



KEMENTERIAN
PENDIDIKAN DASAR
DAN MENENGAH



Konsep Berpikir Komputasional

Konsep Berpikir Komputasional

- Berpikir komputasional berarti peserta didik memiliki kemampuan berpikir yang dapat menyelesaikan permasalahan ataupun persoalan sehari-hari dengan menggunakan teknik ilmu komputer
- Dengan demikian, berpikir komputasional merupakan kemampuan menyusun solusi masalah secara logis dan sistematis sehingga dapat diproses oleh manusia maupun komputer.

Berpikir Komputasional dan Kompetensi Abad 21

- Berpikir komputasional berkaitan erat dengan konsep 4C, yaitu Critical Thinking (berpikir kritis), Creativity (kreativitas), Communication (komunikasi), dan Collaboration (kolaborasi) dari kebutuhan abad 21 karena mendukung penyelesaian masalah secara sistematis, efektif, efisien, dan optimal.

Orientasi Kritis dalam Berpikir Komputasional

- Keterkaitan hubungan antara berpikir komputasional dengan berpikir kritis tidak dapat dipisahkan karena prinsip-prinsip yang ada di dalam berpikir komputasional membangun cara berpikir sehingga membantu seseorang dapat menyelesaikan suatu masalah dengan logis dan efektif

Prinsip Berpikir Komputasional

- Dekomposisi
- Pattern Recognition
- Abstraksi
- Algorithmic Thinking

Dekomposisi

Memecah masalah kompleks menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola atau bisa dipahami dengan sederhana sebagai pemecahan masalah besar menjadi bagian yang lebih kecil.

- **Cara: Berkelompok 2 - 3 orang**
- **Kegiatan: Merancang pesta ulang tahun dengan berdiskusi**
- **Baca modul 2 fase C untuk detail dan contoh**

Pattern Recognition (Pengenalan Pola)

Pengenalan pola berarti mengidentifikasi kesamaan atau pola dalam berbagai permasalahan untuk menemukan solusi yang dapat diterapkan kembali atau dengan kata lain adalah menemukan pola yang bisa digunakan kembali.

- **Cara: Berkelompok 4 - 7 orang**
- **Kegiatan: Membuat pola di dalam kelompok sendiri dan menebak pola orang lain**
- **Baca modul 2 fase C untuk detail dan contoh**

Abstraction (Abstraksi)

Abstraksi adalah menyaring informasi yang tidak relevan dan hanya mengambil bagian yang penting untuk menyelesaikan masalah, dengan kata lain hanya terfokus pada informasi penting dan menyaring detail yang tidak relevan.

- **Cara:** Satu orang menjadi pemberi informasi dan peserta lain akan menjawab.
- **Kegiatan:** Mencari jawaban yang paling tepat dari informasi yang diberikan
- **Baca modul 2 fase C untuk detail dan contoh**

Algorithmic Thinking (Berpikir Algoritmik)

Berpikir algoritmik yaitu menyusun langkah-langkah, instruksi ataupun prosedur yang sistematis untuk menyelesaikan suatu masalah tertentu.

- **Cara:** Buatlah beberapa kelompok dengan jumlah anggota 3 - 5 orang.
- **Kegiatan:** Bermain game susun instruksi atau berpasangan.
- Baca modul 2 fase C untuk detail dan contoh.

Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial (KA)

Menghubungkan Prinsip Berpikir Komputasional dengan Koding

Pembelajaran Berpikir Komputasional sebagai Landasan untuk Kecerdasan Artifisial (KA)

Menurut naskah akademik Pembelajaran Koding Dan Kecerdasan Artifisial Pada Pendidikan Dasar dan Menengah Tujuan dari pembelajaran ini tidak hanya berhenti pada menyiapkan peserta didik untuk memiliki kemampuan penyelesaian masalah, namun juga mengajarkan peserta didik untuk mengerti inti dari esensi pembelajaran berpikir komputasional dan kecerdasan artifisial.

Pembelajaran Berpikir Komputasional sebagai Landasan untuk Kecerdasan Artifisial (KA)

Hal esensi yang dimaksud:

Berpikir komputasional, analisis data, algoritma pemrograman, etika KA, human-centered mindset, design system KA, dan teknik KA.

Menghubungkan Prinsip Berpikir Komputasional dengan Koding

Tabel 1.2. Hubungan berpikir komputasional dengan koding

Prinsip Berpikir Komputasional	Penerapan dalam koding
Dekomposisi (Decomposition)	Pembagian proyek ke dalam kelompok
	seperti gerakan, suara, karakter dan efek.
Pengenalan Pola (Pattern Recognition)	Penggunaan kode <i>loop</i> atau pengulangan untuk menghindari penulisan kode yang tidak efisien.
Abstraksi (Abstraction)	Menggunakan blok kode yang sudah ada (seperti fungsi atau variabel) agar kode lebih ringkas dan mudah dipahami.
Berpikir Algoritma (Algorithmic Thinking)	Menulis kode dengan urut dan logis sehingga program dapat digunakan sesuai dengan instruksi yang benar.

IPO (Input-Proses-Output)

Tabel 1.3. Penjelasan konsep IPO.

Komponen	Penjelasan	Contoh dalam Kehidupan Sehari-hari	Contoh dalam Koding
Input (Masukan)	Data atau informasi yang dimasukkan ke dalam sistem.	Pada mesin cuci, input berupa: Baju kotor Deterjen Air	Menekan tombol "Start" pada gim.
Proses (Proses)	Sistem memproses input berdasarkan aturan yang sudah ditentukan.	Mesin mencuci (mengaduk dan membersihkan baju), membilas (menghilangkan sisa deterjen) dan mengeringkan.	Aplikasi program menjalankan perintah (misalnya, karakter berjalan ke kanan).
Output (Luaran)	Hasil akhir setelah proses dijalankan.	Hasilnya baju bersih dan cukup kering	Karakter di aplikasi bergerak sesuai perintah (misalkan bergerak ke kanan). ▾

Contoh kegiatan: *Unplugged IPO*

Dalam permainan "Simon Says", peserta didik akan membentuk lingkaran atau kelompok besar, dengan satu peserta didik berperan sebagai "Simon". Tugas "Simon" adalah memberikan instruksi kepada peserta lain, yang hanya boleh diikuti jika instruksi diawali dengan kata "Simon says".

Variabel

Variabel adalah wadah atau bentuk yang dapat dilihat, diukur ataupun diamati. Dalam koding, variabel digunakan untuk menyimpan data yang akan digunakan.

Alur Logika dalam Koding

Sekuensial:

Perintah dieksekusi satu per satu sesuai urutan.

Percabangan:

Program memilih jalur berbeda berdasarkan kondisi tertentu.

Perulangan:

Mengulangi perintah hingga kondisi tertentu terpenuhi.

Contoh kegiatan: Perulangan dalam Pola Gerakan Karakter

Cobalah untuk membuat program menggunakan aplikasi/ halaman web untuk menggunakan blok "repeat until" atau "forever" untuk mengulang gerakan tertentu. Perhatikan apa yang terjadi saat perintah tersebut diberikan.

Kegiatan Yang Bisa Dilakukan dalam Kelas

Strategi Pembelajaran Berbasis Koding (*plugged*) & Aktivitas Interaktif Berbasis Berpikir Komputasional dan Koding (*unplugged*)

Strategi Pembelajaran Berbasis Koding (plugged)

Kegiatan Plugged berarti kegiatan yang menggunakan perangkat teknologi seperti komputer, tablet, atau aplikasi untuk mengajarkan konsep berpikir komputasional.

1. Menggunakan aplikasi kode blok: Scratch atau Scratch Junior
2. Menggunakan aplikasi kode blok: Blockly atau [Code.org](https://code.org)

Kegiatan ini bisa dilakukan dengan mencobanya langsung apabila fasilitas memadai.

Contoh algoritma dalam kehidupan sehari-hari

Aktivitas 5. Algoritma

Tujuan: Peserta pelatihan belajar pemecahan masalah logistik dalam menghitung waktu tempuh berkendara.

Plugged: Aplikasi blockly (Scratch, Pictoblox, dll)

Unplugged: Kertas, pensil, atau papan tulis.

Langkah-langkah:

Berikut ini adalah contoh langkah-langkah yang dapat digunakan dalam menghitung waktu tempuh:

Langkah 1: Tentukan jarak yang akan ditempuh

Tentukan jarak antara titik asal dan tujuan, misalnya 100 km.

Langkah 2: Tentukan kecepatan rata-rata kendaraan

Tentukan kecepatan rata-rata kendaraan berdasarkan kondisi jalan dan jenis kendaraan, misalnya 60 km/jam.

Langkah 3: Hitung waktu tempuh dasar

Gunakan rumus dasar:

Waktu Tempuh = Jarak / Kecepatan Rata-rata

Contoh jika jarak adalah 100 km dan kecepatan rata-rata 60 km/jam, maka:

$$\begin{aligned} \text{Waktu Tempuh} &= 100 / 60 \\ &= 1.67 \text{ jam} \end{aligned}$$

Langkah 4: Tambahkan waktu ekstra sebagai faktor eksternal

Mempertimbangkan faktor seperti kemacetan atau cuaca buruk. Misalnya, jika perjalanan diperkirakan melalui daerah yang padat pada jam sibuk, tambah waktu tempuh sebesar 30 menit.

Langkah 5: Tambahkan waktu istirahat

Jika perlu berhenti untuk istirahat, tambahkan waktu istirahat. Misalnya, waktu istirahat selama 15 menit.

Langkah 6: Hitung total waktu tempuh

Total waktu tempuh keseluruhan adalah jumlah waktu tempuh dasar, waktu tambahan karena kemacetan, dan waktu istirahat.

Contoh:

5.1. Perhitungan dengan Formula Dasar

Waktu Tempuh (jam) = Jarak (km) ÷ Kecepatan Rata-rata (km/jam)

$$\text{Waktu Tempuh} = 100 / 60 = 1.67 \text{ jam}$$

5.2. Perhitungan dengan Formula Koreksi

Total Waktu Tempuh = Waktu Tempuh + Faktor Lalu Lintas + Faktor Jenis Jalan

$$\begin{aligned}\text{Total Waktu Tempuh} &= 1.67 \text{ jam} + 0.5 \text{ jam (kemacetan)} + 0.25 \text{ jam (istirahat)} \\ &= 2.42 \text{ jam}\end{aligned}$$

Penerapan perhitungan waktu tempuh dengan aplikasi pemrograman berbasis Blockly



A Scratch script consisting of the following blocks:

- when green flag clicked
- ask [Masukkan waktu tempuh?] and wait
- set `waktu_tempuh` to [answer]
- ask [Masukkan estimasi macet] and wait
- set `kemacetan` to [answer]
- ask [Masukkan waktu Istirahat] and wait
- set `istirahat` to [answer]
- set `Hasil` to [`waktu_tempuh` + `kemacetan` + `istirahat`]
- say [join [Waktu perjalanan anda:] [join `Hasil` [Jam]]]

The script prompts the user for three inputs: travel time, estimated traffic, and break time. It then calculates the total travel time by summing these values and outputs the result along with the word "Jam". The "Hasil" variable is highlighted with a green oval.

Aktivitas Interaktif Berbasis Berpikir Komputasional dan Koding (*unplugged*)

Kegiatan Unplugged berarti kegiatan yang tidak menggunakan perangkat elektronik, tetapi tetap mengajarkan konsep komputasional melalui aktivitas fisik dan permainan.

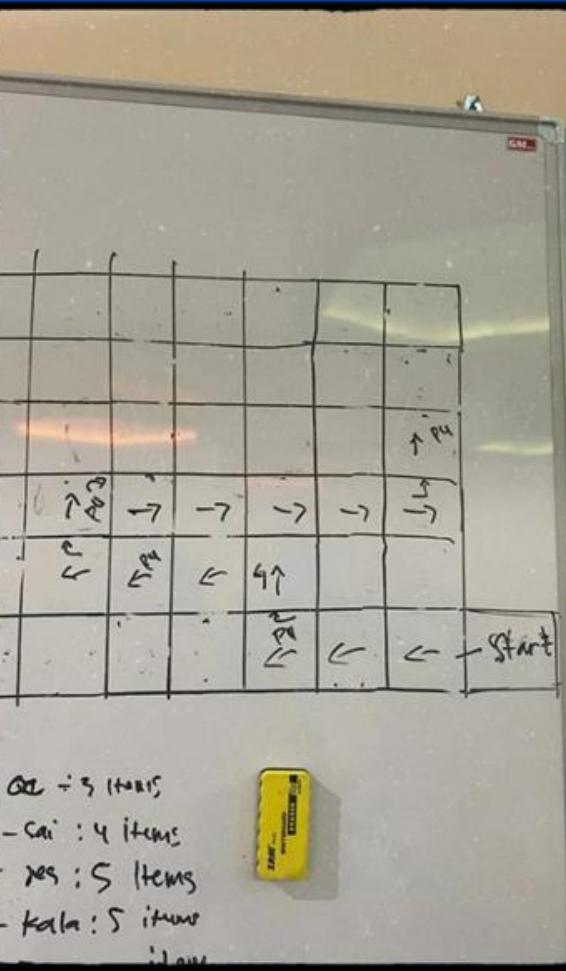
1. Kegiatan 1: Dekomposisi - “Program Your Pals”.
2. Kegiatan 2: Algoritma - “Menulis Kode Resep”.
3. Kegiatan 3: Abstraksi - “Filter Informasi”.

Kegiatan 1 - 3 bisa dilakukan dan dicoba bersama apabila ingin mengalami langsung pengalaman *unplugged*.

Contoh kegiatan: *Unplugged*



Siswa memainkan permainan "*human grid*", siswa berpura-pura menjadi robot yang diberikan instruksi.



Siswa memberikan instruksi kepada "robot" dengan menuliskan instruksi di papan tulis untuk diproses.



Siswa mengikuti gerakan guru pada materi Pengenalan Pola