

PETA KONSEP

A

Konsep Berpikir Komputasional

B

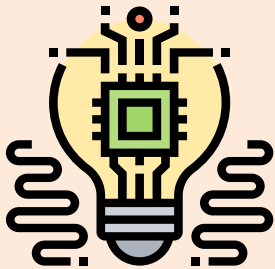
Berbagai Kasus Berpikir Komputasional

A

Konsep Berpikir Komputasional

01

Berpikir Komputasional



Berpikir komputasional (computational thinking) adalah konsep bagaimana menemukan masalah yang ada di sekitar kita, memahaminya, kemudian mengembangkan solusi yang inovatif dengan bantuan perangkat teknologi komputer.

Agar dapat menangani permasalahan kompleks dan memberikan solusi yang inovatif, berpikir komputasional mempunyai empat teknik kunci pendekatan, yaitu **dekomposisi**, **pengenalan pola**, **abstraksi**, dan **algoritma**.



A

Konsep Berpikir Komputasional

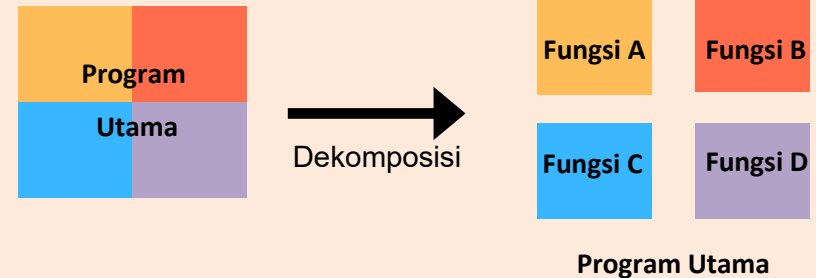
01

Berpikir Komputasional

a. Dekomposisi

Pendekatan dengan memecah masalah besar dan kompleks menjadi masalah-masalah yang lebih kecil dan lebih sederhana sehingga lebih mudah dikelola dan dipahami, kemudian dicari solusinya.

Ilustrasi Dekomposisi Fungsi












A Konsep Berpikir Komputasional

01 Berpikir Komputasional

b. Pengenalan Pola

Pengenalan pola dilakukan untuk melihat kesamaan pola dari masalah yang ada maupun masalah sebelumnya.

a.	2	6	10	14	18
b.					
c.					

A**Konsep Berpikir Komputasional****01****Berpikir Komputasional****c. Abstraksi**

Abstraksi dilakukan untuk menghilangkan informasi yang tidak penting dan fokus pada informasi yang penting dalam penyelesaian masalah.

Contoh terdapat pola bilangan berurutan

6, 4, 8, 6, 10, 8, 12, ...

Melakukan abstraksi untuk memecah pola urutan bilangan

Indeks	Nilai	Rumus/pola
1	6	$n + 5$
2	4	$n + 2$
3	8	$n + 5$
4	6	$n + 2$
5	10	$n + 5$
6	8	$n + 2$
7	12	$n + 5$
8	?	$n + 2$

A

Konsep Berpikir Komputasional

01

Berpikir Komputasional







d. Algoritma

Penyusunan algoritma adalah proses membangun penyelesaian masalah dengan langkah-langkah dan aturan yang sesuai.

Ada dua cara untuk menuliskan algoritma, yaitu menggunakan pseudocode dan flowchart.

- **Pseudocode** adalah rangkaian instruksi yang menggambarkan langkah-langkah untuk menyelesaikan permasalahan
- **Flowchart** adalah diagram yang mewakili kumpulan dari instruksi-instruksi.

Simbol flowchart

No.	Nama	Simbol	Kegunaan
1.	Mulai/berhenti		Digunakan untuk memulai dan mengakhiri proses
2.	Proses		Menjalankan perintah
3.	Keputusan		Membuat keputusan/pilihan
4.	Masukan/keluaran (input/output)		Menerima input atau menampilkan output pada pengguna
5.	Penghubung		Berpindah ke bagian lain di flowchart
6.	Arah aliran		Menunjukkan arah aliran flowchart

A

Konsep Berpikir Komputasional

02

Mengevaluasi Solusi

a. Solusi Tidak Didekomposisi dengan Benar

Solusi tidak didekomposisi dengan benar, artinya pada proses dekomposisi, kita tidak melakukannya dengan benar.

Contoh kita tidak menyertakan bagaimana proses membuat bumbu yang benar, atau sebagai ganti bagaimana kriteria daging yang baik, kita justru membahas bagaimana cara memotong kambing dengan benar.

b. Solusi Tidak Lengkap

Solusi yang tidak lengkap, artinya ada bagian permasalahan yang belum diselesaikan.

Contoh, kita tidak membuat solusi untuk menemukan bumbu terbaik, atau lupa mencari tahu bagaimana cara membakar sate yang baik. Ketika solusi dihasilkan, masih ada bagian permasalahan yang belum diselesaikan

A

Konsep Berpikir Komputasional

02

Mengevaluasi Solusi

c. Solusi Tidak Efisien

Hal ini berarti, ketika solusi dijalankan, maka penggunaan sumber daya seperti penggunaan sumber daya manusia, dana, waktu, peralatan, dan bahan-bahan tidak dilakukan dengan efisien.

d. Solusi Tidak Sesuai dengan Kriteria yang Ditentukan

Hasil dari solusi yang diberikan tidak sesuai dengan yang ditetapkan pada awalnya.

Sebagai contoh, jika di awal kita ingin membuat sate kambing madura yang terbaik, solusi yang diberikan haruslah memenuhi kriteria tersebut, yaitu resep untuk sate kambing madura, bukan sate kambing maranggi atau lainnya.

B**Berbagai Kasus Berpikir Komputasional****01****Menjumlahkan Bilangan**

Contoh : Hitung $1 + 2 + 3 + \dots + 100$ dengan cepat.

Prinsip yang digunakan : Pengenalan Pola dalam Berpikir Komputasional.

Penyelesaian

Susun angka menjadi 2 baris:

Baris 1: 1, 2, 3, ..., 50

Baris 2: 100, 99, 98, ..., 51

Setiap kolom: jumlah = 101

untuk menghitung jumlah dari angka 1 sampai 100:

$$101 \times 50 = 5050$$

	N1	N2	N3	N4	N5	...	N50
Baris 1	1	2	3	4	5	...	50
Baris 2	100	99	98	97	96		51
Jumlah	101	101	101	101	101		101

Kesimpulan:

Berpikir komputasional membantu menyederhanakan masalah.

Pengenalan pola menjadi solusi lebih cepat & efisien.

Bisa diterapkan untuk berbagai masalah matematika & logika.

B

Berbagai Kasus Berpikir Komputasional

02

Membuat Prosedur

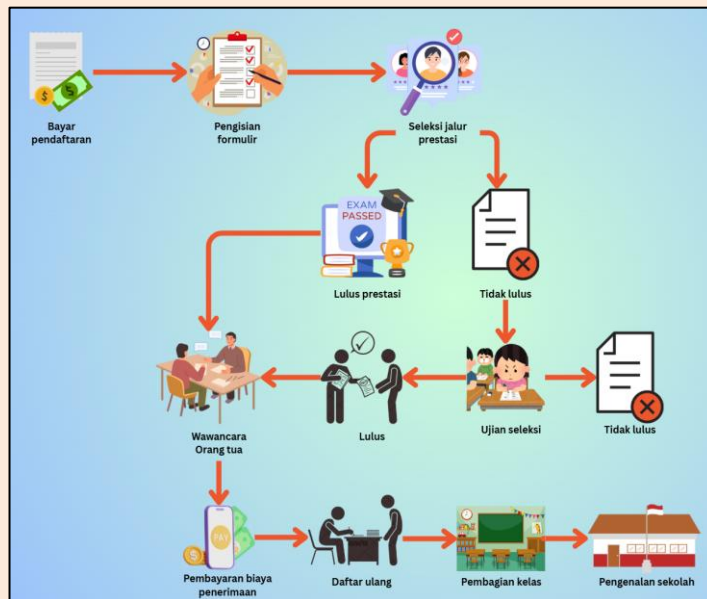
Contoh: Menyusun prosedur penerimaan peserta didik baru

Prinsip yang digunakan: Algoritma untuk menyusun langkah terurut sampai mencapai tujuan.

Menyusun prosedur berarti membuat urutan langkah yang jelas, sistematis, dan dapat diikuti siapa saja untuk mencapai tujuan tertentu.

Prinsip algoritma membantu memastikan bahwa: Tidak ada langkah yang terlewat.

Urutan pekerjaan logis dan efisien.



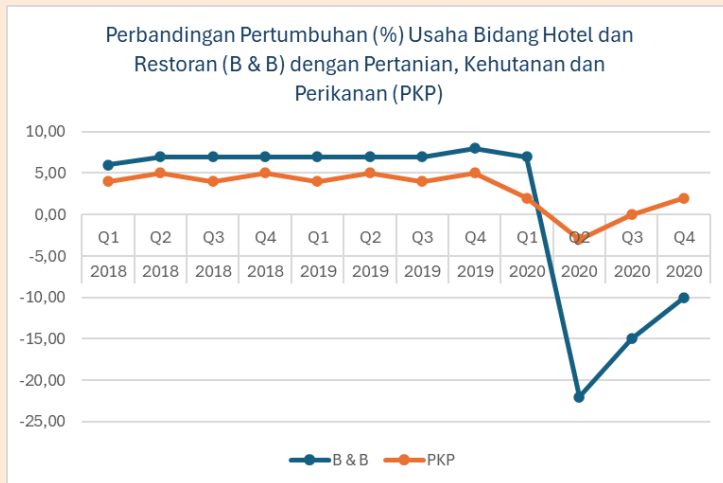
B

Berbagai Kasus Berpikir Komputasional

03

Menerapkan Berpikir Komputasional dengan Data

Contoh: Pertumbuhan bidang usaha dalam berbagai bidang.



a. Dekomposisi

Pecah grafik kompleks jadi bagian lebih kecil (stabil, turun, naik) untuk memudahkan analisis.

b. Abstraksi

Fokus pada bagian penting (titik tertinggi, terendah, selisih terbesar/kecil), abaikan yang tidak relevan.

c. Pengenalan Pola

Temukan pola naik/turun/stabil untuk memahami tren dan mengevaluasi tindakan.

d. Algoritma

Susun langkah sistematis untuk mencari penyebab dan merancang solusi berdasarkan data.