



Modul 2

# Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial

(Bimbingan Teknis Guru Koding dan Kecerdasan Artifisial  
Jenjang SD)



# Modul 2

## Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial

### Pengarah:

Direktur Jenderal Guru, Tenaga Kependidikan dan Pendidikan Guru

### Penanggung Jawab:

Direktur Guru Pendidikan Dasar

### Koordinator:

Dr. Nita Isaeni, M.Pd.  
Dr. Medira Ferayanti, S.S., M.A

### Penulis:

Ester Margareth Wagiu, S.Pd.

### Tim Ahli Materi:

Dr. Asep Wahyudin  
Septiaji Eko Nugroho, S.T, M.Sc.,  
Dr. Asep Jihad, M.Pd.

### Kontributor:

Dwi Setiyowati, S.Si.  
Irwan Nuriwansyah, S.Pd.  
Ita Utari  
Rohmi Nurwiyati  
Isti Marina Sarida  
Amar Nugraha

### Layout/desain:

Yane Hendarrita

### Dikeluarkan oleh:

Direktorat Guru Pendidikan Dasar  
Direktorat Jenderal Guru, Tenaga Kependidikan dan Pendidikan Guru  
Kementerian Pendidikan Dasar dan Menengah  
Kompleks Kemendikbud, Jalan Jenderal Sudirman, Senayan, Jakarta, 10270

Copyright © 2025

Hak Cipta Dilindungi Undang-Undang

Karya ini dilisensikan di bawah lisensi Creative Commons Attribution-NonCommercial-No

Derivatives 4.0 International (CC BY-NC-SA 4.0) Dilarang memperbanyak sebagian atau keseluruhan isi buku ini untuk kepentingan komersil tanpa izin tertulis dari Kementerian Pendidikan Dasar dan Menengah

# Daftar Isi

Daftar Isi.....	3
Kata Pengantar.....	4
A. Deskripsi Umum Modul.....	6
A.1. Capaian Pelatihan.....	6
A.2. Tujuan Pelatihan.....	6
A.3. Indikator Capaian Pelatihan.....	6
A.4. Pokok Bahasan.....	6
A.5. Alur Pelatihan.....	7
B. Konsep Berpikir Komputasional.....	9
B.1. Definisi Berpikir Komputasional.....	9
B.2. Berpikir Komputasional dan Kompetensi Abad 21.....	9
B.3. Orientasi Kritis dalam Berpikir Komputasional.....	9
B.4. Prinsip Berpikir Komputasional.....	9
B.5. Pentingnya Berpikir Komputasional untuk Memecahkan Berbagai Masalah dalam Kehidupan sehari-hari.....	14
C. Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial (KA).....	15
C.1. Pembelajaran Berpikir Komputasional yang Terintegrasi dengan Koding.....	15
C.1.1 Menghubungkan Prinsip Berpikir Komputasional dengan Koding.....	15
C.1.2 Kegiatan Yang Bisa Dilakukan dalam Kelas.....	15
C.2. Pembelajaran Berpikir Komputasional sebagai Landasan untuk Kecerdasan Artifisial (KA).....	17
C.3. Masukan ( <i>Input</i> ), Proses, Luaran ( <i>Output</i> ).....	18
C.4. Variabel.....	19
C.5. Alur Logika: Sekuensial, Percabangan dan Perulangan.....	20
C.6. Contoh Penerapan Pembelajaran Berpikir Komputasional.....	21
D. Pembelajaran Berpikir Komputasional.....	22
D.1. Pembelajaran <i>Plugged</i> dan <i>Unplugged</i> .....	22
D.2. Menyiastasi Pembelajaran <i>Plugged</i> menjadi <i>Unplugged</i> .....	23
Aktivitas dan Lembar Kerja.....	24
Lembar Kerja 2.1 Membuat Rancangan Penyelesaian.....	24
Lembar kerja 2.2 Membuat <i>Mind Map</i> .....	26
Kuis Singkat untuk Memahami Berpikir Komputasional Sebagai Dasar Koding.....	27
Daftar Pustaka.....	29

## Kata Pengantar

Assalamualaikum Warahmatullahi Wabarakatuh, Salam sejahtera bagi kita semua,

Puji syukur kehadiran Allah Subhanahu Wa Ta'ala atas rahmat dan karunia-Nya, sehingga Modul Bimbingan Teknis (Bimtek) Training of Trainers (TOT) Koding dan Kecerdasan Artifisial untuk Fase C ini siap digunakan. Modul ini menjadi bagian penting dalam upaya kita bersama untuk mengakselerasi penguasaan kompetensi abad ke-21 bagi murid di seluruh pelosok negeri.

Sebagaimana kita ketahui, arah kebijakan pendidikan saat ini menempatkan penguatan sumber daya manusia sebagai prioritas utama. Salah satu pilar penting dalam mewujudkan visi tersebut adalah melalui transformasi pendidikan yang adaptif terhadap perkembangan teknologi. Koding dan Kecerdasan Artifisial (KA) bukan lagi sekadar tren, melainkan fondasi krusial bagi kemajuan bangsa di era digital ini.

Kementerian Pendidikan Dasar dan Menengah telah merilis Naskah Akademik Pembelajaran Koding dan Kecerdasan Artifisial yang menjadi landasan filosofi. Naskah akademik ini menggarisbawahi pentingnya integrasi Koding dan KA dalam proses pembelajaran yang memberdayakan guru dan murid untuk menjadi inovator di masa depan.

Modul Bimtek TOT Fase C ini dirancang secara khusus untuk membekali para fasilitator dengan pemahaman yang terintegrasi dengan pembelajaran mendalam serta keterampilan yang dapat mentransformasikan konsep pembelajaran Koding dan KA kepada para guru. Modul ini mencakup materi-materi esensial, mulai dari kedudukan mata pelajaran Koding dan Kecerdasan Artifisial dalam kurikulum nasional, penguatan Berpikir Komputasional sebagai landasan utama, hingga pemahaman konsep dasar dan pemanfaatan Kecerdasan Artifisial yang relevan untuk peserta didik Fase C. Modul ini juga membekali para fasilitator dengan pendekatan pembelajaran mendalam serta pedagogik yang efektif dan inovatif dalam memfasilitasi pembelajaran Koding dan KA.

Kami berharap, melalui Bimtek ini, para fasilitator dapat menjadi agen perubahan yang mampu menularkan semangat dan pengetahuan tentang Koding dan KA kepada para guru pendidikan dasar di seluruh Indonesia. Dengan guru yang kompeten dan berkualitas, diharapkan akan mampu mendidik generasi penerus bangsa yang tidak hanya cakap dalam menggunakan teknologi, tetapi juga mampu mencipta dan berinovasi dengan memanfaatkan potensi Koding dan KA.

Akhir kata, saya menyampaikan apresiasi yang setinggi-tingginya kepada tim penyusun modul yang telah bekerja keras menghasilkan materi yang komprehensif dan relevan ini. Semoga Allah SWT senantiasa melimpahkan rahmat dan hidayah-Nya kepada kita semua dalam upaya memajukan pendidikan untuk semua dengan memberikan standar pelayanan yang Responsif, Akuntabel, Melayani, Adaptif, dan Harmonis (RAMAH).

Wassalamualaikum Warahmatullahi Wabarakatuh.

Hormat saya,  
Direktur Guru Pendidikan Dasar  
Kementerian Pendidikan Dasar dan  
Menengah



Dr. Rachmadi Widdiharto, MA.

## A. Deskripsi Umum Modul

### A.1. Capaian Pelatihan

Pelatihan berpikir komputasional ini ditujukan kepada setiap guru yang akan mengajarkan kecerdasan artifisial dan koding di jenjang sekolah dasar. Pada akhir pelatihan peserta tidak hanya mengerti namun juga memiliki pemahaman yang dalam mengenai konsep berpikir komputasional sehingga pelaksanaan di dalam kelas dapat dilakukan dengan maksimal. Pada pelatihan ini peserta pelatihan akan mendapatkan penjelasan mengenai apa itu berpikir komputasional, aktivitas dan juga kerangka berpikir untuk berpikir komputasional.

### A.2. Tujuan Pelatihan

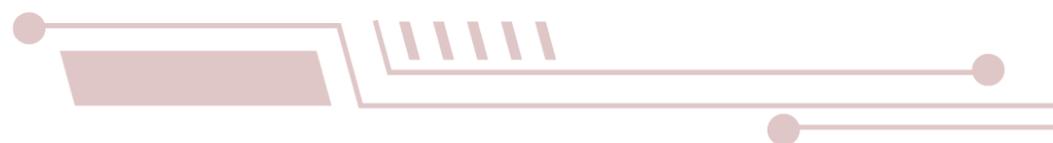
1. Peserta pelatihan mampu memahami konsep dasar berpikir komputasional.
2. Peserta pelatihan mampu menguraikan dan memberi contoh penerapan berpikir komputasional dalam menyelesaikan berbagai permasalahan dalam kehidupan sehari-hari.

### A.3. Indikator Capaian Pelatihan

- 1.1. Peserta pelatihan dapat mengenali dan mengingat konsep dasar berpikir komputasional, termasuk definisi, prinsip, dan karakteristiknya.
- 1.2. Peserta pelatihan dapat menjelaskan keterkaitan berpikir komputasional dengan cara berpikir logis dan pemecahan masalah dalam berbagai konteks.
- 2.1. Peserta pelatihan dapat menguraikan bagaimana berpikir komputasional diterapkan dalam menyelesaikan berbagai permasalahan kehidupan sehari-hari.
- 2.2. Peserta pelatihan dapat memberikan contoh penerapan berpikir komputasional dalam menyelesaikan masalah nyata

### A.4. Pokok Bahasan

1. Definisi berpikir komputasional.
2. Prinsip berpikir komputasional
3. Komputasional sebagai dasar koding.
4. Aktivitas *plugged* dan *unplugged*.



## A.5. Alur Pelatihan

Alur pelatihan modul Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial menggunakan *SOLO Taxonomy* yang dirinci pada Tabel 1.1.

Tabel 1.1. Alur Pelatihan

Hari ke-	Bab/Topik Utama	Tahapan dalam Pembelajaran Mendalam	Indikator	Aktivitas	Evaluasi	JP
Hari ke-2	Konsep Berpikir Komputasional I	Memahami (Unistruktural dan Multistruktural)	<ul style="list-style-type: none"> <li>Peserta mampu mengenali dan memahami konsep berpikir komputasional, prinsip-prinsipnya, serta relevansinya dalam pemecahan masalah.</li> </ul>	<ul style="list-style-type: none"> <li>Fasilitator memaparkan materi dari definisi, orientasi berpikir, prinsip dan penjelasan mengenai pentingnya berpikir komputasional dalam kehidupan sehari-hari.</li> </ul>	Membaca modul 2 Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial.	1
		Mengaplikasi (Relasional)	<ul style="list-style-type: none"> <li>Peserta mampu menerapkan berpikir komputasional dalam menyelesaikan masalah dan memberikan contoh nyata penerapannya.</li> </ul>	<ul style="list-style-type: none"> <li>Fasilitator memberikan contoh kegiatan yang bisa digunakan untuk membantu peserta mengerti cara berpikir komputasional sebagai landasan Kecerdasan Artifisial.</li> <li>Fasilitator memberikan waktu kepada peserta untuk mengerjakan LK 2.1 mengenai studi kasus.</li> </ul>	<ul style="list-style-type: none"> <li>Membaca modul 2</li> <li>Mengerjakan <a href="#">LK 2.1</a> Membuat rancangan penyelesaian masalah (studi kasus).</li> </ul>	2
		Merefleksi (Abstrak Meluas)	<ul style="list-style-type: none"> <li>Peserta mampu menuliskan bagaimana berpikir komputasional dapat membantu dalam pembelajaran dan kehidupan sehari-hari</li> </ul>	<ul style="list-style-type: none"> <li>Fasilitator membimbing peserta untuk melakukan refleksi.</li> </ul>	Mengisi jurnal refleksi.	1
	Berpikir Komputasional I sebagai Dasar Koding & Pembelajaran	Memahami (Unistruktural dan Multistruktural)	<ul style="list-style-type: none"> <li>Peserta mampu mengenali dan memahami bagaimana berpikir komputasional dapat diterapkan dalam pembelajaran koding, serta membandingkan metode <i>plugged</i> dan <i>unplugged</i>.</li> </ul>	<ul style="list-style-type: none"> <li>Fasilitator memaparkan kegiatan dan manfaat kegiatan <i>plugged</i> dan <i>unplugged</i> dalam pembelajaran berpikir komputasional dan Kecerdasan Artifisial.</li> </ul>	Membaca modul 2	1



<b>Berpikir Komputasiona I</b>	Mengaplikasi (Relasional)	<ul style="list-style-type: none"> <li>• Peserta mampu menerapkan konsep yang telah dipelajari dengan menyusun rencana pembelajaran berpikir komputasional menggunakan metode <i>plugged</i> dan <i>unplugged</i>, serta mendemonstrasikan cara penggunaannya dalam pembelajaran koding.</li> </ul>	<ul style="list-style-type: none"> <li>• Fasilitator melakukan demonstrasi kegiatan <i>plugged</i> dan <i>unplugged</i>.</li> <li>• Peserta mencoba melakukan kegiatan <i>plugged</i> dan <i>unplugged</i> bersama-sama.</li> <li>• Fasilitator memberikan waktu kepada peserta untuk mengerjakan Lembar kerja 2.2 membuat <i>mind map</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Mengerjakan LK <a href="#">2.2 Membuat Mind Map</a>.</li> </ul>	2
	Merefleksi (Abstrak Meluas)	<ul style="list-style-type: none"> <li>• Peserta mampu mengevaluasi efektivitas metode yang digunakan dan bagaimana berpikir komputasional dapat diterapkan lebih baik dalam pembelajaran.</li> </ul>	<ul style="list-style-type: none"> <li>• Fasilitator membimbing peserta untuk melakukan refleksi</li> <li>• Peserta diberikan waktu menyelesaikan Kuis Singkat di LMS</li> </ul>	<ul style="list-style-type: none"> <li>• Mengisi jurnal refleksi</li> <li>• Mengerjakan <a href="#">Kuis Singkat</a></li> </ul>	1

## B. Konsep Berpikir Komputasional

### B.1. Definisi Berpikir Komputasional

Berpikir komputasional berarti peserta didik memiliki kemampuan berpikir yang dapat menyelesaikan permasalahan ataupun persoalan sehari-hari dengan menggunakan teknik ilmu komputer (Bebras Indonesia, n.d.). Berpikir komputasional juga dapat diartikan dengan pemecahan masalah dengan solusi menggunakan konsep dasar berpikir dalam informatika (Tresnawati et al., n.d.). Dengan demikian, berpikir komputasional merupakan kemampuan menyusun solusi masalah secara logis dan sistematis sehingga dapat diproses oleh manusia maupun komputer. Dalam Kepka BSKAP nomor 032 tahun 2024 halaman 239 mendeskripsikan berpikir komputasional adalah keterampilan *problem solving* yang berjenjang melalui pemodelan dan melalui simulasi untuk menghasilkan solusi efektif, efisien, dan optimal yang dapat dijalankan oleh manusia atau mesin meliputi penalaran logis, kritis, dan kreatif berdasarkan data, baik secara mandiri maupun berkolaborasi.

### B.2. Berpikir Komputasional dan Kompetensi Abad 21

Keterampilan berpikir komputasional berperan penting dalam kehidupan sehari-hari dan semakin dibutuhkan di era Abad 21. Kompetensi ini berkaitan erat dengan konsep 4C, yaitu *Critical Thinking* (berpikir kritis), *Creativity* (kreativitas), *Communication* (komunikasi), dan *Collaboration* (kolaborasi) (Kemendikbud Ristek, 2024, 188). Keempat keterampilan ini mendukung penyelesaian masalah secara sistematis, inovatif, dan efektif. Oleh karena itu, berpikir komputasional menjadi dasar dalam mengembangkan kemampuan yang relevan dengan tantangan masa kini.

### B.3. Orientasi Kritis dalam Berpikir Komputasional

Orientasi Kritis dalam Berpikir Komputasional mengacu pada pendekatan berpikir komputasional yang tidak hanya berfokus pada pemecahan masalah secara sistematis, tetapi juga mempertimbangkan aspek kritis seperti etika, dampak sosial, dan keberlanjutan solusi yang dihasilkan (Kemendikbud Ristek, 2024, 235). Ini berarti bahwa dalam menerapkan berpikir komputasional, seseorang tidak hanya mengikuti prosedur algoritmik, tetapi juga menganalisis dan mengevaluasi masalah serta solusi secara lebih mendalam. 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 (Meitjing & Fuad, 2023, 2).

### B.4. Prinsip Berpikir Komputasional

Prinsip berpikir komputasional mencakup dekomposisi, pengenalan pola, abstraksi, dan berpikir algoritma (Supiarmo et al., 2021, 372). Keempat prinsip ini kemudian akan dijelaskan dan diuraikan dengan penggunaan aktivitas untuk mempermudah pengenalan dengan prinsip berpikir komputasional tersebut. Adapun prinsip dan aktivitas tersebut terdiri dari:

## 1. Decomposition (Dekomposisi/Pemecahan Masalah)

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. (Supiarmono et al., 2021, 4).

### Kegiatan yang bisa dilakukan untuk semakin memahami arti dari *Decomposition*:

#### 1. Pesta Ulang Tahun.

**Tujuan:** Peserta didik belajar memecah masalah besar menjadi bagian kecil yang lebih mudah diselesaikan.

**Alat:** Kertas, pensil, atau papan tulis.

#### Langkah-langkah:

1. Guru memberikan tantangan: *“Bayangkan jika ingin membuat sebuah pesta ulang tahun yang sempurna. Apa saja yang harus dipersiapkan?”*
2. Peserta didik menuliskan **bagian-bagian kecil** dari perencanaan pesta, misalnya:
  - Memilih tempat
  - Mengundang teman
  - Menyiapkan makanan
  - Menentukan permainan
3. Setelah semua bagian terkumpul, diskusikan dan dijelaskan kembali kepada peserta didik bagaimana dekomposisi membuat pekerjaan lebih mudah dikelola.

CHECKLIST

### Keperluan Pesta Ulang Tahun

TULISKAN KEBUTUHAN YANG PERLU DISIAPKAN  
UNTUK ACARA ULANG TAHUN

- Memilih tempat
- Mengundang teman
- Menyiapkan makanan
- Menentukan permainan

Gambar 1.1 Contoh kegiatan dekomposisi

## 2. *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 (Supiarmino et al., 2021, 4).

### Kegiatan yang bisa dilakukan untuk semakin memahami arti dari *Pattern Recognition*:

**Tujuan:** Peserta didik belajar menemukan pola dari berbagai situasi agar dapat digunakan kembali.

**Nama Kegiatan:** Tebak bentuk!

**Alat:** Kartu bergambar atau papan tulis.

**Langkah-langkah:**

1. Guru menampilkan beberapa contoh pola, misalnya:
  - Pola angka: 2, 4, 6, 8, ... (lanjutannya?)
  - Pola bentuk: ● ▲ ● ▲ ... (apa yang selanjutnya?)
  - Pola dalam aktivitas sehari-hari (jadwal sekolah, langkah dalam bermain gim).
2. Peserta didik mencoba menebak dan menjelaskan polanya.
3. Guru menjelaskan bahwa **dalam komputer, pola sering digunakan untuk membuat program lebih efisien.**

**Nama Kegiatan:** *Code Breakers!*

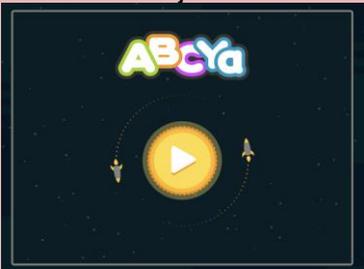
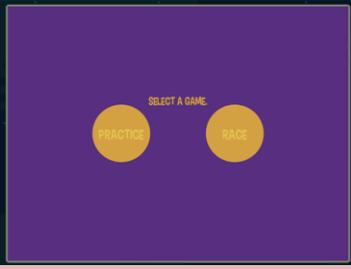
**Alat:** Ruang untuk bermain.

**Langkah-langkah:**

1. Terdiri dari 4 siswa dan 3 siswa lainnya (jumlahnya bisa menyesuaikan sesuai kebutuhan)
2. Guru memberikan instruksi kepada 4 siswa pertama untuk membentuk sebuah pola (seperti pose, atau gerakan).
3. Siswa memperagakan pola yang sudah mereka diskusikan, tiga siswa lainnya melengkapi pola yang dibentuk oleh empat siswa sebelumnya.

Untuk aktivitas kelas menggunakan komputer bisa menggunakan:

[https://www.abcya.com/games/shape\\_patterns](https://www.abcya.com/games/shape_patterns)

<p>Tautan Pratinjau:</p> 		
<p>Gambar 1.2 Tampilan awal dari tautan</p>	<p>Gambar 1.3 Tampilan setelah icon play ditekan</p>	<p>Gambar 1.4 Tampilan pilihan kegiatan yang bisa digunakan</p>

### 3. *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 (Supiarmono et al., 2021, 4).

Kegiatan yang bisa dilakukan untuk semakin memahami arti dari *Abstraction*:

1. **Tujuan:** Peserta didik belajar mengambil informasi penting dan mengabaikan hal yang tidak relevan.

**Alat:** Kartu berisi deskripsi benda atau tempat tanpa menyebutkan namanya.

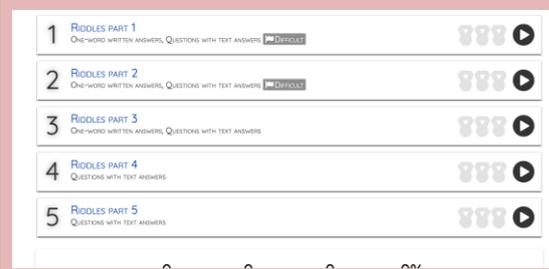
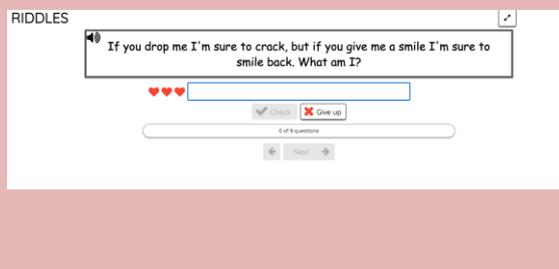
**Langkah-langkah:**

1. Guru menyiapkan beberapa kartu berisi deskripsi singkat, misalnya:
  - o Aku adalah tempat orang membeli makanan, ada meja dan kursi, tapi bukan di rumah. Aku di mana? (Restoran)
  - o Aku adalah alat untuk menulis, biasanya panjang dan runcing, sering dipakai di sekolah. Aku apa? (Pensil)
2. Peserta didik mencoba menebak hanya dari **informasi penting** tanpa detail berlebihan.
3. Setelah beberapa ronde, guru menjelaskan bahwa **abstraksi membantu untuk fokus pada informasi yang benar-benar dibutuhkan** dalam menyelesaikan masalah, seperti tampilan sederhana dalam aplikasi *Google Maps*.

Untuk aktivitas kelas menggunakan komputer bisa menggunakan *game riddle*:

<https://www.helpfulgames.com/subjects/brain-training/1088-riddles.html>

Tautan Pratinjau:

	
<p>Gambar 1.6 Tampilan awal halaman dan permainan yang akan digunakan</p>	<p>Gambar 1.7 Tampilan contoh permainan Keterangan: Permainan <i>riddle</i> yang akan dimainkan dengan cara mengetikkan jawaban di dalamnya dan jika teks berwarna hijau maka jawaban benar dan jika merah maka jawaban tidak tepat</p>

### 4. *Algorithmic Thinking* (Berpikir Algoritmik)

Berpikir algoritmik yaitu menyusun langkah-langkah, instruksi ataupun prosedur yang sistematis untuk menyelesaikan suatu masalah tertentu (Supiarmono et al., 2021, 4)

Kegiatan yang bisa dilakukan untuk semakin memahami arti dari **Algorithmic Thinking**:

1. Tujuan: Peserta didik memahami bagaimana menyusun langkah-langkah sistematis untuk menyelesaikan masalah.

Alat: Buku resep atau instruksi acak.

Langkah-langkah:

1. Guru menyiapkan beberapa lembar/ buku berisi langkah-langkah membuat sesuatu, tetapi dalam urutan yang acak (misalnya: membuat roti lapis, membuat origami, atau menggambar rumah).
2. Peserta didik bekerja dalam kelompok untuk menyusun ulang langkah-langkah menjadi urutan yang benar.
3. Setelah selesai, satu kelompok membaca urutan mereka, dan kelompok lain memeriksa apakah bisa diikuti dengan logis.
4. Guru menjelaskan bahwa berpikir algoritmik adalah menyusun langkah-langkah yang benar agar suatu tugas bisa diselesaikan dengan baik, seperti instruksi dalam program komputer.

Untuk aktivitas kelas menggunakan komputer bisa menggunakan:

<https://planeta42.com/it/algorithms/game.html> gim daring yang memperkenalkan sistem *flowchart*.

atau: <https://planeta42.com/it/gameinstall/game.html> untuk mencoba mengerti apa saja langkah-langkah dalam memasang aplikasi.

		
<p>Gambar 1.8 Tampilan awal halaman web</p>	<p>Gambar 1.9 Tampilan halaman sebelum memulai permainan mengenai bagaimana cara flowchart bekerja</p>	<p>Gambar 2.1 Permainan di halaman web dimulai</p>
		
<p>Gambar 2.2 Tampilan awal halaman web.</p>	<p>Gambar 2.3 Tampilan halaman sebelum memulai permainan mengenai bagaimana cara menginstalasi aplikasi pertahapannya.</p>	<p>Gambar 2.4 Permainan di halaman web dimulai</p>

## B.5.Pentingnya Berpikir Komputasional untuk Memecahkan Berbagai Masalah dalam Kehidupan sehari-hari

Di era digital saat ini, berpikir komputasional bukan hanya keterampilan bagi ilmuwan komputer, tetapi juga kemampuan esensial dalam kehidupan sehari-hari. Kemampuan ini membantu individu dalam berbagai situasi, mulai dari pengambilan keputusan, manajemen waktu, hingga pemecahan masalah yang kompleks (Christi & Rajiman, 2023, 2).

Kemampuan berpikir komputasional yang dimaksud dapat digunakan dalam keseharian seperti berikut.

1. Membantu menyelesaikan masalah dengan lebih mudah
2. Membantu membuat keputusan yang lebih baik
3. Mengajarkan cara berpikir kreatif dan terstruktur
4. Membantu menghemat waktu dan lebih efisien
5. Membantu beradaptasi dengan teknologi

Berpikir komputasional melatih otak manusia agar tetap dapat berpikir secara logis dan sistematis, sehingga mampu menyelesaikan masalah lainnya yang berhubungan dengan kehidupan sehari-hari.

### [Modul Lembar Kerja 2.1 Studi Kasus](#) (Selengkapnya Tekan Tautan)

Skenario:

Banyak peserta didik tidak mampu menyelesaikan permasalahan sehari-hari secara sistematis. Bagaimana Anda akan membimbing mereka untuk dapat berpikir secara logis, terstruktur, prosedural dan sistematis?

**Buatlah desain infografis rencana implementasi sederhana tentang bagaimana Anda akan mengajarkan konsep berpikir komputasional yang dapat mendukung pembelajaran koding pada peserta didik melalui penyelesaian masalah sehari-hari menggunakan media visual (seperti canva, photoshop, photopea, dll).**

## C. Berpikir Komputasional sebagai Dasar Koding dan Kecerdasan Artifisial (KA)

### C.1. Pembelajaran Berpikir Komputasional yang Terintegrasi dengan Koding

#### C.1.1 Menghubungkan Prinsip Berpikir Komputasional dengan Koding

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

Tabel 1.2. Hubungan berpikir komputasional dengan koding

#### C.1.2 Kegiatan Yang Bisa Dilakukan dalam Kelas

- Strategi Pembelajaran Berbasis Koding (*plugged*)
  - Menggunakan aplikasi kode blok: **Scratch** atau **Scratch Junior**  
Aktivitas:
    - Membuat Animasi Sederhana: peserta didik dapat diminta untuk membuat animasi sederhana menggunakan urutan kode yang benar.
    - Membuat Permainan Sederhana: Permainan sederhana dibuat menggunakan konsep percabangan (*if-else*) atau perulangan(loop) dan kondisi di blok kode.
  - Menggunakan aplikasi kode blok: **Blockly** atau **Code.org**  
Aktivitas:
    - Menggunakan *blockly* untuk menyusun blok kode dan menyelesaikan misi.
    - Menggunakan *Loop* dan *If-Else*: Menerapkan pengulangan dalam permainan koding sederhana.

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

- **Kegiatan 1: Dekomposisi - “Program Your Pals.”**

- Tujuan: peserta didik belajar memecah tugas besar menjadi beberapa langkah kecil.
- **Aktivitas:**
  - Guru meminta peserta didik menuliskan langkah-langkah detail untuk menggerakkan “robot” ke titik tertentu di kelas.
  - peserta didik lain mengikuti instruksi tersebut (seperti robot).
  - Guru menghubungkan konsep ini dengan menulis kode dalam koding sesungguhnya, di mana setiap langkah kecil adalah bagian dari kode.



Gambar 2.5 Contoh permainan kelas  
©Meghan Zigmond - Unplugged: Human Coding Grid

- **Kegiatan 2: Algoritma - “Menulis Kode Resep”**

- Tujuan: peserta didik memahami konsep algoritma dalam koding.
- **Aktivitas:**
  - Peserta didik menulis langkah-langkah membuat roti lapis seperti kode program.
  - Guru menunjukkan bagaimana logika ini diterapkan dalam program koding menggunakan blok kode berurutan.



Gambar 2.6 Contoh penjelasan algoritma

- **Kegiatan 3: Abstraksi - “Filter Informasi”**

- Tujuan: peserta didik memahami pentingnya menyaring informasi yang relevan dalam pemrograman.
- **Aktivitas:**
  - Guru memberikan deskripsi panjang tentang sesuatu deskripsi, tetapi hanya beberapa bagian penting yang berguna.
  - Peserta didik harus menyusun ulang informasi agar lebih ringkas dan jelas.
  - Guru menjelaskan bagaimana dalam koding, kita hanya menggunakan variabel yang diperlukan agar kode tidak terlalu rumit.
  - **Contoh Deskripsi panjang:** Hewan adalah makhluk hidup yang memiliki berbagai bentuk, ukuran, dan habitat. Beberapa hewan bisa hidup di air, seperti ikan dan paus, sementara yang lain hidup di darat, seperti singa dan gajah. Ada juga hewan yang bisa terbang, seperti burung dan kelelawar. Hewan memiliki cara berkembang biak yang berbeda-beda, seperti bertelur atau melahirkan. Mereka juga memiliki sistem pencernaan yang berbeda tergantung pada jenis makanannya, seperti herbivora yang makan tumbuhan dan karnivora yang makan daging. Hewan sangat beragam, mulai dari yang kecil seperti semut hingga yang besar seperti paus biru (OpenAI, 2025).
  - **Deskripsi yang telah difilter:** Hewan adalah makhluk hidup yang memiliki berbagai bentuk dan habitat. Mereka berkembang biak dengan cara bertelur atau melahirkan serta memiliki sistem pencernaan yang beragam sesuai makanannya (herbivora atau karnivora) (OpenAI, 2025).
  - Penjelasan keterhubungan dengan pembelajaran koding: Dalam pemrograman, kita hanya menggunakan variabel dan kode yang relevan agar program tidak terlalu rumit, *programmer* menyaring data agar efisien dan mudah dipahami.

## **C.2. Pembelajaran Berpikir Komputasional sebagai Landasan untuk Kecerdasan Artifisial (KA)**

Pembelajaran berpikir komputasional sebagai landasan untuk kecerdasan artifisial (KA) bukan lagi hal yang berdiri sendiri, melainkan satu kesatuan yang tidak terpisahkan. Menurut Pembelajaran Koding Dan Kecerdasan Artifisial Pada Pendidikan Dasar dan Menengah pada Naskah Akademik 2025 menyatakan bahwa pembelajaran berpikir komputasional sebagai landasan untuk kecerdasan artifisial tidak hanya sekedar untuk meningkatkan literasi digital ataupun hanya sekedar

menyelesaikan masalah sehari-hari, namun tujuan dari pada pembelajaran ini adalah untuk menyiapkan peserta didik di masa depan agar siap menghadapi tantangan global. Tujuan dari pembelajaran ini juga 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.

### C.3. Masukan (*Input*), Proses, Luaran (*Output*)

Konsep Masukan (*Input*) - Proses - Luaran (*Output*) (*IPO*) adalah dasar dari cara komputer bekerja dan juga digunakan dalam pemecahan masalah secara sistematis. Konsep ini juga bisa diaplikasikan dalam kehidupan sehari-hari dan pemrograman sederhana. Berikut adalah tabel penjelasan konsep *IPO* yang dapat membantu kita semakin mengerti prosesnya.

Tabel 1.3. Penjelasan konsep *IPO*.

Komponen	Penjelasan	Contoh dalam Kehidupan Sehari-hari	Contoh dalam Koding
<b>Input</b> (Masukan)	Data atau informasi yang dimasukkan ke dalam sistem.	Pada mesin cuci, input berupa: Baju kotor Deterjen Air	Menekan tombol "Start" pada gim.
<b>Proses</b> (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).
<b>Output</b> (Luaran)	Hasil akhir setelah proses dijalankan.	Hasilnya baju bersih dan cukup kering	Karakter di aplikasi bergerak sesuai perintah (misalkan bergerak ke kanan).

Kegiatan yang bisa dilakukan untuk peserta didik mengerti *IPO* bisa menggunakan:

#### Kegiatan 1: Simulasi *IPO* dengan "Mesin Jus Buah"

**Tujuan:** Memahami konsep masukan, proses, dan keluaran dengan contoh nyata.

- **Masukan:** Peserta didik memasukkan bahan jus dan menekan tombol mesin jus (misalnya, kiwi, semangka dan apel).



©Ecentio

Gambar 2.7 Gambar *blender*

- **Proses:** Mesin *blender* mencampur bahan menjadi jus.
- **Luaran:** Jus siap diminum.  
→ **Diskusi:** Hubungkan konsep ini dengan bagaimana komputer mengolah data.

Gim daring yang mendukung:

<https://studio.code.org/s/course1/lessons/4/levels/1>

		
<p>Gambar 2.8 Tampilan awal halaman</p>	<p>Gambar 2.9 Tampilan setelah video selesai ditonton</p>	<p>Gambar 3.1 Halaman web dan blok kode yang bisa dimainkan untuk simulasi IPO</p>
<p>Blok kode diletakan sesuai instruksi dan karakter akan bergerak sesuai instruksi</p>		

### Kegiatan 2: Permainan Interaktif - Mesin Penjawab Otomatis

**Tujuan:** Melatih peserta didik berpikir logis menggunakan konsep IPO.

- Peserta didik membuat *gim* menggunakan aplikasi pemrograman berbasis blok visual di mana karakter memberikan jawaban berdasarkan masukan.
- **Masukan:** Pengguna memilih angka (misalnya, 1 untuk "Halo", 2 untuk "Apa kabar?").
- **Proses:** Program memilih jawaban berdasarkan angka yang ditekan.
- **Keluaran:** Karakter menampilkan teks yang sesuai.

### Kegiatan 3: Permainan *unplugged* alternatif konsep IPO

**Tujuan:** Melatih peserta didik memahami konsep IPO melalui permainan sederhana.

- 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**".
- **Masukan:** "Simon" memberikan instruksi (seperti duduk, berdiri, hadap kanan, hadap kiri).
- **Proses:** Peserta mendengarkan, mengenali pola perintah, dan mengikuti instruksi.
- **Luaran:** Peserta didik bergerak sesuai instruksi.

## C.4. Variabel

Variabel dikutip dari jurnal *National Council of Teacher of Mathematics 2022* mengatakan bahwa variabel biasanya diwakili dengan huruf, mewakili angka, mewakili nilai yang sebenarnya berubah dari waktu ke waktu. Ini menandakan bahwa sesungguhnya variabel adalah wadah atau bentuk yang dapat dilihat, diukur ataupun diamati. Dalam koding, variabel digunakan untuk menyimpan data yang akan digunakan.

### Contoh dalam koding:

- Membuat variabel "Nyawa" yang berkurang jika karakter terkena rintangan.
- Membuat variabel "Waktu" yang menghitung mundur dalam *gim*.

## C.5. Alur Logika: Sekuensial, Percabangan dan Perulangan

Dalam pemrograman dan berpikir komputasional, alur logika terdiri dari tiga konsep utama:

Tabel 1.4. Alur logika dalam penerapannya.

Konsep	Penjelasan	Contoh dalam Kehidupan Sehari-hari	Contoh dalam Koding
Sekuensial (Urutan langkah-langkah atau prosedur)	Perintah dieksekusi satu per satu sesuai urutan.	Membuat teh: <ul style="list-style-type: none"><li>• Siapkan alat dan bahan: panci, kompor, air, teh celup, gula, sendok, dan gelas/cangkir.</li><li>• Tuang air secukupnya ke dalam panci.</li><li>• Nyalakan kompor dan letakkan panci di atasnya.</li><li>• Rebus air hingga mendidih.</li><li>• Masukkan teh celup ke dalam panci berisi air mendidih.</li><li>• Tambahkan gula sesuai selera.</li><li>• Aduk hingga teh dan gula tercampur rata.</li><li>• Matikan kompor setelah teh siap.</li><li>• Tuang teh ke dalam gelas atau cangkir.</li><li>• Sajikan dan nikmati.</li></ul>	Tekan tombol mulai, aplikasi memproses, visualisasi bergerak ke kanan sesuai instruksi.
Percabangan (Pemilihan/ <i>Decision Making</i> )	Program memilih jalur berbeda berdasarkan kondisi tertentu.	Jika hujan, maka membawa payung. Jika tidak hujan, maka tidak perlu membawa payung.	Jika skor $\geq 50$ , maka karakter naik level; jika tidak, karakter tetap di level awal.
Perulangan ( <i>Looping</i> )	Mengulangi perintah hingga kondisi tertentu terpenuhi.	Saat berjalan, kaki kanan melangkah ke depan dan dilanjutkan oleh kaki	Karakter di gim terus berlari sampai tombol "STOP" ditekan.

kiri kemudian dilakukan terus menerus hingga bergerak ke depan.

Kegiatan yang bisa dilakukan secara daring:  
<https://www.kidlocoding.com/>



Gambar 3.2

Salah satu situs yang bisa digunakan untuk mempelajari secara lengkap mulai dari *sequences*.

## C.6. Contoh Penerapan Pembelajaran Berpikir Komputasional.

### 1. Kegiatan 1: Mengetahui Variabel dengan Simulasi Skor Gim.

**Tujuan:** Memahami konsep variabel dengan permainan interaktif.

**Kegiatan:**

1. Guru membagi peserta didik menjadi dua kelompok dan memberikan permainan sederhana (misalnya, setiap jawaban benar mendapat 10 poin).
2. Peserta didik mencatat perubahan skor menggunakan variabel (misalnya, skor awal 0, lalu bertambah 10 setiap kali menjawab benar).
3. Peserta didik memahami bahwa variabel dapat bertambah atau berkurang sesuai kondisi.

### 2. Kegiatan 2: Percabangan dengan "Jalan ke Sekolah".

**Tujuan:** Memahami logika percabangan dalam kehidupan nyata.

**Kegiatan:**

1. Peserta didik diberikan skenario: "Jika hujan, maka gunakan payung; jika tidak hujan, maka tidak menggunakan payung." atau "Jika hujan, maka memesan taksi; jika tidak hujan, maka tidak memesan taksi".
2. Guru menghubungkan konsep ini dengan percabangan dalam koding, seperti if-else statement di aplikasi pemrograman berbasis blok visual.

### 3. Kegiatan 3: Perulangan dalam Pola Gerakan Karakter.

**Tujuan:** Memahami konsep perulangan melalui aplikasi pemrograman berbasis blok visual.

**Kegiatan:**

1. Peserta didik membuat karakter yang melompat terus-menerus selama tombol ditekan.
2. Menggunakan blok "*repeat until*" atau "*forever*" untuk mengulang gerakan tertentu.
3. Peserta didik menyadari bahwa *looping* membantu membuat program lebih efisien.

## D. Pembelajaran Berpikir Komputasional

### D.1. Pembelajaran *Plugged* dan *Unplugged*

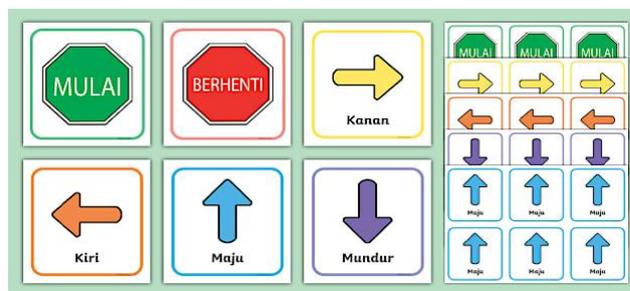
Pembelajaran berpikir komputasional dapat dilakukan dengan dua pendekatan utama: *Plugged* dan *Unplugged*.

#### 1. Pembelajaran *Plugged*

- Menggunakan perangkat teknologi seperti komputer, tablet, atau aplikasi untuk mengajarkan konsep berpikir komputasional.
- Contoh:
  - **Scratch** atau **Scratch Junior** untuk memahami algoritma dan pemrograman.
  - **Code.org** untuk belajar pemrograman berbasis blok dan lainnya.

#### 2. Pembelajaran *Unplugged*

- Tidak menggunakan perangkat elektronik, tetapi tetap mengajarkan konsep komputasional melalui aktivitas fisik dan permainan.
- Contoh:
  - **Permainan kartu algoritma** di mana peserta didik menyusun langkah-langkah membuat roti lapis dan lain sebagainya.
  - **Maze Challenge/ Human Grid** di mana peserta didik membuat rute berdasarkan instruksi yang diberikan.



Gambar 3.3 Contoh permainan algoritma ©Twinkl

- **Sorting Network** di mana peserta didik bekerja dalam kelompok untuk menyusun angka dalam urutan yang benar menggunakan aturan logika.

## D.2. Menyasati Pembelajaran *Plugged* menjadi *Unplugged*

Jika akses ke teknologi terbatas, pembelajaran berpikir komputasional tetap bisa dilakukan dengan metode *unplugged*. Berikut adalah beberapa strategi:

- **Mengubah Kode Blok menjadi Kartu Kode**
  - Jika peserta didik belajar pemrograman berbasis blok visual, buatlah kartu dengan simbol perintah seperti "Maju 10 langkah", "Putar 90 derajat", dll.
  - Peserta didik menyusun kartu tersebut untuk menyelesaikan tantangan tanpa komputer.
- **Role-Playing untuk Pemahaman Algoritma**
  - Terdiri dari tiga peserta didik yang berperan sebagai: Masukan (*Input*) (yang memberikan instruksi), Proses (yang berperan sebagai penggerak instruksi), Luaran (*Output*) (sebagai hasil dari proses).
  - Peserta didik yang berperan sebagai "Masukan" menginstruksikan untuk berputar ke kanan, peserta didik yang berperan sebagai "Proses" menggerakkan tubuh peserta didik "Luaran" berputar ke kanan dan peserta didik "Luaran" mengikuti arahan "Proses".
- **Menggunakan Storytelling untuk Konsep Pemrograman**
  - Guru memberikan sebuah cerita interaktif, dari cerita tersebut terdapat dua alternatif pilihan tindakan yang akan menentukan akhir cerita.
  - Kemudian peserta didik diminta untuk memilih keputusan atau alternatif tindakan mana yang akan dilakukan.
  - Guru menjelaskan bahwa pemilihan keputusan oleh peserta didik meniru konsep percabangan (*if-else*) dalam pemrograman.

### [Modul Lembar Kerja 2 Mind Map](#) (Selengkapnya Tekan Tautan)

#### Membuat *Mind Map*

Buatlah *mind map* di kertas atau diaplikasi yang berisikan penjelasan salah satu dari topik berikut dengan menarik:

- Menjelaskan konsep *IPO* dengan aktivitas interaktif (*unplugged* ataupun *plugged*).
- Menjelaskan Variabel dalam aplikasi koding menggunakan contoh sederhana.
- Menjelaskan Percabangan dan Perulangan dengan contoh permainan atau latihan sederhana.

# Aktivitas dan Lembar Kerja

## Lembar Kerja 2.1 Membuat Rancangan Penyelesaian Masalah (Studi Kasus)

### Lembar Kerja 2.1

#### Skenario:

Banyak siswa tidak bisa menyelesaikan tantangan koding karena terlalu mengikuti langkah tanpa berpikir kritis. Bagaimana Anda akan membimbing mereka untuk menggunakan Analisis Kritis terhadap Masalah?

Buatlah rencana implementasi sederhana menggunakan media presentasi tentang bagaimana anda akan mengajarkan konsep koding kepada siswa.

#### Daftar Checklist Indikator Keberhasilan:

- Guru telah menetapkan tujuan pembelajaran yang terukur sesuai dengan Taksonomi Solo(misalnya: Menghafal, Klasifikasikan, menerapkan dan Berteor).  
• Guru menggunakan pendekatan berbasis **problem-solving**, bukan hanya "ikuti tutorial".  
• Ada aktivitas **menganalisis dan memecah masalah** sebelum menulis kode.  
• Ada aktivitas yang melibatkan siswa, siswa tidak hanya menjadi **pengikut instruksi**, tetapi juga menjadi **pemecah masalah aktif**.

#### Rubrik Penilaian keberhasilan guru:

Aspek Penilaian	Deskripsi Kriteria	Skor 3 (Sangat Baik)	Skor 2 (Cukup Baik)	Skor 1 (Kurang Baik)
<b>Tujuan Pembelajaran yang Terukur</b>	Guru menetapkan tujuan pembelajaran yang jelas sesuai dengan tingkat pemahaman siswa SD.	Tujuan pembelajaran sangat jelas, menggunakan bahasa yang sederhana dan sesuai dengan perkembangan siswa kelas 5 SD.	Tujuan pembelajaran cukup jelas tetapi masih agak abstrak bagi siswa SD.	Tujuan tidak spesifik atau sulit dipahami oleh siswa.
<b>Pendekatan Berbasis Problem-Solving</b>	Guru menggunakan tantangan atau permainan untuk mengajarkan koding dan berpikir komputasional.	Siswa diberikan tantangan yang menarik, misalnya gim interaktif atau teka-teki koding yang mendorong mereka berpikir kritis.	Ada tantangan, tetapi masih banyak instruksi langsung tanpa eksplorasi.	Guru hanya memberikan perintah langkah demi langkah tanpa meminta siswa mencari solusi sendiri.
<b>Aktivitas Menganalisis Sebelum Koding</b>	Siswa diajak untuk berpikir sebelum menulis kode.	Siswa diberikan latihan sederhana seperti menyusun kartu perintah, atau	Siswa diberi sedikit latihan berpikir sebelum koding, tetapi masih	Siswa langsung mengetik kode tanpa berpikir atau merancang solusi

		menebak hasil kode sebelum menulisnya.	banyak langsung menetik kode tanpa analisis.	terlebih dahulu.
<b>Keterlibatan Siswa dalam Proses Belajar</b>	Siswa tidak hanya mengikuti instruksi, tetapi juga aktif berpartisipasi.	Siswa aktif bertanya, berdiskusi, mencoba berbagai solusi, dan bekerja sama dengan teman.	Siswa cukup terlibat, tetapi masih ada yang hanya mengikuti arahan tanpa mencoba sendiri.	Siswa hanya menyalin kode tanpa memahami atau mencoba eksplorasi mandiri.

## Lembar kerja 2.2 Membuat *Mind Map*

### Lembar Kerja 2.2

#### Membuat *Mind Map*

Buatlah *mind map* di kertas atau diaplikasi yang berisikan penjelasan salah satu dari topik berikut dengan menarik:

- Menjelaskan *Input-Proses-Output* dengan aktivitas interaktif (*unplugged* ataupun *plugged*).
- Menjelaskan Variabel dalam aplikasi koding menggunakan contoh sederhana.
- Menjelaskan Percabangan dan Perulangan dengan contoh permainan atau latihan sederhana.

#### Kriteria Penilaian *Mind Map* Guru:

Aspek	Topik: Masukan - proses - luaran	Topik: Variabel dalam Koding	Topik: Percabangan & Perulangan
<b>Keakuratan Konsep</b>	Menjelaskan konsep <i>IPO</i> dengan contoh nyata yang jelas dan aplikatif.	Menjelaskan apa itu variabel, cara menggunakannya, dan contoh penerapannya dalam kode sederhana.	Menjelaskan percabangan ( <i>if-else</i> ) dan perulangan ( <i>loop</i> ) dengan contoh yang relevan.
<b>Penerapan dalam Koding</b>	Menggunakan contoh interaktif, baik <i>unplugged</i> (aktivitas tanpa gawai) atau <i>plugged</i> (dengan gawai).	Menampilkan penggunaan variabel dalam aplikasi blok visual, atau alat koding lainnya.	Memberikan contoh langsung dalam aplikasi koding seperti Scratch, Blockly, Python.
<b>Kejelasan Demonstrasi</b>	Memberikan ilustrasi atau simulasi sederhana yang memudahkan siswa memahami proses <i>IPO</i> .	Memperlihatkan bagaimana variabel bekerja dalam kode melalui eksperimen kecil.	Menjelaskan bagaimana percabangan dan perulangan digunakan dalam koding dengan contoh logis.

## Kuis Singkat untuk Memahami Berpikir Komputasional Sebagai Dasar Koding.

### Lembar kerja 2.3

**Skor maksimal: 100**

**Bobot tiap soal: 10 poin**

Bacalah pertanyaan di bawah ini dengan seksama dan pilih 1 jawaban yang paling tepat!

1. Dari definisi berikut, manakah yang paling tepat menggambarkan berpikir komputasional?
  - A. Kemampuan menyelesaikan masalah dengan cepat
  - B. Kemampuan membuat program komputer secara langsung
  - C. Kemampuan menyelesaikan masalah dengan teknik berpikir seperti komputer
  - D. Kemampuan menghafal rumus dan algoritma
2. Berpikir komputasional dalam artian lain juga berarti....
  - A. Menghafal urutan kode pemrograman
  - B. Pemecahan masalah melalui pemodelan dan simulasi
  - C. Mengerjakan soal matematika secara manual
  - D. Mengandalkan komputer dalam setiap penyelesaian masalah
3. Apa saja 4C yang ada di dalam keterampilan abad 21?
  - A. *Communication, Coding, Creativity, Confidence*
  - B. *Collaboration, Communication, Calculation, Coding*
  - C. *Critical Thinking, Creativity, Communication, Collaboration*
  - D. *Critical Thinking, Creativity, Coding, Confidence*
4. Dari pernyataan di bawah ini, yang manakah yang paling tepat?
  - A. Berpikir komputasional hanya digunakan saat menggunakan komputer
  - B. Berpikir komputasional dapat digunakan dalam kehidupan sehari-hari untuk menyelesaikan masalah
  - C. Semua orang harus belajar koding agar bisa berpikir komputasional
  - D. Berpikir komputasional tidak penting untuk anak sekolah dasar
5. Apa saja konsep berpikir komputasional?
  - A. Decomposition, Pattern Recognition, Abstraction, Algorithmic Thinking
  - B. Algoritma, Logika, Analisis, Simulasi
  - C. Input, Proses, Output, Visualisasi
  - D. Urutan, Cabang, Perulangan, Debugging
6. Pernyataan di bawah ini merupakan bentuk berpikir komputasional dalam kehidupan sehari-hari, kecuali...
  - A. Mengatur jadwal belajar menggunakan aplikasi

- 
- B. Membuat langkah-langkah memasak mie instan
  - C. Menghafal soal tanpa memahami konsep
  - D. Menentukan rute tercepat menggunakan Google Maps
7. Cara kerja komputer melibatkan tiga tahapan utama. Manakah urutan yang benar?
- A. Proses - Input - Output
  - B. Output - Input - Proses
  - C. Input - Proses - Output
  - D. Input - Output - Proses
8. Dalam pembelajaran *unplugged*, peserta didik...
- A. Harus selalu menggunakan laptop atau tablet
  - B. Belajar konsep komputasional tanpa menggunakan perangkat digital
  - C. Menonton video pembelajaran di rumah
  - D. Membuat koding menggunakan robot canggih
9. Tujuan dari prinsip *abstraction* dalam berpikir komputasional adalah...
- A. Mengulang langkah secara terus-menerus
  - B. Mengambil informasi penting dan mengabaikan yang tidak relevan
  - C. Membagi masalah ke dalam langkah-langkah kecil
  - D. Mengenali pola dalam permainan
10. Contoh logika percabangan (*branching*) yang tepat dalam kehidupan nyata adalah...
- A. Menyapu lantai setiap hari
  - B. Membaca buku setiap pagi
  - C. Jika hujan, maka membawa payung; jika tidak hujan, maka tidak membawa payung
  - D. Memasak dengan mengikuti resep urut dari awal hingga akhir

## Daftar Pustaka

- Bebras Indonesia. (n.d.). *Apa itu Berpikir Komputasional?* Bebras Indonesia. <https://bebras.or.id/v3/apa-itu-berpikir-komputasional/>
- Christi, S. R., & Rajiman, W. (2023, Maret 15). Pentingnya Berpikir Komputasional dalam Pembelajaran Matematika. *Journal on Education*, 05(Mei-Agustus 2023), Kemendikbudristek. (2024). Kepka BSKAP Nomor 032 Tahun 2024 tentang capaian pembelajaran pada pendidikan anak usia dini, jenjang pendidikan dasar, dan jenjang pendidikan menengah pada Kurikulum Merdeka. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. 188.
- Meitjing, P. R., & Fuad, Y. (2023, Juli 8). EduMatSains; Jurnal Pendidikan Matematika dan Sains. *BERPIKIR KOMPUTASIONAL SISWA SMP DALAM MENYELESAIKAN MASALAH MATEMATIKA*.
- OpenAI. (2025). (*Model GPT-4*). [Large language model]. <https://openai.com>
- Riady, S., Setiawan, D., Ismah, Purwarianti, E. A., Jihad, A., Rachmahana, R., Audah, A., Izzatusholekha, Muhaemin, Siadari, T., Zakaria, A., Nugroho, S., Rahayu, B., Jumail, Irawan, D., Rosanti, N., Mutaqin, M., Agasta, M., Anggraena, Y., ... dan lainnya. (2025, Februari). Naskah Akademik Pembelajaran Koding dan Kecerdasan Artifisial Pada Pendidikan Dasar dan Menengah. *Kementerian Pendidikan Dasar dan Menengah Republik Indonesia*. [https://kurikulum.kemdikbud.go.id/file/1741766787\\_manage\\_file.pdf](https://kurikulum.kemdikbud.go.id/file/1741766787_manage_file.pdf)
- SCHOENFELD, A. H., & ARCAVI, A. (1988). National Council of Teachers of Mathematics. *On the Meaning of Variable*, 81.
- Supiarmo, M. G., Mardhiyatirrahmah, L., & Turmudi. (2021). Pemberian Scaffolding untuk Memperbaiki Proses Berpikir Komputasional Siswa dalam Memecahkan Masalah Matematika. *Jurnal Cendekia: Jurnal Pendidikan Matematika*, 05, 01, 378.
- Tresnawati, D., Setiawan, R., Fitriani, L., Mulyani, A., Rahayu, S., Nasrullah, M., Septiana, Y., Fatimah, D., Satria, E., Gunadhi, E., Latifah, A., & Sutedi, A. (n.d.). Jurnal PKM MIFTEK. *MEMBENTUK CARA BERPIKIR KOMPUTASI SISWA DI GARUT DENGAN TANTANGAN BEBRAS*, 1(2020), 1.