeriksa fakta dan validasi sumber
D. Menghapus semua media sosial

19. Di kota Kudus, kemacetan lalu lintas kerap terjadi sekitar pukul 06.30 pagi. Seorang karyawan pabrik yang harus masuk kerja pukul 07.00 menggunakan berpikir komputasional untuk memecahkan masalah ini. Jika ia menerapkan pilar "Pengenalan Pola" dan "Abstraksi", tindakan manakah yang paling mencerminkan kedua pilar tersebut?

A. Menghitung detail biaya bensin bulanan dan membeli motor baru
B. Mengingat bahwa kemacetan selalu berpola di jam 06.30 pada jalur utama, lalu mengabaikan rute sekunder yang jauh namun fokus mencari jalan tikus yang terbukti selalu lengang
C. Memecah masalah menjadi jam bangun tidur, jenis sarapan, dan warna baju yang dipakai kerja
D. Menulis langkah demi langkah cara menghidupkan motor dari kunci kontak hingga menarik gas

20. Saat merancang game menembak bola di Scratch, Anda membagi proyek menjadi animasi bola, deteksi tabrakan, dan pembaruan skor. Mengapa pendekatan dekomposisi ini dianggap lebih efektif dalam pemecahan masalah kompleks dibanding langsung menyatukan seluruh kode?
    A. Karena membuat program utama berjalan lebih lambat sehingga mudah diamati
    B. Karena memecah game menjadi bagian kecil membuat logika masalah lebih mudah dikelola, dipahami, dan dikerjakan secara fokus
    C. Karena dekomposisi otomatis menghilangkan kebutuhan untuk menggunakan library pihak ketiga
    D. Karena dengan dekomposisi, kita tidak perlu lagi mendesain langkah algoritma
21. Anda membuat Custom Block bernama [GambarPersegi] dengan parameter [UkuranSisi]. Analisis apa yang akan terjadi pada program Anda jika Anda menghilangkan parameter [UkuranSisi] dari modul tersebut?
    A. Modul tidak akan bisa dipanggil lagi oleh program utama
    B. Modul kehilangan sifat dinamisnya, sehingga hanya bisa menggambar persegi dengan satu ukuran yang kaku/tetap
    C. Modul secara otomatis menggambar berbagai jenis geometri lain seperti segitiga dan lingkaran
    D. Program Scratch akan mendeteksi *error runtime* dan langsung menutup aplikasi

22. Materi memberikan analogi montir motor yang hanya perlu memeriksa atau mengganti modul rem jika remnya rusak, tanpa harus membongkar silinder mesin. Jika prinsip ini diterapkan dalam penanganan eror (*debugging*) program komputer, apa keuntungan terbesar yang didapatkan oleh programmer?
    A. Programmer tidak perlu lagi mempelajari bahasa pemrograman baru
    B. Pelacakan eror menjadi lebih efisien karena programmer cukup memeriksa modul spesifik yang bermasalah tanpa mengaduk-aduk seluruh kode program
    C. Program akan terbebas dari serangan virus komputer selamanya
    D. Kode program utama akan terhapus otomatis sehingga memori menjadi bersih
23. Seorang pelajar harus menulis makalah sekolah dengan menentukan rumusan masalah, mengumpulkan data, menyajikan pembahasan, dan menarik kesimpulan. Mengapa aktivitas menulis makalah ini disebut menggunakan teknik berpikir komputasional?
    A. Karena menulis makalah harus diketik menggunakan software Microsoft Word di komputer
    B. Karena prosesnya membantu memecahkan masalah rumit secara sistematis, mengeksplorasi ide baru, dan menghasilkan penemuan terstruktur
    C. Karena kesimpulan makalah harus berupa kode-kode bahasa pemrograman
    D. Karena mengumpulkan data makalah sama ringannya dengan memuat blok standar
24. Seseorang yang impulsif saat melakukan belanja bulanan akan membeli terlalu banyak produk tidak penting dan menjadi boros. Solusi preventif terbaik dengan menerapkan pilar "Abstraksi" dan "Desain Algoritma" sebelum pergi ke toko adalah...
    A. Membawa uang sebanyak-banyaknya lalu memilih barang termurah yang langsung terlihat di rak toko
    B. Mencari tahu keperluan yang habis, membuat daftar catatan belanja (mengabaikan barang promo yang tidak perlu), lalu menyusun langkah belanja sesuai anggaran biaya yang ada
    C. Meminta kasir toko untuk menghitung seluruh total harga barang yang ada di dalam toko
    D. Memecah tugas belanja kepada seluruh anggota keluarga agar belanjaan cepat selesai
25. Di tingkat lanjut, programmer dapat membungkus custom blocks yang mereka buat menjadi sebuah pustaka (.json) untuk dibagikan ke teman sekelas. Berdasarkan materi, apa dampak sosial-kolaboratif yang paling positif dari tindakan ini?
    A. Membuat teman sekelas tidak perlu belajar memprogram lagi karena tinggal memakai
    B. Memupuk budaya kolaborasi pengembangan perangkat lunak (open-source) sehingga proyek kelompok dapat diselesaikan lebih efisien secara bersama-sama

C. Menurunkan beban memori komputer masing-masing siswa secara otomatis
D. Mengubah bahasa pemrograman visual Scratch menjadi bahasa pemrograman berbasis teks kaku

II. Jawablah pertanyaan berikut dengan jelas

1. Di lingkungan sekolah sering terjadi keterlambatan siswa saat upacara hari Senin karena jalan menuju sekolah macet.
   Analisislah permasalahan tersebut menggunakan langkah berpikir komputasional yang meliputi:
   a. Dekomposisi
   b. Pengenalan pola
   c. Abstraksi
   d. Algoritma solusi
2. SMP 1 Jekulo mengalami masalah penumpukan sampah plastik setiap hari. Buatlah solusi berbasis komputasional untuk mengatasi masalah tersebut!
3. Jelaskan bagaimana modularisasi dapat membantu programmer dalam membuat program yang kompleks!
4. Bandingkan penggunaan pseudocode dan flowchart dalam penulisan algoritma!
5. Mengapa kemampuan berpikir komputasional penting dimiliki pelajar di era digital saat ini?

BAB III
DASAR PEMROGRAMAN: TEKS (PSEUDOCODE)

Tujuan Pembelajaran

Setelah mengikuti pembelajaran ini, murid mampu:

1. Menjelaskan pengertian pseudocode dengan bahasa sendiri.
2. Mengidentifikasi fungsi pseudocode dalam proses pembuatan program.
3. Membedakan algoritma dalam bahasa sehari-hari, flowchart, pseudocode, dan program.
4. Mengenali aturan dasar penulisan pseudocode.
5. Menulis pseudocode sederhana menggunakan struktur urutan instruksi.
6. Menulis pseudocode yang menggunakan masukan, proses, keluaran, percabangan dan pengulangan sederhana.
7. Mengubah masalah sehari-hari menjadi langkah-langkah algoritmik dalam bentuk pseudocode.

A. PENGERTIAN PSEUDOCODE

Pseudocode adalah cara menuliskan langkah-langkah penyelesaian masalah menggunakan bahasa sederhana yang menyerupai bahasa pemrograman, tetapi belum menjadi program yang dapat dijalankan oleh komputer.

Kata *pseudo* berarti "semu" atau "tiruan", sedangkan *code* berarti "kode". Jadi, pseudocode dapat diartikan sebagai **kode semu**.

Pseudocode digunakan untuk merancang algoritma sebelum ditulis ke dalam bahasa pemrograman yang sebenarnya.

Contoh pseudocode sederhana:

```
Mulai
Tampilkan "Halo, Murid Kelas IX"
Selesai
```

Pseudocode tersebut belum dapat dijalankan langsung oleh komputer, tetapi sudah menunjukkan urutan instruksi yang jelas.

B. FUNGSI PSEUDOCODE

Pseudocode memiliki beberapa fungsi penting, yaitu:

1. **Membantu menyusun logika program**
   Menentukan langkah-langkah penyelesaian masalah sebelum menulis kode.
2. **Memudahkan komunikasi**
   Pseudocode mudah dibaca oleh orang yang belum menguasai bahasa pemrograman tertentu.
3. **Mengurangi kesalahan saat membuat program**
   Jika alur sudah dirancang dengan baik, proses penulisan program menjadi lebih mudah.
4. **Menjadi jembatan antara algoritma dan kode program**
   Pseudocode berada di antara bahasa sehari-hari dan bahasa pemrograman.
5. **Membantu memperbaiki program**
   Jika program tidak berjalan sesuai harapan, murid dapat memeriksa kembali logika pada pseudocode.

C. PERBEDAAN ALGORITMA, PSEUDOCODE, DAN PROGRAM

BENTUK	PENGERTIAN	CONTOH
Algoritma	Langkah-langkah untuk menyelesaikan masalah	Ambil gelas, masukkan teh, tuangkan air panas
Pseudocode	Penulisan algoritma yang menyerupai bahasa pemrograman	INPUT nilai, HITUNG rata-rata, TAMPILKAN hasil

Program	Instruksi yang ditulis dengan bahasa pemrograman dan dapat dijalankan komputer	Kode Python, Java, Scratch, dan lain-lain

Contoh perbandingan:
Algoritma bahasa **sehari-hari:**

1. Masukkan nilai pertama.
2. Masukkan nilai kedua.
3. Jumlahkan kedua nilai.
4. Tampilkan hasilnya.

Pseudocode:	**Contoh dalam Python:**
Mulai INPUT nilai1 INPUT nilai2 hasil ← nilai1 + nilai2 TAMPILKAN hasil Selesai	nilai1 = int(input("Masukkan nilai pertama: ")) nilai2 = int(input("Masukkan nilai kedua: ")) hasil = nilai1 + nilai2 print(hasil)

D. Aturan Dasar Penulisan Pseudocode

Tidak ada satu aturan yang benar-benar wajib untuk semua pseudocode. Namun, agar mudah dibaca, perlu menggunakan aturan yang konsisten.

1. Gunakan Kata Kunci yang Jelas

Beberapa kata kunci yang sering digunakan:

KATA KUNCI	FUNGSI
MULAI	Menandai awal algoritma
SELESAI	Menandai akhir algoritma
INPUT	Memasukkan data
OUTPUT / TAMPILKAN	Menampilkan hasil
BACA	Membaca data dari pengguna
JIKA	Memulai percabangan
MAKA	Menunjukkan tindakan jika kondisi benar
JIKA TIDAK	Menunjukkan tindakan jika kondisi salah
ULANGI	Memulai pengulangan
SELAMA	Menjalankan pengulangan berdasarkan kondisi
UNTUK	Melakukan pengulangan dengan jumlah tertentu

2. Gunakan Nama Variabel yang Mudah Dipahami

Variabel adalah tempat untuk menyimpan data.

Contoh nama variabel yang baik:	Contoh nama variabel yang kurang baik:
namaMurid nilaiMatematika jumlahBelanja rataRata umur	a x data1 z

Nama variabel sebaiknya menggambarkan isi data yang disimpan.

3. Gunakan Urutan yang Logis

Instruksi harus ditulis secara berurutan. Komputer akan menjalankan perintah dari atas ke bawah.

Contoh yang benar:	Contoh yang kurang tepat:
Mulai INPUT panjang INPUT lebar luas ← panjang × lebar TAMPILKAN luas Selesai	Mulai TAMPILKAN luas luas ← panjang × lebar INPUT panjang INPUT lebar Selesai

Pada contoh kedua, nilai luas belum dihitung saat akan ditampilkan.

4. Gunakan Indentasi

Indentasi adalah penulisan menjorok ke dalam untuk menunjukkan bagian instruksi yang berada di dalam percabangan atau pengulangan.

Contoh:

```
JIKA nilai >= 75 MAKA
    TAMPILKAN "Lulus"
JIKA TIDAK
    TAMPILKAN "Belum Lulus"
```

Indentasi membantu murid membaca struktur program dengan lebih mudah.

PRAKTIK 1

Buatlah pseudocode untuk menghitung luas persegi.

Petunjuk

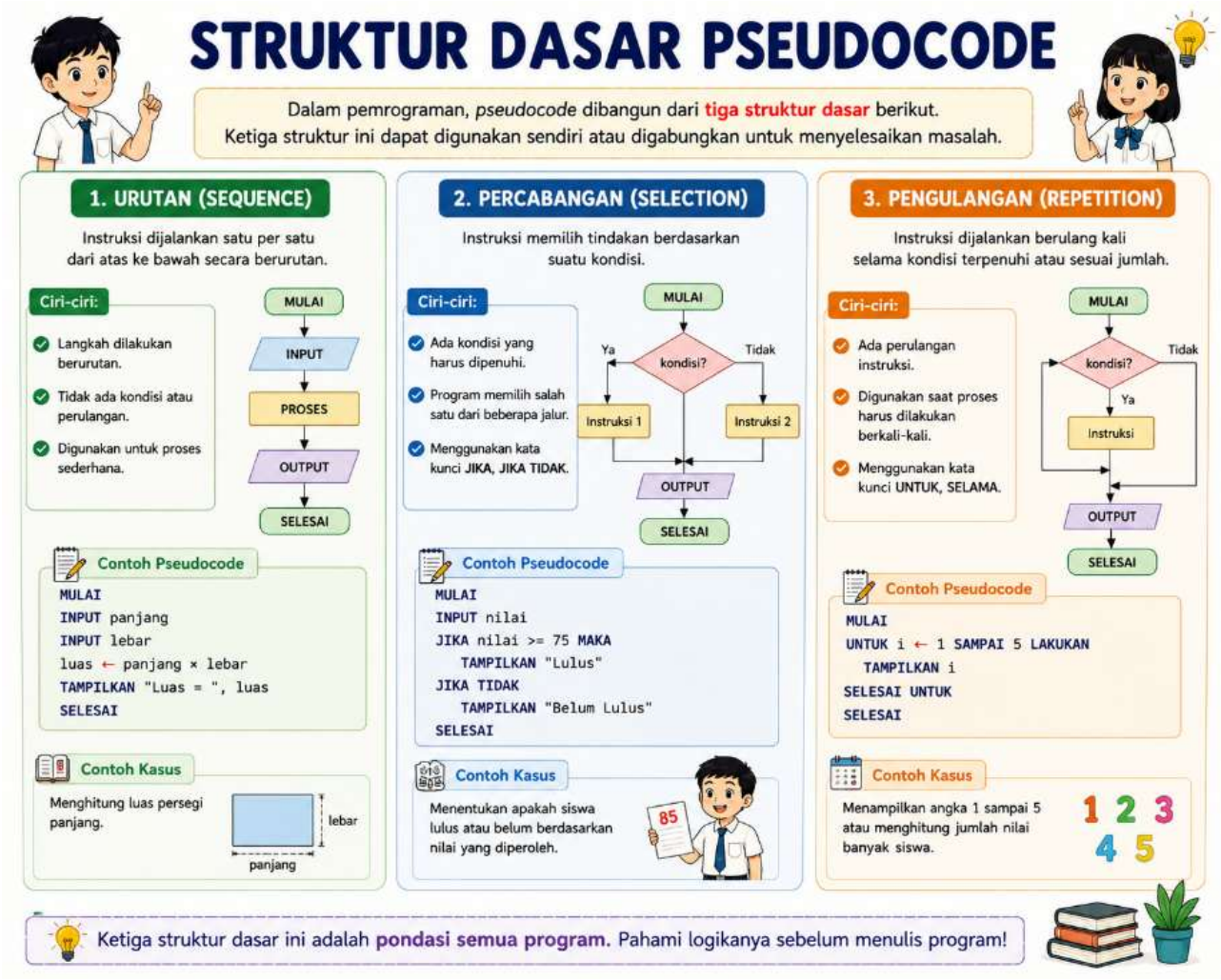
- Input: panjang sisi persegi.
- Proses: sisi × sisi.
- Output: luas persegi.

E. STRUKTUR DASAR PSEUDOCODE

Dalam pemrograman, terdapat tiga struktur utama:

1. **Urutan (Sequence)**
2. **Percabangan (Selection)**
3. **Pengulangan (Repetition)**

Ketiga struktur tersebut menjadi dasar hampir semua program komputer.



F. STRUKTUR URUTAN (SEQUENCE)

1. Pengertian

Struktur urutan adalah instruksi yang dijalankan satu per satu dari atas ke bawah.
Contoh kegiatan sehari-hari:

1. Membuka buku.
2. Membaca materi.
3. Mengerjakan latihan.
4. Mengumpulkan tugas.

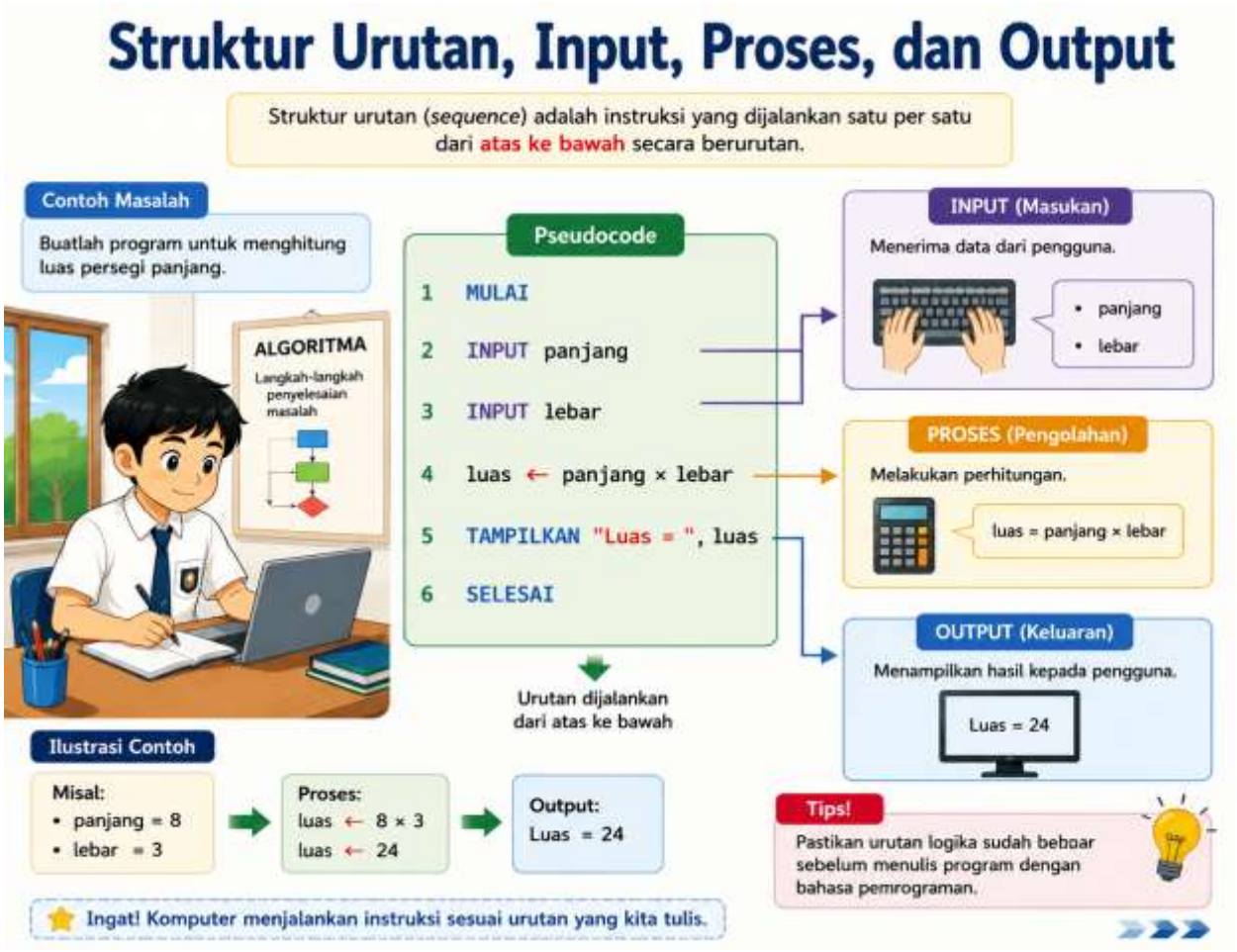
Contoh pseudocode:

Mulai
Tampilkan "Selamat Datang"
Tampilkan "Mari Belajar Pseudocode"
Tampilkan "Semangat Belajar"
Selesai

2. Input, Proses, dan Output

Dalam sebuah program sederhana biasanya terdapat tiga bagian:

- **Input:** data yang dimasukkan.
- **Proses:** pengolahan data.
- **Output:** hasil yang ditampilkan.



Contoh menghitung luas persegi panjang:

Mulai
INPUT panjang
INPUT lebar
luas ← panjang × lebar
TAMPILKAN "Luas persegi panjang = ", luas
Selesai

Keterangan:

BAGIAN	ISI
Input	panjang dan lebar
Proses	luas = panjang × lebar
Output	luas persegi panjang

3. Contoh Pseudocode Menghitung Keliling Persegi

Mulai
INPUT sisi
keliling ← 4 × sisi
TAMPILKAN "Keliling persegi = ", keliling
Selesai
Jika nilai sisi = 8, maka hasilnya adalah:
Keliling persegi = 32

4. Contoh Pseudocode Menghitung Rata-Rata Nilai

Mulai
INPUT nilai1
INPUT nilai2

```
INPUT nilai3
jumlah ← nilai1 + nilai2 + nilai3
rataRata ← jumlah / 3
TAMPILKAN "Rata-rata nilai = ", rataRata
Selesai
```

G. OPERATOR DALAM PSEUDOCODE

Operator adalah simbol yang digunakan untuk melakukan operasi.

1. Operator Aritmetika

OPERATOR	FUNGSI	CONTOH
+	Penjumlahan	5 + 3
-	Pengurangan	8 – 2
× atau *	Perkalian	4 × 6
/	Pembagian	10 / 2
MOD	Sisa hasil bagi	10 MOD 3

Contoh:
sisa ← 10 MOD 3
Hasilnya adalah 1.

2. Operator Perbandingan

OPERATOR	ARTI
=	Sama dengan
≠	Tidak sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

Contoh:
nilai >= 75
Artinya nilai lebih besar atau sama dengan 75.

3. Operator Logika

OPERATOR	ARTI
DAN	Kedua kondisi harus benar
ATAU	Salah satu kondisi benar
TIDAK	Membalik kondisi

Contoh:

```
JIKA nilai >= 75 DAN kehadiran >= 80 MAKA
   TAMPILKAN "Lulus"
```

TUGAS 1

1. Jelaskan pengertian pseudocode dengan bahasa sendiri.
2. Sebutkan tiga manfaat pseudocode.
3. Jelaskan perbedaan antara algoritma dan program.
4. Buatlah pseudocode untuk menghitung keliling persegi panjang.
5. Buatlah pseudocode untuk menentukan apakah sebuah bilangan positif, negatif, atau nol.

H. STRUKTUR PERCABANGAN

1. Pengertian Percabangan

Percabangan digunakan ketika program harus memilih tindakan berdasarkan suatu kondisi.
Contoh dalam kehidupan sehari-hari:

- Jika hujan, gunakan payung.
- Jika tidak hujan, tidak perlu membawa payung.

2. Percabangan Satu Kondisi

```
Mulai
INPUT nilai
JIKA nilai >= 75 MAKA
   TAMPILKAN "Selamat, kamu lulus"
SELESAI
```

Pada pseudocode tersebut, pesan hanya muncul jika nilai murid minimal 75.

3. Percabangan Dua Kondisi

```
Mulai
INPUT nilai
JIKA nilai >= 75 MAKA
   TAMPILKAN "Lulus"
JIKA TIDAK
   TAMPILKAN "Belum Lulus"
SELESAI
```

4. Contoh Menentukan Bilangan Genap atau Ganjil

Mulai
INPUT bilangan
JIKA bilangan MOD 2 = 0 MAKA
  TAMPILKAN "Bilangan Genap"
JIKA TIDAK
  TAMPILKAN "Bilangan Ganjil"
SELESAI
	Penjelasan:
	Jika sisa pembagian bilangan dengan 2 adalah 0, maka bilangan tersebut genap.

5. Contoh Menentukan Kategori Nilai

```
Mulai
INPUT nilai
JIKA nilai >= 90 MAKA
   TAMPILKAN "Sangat Baik"
JIKA TIDAK JIKA nilai >= 75 MAKA
   TAMPILKAN "Baik"
JIKA TIDAK JIKA nilai >= 60 MAKA
   TAMPILKAN "Cukup"
JIKA TIDAK
   TAMPILKAN "Perlu Bimbingan"
SELESAI
```

PRAKTIK 2

Buatlah pseudocode untuk menentukan apakah suhu tubuh seseorang normal atau demam.
Ketentuan:

- Jika suhu lebih dari atau sama dengan 38, tampilkan "Demam".
- Jika suhu kurang dari 38, tampilkan "Normal".

I. STRUKTUR PENGULANGAN

1. Pengertian Pengulangan

Pengulangan digunakan ketika suatu instruksi perlu dilakukan lebih dari satu kali.
Contoh kegiatan sehari-hari:

- Menulis nama sebanyak 10 kali.
- Menghitung angka 1 sampai 10.
- Memanggil nomor absen murid satu per satu.

2. Pengulangan UNTUK

Digunakan jika jumlah pengulangan sudah diketahui.

Mulai UNTUK i ← 1 SAMPAI 5 LAKUKAN TAMPILKAN "Belajar Pseudocode" SELESAI UNTUK Selesai	Hasil: Belajar Pseudocode Belajar Pseudocode Belajar Pseudocode Belajar Pseudocode Belajar Pseudocode

3. Pengulangan SELAMA

Digunakan jika pengulangan bergantung pada kondisi tertentu.

Mulai
angka ← 1
SELAMA angka <= 5 LAKUKAN
  TAMPILKAN angka
  angka ← angka + 1
SELESAI SELAMA
Selesai
	Hasil:
	1
	2
	3
	4
	5

4. Contoh Menghitung Jumlah Bilangan 1 sampai 10

Mulai jumlah ← 0 UNTUK i ← 1 SAMPAI 10 LAKUKAN jumlah ← jumlah + i SELESAI UNTUK TAMPILKAN "Jumlah = ", jumlah Selesai	Hasilnya: Jumlah = 55

PRAKTIK 3

Buatlah pseudocode untuk menampilkan angka 1 sampai 10.

J. CONTOH PENERAPAN PSEUDOCODE DALAM KEHIDUPAN SEHARI-HARI

1. Membuat Teh

```
Mulai
Siapkan gelas
Masukkan gula ke dalam gelas
Masukkan teh ke dalam gelas
Tuangkan air panas
Aduk teh
Tampilkan "Teh siap disajikan"
Selesai
```

2. Menentukan Diskon Belanja

```
Mulai
INPUT totalBelanja

JIKA totalBelanja >= 100000 MAKA
   diskon ← totalBelanja × 10 / 100
JIKA TIDAK
   diskon ← 0

bayar ← totalBelanja - diskon

TAMPILKAN "Diskon = ", diskon
TAMPILKAN "Total bayar = ", bayar
Selesai
```

3. Menghitung Total Belanja

```
Mulai
INPUT hargaBarang
INPUT jumlahBarang
total ← hargaBarang × jumlahBarang
TAMPILKAN "Total belanja = ", total
Selesai
```

Langkah Menyusun Pseudocode

Murid dapat mengikuti langkah berikut saat membuat pseudocode:

1. Pahami masalah yang diberikan.
2. Tentukan data masukan yang diperlukan.
3. Tentukan proses yang harus dilakukan.
4. Tentukan hasil yang ingin ditampilkan.
5. Pilih struktur yang sesuai: urutan, percabangan, atau pengulangan.
6. Tuliskan pseudocode secara runtut.
7. Periksa kembali logika instruksi.
8. Lakukan uji coba menggunakan contoh data.

K. CONTOH ANALISIS MASALAH

Masalah

Buatlah pseudocode untuk menentukan apakah seorang murid dinyatakan lulus atau belum lulus. Murid dinyatakan lulus jika nilai minimal 75.

Analisis

KOMPONEN	ISI
Input	Nilai murid
Proses	Membandingkan nilai dengan 75
Output	Lulus atau Belum Lulus

Pseudocode

```
Mulai
INPUT nilai
JIKA nilai >= 75 MAKA
   TAMPILKAN "Lulus"
JIKA TIDAK
   TAMPILKAN "Belum Lulus"
Selesai
```

L. KESALAHAN UMUM DALAM MENULIS PSEUDOCODE

1. **Instruksi tidak berurutan**
   Contoh: menampilkan hasil sebelum melakukan perhitungan.
2. **Nama variabel tidak jelas**
   Sebaiknya gunakan nilaiAkhir, bukan hanya x.
3. **Tidak menuliskan input**
   Data yang digunakan harus dijelaskan asalnya.
4. **Kondisi percabangan tidak lengkap**
   Misalnya hanya menuliskan kondisi lulus tanpa menjelaskan kondisi tidak lulus.
5. **Pengulangan tidak memiliki batas atau perubahan nilai**
   Hal ini dapat menyebabkan pengulangan berjalan terus-menerus.
6. **Tidak menggunakan indentasi**
   Pseudocode menjadi sulit dibaca, terutama pada percabangan dan pengulangan.

PRAKTIK 4

Buatlah pseudocode untuk menghitung total harga pembelian. Jika total belanja lebih dari Rp50.000, murid mendapatkan diskon 5%.

UJI KOMPETENSI BAB. III
DASAR PEMROGRAMAN: TEKS (PSEUDOCODE)

I. Berilah tanda silang (X) pada huruf A, B, C, atau D pada jawaban yang paling tepat!

1. Pseudocode adalah ...
   A. bahasa pemrograman yang langsung dijalankan komputer
   B. perangkat keras untuk membuat program
   C. kode semu untuk menuliskan langkah-langkah algoritma
   D. aplikasi untuk menggambar flowchart
2. Kata kunci yang digunakan untuk menandai awal pseudocode adalah ...
   A. INPUT
   B. MULAI
   C. TAMPILKAN
   D. JIKA
3. Kata kunci yang digunakan untuk menandai akhir pseudocode adalah ...
   A. STOP
   B. OUTPUT
   C. AKHIR
   D. SELESAI
4. Perintah yang digunakan untuk menerima data dari pengguna adalah ...
   A. INPUT
   B. TAMPILKAN
   C. JIKA
   D. ULANGI
5. Perintah yang digunakan untuk menampilkan hasil kepada pengguna adalah ...
   A. MULAI
   B. INPUT
   C. TAMPILKAN
   D. SELAMA
6. Data yang dimasukkan oleh pengguna ke dalam program disebut ...
   A. Proses
   B. Input
   C. output
   D. kondisi
7. Hasil yang ditampilkan oleh program disebut ...
   A. Input
   B. Proses
   C. output
   D. variabel
8. Variabel digunakan untuk ...
   A. menyimpan data
   B. menggambar gambar
   C. menghapus program
   D. menjalankan komputer
9. Operator + digunakan untuk ...
   A. Pembagian
   B. Perkalian
   C. penjumlahan
   D. perbandingan
10. Operator **MOD** digunakan untuk ...
    A. mencari hasil perkalian
    B. mencari sisa hasil bagi
    C. mencari nilai terbesar
    D. mencari nilai rata-rata
11. Perhatikan pseudocode berikut.

```
Mulai
INPUT panjang
INPUT lebar
luas ← panjang × lebar
TAMPILKAN luas
Selesai
```

    Bagian luas ← panjang × lebar merupakan bagian ...
    A. input
    B. proses
    C. output
    D. percabangan
12. Perhatikan pseudocode berikut.

```
Mulai
INPUT nilai
JIKA nilai >= 75 MAKA
   TAMPILKAN "Lulus"
JIKA TIDAK
   TAMPILKAN "Belum Lulus"
Selesai
```

    Struktur pemrograman yang digunakan pada pseudocode tersebut adalah ...

A. Urutan
B. Percabangan
C. pengulangan
D. variabel

13. Jika nilai bilangan = 14, hasil dari operasi berikut adalah ...
    bilangan **MOD 2**
    A. 0
    B. 1
    C. 2
    D. 7
14. Perhatikan pseudocode berikut.

```
Mulai
UNTUK i ← 1 SAMPAI 3 LAKUKAN
   TAMPILKAN "Halo"
SELESAI UNTUK
Selesai
```

    Tulisan "Halo" akan tampil sebanyak ...
    A. 1 kali
    B. 2 kali
    C. 3 kali
    D. 4 kali
15. Operator yang digunakan untuk menyatakan "lebih besar atau sama dengan" adalah **...**
    A. =
    B. >
    C. >=
    D. <=
16. Perhatikan pseudocode berikut.

```
Mulai
INPUT sisi
keliling ← 4 × sisi
TAMPILKAN keliling
Selesai
```

    Jika nilai sisi = 7, hasil yang ditampilkan adalah ...
    A. 11
    B. 21
    C. 28
    D. 49
17. Perhatikan pseudocode berikut!

```
Mulai
INPUT nilai1
INPUT nilai2
rataRata ← (nilai1 + nilai2) / 2
TAMPILKAN rataRata
Selesai
```

    Jika nilai1 = 80 dan nilai2 = 90, nilai rata-rata yang ditampilkan adalah ...
    A. 80
    B. 85
    C. 90
    D. 170
18. Perintah yang paling tepat untuk menentukan apakah suatu bilangan genap adalah ...
    A. JIKA bilangan > 2
    B. JIKA bilangan MOD 2 = 0
    C. JIKA bilangan = 2
    D. JIKA bilangan / 2 = 1
19. Perhatikan pseudocode berikut!

```
Mulai
angka ← 1
SELAMA angka <= 3 LAKUKAN
   TAMPILKAN angka
   angka ← angka + 1
SELESAI SELAMA
Selesai
```

    Hasil yang ditampilkan adalah ...
    A. 1, 2
    B. 1, 2, 3
    C. 1, 2, 3, 4
    D. 0, 1, 2, 3
20. Salah satu alasan penggunaan indentasi dalam pseudocode adalah ...
    A. membuat pseudocode lebih panjang
    B. memperindah warna teks
    C. menunjukkan bagian instruksi dalam percabangan atau pengulangan

D. mengganti nama variabel

21. Perhatikan pseudocode berikut!

```
Mulai
INPUT nilai
JIKA nilai >= 90 MAKA
   TAMPILKAN "A"
JIKA TIDAK JIKA nilai >= 75 MAKA
   TAMPILKAN "B"
JIKA TIDAK JIKA nilai >= 60 MAKA
   TAMPILKAN "C"
JIKA TIDAK
   TAMPILKAN "D"
Selesai
```

Jika nilai yang dimasukkan adalah 72, hasil yang ditampilkan adalah ...

A. A
B. B
C. C
D. D

22. Perhatikan pseudocode berikut!

```
Mulai
jumlah ← 0
UNTUK i ← 1 SAMPAI 5 LAKUKAN
   jumlah ← jumlah + i
SELESAI UNTUK
TAMPILKAN jumlah
Selesai
```

Nilai jumlah yang ditampilkan adalah ...

A. 5
B. 10
C. 15
D. 20

23. Perhatikan pseudocode berikut!

```
Mulai
INPUT totalBelanja
JIKA totalBelanja >= 100000 MAKA
   diskon ← totalBelanja × 10 / 100
JIKA TIDAK
   diskon ← 0
bayar ← totalBelanja - diskon
TAMPILKAN bayar
Selesai
```

Jika totalBelanja = 150000, nilai bayar adalah ...

A. 135000
B. 140000
C. 145000
D. 150000

24. Perhatikan pseudocode berikut!

```
Mulai
INPUT bilangan
JIKA bilangan MOD 2 = 0 MAKA
   TAMPILKAN "Genap"
JIKA TIDAK
   TAMPILKAN "Ganjil"
Selesai
```

Jika nilai bilangan = 25, hasil yang ditampilkan adalah ...

A. Genap
B. Ganjil
C. Positif
D. Negatif

25. Perhatikan pseudocode berikut!

```
Mulai
nilai ← 70
JIKA nilai >= 75 DAN nilai <= 100 MAKA
   TAMPILKAN "Lulus"
JIKA TIDAK
   TAMPILKAN "Belum Lulus"
Selesai
```

Hasil yang ditampilkan adalah ...

a. Lulus
b. Belum Lulus
C. Nilai tidak valid
D. Tidak ada output

II.Jawablah pertanyaan-pertanyaan di bawah ini dengan tepat!

1. Jelaskan pengertian pseudocode dan sebutkan dua manfaatnya dalam pembuatan program!
2. Buatlah pseudocode untuk menghitung luas persegi panjang!
   Ketentuan:
   - Murid memasukkan nilai panjang.
   - Murid memasukkan nilai lebar.
   - Program menampilkan luas persegi panjang.
3. Buatlah pseudocode untuk menentukan apakah seorang murid lulus atau belum lulus!
   Ketentuan:
   - Murid memasukkan nilai.
   - Jika nilai minimal 75, tampilkan "Lulus".
   - Jika nilai kurang dari 75, tampilkan "Belum Lulus".
4. Buatlah pseudocode untuk menampilkan angka 1 sampai 10 menggunakan pengulangan!
5. Buatlah pseudocode untuk menghitung total pembayaran belanja dengan ketentuan berikut:
   - Murid memasukkan harga barang.
   - Murid memasukkan jumlah barang.
   - Total belanja dihitung dari harga barang × jumlah barang.
   - Jika total belanja minimal Rp100.000, murid mendapatkan diskon 10%.
   - Jika total belanja kurang dari Rp100.000, tidak mendapatkan diskon.
   - Program menampilkan total belanja, diskon, dan total pembayaran.

BAB IV
MERANGKAI LOGIKA MENJADI PSEUDOCODE

Tujuan Pembelajaran
Setelah mengikuti pembelajaran ini, murid mampu:
1. Menjelaskan pengertian penulisan instruksi menggunakan kosakata terbatas dalam pseudocode.
2. Mengidentifikasi kata kunci yang digunakan dalam format pseudocode.
3. Mengidentifikasi simbol yang digunakan dalam pseudocode.
4. Menjelaskan fungsi setiap kata kunci dan simbol dalam pseudocode.
5. Menulis sekumpulan instruksi sederhana menggunakan format pseudocode.
6. Menggunakan variabel untuk menyimpan data, operasi aritmetika, operator perbandingan dan logika dalam pseudocode.
7. Menulis pseudocode dengan struktur urutan, percabangan, dan pengulangan.
8. Menyusun pseudocode untuk menyelesaikan masalah sederhana dalam kehidupan sehari-hari.
9. Memeriksa kesalahan penulisan pseudocode dan memperbaikinya.

A. KONSEP DASAR INSTRUKSI DALAM PSEUDOCODE

1. PENGERTIAN INSTRUKSI

Instruksi adalah perintah yang diberikan untuk melakukan suatu tindakan.
Contoh instruksi dalam kehidupan sehari-hari:
1. Buka buku.
2. Bacalah halaman 10.
3. Kerjakan latihan.
4. Kumpulkan tugas.

Dalam pemrograman, instruksi ditulis dengan lebih teratur agar dapat dipahami oleh komputer.
Contoh instruksi dalam pseudocode:

```
MULAI
TAMPILKAN "Selamat Belajar Informatika"
SELESAI
```

Instruksi tersebut berarti program dimulai, menampilkan sebuah pesan, kemudian selesai.

2. SEKUMPULAN INSTRUKSI

Sekumpulan instruksi adalah beberapa perintah yang disusun secara runtut untuk menyelesaikan sebuah masalah.
Contoh masalah: menghitung jumlah dua bilangan.

```
MULAI
INPUT bilangan1
INPUT bilangan2
jumlah ← bilangan1 + bilangan2
TAMPILKAN jumlah
SELESAI
```

Pseudocode tersebut terdiri atas beberapa instruksi:
1) Memulai program.
2) Memasukkan bilangan pertama.
3) Memasukkan bilangan kedua.
4) Menjumlahkan kedua bilangan.
5) Menampilkan hasil.
6) Mengakhiri program.

B. Kosakata Terbatas dalam Pseudocode

Kosakata terbatas adalah kata-kata khusus yang digunakan secara konsisten dalam penulisan pseudocode. Kata-kata tersebut disebut juga **kata kunci**.

1. Kata Kunci Dasar

KATA KUNCI	FUNGSI
MULAI	Menandai awal program
SELESAI	Menandai akhir program
INPUT	Menerima data dari pengguna
BACA	Membaca data masukan
TAMPILKAN	Menampilkan hasil atau pesan
OUTPUT	Menampilkan hasil program
JIKA	Memulai pemeriksaan kondisi
MAKA	Menunjukkan tindakan jika kondisi benar
JIKA TIDAK	Menunjukkan tindakan jika kondisi salah
UNTUK	Memulai pengulangan dengan jumlah tertentu
SELAMA	Memulai pengulangan berdasarkan kondisi
LAKUKAN	Menunjukkan instruksi yang dikerjakan
SELESAI UNTUK	Menandai akhir pengulangan UNTUK
SELESAI SELAMA	Menandai akhir pengulangan SELAMA

2. Contoh Penggunaan Kata Kunci

```
MULAI
INPUT nama
TAMPILKAN "Halo, ", nama
SELESAI
```

Penjelasan:

- MULAI menunjukkan program dimulai.
- INPUT nama menerima data nama dari pengguna.
- TAMPILKAN digunakan untuk menampilkan sapaan.
- SELESAI menunjukkan program berakhir.

PRAKTIK 1

Buatlah pseudocode untuk menampilkan nama murid dan kelasnya.

Petunjuk

- Gunakan MULAI, INPUT, TAMPILKAN, dan SELESAI.
- Data yang dimasukkan adalah nama dan kelas.

C. SIMBOL DALAM PSEUDOCODE

Selain kata kunci, pseudocode menggunakan simbol untuk membantu menuliskan perhitungan, perbandingan, dan penyimpanan data.

1. Simbol Penugasan

SIMBOL	FUNGSI	CONTOH
←	Menyimpan hasil ke dalam variabel	jumlah ← a + b
=	Dapat digunakan untuk menyatakan nilai atau perbandingan	nilai = 75

Simbol ← dibaca "diisi dengan" atau "mendapat nilai dari".
Contoh:
luas ← panjang × lebar
Artinya, hasil perkalian panjang × lebar disimpan ke dalam variabel luas.

2. Simbol Aritmetika

SIMBOL	NAMA OPERASI	CONTOH
+	Penjumlahan	jumlah ← a + b
-	Pengurangan	selisih ← a – b
× atau *	Perkalian	luas ← panjang × lebar
/	Pembagian	rataRata ← jumlah / 2
MOD	Sisa hasil bagi	sisa ← bilangan MOD 2

Contoh: total ← harga × jumlahBarang

PRAKTIK 2

Buatlah pseudocode untuk menghitung luas segitiga.
Petunjuk

- Input: alas dan tinggi.
- Rumus: luas = alas × tinggi / 2.
- Output: luas segitiga..

3. Simbol Perbandingan

Simbol	Arti	Contoh
=	Sama dengan	nilai = 75
≠	Tidak sama dengan	nilai ≠ 0
>	Lebih besar dari	nilai > 75
<	Lebih kecil dari	nilai < 75
>=	Lebih besar atau sama dengan	nilai >= 75
<=	Lebih kecil atau sama dengan	umur <= 17

Simbol perbandingan biasanya digunakan dalam percabangan.
Contoh:

```
JIKA nilai >= 75 MAKA
   TAMPILKAN "Lulus"
```

4. Simbol dan Operator Logika

Operator	Arti	Contoh
DAN	Kedua kondisi harus benar	nilai >= 75 DAN kehadiran >= 80
ATAU	Salah satu kondisi benar	hari = "Sabtu" ATAU hari = "Minggu"
TIDAK	Membalik kondisi	TIDAK hujan

Contoh:
JIKA nilai >= 75 DAN kehadiran >= 80 MAKA
  TAMPILKAN "Lulus"

D. VARIABEL DALAM PSEUDOCODE

1. Pengertian Variabel

Variabel adalah tempat untuk menyimpan data yang dapat berubah nilainya.
Contoh:
namaMurid ← "Alya"
nilai ← 85
Pada contoh tersebut:

- namaMurid menyimpan data teks.
- nilai menyimpan data angka.

2. Aturan Penulisan Nama Variabel

Nama variabel sebaiknya:

1. Menggunakan kata yang mudah dipahami.
2. Menggambarkan data yang disimpan.
3. Tidak menggunakan spasi.
4. Tidak dimulai dengan angka.
5. Konsisten dalam penulisan.

Contoh nama variabel yang baik:
namaMurid
nilaiUlangan
jumlahBarang
totalBelanja
rataRata
Contoh nama variabel yang kurang baik:
a
x
data1
nilai murid
1nilai

E. FORMAT DASAR PENULISAN DAN MENULIS INSTRUKSI DENGAN STRUKTUR URUTAN PSEUDOCODE

1. Format dasar pseudocode dapat ditulis seperti berikut:

Format dasar pseudocode dapat ditulis seperti berikut: MULAI INPUT data proses ← perhitungan atau pengolahan data TAMPILKAN hasil SELESAI	Contoh menghitung luas persegi: MULAI INPUT sisi luas ← sisi × sisi TAMPILKAN "Luas persegi = ", luas SELESAI

2. Pengertian Struktur Urutan

Struktur urutan adalah instruksi yang dijalankan satu per satu dari atas ke bawah.
Contoh:

MULAI INPUT harga INPUT jumlah total ← harga × jumlah TAMPILKAN total SELESAI	Program menjalankan instruksi sesuai urutan: 1. Meminta harga barang. 2. Meminta jumlah barang. 3. Menghitung total. 4. Menampilkan total.

Contoh: Menghitung Keliling Persegi	**Contoh: Menghitung Rata-Rata Dua Nilai**
MULAI INPUT sisi keliling ← 4 × sisi TAMPILKAN "Keliling persegi = ", keliling SELESAI	MULAI INPUT nilai1 INPUT nilai2 jumlah ← nilai1 + nilai2 rataRata ← jumlah / 2 TAMPILKAN "Rata-rata = ", rataRata SELESAI

3. Menulis Instruksi dengan Struktur Percabangan

Percabangan digunakan ketika program harus memilih tindakan berdasarkan kondisi tertentu.
Contoh kehidupan sehari-hari:

- Jika hujan, bawa payung.
- Jika tidak hujan, gunakan topi.

a. Percabangan Satu Kondisi	**b. Percabangan Dua Kondisi**

MULAI
INPUT nilai
JIKA nilai >= 75 MAKA
  TAMPILKAN "Lulus"
SELESAI
	MULAI
	INPUT nilai
	JIKA nilai >= 75 MAKA
	  TAMPILKAN "Lulus"
	JIKA TIDAK
	  TAMPILKAN "Belum Lulus"
	SELESAI

Contoh: Menentukan Bilangan Genap atau Ganjil

MULAI
INPUT bilangan
JIKA bilangan MOD 2 = 0 MAKA
  TAMPILKAN "Bilangan Genap"
JIKA TIDAK
  TAMPILKAN "Bilangan Ganjil"
SELESAI
	Penjelasan:
	• MOD 2 digunakan untuk mencari sisa pembagian dengan 2.
	• Jika sisa pembagian adalah 0, bilangan tersebut genap.
	• Jika sisa pembagian bukan 0, bilangan tersebut ganjil.

PRAKTIK 3

Buatlah pseudocode untuk menentukan apakah umur seseorang sudah dapat membuat KTP.

Ketentuan

- Jika umur minimal 17 tahun, tampilkan "Dapat membuat KTP".
- Jika umur kurang dari 17 tahun, tampilkan "Belum dapat membuat KTP".

F. MENULIS INSTRUKSI DENGAN STRUKTUR PENGULANGAN

Pengulangan digunakan ketika suatu instruksi perlu dilakukan berulang kali.
Contoh: menampilkan angka 1 sampai 5.

1. Pengulangan UNTUK	**2. Pengulangan SELAMA**

MULAI
UNTUK i ← 1 SAMPAI 5 LAKUKAN
  TAMPILKAN i
SELESAI UNTUK
SELESAI
	MULAI
	angka ← 1
	SELAMA angka <= 5 LAKUKAN
	  TAMPILKAN angka
	  angka ← angka + 1
	SELESAI SELAMA
	SELESAI

Contoh: Menampilkan Pesan Lima Kali

```
MULAI
UNTUK i ← 1 SAMPAI 5 LAKUKAN
   TAMPILKAN "Semangat Belajar Informatika"
SELESAI UNTUK
SELESAI
```

G. MENGGABUNGKAN URUTAN, PERCABANGAN, DAN PENGULANGAN

Pseudocode dapat menggunakan lebih dari satu struktur.

Contoh: menampilkan status kelulusan lima murid.

MULAI
UNTUK i ← 1 SAMPAI 5 LAKUKAN
  INPUT nilai
  JIKA nilai >= 75 MAKA
  TAMPILKAN "Lulus"
  JIKA TIDAK
  TAMPILKAN "Belum Lulus"
SELESAI UNTUK
SELESAI
	Pseudocode tersebut menggunakan:
	• struktur pengulangan UNTUK;
	• struktur percabangan JIKA;
	• input nilai;
	• output status kelulusan.

Langkah menulis pseudocode:

1. Pahami masalah yang diberikan.
2. Tentukan tujuan program.
3. Tentukan data masukan atau input.
4. Tentukan proses yang dilakukan.
5. Tentukan hasil atau output.
6. Tentukan struktur yang digunakan: urutan, percabangan, atau pengulangan.
7. Gunakan kata kunci pseudocode secara tepat.
8. Gunakan nama variabel yang jelas.
9. Tuliskan instruksi secara runtut.
10. Periksa kembali kesesuaian logika pseudocode.
11. Uji pseudocode menggunakan contoh data.

PRAKTIK 4

Buatlah pseudocode untuk menghitung nilai rata-rata tiga nilai. Jika rata-rata minimal 75, tampilkan “Lulus”. Jika kurang dari 75, tampilkan “Belum Lulus”.

H. CONTOH ANALISIS MASALAH DAN PSEUDOCODE

Masalah

Buatlah Pseudocode untuk menghitung total harga pembelian. Jika total belanja minimal Rp100.000, pembeli mendapat diskon 10%.

Analisis Masalah

Bagian	Isi
Input	hargaBarang, jumlahBarang
Proses	totalBelanja = hargaBarang × jumlahBarang
Kondisi	totalBelanja >= 100000
Output	totalBelanja, diskon, totalBayar

Pseudocode

```
MULAI
```

```
INPUT hargaBarang
INPUT jumlahBarang

totalBelanja ← hargaBarang × jumlahBarang

JIKA totalBelanja >= 100000 MAKA
   diskon ← totalBelanja × 10 / 100
JIKA TIDAK
   diskon ← 0

totalBayar ← totalBelanja - diskon

TAMPILKAN "Total belanja = ", totalBelanja
TAMPILKAN "Diskon = ", diskon
TAMPILKAN "Total pembayaran = ", totalBayar

SELESAI
```

UJI KOMPETENSI BAB. IV
MERANGKAI LOGIKA MENJADI PSEUDOCODE

I. Berilah tanda silang (X) pada huruf A, B, C, atau D pada jawaban yang paling tepat!

1. Fungsi utama penggunaan pseudocode sebelum membuat program adalah ...
   A. mempercepat komputer bekerja
   B. menyusun langkah penyelesaian masalah secara teratur
   C. mengganti perangkat keras komputer
   D. membuat gambar animasi
2. Kata kunci yang tepat untuk meminta murid memasukkan umur adalah ...
   A. TAMPILKAN umur
   B. umur ← INPUT
   C. INPUT umur
   D. JIKA umur
3. Perintah yang tepat untuk menampilkan pesan "Selamat Datang" adalah ...
   A. INPUT "Selamat Datang"
   B. TAMPILKAN "Selamat Datang"
   C. MULAI "Selamat Datang"
   D. JIKA "Selamat Datang"
4. Pada pseudocode berikut, jumlah merupakan ...
   jumlah ← harga + pajak
   A. kata kunci
   B. Variabel
   C. operator logika
   D. kondisi
5. Simbol yang tepat untuk operasi perkalian dalam pseudocode adalah ...
   A. +
   B. -
   C. ×
   D. >=
6. Perintah nilaiAkhir ← nilaiTugas + nilaiUlangan berarti ...
   A. nilai tugas dibandingkan dengan nilai ulangan
   B. hasil penjumlahan disimpan dalam variabel nilaiAkhir
   C. nilai akhir ditampilkan ke layar
   D. nilai tugas dihapus dari program
7. Operator yang digunakan untuk memeriksa apakah dua nilai sama adalah ...
   A. =
   B. >
   C. <
   D. MOD
8. Pernyataan yang benar tentang nama variabel adalah ...
   A. nama variabel harus selalu berupa angka
   B. nama variabel boleh menggunakan spasi
   C. nama variabel sebaiknya menggambarkan data yang disimpan
   D. nama variabel harus menggunakan kata INPUT
9. Kata kunci JIKA TIDAK digunakan ketika ...
   A. program dimulai
   B. kondisi pada JIKA tidak terpenuhi
   C. data dimasukkan oleh pengguna
   D. pengulangan selesai dilakukan
10. Kata kunci yang digunakan untuk melakukan pengulangan dengan jumlah tertentu adalah ...
    A. JIKA
    B. UNTUK
    C. INPUT
    D. TAMPILKAN
11. Perhatikan pseudocode berikut!

```
MULAI
INPUT harga
INPUT jumlah
total ← harga × jumlah
TAMPILKAN total
SELESAI
```

    Jika harga = 5000 dan jumlah = 4, hasil yang ditampilkan adalah ...
    A. 9000
    B. 10000
    C. 20000
    D. 25000
12. Perhatikan pseudocode berikut!

```
MULAI
INPUT suhu
```

```
JIKA suhu > 30 MAKA
   TAMPILKAN "Panas"
JIKA TIDAK
   TAMPILKAN "Tidak Panas"
SELESAI
```

Jika suhu yang dimasukkan adalah 30, hasil yang ditampilkan adalah ...

A. Panas
B. Tidak Panas
C. Sangat Panas
D. Tidak ada output

13. Perhatikan pseudocode berikut!

```
MULAI
a ← 12
b ← 5
selisih ← a - b
TAMPILKAN selisih
SELESAI
```

Nilai yang ditampilkan adalah ...

A. 5
B. 7
C. 12
D. 17

14. Perhatikan pseudocode berikut!

```
MULAI
UNTUK nomor ← 2 SAMPAI 6 LAKUKAN
   TAMPILKAN nomor
SELESAI UNTUK
SELESAI
```

Banyak angka yang ditampilkan adalah ...

A. 4
B. 5
C. 6
D. 8

15. Perhatikan pseudocode berikut!

```
MULAI
nilai ← 64
JIKA nilai >= 60 MAKA
   TAMPILKAN "Tuntas"
JIKA TIDAK
   TAMPILKAN "Belum Tuntas"
SELESAI
```

Output yang benar adalah ...

A. Tuntas
B. Belum Tuntas
C. Nilai Salah
D. Tidak ada output

16. Perhatikan pseudocode berikut.

```
MULAI
bilangan ← 27
hasil ← bilangan MOD 5
TAMPILKAN hasil
SELESAI
```

Nilai hasil adalah ...

A. 0
B. 1
C. 2
D. 5

17. Perhatikan Pseudocode berikut :

```
MULAI
INPUT alas
```

```
    luas ← alas × tinggi / 2
    TAMPILKAN luas
    SELESAI
```

Pseudocode tersebut memiliki kesalahan karena ...

A. tidak ada kata MULAI
B. variabel tinggi belum diberi nilai atau input
C. luas tidak boleh menggunakan simbol ←
D. TAMPILKAN harus ditulis sebelum INPUT

18. Perhatikan Pseudocode berikut.

```
MULAI
x ← 3
SELAMA x < 8 LAKUKAN
   TAMPILKAN x
   x ← x + 2
SELESAI SELAMA
SELESAI
```

Urutan angka yang ditampilkan adalah ...

A. 3, 4, 5, 6, 7
B. 3, 5, 7
C. 3, 5, 7, 9
D. 2, 4, 6, 8

19. Perhatikan pernyataan berikut.

```
JIKA umur >= 17 ATAU sudahMenikah = benar MAKA
   TAMPILKAN "Memenuhi syarat"
```

Operator ATAU berarti ...

A. kedua kondisi harus benar
B. salah satu kondisi bernilai benar
C. semua kondisi bernilai salah
D. kondisi harus diulang

20. Manakah pseudocode yang paling tepat untuk menghitung sisa uang?

A. sisa ← uangAwal + uangBelanja
B. sisa ← uangBelanja – uangAwal
C. sisa ← uangAwal - uangBelanja
D. sisa ← uangAwal MOD uangBelanja

21. Perhatikan pseudocode berikut.

```
MULAI
INPUT angka
JIKA angka > 0 MAKA
   TAMPILKAN "Positif"
JIKA TIDAK JIKA angka < 0 MAKA
   TAMPILKAN "Negatif"
JIKA TIDAK
   TAMPILKAN "Nol"
SELESAI
```

Jika nilai angka = 0, output yang benar adalah ...

A. Positif
B. Negatif
C. Nol
D. Tidak ada output

22. Perhatikan pseudocode berikut!

```
MULAI
total ← 1
UNTUK i ← 1 SAMPAI 4 LAKUKAN
   total ← total × 2
SELESAI UNTUK
TAMPILKAN total
SELESAI
```

Nilai total yang ditampilkan adalah ...

A. 4
B. 8
C. 12
D. 16

23. Perhatikan pseudocode berikut!

```
MULAI
INPUT nilai
JIKA nilai >= 85 MAKA
   predikat ← "Sangat Baik"
JIKA TIDAK JIKA nilai >= 70 MAKA
   predikat ← "Baik"
JIKA TIDAK JIKA nilai >= 55 MAKA
   predikat ← "Cukup"
JIKA TIDAK
   predikat ← "Perlu Bimbingan"
TAMPILKAN predikat
SELESAI
```

Jika nilai yang dimasukkan adalah 69, predikat yang ditampilkan adalah ...

A. Sangat Baik
B. Baik
C. Cukup
D. Perlu Bimbingan

24. Perhatikan pseudocode berikut!

```
MULAI
jumlah ← 0
UNTUK i ← 2 SAMPAI 8 LAKUKAN
   JIKA i MOD 2 = 0 MAKA
      jumlah ← jumlah + i
SELESAI UNTUK

TAMPILKAN jumlah
SELESAI
```

Nilai jumlah yang ditampilkan adalah ...

A. 16
B. 18
C. 20
D. 28

25. Perhatikan pseudocode berikut.

```
MULAI
INPUT nilaiUjian
INPUT nilaiTugas
nilaiAkhir ← (nilaiUjian × 70 / 100) + (nilaiTugas × 30 / 100)
JIKA nilaiAkhir >= 75 DAN nilaiUjian >= 60 MAKA
   TAMPILKAN "Lulus"
JIKA TIDAK
   TAMPILKAN "Belum Lulus"
SELESAI
```

26. Jika nilaiUjian = 70 dan nilaiTugas = 90, hasil yang ditampilkan adalah ...

A. Lulus
B. Belum Lulus
C. Nilai Tidak Valid
D. Tidak ada output

II.Jawablah pertanyaan-pertanyaan di bawah ini dengan tepat!

1. Jelaskan pengertian kosakata terbatas dalam pseudocode dan berikan tiga contohnya!
2. Jelaskan fungsi dari kata kunci berikut: INPUT, TAMPILKAN, dan SELESAI!
3. Buatlah pseudocode untuk menghitung luas persegi dengan ketentuan berikut:
   a. Input: panjang sisi persegi.
   b. Proses: sisi × sisi.
   c. Output: luas persegi.
4. Buatlah pseudocode untuk menentukan apakah sebuah bilangan merupakan bilangan genap atau ganjil.
5. Buatlah pseudocode untuk menghitung nilai rata-rata tiga nilai murid dengan ketentuan berikut:
   a. Murid memasukkan nilai pertama, nilai kedua, dan nilai ketiga.
   b. Program menghitung nilai rata-rata.
   c. Jika rata-rata minimal 75, tampilkan "Lulus".
   d. Jika rata-rata kurang dari 75, tampilkan "Belum Lulus".
   e. Program juga menampilkan nilai rata-rata.