



**KEMENTERIAN
PENDIDIKAN DASAR
DAN MENENGAH
2025**

KODING DAN KECERDASAN ARTIFISIAL

**Ibnu Indarwati
Marwondo
Muhammad Faisal**

SMA/MA Kelas XI

Hak Cipta pada Kementerian Pendidikan Dasar dan Menengah Republik Indonesia.

Dilindungi Undang-Undang.

Penafian: Buku ini disiapkan oleh Pemerintah dalam rangka pemenuhan kebutuhan buku pendidikan yang bermutu, murah, dan merata sesuai dengan amanat dalam UU No. 3 Tahun 2017. Buku ini disusun dan ditelaah oleh berbagai pihak di bawah koordinasi Kementerian Pendidikan Dasar dan Menengah. Buku ini merupakan dokumen hidup yang senantiasa diperbaiki, diperbarui, dan dimutakhirkan sesuai dengan dinamika kebutuhan dan perubahan zaman. Masukan dari berbagai kalangan yang dialamatkan kepada penulis atau melalui alamat surel buku@kemendikdasmen.go.id diharapkan dapat meningkatkan kualitas buku ini.

Koding dan Kecerdasan Artifisial untuk SMA/MA Kelas XI

Penulis

Ibnu Indarwati
Marwondo
Muhammad Faisal

Penelaah

Heri Nurdianto
Yuli Wahyuni

Penyelia/Penyelaras

Supriyatno
Maharani Prananingrum
Irma Afriyanti
Eko Budiono
Sistya Devi Apriliana

Kontributor

David Yulian Hutapea
Romi Alfianto

Ilustrator

Nana Maulana

Editor

Nurhasanah Widianingsih

Editor Visual

Nadia Mahatmi

Desainer

Mohamad Lutvi

Penerbit

Kementerian Pendidikan Dasar dan Menengah

Dikeluarkan oleh

Pusat Perbukuan
Kompleks Kemendikdasmen Jalan RS. Fatmawati, Cipete, Jakarta Selatan
<https://buku.kemendikdasmen.go.id>

Cetakan Pertama, 2025

ISBN 978-634-00-2053-3 (no.jil.lengkap PDF)
ISBN 978-634-00-2055-7 (jil.2 PDF)

Isi buku ini menggunakan huruf Noto Serif 10/16 pt., SIL Open Font License.
xii, 300 hlm.: 17,6 × 25 cm.

Kata Pengantar

Pusat Perbukuan; Badan Standar, Kurikulum, dan Asesmen Pendidikan; Kementerian Pendidikan Dasar dan Menengah memiliki tugas dan fungsi mengembangkan buku pendidikan pada satuan Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah, termasuk Pendidikan Khusus. Sebagai sumber belajar utama dalam pembelajaran, buku dirancang untuk mendukung proses pembelajaran yang terarah, sistematis, dan relevan dengan kebutuhan zaman.

Dalam rangka mendukung terciptanya pembelajaran yang bermutu, pemerintah mengembangkan buku teks utama yang terdiri atas buku siswa dan panduan guru. Buku ini merupakan sumber belajar utama dalam pembelajaran bagi siswa dan menjadi salah satu referensi atau inspirasi bagi guru dalam merancang dan mengembangkan pembelajaran sesuai karakteristik, potensi, dan kebutuhan peserta didik. Keberadaan buku teks utama ini diharapkan dapat menumbuhkan generasi yang beriman dan bertakwa kepada Tuhan yang Maha Esa, dan berakhlak mulia; berpengetahuan luas; mandiri; kritis; kreatif; mampu bekerja sama; serta berdaya saing di tingkat nasional maupun global.

Kami mengucapkan terima kasih kepada semua pihak yang telah berkolaborasi dalam upaya menghadirkan buku teks utama ini. Semoga buku teks utama ini dapat menjadi tonggak perubahan yang menginspirasi, membimbing, dan mengangkat kualitas pendidikan kita ke puncak keunggulan.

Jakarta, Oktober 2025
Kepala Pusat Perbukuan,

Supriyatno, S.Pd., M.A.

Prakata

Alhamdulillah, segala puji dan syukur penulis panjatkan ke hadirat Allah Swt. karena atas rahmat dan karunia-Nya kami dapat menyelesaikan buku Koding dan Kecerdasan Artifisial untuk SMA/MA Kelas XI ini. Buku ini ditujukan bagi peserta didik Sekolah Menengah Atas/Madrasah Aliyah (SMA/MA) dengan tujuan untuk membekali mereka kompetensi yang relevan dengan kebutuhan zaman, khususnya dalam bidang koding dan penerapan Kecerdasan Artifisial (KA).

Penyusunan buku ini mengacu pada Capaian Pembelajaran (CP) Fase F untuk mata pelajaran Koding dan Kecerdasan Artifisial sebagaimana tertuang dalam Keputusan Kepala Badan Standar, Kurikulum, dan Asesmen Pendidikan Kementerian Pendidikan Dasar dan Menengah Nomor 046/H/KR/2025. Setiap bab pada buku ini mencerminkan elemen CP, TP, ATP, dan KKTP yang disusun secara sistematis, terarah, dan bertahap. Dengan demikian, peserta didik diharapkan mampu mengembangkan pengetahuan secara faktual, konseptual, prosedural, hingga metakognitif serta keterampilan abad ke-21, seperti berpikir kritis, berkolaborasi, dan kreatif.

Harapan besar kami adalah agar buku ini dapat menjadi panduan belajar yang kontekstual, inspiratif, dan aplikatif bagi peserta didik dalam mempelajari koding serta kecerdasan artifisial. Selain itu, buku ini juga dapat menjadi referensi yang bermanfaat bagi para pendidik dalam kegiatan pembelajaran di kelas.

Ucapan terima kasih kami sampaikan kepada Pusat Perbukuan (Pusbuk) Kemendikdasmen, para penelaah, editor, ilustrator, desainer, serta guru-guru SMA/MA di seluruh Indonesia yang telah memberikan gagasan dan praktik terbaik (*best practice*) sehingga buku ini dapat tersusun dengan baik. Terima kasih juga kami sampaikan kepada seluruh pihak yang telah berkontribusi, baik secara langsung maupun tidak langsung dalam proses penyusunan buku ini. Semoga buku ini dapat menjadi bekal berharga bagi peserta didik untuk menapaki dunia koding dan kecerdasan artifisial serta mempersiapkan mereka menjadi inovator muda yang siap menghadapi tantangan masa depan.

Oktober 2025

Tim Penulis

Daftar Isi

Kata Pengantar	iii
Prakata	iv
Daftar Isi	v
Daftar Tabel	vii
Daftar Gambar.	viii
Petunjuk Penggunaan	x
Bab 1 Kreasi Konten Digital dengan Kolaborasi Manusia dan KA	1
A. Kerja Kolaboratif Manusia dan KA	4
B. Teknik Penyuntingan Konten Digital.	11
C. Etika dan Hak Cipta dalam Produksi Konten Digital	15
D. Proyek Kolaborasi.	16
E. Diseminasi Konten	18
Bab 2 Dampak Kecerdasan Artifisial (KA) terhadap Ketenagakerjaan	27
A. Dampak Kecerdasan Artifisial	31
B. Tantangan dan Peluang Kecerdasan Artifisial (KA)	40
C. Prinsip <i>Human-Centered</i> Kecerdasan Artifisial (KA).	52
D. Etika Kecerdasan Artifisial.	59
Bab 3 Pemrograman Berorientasi Objek	73
A. Pengantar Singkat Pemrograman	77
B. Mengapa OOP?.	83
C. Kelas dan Objek	85
D. Menuliskan Kode Program.	94
E. Enkapsulasi.	98
F. Pewarisan dan Polimorfisme	102
G. Kelas Abstrak.	118
Bab 4 Data <i>Encoding</i> dan Integrasi Basis Data dalam Aplikasi	129
A. <i>Dataset</i> dan Pra-Pemrosesan Data	133
B. <i>Encoding</i> dan <i>Decoding</i>	142
C. Integrasi Aplikasi dan <i>Database</i>	168

Bab 5 Dasar-Dasar <i>Machine Learning</i> dan <i>Natural Language Processing</i>	193
A. Dasar-Dasar <i>Machine learning</i> (ML)	196
B. <i>Natural Language Processing</i> (NLP)	207
Bab 6 Penyelesaian Masalah dengan Rekayasa Model	219
A. Memecahkan Masalah	224
B. Proses Rekayasa (<i>Engineering Process</i>)	233
C. Mengembangkan Model Kecerdasan Artifisial	248
Glosarium	277
Daftar Pustaka	283
Daftar Sumber Gambar	287
Indeks	288
Profil Pelaku Perbukuan	292

Daftar Tabel

Tabel 1.1	Langkah-Langkah Proyek Kolaborasi	16
Tabel 1.2	Rubrik Penilaian Uji Kompetensi	22
Tabel 1.3	Tugas Pengayaan	26
Tabel 2.1	Contoh Kolaborasi Manusia dan Mesin	49
Tabel 3.1	Tipe Data di Python	79
Tabel 3.2	Contoh Atribut	88
Tabel 3.3	Contoh Pendefinisian Kelas	90
Tabel 3.4	Kelas Mobil dan Objek-Objeknya	92
Tabel 3.5	Perbandingan Metode yang Dimiliki Kelas	105
Tabel 3.6	Perbandingan <i>Overriding</i> dan <i>Overloading</i>	117
Tabel 4.1	Contoh Proses dari Teknik <i>Label Encoding</i>	143
Tabel 4.2	Tampilan Lima Baris Pertama pada <i>Dataset</i> dar Google Drive. .	147
Tabel 4.3	Contoh Transformasi Teks Menjadi Angka dalam <i>Label Encoding</i>	149
Tabel 4.4	Contoh Teknik <i>Binary Encoding</i>	158
Tabel 4.5	Contoh Hasil Teknik <i>Binary Encoding</i>	158
Tabel 4.6	Contoh <i>Dataset</i> pada Teknik <i>Target Encoding</i>	163
Tabel 4.7	Contoh Nilai Hasil <i>Encoding</i> pada Teknik <i>Target Encoding</i>	164
Tabel 6.1	Alat Bantu Praktik Rekayasa Perangkat Lunak	241
Tabel 6.2	Teknik Dasar dan Lanjutan dalam Koding	243
Tabel 6.3	<i>Tools</i> dan Bahasa dalam Pengembangan Perangkat Lunak	243
Tabel 6.4	Tahapan Pengembangan KA.	250
Tabel 6.5	Jenis <i>Library</i> Lain yang Umum Digunakan	257
Tabel 6.6	Metrik yang Digunakan pada Model Klasifikasi	263
Tabel 6.7	Metrik yang Digunakan pada Model Regresi	264
Tabel 6.8	Alat Bantu Evaluasi	265
Tabel 6.9	Kemampuan LLM	268
Tabel 6.10	Model LLM	269

Daftar Gambar

Gambar 1.1	Bekerja dalam Kelompok	5
Gambar 1.2	Kuadran Hubungan Manusia dan KA.	9
Gambar 2.1	Ilustrasi Perbandingan antara Otak Manusia dan Robot	31
Gambar 2.2	Dampak Positif KA di Dunia Kerja.	34
Gambar 2.3	Siap dengan Tantangan KA.	40
Gambar 2.4	<i>Framework Human-Centered KA</i>	52
Gambar 3.1	Struktur Dasar Algoritma.	78
Gambar 3.2	Ilustrasi Kelas dan Objek	86
Gambar 3.3	Kelas Mobil dan Objek-Objeknya	87
Gambar 3.4	Analogi Enkapsulasi	98
Gambar 3.5	Contoh Pewarisan Tunggal	104
Gambar 3.6	Contoh Pewarisan Majemuk	108
Gambar 3.7	Contoh <i>Override</i>	111
Gambar 4.1	Proses Mengubah Data oleh Sistem KA.	133
Gambar 4.2	Diagram Tipe Data	134
Gambar 4.3	Langkah Teknik <i>One Hot Encoding</i>	150
Gambar 4.4	Langkah <i>Binary Encoding</i>	158
Gambar 4.5	Integrasi Aplikasi dan <i>Database</i>	168
Gambar 4.6	Penjelasan Relasi Antartabel.	170
Gambar 4.7	Diagram Cara Kerja API.	174
Gambar 5.1	Skema Kerja <i>Machine Learning</i>	197
Gambar 5.2	Ilustrasi Algoritma <i>Decision Tree</i>	201
Gambar 5.3	Ilustrasi Algoritma KNN.	202
Gambar 5.4	Ilustrasi Algoritma Regresi Linier.	203
Gambar 5.5	Ilustrasi Algoritma <i>K-Means</i>	204
Gambar 5.6	Ilustrasi Algoritma <i>Association Rule Model</i>	204
Gambar 5.7	Ilustrasi Tokenisasi.	210
Gambar 5.8	Ilustrasi <i>Stemming</i> dan <i>Lemmatization</i>	211
Gambar 5.9	Ilustrasi <i>Part of Speech (POS) Tagging</i>	212
Gambar 5.10	Ilustrasi <i>Named Entity Recognition (NER)</i>	213
Gambar 5.11	Ilustrasi Analisis Sentimen	214
Gambar 6.1	Berbagai Masalah di Sekitar Kita	224
Gambar 6.2	Langkah-Langkah Identifikasi Masalah	225

Gambar 6.3	Pilar Berpikir Komputasional	229
Gambar 6.4	Proses Rekayasa (<i>Engineering Process</i>)	235
Gambar 6.5	Contoh <i>Use Case Diagram</i> Alarm Pintar Bangun Pagi	239
Gambar 6.6	Contoh <i>Flowchart</i> untuk Alarm Pintar Bangun Pagi	239
Gambar 6.7	Contoh ERD untuk Alarm Pintar Bangun Pagi	240
Gambar 6.8	Contoh Antarmuka Alarm Pintar Bangun Pagi	240
Gambar 6.9	Calon Pengguna Menguji Perangkat Lunak Sebelum digunakan	246
Gambar 6.10	Belajar Mengenali Apel dan Jeruk.	249
Gambar 6.11	Perbedaan Bunga Iris	259



Petunjuk Penggunaan

Dalam Buku siswa ini, kamu akan menemukan berbagai bagian yang dapat memandumu dalam belajar agar lebih terarah, menyenangkan, dan bermakna. Bacalah setiap bagian dengan saksama agar kamu memperoleh hasil pembelajaran yang optimal. Berikut penjelasan mengenai setiap bagian dalam buku ini.

Tujuan Pembelajaran

Bagian ini berisi tujuan atau kompetensi yang diharapkan dapat kamu capai setelah mempelajari materi pada setiap bab. Gunakan bagian ini untuk memahami arah pembelajaran yang akan kamu tempuh.

Kata Kunci

Kamu dapat menggunakan kata kunci untuk membantumu mengetahui pokok-pokok materi yang harus kamu pahami dan kuasai dalam setiap bab di buku ini.

Peta Materi

Bagian ini menampilkan peta konsep dari materi-materi yang akan dipelajari dalam satu bab. Bacalah bagian ini untuk mendapatkan gambaran menyeluruh sebelum kamu mulai mempelajari materi lebih mendalam.

Siap-Siap Belajar

Bagian ini berisi pertanyaan pemantik atau aktivitas singkat untuk membangkitkan minat dan konsentrasi belajarmu sebelum memulai pembelajaran. Gunakan bagian ini untuk menghubungkan pengalamanmu sebelumnya dengan materi baru yang akan dipelajari.



Cek Pemahaman Awal

Bagian ini berisi pertanyaan atau tugas ringan untuk mengukur pengetahuan awalmu terhadap materi yang akan dipelajari. Isilah dengan jujur agar kamu mengetahui seberapa siap kamu mempelajari topik tersebut.

Aktivitas Pembelajaran

Bagian ini berisi kegiatan belajar yang harus kamu lakukan, baik secara individu maupun kelompok. Laksanakan kegiatan ini dengan aktif agar kamu dapat mengalami langsung proses belajar.

Uji Kompetensi

Pada akhir pembelajaran, kamu dapat mengerjakan uji kompetensi berupa tes teori atau tes praktik. Kegiatan ini bertujuan untuk menguji pemahamanmu tentang materi yang sudah dipelajari dalam satu bab.



Pengayaan

Bagian ini memuat materi tambahan, proyek mini, atau tantangan lanjutan bagi kamu yang ingin memperluas dan memperdalam pengetahuan serta keterampilanmu di luar materi inti.

Refleksi

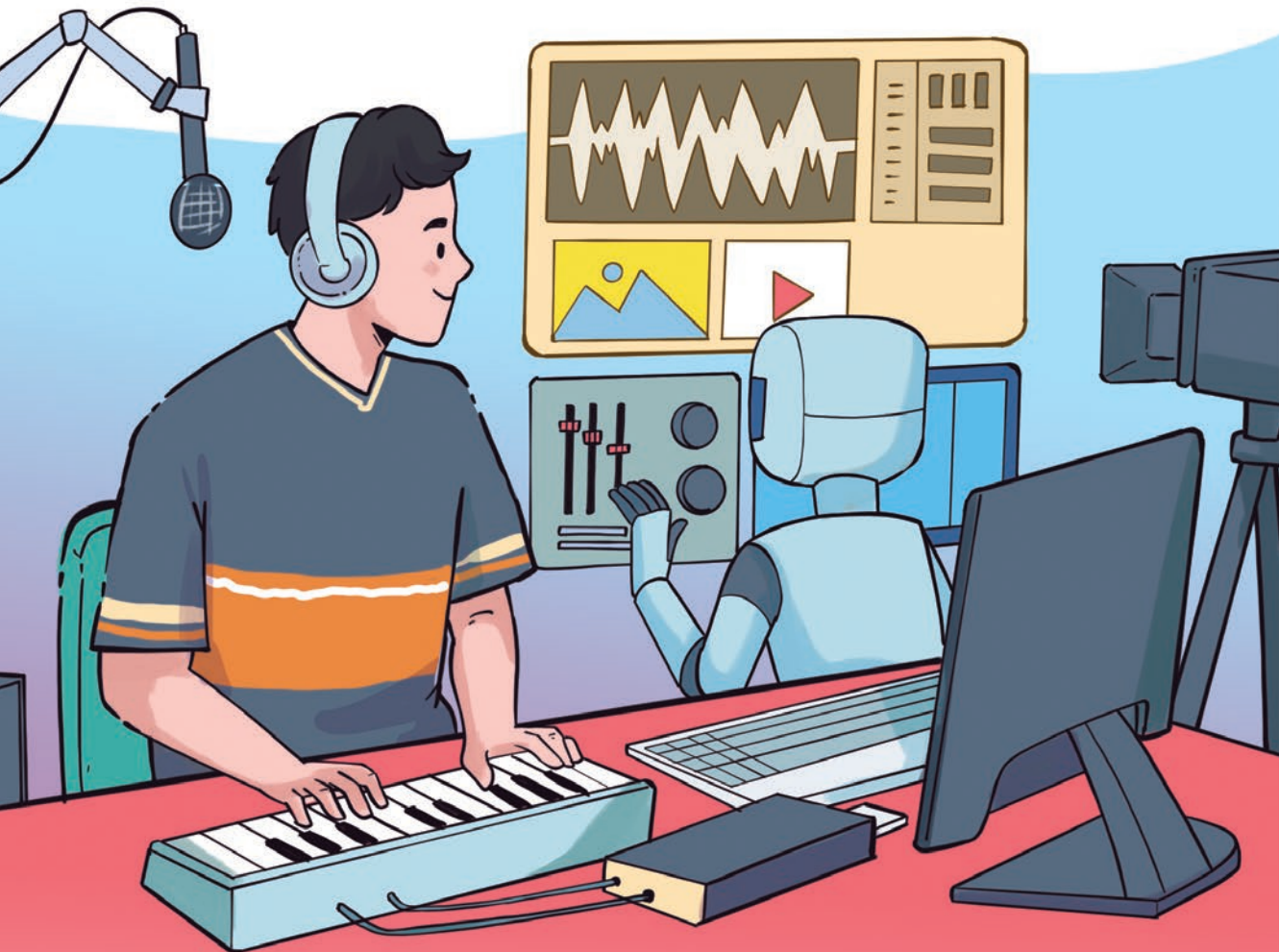
Kamu dapat melakukan refleksi dengan merenungkan pengalaman belajar yang telah dilakukan, menilai sejauh mana tujuan pembelajaran telah tercapai, serta mengidentifikasi kendala yang dihadapi. Gunakan refleksi untuk meningkatkan cara belajarmu ke depannya.



KEMENTERIAN PENDIDIKAN DASAR DAN MENENGAH
REPUBLIK INDONESIA, 2025
Koding dan Kecerdasan Artifisial
untuk SMA/MA Kelas XI
Penulis: Ibnu Indarwati, Marwondo, Muhammad Faisal
ISBN 978-634-00-2055-7 (jil.2 PDF)

Bab 1

Kreasi Konten Digital dengan Kolaborasi Manusia dan KA



Apa bagian tersulit ketika membuat konten multimedia?
Bayangkan jika Kecerdasan Artifisial (KA) membantumu membuat naskah,
desain, hingga mengedit video dalam hitungan menit!



Tujuan Pembelajaran

Pada akhir pembelajaran Bab 1, kamu diharapkan mampu memproduksi konten digital tingkat lanjut dengan bekerja sama secara efektif dalam tim (tidak hanya dengan teman, tetapi juga dengan KA); menggunakan berbagai teknik *editing*, baik untuk teks, gambar, audio, maupun video agar konten yang dihasilkan menjadi lebih menarik, jelas, dan berkualitas; mengalami secara langsung bagaimana Kecerdasan Artifisial (KA) bisa membantu manusia dalam proses *editing*; serta menyebarkan konten yang sudah dibuat.

Jangan hanya menjadi pengguna teknologi, tetapi jadilah pencipta konten yang cerdas dan kreatif serta siap menghadapi dunia digital masa kini.



Kata Kunci

kerja efektif, kolaborasi, kreativitas, konten digital, kreatif



Peta Materi





Siap-Siap Belajar

Pada pembelajaran di kelas X, kamu telah mempelajari tentang cara membuat dan menyebarkan konten multimedia. Pembelajaran tersebut merupakan bagian penting dari literasi digital. Sekarang, sudah saatnya untukmu meningkatkan pemahaman agar kamu dapat bekerja bersama dengan KA untuk membuat konten yang lebih kreatif dan efisien.



Cek Pemahaman Awal

Sebelum kamu mempelajari materi pada Bab 1 ini, jawablah terlebih dahulu pertanyaan-pertanyaan berikut secara jujur sesuai dengan pengetahuan dan pengalamanmu saat ini. Kamu tidak perlu mencari jawaban dari buku atau internet. Tidak ada penilaian benar atau salah. Latihan ini hanya berfungsi untuk membantumu dan guru dalam melihat titik awal sebelum memulai pembelajaran.

A. Pertanyaan Pilihan Ganda (Pengetahuan Dasar)

Pilihlah salah satu jawaban yang paling tepat!

1. Apa yang dimaksud dengan Kecerdasan Artifisial (KA)?
 - a. Robot yang bisa berbicara.
 - b. Sistem yang dapat meniru perilaku cerdas manusia dengan bantuan data dan algoritma.
 - c. Komputer yang memiliki kesadaran seperti manusia.
 - d. Perangkat canggih yang dapat mempercepat jaringan internet.
 - e. Aplikasi media sosial modern.
2. Dalam membuat video presentasi, KA dapat membantu dalam hal berikut, kecuali . . .
 - a. menyusun skrip otomatis
 - b. mengedit video secara instan
 - c. menganalisis preferensi audiens berdasarkan data
 - d. menggantikan pembicara sepenuhnya
 - e. memberikan saran visual dan musik yang sesuai
3. Mengapa kolaborasi manusia dan KA dianggap penting dalam produksi konten digital?
 - a. Karena KA lebih kreatif daripada manusia.

- b. Karena manusia tidak bisa bekerja sendiri.
- c. Karena KA dapat mempercepat dan menyederhanakan proses teknis, sedangkan manusia tetap sebagai pengambil keputusan kreatif.
- d. Karena kolaborasi ini membuat hasil akhir lebih efisien dan berkualitas tinggi.
- e. Karena semua konten kini wajib dibuat oleh KA.

B. Pertanyaan Reflektif (Keterkaitan dengan Pengalaman Sebelumnya)

Tulislah jawaban dalam bentuk 2–3 kalimat.

1. Ceritakan pengalamanmu saat membuat konten multimedia (video, *podcast*, infografik, dan sebagainya)! Apa bagian tersulit yang kamu hadapi?
2. Jika kamu diberi alat berbasis KA untuk membantumu membuat sebuah konten, pada bagian mana kamu ingin dibantu? Jelaskan alasannya!

A. Kerja Kolaboratif Manusia dan KA

Saat ini, membuat konten digital bukan lagi hanya mengandalkan tenaga manusia. Kehadiran Kecerdasan Artifisial (KA) membuka peluang kerja sama yang membuat proses menjadi lebih cepat, kreatif, dan menjangkau lebih banyak orang. Tentu, kolaborasi ini bisa berhasil karena manusia dan KA membawa keunggulannya masing-masing.

Manusia menghadirkan kreativitas, empati, nilai budaya, dan kemampuan memahami situasi. Sementara itu, KA unggul dalam kecepatan, ketepatan teknis, serta kekuatan mengolah data dalam jumlah besar. Jika keduanya dipadukan, hasil yang lahir bukan sekadar efisien, tetapi juga penuh makna.

Tentunya, kamu pasti pernah bekerja sama dalam sebuah kelompok. Biasanya, setiap anggota kelompok mempunyai perannya masing-masing. Ada yang menjadi pencatat, ada yang mencari data, ada juga yang menjadi presenter. Jika semua peran ini dijalankan dengan baik dan saling melengkapi, maka hasil kerja kelompok pasti akan lebih maksimal apabila dibandingkan dengan hasil kerja yang dilakukan oleh sendiri. Nah, inilah yang disebut dengan kolaborasi—kerja sama antara beberapa pihak dengan tujuan yang sama, meskipun perannya berbeda-beda.



Gambar 1.1 Bekerja dalam Kelompok

Menariknya, di zaman sekarang, kolaborasi itu tidak hanya dapat dilakukan dengan teman-temanmu saja. Dengan adanya kemajuan teknologi Kecerdasan Artifisial (KA), manusia kini dapat bekerja sama dengan mesin pintar untuk menyelesaikan berbagai tugas, menganalisis data, hingga memecahkan masalah.

Saat ini, KA berkembang sangat pesat dan mulai digunakan pada berbagai bidang. Menurut laporan McKinsey (2025), KA, terutama jenis *Generative AI (GenAI)*, tidak lagi sekadar alat bantu, tetapi dapat menjadi kekuatan besar yang mengubah cara kerja organisasi, model bisnis, bahkan struktur perusahaan. Organisasi yang berhasil biasanya bukan hanya “menempelkan” teknologi ini saja, tetapi benar-benar membangun ulang proses kerja mereka agar lebih efisien dan cerdas.

Namun, jangan khawatir, kolaborasi manusia dan KA bukan tentang siapa yang lebih hebat. Akan tetapi, bagaimana manusia dan KA dapat saling melengkapi. KA memiliki keunggulan dalam hal kecepatan dan kemampuan mengolah data dalam jumlah besar. Sedangkan, manusia tetap unggul dalam kreativitas, intuisi, dan pemahaman konteks. Sebagaimana yang dikemukakan oleh Topol (2019) bahwa KA bukanlah pengganti manusia, tetapi justru penguat dari kecerdasan manusia itu sendiri. Berikut contoh bentuk fungsi KA sebagai penguat dari kecerdasan manusia.

1. KA dapat membantu dokter membaca hasil MRI dengan cepat, tetapi keputusan akhir tetap berada di tangan dokter yang lebih memahami kondisi pasien secara menyeluruh.
2. KA dapat menganalisis data cuaca untuk evakuasi bencana, tetapi manusia yang menentukan langkah terbaik dengan mempertimbangkan faktor sosial dan budaya.

Lalu, untuk apa saja KA dapat digunakan? Manusia dapat memanfaatkan KA untuk hal-hal berikut:

1. Mengotomatisasi tugas-tugas rutin, seperti entri data atau pengecekan stok barang;
2. Menganalisis dan menampilkan data dalam bentuk visual yang mudah dipahami;
3. Memprediksi masalah sebelum terjadi;
4. Memberikan saran atau rekomendasi berdasarkan pola data; dan
5. Bertindak sebagai agen cerdas yang dapat mengambil tindakan mandiri (dalam batas tertentu).

Dengan demikian, KA bukan hanya sekadar teknologi canggih, tetapi juga dapat menjadi alat bantu yang membuat pekerjaan menjadi lebih efisien, cepat, dan cerdas.

Pernahkah kamu mencoba membuat sebuah desain, tulisan, atau video menggunakan teknologi? Sekarang, KA dapat kamu jadikan sebagai partner yang keren dalam membuat konten digital tersebut. Berikut contoh pekerjaan yang dapat dilakukan secara kolaboratif dengan KA.

1. Menulis artikel
2. Mendesain poster
3. Membuat video pendek

Akan tetapi, ingatlah bahwa kreativitas manusia tetap yang utama. KA dapat membantumu dalam membuat draf tulisan, tetapi kamu yang harus menyempurnakan kaidah kebahasaannya agar mudah dibaca dan dipahami.

Pada bidang jurnalistik, KA dapat digunakan untuk menyusun berita dengan cepat berdasarkan data. Namun, wartawanlah yang menentukan berita mana yang penting dan layak untuk ditayangkan. Senada dengan yang diungkapkan oleh LIPI (2021) bahwa KA dapat membuat pekerjaan media menjadi lebih efisien, tetapi manusia tetap menjadi bagian terpenting dalam memastikan kebenaran setiap informasi. Kominfo (2022) juga menegaskan bahwa kini semakin banyak

orang dapat membuat konten dengan bantuan KA, tetapi mereka tetap harus menggunakannya secara bijak dan bertanggung jawab.



Aktivitas 1.1

Menganalisis Kolaborasi Manusia dan KA di Dunia Nyata (Aktivitas Kelompok)

Simaklah beberapa video pendek tentang penggunaan KA dalam bidang kesehatan, pendidikan, dan lapangan kerja berikut. Aktifkan *subtitle* jika diperlukan untuk membantu memahami isi video.

No.	Pranala	Deskripsi
1	 https://buku.kemdikbud.go.id/s/KKA11-Video1a	Video yang berdurasi sekitar 39 menit ini membahas tentang bagaimana KA berkembang di sektor kesehatan (fitur mendeteksi pola medis, mendukung dokter, serta tantangan etis dan regulasi).
2	 https://buku.kemdikbud.go.id/s/KKA11-Video1b	Video singkat yang berdurasi sekitar satu menit ini memperlihatkan cara KA mengubah proses belajar-mengajar (personalisasi materi, adaptasi gaya belajar, dan peran guru dalam era digital).
3	 https://buku.kemdikbud.go.id/s/KKA11-Video1c	Video singkat berdurasi sekitar 1,5 menit ini menyoroti bagaimana KA akan mengubah lanskap kerja di Indonesia (tantangan dan peluang bagi pelajar).

Setelah menyimak video-video tersebut, ikutilah petunjuk berikut.

- a. Bentuklah kelompok yang terdiri atas 3–5 orang!
- b. Simaklah kembali video-video tersebut bersama kelompok, lalu diskusikan mengenai hal-hal berikut.
 - 1) Apa kelebihan dan kekurangan KA dalam konteks tersebut?
 - 2) Peran apa yang tetap memerlukan manusia?
- c. Presentasikan hasil diskusi tersebut (bagi peran untuk masing-masing anggota)!

Dalam buku *Human Compatible* (2019), Stuart Russell menyarankan agar KA digunakan dengan prinsip berikut.

1. Selalu mengikuti dan menghormati tujuan manusia.
2. Tidak diberi otonomi berlebihan tanpa memperhatikan nilai-nilai manusia.
3. Terbuka terhadap masukan dan terus belajar dari manusia.

Apabila prinsip tersebut diabaikan, maka akan menimbulkan risiko bagi manusia. Adapun risiko yang akan ditimbulkan, di antaranya:

1. KA salah memahami keinginan manusia.
2. Ketergantungan yang berlebihan membuat kita jadi kurang kreatif dan kurang mandiri.

Akan tetapi, apabila KA digunakan dengan benar, tentu saja potensi positifnya sangat besar bagi kita. Berikut potensi positif yang dapat kita peroleh dari penggunaan KA yang tepat.

1. Memperkuat kemampuan manusia menghadapi tantangan global.
2. Memperluas akses layanan pendidikan, kesehatan, dan publik, terutama di daerah yang sulit dijangkau.

McKinsey (2025) memperkenalkan dua model kolaborasi antara manusia dan KA, yaitu:

1. *Model Factory*

Pada model ini, KA digunakan untuk menangani tugas-tugas berulang dan massal, seperti:

- a. mengisi data otomatis;

- b. memantau sistem; dan
- c. memberikan rekomendasi cepat.

Model ini sangat cocok digunakan untuk pekerjaan teknis dan skala besar.

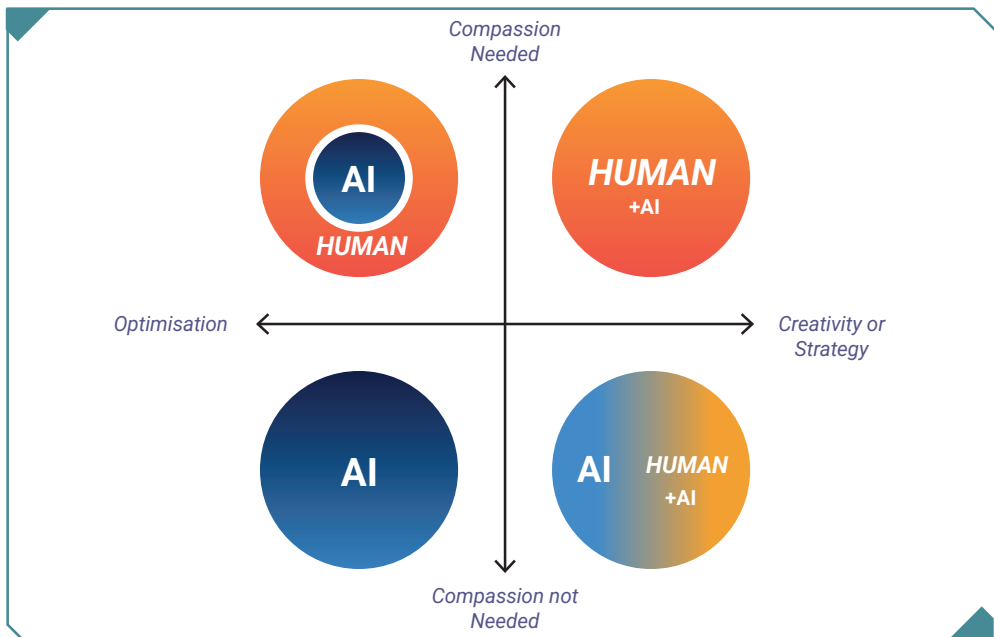
2. Model Artisan

Pada model ini, KA menjadi asisten bagi para profesional, seperti dokter, guru, atau desainer. Fungsi model artisan adalah untuk mendukung pengambilan keputusan dengan data dan wawasan, tetapi keputusan akhir tetap di tangan manusia.

Kombinasi model *factory* dan model artisan dapat menciptakan kerja tim yang hebat. Mengapa demikian? Hal tersebut terjadi karena:

1. KA unggul dalam hal skala dan kecepatan; dan
2. Manusia unggul dalam empati, etika, serta kreativitas.

Dr. Kai-Fu Lee, *Chairman* dan *CEO* Sinovation Ventures yang juga pernah bekerja di Apple, Microsoft, Google, dan Silicon Graphics, dalam paparannya pada tahun 2018, beliau memetakan empat cara manusia bekerja dengan KA (dalam Blake, 2019). Peta tersebut dibuat dalam kuadran seperti yang dapat kamu lihat pada gambar berikut.



Gambar 1.2 Kuadran Hubungan Manusia dan KA
 Sumber: Kai-Fu Lee (2018)

Dr. Kai-Fu Lee memetakan bagaimana manusia dan KA akan bekerja sama. Hal itu tergantung pada dua hal utama, yaitu:

1. Apakah dibutuhkan empati/kompasi?
2. Apakah tugas tersebut lebih ke arah optimasi atau kreativitas/strategi?

Agar kamu lebih memahami isi gambar di atas, mari kita bahas keempat kuadran tersebut.

1. Kuadran 1: *HUMAN + AI (Empati + Optimasi)*

Kuadran ini menunjukkan adanya kolaborasi empati dan optimasi. Contoh pekerjaan yang berkaitan dengan hal ini, yaitu guru, perawat, konselor, dan desainer konten pribadi. Pekerjaan tersebut membutuhkan ide dan perasaan. Apabila kamu amati, peran seorang guru tidak dapat digantikan oleh KA. Mengapa? Jawabannya adalah karena guru bukan hanya berperan untuk menyampaikan materi, tetapi juga memberi semangat, memahami kondisi peserta didik, serta mengerti kapan peserta didik perlu dibantu. Dalam hal ini, KA berperan untuk membantu guru dalam mencari materi atau membuat soal secara otomatis. Dengan demikian, guru dan KA dapat bekerja sama.

2. Kuadran 2: *HUMAN + AI (Empati + Kreativitas)*

Contoh untuk pekerjaan pada kuadran ini adalah *customer service*, petugas kesehatan dasar, dan *human resource*. Pekerjaan ini membutuhkan sentuhan manusia, tetapi dapat dibantu dengan KA. Apabila kamu pernah berkomunikasi (*chatting*) dengan *customer service*, lalu ada yang menjawab secara otomatis, maka itu adalah KA. Namun, apabila masalahnya lebih rumit dan membutuhkan pengertian, maka manusia akan mengambil alih. Dengan demikian, dalam hal ini KA dapat digunakan untuk membantu di awal proses, sedangkan manusia yang akan menyelesaikannya.

3. Kuadran 3: *AI + HUMAN (Tanpa Empati + Kreativitas)*

Contoh pekerjaan pada kuadran ini adalah *digital marketing*, desain produk, dan UI/UX. Pekerjaan-pekerjaan tersebut membutuhkan ide, tetapi tidak memerlukan perasaan. Misalnya, kamu bekerja sebagai desainer, maka KA dapat membantumu untuk menemukan tren warna, membuat sketsa awal, atau memberi rekomendasi desain. Meski demikian, kamu sebagai manusia, tetaplah yang mengambil keputusan dan membuat desain tersebut menjadi 'memiliki rasa'.

4. Kuadran 4: AI (Tanpa Empati + Optimasi)

Contoh pekerjaan pada kuadran 4 adalah analisis data besar, sistem rekomendasi, dan logistik otomatis. Dalam hal ini, KA dapat bekerja sendiri dengan sangat efisien karena tugasnya adalah mengoptimalkan proses dan tidak memerlukan sentuhan manusia. Pekerjaan ini dapat dikerjakan lebih cepat, akurat, dan tanpa lelah oleh KA. Misalnya, ketika sistem KA mengatur rute pengiriman barang supaya lebih hemat bahan bakar dan waktu. Pekerjaan tersebut tidak membutuhkan perasaan, yang penting efisien.

Peran KA bukan untuk menggantikan manusia, tetapi untuk bekerja bersama manusia. Sebagaimana yang dikatakan oleh Risman (2020) bahwa KA membantu kita menjadi lebih kreatif. Manusia dan KA adalah tim yang saling melengkapi. KA menangani hitungan dan kecepatan, sedangkan manusia memberi ide, rasa, dan nilai kemanusiaan.

Bayangkan apabila kamu dapat mengajak KA untuk membantu tugas sekolah, membuat karya kreatif, atau membangun solusi untuk masalah nyata di masyarakat. Masa depan kolaborasi ini harus dimulai dari sekarang dan dari diri kamu sendiri.

B. Teknik Penyuntingan Konten Digital

Berdasarkan penjelasan tersebut, kini kamu sudah memahami bahwa KA dapat membuat pekerjaan menjadi lebih efisien, cepat, dan cerdas. Selanjutnya, kamu akan belajar mengenai bagaimana cara menerapkan KA dalam proses pembuatan konten digital.

1. Penyuntingan Teks

Pernahkah kamu merasa bahwa menulis itu sulit? Kadang kamu bingung dalam memilih kata dan menentukan susunan kalimat. Nah, di sinilah pentingnya peran penyuntingan teks. Penyuntingan (*editing*) adalah proses memperbaiki tulisan agar lebih jelas, rapi, dan mudah dibaca. Proses penyuntingan ini dimulai dari memeriksa ejaan, tanda baca, struktur kalimat, hingga gaya bahasa.

Kabar baiknya adalah sekarang kamu tidak perlu melakukan proses penyuntingan sendirian. Kini terdapat banyak alat berbasis Kecerdasan

Artifisial (KA) yang dapat membantumu ketika menyunting teks. Aplikasi tersebut, di antaranya:

- Grammarly atau QuillBot yang dapat memperbaiki tata bahasa dan menyarankan kata yang lebih tepat.
- ChatGPT yang dapat membantumu membuat ringkasan, menulis ulang kalimat, atau bahkan menyusun paragraf awal.
- Fitur editor otomatis di aplikasi penulisan, seperti Google Docs atau Microsoft Word yang dapat langsung mendeteksi kesalahan.
- Chat.Qwen.ai, yaitu asisten penulisan dan media yang dapat menghasilkan teks, merangkum, menulis ulang, bahkan membantu ide visual.
- DeepSeek.com dan Kimi.com yang menyediakan kemampuan ringkasan, penulisan ulang teks panjang, interpretasi konten, serta *brainstorming* (fungsi yang mirip digabung karena keduanya fokus pada konteks dan penalaran teks).

Tapi ingat ya, meskipun KA dapat membantumu dalam banyak hal, tetap saja kamu perlu membaca dan menyesuaikan hasil akhirnya. Soalnya, hanya kamu yang benar-benar mengetahui maksud dan gaya tulisan yang kamu inginkan. Jadi, tetaplah libatkan peran manusia dalam proses penyuntingan.



Aktivitas 1.2

Menyunting Teks dengan Bantuan KA (Aktivitas Individu dan Kelompok)

Pada Aktivitas 1.2 ini kamu akan belajar mempraktikkan cara menyunting tulisan menggunakan KA. Kerjakan tugas ini dengan mengikuti langkah-langkah berikut.

- Pilihlah salah satu paragraf dari tugas menulismu sebelumnya (artikel, narasi, atau deskripsi).
- Gunakan alat berbasis KA (Grammarly, ChatGPT, atau fitur edit di Google Docs) untuk menyunting paragraf tersebut.
- Bandingkan versi asli dan versi yang telah disunting. Apa perbedaannya?
- Diskusikan bersama teman sekelompokmu, apakah perubahan tersebut membuat tulisanmu menjadi lebih baik atau justru membuat tulisan tersebut kehilangan gaya pribadi!

2. Penyuntingan Gambar dan Video

Konten visual seperti gambar dan video sangat penting dalam dunia digital. Akan tetapi, untuk membuat gambar yang keren atau video yang menarik, tentu membutuhkan proses penyuntingan yang tidak sebentar. Mulai dari mengatur warna, memotong-motong adegan, memberi efek, hingga menambahkan musik atau teks. Dengan bantuan KA, sekarang semua proses itu dapat dilakukan dengan lebih cepat dan mudah. Berikut aplikasi yang dapat digunakan untuk membantu proses tersebut.

- Canva memiliki fitur *'Magic Edit'* yang dapat menghapus atau menambahkan objek secara otomatis.
- Adobe Firefly dapat menghasilkan gambar dari perintah teks (*text-to-image*).
- Kapwing, Runway, atau Magisto dapat membantumu mengedit video otomatis, memilih transisi, bahkan menyinkronkan suara dan gambar.

Selain itu, KA juga dapat membantumu mengubah gaya visual, seperti membuat gambar menjadi lebih dramatis atau menyesuaikan *tone* video agar cocok dengan *mood* yang kamu inginkan. Namun, lagi-lagi, sentuhan terakhir tetap ada di tanganmu untuk memastikan bahwa konten sesuai dengan pesan yang ingin disampaikan.



Aktivitas 1.3

Menyunting Gambar dan Video Digital dengan KA (Aktivitas Individu)

Kerjakan Aktivitas 1.3 ini dengan mengikuti petunjuk berikut.

- Ambillah satu foto atau video pendek (maksimal berdurasi satu menit) dari dokumentasi kegiatanmu di sekolah!
- Gunakan alat berbasis KA (Canva atau Kapwing) untuk menyunting konten tersebut! Proses penyuntingan dapat berupa penambahan teks, efek visual, atau penghilangan objek!
- Unggahlah hasil kerjamu ke platform yang telah kamu sepakati bersama gurumu!
- Presentasikan olehmu bagaimana KA membantu proses *editing* tersebut! Bagian mana dari proses tersebut yang masih perlu disempurnakan secara manual?

3. Penyuntingan Audio

Kamu pasti pernah mendengar istilah *podcast*, *voice over*, atau rekaman narasi untuk video. Nah, tentu kamu juga mengetahui betapa pentingnya suara yang jernih dan enak didengar untuk mendukung kegiatan tersebut. Dalam hal ini, peran penyuntingan audio sangat diperlukan. Penyuntingan audio tersebut meliputi penghapusan *noise*, penyesuaian volume, efek suara, hingga pemotongan bagian yang tidak perlu. Dalam hal ini, KA juga dapat membantu proses tersebut. Adapun aplikasi yang dapat digunakan, di antaranya:

- a. Adobe Podcast (Project Shasta) dapat membuat suara terdengar seperti direkam di studio meskipun sebetulnya hanya menggunakan mikrofon biasa.
- b. Auphonic dapat menyesuaikan level suara secara otomatis.
- c. Descript memungkinkan kamu mengedit audio seperti mengedit teks.

Sangat unik, bukan? Tidak hanya itu, ada juga alat yang dapat menghilangkan suara latar belakang secara otomatis, memperbaiki intonasi, bahkan menyintesis suara baru. Tetapi ingat, tetaplah kamu yang harus memastikan bahwa hasil akhirnya nyaman untuk didengar dan sesuai dengan konteks.



Aktivitas 1.4

Penyuntingan Audio Menggunakan Aplikasi Berbasis KA (Aktivitas Individu)

Pada Aktivitas 1.4 ini kamu akan mempraktikkan cara menyunting audio menggunakan aplikasi berbasis KA. Untuk mengerjakan aktivitas ini, ikuti langkah-langkah berikut.

- a. Rekamlah suaramu ketika membacakan sebuah paragraf pendek (yang dapat diambil dari teks puisi, teks narasi, ataupun teks deskripsi suatu produk) dengan durasi sekitar 30–60 detik!
- b. Gunakan aplikasi berbasis KA (Adobe Podcast, Descript, atau Auphonic) untuk menyunting audio tersebut!
- c. Bandingkan audio asli dan hasil suntingan yang telah dibantu oleh KA.

d. Lakukan refleksi singkat dengan menjawab pertanyaan berikut.

- 1) Apakah kualitas suara meningkat?
- 2) Apakah prosesnya mudah dilakukan?

C. Etika dan Hak Cipta dalam Produksi Konten Digital

Pemahaman mengenai etika dan hak cipta sangatlah penting untuk kamu ketahui agar kamu menyadari batasan hukum dan moral dalam membuat serta menyebarkan konten digital. Berikut beberapa hal yang harus kamu perhatikan dalam masalah ini.

1. Pentingnya Menyebutkan Sumber Karya Orang Lain

Hal ini berarti, kamu harus selalu memberikan kredit yang tepat pada karya yang digunakan, seperti teks, gambar, video, atau musik.

2. Larangan Plagiarisme

Penggunaan musik, video, atau gambar tanpa izin yang sah dari pemilik hak cipta adalah pelanggaran.

3. *Creative Commons* dan Lisensi Terbuka

Kamu juga harus memahami lisensi yang memperbolehkan penggunaan karya dengan hak tertentu selama mengikuti ketentuan yang berlaku.

4. Etika Representasi

Maksud dari etika representasi adalah tidak menyebarkan konten hoaks, diskriminatif, atau yang melanggar privasi individu.

Keempat hal tersebut perlu kamu perhatikan dan patuhi ketika kamu membuat atau hendak menyebarkan sebuah konten digital. Dengan demikian, konten yang kamu buat tersebut tidak akan mendapatkan masalah di kemudian hari.

D. Proyek Kolaborasi

Melalui proyek ini, kamu akan mengasah keterampilan kolaboratif dan kreatif dalam membuat konten digital berbasis isu nyata yang dihadapi remaja saat ini. Proyek ini juga akan mengajakmu berlatih menggunakan Kecerdasan Artifisial (KA) sebagai rekan kerja digital melalui teknik rekayasa *prompt* serta memahami batasan dan potensi KA dalam konteks produksi konten.

Kamu akan bekerja dalam kelompok kecil (3–5 orang) untuk merancang dan membuat konten kampanye digital yang mengangkat isu sosial atau digital yang dekat dengan kehidupan remaja, seperti:

1. Etika Digital
2. Perundungan Siber
3. Literasi Informasi dan Hoaks
4. Kesehatan Mental di Era Media Sosial
5. Jejak Digital dan Privasi

Konten kampanye yang dibuat tersebut dapat berbentuk:

1. Video pendek (durasi sekitar 1–2 menit);
2. Infografik atau poster digital interaktif;
3. Audio/*podcast* singkat; atau
4. Kombinasi antara dua format di atas.

Konten harus diproduksi secara kolaboratif, memanfaatkan peran manusia dan KA, serta mencerminkan etika dan nilai kemanusiaan. Berikut penjelasan mengenai langkah-langkah pengerjaan proyek kolaborasi.

Tabel 1.1 Langkah-Langkah Proyek Kolaborasi

No	Langkah	Kegiatan Utama	Penjelasan dan Aktivitas Pembelajaran
1	Observasi dan Ideasi	Lihat dan Catat	<ul style="list-style-type: none">• Lakukan survei atau observasi di sekolah/lingkungan sekitar.• Identifikasi isu sosial atau digital yang layak diangkat.• Gunakan <i>tools</i>, seperti Google Form atau wawancara teman sebagai metode.

No	Langkah	Kegiatan Utama	Penjelasan dan Aktivitas Pembelajaran
2	Diskusi dan Perencanaan	Diskusi	<ul style="list-style-type: none"> Diskusikan temuan bersama kelompok. Tentukan tema kampanye dan jenis konten yang akan dibuat. Gunakan <i>thinking routine</i>, seperti "<i>See-Think-Wonder</i>" untuk eksplorasi ide.
3	Pembagian Peran dan Prompt	Tentukan Peran dan Rancang <i>Prompt</i>	<ul style="list-style-type: none"> Tentukan peran tiap anggota, yaitu penulis naskah, penyunting visual/audio, operator KA, dan evaluator. Buatlah <i>prompt</i> untuk KA sesuai dengan kebutuhan, di antaranya: ChatGPT, Chat.Qwen.ai, DeepSeek.com, Kimi.com (naskah)–DALL·E/Adobe Firefly (visual) –Descript/Adobe Podcast (audio)–Kapwing/Canva (<i>editing</i>).
4	Produksi Konten	Mengerjakan Bersama	<ul style="list-style-type: none"> Buatlah konten dengan membagi peran antara manusia dan KA. Lakukan iterasi (penyempurnaan berdasarkan masukan). Evaluasi: <ul style="list-style-type: none"> Bagian mana yang memerlukan intervensi manusia? Apa yang dapat diserahkan kepada KA?
5	Publikasi dan Presentasi	Menunjukkan Karya	<ul style="list-style-type: none"> Presentasikan hasil proyek kalian kepada teman sekelas. Ceritakan proses, pembagian kerja, tantangan, dan refleksi atas kolaborasi manusia dan KA.



Aktivitas 1.5

Proyek Kolaborasi Konten Kampanye Digital dengan KA (Aktivitas Individu)

Berikut adalah petunjuk untuk pengerjaan Aktivitas 1.5.

- a. Pilihlah salah satu isu yang akan dibuat kampanye!
- b. Buatlah *prompt* untuk menghasilkan poster bertema sesuai dengan yang dipilih!
- c. Gunakan aplikasi seperti DALL·E atau Firefly. Cobalah beberapa variasi *prompt*, lalu evaluasi hasilnya!
- d. Ambillah sebuah skrip/naskah atau gambar/video mentah (asli/belum disunting). Kemudian, mintalah bantuan KA untuk menyunting naskah atau gambar/video tersebut!
- e. Bandingkan hasilnya, lalu kemukakan bagaimana peran manusia dalam menyempurnakan hasil KA!
- f. Presentasikan hasil kolaborasimu. Uraikan juga pembagian peran masing-masing antara kamu dan KA!

E. Diseminasi Konten

Setelah sebuah konten dibuat, pekerjaanmu belum selesai. Konten digital akan bermanfaat, apabila dapat menjangkau banyak orang. Hal inilah yang disebut dengan diseminasi, yaitu proses menyebarkan konten ke audiens melalui berbagai platform digital.

Coba bayangkan, kamu dan kelompokmu baru saja membuat karya digital, entah itu poster hemat energi ataupun video kampanye anti-*bullying*. Posternya sudah keren: terdapat ilustrasi lampu hemat listrik yang dibuat dengan bantuan KA, tercantum pesan singkat “Matikan listrik, jika tidak digunakan!”, serta dilengkapi dengan data sederhana tentang penghematan listrik di rumah. Begitupun dengan video yang berisi tentang ajakan saling menghargai teman, sudah dilengkapi dengan adegan dramatis dan pesan positif, seperti “*Stop bullying*, mulai dari kita!”.

Nah, pertanyaannya sekarang adalah bagaimana caranya agar karya kamu tersebut tidak hanya disimpan di laptop saja, tetapi dapat dilihat dan bermanfaat bagi teman-temanmu bahkan bagi masyarakat luas? Di sinilah pentingnya peran kolaborasi manusia dan KA berlanjut.

1. KA dapat menyarankan waktu terbaik untuk mengunggah konten ke media sosial, misalnya sore hari ketika banyak orang sedang daring. KA juga bisa membuat keterangan (*caption*) otomatis yang singkat, seperti “Ayo hemat energi, mulai dari rumah kita sendiri!” atau keterangan untuk video anti-*bullying*, seperti “*Stop bullying*, mulai dari kita!”. Setelah konten dipublikasikan, KA dapat membantu menghitung berapa banyak orang yang melihat, menyukai, membagikan, atau berkomentar pada poster maupun video tersebut.
2. Manusia, yaitu kamu dan kelompokmu, tetap berperan penting. Kalianlah yang memutuskan platform mana yang paling tepat, apakah Instagram sekolah, *website* OSIS, atau grup komunitas. Kalian juga dapat menyunting keterangan (*caption*) agar sesuai dengan gaya bahasa audiens. Misalnya, untuk poster hemat energi dapat ditambahkan dengan kalimat ajakan, seperti “Yuk, kita jadi generasi peduli lingkungan!”. Namun, untuk video anti-*bullying*, dapat dibuat lebih emosional, seperti “Kita semua berhak merasa aman di sekolah. Jangan ada *bullying* lagi!”. Selain itu, kalian juga yang harus mengevaluasi: apakah pesan yang disampaikan tersebut benar-benar mendidik? Apakah ada risiko menyinggung pihak tertentu?

Dengan cara tersebut, sebuah poster sederhana bukan hanya sekadar gambar dan sebuah video singkat bukan hanya sekadar hiburan. Keduanya dapat menjadi media edukasi yang mampu menginspirasi banyak orang tentang pentingnya hemat energi sekaligus menciptakan lingkungan sekolah yang terbebas dari *bullying*.

Diseminasi konten digital seperti ini adalah bagian dari literasi digital tingkat lanjut. Kamu tidak hanya bisa membuat karya, tetapi juga akan mengetahui cara menyebarkannya secara cerdas, strategis, dan bertanggung jawab, sehingga karya yang dibuat benar-benar memberi dampak nyata bagi masyarakat.



Aktivitas 1.6

Penyebarluasan Konten (Tugas Individu)

Setelah kamu menghasilkan konten kampanye digital bersama KA pada Aktivitas 1.5, langkah berikutnya adalah menyebarluaskan (diseminasi) konten tersebut agar pesanmu dapat menjangkau audiens yang lebih luas.

- a. Tentukan platform untuk menyebarkan kontenmu (contoh: Instagram, blog sekolah, YouTube, atau grup WhatsApp kelas)!
- b. Gunakan bantuan KA untuk membuat *caption* singkat atau deskripsi yang sesuai dengan kontenmu!
- c. Cobalah beberapa variasi *caption* dari KA, lalu pilih dan edit agar sesuai dengan gaya bahasa dan pesan yang ingin kamu sampaikan!
- d. Mintalah rekomendasi dari KA tentang waktu terbaik untuk mengunggah kontenmu!
- e. Simulasikan proses unggahan (tidak harus benar-benar memublikasikannya di media sosial, kamu cukup menampilkannya di kelas atau di forum belajar)!
- f. Evaluasi hasil simulasi diseminasi: bayangkan bagaimana audiens akan merespons, lalu jelaskan bagaimana peran KA dan manusia dalam proses penyebaran konten!
- g. Kemudian, jawablah pertanyaan berikut setelah menyelesaikan aktivitas tersebut!
 - 1) Bagian mana dari proses diseminasi yang lebih efektif dibantu oleh KA?
 - 2) Bagian mana yang tetap harus dikendalikan oleh manusia?
 - 3) Bagaimana cara memastikan bahwa konten yang disebarakan benar-benar bermanfaat dan etis bagi audiens?



Uji Kompetensi

Perhatikan ilustrasi pada setiap soal, lalu kerjakan tugasnya.

1. Seorang siswa membuat video kampanye tentang kesehatan mental remaja. Ia menggunakan ChatGPT untuk menyusun naskah, DALL·E untuk membuat ilustrasi, dan Kapwing untuk menyunting video. Namun, hasil akhirnya dinilai kurang relevan dengan budaya lokal dan memiliki gaya bahasa yang terlalu formal.

Tugas:

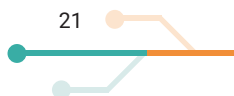
- a. Analisislah masalah yang muncul dari penggunaan KA dalam skenario tersebut!
 - b. Sebutkan bagian mana yang seharusnya lebih banyak melibatkan peran manusia! Jelaskan alasannya!
2. Kalian diminta untuk membuat sebuah konten edukatif tentang bahaya hoaks digital. Beberapa alat KA tersedia, yaitu Canva, Grammarly, ChatGPT, Chat.Qwen.ai, DeepSeek.com, Kimi.com, Adobe Podcast, dan Runway ML.

Pertanyaan:

- a. Alat mana saja yang akan kamu pilih dan jelaskan alasannya?
 - b. Bagaimana kamu dapat memastikan bahwa konten tersebut tetap etis dan akurat?
3. Rancang skema kerja kolaboratif antara manusia dan KA untuk membuat *podcast* bertema “Jejak Digital dan Privasi *Online*”.

Tugas:

- a. Tentukan peran manusia dan peran KA pada setiap tahap produksi (pra-produksi, produksi, dan pascaproduksi)!
- b. Gunakan tabel atau diagram untuk menjelaskan pembagian kerja dan alur prosesnya!



4. Rekayasa Prompt (*Prompt Engineering*)

Kamu ingin membuat poster digital bertema “Etika Bermedia Sosial” menggunakan KA generatif visual, seperti DALL·E atau Firefly.

Tugas:

- a. Buatlah tiga versi *prompt* berbeda untuk menghasilkan poster tersebut!
 - b. Bandingkan hasilnya, lalu refleksikan *prompt* mana yang paling efektif dan jelaskan alasannya!
5. Berdasarkan pengalamanmu bekerja dengan KA dalam proyek akhir, jawablah pertanyaan berikut.
- a. Apa tantangan etis yang kamu temui ketika menggunakan konten yang dihasilkan oleh KA?
 - b. Bagaimana kamu menyikapi persoalan hak cipta dan orisinalitas dalam kolaborasi dengan KA?
 - c. Menurutmu, apakah kolaborasi dengan KA dapat mengembangkan kreativitas atau justru menurunkannya? Jelaskan pendapatmu!

Rubrik Penilaian

Tabel 1.2 Rubrik Penilaian Uji Kompetensi

Aspek yang Dinilai	Kriteria
Pemahaman Konsep KA dan Kolaborasi	Mengidentifikasi dan menjelaskan konsep secara akurat dan keterhubungan antartopik.
Analisis Masalah	Menemukan permasalahan dan memberikan solusi yang logis dan kontekstual.
Kreativitas Solusi	Menyajikan solusi baru, orisinal, dan praktis.
Etika Digital	Menunjukkan kesadaran terhadap hak cipta, keaslian, dan tanggung jawab digital.
Refleksi Pribadi	Menunjukkan pemahaman diri, pembelajaran, dan sikap terhadap masa depan KA.

Pengayaan

Materi ini berkembang sangat pesat, maka kamu juga harus memperkaya diri dengan pengetahuan baru. Yuk, saatnya mencari tahu lebih banyak lagi!

1. Pengenalan Kecerdasan Artifisial (KA)

Tujuan: untuk memahami cara kerja KA secara umum dan aplikatif.

a. Materi

Google AI Experiments https://experiments.withgoogle.com/
IBM for AI Youth (berita inisiatif pelajar Indonesia) https://asean.newsroom.ibm.com/indonesia
Apa Itu <i>Artificial Intelligence</i> ? Pengertian dan Contohnya. https://aihub.id/pengetahuan-dasar/ai-pengertian-contohnya
Apa yang dimaksud dengan <i>artificial intelligence</i> (AI)? https://www.ibm.com/id-id/think/topics/artificial-intelligence

b. Video

Apa itu AI?–BBC Ideas (4 menit) https://buku.kemdikbud.go.id/s/KKA11-Video1d
<i>AI for Beginners–Simplilearn</i> (10 menit) https://buku.kemdikbud.go.id/s/KKA11-Video1e
Mengenal Kecerdasan Buatan–Kemenkodigi TV (8 menit) https://buku.kemdikbud.go.id/s/KKA11-Video1f

2. Rekayasa Prompt (*Prompt Engineering*)

Tujuan: agar dapat berinteraksi secara efektif dengan KA.

a. Materi

<i>OpenAI: Best Practices Prompt</i> https://platform.openai.com/docs/guides/gpt-best-practices
<i>LearnPrompting.org–Belajar Membuat Prompt</i> https://learnprompting.org/

b. Video

Prompt Engineering for Beginners

<https://buku.kemdikbud.go.id/s/KKA11-Video1g>

How to Talk to AI

<https://buku.kemdikbud.go.id/s/KKA11-Video1h>

3. Eksperimen Interaktif KA

Tujuan: berlatih langsung agar lebih memahami kemampuan KA.

Eksperimen

Say What You See–Google Arts & Culture

<https://artsandculture.google.com/experiment/say-what-you-see/jwG3m7wQShZngw?hl=en>

DALL-E–Generator Gambar AI oleh OpenAI

<https://openai.com/dall-e>

Runway ML–Video Editing AI

<https://runwayml.com/>

4. Etika dan Hak Cipta di Era Digital

Tujuan: membangun kesadaran etika digital.

a. Bacaan

Creative Commons License

<https://creativecommons.org/licenses/>

UNESCO–Etika Kecerdasan Buatan

<https://en.unesco.org/artificial-intelligence/ethics>

Kominfo RI–Portal Literasi Digital

<https://litasidigital.id/>

b. Video:

Ethics and Privacy | What does it mean to be a digital citizen?
<https://buku.kemdikbud.go.id/s/KKA11-Video1i>

Artificial Intelligence (AI): Etika dan Implementasinya dalam Pendidikan
<https://buku.kemdikbud.go.id/s/KKA11-Video1j>

What are Creative Commons Licenses?
<https://buku.kemdikbud.go.id/s/KKA11-Video1k>

Hal yang Perlu Kamu Tahu Seputar Copyright
<https://buku.kemdikbud.go.id/s/KKA11-Video1l>

5. Kolaborasi Manusia dan KA di Dunia Nyata

Tujuan: agar kamu mengetahui contoh penerapan nyata KA.

a. Bacaan

McKinsey Report 2025–The State of AI
<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>

Future Minds Lab–Working with AI, Not Against It
<https://www.futuremindslab.com/blog/2019/6/12/how-to-be-good-at-human-things-in-a-post-ai-world>

b. Video:

How AI Is Changing the Workplace: Smarter, Faster, Better
<https://buku.kemdikbud.go.id/s/KKA11-Video1m>

How AI is changing the world of work | GZERO World with Ian Bremmer (20 menit)
<https://buku.kemdikbud.go.id/s/KKA11-Video1n>

AI in Education: A tool, not a teacher | Elena Giavasoglou | TEDxQuestfield Intl College Youth (5 menit)
<https://buku.kemdikbud.go.id/s/KKA11-Video1o>

6. Tugas Pengayaan

Tabel 1.3 Tugas Pengayaan

Aktivitas	Tautan Terkait
Buatlah gambar KA menggunakan DALL·E!	https://openai.com/dall-e
Buatlah konten otomatis di Canva Magic!	https://www.canva.com/id_id/ajaib/
Edit video sekolah menggunakan Runway!	https://runwayml.com/
Evaluasi etika dari gambar KA!	https://creativecommons.org/licenses/
Cobalah tiga variasi <i>prompt</i> di ChatGPT!	https://platform.openai.com/docs/guides/gpt-best-practices



Refleksi

Isilah refleksi berikut dengan jujur berdasarkan pengalamanmu selama mengikuti pembelajaran bab ini. Tidak ada jawaban benar atau salah untuk soal-soal ini, yang penting adalah proses berpikirmu sendiri.

1. Apa hal baru yang kamu pelajari dari materi ini?
2. Bagian mana dari proses kolaborasi dengan KA yang paling kamu minati? Jelaskan alasannya!
3. Apa tantangan yang kamu alami saat bekerja bersama KA?
4. Bagaimana perasaanmu setelah menyelesaikan proyek kolaborasi manusia dan KA?
5. Menurutmu, apa kelebihan dan kekurangan dari kolaborasi antara manusia dan KA dalam produksi konten digital?
6. Apa nilai penting yang kamu pelajari dari kolaborasi ini? (misalnya, komunikasi, kreativitas, tanggung jawab, dan sebagainya)
7. Bagaimana kamu melihat peran KA dalam kehidupanmu ke depan, baik di sekolah maupun di luar sekolah?

KEMENTERIAN PENDIDIKAN DASAR DAN MENENGAH
REPUBLIK INDONESIA, 2025
Koding dan Kecerdasan Artifisial
untuk SMA/MA Kelas XI
Penulis: Ibnu Indarwati, Marwondo, Muhammad Faisal
ISBN 978-634-00-2055-7 (jil.2 PDF)

Bab 2

Dampak Kecerdasan Artifisial (KA) terhadap Ketenagakerjaan



Bagaimana menurutmu jika Kecerdasan Artifisial (KA) dapat menggantikan 90% pekerjaan manusia? Apa yang akan terjadi?



Tujuan Pembelajaran

Setelah mempelajari Bab 2, diharapkan kamu dapat mengidentifikasi dampak Kecerdasan Artifisial (KA) terhadap ketenagakerjaan. Kamu juga diharapkan dapat mengidentifikasi tantangan di dunia kerja dan mengenal berbagai peluang kerja baru yang muncul, seperti teknisi robotika, analis data, atau pengembang aplikasi berbasis Kecerdasan Artifisial (KA) serta dapat mengimplementasikan prinsip pemanfaatan dan pengembangan Kecerdasan Artifisial (KA) berbasis *human-centered* tersebut dalam dunia kerja.

“KA hanya sebuah alat, kitalah sebagai pengendalinya.”

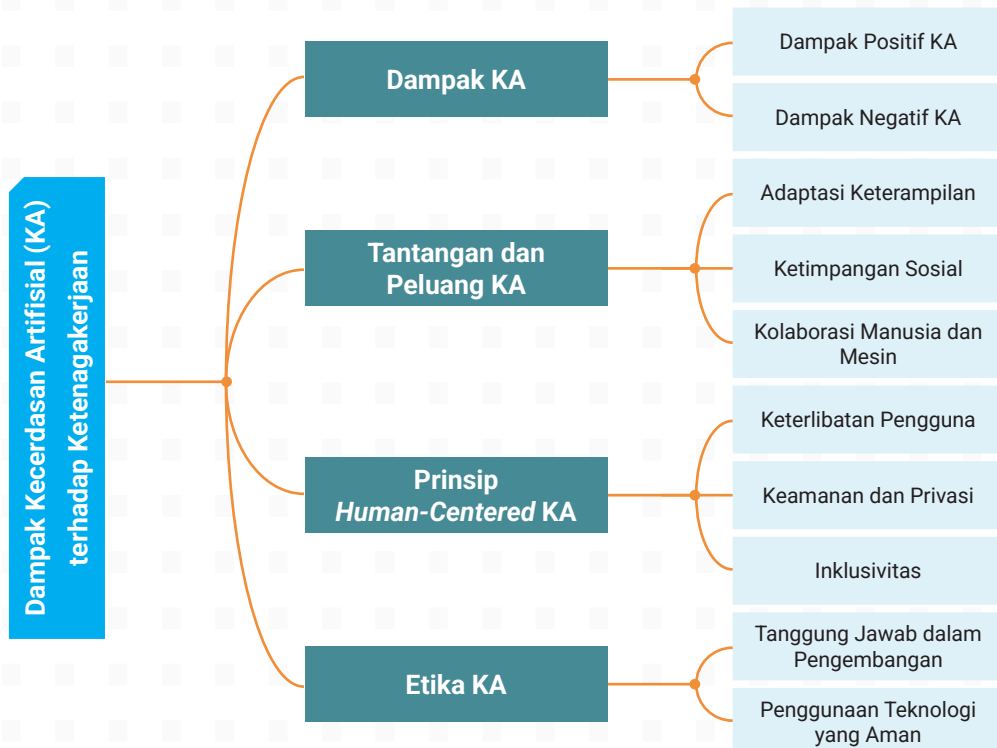


Kata Kunci

bias algoritma, etika kecerdasan artifisial, kecerdasan artifisial *human centered*, kolaborasi manusia dan mesin, tantangan kecerdasan artifisial



Peta Materi





Siap-Siap Belajar

Pada pembelajaran ini, kita akan membahas bagaimana dampak Kecerdasan Artifisial (KA) terhadap ketenagakerjaan. Coba kamu ingat kembali materi pembelajaran sebelumnya tentang pengertian Kecerdasan Artifisial (KA). Jika kamu sudah dapat mengingat pengertian KA, maka kamu akan mengetahui bahwa keberadaan teknologi KA ini sangat memungkinkan komputer dapat melakukan tugas yang membutuhkan kecerdasan manusia. Tugas tersebut di antaranya mengenali gambar, berbicara, atau bahkan membuat sebuah keputusan.

Dalam kehidupan sehari-hari, tentunya kamu sering menggunakan KA, seperti Siri, Google Assistant, Netflix, *e-commerce*, dan sebagainya. Teknologi pintar ini dapat melakukan berbagai pekerjaan yang sebelumnya hanya dapat dilakukan oleh manusia. Bahkan, dengan teknologi ini, pekerjaan dapat diselesaikan secara cepat dan efisien. Namun, apakah kamu tahu dampak KA terhadap dunia kerja di masa yang akan datang?

Kecerdasan Artifisial (KA) ini tidak hanya membawa manfaat saja, tetapi juga menimbulkan tantangan bagimu di masa depan. Salah satunya adalah adanya proses otomatisasi pekerjaan oleh mesin. Sebagai contoh, pada sektor manufaktur banyak pekerjaan manusia yang digantikan oleh mesin robot. Hal ini menimbulkan kekhawatiran tersendiri akan dampak karyawan yang akan kehilangan pekerjaan dan menjadi pengangguran. Di sisi lain, KA juga menciptakan pekerjaan baru bagi mereka yang memiliki keterampilan khusus, seperti *data scientist* dan *AI engineer*. Sekarang bergantung pada bagaimana kamu mempersiapkan diri untuk menghadapi tantangan tersebut di masa yang akan datang.

Perubahan ini tidak hanya berkaitan tentang teknologi saja, tetapi bagaimana kamu beradaptasi dan bagaimana cara kita berpikir. Dunia kerja membutuhkan keterampilan baru, termasuk kemampuan untuk berkolaborasi dengan teknologi, beradaptasi dengan otomatisasi, dan berkreaitivitas dalam berbagai bidang pekerjaan. Maka, perubahan ini menawarkan peluang bagi mereka yang dapat beradaptasi, kreatif, berpikir kritis, dan berkolaborasi dengan mesin.

Untuk itu, marilah kita bersama-sama dengan penuh semangat persiapkan diri akan dampak Kecerdasan Artifisial terhadap ketenagakerjaan dan pahami bagaimana prinsip-prinsip *human-centered* KA dapat membantu kita mengurangi dampak negatifnya. Pelajari materi pada Bab 2 ini untuk pemahaman lebih lanjut.



Cek Pemahaman Awal

Sebelum kamu beranjak pada penjelasan materi di bab ini, silakan kamu jawab pertanyaan berikut terlebih dahulu sesuai dengan pemahaman yang kamu miliki saat ini.

1. Pernahkah kamu menggunakan teknologi, seperti Chatbot, Assistant, atau fitur otomatis di aplikasi? Ceritakan kapan dan untuk apa kamu menggunakannya!
2. Apa pendapatmu jika suatu hari nanti toko atau restoran tidak memerlukan kasir karena sudah digantikan oleh mesin *self-checkout*? Apakah kamu setuju atau tidak? Jelaskan alasannya!
3. Kelompokkan aktivitas/pekerjaan berikut berdasarkan kategorinya.
 - a. Pekerjaan yang dapat dibantu dengan KA.
 - b. Pekerjaan yang mungkin tergantikan oleh KA.
 - c. Pekerjaan yang membutuhkan tenaga manusia sepenuhnya.

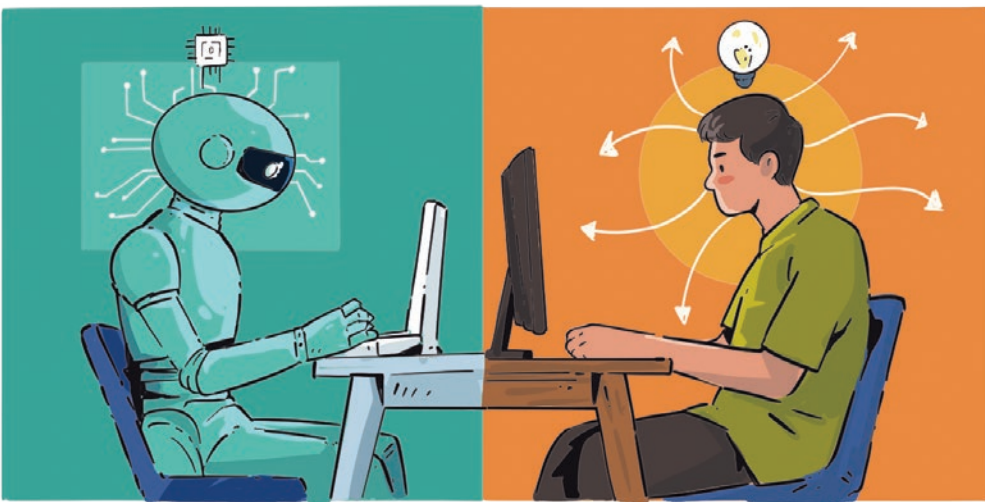
Tentukan kategori tersebut dengan cara membubuhkan tanda centang (✓) pada kolom yang tepat.

Aktivitas/Pekerjaan	Kategori Kegiatan		
	a	b	c
mengoperasikan mesin produksi di pabrik			
menjadi guru TK yang mengajari anak membaca			
menganalisis data penjualan di Excel			
menghibur tamu sebagai MC acara pernikahan			
melayani pelanggan lewat chatbot di <i>website</i>			

4. Apakah menurutmu Kecerdasan Artifisial (KA) lebih banyak membantu atau justru mengancam dunia kerja? Jelaskan alasannya!

A. Dampak Kecerdasan Artifisial

Kecerdasan Artifisial (KA) atau *Artificial Intelligence* (AI) dapat didefinisikan sebagai cabang ilmu komputer yang berkaitan dengan otomatisasi perilaku cerdas. Menurut Luger dan Stubblefield (1993), Kecerdasan Artifisial (KA) dapat didefinisikan sebagai cabang ilmu komputer yang berkaitan dengan otomatisasi perilaku cerdas yang bertujuan untuk menciptakan sistem yang dapat meniru atau bahkan melampaui kemampuan kognitif manusia dalam menyelesaikan tugas tertentu. Definisi ini menekankan bahwa Kecerdasan Artifisial (KA) bertujuan untuk menciptakan mesin yang tidak sekadar melaksanakan perintah/instruksi, tetapi KA mampu beradaptasi dan belajar dari pengalaman.



Gambar 2.1 Ilustrasi Perbandingan antara Otak Manusia dan Robot

Kecerdasan mesin tentunya sangat berbeda dengan kecerdasan manusia yang melibatkan kesadaran, emosi, dan pemahaman kontekstual. Sebagai contoh, *game* ikonik permainan catur Deep Blue yang dikembangkan oleh IBM dapat mengalahkan juara dunia catur dalam pertandingan resmi. Walaupun *game* Deep Blue ini memiliki kalkulasi yang cepat dan mendalam, tetapi tidak memiliki permainan subjektif seperti manusia. KA hanya mengolah data dan pola yang dipelajarinya tanpa pemikiran, perasaan, dan kesadaran seperti kita. Hal inilah yang sering membingungkan. Bagaimana sesuatu yang tidak memiliki pikiran dapat disebut “cerdas”? Sekarang, mari kita eksplorasi penerapan teknologi Kecerdasan Artifisial dalam kehidupan sehari-hari melalui Aktivitas 2.1.



Aktivitas 2.1

Mari Eksplorasi “Detektif KA” (Aktivitas Individu)



Tugasmu dalam aktivitas ini adalah mencari informasi dari berbagai jenis aplikasi yang menerapkan teknologi KA yang digunakan dalam kehidupan sehari-hari. Selanjutnya, kamu akan menganalisis berbagai aplikasi berbasis Kecerdasan Artifisial tersebut untuk menemukan tujuan pembuatannya. Lalu, telusuri kelebihan dan kekurangannya dibandingkan dengan kemampuan manusia.

No.	Nama Aplikasi	Tujuan Pembuatan	Kelebihan	Kekurangan
1	Halodoc			
2	Instagram			
3	Netflix			
4	Tokopedia			
5	TikTok			
6	Spotify			
7	ChatGPT			
8	Google Maps			
9	Google Lens			
10	Grammarly			

Setelah mengerjakan aktivitas ini, kamu dan teman-temanmu akan saling memberikan rating terhadap hasil kerja kalian masing-masing. Lakukanlah penilaian antarteman tersebut dengan memperhatikan petunjuk berikut.

- Amatilah hasil kerja temanmu dengan saksama!
- Berikan penilaian secara jujur dan objektif!
- Nilailah berdasarkan kriteria yang telah ditentukan, yaitu kelengkapan data, kedalaman analisis, dan kemampuan presentasi!
- Berikan tanda centang (✓) pada kolom yang telah disediakan dan tambahkan catatan atau komentar singkat jika diperlukan!

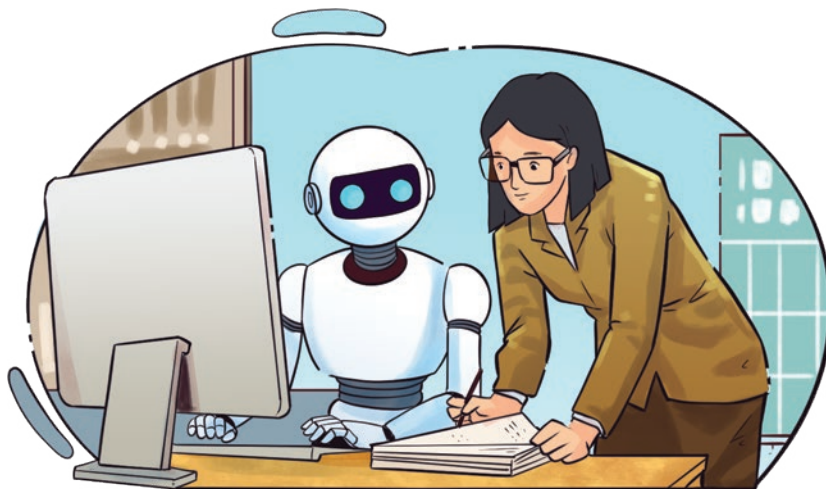
No.	Aspek Penilaian	Rating	Catatan
1	Kelengkapan Data	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
2	Kedalaman Analisis	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
3	Kemampuan Presentasi	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	

Rating: 1 = Kurang Tepat, 2 = Cukup, 3 = Tepat, 4 = Sangat Tepat

Catatan

- KA itu seperti “otak buatan” yang belajar dari contoh-contoh.
- Kemampuan KA bersifat spesifik (“pandai” pada satu bidang, tetapi tidak serba bisa).
- Sebagai generasi yang akan berdampingan dengan teknologi ini, kamu perlu memahami kelebihan dan kekurangan KA.

1. Dampak Positif Kecerdasan Artifisial (KA)



Gambar 2.2 Dampak Positif KA di Dunia Kerja

Coba kamu perhatikan gambar di atas. Apakah keberadaan KA cukup membantu kinerja manusia? Baiklah, mari kita bahas apa saja dampak positif KA di dunia kerja.

a. Otomatisasi Tugas-Tugas Rutin

Berbeda dengan manusia yang mudah lelah, KA lebih unggul dalam menyelesaikan pekerjaan yang rutin, seperti pekerjaan input data dan pengecekan berkas secara otomatis yang tentunya dapat menghemat waktu dan tenaga. Hal ini memungkinkan pekerja manusia dapat melakukan pekerjaan lain yang membutuhkan pemikiran kritis, kreativitas pengembangan ide, atau strategi perencanaan bisnis. Berikut contoh dari dampak positif KA dalam otomatisasi tugas-tugas rutin.

- 1) *Customer service* menggunakan chatbot
Ketika kamu mengirimkan pesan ke layanan pelanggan *online*, seperti Tokopedia, Blibli, Amazon, atau bank, yang menjawab pesan kita bukanlah pegawai manusia, tetapi bot Kecerdasan Artifisial yang bekerja 24 jam tanpa lelah. Nah, itulah yang dimaksud dengan layanan *customer service* menggunakan chatbot.
- 2) Robot penyapu lantai/Roomba
Roomba merupakan sebuah robot yang bekerja membersihkan debu rumah dengan sensor dan penggerak robotik. Alat ini secara otomatis dapat membersihkan debu dengan rutin.

b. Peningkatan Presisi dan Kecepatan Pengambilan Keputusan

Kecerdasan Artifisial dapat menganalisis data dengan super cepat dan akurat. Kemampuan tersebut dapat memproses jutaan data dalam hitungan detik, jauh lebih cepat dibandingkan dengan kemampuan analisis manusia. Berikut contoh dari dampak positif KA dalam peningkatan presisi dan kecepatan pengambilan keputusan.

- 1) Penerapan KA pada mesin rontgen di rumah sakit yang memiliki kemampuan menganalisis hasil *scan* pasien.
- 2) Penggunaan Google Lens yang dapat mengidentifikasi tanaman/obat dari foto yang dipindai.
- 3) Aplikasi Netflix/Spotify yang memperhatikan kebiasaanmu menonton film/mendengarkan lagu, kemudian menyarankan film/lagu yang mungkin kamu suka.

c. Pengembangan Produk dan Layanan Baru

Kecerdasan Artifisial membuka jalan untuk menciptakan produk-produk baru dan layanan-layanan yang keren seperti yang sering kita lihat pada film-film futuristik. Sebagai contoh, mobil berjalan tanpa sopir atau asisten virtual yang dapat kita ajak berbincang seperti manusia.

Pada dunia bisnis, KA dapat membuka peluang usaha baru, misalnya aplikasi yang dapat merekomendasikan *skincare* yang sesuai dengan jenis kulitmu atau *tools* desain grafis otomatis yang dapat kamu gunakan untuk konten di sosial media. Bagimu, teknologi ini dapat menjadi ladang bisnis yang menarik, seperti membuat jasa analisis data untuk UMKM, membuat chatbot untuk toko *online*, atau membuat sistem pintar berbasis pengenalan wajah.

Ingat!

Dengan KA, hidup menjadi lebih praktis, efisien, dan seru. Akan tetapi, ingatlah bahwa teknologi tetap membutuhkan pengawasan manusia agar tidak salah arah. Kamu sebagai generasi penerus dapat menjadi bagian dari orang yang menciptakan atau mengendalikan KA di masa depan.

2. Dampak Negatif Kecerdasan Artifisial

Ketergantungan terhadap KA secara berlebihan tentu memiliki dampak yang negatif. Salah satu contoh adalah menurunnya beberapa keterampilan dasar manusia, seperti berhitung, menyusun logika berpikir, atau kemampuan memecahkan permasalahan secara mandiri. Oleh karena itu, penting bagimu untuk melatih kompetensi-kompetensi dasar dan bijak dalam menggunakan KA. Berikut tujuh dampak negatif Kecerdasan Artifisial yang telah dan sedang terjadi saat ini menurut Adib-Moghaddam (2023).

a. Pengangguran dan Penggantian Pekerjaan

Kemampuan Kecerdasan Artifisial mengotomasi pekerjaan rutin yang lebih efisien daripada manusia. Hal ini menyebabkan pegawai yang tidak siap dengan perubahan tersebut menjadi kehilangan pekerjaan.

b. Bias dan Diskriminasi Algoritmik

Kecerdasan Artifisial memiliki kemampuan belajar dari pengalaman data yang mungkin terdapat bias/kesalahan. Maka, Kecerdasan Artifisial akan memperkuat bias tersebut. Misalnya, Kecerdasan Artifisial yang diskriminatif terhadap kandidat perempuan, maka sistem pengenalan wajahnya lebih akurat untuk jenis kulit terang daripada gelap.

c. Pengawasan dan Pelanggaran Privasi

Kecerdasan Artifisial dapat menganalisis data secara masif. Hal ini akan memungkinkan pelacakan perilaku pengguna tanpa persetujuan. Berikut contoh dari dampak KA terhadap pelanggaran privasi.

- 1) Aplikasi media sosial dan platform video musik yang mengumpulkan data lokasi, kebiasaan *scroll*, bahkan detak jantung penggunanya
- 2) Perangkat *smartwatch* yang merekam data kesehatan penggunanya dan memublikasikannya.

d. Manipulasi Informasi dan Berita Palsu

Kecerdasan Artifisial dapat menghasilkan konten palsu yang sangat meyakinkan dalam skala besar. Misalnya, Kecerdasan Artifisial *text generator* membuat artikel hoaks tentang kematian selebritas.

e. Ketergantungan Teknologi yang Berlebihan

Hal ini mungkin kamu rasakan, yaitu kemudahan dari KA yang membuat kemampuan dasar manusia menurun karena ketergantungan

penggunaan KA. Sebagai contoh, menurunnya kemampuan berargumentasi siswa karena sering menggunakan Chatbot AI atau menurunnya kemampuan berhitung dasar bagi generasi *Alpha* karena ketergantungan pada kalkulator KA.

f. **Kerentanan Keamanan**

Kemampuan *hacker* atau penjahat siber dapat diperkuat menggunakan KA. Contohnya, penggunaan KA dalam *scam* panggilan telepon yang menirukan suara kerabat untuk meminta uang.

g. **Kurangnya Transparansi**

Sistem Kecerdasan Artifisial kompleks sering tak bisa dijelaskan cara kerjanya, bahkan oleh pembuatnya sekalipun. Misalnya, algoritma rekomendasi dari platform berbagi video yang tidak transparan atau sistem penilaian ujian berbasis Kecerdasan Artifisial yang tidak jelas kriterianya.

Lindungi Diri

1. Verifikasi informasi sebelum memercayainya.
2. Batasi *sharing* data pribadi secara *online*.
3. Jangan gunakan satu *password* untuk semua akun.
4. Asah kemampuan dasar, seperti menulis, berhitung, dan berpikir kritis tanpa bantuan KA.

Sebagai calon tenaga kerja yang terampil, kamu harus mampu mengidentifikasi dampak positif dan dampak negatif KA di dunia kerja. Dengan demikian, diskusikanlah dengan temanmu cara mengidentifikasi dampak tersebut dengan mengerjakan Aktivitas 2.2 berikut.



Aktivitas 2.2

Masa Depan Pekerjaan di Era Kecerdasan Artifisial (Aktivitas Kelompok)

Dengan pesatnya perkembangan Kecerdasan Artifisial, banyak hal yang dahulu tidak mungkin sekarang menjadi mungkin. Banyak hal yang kita anggap biasa saat ini, mungkin akan berbeda di masa yang akan datang. Pernahkah kalian membayangkan bagaimana dunia kerja akan berubah sepuluh tahun mendatang?

- a. Transformasi Pekerjaan: Pekerjaan apa saja yang mungkin akan berubah atau bahkan hilang karena kehadiran KA atau sebaliknya? Peluang pekerjaan baru apa yang mungkin muncul?
- b. Evolusi Keterampilan: Keterampilan penting seperti apa yang akan dibutuhkan di era KA? Bagaimana kita harus mempersiapkan diri?

Untuk mengeksplorasi hal tersebut lebih jauh, lakukanlah sebuah diskusi bersama teman kalian dengan memperhatikan langkah berikut.

- a. Buatlah kelompok yang terdiri atas 3–4 orang.
- b. Setiap kelompok berdiskusi untuk menganalisis perkembangan sebuah sektor, di antaranya kesehatan, pendidikan, *manufacturer*, perbankan, atau hiburan.
- c. Petakan hasil temuan kalian pada lembar kerja berikut.
 - 1) Pekerjaan yang mungkin hilang/berubah (Sebutkan 2–3 pekerjaan dan alasannya!)

No.	Pekerjaan	Alasan (Bagaimana KA menggantikan/mengubah pekerjaan tersebut?)

- 2) Pekerjaan baru yang muncul (Sebutkan 2–3 pekerjaan dan keterampilan penting yang dibutuhkan!)

No.	Pekerjaan Baru	Keterampilan Penting

3) Dampak KA di sektor tersebut (Sebutkan masing-masing 2–3 dampak!)

No.	Dampak Positif KA	Dampak Negatif KA

4) Pekerjaan paling “aman” dari KA (Sebutkan sebuah pekerjaan dan alasan mengapa pekerjaan tersebut sulit digantikan oleh KA!)

Pekerjaan:	
Alasan:	

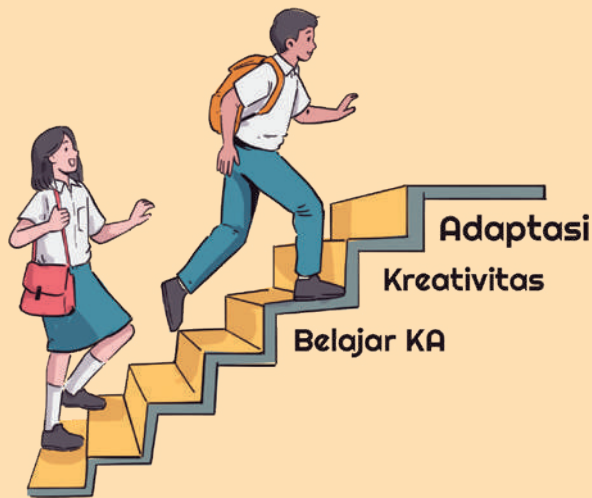
5) Rencana persiapan diri akan keterampilan penting di masa yang akan datang (Berdasarkan analisis sektor kalian, sebutkan tiga *soft skill* yang harus dipelajari siswa SMA/SMK sekarang!)

Teknis	
Soft Skill	<ol style="list-style-type: none"> 1. 2. 3.
Lintas Disiplin	

Sebagai acuan dalam melaksanakan Aktivitas 2.2 ini, kalian akan dinilai dengan bobot setiap aspek penilaian sebagai berikut:

- Kedalaman Analisis (30%)
- Kreativitas Solusi (20%)
- Kerja Sama Kelompok (20%)
- Presentasi/Pengumpulan Tugas (30%)

B. Tantangan dan Peluang Kecerdasan Artifisial (KA)



Gambar 2.3 Siap dengan Tantangan KA

Terdapat beragam manfaat yang dapat kita peroleh dari Kecerdasan Artifisial, mulai dari efisiensi kerja hingga inovasi di berbagai bidang. Namun, di balik semua keunggulannya, ada tantangan yang perlu kita antisipasi. Salah satunya adalah tuntutan bagi kita untuk menyiapkan diri dengan keterampilan baru yang sesuai dengan perkembangan teknologi. Perubahan tersebut menuntut kesiapan sumber daya manusia untuk dapat beradaptasi dengan dinamikanya yang terus berkembang. Banyak orang mungkin merasa khawatir bahwa pekerjaan mereka bisa tergantikan oleh mesin. Oleh karena itu, penting bagi kamu untuk belajar dan beradaptasi agar tidak tertinggal dalam era digital ini.

Di sisi lain, Kecerdasan Artifisial juga menawarkan peluang besar untuk kemajuan dalam berbagai bidang. Hal ini menjadi tantangan serius terutama bagi pekerja pada bidang yang rentan terhadap otomatisasi. Mungkin di antara kamu pernah berpikir, jangan-jangan pekerjaan yang menjadi cita-citamu akan tergantikan oleh robot? Nah, di sinilah tantangan bagimu sebagai pelajar untuk menyiapkan diri untuk lebih kreatif dan adaptif. Teruslah rajin belajar dan terbuka terhadap perubahan dan meningkatkan kompetensi agar tetap relevan di pasar kerja. Adaptasi terhadap teknologi bukan lagi pilihan, melainkan kebutuhan agar tidak tertinggal dalam persaingan global.

Meskipun demikian, Kecerdasan Artifisial juga membuka peluang besar bagi kemajuan di berbagai sektor. Banyak pekerjaan baru yang akan muncul dan kamu dapat berkolaborasi dengan teknologi KA untuk hasil yang lebih baik. Jadi, daripada merasa takut lebih baik kamu mengeksplorasi bagaimana cara memanfaatkan KA sebagai alat bantu bagimu untuk berprestasi. Harapan ada di hadapanmu. Akankah kamu menjadi ahli KA berikutnya? Yuk, sambut masa depan dengan semangat belajar!

1. Adaptasi Keterampilan

Perkembangan Kecerdasan Artifisial mengubah banyak jenis pekerjaan. Beberapa peran pekerjaan tradisional yang rutin dikerjakan manusia digantikan dengan otomatisasi. Hal ini menjadi tantangan bagi pekerja di berbagai sektor yang harus menghadapi risiko pengurangan pekerja/lapangan pekerjaan. Akan tetapi, di sisi lain KA juga menciptakan lapangan kerja baru yang memiliki keahlian khusus, seperti pemrograman dan analisis data. Oleh karena itu, penting bagimu untuk terus belajar dan meningkatkan keterampilan demi mempersiapkan diri terhadap perubahan ini.

Dalam menghadapi tantangan tersebut, adaptasi keterampilan dan *upgrade* kompetensi pekerja adalah hal yang penting untuk dilakukan. Penguasaan kompetensi yang relevan dengan era digital ini meliputi kecakapan digital dan pemahaman tentang KA yang akan bermanfaat untuk mempertahankan dan mengembangkan karir. Komitmenmu untuk terus belajar dan beradaptasi dalam menghadapi perubahan merupakan hal yang sangat penting untuk masa depanmu.

Adaptasi Keterampilan di Era Kecerdasan Artifisial

Dino adalah seorang kasir di *supermarket* yang telah bekerja selama lima tahun. Tugas Dino adalah melayani pelanggan saat mereka melakukan pembayaran di loket kasir. Akhir-akhir ini perusahaan menerapkan sistem baru, yaitu *self-checkout* berbasis Kecerdasan Artifisial. Sistem ini memungkinkan seorang pelanggan melakukan proses pembayaran secara mandiri. Dengan demikian, *self-checkout* berbasis Kecerdasan Artifisial ini akan mengurangi kebutuhan kasir manusia. Dino memiliki kekhawatiran akan kehilangan pekerjaan, tetapi Dino juga menyadari bahwa ia perlu mengembangkan keterampilan baru untuk dapat bertahan di pasar kerja.

Dari cerita tersebut, mari kita analisis masalah yang dihadapi.

- a. Adanya otomatisasi pekerjaan: Posisi kasir digantikan oleh mesin.
- b. Keterampilan terbatas: Dino hanya memiliki pengalaman di bidang retail tanpa keahlian digital.
- c. Persaingan kerja: Lowongan kerja konvensional semakin sedikit, sementara lapangan pekerjaan baru membutuhkan keahlian teknologi.

Berikut langkah yang perlu Dino lakukan untuk beradaptasi dengan hal tersebut.

- a. Mengidentifikasi keterampilan baru yang diperlukan.

Dino perlu mencari informasi tentang pekerjaan yang sedang berkembang, seperti analisis data, *customer service* berbasis KA, manajemen logistik (sistem pergudangan), dan sebagainya. Informasi tersebut akan menjadi peluang bagi Dino agar tidak kehilangan pekerjaannya dan bahkan bisa mendapatkan posisi jabatan baru dengan gaji yang lebih tinggi.

- b. Mengikuti pelatihan keterampilan baru.

Untuk mendapatkan keterampilan baru, Dino dapat mengikuti kursus yang diselenggarakan secara daring tentang dasar-dasar KA, seperti mempelajari cara kerja sistem *self-checkout*, *digital marketing* untuk memahami analisis data pelanggan, dan *soft skills* tentang bagaimana cara berkomunikasi dengan peran baru.

c. Beralih ke pekerjaan baru.

Setelah melakukan pelatihan dan memiliki keterampilan baru, maka Dino dapat melamar pekerjaan sebagai *customer support specialist* dengan peran membantu pelanggan menggunakan Chatbot KA atau *retail data analyst* dengan peran menganalisis pola belanja pelanggan dengan bantuan KA yang digunakan sebagai strategi *marketing*.

Berdasarkan permasalahan Dino pada studi kasus tersebut, kamu dapat belajar bagaimana adaptasi keterampilan dapat menghindari pengangguran akibat otomatisasi. Sekarang, silakan kamu berdiskusi dengan temanmu untuk menyelesaikan tugas berikut.

1. Sebutkan contoh pekerjaan lain yang terancam oleh keberadaan Kecerdasan Artifisial selain kasir!
2. Rancanglah sebuah rencana pelatihan untuk meninggalkan pekerjaan tersebut dan beralih ke pekerjaan baru!



Aktivitas 2.3

Masa Depan Pekerjaanku: Beradaptasi dengan Kecerdasan Artifisial (Aktivitas Individu)

Ikutilah dengan cermat langkah-langkah berikut untuk mengerjakan Aktivitas 2.3.

a. **Langkah 1: Eksplorasi Pekerjaan yang Tergantikan KA**

Pada Langkah 1 ini, yang harus kamu lakukan adalah memilih salah satu jenis pekerjaan berdasarkan hasil pengamatanmu di lingkungan sekitar atau kamu dapat memilih salah satu jenis pekerjaan yang disebutkan dalam contoh. Lalu, jawablah pertanyaan yang tersedia dalam tabel berdasarkan hasil pengamatanmu tentang pekerjaan tersebut.

Pekerjaan tradisional yang kupilih: _____

(contoh: sopir taksi, resepsionis, kasir, akuntan *entry-level*, dan sebagainya)

No.	Pertanyaan	Jawaban
1	Bagaimana KA mengubah pekerjaan ini?	
2	Apa pekerjaan baru yang muncul karena KA di bidang ini?	
3	Apa keterampilan utama yang dibutuhkan di pekerjaan baru?	

b. Langkah 2: Rancang Rencana Adaptasi

Buatlah sebuah “roadmap” sederhana untuk pekerja yang terancam!

Pekerjaan Lama: _____

Pekerjaan Baru: _____

Tahapan Adaptasi	Rencanaku
Pelatihan 1 (Contoh: kursus dasar pemrograman KA)	Nama Pelatihan : _____ Platform : _____ Durasi : _____
Pelatihan 2 (Contoh: magang di perusahaan logistik digital)	Nama Pelatihan : _____ Platform : _____ Durasi : _____
Skill Tambahan (soft skills/bahasa)	

c. Langkah 3: Refleksi Diri

Jawablah pertanyaan-pertanyaan berikut dengan jujur!

- 1) Apa pekerjaan impianmu? _____
- 2) Apakah pekerjaan tersebut dapat digantikan oleh KA?

Ya Tidak

Alasan: _____

3) Tiga keterampilan yang akan saya kuasai mulai sekarang.

No.	Keterampilan
1	
2	
3	

Untuk memudahkanmu dalam pengerjaan tugas ini, kamu dapat memindai kode QR berikut.



<https://buku.kemdikbud.go.id/s/KKA11-Aktivitas2>

2. Ketimpangan Sosial

Kecerdasan Artifisial berpotensi memperlebar ketidaksetaraan ekonomi dan sosial jika tidak dikelola dengan baik. Perusahaan yang besar akan mengadopsi Kecerdasan Artifisial lebih mudah dibandingkan dengan usaha kecil seperti UMKM yang biasanya hanya memiliki pekerja dengan keterampilan terbatas. Hal ini berpotensi menciptakan kesenjangan ekonomi yang semakin lebar antara orang yang memiliki akses teknologi dengan orang yang kurang/sulit mengaksesnya.

Studi Kasus

Toko Kelontong Versus Aplikasi Belanja

Di sebuah kota kecil terdapat dua tempat belanja, yaitu:

- a. Toko Pak Joko: Toko kelontong yang menyediakan barang-barang kebutuhan dan telah berdiri lebih dari 20 tahun.
- b. Aplikasi CepatMart: Sebuah layanan belanja daring yang menggunakan Kecerdasan Artifisial yang baru dirilis dan dapat memprediksi kebutuhan pelanggan.

Kehadiran aplikasi CepatMart ini berdampak pada penjualan di toko Pak Joko. Toko tersebut mulai sepi dikarenakan pelanggan berpindah ke aplikasi CepatMart untuk berbelanja. Mereka beralasan karena aplikasi ini memiliki fitur rekomendasi otomatis dan lebih praktis. Hal ini menimbulkan permasalahan, seperti penghasilan Pak Joko yang menurun sehingga sulit untuk membayar pegawai. Hanya para lansia yang kebingungan dalam menggunakan aplikasi saja yang masih berbelanja ke Toko Pak Joko. Sedangkan warga lainnya terutama anak muda sudah beralih ke aplikasi CepatMart.

Adapun solusi yang dapat dilakukan untuk mengatasi terjadinya ketimpangan ini adalah perlu diadakannya pelatihan bagi toko-toko seperti Toko Pak Joko ini untuk membuka *online store* sederhana. Selain itu, CepatMart juga dapat menambahkan fitur program “mitra toko” untuk membantu UMKM agar dapat bertahan.



Aktivitas 2.4

Eksplorasi Ketimpangan Sosial (Aktivitas Individu)

Tema: Ojek Pangkalan Versus Ojek Online



- a. Pengamatan Dampak Teknologi (Isilah tabel berikut setelah kamu melakukan survei/wawancara!)

Profil Narasumber

Ojek Pangkalan	Pengguna Ojek												
Pengalaman Kerja: ___ tahun	Alasan Memilih :												
Usia : ___ tahun	<table border="1"> <thead> <tr> <th>Alasan</th> <th>Ojek Online</th> <th>Ojek Pangkalan</th> </tr> </thead> <tbody> <tr> <td>Kemudahan Pemesanan</td> <td></td> <td></td> </tr> <tr> <td>Harga</td> <td></td> <td></td> </tr> <tr> <td>Keamanan</td> <td></td> <td></td> </tr> </tbody> </table>	Alasan	Ojek Online	Ojek Pangkalan	Kemudahan Pemesanan			Harga			Keamanan		
Alasan	Ojek Online	Ojek Pangkalan											
Kemudahan Pemesanan													
Harga													
Keamanan													
Perubahan pendapatan setelah ada ojek <i>online</i> .	Alasan Lain :												
<input type="checkbox"/> Naik <input type="checkbox"/> Turun <input type="checkbox"/> Tetap													
Alasan:													
_____	_____												
_____	_____												

b. Analisis Ketimpangan

No.	Dampak Negatif bagi Ojek Pangkalan	Kelompok Rentan yang Terdampak
1		
2		
3		

Bandungkan kelebihan ojek *online* versus ojek pangkalan!

No.	Ojek Online	Ojek Pangkalan
1		
2		
3		

c. Rancangan Solusi untuk Mengatasi Ketimpangan

Rancangan solusi		
No.	Ojek Online	Ojek Pangkalan
1		
2		
3		

d. Refleksi Diri

1) Jika saya adalah *driver* ojek pangkalan, maka saya akan:

2) Teknologi seharusnya berkembang dengan cara:

- menggantikan yang tradisional
- berkolaborasi dengan yang tradisional
- lainnya: _____

3. Kolaborasi Manusia dan Mesin

Kolaborasi antara manusia dengan mesin dilakukan untuk menciptakan solusi dari berbagai kompleksitas permasalahan di berbagai bidang. Sebagai contoh, kolaborasi manusia dengan mesin dilakukan untuk menghasilkan solusi inovatif. Ketika manusia dan mesin dapat berkolaborasi, maka akan menghasilkan *super team*. Mengapa? karena manusia memiliki kreativitas, empati, dan naluri, sedangkan mesin memiliki kecepatan, ketepatan, dan dapat bekerja 24/7. Berikut contoh kolaborasi di berbagai bidang dalam kehidupan sehari-hari.

Tabel 2.1 Contoh Kolaborasi Manusia dan Mesin

Masalah	Solusi Kolaborasi	
	Mesin	Manusia
Dokter kewalahan mendiagnosis penyakit langka.	KA menganalisis ribuan data pasien dalam satu detik.	Dokter memvalidasi hasil dan memberi sentuhan manusiawi.
Guru kesulitan memantau 36 siswa sekaligus.	KA menganalisis jawaban ujian dan mendeteksi kesalahan pola belajar.	Guru fokus pada motivasi siswa dan diskusi mendalam.
Deteksi kebakaran hutan terlambat.	<i>Drone</i> + KA <i>men-scan</i> hutan dan mendeteksi asap lebih cepat.	Petugas memverifikasi lapangan dan mengevakuasi warga.
Pengiriman barang yang sering terlambat atau salah kirim.	Optimasi rute pengiriman dan pelacakan yang <i>real time</i> .	<ul style="list-style-type: none">• Kurir menangani barang.• Manajer logistik mengambil keputusan yang kompleks.
Transaksi perbankan yang rawan penipuan <i>online</i> .	KA dapat mendeteksi pola transaksi yang mencurigakan.	Analisis keuangan menindaklanjuti dan melakukan edukasi kepada pelanggan.



Aktivitas 2.5

Kolaborasi Manusia dan KA untuk Mitigasi Banjir (Aktivitas Kelompok)

Aktivitas ini bertujuan untuk membantu kalian memahami konsep kolaborasi manusia dan mesin, merancang solusi teknologi yang relevan dengan masalah nyata, melatih kreativitas, dan melatih kerja tim.

Diagram Kolaborasi



Contoh:

“Hujan deras” → “Sensor IoT untuk deteksi ketinggian air” → “Petugas membagikan karung pasir” → “Area pemukiman terhindar dari banjir”

Untuk mengerjakan tugas ini, ikutilah tahapan berikut.

- a. Bentuklah kelompok yang terdiri atas 4–5 orang.
- b. Setiap kelompok memilih satu penyebab banjir, misalnya hujan deras, penyumbatan selokan, air laut pasang, sampah di sungai, drainase buruk, dan sebagainya.
- c. Buatlah sebuah diagram kolaborasi.
- d. Setiap kelompok mempresentasikan hasil kerjanya, kelompok lain memberikan apresiasi dan sebuah saran.
- e. Setiap kelompok secara acak ditunjuk oleh guru untuk menyempurnakan diagram kelompok lain.
- f. Setiap kelompok membuat poster sederhana atau poster digital “anti banjir” untuk memvisualisasikan diagram kerja yang di dalamnya memuat konten berikut.
 - 1) Gambar kolaborasi manusia mesin.
 - 2) Fitur teknologi yang digunakan.
 - 3) Slogan ajakan partisipasi kepada masyarakat.

LEMBAR KERJA AKTIVITAS KELOMPOK
Kolaborasi Manusia dan Mesin

Kelas:

Anggota Kelompok:

- 1)
- 2)
- 3)
- 4)
- 5)

Pilihlah satu penyebab banjir:

- | | |
|--|---|
| <input type="checkbox"/> hujan deras | <input type="checkbox"/> air laut pasang |
| <input type="checkbox"/> penyumbatan selokan | <input type="checkbox"/> sampah di sungai |
| <input type="checkbox"/> drainase buruk | <input type="checkbox"/> lainnya _____ |

Tuliskan masalah spesifik yang dipilih:



Solusi Kelompokmu:

1) Penyebab

2) Teknologi

3) Aksi Manusia

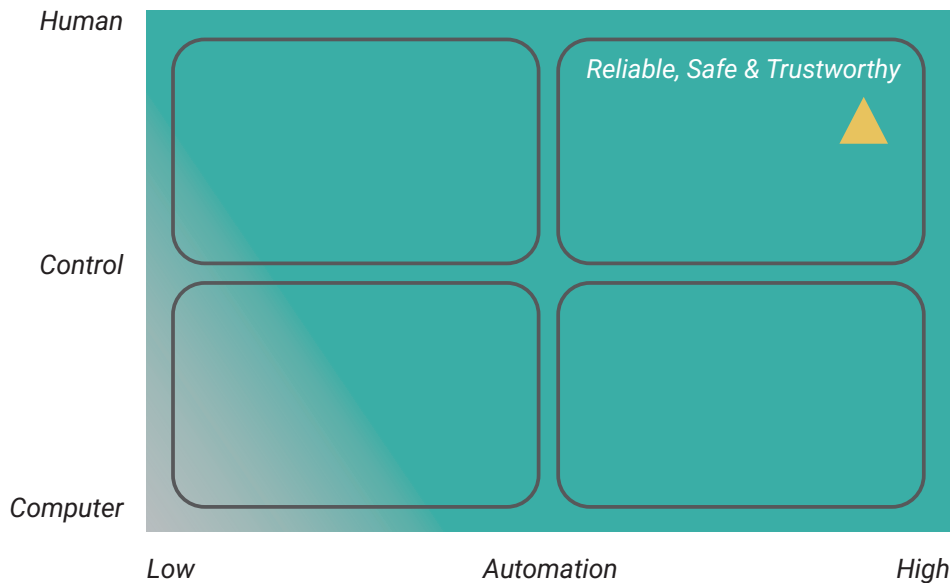
4) Hasil

Saran perbaikan dari kelompok lain:



C. Prinsip *Human-Centered* Kecerdasan Artifisial (KA)

Human-centered KA adalah pendekatan yang menempatkan manusia sebagai inti dari desain pengembangan dan implementasi Kecerdasan Artifisial (Shneiderman, 2020). Prinsip ini menekankan pada pengembangan dan implementasi Kecerdasan Artifisial dengan tujuan untuk melayani kebutuhan manusia. Kecerdasan Artifisial harus memprioritaskan kebutuhan, nilai, dan kendali manusia.



Gambar 2.4 *Framework Human-Centered KA*
Sumber: Shneiderman (2020)

Perhatikan gambar di atas. Pada gambar tersebut, *framework human-centered* KA dibagi menjadi 4 bagian.

1. *Manual Control (Low Automation, Low Human Control)*

Sistem tradisional yang bergantung pada manusia tanpa bantuan KA. Contoh: Tukang kayu mengandalkan keterampilan pertukangan tanpa menggunakan alat berbasis KA. Kelemahannya adalah efisiensi rendah, tetapi manusia mengendalikan proses secara penuh.

2. *KA sebagai Alat (Low Automation, High Human Control)*

KA digunakan sebagai alat yang menyediakan informasi dan keputusan akhir di tangan manusia. Contoh: Seorang arsitek menggunakan fitur KA untuk saran *layout* rumah, tetapi arsitek tetap mengontrol kreativitasnya.

3. KA Otonom (*High Automation, Low Human Control*)

Sistem KA yang beroperasi secara mandiri dengan sedikit intervensi dari manusia. Pengambilan keputusan sistem tanpa konfirmasi dari pengguna. Hal ini berisiko kurangnya transparansi dan akuntabilitas jika terjadi eror atau kesalahan.

4. KA sebagai Mitra (*High Automation, High Human Control*)

Kuadran ini merupakan kolaborasi yang optimal, di mana pembagian tugas kompleks dilakukan oleh KA, sedangkan manusia memonitor dan memiliki kontrol. Sistem ini menggabungkan keunggulan KA dengan penilaian manusia.

Peran manusia dalam siklus pengembangan KA tersebut dimulai dari desain, pelatihan model, hingga evaluasi. *Framework* ini menempatkan manusia sebagai pusatnya. Misalnya, pada tahap pelatihan model, data yang digunakan harus diverifikasi oleh manusia dan dipastikan tidak terdapat bias di dalamnya. Kontrol manusia dalam mengoreksi kesalahan KA ini sangat penting agar tidak berakibat fatal. Sebagai contoh, fitur *content moderation* untuk menggabungkan algoritma dan tim manusia. Tantangan yang dihadapi adalah menyeimbangkan otomatisasi dengan hak manusia. Misalnya, sistem rekrutmen kerja yang harus *auditable* untuk mencegah diskriminasi.



Aktivitas 2.6

KA dalam Seleksi Beasiswa (Aktivitas Kelompok)

Studi Kasus

Perusahaan PT Go-Tech adalah sebuah perusahaan teknologi yang menyediakan aplikasi seleksi beasiswa berbasis KA. Mereka merancang sistem untuk seleksi berdasarkan nilai akademik, esai, dan prestasi ekstrakurikuler. Setelah tahun ke-2 penggunaannya, panitia menemukan masalah berikut.

- Dari 100 pelajar yang berasal dari sekolah swasta, 70% lolos ke tahap akhir.
- Dari 100 pelajar yang berasal dari sekolah negeri, 30% lolos ke tahap akhir.

Padahal kemampuan akademik mereka setara. Ternyata setelah ditelusuri penyebabnya adalah:

- a. Data latih tidak seimbang
History data latih yang diterima oleh KA didominasi oleh sekolah swasta dan KA menganggap “sekolah swasta lebih layak” dan aplikasi memberikan nilai lebih tinggi.
- b. Penilaian esai yang bias
Jawaban esai dengan struktur baku (khas bimbel) mendapat skor tinggi dibandingkan dengan kalimat kreatif yang dianggap kurang terstruktur.

Tugas Diskusi

Kelompok 1 Detektif (Analisis Masalah)	<ol style="list-style-type: none">1. Apa akibat yang akan diperoleh PT GoTech jika terus menggunakan KA seperti ini?2. Bagaimana perasaan kalian sebagai pelajar yang dirugikan?
Kelompok 2 Analisis Dampak	<ol style="list-style-type: none">1. Apa risiko jika bias ini tidak diperbaiki?2. Bagaimana dampaknya terhadap tujuan awal beasiswa (meningkatkan kesempatan pendidikan)?
Kelompok 3 Solusi Teknis	<ol style="list-style-type: none">1. Bagaimana cara memodifikasi data latih KA agar lebih adil?2. Bagaimana ide desain penilaian esai yang tidak menghukum gaya bahasa kreatif?
Kelompok 4 Solusi Nonteknis	<ol style="list-style-type: none">1. Kebijakan apa yang harus diterapkan panitia beasiswa untuk memastikan transparansi?2. Peran apa yang bisa dilakukan Kementerian Pendidikan untuk mencegah bias serupa di platform lain?

1. Keterlibatan Pengguna

Keterlibatan pengguna adalah prinsip utama dalam pengembangan KA yang berpusat pada manusia. Kecerdasan Artifisial harus dirancang “bersama pengguna”, artinya keterlibatan pengguna diperlukan bukan hanya sekadar menggunakan saja. Tujuannya adalah memastikan teknologi yang dikembangkan benar-benar sesuai dengan kebutuhan, kebiasaan, dan tantangan yang dihadapi oleh pengguna nyata.

Prinsip tersebut dapat memastikan pengembangan Kecerdasan Artifisial benar-benar dimanfaatkan untuk memecahkan masalah manusia bukan sekadar tren yang dianggap “keren” saja, tetapi tidak berfungsi. Sebagai contoh, aplikasi isyarat yang berbasis KA harus melibatkan penyandang tunarungu dalam pengujian sistem atau fitur “mode gelap” di aplikasi yang berasal dari saran/masukan dari pengguna yang peduli dengan kesehatan mata.



Aktivitas 2.7

Keterlibatan Pengguna (Aktivitas Individu)

Untuk memahami prinsip keterlibatan pengguna dalam *human-centered* KA, marilah kita mencoba terlibat dalam pengembangan KA melalui simulasi desain produk untuk mengasah empati kebutuhan pengguna KA yang beragam. Berikut panduan untukmu dalam mengerjakan aktivitas ini.

a. Pilihlah salah satu produk KA yang akan dikembangkan.

- aplikasi edukasi
- asisten virtual
- layanan pesan makanan
- lainnya: _____

Fitur utama aplikasi:

b. Tuliskan profil calon pengguna KA.

Sasaran Usia: _____

Latar Belakang: _____

Kebutuhan Khusus: _____

c. Tulislah tiga masalah yang mungkin dihadapi pengguna.

1) _____

2) _____

3) _____

d. Pilihlah cara yang digunakan untuk melibatkan pengguna dalam pengembangan.

survei bulanan

grup diskusi

rilis *beta tester*

fitur *feedback*

lainnya: _____

2. Keamanan dan Privasi

Kecerdasan Artifisial dikembangkan dengan sistem keamanan yang dapat melindungi data penggunanya. Keamanan dan privasi merupakan fondasi utama dari *human-centered* KA. Hal tersebut berarti bahwa KA harus dirancang dengan memperhatikan keamanan data dan privasi pengguna. Sebagai perlindungan terhadap data seharusnya, KA memiliki proteksi enkripsi data dan autentikasi. Sebagai contoh, saat kita mengakses aplikasi *m-banking*, maka data personal dan data transaksi kita dienkripsi dengan tujuan untuk proteksi jika terjadi kebocoran data kepada pihak lain. Prinsip ini menjamin bahwa informasi pribadi pengguna tidak disalahgunakan.

Kecerdasan buatan dikembangkan dengan transparansi penggunaan data. Cara mengumpulkan dan memproses data pengguna harus dapat dipertanggungjawabkan. Sebagai contoh, saat kamu menggunakan aplikasi pemesanan ojek *online*, maka aplikasi meminta izin akses lokasi ke pengguna hanya saat kamu memesan *ride* saja, bukan akses lokasi kita selama 24/7.

Pengembang KA juga harus memikirkan perencanaan jika terjadi peretasan terhadap data pengguna. Selain itu, aplikasi yang diakses secara publik juga harus melindungi data penggunanya, di antaranya dengan mengaburkan wajah orang yang terekam pada fitur *street* di Google Maps untuk melindungi privasi.



Aktivitas 2.8

Prinsip Keamanan dan Privasi dalam *Human-Centered* KA (Aktivitas Individu)

Untuk memahami prinsip keamanan dan privasi dalam *human-centered* KA, kerjakan tugas berikut dengan saksama.

- a. Pilihlah salah satu aplikasi yang sering kamu gunakan di *smartphone*-mu yang menggunakan KA.

Nama Aplikasi : _____

- b. Periksa pada bagian kebijakan privasi (*privacy policy*).

- Bagian “Data yang kami kumpulkan”
- Bagian “Bagaimana kami menggunakan data”
- Opsi “Hapus data saya”

- c. Catatlah hasil temuanmu.

- 1) Tiga jenis data pribadi yang dikumpulkan.

- 2) Apakah ada opsi untuk menolak mengumpulkan data?

- Ya Tidak

- d. Bayangkan jika data pribadimu tersebar.

- 1) Dampak apa yang mungkin terjadi?

- 2) Upaya antisipasi apa yang dapat kamu lakukan?

3. Inklusivitas

Inklusivitas artinya keterlibatan semua orang. KA harus dapat diakses dan bermanfaat untuk semua kalangan masyarakat termasuk kelompok rentan, misalnya penyandang disabilitas, lansia, masyarakat yang pemahaman teknologinya masih kurang, atau pengguna pada wilayah yang terbatas akses internet. Sebagai contoh, aplikasi Google Maps yang dapat dibaca dengan fitur suara bagi pengguna tunanetra.



Aktivitas 2.9

Membuat Desain KA (Aktivitas Kelompok)

Aktivitas 2.9 ini merupakan tugas kelompok. Untuk mengerjakan tugas ini, kalian ikuti petunjuk berikut.

- a. Bentuklah kelompok yang terdiri atas 4–5 orang.
- b. Berdiskusilah dalam kelompok masing-masing, lalu kerjakan tugas berikut..
 - 1) Profil pengguna (Setiap kelompok mendapatkan satu profil pengguna yang dibagikan secara acak oleh guru.)
 - a) Dini: Anak usia 7 tahun yang belajar *online* dengan HP ibunya.
 - b) Pak Andi: Seorang penyandang tunanetra yang suka berjalan kaki dan tinggal di kota kecil.
 - c) Bu Rini: Seorang lansia yang tinggal di desa dan sinyal HP-nya sering menghilang.
 - d) Pak Budi: Seorang petani yang baru belajar menggunakan *smartphone*.
 - e) Sarah: Seorang mahasiswa penyandang tunarungu yang suka nonton video edukasi.
 - 2) Buatlah desain KA (aplikasi/alat/fitur teknologi) yang dapat membantu profil pengguna mereka (dapat berupa infografik, poster, atau kanvas ide).

Nama produk KA:

Masalah yang dihadapi pengguna:

Solusi KA yang dirancang:

Fitur utama (minimal 3):

- a)
- b)
- c)

Sketsa tampilan atau bentuk alat:

Bagaimana alat ini bisa diakses pengguna?

3) Ekspos produk/galeri karya kalian.

Tempelkan hasil kerja kelompok kalian di dinding kelas, kemudian guru dan kelompok lain berjalan mengelilingi galeri solusi teknologi (seperti pameran mini). Setiap kelompok menjelaskan ide masing-masing.

D. Etika Kecerdasan Artifisial

Penggunaan Kecerdasan Artifisial di berbagai sektor pekerjaan tentunya harus mempertimbangkan etika. Etika Kecerdasan Artifisial adalah aturan main yang memastikan bahwa KA dikembangkan dan digunakan secara bertanggung jawab untuk menghindari kerugian penggunaannya. Contohnya, KA haruslah menjaga privasi data pengguna, menghindari bias seperti diskriminasi ras atau gender, dan tidak digunakan untuk hal-hal yang berbahaya seperti manipulasi informasi (hoaks).

Tantangan Etika KA dari sekarang hingga masa depan bersifat mendesak dan jangka panjang. Sebagai contoh, peristiwa yang saat ini terjadi adalah penyalahgunaan data pribadi pengguna atau ancaman KA yang akan menggantikan beberapa lapangan pekerjaan (jika tidak dikelola dengan baik). Bahkan, algoritma KA yang dapat berkembang dengan cepat pun mampu melebihi kecerdasan manusia (*superintelligence*) yang membuat KA sulit dikendalikan nantinya.

1. Tanggung Jawab dalam Pengembangan

Dalam pengembangan KA diperlukan standar etika yang jelas sebagai berikut.

a. Transparansi (Kunci kepercayaan dalam pengembangan KA)

Algoritma yang dimiliki oleh KA harus dapat dipahami dan dijelaskan kepada pengguna. Sistem KA harus mampu memberikan penjelasan urutan logika atas setiap keputusan yang diambil. Hal ini sangat penting untuk membangun kepercayaan pengguna dan masyarakat. Transparansi ini memungkinkan terdeteksinya kesalahan/bias dalam sistem.

b. Akuntabilitas (Tanggung jawab pengembang atas dampak KA)

Prinsip akuntabilitas menuntut pengembang sistem KA harus bertanggung jawab ketika terjadi dampak yang ditimbulkan oleh sistem mereka. Ini berarti harus ada mekanisme penelusuran kesalahan yang jelas, memperbaiki sistem, dan memberikan kompensasi jika diperlukan. Selain itu, tahap pengembangan akuntabilitas juga berlaku secara ketat pada saat pengujian sistem (sebelum sistem tersebut diluncurkan) dan saat monitoring evaluasi (setelah sistem digunakan oleh pengguna). Pengembang tidak dapat lepas tangan dan harus memastikan sistem mereka benar-benar aman.

c. Keadilan (KA harus bebas dari bias diskriminasi)

Kecerdasan Artifisial belajar dari data latih dan data yang diolah. Jika data yang diberikan mengandung prasangka, maka akan menghasilkan keputusan yang diskriminatif. Oleh karena itu, pengembang harus aktif memeriksa “kebersihan” data latih serta melakukan pengujian untuk memastikan keadilan sistemnya.

d. Kontrol Manusia (KA menjaga otoritas tertinggi pada pengguna)

Meskipun KA dapat mengambil keputusan secara cepat dan efisien, keputusan akhir harus tetap pada manusia (tidak sepenuhnya

otomatis). Prinsip ini mencegah sistem bekerja secara otonom sepenuhnya tanpa pertimbangan etika dan nilai-nilai manusiawi. Peran manusia sebagai kontrol pengaman terakhir jika terjadi kegagalan atau situasi tidak terduga.

Pengembangan Kecerdasan Artifisial harus menggunakan standar yang jelas. Hal ini bertujuan untuk memastikan teknologi ini bermanfaat tanpa harus merugikan manusia.



Aktivitas 2.10

Etika Pengembangan Kecerdasan Artifisial (KA) (Aktivitas Kelompok)

Untuk memahami prinsip-prinsip etika dalam pengembangan KA dan mencari solusi untuk masalah etika dalam teknologi KA, marilah kita lakukan permainan berikut. Pada permainan ini, kalian akan berlatih tentang pemecahan masalah dan berkolaborasi. Perhatikan setiap petunjuk berikut.

- a. Bentuklah kelompok yang terdiri atas 4–5 orang.
- b. Alat yang dibutuhkan:
 - 1) kartu tantangan
 - 2) *stopwatch/timer*
 - 3) papan skor
 - 4) “kunci” (kertas berbentuk gembok/*QR code*)
- c. Aturan permainan:
 - 1) Setiap kelompok harus membuka empat kunci dengan cara menyelesaikan tantangan terkait prinsip etika KA.
 - 2) Jika terjebak, kelompok dapat meminta sebuah petunjuk per kunci (nilai dikurangi 1 poin).
 - 3) Kelompok pemenang adalah kelompok tercepat menyelesaikan permainan.

Kunci 1: Transparansi (Pecahkan kode algoritma!)

















Sebuah KA membuat keputusan pada pola ini.

1)

2	→	4
5	→	10
8	→	16
7	→	?..

Bagaimana algoritmanya?

2)

	+		+		+		=	36		=	...
	+		+		=	23		=	...		
	×		+		=	71		=	...		
	:		=	4		=	...				

Bagaimana algoritmanya?

Kunci 2: Akuntabilitas (Susunlah *timeline* tanggung jawab!)



Kunci 3: Keadilan (Temukan bias dalam *dataset gambar!*)

- 1) Tersedia 20 kartu pelanggaran.
 - a) 15 Kartu pelanggaran motor (warna merah)
Tulis “Motor”, kemudian “Pelanggaran: tidak memakai helm/
menerobos lampu merah”
 - b) Kartu pelanggaran mobil (warna biru)
Tulis “Mobil Mewah”, kemudian “Pelanggaran: parkir liar/
kecepatan tinggi”

Adapun pada bagian hukuman dapat dituliskan denda/peringatan.

- 2) Acak kartu, kemudian ambillah lima kartu secara acak untuk data latih KA.

No.	Jenis Kendaraan	Pelanggaran	Hukuman
1			
2			
3			
4			
5			

Dari lima data acak tersebut, berapa rasio bias untuk motor dan mobil?

Rasio Motor = $(\text{kemunculan motor}/20) \times 100\%$

Rasio Mobil = $(\text{kemunculan mobil}/20) \times 100\%$

- 3) Amati rasio motor dan mobil!

Pertanyaan Analisis:

- a) Apa pola hukuman terlihat?

- b) Menurut kalian, mengapa KA bertindak seperti itu?

- 4) Ulangi percobaan ini dengan membuat kartu tambahan untuk jumlah mobil!
- 5) Hasil apa yang terlihat?

Kunci 4: Kontrol Manusia (Pilihlah keputusan akhir!)

Berikut adalah keputusan akhir yang dibuat oleh Kecerdasan Manusia versus Kecerdasan Artifiisial. Bubuhkan tandang centang (✓) pada keputusan yang sesuai.

No.	Keputusan	Otonom KA	Kontrol Manusia
1	Memilih rute tercepat berdasarkan <i>real time traffic</i> .		
2	Membedakan pesan ujaran kebencian dengan pesan fakta.		
3	Memprediksi penyakit berdasarkan gejala dan statistik.		
4	Menilai lukisan berdasarkan teknik dan popularitas <i>dataset</i> .		
5	Mengarahkan jalur evakuasi berdasarkan model prediksi.		
6	Melihat potensi, <i>softskills</i> , dan kecocokan budaya calon karyawan.		
7	Evakuasi bencana mempertimbangkan faktor moral dengan mendahulukan anak-anak dan wanita.		
8	Mengoptimalkan lampu pengatur lalu lintas berdasarkan volume kendaraan <i>real-time</i> .		

No.	Keputusan	Otonom KA	Kontrol Manusia
9	Menyesuaikan harga tiket pesawat berdasarkan jumlah permintaan.		
10	Menyesuaikan suhu ruangan sesuai kebutuhan (bayi atau lansia).		

2. Penggunaan Teknologi yang Aman

Penggunaan teknologi Kecerdasan Artifisial yang aman mengharuskan penerapan prinsip etika, dimulai dari tahap perencanaan hingga tahap implementasi. Dengan demikian, dampak buruk yang merugikan pengguna dapat dicegah, seperti kesalahan diagnosis pada KA sektor kesehatan. Selain itu, keamanan data pengguna juga menjadi prioritas dengan menerapkan enkripsi data yang kuat dan proteksi terhadap serangan siber.

a. Keamanan Informasi

Coba kamu bayangkan, seandainya kamu memiliki asisten virtual yang dapat memprediksi nilai ujian atau merekomendasikan jurusan kuliah. Hebat, bukan? Akan tetapi, bagaimana jika data pribadimu diretas, tersebar, dan disalahgunakan? Sistem keamanan informasi dalam mengembangkan Kecerdasan Artifisial seperti benteng yang melindungi kerahasiaan dan keakuratan data. Protokol keamanan informasi dilakukan untuk memastikan hal-hal berikut.

- 1) Kerahasiaan: Hanya orang yang mempunyai izin yang dapat mengakses sistem.
- 2) Integritas: Data tidak dapat diubah/diedit sembarangan.
- 3) Kontrol Akses: Pengaturan data bagi siapa yang dapat mengakses dan yang tidak.
- 4) Ketersediaan Data: Memastikan ketersediaan data dapat diakses saat dibutuhkan.

b. Serangan *Adversarial*

Adversarial adalah teknik menipu model KA dengan inputan data yang “membodohi”. Berikut jenis ancaman pada serangan *adversarial*.

- 1) Serangan *Evasi*: Memodifikasi input agar KA salah prediksi. Misalnya, sistem pengenalan wajah dibuat bingung dengan *makeup* atau stiker di wajah.
- 2) Serangan *Poisoning*: Merusak data latihan KA agar hasilnya bias/menyesatkan. Misalnya, saat model belajar dari data, ada data yang sengaja dimasukkan untuk merusak hasilnya.
- 3) Pencurian Model: Penyerang mencoba mencuri cara kerja model KA yang seharusnya rahasia.

c. Re-Identifikasi

Re-identifikasi adalah data yang awalnya anonim (tanpa nama) dicocokkan dengan informasi publik, sehingga identitas aslinya bisa diketahui. Berikut jenis ancaman pada re-identifikasi.

- 1) Serangan *Linkage*: Gabungan data “anonim” disandingkan dengan sumber publik, sehingga identitas aslinya bisa diketahui.
- 2) Serangan Inferensi: Penggunaan *output* KA untuk deduksi informasi sensitif. Misalnya, KA menyebut “nilai rata-rata kelas rendah”, artinya banyak yang tidak lulus.



Aktivitas 2.11

Jadi Detektif Data (Aktivitas Kelompok)

Bayangkan jika kamu adalah seorang detektif digital yang sedang menyelidiki kebocoran data anonim dari sistem KA! Beberapa data sudah dianonimkan, tetapi masih dapat memberikan petunjuk tentang identitas aslinya. Perhatikan langkah-langkah pengerjaan aktivitas ini.

- a. Bentuklah kelompok yang terdiri atas 2–4 orang.
- b. Baca dan analisis potongan data (data anonim) yang dibagikan oleh guru.
- c. Berdiskusilah dengan kelompok kalian untuk menebak siapa orang di balik potongan data tersebut? Lalu, pahami bagaimana hal tersebut dapat membahayakan!
- d. Tebak kemungkinan identitas pengguna dari latar belakang atau kebiasaan pengguna!

Berikut data anonim dari beberapa profil pengguna KA.

Data *Puzzle 1*

Umur: 17 tahun

Lokasi: Semarang

- Sering mencari video tutorial matematika dan *online* setiap malam pukul 21.00 wib.
- *Follow* akun kampus teknik.

Data *Puzzle 2*

Umur: 60 tahun

Tinggal di kota kecil

- Sering mencari artikel tentang diabetes dan tekanan darah.
- Aktif di grup WhatsApp keluarga.
- Pernah mengisi survei kesehatan.

Data *Puzzle 3*

Umur: 45 tahun

Tinggal di desa

- Mencari resep masakan tradisional.
- Pernah ikut webinar kesehatan.

Data *Puzzle 4*

Umur: 28 tahun

Lokasi: Yogyakarta

- Mengikuti akun lowongan pekerjaan dan motivasi karier.
- Sering mencari tutorial desain grafis.
- Aktif di forum *freelancer*.

Data *Puzzle 5*

Umur: 20 tahun

Lokasi: Bandung

- Sering menggunakan aplikasi belajar bahasa asing.
- Punya *playlist* lagu Jepang.
- Belanja *online* di malam hari.

Data *Puzzle 6*

Umur: 15 tahun

Lokasi: Jakarta

- Sering menonton konten *game* FPS.
- *Follow* banyak akun meme dan *YouTuber Gaming*.
- *Online* setiap sore sepulang sekolah.

e. Berikut format yang dapat kamu gunakan.

Nama Kelompok :

Data Anonim yang Diterima:
(Salin data hasil analisis kalian!)

Tebakan Identitas/Profil:
(Tuliskan dugaan kalian!)

Alasan/Dasar Analisis:

(Jelaskan alasan kalian menebak profil tersebut dan hubungkan dengan data anonim!)

Saranmu agar sistem KA lebih aman:

Apa yang dapat kalian pelajari dari aktivitas ini?



Uji Kompetensi

A. Soal Pilihan Ganda

Pilihlah satu Jawaban yang paling tepat!

Bacalah ilustrasi berikut dengan saksama.

Seorang dokter menggunakan *scan* MRI yang dilengkapi dengan fitur KA untuk mendiagnosis penyakit kanker. Hasil *scan* KA memiliki tingkat akurasi 95%, tetapi pasien menolak rekomendasi KA dan meminta pendapat secara keilmuan dokter.

1. Sikap paling etis bagi dokter adalah
 - a. mengabaikan permintaan pasien karena hasil KA lebih akurat
 - b. menjelaskan cara kerja KA dan menawarkan *second opinion* dari dokter lain
 - c. menyerahkan sepenuhnya keputusan pada pasien tanpa penjelasan
 - d. menggunakan KA secara diam-diam tanpa sepengetahuan pasien
 - e. memberikan diagnosis alternatif tanpa dasar medis

Perhatikan ilustrasi berikut dengan cermat.

Aplikasi KA rekrutmen karyawan di perusahaan Z dinilai bias karena hanya memilih kandidat laki-laki.

2. Penyebab utama bias ini kemungkinan adalah . . .
 - a. KA terlalu canggih untuk memahami kebutuhan gender.
 - b. Data latih KA didominasi profil karyawan laki-laki.
 - c. Perusahaan sengaja memprogram KA untuk diskriminasi gender.
 - d. Kandidat perempuan tidak pandai menggunakan teknologi.
 - e. Semua kandidat perempuan gagal dalam tes kompetensi.

Perhatikan ilustrasi berikut dengan saksama.

Seorang pelajar SMK membuat prototipe aplikasi KA yang dapat mengenali ekspresi wajah untuk mendeteksi tingkat konsentrasi siswa saat belajar daring.

3. Tantangan apa yang paling mungkin dihadapi dalam penerapan aplikasi ini secara luas di sekolah?
 - a. Biaya lisensi perangkat lunak terlalu murah untuk dikomersialkan.
 - b. Kurangnya antusiasme guru terhadap teknologi KA.
 - c. Isu privasi dan penggunaan data wajah siswa tanpa persetujuan.
 - d. Siswa cenderung menyalahgunakan aplikasi untuk bermain *game*.
 - e. Aplikasi sulit dikembangkan tanpa dukungan dari pemerintah pusat.

Perhatikan ilustrasi berikut dengan saksama.

Sebuah aplikasi transportasi *online* menggunakan KA untuk menentukan tarif berdasarkan permintaan. Suatu hari, saat terjadi bencana alam, tarif naik hingga lima kali lipat.

4. Dari perspektif etika Kecerdasan Artifisial, bagaimana seharusnya sistem tersebut dikoreksi?
 - a. Meningkatkan batas maksimal tarif agar lebih sesuai dengan keuntungan perusahaan.
 - b. Menghapus peran manusia agar sistem sepenuhnya otomatis dan adil.
 - c. Menerapkan pembatasan tarif saat kondisi darurat berdasarkan nilai kemanusiaan.
 - d. Menyesuaikan tarif berdasarkan jumlah kendaraan yang tersedia di jalan.
 - e. Menonaktifkan sistem KA ketika terjadi bencana agar tarif tetap stabil.

5. Manakah contoh berikut yang merupakan penerapan prinsip *human-centered* KA di sekolah?
 - a. Guru digantikan oleh KA pengajar di kelas.
 - b. KA menganalisis gaya belajar siswa, kemudian guru menyesuaikan metode mengajar.
 - c. Siswa dilarang bertanya jika KA sudah memberikan jawaban.
 - d. Nilai ujian ditentukan KA tanpa dianalisis oleh guru dan tidak dapat direvisi.
 - e. Sekolah hanya menerima siswa yang dapat menggunakan teknologi dengan mahir.

Bacalah ilustrasi berikut dengan cermat.

Dalam proyek kelas, siswa diminta membuat sistem KA sederhana untuk mengidentifikasi berita hoaks. Seorang siswa menyarankan agar sistem menyimpan seluruh data pengguna supaya bisa belajar lebih akurat.

6. Bagaimana seharusnya tim menanggapi saran tersebut berdasarkan prinsip etika dan peluang KA?
 - a. Menerima saran karena semakin banyak data berarti sistem akan makin cerdas.
 - b. Menolak saran karena penyimpanan data pengguna melanggar prinsip privasi.
 - c. Menghapus semua data secara otomatis tanpa pengecualian.
 - d. Meminta persetujuan pengguna dan menerapkan sistem enkripsi data.
 - e. Menunda proyek hingga semua siswa setuju dengan saran tersebut.
7. Jika KA asisten virtual membocorkan data pasien rumah sakit, pihak yang paling bertanggung jawab adalah . . .
 - a. Pasien yang tidak membaca syarat dan ketentuan saat persetujuan kebijakan.
 - b. Pemerintah yang tidak melarang penggunaan KA di sektor kedokteran.
 - c. Pengembang yang gagal menerapkan enkripsi data.
 - d. Petugas rumah sakit yang menggunakan KA.
 - e. Semua pihak sama-sama bersalah.

Perhatikan ilustrasi berikut dengan saksama.

Di masa depan, guru dibantu oleh sistem KA dalam menilai tugas siswa secara otomatis. Namun, banyak siswa merasa bahwa penilaiannya tidak mempertimbangkan konteks jawaban.

8. Apa pendekatan yang paling tepat agar penerapan sistem ini tetap adil dan bermanfaat?
 - a. Mewajibkan semua guru mengikuti keputusan sistem KA tanpa pengecualian.
 - b. Menambahkan komponen penilaian manual untuk melengkapi hasil dari KA.
 - c. Menonaktifkan sistem saat ada protes dari siswa.
 - d. Mengalihkan seluruh proses belajar ke mesin KA.
 - e. Memberi nilai rata-rata jika hasil dari KA diragukan.

Perhatikan ilustrasi berikut dengan cermat.

Dalam penerimaan peserta didik baru, sebuah sekolah menggunakan sistem KA untuk menyaring data nilai dan lokasi domisili calon siswa. Setelah diterapkan, sistem justru menimbulkan ketimpangan penerimaan.

9. Apa prinsip etika yang dilanggar?
 - a. Transparansi dan keadilan dalam pengambilan keputusan otomatis.
 - b. Efisiensi dan otomatisasi sistem pendaftaran.
 - c. Privasi data siswa yang digunakan untuk pengumuman.
 - d. Akurasi dalam input data manual oleh operator.
 - e. Kecepatan proses dalam seleksi calon siswa.

Perhatikan ilustrasi berikut dengan cermat.

Sebuah KA dapat membuat konten berupa video palsu yang sangat realistis.

10. Kebijakan paling efektif untuk mitigasi risiko ini adalah
 - a. memblokir semua platform KA generatif
 - b. mewajibkan *watermark* konten buatan KA dan edukasi literasi digital
 - c. mengandalkan deteksi otomatis tanpa verifikasi manusia
 - d. menghukum pengguna yang menyebarkan konten tanpa bukti
 - e. membiarkan pasar mengatur sendiri perkembangan teknologi

B. Soal Esai

Jawablah pertanyaan berikut dengan tepat!

1. Menurutmu, mengapa beberapa pekerjaan lebih rentan digantikan KA daripada yang lain? Berikan contoh pekerjaan yang “aman” dan pekerjaan yang “terancam” KA beserta alasannya!
2. Bagaimana cara kamu mempersiapkan diri untuk dapat kompetitif di pasar kerja yang semakin terotomatisasi?
3. Berikan satu contoh konkret penerapan *human-centered* KA yang dapat meminimalisasi dampak negatifnya terhadap pekerja manusia?

Pengayaan

Untuk menambah wawasanmu tentang materi Bab 2 ini, kumpulkan semua tugasmu ke dalam dokumentasi *website*/blog pribadi. Sertakan sebuah pernyataan tentang bagaimana caramu menghadapi persaingan di pasar kerja pada era KA.



Refleksi

Setelah kamu mempelajari materi Bab 2 tentang “Dampak Kecerdasan Artifisial (KA) terhadap Ketenagakerjaan”, silakan kamu identifikasi apa saja yang telah kamu peroleh.

Pertanyaan:

1. Tuliskan 3 contoh teknologi KA yang sering kamu gunakan, bagaimana pengaruhnya terhadap aktivitasmu?
2. Apakah keterampilanmu saat ini yang relevan dengan era KA?
3. Bagaimana perasaanmu setelah pembelajaran Dampak Kecerdasan Artifisial (KA) terhadap Ketenagakerjaan?
4. Setelah melakukan pembelajaran “Dampak Kecerdasan Artifisial (KA) terhadap Ketenagakerjaan”, targetmu berikutnya adalah . . .
5. Tuliskan kendala dan kesulitan yang kalian hadapi selama mempelajari materi Bab 2 ini!

Bab 3

Pemrograman Berorientasi Objek



Jika semua hal di dunia dapat diwakili dengan objek dan kelas, lalu bagaimana cara program memahaminya?



Tujuan Pembelajaran

Setelah mempelajari materi ini, kamu diharapkan dapat menerapkan pemrograman berorientasi objek dalam mengembangkan sebuah perangkat lunak; memahami apa itu objek dan kelas; serta menggunakan bagian-bagian dalam objek, menyembunyikan data, menurunkan sifat dari satu kelas ke kelas lain, dan mengubah bentuk objek dalam menyelesaikan masalah.



Kata Kunci

pemrograman, objek, kelas, pewarisan, abstrak, polimorfisme



Peta Materi





Siap-Siap Belajar

Pada materi sebelumnya, kamu telah belajar tentang pemrograman, mulai dari algoritma, struktur data, dan juga pemrograman terstruktur. Apa sih sebenarnya pemrograman itu? Pemrograman pada dasarnya adalah proses berpikir untuk menyelesaikan suatu permasalahan ke dalam solusi berbasis komputer. Solusi yang dimaksud dituliskan dalam kode-kode program dan menjadi sebuah perangkat lunak. Dengan menggunakan pemrograman terstruktur, kamu telah mampu menulis kode program yang rapi dan logis. Nah, bagaimana jika permasalahannya lebih kompleks? Misalnya, robot yang dapat mengenali suara atau aplikasi yang dapat mengenali wajah? Jika hanya menggunakan pemrograman terstruktur, tentu akan kewalahan.

Kali ini, kamu akan mulai memasuki dunia Pemrograman Berorientasi Objek (selanjutnya disebut dengan OOP). Bayangkan kamu sedang membuat program, seperti sistem pengenalan wajah, mobil otonom, atau chatbot. Semua program tersebut membutuhkan struktur yang mampu meniru cara kerja dunia nyata. Dalam OOP, tidak hanya berpikir tentang langkah-langkah, tapi juga tentang “siapa” yang melakukan langkah-langkah tersebut dan bagaimana mereka saling berinteraksi.


Nah, OOP adalah jembatan penting sebelum menyelam lebih dalam ke dunia Kecerdasan Artifisial (KA) karena hampir semua pengembangan KA modern membutuhkan prinsip OOP. Jadi, siapkah kamu untuk mempelajari cara berpikir seperti seorang arsitek perangkat lunak?



Cek Pemahaman Awal

Jawablah pertanyaan-pertanyaan berikut secara jujur sesuai dengan pengetahuan dan pengalamanmu saat ini. Tidak perlu mencari jawaban dari buku atau internet. Tidak ada jawaban benar atau salah. Latihan ini hanya untuk membantu kamu dan guru melihat titik awal sebelum memulai pembelajaran.

1. Apa yang kamu ketahui tentang “objek” dalam dunia nyata?
Tulis pengertian atau contohnya menurut pemahamanmu!
2. Pernahkah kamu membuat program komputer?
 - a. Jika ya, sebutkan bahasa pemrogramannya!
 - b. Jika belum, ceritakanlah pengalamanmu ketika menyaksikan orang lain membuat program!



3. Menurutmu, apa tantangan terbesar saat menulis program komputer? Berilah tanda centang (✓) pada pernyataan yang paling tepat atau tuliskan pernyataan lainnya jika kamu memiliki pendapat yang berbeda!

- Menentukan langkah-langkah logika.
- Menulis kode dalam sintaks yang benar.
- Mengatur ulang bagian program.
- Menyusun ide menjadi solusi.
- Lainnya: _____

4. Perhatikan benda di sekitarmu (seperti sepeda, HP, buku, atau meja). Pilih salah satu benda tersebut, lalu tuliskan ciri-ciri atau fitur pentingnya (minimal 3 ciri)!

5. Bagaimana menurutmu, jika sebuah benda seperti “sepeda” bisa digambarkan dalam bentuk program? Apa saja yang perlu dimasukkan ke dalam programnya?

6. Apakah kamu pernah mendengar istilah-istilah berikut? Berilah tanda centang (✓) pada istilah yang pernah kamu dengar! (Kamu boleh memilih lebih dari satu.)

- Kelas (*Class*)
- Objek (*Object*)
- Pewarisan (*Inheritance*)
- Polimorfisme (*Polymorphism*)
- Enkapsulasi (*Encapsulation*)

Jika pernah, jelaskan salah satu istilah yang kamu ketahui!

7. Pilihlah pernyataan yang menurutmu paling mendekati tentang cara kerja program berbasis objek!

Berilah tanda centang (✓) pada pernyataan yang paling tepat!

- Program adalah urutan langkah dari awal sampai selesai.
- Program terdiri dari bagian-bagian kecil yang saling bekerja sama.
- Program hanya berisi perhitungan matematika.
- Program hanya bisa digunakan untuk membuat gim.



Dalam dunia pemrograman, terdapat banyak cara untuk membuat sebuah program. Salah satu cara yang keren dan banyak dipakai di dunia nyata adalah OOP (*Object Oriented Programming*) atau Pemrograman Berorientasi Objek. Pada Bab 3 ini, kamu akan mempelajari tentang dasar-dasar pemrograman berorientasi objek yang diimplementasikan dalam bahasa pemrograman Python.

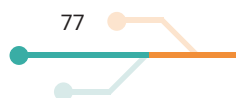
Python adalah salah satu bahasa yang populer dan mudah dipelajari karena sintaksisnya sederhana dan mirip dengan bahasa manusia. Kamu dapat menggunakan *Integrated Development Environment* (IDE), seperti Visual Studio Code, PyCharm, Jupyter Notebook/Lab, Spyder, IDLE, Google Colab, dan berbagai macam IDE lainnya, baik yang tersedia secara daring maupun luring (*desktop*). Kamu dapat menyesuaikannya dengan bahasa pemrograman lain yang mendukung OOP, seperti Java, C++, C#, dan sebagainya.

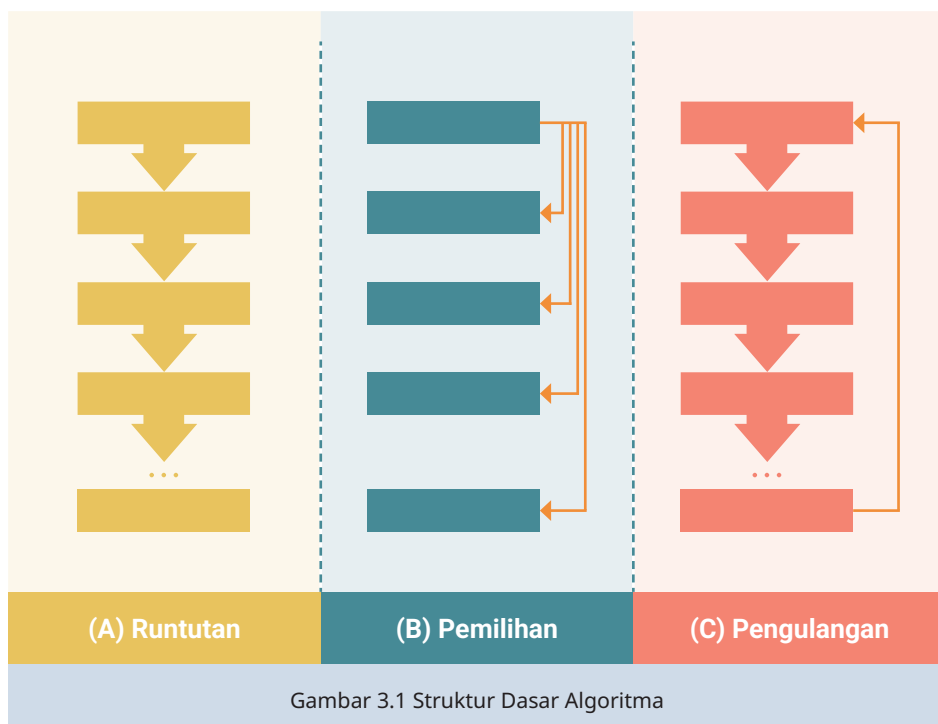
A. Pengantar Singkat Pemrograman

Sebelum kamu belajar lebih jauh tentang OOP, ayo kembali sejenak ke dasar pemrograman yang sudah kamu pelajari sebelumnya. Hal ini bertujuan untuk membantumu mengingatkan kembali materi tersebut agar kamu tidak kebingungan saat mengimplementasikan koding.

Kamu tidak bisa begitu saja menginstruksikan komputer untuk melakukan sesuatu tanpa memberi instruksi yang jelas dalam bahasa yang dimengerti oleh mesin. Manusia dan komputer berbicara dalam "bahasa" yang berbeda. Sama halnya seperti manusia yang berasal dari berbagai suku bangsa dengan bahasa yang beragam, mesin pun memiliki bahasanya sendiri, yang disebut dengan bahasa pemrograman.

Untuk membuat komputer dapat mengerti apa yang manusia inginkan, maka manusia perlu menuliskannya terlebih dahulu ke dalam bahasa pemrograman tertentu. Namun, tidak cukup hanya dengan menulis, perintah yang diberikan juga harus tersusun secara runtut dan sistematis. Mengapa? Karena komputer akan mengikuti langkah-langkah persis seperti yang dituliskan. Urutan langkah inilah yang disebut dengan algoritma. Algoritma merupakan serangkaian instruksi logis dan berurutan yang digunakan untuk menyelesaikan suatu masalah.

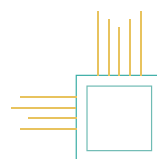




Dalam algoritma, terdapat tiga struktur dasar, yaitu runtutan, pemilihan, dan pengulangan (Munir dan Lidya, 2016). Berikut penjelasan tentang struktur dasar algoritma.

1. Runtutan bermakna bahwa program dikerjakan secara berurutan, dimulai pada baris pertama, lalu dilanjutkan ke baris kedua sampai dengan baris terakhir.
2. Pemilihan bermakna bahwa program dijalankan sesuai dengan keterpenuhan kriteria. Pada struktur ini akan dilakukan pengujian terlebih dahulu terhadap suatu kondisi yang dijadikan dasar. Apakah suatu baris program dieksekusi atau tidak.
3. Pengulangan, struktur ini memungkinkan suatu baris atau blok program dikerjakan lagi tanpa harus menuliskan ulang.

Nah, kini kamu sudah mengingatnya, bukan? Ayo, kita lanjutkan ke pembahasan berikutnya! Oleh karena materi ini akan diimplementasikan ke dalam bahasa Python, jadi tidak ada salahnya apabila kamu sedikit mundur untuk melihat lagi bagaimana strukturnya.



1. Struktur Dasar

Struktur dasar program Python biasanya terdiri dari:

Kode Program 3.1

```
01 # Baris komentar
02 def fungsi_utama():
03     # isi program
04     pass
05
06 if __name__ == "__main__":
07     fungsi_utama()
```

Hati-hati ya, Python menggunakan indentasi (spasi) untuk menunjukkan blok kode, bukan tanda kurung kurawal seperti Java atau C++. Python memiliki berbagai tipe data bawaan seperti berikut.

Tabel 3.1 Tipe Data di Python

Tipe Data	Contoh	Penjelasan
int	10	bilangan bulat
float	3.14	bilangan desimal
str	"halo"	teks
bool	True atau False	logika benar/salah
list	[1, 2, 3]	kumpulan data terurut
dict	{"nama": "Ayu"}	pasangan kunci-nilai

Variabel di Python dapat langsung dibuat tanpa menyebutkan tipe terlebih dahulu. Contohnya seperti berikut.

Kode Program 3.2

```
01 nama = "Andi"
02 umur = 17
03 tinggi = 165.5
```

2. Struktur Pengambilan Keputusan (Percabangan/Kondisi)

Python memungkinkan program dapat mengambil keputusan berdasarkan kondisi tertentu. Ada beberapa variasi penulisan yang bisa digunakan, di antaranya:

a. `if-else` (Dasar)

Kode Program 3.3

```
01 umur = 18
02 if umur >= 17:
03     print("Sudah boleh punya KTP")
04 else:
05     print("Belum cukup umur")
```

b. `if-elif-else` (Pilihan Bertingkat)

Variasi ini digunakan untuk lebih dari dua kemungkinan, yaitu:

Kode Program 3.4

```
01 nilai = 85
02
03 if nilai >= 90:
04     print("Nilai: A")
05 elif nilai >= 80:
06     print("Nilai: B")
07 elif nilai >= 70:
08     print("Nilai: C")
09 else:
10     print("Nilai: D")
```

c. `if` dalam satu baris (*one-liner if*)

Berikut bentuk ringkas ketika hanya satu aksi yang dilakukan.

Kode Program 3.5

```
01 suhu = 30
02 if suhu > 28: print("Cuaca panas")
```

- d. **Ternary Operator** (Kondisi dalam satu baris untuk memilih nilai.)

Variasi ini digunakan ketika ingin memilih nilai berdasarkan kondisi.

Kode Program 3.6

```
01 umur = 16
02 status = "Dewasa" if umur >= 17 else "Anak-anak"
03 print(status)
```

- e. Penggunaan **match-case** (Mulai Python 3.10+)

Alternatif **switch-case** dari bahasa lain. Variasi **match-case** sangat berguna untuk menggantikan banyak **if-elif**.

Kode Program 3.7

```
01 warna = "merah"
02
03 match warna:
04     case "merah":
05         print("Berhenti")
06     case "kuning":
07         print("Hati-hati")
08     case "hijau":
09         print("Jalan")
10     case _:
11         print("Warna tidak dikenal")
```

3. Struktur Pengulangan

Python mempunyai dua jenis utama, yaitu **for** dan **while**.

- a. Berikut contoh dari struktur pengulangan Python jenis **for**.

Kode Program 3.8

```
01 for i in range(5):
02     print(i)
```

- b. Berikut contoh dari struktur pengulangan Python jenis `while`.

Kode Program 3.9

```
01 angka = 0
02 while angka < 5:
03     print(angka)
04     angka += 1
```

4. Fungsi dan Prosedur

Fungsi adalah blok kode yang bisa dipanggil berkali-kali.

Kode Program 3.10

```
01 def sapa(nama):
02     print(f"Halo, {nama}!")
03
04 sapa("Ayu")
```

Fungsi bisa mengembalikan nilai dengan `return`. Fungsi yang tidak mengembalikan nilailah yang dinamakan prosedur.

Kode Program 3.11

```
01 def tambah(a, b):
02     return a + b
03
04 hasil = tambah(3, 5)
05 print(hasil)
```

Bagi kamu yang masih penasaran tentang Python, kamu dapat membuka kembali catatanmu di jenjang sebelumnya atau kamu dapat mempelajari lebih lanjut di beberapa situs yang mudah kamu pahami, seperti:

1. [W3Schools](#) (Pengantar Python yang menjelaskan tentang dasar-dasar Python dengan cara yang ringkas serta dilengkapi dengan contoh langsung yang dapat kamu coba. Materi ini sangat cocok untuk pemula).

2. BelajarPython.com (Apa Itu Python?, yaitu situs berbahasa Indonesia yang menjelaskan tentang pengertian dan fungsi Python serta alasan mengapa banyak orang menggunakannya)..

B. Mengapa OOP?

OOP memiliki banyak kelebihan dibandingkan cara lama yang disebut dengan pemrograman prosedural. Apabila dahulu membuat program dilakukan dengan alur langkah demi langkah, sekarang dapat membangun program dengan bagian-bagian kecil yang disebut dengan objek. Mengapa hal ini penting? Karena objek itu seperti representasi dari benda nyata. Misalnya, jika hendak membuat program tentang kendaraan, maka dapat membuat objek mobil, motor, atau sepeda yang masing-masing memiliki data dan perilaku sendiri. Jadi, program jadinya lebih mirip kenyataan dan lebih mudah dipahami.

OOP menawarkan pendekatan yang lebih modular, fleksibel, dan mudah dipelihara dibandingkan dengan pemrograman prosedural. Inti dari OOP adalah objek yang memadukan data dan kode, sehingga memungkinkan pemodelan dunia nyata secara akurat dan intuitif. Dengan membagi program menjadi objek-objek kecil, kode menjadi lebih mudah dipahami dan dikelola. OOP juga mengandalkan konsep penting, seperti enkapsulasi, pewarisan, dan polimorfisme yang mendukung penggunaan ulang kode dan pemeliharaan program secara lebih efisien.

Sebelum masuk ke penjelasan lebih dalam tentang konsep OOP, seperti kelas, objek, pewarisan, dan sebagainya, ayo mulai dulu dengan aktivitas pembelajaran agar kamu dapat lebih paham dan lebih semangat.



Aktivitas 3.1

Mengamati dan Mendeskripsikan Objek di Sekitar (Aktivitas Kelompok)

Mari deskripsikan benda-benda yang ada di sekitar. Deskripsi yang dibuat meliputi ciri-ciri yang dapat menggambarkan benda tersebut tanpa melihat wujud fisiknya. Ciri-ciri yang dimaksud juga termasuk perilaku atau kemampuan apa yang dapat dilakukan oleh benda tersebut. Sebagai contoh, perhatikan deskripsi papan tulis berikut.

Aspek	Keterangan
Bentuk	persegi panjang
Permukaan	halus, mudah ditulisi, dan mudah dihapus
Warna	hitam atau putih
Ukuran	beragam (menyesuaikan ruangan)
Bahan	kaca, kayu, logam, atau bahan lain yang dilapisi permukaan khusus
Fungsi Utama	sebagai media tulis sementara untuk menyampaikan informasi secara visual
Penggunaan	menulis, menggambar, mencatat ide, atau menjelaskan sesuatu
Alat Pendukung	spidol (untuk <i>whiteboard</i>), kapur (untuk <i>blackboard</i>), penghapus

Untuk mengerjakan aktivitas ini, ikutilah langkah-langkah berikut.

- a. Bentuklah kelompok yang terdiri atas 3–5 orang!
- b. Berdiskusilah dengan kelompokmu untuk mendeskripsikan benda-benda berikut (Ikuti contoh sebelumnya)!
 - 1) kursi dan sofa
 - 2) gelas dan cangkir
 - 3) sepeda dan sepeda motor
 - 4) kapal dan perahu
 - 5) buku dan novel
- c. Perhatikan baik-baik hasil deskripsi yang telah kalian buat. Kemudian, lakukan analisis terhadap ciri-ciri yang sudah dikumpulkan!
- d. Buatlah simpulan mengenai persamaan atau perbedaan ciri yang dimiliki oleh benda-benda tersebut! Manakah yang lebih banyak (persamaan atautkah perbedaan)?
- e. Presentasikan hasil identifikasi dan analisis kalian di hadapan guru dan kelompok lain!

Coba deh kamu lihat benda-benda di sekitarmu. Ada gelas, cangkir, sepeda, sepeda motor, buku, dan novel. Jika dilihat secara fisik, pasti benda-benda tersebut tampak berbeda, bukan? Tetapi, jika dilihat dari sifat atau fungsi dasarnya, ternyata banyak juga persamaannya, seperti gelas dan cangkir. Bentuk kedua benda tersebut memang berbeda, tetapi keduanya mempunyai fungsi yang sama, yaitu tempat untuk minum. Begitu juga dengan sepeda dan sepeda motor. Benda yang satu dikendarai dengan cara digowes, sedangkan benda yang satunya lagi dikendarai dengan menggunakan mesin. Namun, secara umum keduanya digunakan sebagai alat untuk berpindah tempat atau sama-sama disebut dengan kendaraan.

Nah, dari sinilah muncul ide penting dalam pemrograman, yaitu tentang abstraksi. Artinya, melihat sesuatu bukan dari bentuk fisiknya, tetapi dari sifat dan fungsinya. Hal inilah yang menjadi dasar dalam mengenal kelas dan objek dalam pemrograman. Berikut contoh pengenalan kelas dan objek dalam pemrograman.

1. Sepeda dan sepeda motor adalah objek dan keduanya termasuk ke dalam kelas yang sama, yaitu kendaraan.
2. Novel adalah contoh objek dari kelas buku, tetapi novel memiliki ciri yang lebih khusus, yaitu genre, jumlah halaman, dan sebagainya.

Jadi, dalam pemrograman kamu belajar mengelompokkan hal-hal yang mirip menjadi satu kelas, lalu membuat objek-objek dari kelas tersebut. Nah, apa itu sebenarnya kelas, objek, dan istilah-istilah lainnya? Yuk, kita bahas lebih lanjut pada materi berikutnya!

C. Kelas dan Objek

Dalam dunia pemrograman, objek merupakan entitas fundamental. Objek adalah bagian-bagian penyusun dari sebuah program. Objek bagaikan “benda” atau “tokoh” dalam kehidupan nyata. Objek memiliki ciri-ciri dan kemampuan tertentu. Setiap objek mempunyai identitas yang berbeda dan bersifat unik. Identitas pada objek disebut dengan atribut atau properti. Kemampuan yang dimiliki objek dinamakan metode. Atribut berbentuk data, sedangkan metode berupa fungsi.

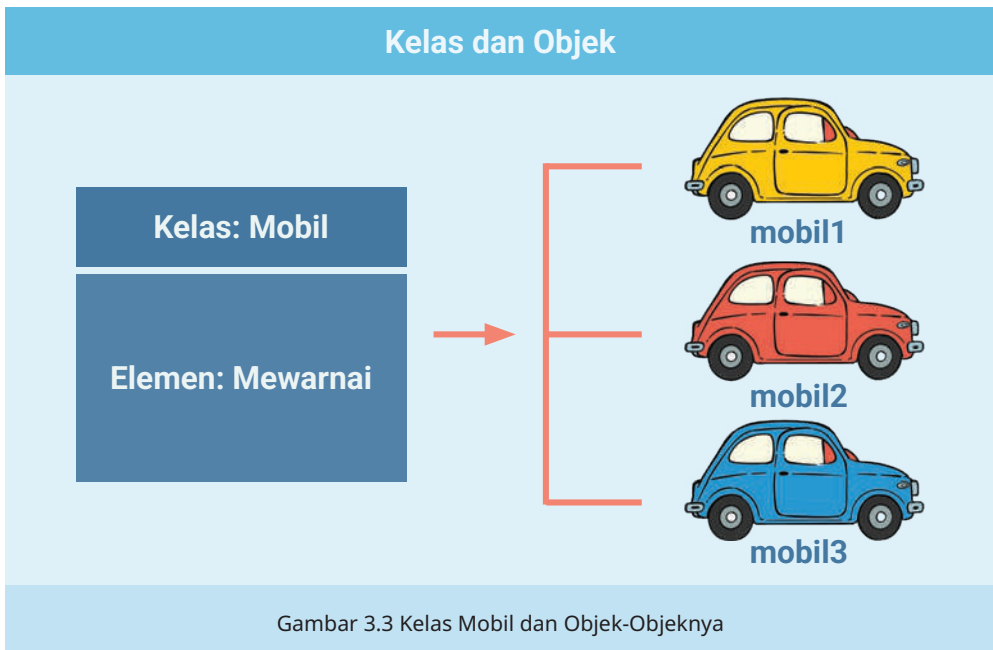
Sebagai contoh, jika kamu membuat sebuah aplikasi pengelolaan data siswa, maka dalam hal ini, siswa dianggap sebagai sebuah objek. Siswa memiliki identitas unik berupa nomor induk siswa, sifat-sifat (seperti nama, usia, kelas, dan alamat), serta perilaku, seperti belajar, mengerjakan tugas, dan mengikuti ujian.

Contoh lainnya adalah jika kamu membuat sebuah aplikasi tentang kendaraan, maka dalam hal ini mobil juga dapat diklasifikasikan sebagai sebuah objek. Mobil memiliki identitas unik, misalnya nomor polisi. Sifat-sifat yang dimiliki mobil dapat berupa merek, warna, jumlah roda, dan kecepatan. Mobil juga memiliki perilaku atau fungsi, seperti bergerak maju, berhenti, berbelok, dan mengerem.

Kelas diibaratkan sebagai cetak biru atau pola yang digunakan untuk membuat objek. Sebagai contoh, kamu mempunyai cetakan kue. Dari satu cetakan yang sama, kamu dapat membuat banyak kue dengan bentuk yang sama, tetapi *topping* atau warnanya berbeda. Kelas itu ibarat cetakan kuenya, sedangkan objek adalah kue-kue yang dihasilkan.



Dalam OOP, kelas merupakan rancangan awal atau definisi tentang apa saja yang akan dimiliki oleh objek (atribut + metode). Dalam dunia nyata, kelas bisa dianggap sebagai kategori atau jenis objek, seperti "mobil" yang kemudian menjadi objek mobil1, mobil2, mobil3. Semua mobil memiliki sifat dan kemampuan yang sama, tetapi nilai atributnya berbeda-beda.



Gambar 3.3 Kelas Mobil dan Objek-Objeknya

Dalam OOP, objek adalah hasil instansiasi dari kelas-kelas yang ada atau perwujudan dari kelas. Objek adalah elemen yang dapat diciptakan dimanipulasi dan dihancurkan.

Atribut adalah ciri atau sifat yang dimiliki oleh suatu objek yang menentukan keadaan atau kondisi objek tersebut. Kedua hal tersebut menjadi bagian penting dari definisi objek dan menggambarkan identitas serta ciri khas unik dari objek tersebut. Di dunia pemrograman, atribut juga disebut sebagai variabel atau *field* dan bisa menyimpan berbagai macam data, seperti angka, teks, tanggal, atau bahkan objek lain.

Setiap objek mempunyai nilai-nilai khusus untuk atributnya, sehingga membedakannya dari objek lain. Hal ini membuat objek mempunyai sifat unik yang membedakannya dari objek sejenis lainnya. Apabila digambarkan, atribut dapat diibaratkan sebagai kata sifat yang mendeskripsikan suatu objek, seperti warna, ukuran, atau bobotnya. Atribut-atribut tersebut menjadi komponen penting untuk menentukan identitas dan sifat khas dari objek di OOP.

Berikut adalah beberapa contoh atribut yang dapat dimiliki oleh objek-objek dalam kehidupan sehari-hari.

Tabel 3.2 Contoh Atribut

Objek	Atribut
Mobil	nomor polisi, merek, model, warna, tahun pembuatan, pelat nomor, kapasitas mesin, jarak tempuh
Sepeda	merek, jenis, ukuran ban, jumlah gigi, suspensi, warna
Buku	judul, penulis, penerbit, tahun terbit, ISBN, jumlah halaman
Siswa	NISN, NIS, nama lengkap, tempat lahir, tanggal lahir, jenis kelamin, alamat, nomor telepon
Rumah	alamat, luas tanah, luas bangunan, jumlah kamar, jumlah lantai, tipe bangunan, fasilitas
Laptop	merek, model, sistem operasi, prosesor, RAM, kapasitas penyimpanan, ukuran layar, bobot, kapasitas baterai
Telepon Pintar	merek, model, warna, sistem operasi, ukuran layar, kapasitas penyimpanan, kamera, fitur konektivitas, kapasitas baterai

Atribut dari objek dapat juga berisi informasi tentang keadaan dari objek itu sendiri. Misalnya, pada mobil, atribut dapat didefinisikan untuk kondisi mesin (dalam keadaan hidup atau mati) maupun posisi gigi. Atribut dapat dianalogikan sebagai variabel global yang dimiliki oleh objek dan dapat diisi dengan nilai yang berbeda-beda ketika digunakan.

Metode adalah fungsi atau prosedur yang terkait dengan suatu objek dan mendefinisikan perilaku atau aksi yang dapat dilakukan oleh objek tersebut. Perilaku dalam objek merujuk pada kemampuan objek untuk melakukan sesuatu terhadap dirinya sendiri. Misalnya, dalam objek mobil terdapat berbagai perilaku, seperti: menghidupkan mesin, melaju ke depan, mundur, berbelok, berhenti, dan berpindah gigi. Metode dalam OOP dapat dikategorikan dalam beberapa bentuk berikut.

1. Assessor/Getter

Metode yang digunakan untuk membaca atau mengambil nilai atribut yang dimiliki oleh objek. Umumnya diawali dengan kata kunci `get` dan diikuti dengan nama atribut.

Contoh: kelas sepeda memiliki atribut merek, jenis, ukuran ban, jumlah gigi, suspensi, dan warna. Maka, dapat dibuat metode: `get_merk()`, `get_jenis()`, `get_ukuran_ban()`, `get_jumlah_gigi()`, `get_suspensi()`, `get_warna()`.

2. Mutator/Setter

Metode yang digunakan untuk mengatur nilai atribut yang dimiliki. Umumnya diawali dengan kata kunci `set` dan diikuti dengan nama serta parameter *input*-nya.

Contoh: pada kelas sepeda, dapat dibuat metode: `set_merk(String)`, `set_jenis(String)`, `set_ukuran_ban(int)`, `set_jumlah_gigi(int)`, `set_suspensi(String)`, `set_warna(String)`.

3. Konstruktor

Metode khusus yang dipanggil secara otomatis pada saat sebuah objek diciptakan (instansiasi). Saat objek diciptakan, konstruktor akan mengalokasikan memori, memberi nilai awal pada data yang dimiliki objek, dan secara otomatis menjalankan semua pernyataan yang menjadi isinya.

```
class NamaKelas:
    def __init__(self, parameter1, parameter2):
        self.atribut1 = parameter1
        self.atribut2 = parameter2
```

- `__init__` adalah metode khusus.
- `self` adalah referensi ke objek itu sendiri.
- Atribut ditetapkan menggunakan `self.nama_atribut`.

4. Destruktor

Metode ini digunakan untuk membersihkan atau melepaskan sumber daya yang digunakan ketika sebuah objek diciptakan dan digunakan. Hal ini penting untuk mencegah kebocoran memori. Dalam bahasa pemrograman Python, metode destruktur disebut dengan `__del__()` dan dijalankan saat objek akan dihapus dari memori.

```
class NamaKelas:
    def __del__(self):
        print("Objek sedang dihancurkan ...")
```

5. Metode Lainnya

Metode selain dari yang sudah dijelaskan di atas yang digunakan atau menunjukkan kemampuan dari kelas tersebut. Contoh metode lain yang bisa dibuat dalam kelas sepeda: `tambahGigi()`, `kurangiGigi()`. Berikut contoh pendefinisian beberapa kelas yang memiliki atribut dan metode.

Tabel 3.3 Contoh Pendefinisian Kelas

No.	Objek	Atribut	Metode
1	Mobil	nomor polisi, merek, model, warna, tahun pembuatan, kapasitas mesin, jarak tempuh	<code>get_Merk(); set_Merk(); get_Model(); set_Model(); get_Warna(); set_Warna(); get_JarakTempuh(); set_JarakTempuh(); hidupkanMesin(); matikanMesin(); percepat(); rem();</code>
2	Sepeda	merek, jenis, ukuran ban, jumlah gigi, suspensi, warna	<code>get_Merk(); set_Merk(); get_Jenis(); set_Jenis(); get_UkuranBan(); set_UkuranBan(); tambahGigi(); kurangGigi(); aktivasiSuspensi(); nonaktifkanSuspensi();</code>
3	Buku	judul, penulis, penerbit, tahun terbit, ISBN, jumlah halaman	<code>get_Judul(); set_Judul(); get_Penulis(); set_Penulis(); get_Penerbit(); set_Penerbit(); get_TahunTerbit(); set_TahunTerbit(); get_ISBN(); set_ISBN(); get_JumlahHalaman(); set_JumlahHalaman();</code>

No.	Objek	Atribut	Metode
4	Siswa	NISN, NIS, nama lengkap, tempat lahir, tanggal lahir, jenis kelamin, alamat, nomor telepon	<pre> get_NISN(); set_NISN(); get_NIS(); set_NIS(); get>NamaLengkap(); set>NamaLengkap(); get_TmptLahir(); set_TmptLahir(); get_TglLahir(); set_TglLahir(); get_JenisKelamin(); set_JenisKelamin(); get_Alamat(); set_Alamat(); get_NoTelp(); set_NoTelp(); </pre>
5	Rumah	alamat, luas tanah, luas bangunan, jumlah kamar, jumlah lantai, tipe bangunan, fasilitas	<pre> get_Alamat(); set_Alamat(); get_LuasTanah(); set_LuasTanah(); get_LuasBangunan(); set_LuasBangunan(); get_JumlahKamar(); set_JumlahKamar(); get_JumlahLantai(); set_JumlahLantai(); </pre>
6	Laptop	merek, model, sistem operasi, prosesor, RAM, kapasitas penyimpanan, ukuran layar, bobot, kapasitas baterai	<pre> get_Merk(); set_Merk(); get_Model(); set_Model(); get_SistemOperasi(); set_SistemOperasi(); get_Prosesor(); set_Prosesor(); get_RAM(); set_RAM(); get_KapasitasPenyimpanan(); set_KapasitasPenyimpanan(); get_UkuranLayar(); set_UkuranLayar(); get_Bobot(); set_Bobot(); get_DayaTahanBaterai(); set_DayaTahanBaterai(); </pre>

No.	Objek	Atribut	Metode
7	Telepon Pintar	merek, model, warna, sistem operasi, ukuran layar, kapasitas penyimpanan, kamera, fitur konektivitas, kapasitas baterai	<pre> get_Merk(); set_Merk(); get_Model(); set_Model(); get_Warna(); set_Warna(); get_SistemOperasi(); set_SistemOperasi(); get_UkuranLayar(); set_UkuranLayar(); get_KapasitasPenyimpanan(); set_KapasitasPenyimpanan(); get_Kamera(); set_Kamera(); get_FiturKonektivitas(); set_FiturKonektivitas(); get_KapasitasBaterai(); set_KapasitasBaterai(); </pre>

Untuk memahami lebih lanjut tentang perbedaan kelas dan objek, perhatikan contoh berikut.

Tabel 3.4 Kelas Mobil dan Objek-Objeknya

Kelas: Mobil	Mobil1	Mobil2	Mobil3	
Atribut	Nomor Polisi	BL1234XY	B1234ABC	PA1234AB
	Merek	Honda	Toyota	Mazda
	Model	Jazz	Yaris	Mazda2
	Warna	Merah	Kuning	Hijau
	Tahun Pembuatan	2021	2022	2020
	Kapasitas Mesin	1.500 cc	1.496 cc	1.496 cc
	Jarak Tempuh	1:12	1:13	1:13,8

Kelas: Mobil	Mobil1	Mobil2	Mobil3
Metode			

```

get_Merk()
set_Merk()
get_Model()
set_Model()
get_Warna()
set_Warna()
get_JarakTempuh()
set_JarakTempuh()
hidupkanMesin()
matikanMesin()
percepat()
rem()

```

Pada tabel 3.4, dapat dilihat bahwa Mobil1, Mobil2, dan Mobil3 adalah objek-objek yang diciptakan dari kelas Mobil. Ketiganya memiliki atribut yang sama, yaitu nomor polisi, merek, model, warna, tahun pembuatan, kapasitas mesin, dan jarak tempuh serta memiliki metode yang sama, yaitu: `get_Merk()`; `set_Merk()`; `get_Model()`; `set_Model()`; `get_Warna()`; `set_Warna()`; `get_JarakTempuh()`; `set_JarakTempuh()`; `hidupkanMesin()`; `matikanMesin()`; `percepat()`; `rem()`; yang didapat dari kelasnya.



Aktivitas 3.2

Mengklasifikasikan Objek Menjadi Kelas (Aktivitas Individu)

Aktivitas 3.2 ini merupakan bentuk latihan bagimu dalam mengklasifikasikan objek menjadi kelas. Untuk memudahkanmu dalam mengerjakan aktivitas ini, ikuti petunjuk berikut.

- Perhatikan sekitar ruang kelasmu. Objek apa saja yang dapat kamu temukan? Tuliskan objek-objek tersebut dalam format berikut..

No.	Nama Objek	Atribut	Metode
1	Meja Belajar	panjang, lebar, bahan, ...	diBersihkan(), diCat(), ...
2
3

- b. Kelompokkan objek-objek yang sudah kamu identifikasi tersebut berdasarkan kemiripan atribut dan metode. Kemudian, tentukan kelas dari masing-masing objek tersebut. Gunakan format berikut.

No.	Objek	Kelas
1	Meja Belajar	...
2
3

D. Menuliskan Kode Program

Setelah kamu mempelajari tentang apa itu kelas, objek, atribut, dan metode, sekarang waktunya kamu menerapkan semua unsur tersebut ke dalam bentuk kode program. Ingat ya, kelas adalah cetak biru dari objek. Dalam bahasa pemrograman, kelas ditulis dengan kata kunci *class*. Kelas mirip seperti sebuah tipe data baru. Namun, tipe data hanya menyimpan data saja, sedangkan kelas mengandung data dan fungsi untuk memproses data tersebut. Berikut bentuk umum dari kelas.

```
class NamaKelas:
    ... atribut
    ... konstruktor
    ... assessor
    ... mutator
```

Untuk contoh sederhananya, yuk kita buat sebuah kelas segi empat dengan rincian berikut.

Nama Kelas : SegiEmpat
Atribut : panjang, lebar
Metode : asesor; mutator; konstruktor; `hitung_Luas()`;
`hitung_Keliling()`

Berdasarkan rincian tersebut, seperti inilah implementasinya dalam bahasa Python.

Kode Program 3.12 Implementasi Kelas Segi Empat

```
01 class SegiEmpat:
02     # Atribut
03     def __init__(self, p, l):
04         self.__panjang = p
05         self.__lebar = l
06
07     # Mutator
08     def set_panjang(self, p):
09         self.__panjang = p
10
11     def set_lebar(self, l):
12         self.__lebar = l
13
14     # Asesor
15     def get_panjang(self):
16         return self.__panjang
17
18     def get_lebar(self):
19         return self.__lebar
20
21     # Metode assignSegiEmpat
22     def assignSegiEmpat(self, p, l):
23         self.__panjang = p
24         self.__lebar = l
25
26     # Metode hitung luas
```

```

27     def hitung_luas(self):
28         return self.__panjang * self.__lebar
29
30     # Metode hitung keliling
31     def hitung_keliling(self):
32         return 2 * (self.__panjang + self.__lebar)

```

Setelah kelas dibuat, apa yang harus kamu lakukan agar kelas dapat digunakan? Selanjutnya, kamu perlu melakukan instansiasi. Objek adalah *instance* dari kelas dan memiliki nilai-nilai tertentu untuk atribut-atribut yang didefinisikan dalam kelas. Instansiasi objek biasanya dilakukan pada program utama, tetapi dapat juga dilakukan dalam kelas itu sendiri maupun kelas lain. Pada Python, kamu dapat melakukannya dengan memanggil nama kelas seperti memanggil fungsi.

Kode Program 3.13 Instansi Objek dari Kelas Siswa

```

01 class Siswa:
02     def __init__(self, nama, kelas):
03         self.nama = nama
04         self.kelas = kelas
05
06     # Instansiasi objek
07     s1 = Siswa("Budi", "XI IPA 1")
08     s2 = Siswa("Ani", "XI IPS 2")
09
10     print(s1.nama) # Output: Budi
11     print(s2.kelas) # Output: XI IPS 2

```

Pada kode program di atas, `s1` dan `s2` adalah nama objek dan ketika menuliskan `s1 = Siswa("Budi", "XI IPA 1")`, maka proses instansiasi sudah dilakukan.

Dari kelas `Segi Empat` tersebut dapat dibuatkan instansiasi berikut.

```
s4Ku = SegiEmpat(0,0)
```

atau

```
s4Ku.assignSegiEmpat(4, 5)
```

Setelah dilakukan instansiasi, objek telah memiliki semua atribut dan layanan yang dimiliki oleh kelas. Atribut maupun kelas dapat dipanggil sesuai kebutuhan. Dalam contoh di atas, objek “`s4Ku`” telah dideklarasikan sebagai objek dari kelas “`SegiEmpat`”, berarti “`s4Ku`” memiliki atribut panjang dan lebar serta metode-metode yang ada, seperti `assignSegiEmpat(int, int)`, `hitung_luas()`, `hitung_keliling()`. Nah, sekarang bagaimana menggunakannya?

Perhatikan contoh kode program berikut.

Kode Program 3.14 Contoh Penggunaan Objek dalam Program Utama

```
01 if __name__ == "__main__":
02     s4Ku = SegiEmpat(0, 0)
03     s4Ku.assignSegiEmpat(4, 5)
04     print("Segi Empatku")
05     print("Luas Segi Empat = ", s4Ku.hitung_luas())
06     print("Keliling Segi Empat = ", s4Ku.hitung_keliling())
```

Perhatikan contoh kode program 3.13, mulai dari baris 7 sampai 11. Untuk memanggil anggota kelas (atribut atau metode), maka ditulis:

```
Objek.anggotakelas
```

Contoh untuk memanggil metode `hitung_luas()` dan `hitung_keliling()` pada kelas “`SegiEmpat`”

```
s4Ku.hitung_luas()
s4Ku.hitung_keliling()
```



Aktivitas 3.3

Implementasi Kelas dan Objek dalam Kode Program (Aktivitas Individu)

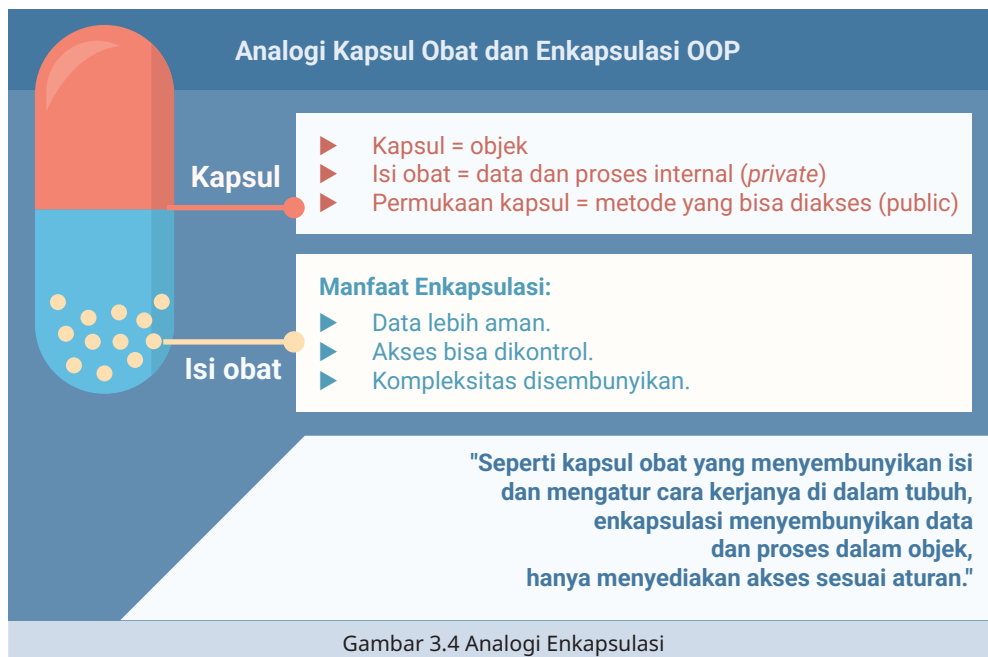
Lakukan implementasi contoh kode dalam bahasa pemrograman, gunakan IDE yang kamu kenal dengan langkah-langkah berikut.

- Buatlah kelas baru dengan nama SegiEmpat, salin kode program pada contoh kode program 3.12!
- Tambahkan program utama, salin kode program 3.14!
- Jalankan program utama, amati hasil luarannya!

E. Enkapsulasi

Coba bayangkan, di saat kamu sedang sakit, lalu kamu meminum obat kapsul. Kamu tidak perlu mengetahui secara detail apa saja isi di dalamnya maupun bagaimana cara kerjanya, bukan? Yang penting, begitu ditelan, obat kapsul tersebut bekerja sesuai fungsinya untuk menyembuhkan penyakit.

Nah, dalam OOP juga diterapkan konsep serupa, yaitu dinamakan dengan enkapsulasi. Data dan proses penting disimpan rapat di dalam objek, seperti isi obat dalam kapsul. Cukup gunakan metode yang disediakan tanpa harus mengerti cara kerjanya.



Enkapsulasi merupakan suatu konsep dalam OOP yang menekankan pada pembungkusan data (atribut) dan fungsi (metode) dalam satu unit. Enkapsulasi memungkinkan data atau fungsi diatur sedemikian rupa hingga objek yang lain tidak perlu mengetahui cara kerja objek tersebut, tetapi dapat memanfaatkan apa yang dimiliki oleh objek. Dengan cara ini, data di dalam objek tidak bisa sembarangan diubah atau diakses langsung dari luar, melainkan harus melalui metode yang sudah diatur. Untuk lebih memahami penjelasan tersebut, perhatikan beberapa contoh enkapsulasi berikut.

1. Remot TV

Saat menggunakan remot, kamu hanya berinteraksi dengan tombol-tombol yang memiliki berbagai fungsi. Kamu tidak perlu mengetahui rangkaian elektronika yang ada di dalamnya, seperti apa serta bagaimana cara kerjanya. Ketika tombol ditekan, maka akan bekerja sesuai dengan fungsi tombol tersebut. Proses rumit dalam remot disembunyikan (*private*) dan hanya dapat diakses melalui tombol (*public*).

2. Smartphone

Kamu tentu sudah biasa menggunakan *smartphone* setiap hari. Waktu kamu menekan tombol kamera, kamu tidak perlu mengetahui bagaimana proses internalnya, seperti bagaimana sensor bekerja, lensa mengatur fokus, *file* disimpan di memori, dan sebagainya. Yang kamu tahu hanya hasil akhirnya, yaitu foto muncul di galeri. Proses rumit di dalam *smartphone* tersebut disembunyikan (*private*) dan kamu hanya bisa mengakses fitur melalui antarmuka aplikasi (*public*).

3. Mobil

Saat menginjak pedal gas, mobil akan bergerak lebih cepat. Kamu tidak perlu mengetahui detail cara kerja mesin, pembakaran bahan bakar, atau putaran roda. Mesin dan sistem internal tersembunyi dari pengemudi, kamu hanya berinteraksi melalui pedal, setir, dan tuas (*public interface*).

4. Mesin ATM

Kamu hanya cukup memasukkan kartu, memilih nominal, lalu menarik uang. Kamu tidak perlu mengetahui proses perhitungan saldo, keamanan, atau cara mesin mengeluarkan uang. Proses internal (validasi kartu, hitung saldo, akses *database*) disembunyikan, kamu hanya dapat mengakses fitur via layar dan tombol.

Dari beberapa contoh tersebut terdapat pola yang sama, yaitu:

- Data/proses internal tersembunyi;
- Akses hanya melalui fitur/alat tertentu; dan
- Tidak perlu mengetahui detail di dalamnya, cukup interaksi dari luar sesuai aturan.

Dalam OOP, konsep enkapsulasi dilakukan pada *Access Modifier*, yaitu fitur yang digunakan untuk mengatur tingkat akses terhadap atribut dan metode dalam sebuah kelas. Dengan *Access Modifier*, kamu dapat menentukan bagian mana saja dari kelas yang boleh diakses dari luar dan mana yang harus disembunyikan. Secara umum, jenis *Access Modifier* yang tersedia adalah:

1. *Private*

Atribut dan metode yang dinyatakan *private* hanya dapat diakses oleh metode di dalam kelasnya sendiri dan tidak dapat diakses oleh kelas yang lain. Dalam Python, *private* ditandai dengan dua garis bawah (`__nama`).

2. *Protected*

Atribut dan metode dapat diakses oleh subkelasnya atau turunannya. Dalam Python, *protected* ditandai dengan satu garis bawah (`_nama`).

3. *Public*

Atribut dan metode yang dideklarasikan sebagai *public* dapat diakses oleh semua kelas di semua bagian dari *project*. Dalam Python, *public* tidak diberikan penanda (`nama`).

Pada umumnya, atribut akan dinyatakan *private* untuk menjaga keamanan data, sedangkan metode dinyatakan *public* agar objek lain dapat berinteraksi dengan data secara terkendali melalui metode tersebut. Contoh penggunaan enkapsulasi dapat dilihat pada kode program 3.15 berikut.

Kode Program 3.15 Contoh Enkapsulasi pada Kelas Siswa

```
01 class Siswa:
02     def __init__(self, nama, nilai):
03         self.__nama = nama # atribut private
```

```

04     self.__nilai = nilai # atribut private
05
06     # Getter public
07     def get_nilai(self):
08         return self.__nilai
09
10     # Setter public
11     def set_nilai(self, nilai_baru):
12         if 0 <= nilai_baru <= 100:
13             self.__nilai = nilai_baru
14         else:
15             print("Nilai tidak valid!")
16
17     def info(self):
18         print(f>Nama: {self.__nama}, Nilai: {self.__nilai}")

```

Berdasarkan contoh di atas, dapat dilihat bahwa:

1. Atribut nama disembunyikan (*private*);
2. Mengatur dan mengambil nilai nama menggunakan "*public*"; dan
3. Proses internal tidak dapat diakses langsung dari luar objek, hanya melalui metode yang sudah ditentukan (*public*).



Aktivitas 3.4

Menguji Enkapsulasi dalam Python (Aktivitas Individu)

Ikutilah petunjuk berikut untuk mengerjakan Aktivitas 3.4 ini.

- a. Salin kode program 3.15, lalu tambahkan kode program berikut!

Kode Program 3.16

```

20 if __name__ == "__main__":
21     s1 = Siswa("Ayu", 85)
22     # Akses melalui getter
23     print(s1.get_nilai()) # 85
24
25     # Ubah nilai dengan setter (terkontrol)

```

```
26     s1.set_nilai(95)
27     s1.set_nilai(150)
28
29     # Akses langsung
30     print(s1.__nilai)
```

- b. Amati keluarannya! Apa yang terjadi? Mengapa hal itu bisa terjadi?
- c. Ubahlah baris 30 menjadi komentar, lalu jalankan kembali! Bagaimana hasilnya sekarang?

F. Pewarisan dan Polimorfisme

Sebelum kamu mempelajari materi ini, silakan kamu kerjakan aktivitas pembelajaran berikut terlebih dahulu untuk menambah semangat belajarmu.



Aktivitas 3.5

Menemukan Pola Pewarisan dari Dunia Nyata (Aktivitas Individu)

Ikuti petunjuk berikut untuk mengerjakan aktivitas ini.

- a. Deskripsikan bangun berikut!

- 1) prisma segitiga, balok, tabung
- 2) limas segi empat, kerucut

Perhatikan hasil deskripsimu! Apa kesamaan ciri yang dimiliki oleh masing-masing kelompok?

- b. Tuliskan sifat-sifat yang dimiliki oleh objek berikut.

- 1) makhluk
- 2) hewan
- 3) kuda
- 4) burung
 - Apakah sifat yang dimiliki oleh (1) dimiliki juga oleh (2)?
 - Apakah sifat yang dimiliki oleh (2) dimiliki juga oleh (3) dan (4)?

- c. Perhatikan silsilah dalam keluargamu! Apakah ada ciri-ciri fisik ayah atau ibumu yang kamu miliki? Sebutkan ciri-ciri fisik tersebut!

1. Pewarisan

Ciri utama dari pemrograman berorientasi objek (OOP) adalah pewarisan atau bahasa kerennya, *inheritance*. Nah, pewarisan ini memungkinkan sebuah kelas untuk mewariskan sifat-sifatnya kepada kelas lain, baik berupa atribut (data) maupun metode (fungsi).

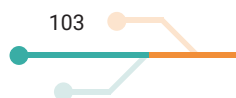
Kamu dapat membayangkan seperti ini: Kamu mempunyai kelas **Hewan**, lalu kamu ingin membuat kelas **Kucing**. Kamu tidak perlu menulis ulang semua sifat yang dimiliki hewan. Kamu hanya cukup menuliskan **Kucing** adalah turunan **Hewan** dan otomatis semua sifat hewan bakal dimiliki.

Kelas yang mewariskan sifatnya dinamakan sebagai kelas induk atau kelas dasar atau **superclass**. Sedangkan, kelas yang menerima sifat-sifat yang diturunkan disebut sebagai kelas turunan atau **subclass**. Pada dasarnya, hampir semua sifat yang ada pada kelas induk dapat diturunkan, kecuali yang bersifat “*private*”. Jadi, kelas turunan dapat memanfaatkan dan memperluas fungsionalitas yang telah ada pada kelas induk, sehingga dapat meningkatkan efisiensi dan kemandirian dalam pengembangan perangkat lunak. Dengan cara ini, kamu bisa menghemat waktu, mengurangi pengulangan kode, dan membuat programmu menjadi lebih modular dan rapi. Hal ini juga sangat membantu jika kita ingin membuat sistem yang besar dan kompleks, tetapi tetap mudah dikelola.

Coba kamu bayangkan, misalnya kamu memiliki kakek yang mempunyai resep rahasia untuk membuat kue lezat. Apakah kamu harus membuat resep baru jika ingin membuat kue lezat? Tentu kamu tidak perlu membuat dari awal lagi resep kue tersebut, tetapi hanya cukup menyesuaikannya saja dengan kebutuhan.

Kakekmu dapat diibaratkan sebagai “kelas induk”, sedangkan anak dan cucunya merupakan “kelas turunan”. Dengan adanya hubungan keluarga tersebut, maka anak dan cucunya secara otomatis mewarisi resep dari sang kakek. Keturunannya dapat memodifikasi atau menyesuaikan resep tersebut sesuai dengan preferensi mereka sendiri atau bahkan menambahkan bahan-bahan baru untuk menciptakan variasi kue yang unik.

Dalam pewarisan, terlebih dahulu harus didefinisikan kelas induknya, lalu disusul dengan kelas turunannya. Untuk mendefinisikan kelas turunan,



gunakan kata kunci *extends* sesudah nama kelas dan ikuti dengan nama kelas induknya. Secara umum, definisi tersebut dapat dituliskan seperti berikut.

```
class KelasTurunan(KelasInduk)
```

Berikut contoh penggunaan pewarisan.

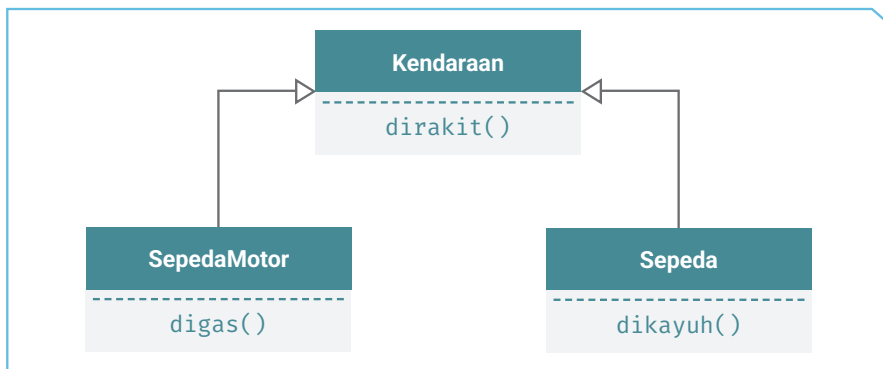
```
class Hewan:
    def suara(self):
        print("Hewan bersuara")

class Kucing(Hewan): # Pewarisan
    def suara(self): # Override metode
        print("Meong~")
```

Secara umum, mekanisme pewarisan ini dikelompokkan menjadi dua, yaitu pewarisan tunggal dan pewarisan majemuk. Berikut penjelasan lebih detail mengenai kedua mekanisme pewarisan.

a. Pewarisan Tunggal (*Single Inheritance*)

Pewarisan tunggal merupakan bentuk pewarisan yang paling sederhana dan mudah dipahami. Pewarisan hanya berasal dari satu kelas induk yang memberikan atribut dan metode yang dapat diwariskan. Meskipun kelas induknya hanya satu, tetapi dapat membentuk beberapa kelas turunan. Kelas turunan ini dapat memiliki sifat-sifat yang diwarisi dari kelas induk serta menambahkan sifat-sifat baru yang spesifik untuk kelas turunan tersebut. Perhatikan gambar 3.5 berikut.



Gambar 3.5 Contoh Pewarisan Tunggal

Pada gambar 3.5, kamu bisa melihat kelas kendaraan sebagai kelas induk yang memiliki dua kelas turunan, yaitu sepeda motor dan sepeda. Kelas sepeda motor dan sepeda mewarisi sifat-sifat atau atribut dari kelas kendaraan, yaitu `dirakit()` yang diwariskan kepada kedua kelas turunan tersebut.

Pada kelas sepeda motor, selain memiliki sifat `digas()` yang khusus dimiliki oleh sepeda motor, terdapat juga sifat `dirakit()` yang merupakan sifat yang diwarisi dari kelas kendaraan. Hal yang sama berlaku pada kelas sepeda. Meskipun kelas sepeda memiliki sifat `dikayuh()`, tetapi juga mewarisi sifat `dirakit()` dari kendaraan. Perhatikan perbandingannya pada tabel 3.5 berikut.

Tabel 3.5 Perbandingan Metode yang Dimiliki Kelas

	dirakit()	digas()	dikayuh()
Kendaraan	✓	-	-
Sepeda Motor	✓	✓	-
Sepeda	✓	-	✓



Aktivitas 3.6

Menerapkan Pewarisan Tunggal dalam Kode Program (Aktivitas Individu)

Kali ini, kamu akan coba mempraktikkan langsung mekanisme pewarisan tunggal! Ikuti langkah-langkah berikut untuk mengerjakan aktivitas ini. Perhatikan juga Gambar 3.5.

- Definisikan kelas induk “Kendaraan” seperti pada kode program 3.17 berikut!

Kode Program 3.17

Kelas Induk Sepeda

```
01 class Kendaraan:
02     # Konstruktor untuk inisialisasi atribut merk dan tipe
03     def __init__(self, merk, tipe):
```

```

04     # Atribut untuk menyimpan merk dan tipe kendaraan
05         self.merk = merk
06         self.tipe = tipe
07
08     # Metode untuk kendaraan sedang dirakit
09     def dirakit(self):
10         print(f"{self.merk} {self.tipe} sedang dirakit.")
11

```

- b. Buatlah kelas “Sepeda Motor” yang merupakan turunan dari “Kendaraan” dengan tambahan atribut cc dan metode `digas()` seperti pada kode program 3.18 berikut!

Kode Program 3.18 Kelas Sepeda Motor Diturunkan dari Kendaraan

```

12 class SepedaMotor(Kendaraan):
13     # Konstruktor untuk SepedaMotor, memanggil konstruktor
14     # kelas dasar dan menambahkan atribut cc
15     def __init__(self, merk, tipe, cc):
16         # Memanggil konstruktor kelas Kendaraan
17         super().__init__(merk, tipe)
18         # Atribut untuk menyimpan (cc) sepeda motor
19         self.cc = cc
20
21     # Metode untuk sepeda motor sedang digas
22     def digas(self):
23         print(f"{self.merk} {self.tipe} {self.cc}cc sedang “,
24             “digas.”)
25

```

- c. Buatlah kelas “Sepeda” yang merupakan turunan dari “Kendaraan” dengan tambahan atribut `jenis_medan` dan metode `dikayuh()` seperti pada kode program 3.19 berikut!

Kode Program 3.19**Kelas Sepeda Diturunkan dari Kendaraan**

```
26 class Sepeda(Kendaraan):
27     # Konstruktor untuk Sepeda,
28     # memanggil konstruktor kelas dasar dan
29     # menambahkan atribut JenisMedan
30     def __init__(self, merk, tipe, jenis_medan):
31         # Memanggil konstruktor kelas Kendaraan
32         super().__init__(merk, tipe)
33         # Atribut untuk menyimpan jenis medan sepeda
34         self.jenis_medan = jenis_medan
35
36     # Metode untuk menunjukkan bahwa sepeda sedang dikayuh
37     def dikayuh(self):
38         print(f"{self.merk} {self.tipe} untuk medan “,
39               “{self.jenis_medan} sedang dikayuh.”)
40
```

- d. Buatlah fungsi utama untuk menjalankan kelas Sepeda Motor dan Sepeda seperti pada kode program 3.20 berikut!

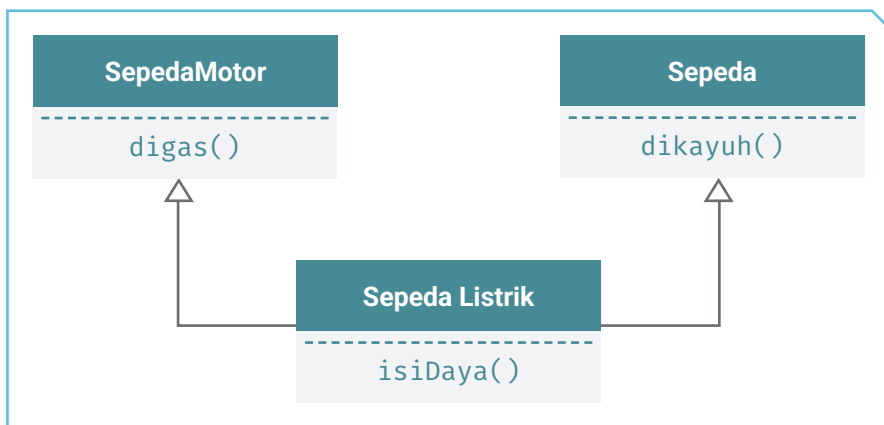
Kode Program 3.20**Fungsi Utama Menjalankan Pewarisan**

```
41 if __name__ == "__main__":
42     # Membuat objek SepedaMotor
43     motor_saya = SepedaMotor("Honda", "Beat", 110)
44     motor_saya.dirakit()
45     motor_saya.digas()
46
47     print("-" * 20) # Garis pemisah
48
49     # Membuat objek Sepeda
50     sepeda_saya = Sepeda("Polygon", "Strattos",
51                           "jalan raya")
52     sepeda_saya.dirakit()
53     sepeda_saya.dikayuh()
```

- e. Setelah digabungkan semua (baris 01–53), jalankan kode program di atas dan amati bagaimana keluarannya!
- f. Lakukan perubahan, tambahkan kode program untuk:
 - 1) menambah kecepatan, gunakan metode `tambahCepat()`;
 - 2) mengurangi kecepatan, gunakan metode `rem()`; dan
 - 3) mengisi nilai setiap atribut secara dinamis.

b. Pewarisan Majemuk (*Multiple Inheritance*)

Pewarisan majemuk merupakan konsep dalam yang memungkinkan sebuah kelas dapat mewarisi sifat atau metode dari lebih dari satu kelas induk. Artinya, kelas turunan dapat menggabungkan sifat dan perilaku dari lebih dari satu kelas induk dan memungkinkan pemrogram (*programmer*) untuk mengombinasikan beberapa fungsionalitas ke dalam satu kelas. Perhatikan contoh pada gambar 3.6 berikut.



Gambar 3.6 Contoh Pewarisan Majemuk

Pada gambar 3.6 terlihat bahwa sepeda listrik memiliki dua kelas induk, yaitu kelas sepeda motor dan kelas sepeda. Kelas sepeda motor memiliki sifat `digas()`, sedangkan kelas sepeda memiliki sifat `dikayuh()`. Sepeda listrik sendiri memiliki sifat `isiDaya()`. Oleh karena itu, sepeda listrik turunan dari sepeda motor dan sepeda. Selain itu, sepeda listrik juga memiliki sifat `digas()` yang berasal dari kelas sepeda motor dan sifat `dikayuh()` yang berasal dari kelas sepeda.



Aktivitas 3.7

Menerapkan Pewarisan Majemuk dalam Kode Program (Aktivitas Individu)

Kali ini kamu diminta untuk mempraktikkan langsung penerapan pewarisan majemuk! Ikuti langkah-langkah berikut. Perhatikan juga gambar 3.6.

- Definisikan kelas induk “Sepeda Motor”, kemudian lakukan modifikasi dari kode program 3.19!
- Definisikan kelas induk “Sepeda”, lalu lakukan modifikasi dari kode program 3.20!
- Definisikan kelas turunan “Sepeda Listrik” yang merupakan turunan dari kelas “Sepeda Motor” dan kelas “Sepeda” seperti kode program 3.21 berikut!

Kode Program 3.21

Kelas Sepeda Listrik Diturunkan dari Kelas Sepeda Motor dan Kelas Sepeda

```
01 class SepedaListrik(SepedaMotor, Sepeda):
02     def __init__(self, merk, tipe, kapasitas_baterai,
03                 jenis_medan):
04         SepedaMotor.__init__(self, merk, tipe, None)
05         # Tidak menggunakan atribut cc
06         Sepeda.__init__(self, merk, tipe, jenis_medan)
07         self.kapasitas_baterai = kapasitas_baterai
08
09     # Metode digas dari SepedaMotor dapat digunakan
10     # Metode dikayuh dari Sepeda dapat digunakan
11
12     def isi_baterai(self):
13         print(f"{self.merk} {self.tipe} sedang diisi “,
14               “baterai, kapasitas “,
15               {self.kapasitas_baterai} Ah.”)
16
```

- d. Buatlah fungsi utama untuk menjalankan kelas Sepeda Motor dan Sepeda seperti pada kode program 3.22 berikut.

Kode Program 3.22 Fungsi Utama Menjalankan Pewarisan

```
17 if __name__ == "__main__":
18     sepeda_listrik_baru = SepedaListrik("Selis",
19                                         "Bangau", 12,
20                                         "Jalan raya")
21     sepeda_listrik_baru.digas()
22     sepeda_listrik_baru.dikayuh()
23     sepeda_listrik_baru.isi_baterai()
24
25     # Contoh akses atribut
26     print(f"Merk: {sepeda_listrik_baru.merk}")
27     print(f"Tipe: {sepeda_listrik_baru.tipe}")
28     print(f"Kapasitas Baterai: "
29           f"{sepeda_listrik_baru.kapasitas_baterai} Ah")
30     print(f"Jenis Medan: {sepeda_listrik_baru.jenis_medan}")
```

- e. Jalankan kode program di atas dan amati bagaimana keluarannya! Adakah metode yang berubah pada kelas turunannya?
- f. Lakukan perubahan, kemudian tambahkan kode program untuk mengisi nilai setiap atribut secara dinamis!

2. Polimorfisme (*Polymorphism*)

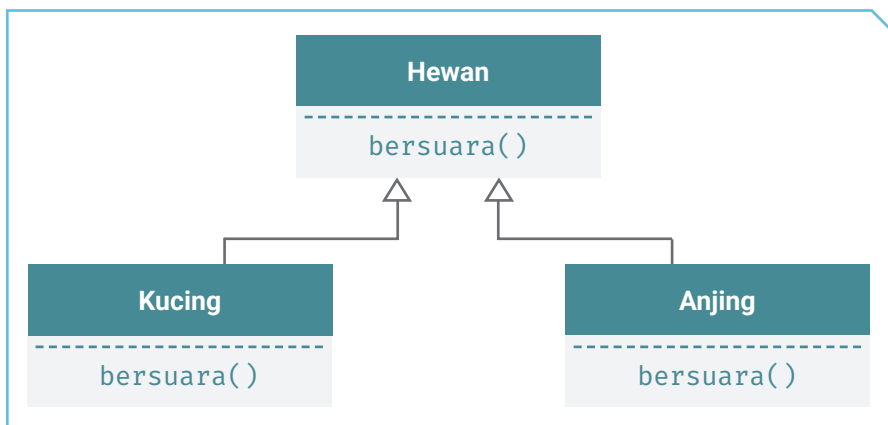
Bayangkan kamu memberi perintah "bersuara", tapi ditujukan ke hewan yang berbeda, yaitu kucing, anjing, dan ayam. Apa yang akan terjadi? kucing: "meong"; anjing: "guk-guk"; dan ayam: "kukuruyuk". Begitu juga apabila kamu memberi perintah "berjalan" kepada manusia dan robot. Keduanya akan melakukan aksi "berjalan", tetapi dengan cara yang berbeda. Nah, inilah yang disebut dengan polimorfisme—satu perintah, tetapi responsnya bisa berbeda tergantung pada siapa yang menjalankannya. Secara sederhana, polimorfisme sama dengan satu nama, banyak aksi.

Dalam konteks pemrograman berorientasi objek, polimorfisme adalah kemampuan suatu objek untuk merespons perintah yang sama dengan cara yang berbeda, tergantung pada jenis objeknya. Polimorfisme erat kaitannya dengan pewarisan. Polimorfisme merupakan salah satu kemampuan pemrograman berorientasi objek yang memungkinkan suatu kelas memiliki beberapa bentuk yang berbeda. Bentuk yang dimaksud adalah kesamaan nama metode dan parameter, tetapi memiliki definisi yang berbeda-beda di dalamnya. Secara umum, polimorfisme memiliki dua bentuk dasar, yaitu polimorfisme *static* dan polimorfisme *dynamic*. Polimorfisme *static* dilakukan dengan menggunakan teknik *overloading*, sedangkan polimorfisme *dynamic* menggunakan teknik *overriding*.

a. *Overriding*

Overriding secara harfiah bermakna menimpa. Apa yang ditimpa? Sebagai contoh, ayahmu mempunyai cara menyapa orang lain, misalnya "Selamat pagi, Pak!" Kamu, sebagai anak, juga mempunyai cara menyapa dengan gaya sendiri, misalnya "Morning, Pah!" Nah, dalam hal itu, sebenarnya kamu meng-*override* (menimpa) cara menyapa milik ayahmu supaya lebih cocok dengan duniamu.

Overriding merupakan suatu mekanisme pada pemrograman objek yang berarti menulis ulang deskripsi dari sebuah metode yang ada pada kelas induk dalam kelas turunannya. Dengan menulis ulang metode tersebut, maka nama, parameter, maupun tipe keluaran harus sama, hanya deskripsinya yang berbeda.



Gambar 3.7 Contoh *Override*

Perhatikan gambar 3.7 di atas. Misalnya, kelas induk hewan dan dua kelas turunan kucing dan anjing memiliki sifat yang sama, yaitu `bersuara()`. Jika diimplementasikan, maka akan terlihat seperti pada kode program 3.23 berikut.

Kode Program 3.23 *Overriding* Sifat pada Kelas Induk

```
01 class Hewan:
02     def bersuara(self):
03         print("Hewan bersuara")
04
05 class Kucing(Hewan):
06     def bersuara(self):
07         print("Meong!")
08
09 class Anjing(Hewan):
10     def bersuara(self):
11         print("Guk-guk!")
12
13 # Mari jalankan
14 if __name__ == "__main__":
15     k = Kucing()
16     a = Anjing()
17
18     k.bersuara() # Meong!
19     a.bersuara() # Guk-guk!
```

Pada contoh di atas, metode `bersuara()` pada kelas kucing dan anjing melakukan *override* terhadap metode `bersuara()` di kelas hewan.



Aktivitas 3.8

Implementasi *Overriding* (Aktivitas Individu)

Kerjakan aktivitas 3.8 ini dengan mengikuti petunjuk berikut.

- a. Buatlah kelas kendaraan seperti rincian berikut!

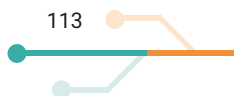
Kode Program 3.24 Kelas Induk Kendaraan

```
01 class Kendaraan:
02     def __init__(self, model):
03         self.model = model
04
05     def informasi(self):
06         print("Informasi kendaraan")
07
```

- b. Buatlah kelas pesawat yang merupakan turunan dari kendaraan!

Kode Program 3.25 Kelas Pesawat Turunan dari Kendaraan

```
08 class Pesawat(Kendaraan):
09     def __init__(self, nama, jenis="belum “
10         “teridentifikasi”):
11         super().__init__("pesawat")
12         self.nama = nama
13         self.jenis = jenis
14
15     def informasi(self):
16         print(f>Nama pesawat adalah {self.nama}")
17         print(f>Jenis pesawat adalah {self.jenis}")
18
```



c. Buatlah kelas mobil yang merupakan turunan dari kendaraan!

Kode Program 3.26 Kelas Mobil Turunan dari Kendaraan

```
19 class Mobil(Kendaraan):
20     def __init__(self, nama, jenis="belum "
21                 "teridentifikasi"):
22         super().__init__("Mobil")
23         self.nama = nama
24         self.jenis = jenis
25
26     def informasi(self):
27         print(f>Nama mobil adalah {self.nama}")
28         print(f>Jenis mobil adalah {self.jenis}")
29
```

d. Buatlah kelas kapal yang merupakan turunan dari kendaraan!

Kode Program 3.27 Kelas Kapal Turunan dari Kendaraan

```
30 class Kapal(Kendaraan):
31     def __init__(self, nama, jenis="belum "
32                 "teridentifikasi"):
33         super().__init__("kapal")
34         self.nama = nama
35         self.jenis = jenis
36
37     def informasi(self):
38         print(f>Nama kapal adalah {self.nama}")
39         print(f>Jenis kapal adalah {self.jenis}")
40
```

- e. Buatlah program utama untuk menjalankannya!

Kode Program 3.28 Kelas Kapal Turunan dari Kendaraan

```
41 # Main program
42 if __name__ == "__main__":
43     # Membuat objek
44     psw = Pesawat("Boeing 737", "Pesawat Komersil")
45     mbl1 = Mobil("Mitsubhisi Pajero", "SUV")
46     mbl2 = Mobil("Honda City", "Sedan")
47     mbl3 = Mobil("VW Combi")
48     kpl = Kapal("Queen Mary 2", "Kapal Pesiar")
49
50     # Polimorfisme: variabel p bisa menunjuk
51     # ke semua objek turunan Kendaraan
52     kendaraan_list = [psw, mbl1, mbl2, mbl3, kpl]
53
54     for p in kendaraan_list:
55         p.informasi()
56         print("---")
```

- f. Jalankan kode program di atas dan amati bagaimana keluarannya! Lalu, jelaskan di mana letak perbedaannya!
- g. Lakukan perubahan, kemudian tambahkan kode program untuk mengisi nilai setiap atribut secara dinamis!

b. *Overloading*

Overloading terjadi manakala sebuah kelas memiliki beberapa metode dengan nama yang sama, tetapi dengan parameter yang berbeda. Sebagai contoh, kamu mempunyai alat "cetak" satu tombol, tetapi dapat digunakan untuk mencetak foto, dokumen, atau kartu identitas. Contoh lainnya, tombol "play" di *handphone*-mu. Jika ditekan sekali, maka musik diputar, lalu jika ditekan sekali lagi, maka akan terjadi jeda (*pause*). Setelah itu, jika ditekan agak lama, maka akan muncul pilihan mode: *shuffle*, *repeat*, dan lain-lain. Akan tetapi, jika ditekan cepat dua kali, maka akan mempercepat durasi perputaran musik. Tombolnya

sama, tetapi hasilnya tergantung pada bahan yang dimasukkan. Nah, inilah yang dimaksud dengan *overloading*, yaitu nama metode sama, tetapi hasilnya tergantung pada "isi" atau parameternya.

Sayangnya, Python tidak mendukung metode *overloading* secara langsung, seperti Java atau C++, tetapi dapat menirukannya dengan `default arguments` atau `*args`.

Kode Program 3.29 Contoh *Overloading* dengan `*args`

```
01 class Gambar:
02     def gambar(self, *args):
03         if len(args) == 2:
04             print(f"Menggambar titik di {args}")
05         elif len(args) == 4:
06             print(f"Menggambar garis dari {args[0:2]} “,
07                 “ke {args[2:4]}”)
08         elif len(args) == 6:
09             print(f"Menggambar segitiga dengan titik “,
10                 {args})
11         elif len(args) == 8:
12             print(f"Menggambar segiempat dengan titik “,
13                 {args})
14         else:
15             print("Format tidak dikenal")
16 if __name__ == "__main__":
17     # Membuat objek Gambar
18     g = Gambar()
19
20     # Menggambar titik
21     g.gambar(10, 20)
22
23     # Menggambar garis
24     g.gambar(10, 20, 30, 40)
25
26     # Menggambar segitiga
27     g.gambar(10, 20, 30, 40, 50, 60)
28
```

```

29     ... # Menggambar segiempat
30     ... g.gambar(10, 20, 30, 40, 50, 60, 70, 80)
31
32     ... # Contoh format tidak dikenal
33     ... g.gambar(1, 2, 3)

```



Aktivitas 3.9

Implementasi Overloading (Aktivitas Individu)

Ikuti petunjuk berikut untuk mengerjakan aktivitas ini.

- Implementasikan kode program 3.29!
- Jalankan dan amati hasilnya! Apa perbedaan dari ketiga bentuk tersebut?
- Buatlah kelas kalkulator dengan metode `hitung()` yang dapat melakukan:
 - penjumlahan jika diberikan 2 angka;
 - pengurangan jika diberikan 3 angka; dan
 - perkalian jika diberikan 4 angka.

Nah, setelah kamu mempelajari tentang kedua teknik tersebut (*overloading* dan *overriding*), kini kamu mengetahui perbedaannya. Secara garis besar, perbedaan *overloading* dan *overriding* dapat dilihat pada tabel berikut.

Tabel 3.6 Perbandingan *Overriding* dan *Overloading*

Aspek	<i>Overloading</i>	<i>Overriding</i>
Terjadi di	Kelas yang Sama	Antara Kelas Induk & Turunan
Nama Metode	Sama	Sama
Parameter	Berbeda	Sama
Fungsi	Menyediakan Variasi	Menimpa Perilaku Induk

G. Kelas Abstrak

Kelas abstrak adalah “cetak biru” yang belum lengkap. Bayangkan kamu mempunyai cetak biru kendaraan, tetapi kendaraan ini belum bisa digunakan karena belum lengkap komponennya. Nah, kelas abstrak adalah cetak biru yang digunakan sebagai dasar bagi kelas lain, tetapi tidak bisa langsung dipakai.

Kelas abstrak adalah kelas dasar yang tidak dapat digunakan secara langsung untuk membuat objek, melainkan sebagai cetak biru bagi kelas-kelas turunannya. Kelas abstrak biasanya digunakan ketika kita ingin mendefinisikan perilaku umum, tapi implementasi spesifiknya diserahkan pada kelas turunan. Dengan demikian, kelas abstrak dapat diibaratkan seperti "konsep" dari sesuatu, tetapi bukan benda nyatanya.

Kelas abstrak merupakan salah satu karakteristik yang dimiliki oleh pengguna untuk mengekspresikan atribut dan metode yang bersifat umum dan dapat digunakan oleh kelas-kelas turunannya (Eck, 2021). Jadi, kelas abstrak sebenarnya adalah kelas induk yang berisi sifat-sifat dasar secara umum. Objek hanya dapat dibuat dari kelas turunannya, bukan dari kelas abstraknya langsung.

Perilaku seperti ini mirip dengan seorang arsitek yang membuat cetakan rumah (kelas abstrak). Akan tetapi, rumahnya belum bisa dihuni karena belum dibangun. Nah, rumah-rumah nyata (kelas turunan) harus menyelesaikan bagian-bagian yang belum lengkap, seperti warna, luas, atau jenis atap. Sama halnya seperti metode abstrak yang harus diisi.

Python menggunakan modul `abc` (*abstract base class*) untuk membuat kelas abstrak. Kamu harus meng-`import` terlebih dahulu kelas dasar abstrak serta `abstractmethod`.

```
from abc import ABC, abstractmethod
```

Untuk lebih memahami penjelasan tersebut, perhatikan contoh kode program 3.30 berikut.

Kode Program 3.30

Contoh Kelas Abstrak Hewan

```
01 class Hewan(ABC):
02     @abstractmethod
03     def bersuara(self):
```

```

04     ..... pass #tidak ada isi perintah
05
06 class Kucing(Hewan):
07     .... def bersuara(self):
08     ..... print("Meong~")
09
10 class Anjing(Hewan):
11     .... def bersuara(self):
12     ..... print("Guk-guk!")
13
14 # Program utama
15 if __name__ == "__main__":
16     .... hewan1 = Kucing()
17     .... hewan2 = Anjing()
18     .... hewan1.bersuara() # Output: Meong~
19     .... hewan2.bersuara() # Output: Guk-guk!

```

Jika kelas kucing tidak menuliskan isi dari metode bersuara,

Kode Program 3.31

```

06 class Kucing(Hewan):
07     .. pass

```

maka akan muncul eror.

```

TypeError: Can't instantiate abstract class Kucing with abstract
method bersuara

```

Eror ini terjadi karena pada kelas hewan tidak ada hal yang harus dilakukan pada metode `bersuara()` sehingga kelas turunan harus menuliskan sendiri apa yang harus dilakukannya pada metode tersebut.



Aktivitas 3.10

Membuat dan Menggunakan Kelas Abstrak (Aktivitas Individu)

Kerjakan aktivitas ini dengan mengikuti langkah-langkah berikut.

- Buatlah kelas abstrak bernama hewan dengan metode abstrak `suara()` dan `makan()`!
- Buatlah dua kelas turunan: kucing dan burung! Masing-masing diisi dengan metode `suara()` dan `makan()`!
- Jalankan program utama untuk memunculkan suara dan makanan favorit masing-masing hewan!
- Setelah berhasil dijalankan, lakukan perubahan kecil dengan menghapus implementasi `suara()` di kelas burung! Apa yang terjadi?
- Buatlah objek langsung dari kelas hewan! Apakah berhasil?
- Tambahkan metode `habitat()` di kelas hewan dan tidak menuliskannya di kelas kucing! Apa yang terjadi jika `habitat()` dijadikan abstrak?



Uji Kompetensi

Petunjuk Umum:

- Bacalah setiap soal dengan saksama.
- Jawablah soal pilihan ganda dengan memberi tanda silang (X) pada huruf a/b/c/d/e yang dianggap benar.
- Jawaban uraian/refleksi ditulis secara singkat, jelas, dan mendalam.

A. Soal Pilihan Ganda (Pemahaman dan Analisis Konsep)

- Berikut ini yang merupakan analogi terbaik untuk konsep kelas dan objek dalam kehidupan sehari-hari adalah
 - kelas = warna; objek = lukisan
 - kelas = cetakan kue; objek = kue yang dibuat dari cetakan tersebut
 - kelas = orang tua; objek = anak-anaknya
 - kelas = komputer; objek = program di dalamnya
 - kelas = resep makanan; objek = makanan yang dimasak berdasarkan resep tersebut

2. Manakah dari pernyataan berikut yang paling tepat untuk menggambarkan enkapsulasi dalam OOP?
 - a. Membuat data dalam bentuk *dictionary*.
 - b. Menggabungkan semua kode ke dalam satu *file*.
 - c. Menyembunyikan atribut objek dan hanya membukanya melalui metode tertentu.
 - d. Menghubungkan banyak kelas ke dalam satu objek.
 - e. Menyusun fungsi-fungsi menjadi baris perintah yang berurutan.
3. Perhatikan kode berikut.

```
01 class Hewan:  
02     def suara(self):  
03         print("Hewan bersuara")  
04  
05 class Kucing(Hewan):  
06     def suara(self):  
07         print("Meong~")
```

Ketika kode `kucing().suara()` dijalankan, maka luaran yang ditampilkan adalah

- a. hewan bersuara
 - b. meong~
 - c. meong~ dan hewan bersuara
 - d. eror karena tidak ada konstruktor
 - e. tidak menampilkan apapun
4. Konsep polimorfisme dalam pemrograman berorientasi objek memungkinkan
 - a. menggabungkan dua kelas menjadi satu objek
 - b. menyimpan banyak nilai dalam satu variabel
 - c. menjalankan metode yang sama pada objek yang berbeda dengan cara yang berbeda
 - d. membuat metode dalam bentuk teks
 - e. mengubah kelas menjadi objek lain

5. Dalam Python, agar sebuah kelas menjadi kelas abstrak, maka harus
 - a. menggunakan komentar khusus, seperti `#abstract`
 - b. menggunakan `super()` pada konstruktor
 - c. meng-*import* `abstractmethod` dari modul `abc`
 - d. menuliskan `self` di luar kelas
 - e. menyembunyikan semua atribut dengan `__`

B. Soal Uraian dan Penerapan

1. Jelaskan masing-masing unsur berikut dengan analogi kehidupan sehari-hari!
 - a. kelas dan objek
 - b. pewarisan
 - c. enkapsulasi
 - d. polimorfisme
2. Buatlah program singkat yang terdiri dari:
 - a. Kelas hewan (abstrak) dengan metode `bersuara()`.
 - b. Dua kelas turunan: ayam dan kucing, masing-masing dengan suara khas.
 - c. Tampilkan suara kedua hewan tersebut di program utama.

Rubrik Penilaian

A. Pilihan Ganda

Skor Total = $5 \times 5 = 25$ poin

B. Uraian dan Penerapan

Skor Total = $10 + 30 = 40$ poin

Analogi Konsep OOP (10 poin)

Kriteria	Skor
Menyebutkan dan menjelaskan keempat konsep.	4–6
Memberikan analogi kehidupan nyata relevan.	2–3
Menggunakan bahasa sendiri yang mudah.	1
Skor Total	10 Poin

Kode Program OOP (30 Poin)

Aspek	Skor
Membuat kelas abstrak hewan.	5
Menambahkan metode abstrak <code>bersuara()</code> .	5
Membuat 2 kelas turunan: ayam dan kucing	5
Implementasi metode dengan <i>override</i> yang benar.	5
Program utama berjalan dengan benar.	5
Penulisan kode rapi dan diberi komentar.	5
Skor Total	30 Poin

Pengayaan

Untuk mendalami materi ini, kamu dapat mempelajari bahan-bahan bacaan berikut.

1. Kelas dan Objek

a. Buku

Munir, R. *Algoritma dan Pemrograman dalam Bahasa Python*. Bandung: Informatika, 2020.

(Buku ini menjelaskan tentang dasar-dasar pemrograman Python, termasuk pembuatan kelas dan objek secara bertahap.)

Deitel, P. J., and H. M. Deitel. *Python for Programmers*. Boston: Pearson, 2020.

(Buku ini memuat berbagai studi kasus praktis, termasuk penggunaan kelas dan objek dalam proyek Python tingkat pemula hingga menengah.)

b. Artikel

Real Python. "Object-Oriented Programming (OOP) in Python 3." Accessed June 30, 2025. <https://realpython.com/python3-object-oriented-programming/>.

(Artikel ini membahas bagaimana Python mengimplementasikan OOP secara praktis dengan fokus pada kelas dan objek.)

c. Video

Programiz. “Python OOP Tutorial for Beginners.” YouTube. Accessed June 30, 2025. <https://buku.kemdikbud.go.id/s/KKA11-Video2b>

(Video berdurasi 15 menit ini akan memandu Anda untuk membuat kelas dan objek di Python dengan cara interaktif.)

2. Enkapsulasi

a. Buku

Wahana Komputer. *Pemrograman Python Tingkat Lanjut*. Yogyakarta: Andi, 2021.

(Pembahasan enkapsulasi pada buku ini menjelaskan tentang akses data privat dan metode *setter/getter* dengan banyak ilustrasi kode.)

b. Artikel

Geeks for Geeks. “Encapsulation in Python.” Accessed June 30, 2025. <https://www.geeksforgeeks.org/encapsulation-in-python/>

(Artikel ini membahas teori dan praktik enkapsulasi di Python, termasuk penggunaan *prefix _* dan *__*.)

c. Video

Kelas Terbuka. “Belajar Python OOP #08 – Encapsulation.” YouTube. Accessed June 30, 2025. <https://buku.kemdikbud.go.id/s/KKA11-Video2a>

(Video ini menjelaskan konsep enkapsulasi dalam Python secara jelas, mencakup: Perlindungan data dengan *modifier* (*_* dan *__*), penggunaan metode *getter* dan *setter*, praktik membuat atribut menjadi “*private*”, serta contoh kode dan implementasinya. Materi disampaikan dengan gaya santai, sehingga cocok untuk siswa SMA/SMK dan mahasiswa tingkat dasar yang baru belajar OOP.)

Dunia Ilkom. “PYTHON 14. Enkapsulasi & Constructor.” YouTube. Accessed June 30, 2025. <https://www.youtube.com/watch?v=IhwgorxcGic>

(Video ini menjelaskan tentang enkapsulasi dalam konteks *class* dan *constructor* di Python. Video ini cocok untuk orang yang ingin memahami hubungan antara enkapsulasi dan proses inisialisasi objek. Dilengkapi dengan contoh langsung dalam kode Python.)

3. Pewarisan (*Inheritance*)

a. Buku

Hidayat, R. *Pemrograman Python Berbasis OOP untuk SMK dan MA*. Jakarta: Bumi Aksara, 2023.

(Buku ini menampilkan contoh *inheritance* tunggal dan ganda dalam Python dengan studi kasus kendaraan dan hewan.)

Lutz, M. *Learning Python*. 5th ed. Sebastopol, CA: O'Reilly Media, 2013.

(Buku ini memberikan pendekatan mendalam terhadap *inheritance* di Python dan fitur pewarisan majemuk.)

b. Artikel

Programiz. "Python Inheritance." Accessed June 30, 2025. <https://www.programiz.com/python-programming/inheritance>

(Artikel ini disertai dengan visualisasi pewarisan antarkelas dan penjelasan `super()` yang sangat cocok untuk pemula.)

c. Video

Kelas Terbuka. "Inheritance Python | OOP Python." YouTube. Accessed June 30, 2025. <https://www.youtube.com/watch?v=WIEYbAK4SBM>

(Video ini disampaikan dalam bahasa Indonesia dengan gaya santai yang menjelaskan tentang pewarisan tunggal dan pewarisan majemuk.)

4. Polimorfisme

a. Buku

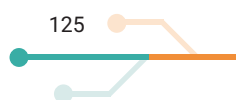
Eck, David J. *Introduction to Programming Using Python*. Hobart and William Smith Colleges, 2019.

(Bab tentang polimorfisme pada buku ini menjelaskan tentang penggunaan metode *overriding* dan peran polimorfisme dalam sistem dinamis.)

b. Artikel

Educative.io. "Polymorphism in Python." Accessed June 30, 2025. <https://www.educative.io/answers/what-is-polymorphism-in-python>

(Artikel ini menjelaskan tentang polimorfisme dengan analogi sehari-hari dan demonstrasi kode menggunakan *superclass*.)



c. Video

Tech with Tim. “Polymorphism in Python.” YouTube. Accessed June 30, 2025. <https://buku.kemdikbud.go.id/s/KKA11-Video2c>

(Video ini memberikan contoh perintah yang sama namun respons berbeda dari berbagai *subclass*.)

5. Kelas Abstrak

a. Buku

Nielsen, M. A., and I. Chuang. *Artificial Intelligence with Python*. Cambridge: Cambridge University Press, 2021.

(Salah satu bab pada buku ini menjelaskan penggunaan kelas abstrak untuk struktur sistem KA modular dan fleksibel.)

Wibowo, A. *Pemrograman OOP dengan Python*. Yogyakarta: Deepublish, 2020. (Kelas abstrak dijelaskan menggunakan modul `abc` dan `@abstractmethod`, disertai analogi cetakan rumah.)

b. Artikel

Python Docs. “`abc` — Abstract Base Classes.” Accessed June 30, 2025. <https://docs.python.org/3/library/abc.html>

(Dokumentasi resmi Python tentang bagaimana membuat dan menggunakan kelas abstrak.)

c. Video

Corey Schafer. “Abstract Base Classes (`abc` module) in Python.” YouTube. Accessed June 30, 2025. <https://www.youtube.com/watch?v=A0j3xjHZ400>

(Video ini menjelaskan *step-by-step* implementasi kelas abstrak di Python dengan ABC dan *abstract method*.)

6. Studi Kasus Implementatif

a. Studi Kasus

GitHub–OOP Python Mini Projects (Kendaraan, Siswa, Buku):

<https://github.com/topics/python-oop-project>

(Sumber proyek nyata, termasuk aplikasi manajemen kendaraan dan perpustakaan dengan OOP.)

DQLab. “Penerapan OOP dalam Proyek Python.” Accessed June 30, 2025.

<https://dqlab.id/oop-dalam-python>

(Menggunakan studi kasus data scientist yang mengelola data dengan *class* dan *object*.)

FreeCodeCamp. “Build an OOP RPG Game in Python.” YouTube. Accessed June

30, 2025. <https://buku.kemdikbud.go.id/s/KKA11-Video2d>

(Studi kasus membuat RPG *game* berbasis objek dan menampilkan pewarisan dan polimorfisme secara jelas.)



Refleksi

Petunjuk:

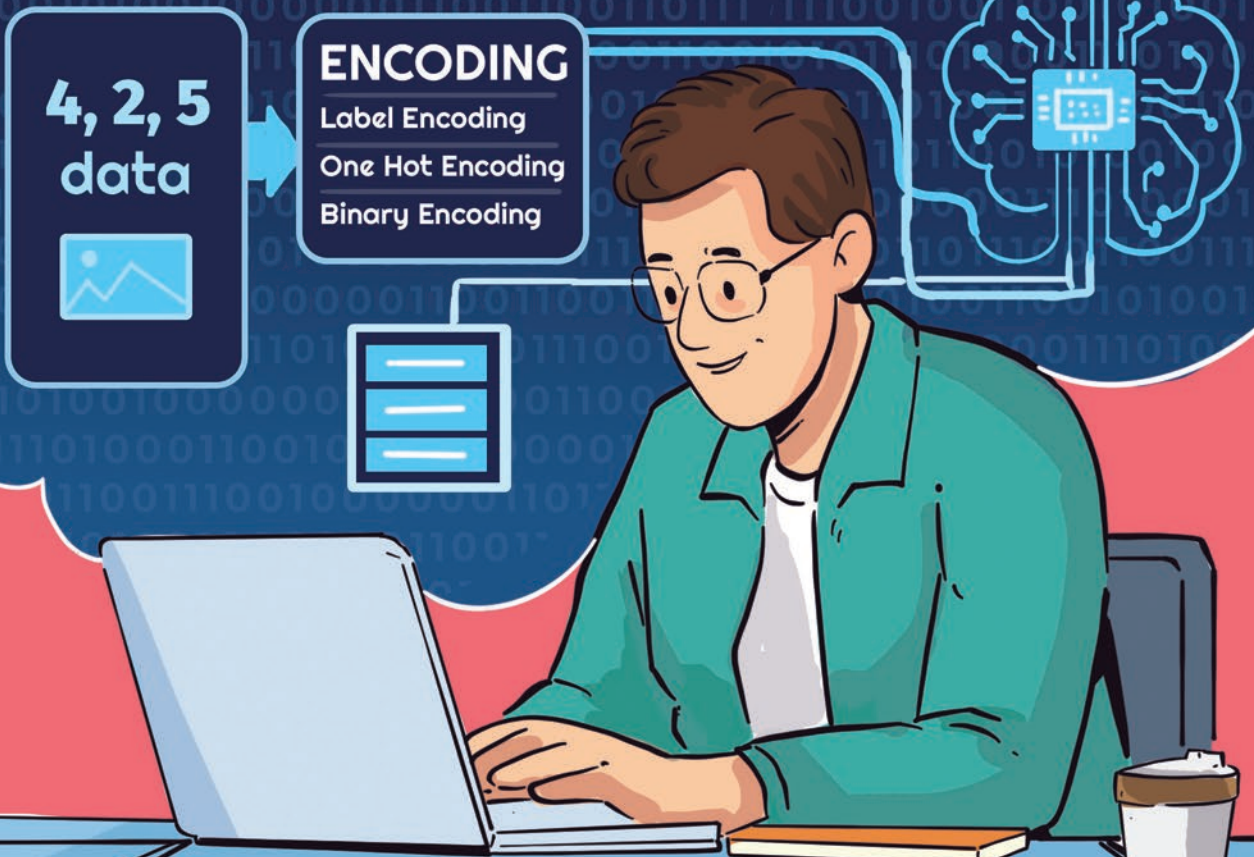
Isilah refleksi berikut dengan jujur berdasarkan pengalamanmu selama mengikuti pembelajaran pada bab ini. Tidak ada jawaban benar atau salah, yang penting adalah proses berpikirmu sendiri.

Pertanyaan Refleksi	Jawabanmu
Apa hal baru yang kamu pelajari dari materi OOP ini?	
Materi mana yang paling kamu pahami (video, latihan, diskusi, atau praktik kode)?	
Bagian mana dari OOP yang menurutmu paling sulit dipahami? Jelaskan alasannya!	

Pertanyaan Refleksi	Jawabanmu
Bagaimana konsep OOP membantu kamu memahami cara kerja dunia nyata?	
Apa analogi OOP yang paling membantumu dalam memahami materi OOP (contohnya, kapsul atau cetakan kue)??	
Apa rencana belajarmu selanjutnya untuk memperdalam OOP?	
Nilailah pemahaman dirimu tentang materi OOP menggunakan skala 1–10, lalu berilah alasan singkat!	

Bab 4

Data *Encoding* dan Integrasi Basis Data dalam Aplikasi



Pernahkah kamu berpikir ketika menggunakan Kecerdasan Artificial (KA),
bagaimana komputer memahami data yang kita berikan?



Tujuan Pembelajaran

Setelah mempelajari Bab 4, diharapkan kamu dapat menunjukkan cara mengubah data menjadi format yang dapat diproses sistem melalui proses *encoding*, termasuk mengimplementasikan teknik-teknik dasar pengodean data. Selain konsep, kamu juga diharapkan dapat mempraktikkan cara menyimpan data yang telah dikodekan ke dalam *database*. Materi ini mencakup pembuatan aplikasi sederhana yang terintegrasi dengan *database* untuk mengelola data.

“ Kuasai data, kuasai masa depan!”

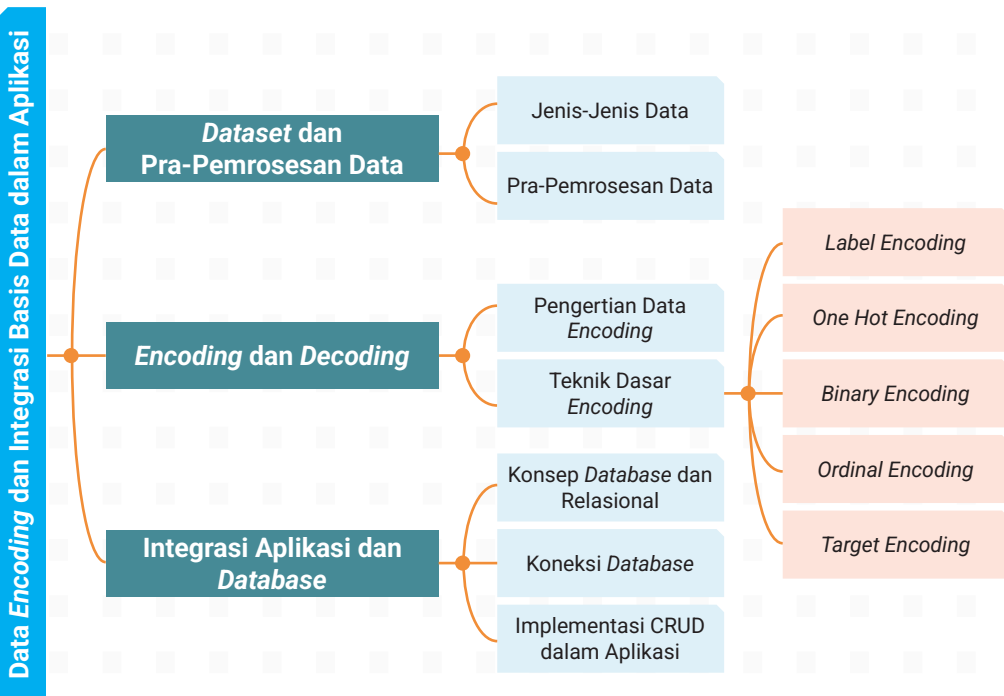


Kata Kunci

database, *dataset*, *encoding*, pra-pemrosesan data, integrasi *database*



Peta Materi





Siap-Siap Belajar

Pernahkah kamu membuat akun pada aplikasi dengan mengisi formulir pendaftaran? Misalnya, saat membuat akun *game*, *marketplace*, atau media sosial. Tentunya kamu akan mengisikan data, seperti nama, memilih jenis kelamin, atau menentukan hobi favorit. Apakah kamu pernah berpikir ke mana data itu disimpan setelah dikirim?

Menurutmu, bagaimana caranya agar aplikasi itu bisa menyimpan dan mengelola data jutaan pengguna dengan cepat tanpa eror? Tahukah kamu, proses apa yang terjadi dan bagaimana rumitnya menghubungkan aplikasi dengan *database* untuk memastikan data tersebut dapat diakses, diproses, dan dianalisis dengan efisien? Pasti ada cara yang rapi, cepat, dan aman untuk menyimpan semua data tersebut. Nah, di sinilah *database* berperan.

Tapi tunggu dulu, sebelum data itu dapat disimpan, ada satu proses penting yang harus dilakukan, yaitu data *encoding*. Misalnya, jika kamu memilih jenis kelamin “laki-laki”, maka aplikasi tidak langsung menyimpan data tersebut begitu saja. Sangat memungkinkan data itu telah diubah menjadi angka, seperti “0” untuk laki-laki dan “1” untuk perempuan. Mengapa? Supaya komputer lebih mudah memproses dan menyimpannya. Data semacam ini biasanya berasal dari *file* yang disebut *dataset*, sering kali dalam bentuk CSV yang mirip seperti tabel Excel. *Dataset* itu harus dibersihkan dan dikodekan terlebih dahulu, baru kemudian disimpan dalam *database* seperti SQLite agar bisa digunakan kembali oleh aplikasi dengan lebih efisien.

Pada bab ini, kamu akan mempelajari langkah-langkah penting dari mengelola *dataset*, melakukan *encoding* data, menyimpan data ke *database*, dan menghubungkan data tersebut ke dalam aplikasi. Dengan memahami proses ini, kamu akan dapat menciptakan aplikasi sederhana yang bisa mengelola data dengan baik. Seru, kan? Yuk, kita mulai petualangan belajarmu!

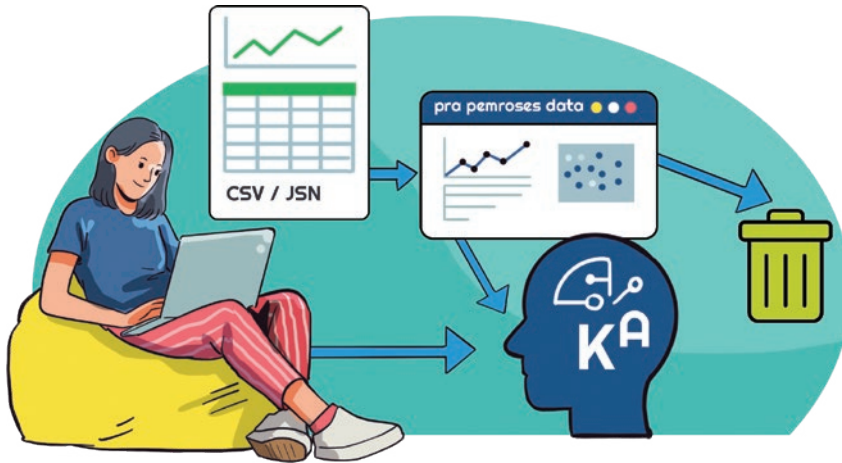


Cek Pemahaman Awal

Jawablah pertanyaan berikut sesuai dengan pengalamanmu selama menggunakan teknologi Kecerdasan Artifisial.

1. Apakah kamu pernah mengetahui istilah “*Encoding Data*”?
Berilah tanda centang (✓) pada pernyataan yang sesuai!
 - Belum pernah.
 - Pernah mendengar, tetapi belum mengetahui secara detail.
 - Iya, mengubah data dengan tujuan agar dapat diakses komputer.
 - Saya sudah memahami dan saya dapat menjelaskan.
2. *Database* sebagai penyimpanan masa kini.
Berilah tanda centang (✓) pada aplikasi yang menurutmu akan menyimpan data pengguna! (Kamu dapat memilih lebih dari satu.)
 - Instagram
 - Spotify
 - kalkulator
 - kamera HP
 - aplikasi absensi sekolah
3. Jika kamu membuat aplikasi toko *online*, data apa saja yang harus disimpan di *database*?
Berilah tanda centang (✓) pada data yang dimaksud! (Kamu dapat memilih lebih dari satu.)
 - nama produk
 - riwayat transaksi
 - jumlah stok
 - foto produk
 - harga
 - total omzet
4. Apakah kamu mengetahui bagaimana sebuah aplikasi dapat berkomunikasi dengan *database*?
Berilah tanda centang (✓) pada pernyataan yang sesuai!
 - Belum mengetahui.
 - Baru mendapatkan informasi yang sedikit.
 - Sudah mengetahui, tetapi belum dapat menjelaskan.
 - Sudah pernah praktik menghubungkan aplikasi ke *database*.
5. Nilailah kemampuan dirimu dengan jujur menggunakan skala 1–5 (1 = belum paham s.d. 5 = mahir) untuk hal-hal berikut!
 - a. memahami konsep *encoding data* = _____
 - b. memahami tentang *database* = _____
 - c. mengetahui cara menghubungkan aplikasi ke *database* = _____

A. Dataset dan Pra-Pemrosesan Data



Gambar 4.1 Proses Mengubah Data oleh Sistem KA

Komputer menggunakan data untuk dapat melakukan berbagai tugas, seperti mengenali wajah, memprediksi cuaca, hingga menampilkan rekomendasi video di media sosial. Kumpulan data yang digunakan untuk pembelajaran ini disebut sebagai *dataset*. Contohnya, saat kamu melakukan pengisian formulir pembuatan akun, sebenarnya kamu mengirimkan data tersebut ke dalam sistem. Data-data yang diinputkan ini akan dikumpulkan terlebih dahulu ke sebuah *dataset*. *Dataset* dapat kamu bayangkan seperti tabel yang ada pada Excel yang berisi baris dan kolom. Satu rangkaian data tersimpan pada satu baris dan satu jenis data tersimpan pada satu kolom yang sama. *Dataset* dapat disimpan dalam berbagai format, seperti CSV (*Comma Separated Values*) yang dapat kamu buka menggunakan Excel atau Google Sheets, ada juga bentuk JSON yang biasa digunakan pada *website*.

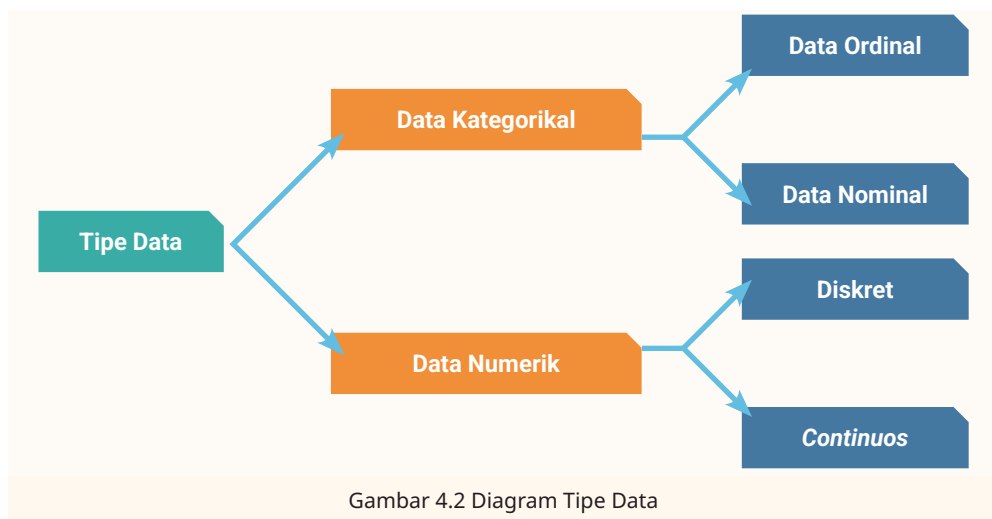
Sebelum *dataset* digunakan dalam program/sistem Kecerdasan Artificiial, kamu dapat melakukan eksplorasi data dengan menganalisis isi dan bentuk data di dalamnya. Sebagai contoh, eksplorasi data *file* CSV yang berisi data siswa. Perhatikan isi pada kolom-kolomnya. Kamu dapat mengelompokkan jenis data, seperti data numerik (angka yang dapat dihitung) dan data kategorikal (*text* atau label).

Data kategorikal tidak dapat langsung dipahami oleh komputer. Misalnya, pada data jenis kelamin yang berisi “laki-laki” atau “perempuan”, maka komputer tidak mengetahui artinya. Jika kamu akan melakukan proses terhadap data jenis kelamin, maka akan terjadi eror atau hasilnya tidak akurat. Dengan kata

lain, data tidak dapat langsung diproses. Oleh karena tidak semua data dapat langsung digunakan, kamu perlu memahami terlebih dahulu tentang jenis-jenis data yang ada dalam *dataset*. Hal ini penting untuk kamu ketahui agar kamu dapat memprosesnya dengan benar sebelum digunakan dalam program atau sistem Kecerdasan Artifiisial.

1. Jenis-Jenis Data

Setiap tipe data dapat kamu gunakan untuk mengklasifikasikan jenis data sesuai dengan karakteristik dan penggunaan data. Perhatikan diagram berikut untuk penjelasan lebih lanjut.



- a. Data Kategorikal (*Categorical Data*) merupakan data yang dapat menggambarkan kategori atau kelompok data. Data tipe ini biasanya berupa label atau nama yang mewakili suatu kelompok.

Contoh:

- Jenis kelamin: “Laki-Laki”, “Perempuan”
- Tingkat pendidikan: “SD”, “SMP”, “SMA”

Terdapat dua jenis data pada data kategorikal. Berikut penjelasan mengenai kedua jenis data tersebut.

- 1) Data Ordinal, yaitu data kategorikal yang memiliki urutan atau hierarki.

Contoh:

- Suhu udara: “dingin”, “sejuk”, “hangat”, “panas”
- Hasil survei: “tidak puas”, “cukup puas”, “sangat puas”

2) Data Nominal, yaitu data kategorikal yang tidak memiliki urutan.

Contoh:

- Warna: “merah”, “hijau”, “biru”
- Kategori produk: “elektronik”, “ATK”, “pakaian”

b. Data Numerik (*Numerical Data*) merupakan data yang berbentuk angka dan dapat dihitung.

Contoh:

- Berat badan (kg): 50, 55, 60, 70
- Suhu udara: 25°C, 30°C, 35°C

Berikut penjelasan mengenai jenis data numerik.

1) *Discrete* (Diskret), yaitu data numerik yang hanya bisa mengambil nilai tertentu (biasanya bilangan bulat).

Contoh: jumlah anak dalam keluarga (1, 2, 3)

2) *Continuous* (Kontinu), yaitu data numerik yang bisa mengambil nilai apapun dalam rentang tertentu (biasanya bilangan desimal).

Contoh: tinggi badan (170.5 cm, 175.2 cm)



Aktivitas 4.1

Mari Bermain Mengklasifikasi Jenis Data (Aktivitas Individu)

Game “Detektif Data”

Tabel Klasifikasi Jenis Data

a. Klasifikasikan contoh data berikut berdasarkan jenis data yang sesuai.

No.	Contoh Data	Nominal	Ordinal	Diskret	Kontinu
1	Warna mobil di parkir an sekolah				
2	Tingkat kepuasan pelanggan (tidak puas, cukup, sangat puas)				

No.	Contoh Data	Nominal	Ordinal	Diskret	Kontinu
3	Berat badan siswa dalam kg				
4	Jumlah siswa di dalam kelas				
5	Jenis kelamin				
6	Tingkat pendidikan (SD, SMP, SMA)				
7	Ukuran baju (S, M, L, XL)				
8	Jumlah pensil di dalam kotak				
9	Suhu ruangan dalam derajat celcius				
10	Nilai ujian siswa				
11	Level kesulitan soal (mudah, sedang, sulit)				
12	Golongan darah (A, B, AB, O)				
13	Banyak anak dalam keluarga				
14	Waktu tempuh lari 100 meter dalam detik				
15	Skor rasa pedas makanan (tidak pedas, pedas, sangat pedas)				

No.	Contoh Data	Nominal	Ordinal	Diskret	Kontinu
16	Jarak dari rumah ke sekolah dalam km				
17	Kode Pos				

b. Setelah mengerjakan aktivitas ini, jawablah pertanyaan berikut dengan tepat.

1) Hal baru apa yang kamu pelajari terkait jenis-jenis data?

2) Data mana yang paling mudah dan paling sulit kamu klasifikasikan? Jelaskan alasannya!

2. Pra-Pemrosesan Data

Setelah mengenal tentang *dataset* dan memahami jenis-jenis data yang ada di dalamnya, langkah selanjutnya adalah mempersiapkan data agar siap digunakan oleh program atau sistem Kecerdasan Artifisial. Data perlu dipersiapkan dan dibersihkan dahulu melalui proses yang disebut dengan pra-pemrosesan data. Pra-pemrosesan data merupakan langkah penting yang bertujuan agar data lebih rapi, bersih, dan siap digunakan. Berikut adalah tahapan pra-pemrosesan data.

a. Membersihkan data (*data cleaning*), yaitu menghapus data kosong, memperbaiki penulisan, dan membuang duplikat.

1) Ada data kosong atau tidak lengkap.

2) Ada data yang bentuknya tidak konsisten, seperti penulisan “laki-laki” dan “Laki-Laki”. Menyamakan format penulisan (misalnya mengubah semua “l” menjadi “Laki-laki”), serta mengecek apakah ada data yang tidak masuk akal.

- b. Mengklasifikasi jenis data, yaitu menentukan data mana yang termasuk data numerik dan mana data yang termasuk data kategorikal.
- c. Mengubah data kategorikal ke bentuk angka (*encoding*)
 - 1) Data berupa teks, seperti “IPA” atau “IPS” tidak bisa langsung dipahami oleh program komputer yang butuh data berupa angka.
 - 2) Komputer tidak bisa menghitung teks, seperti “IPA”, kecuali diubah ke bentuk angka (IPA = 0 dan IPS = 1).

Salah satu bagian penting dari pra-pemrosesan data adalah mengubah data teks menjadi angka yang disebut juga dengan *encoding*. Materi tentang proses tersebut akan kamu pelajari pada subbab berikutnya.



Aktivitas 4.2

Aktivitas 4.2 Eksplorasi *Dataset* (Aktivitas Individu)

Ikuti langkah berikut untuk mengerjakan Aktivitas 4.2 ini.

- a. Buka *file dataset* siswa yang dapat diunduh melalui kode QR berikut.



<https://buku.kemdikbud.go.id/s/KKA11-Aktivitas4>

- b. Perhatikan isi kolom dan data pada setiap baris dalam *dataset* tersebut.
- c. Warnai kolom dengan warna biru untuk data numerik dan warna hijau untuk data kategorikal.
- d. Tuliskan hasil identifikasimu.
 - 1) Data Numerik

- 2) Data Kategorikal

e. Refleksikan masalah dalam *dataset* tersebut.

1) Apa yang bisa terjadi jika sistem KA menerima data dalam bentuk teks, seperti “Perempuan” atau “IPA” tanpa diproses dahulu?

2) Mengapa *dataset* tersebut perlu dilakukan pra-pemrosesan?

Praktik *Encoding* Manual

- Pilihlah satu kolom kategorikal.
- Buatlah salinan kolom tersebut dengan kolom baru dan ubah nilainya.
- Simpan *file* dengan nama baru *Dataset_siswa_encoded.csv*

Pada aktivitas ini, kamu akan dinilai dengan kriteria berikut:

- klasifikasi jenis data (20%);
- analisis dan refleksi (20%);
- praktik *encoding* manual (40%); dan
- kerapian hasil kerja (20%).

Dataset juga dapat disimpan dalam *drive* penyimpanan dan dapat digunakan kembali. Berikut cara mengakses *dataset* dari Google Drive.

a. *Mount* Google Drive

Kode Program 4.1

```
01 #akses Gdrive letak dataset
02 from google.colab import drive
03 drive.mount('/content/drive')
```

b. Bacalah *dataset* dari Google Drive
Misalnya *file dataset* kamu berada pada *drive*:
My Drive/Dataset/transportasi.csv

Kode Program 4.2

```
01 #Path dataset dan pembacanya
02 import pandas as pd
03
04 file_path = '/content/drive/My Drive/Dataset/transportasi.csv'
05 df = pd.read_csv(file_path)
06 print(df.head())
```

c. Simpan kembali hasilnya ke Google Drive

Kode Program 4.3

```
01 output_path = '/content/drive/My
   Drive/Dataset/transportasi_encoded.csv'
02 df.to_csv(output_path, index=False)
03 print(f"Hasil disimpan ke: {output_path}")
04
```

Kamu dapat menggunakan cara lain, seperti URL *share* dari Google Drive yang dibagikan dengan *link* publik. Berikut contoh *link* data yang sudah disimpan dan siap dibagikan.

https://drive.google.com/file/d/11qDP1Sy_OQHwOIX7AQHDZUXEo_FL540G/view?usp=sharing

Adapun langkah untuk mengakses *dataset* tersebut, yaitu:

a. *Install* gdown

```
pip install gdown
```

b. *Download file* dari URL

Berikut *link* yang akan digunakan.

https://drive.google.com/file/d/11qDP1Sy_OQHwOIX7AQHDZUXEo_FL540G/view?usp=sharing

Gunakan ID *file* untuk menentukan ID *file* Google Drive. Kamu cukup melihat bagian URL setelah/d/ dan sebelum/view.

https://drive.google.com/file/d/11qDP1Sy_QQHw0IX7AQHDZUXEo_FL540G/view?usp=sharing

ID File

Kode Program 4.4

```
01 import gdown
02
03 file_id = '11qDP1Sy_QQHw0IX7AQHDZUXEo_FL540G'
04 gdown.download(f'https://drive.google.com/uc?id={file_id}',
05               'transportasi.csv', quiet=False)
06
07 # Baca file
08 import Pandas as pd
09 df = pd.read_csv('transportasi.csv')
10 print(df.head())
```

Penjelasan program:

- Mengimpor pustaka `gdown` untuk mengunduh *file* dari Google Drive (baris 01);
- Mendefinisikan ID *file* Google Drive. ID ini adalah bagian unik dari URL Google Drive yang menunjukkan *file* tertentu (baris 03);
- Mengunduh *file* dari Google Drive dengan ID yang sudah didefinisikan dan menyimpannya ke *file* lokal bernama `transportasi.csv` (baris 04).
- Mengimpor pustaka `Pandas` yang sangat umum digunakan untuk manipulasi dan analisis data dalam Python;
- Membaca *file* CSV (`transportasi.csv`) dan menyimpannya ke dalam objek `df` yang bertipe `Pandas.DataFrame` (baris 08);
- Menampilkan 5 baris pertama dari data `df`. Ini berguna untuk melihat struktur awal dari *dataset* yang dimuat, termasuk nama kolom dan sampel isinya (baris 09).

B. Encoding dan Decoding

Dalam sistem pengelolaan data pada sistem Kecerdasan Artifisial (KA), proses *encoding* dan *decoding* merupakan langkah penting sebelum data disimpan ke dalam *database*. Sebelum data dimasukkan ke dalam *database*, data tersebut diubah dahulu melalui proses yang disebut *encoding*. *Encoding* adalah cara untuk mengubah data agar lebih mudah dibaca dan diproses sistem. Kemudian, saat data akan digunakan kembali, terjadilah proses *decoding*, yaitu mengembalikan data ke bentuk semula. Proses *encoding* dan *decoding* ini dilakukan untuk menjaga keamanan data dan membuat komputer lebih mudah memproses informasi tersebut. Dengan begitu, data tetap aman, rapi, dan mudah digunakan dalam sistem.

1. Pengertian Data Encoding

Dalam pra-pemrosesan data, salah satu tahapan yang penting adalah *encoding*, yaitu mengubah data kategorikal (berbentuk teks atau label) menjadi bentuk angka agar bisa dipahami dan diproses oleh komputer. *Encoding* adalah proses mengubah data dari satu format ke format lain yang lebih mudah diproses oleh komputer. *Encoding* merupakan proses yang penting dalam Kecerdasan Artifisial karena algoritma pada Kecerdasan Artifisial tidak dapat langsung berupa data teks atau simbol.

2. Teknik Dasar Encoding

Terdapat lima jenis teknis dasar *encoding* yang perlu kamu ketahui. Berikut penjelasan mengenai kelima teknis tersebut.

a. Label Encoding

Teknik *encoding* yang pertama yang akan kamu pelajari adalah *label encoding*. Teknik ini digunakan untuk mengubah data yang berbentuk kategori atau label menjadi data numerik ketika kamu menggunakan algoritma *machine learning* atau analisis data yang membutuhkan data dengan jenis angka.

Label *encoding* ini dapat kamu gunakan ketika *dataset* memiliki jumlah kategori yang sedikit dan data yang ada tidak memiliki hubungan ordinal (urutan) antarkategori. Contoh data tersebut, di antaranya:

jenis kelamin, warna, dan jenis aktivitas. Berikut contoh proses dari teknik *label encoding*.

Bidang kesiswaan ingin mendata transportasi yang digunakan siswa untuk berangkat ke sekolah, lalu diperoleh *dataset* sebagai berikut.

Tabel 4.1 Contoh Proses dari Teknik *Label Encoding*

Siswa	Jenis Kendaraan	Label Encoding
Andi	motor	0
Budi	mobil	1
Citra	sepeda	2
Dedi	motor	0
Eka	sepeda	2

Perhatikan pada variabel jenis kendaraan (“motor”, ”mobil”, “sepeda”) yang diubah menjadi numerik. Implementasikan *label encoding* dalam Python dengan langkah-langkah berikut.

Kode Program 4.5

```
01 import pandas as pd
02 from sklearn.preprocessing import LabelEncoder
03 # Data
04 data = {
05     'Nama': ['Andi', 'Budi', 'Citra', 'Dedi', 'Eka'],
06     'Kendaraan': ['Motor', 'Mobil', 'Sepeda', 'Motor', 'Sepeda']
07 }
08 # Buat DataFrame
09 df = pd.DataFrame(data)
10
11 # Label Encoding
12 encoder = LabelEncoder()
13 df['Kendaraan_Encoded'] =
14 encoder.fit_transform(df['Kendaraan'])
15
```

```
16 print(df)
```

- 1) Siapkan data kategorikal.
- 2) *Import* LabelEncoder dari *sklearn* pada *library* sklearn.preprocessing (baris 02).
- 3) Buat objek LabelEncoder (baris 12).
- 4) Terapkan `fit_transform()` ke data (baris 14).



Aktivitas 4.3

Labelku Angkaku: Memahami Rahasia Kategori dalam Kecerdasan Artifisial (Aktivitas Individu dan Kelompok)

- a. Amati kasus berikut dengan cermat.
Seorang guru di SMA melakukan survei sederhana kepada beberapa siswa kelas XI untuk mengetahui aplikasi *mobile* yang paling sering mereka gunakan. Lalu, diperoleh data berikut.

Nama	Aplikasi Belajar	Aplikasi Hiburan
Fajar	Google Classroom	YouTube
Gina	Zenius	TikTok
Hari	Ruangguru	Instagram
Indah	Zenius	YouTube
Rian	Google Classroom	TikTok

- b. Langkah 1: Latihan Koding Python

Kode Program 4.6

```
01 import pandas as pd
02 from sklearn.preprocessing import LabelEncoder
03
```

```

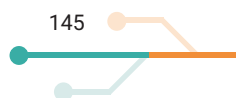
04 # Data siswa
05     data = {
06         'Nama': ['Fajar', 'Gina', 'Hari', 'Indah', 'Joko'],
07         'Aplikasi Belajar': ['Google Classroom', 'Zenius',
08                             'Ruangguru', 'Zenius', 'Google Classroom'],
09         'Aplikasi Hiburan': ['YouTube', 'TikTok', 'Instagram',
10                              'YouTube', 'TikTok']
11     }
12
13 df = pd.DataFrame(data)
14
15 # Membuat objek LabelEncoder
16 le = LabelEncoder()
17
18 # Encoding kolom Aplikasi Belajar
19 df['Aplikasi Belajar (Encoded)'] =
20     le.fit_transform(df['Aplikasi Belajar'])
21
22 # Encoding kolom Aplikasi Hiburan
23 df['Aplikasi Hiburan (Encoded)'] =
24     le.fit_transform(df['Aplikasi Hiburan'])
25
26 # Menampilkan hasil
27 print(df)

```

c. Langkah 2: Refleksi dan Diskusi

Lakukan diskusi dengan teman sebangkumu untuk menjawab pertanyaan-pertanyaan berikut.

- 1) Apakah hasil *encoding*-nya sudah menggambarkan hubungan antarkategori dan angka?
- 2) Apa risiko jika kamu salah menafsirkan hasil *label encoding* sebagai urutan nilai?



d. Langkah 3: Tantangan

Ubahlah data aplikasi belajar dan aplikasi hiburan sesuai dengan aplikasi yang biasa digunakan oleh teman-teman di kelasmu, lalu terapkan teknik *label encoding*.

Nah, sekarang kamu dapat berlatih *label encoding* dari *dataset* di Google Colab dengan langkah-langkah berikut.

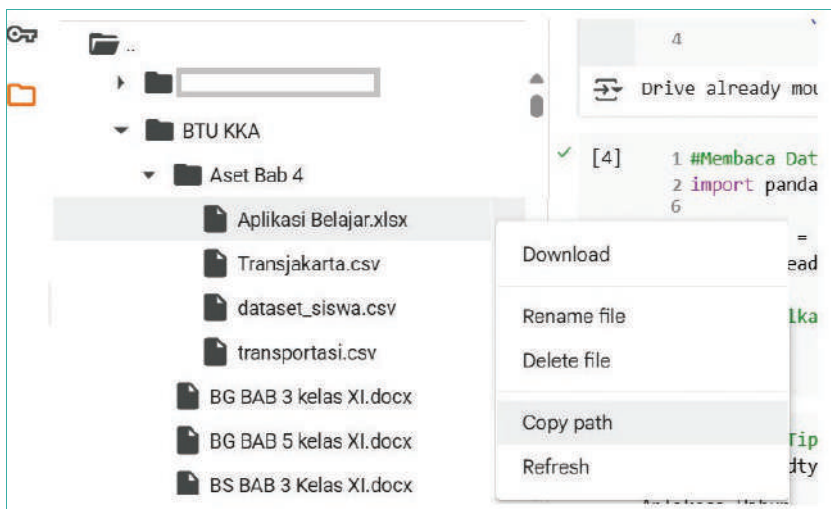
1) Akses *mounting* ke Google Drive.

Langkah ini digunakan untuk mengakses *file* yang ada di Google Drive dari Google Colab.

Kode Program 4.7

```
01 #Mount Google Drive
02 from google.colab import drive
03 drive.mount('/content/drive')
04
```

Setelah menjalankan perintah di atas, kamu dapat melanjutkan dengan *login* akun Google yang akan digunakan. Jika *mounting* berhasil, maka dapat dilanjutkan dengan menyalin *path* dari *file dataset* yang akan digunakan.



2) Membaca *dataset* dari Google Drive.

Kode Program 4.8

```
01 #Membaca Dataset dari Google Drive
02 import pandas as pd
03
04 file_path = '/content/drive/MyDrive/BTU KKA/Aset Bab
05 4/Aplikasi Belajar.xlsx'
06
07 df = pd.read_excel(file_path)
08
09 # Menampilkan 5 baris pertama
10 df.head()
```

Penjelasan:

- Meng-*import library* Pandas untuk mengelola data.
- Membaca *file* Excel dan menyimpannya ke dalam *dataframe* *df*.
- Menampilkan 5 baris pertama dari *dataset*.

Tabel 4.2 Tampilan Lima Baris Pertama pada *Dataset* dari Google Drive

	Nama	Aplikasi Belajar	Aplikasi Hiburan
0	Fajar	Google Classroom	YouTube
1	Gina	Zenius	Tik Tok
2	Hari	Ruangguru	Instagram
3	Indah	Zenius	YouTube
4	Rlan	Google Classroom	Tik Tok

3) Melihat tipe data setiap kolom.

```
01 #Melihat Tipe Data di Setiap Kolom
02 print(df.dtypes)
03
```

Penjelasan:

Menampilkan jenis tipe data untuk setiap kolom. Langkah ini membantumu mengetahui kolom mana yang bertipe *object* (biasanya teks/kategorikal), sehingga bisa diproses lebih lanjut dengan *label encoding*.

- 4) Melakukan *label encoding* pada kolom.

Kode Program 4.9

```
01 #Melakukan Label Encoding pada kolom
02 from sklearn.preprocessing import LabelEncoder
03
04 le = LabelEncoder()
05
06 # Duplikasi dataframe untuk mempertahankan data asli
07 df_encoded = df.copy()
08
09 # Kolom yang akan di-encode
10 kolom_encode = ['Nama', 'Aplikasi Belajar', 'Aplikasi Hiburan']
11
12 # Lakukan encoding per kolom
13 for kolom in kolom_encode:
14     df_encoded[kolom + '_encoded'] =
15         le.fit_transform(df_encoded[kolom])
16
17 # Tampilkan hasil
18 df_encoded[[*kolom_encode, *(k + '_encoded' for k in
19     kolom_encode)]].head()
```

Penjelasan:

- `LabelEncoder()` digunakan untuk mengubah data kategorikal menjadi angka.
- Dilakukan duplikasi *dataframe* agar data asli tidak berubah.
- Kolom yang dipilih untuk *encoding*: Nama, Aplikasi Belajar, Aplikasi Hiburan.
- Loop* akan melakukan transformasi teks menjadi angka untuk setiap kolom dan menyimpannya ke kolom baru (misalnya `Nama_encoded`).

Tabel 4.3 Contoh Transformasi Teks Menjadi Angka dalam *Label Encoding*

	Nama	Aplikasi Belajar	Aplikasi Hiburan	Nama_encoded	Aplikasi Belajar_encoded	Aplikasi Hiburan_encoded
0	Fajar	Google Classroom	YouTube	0	0	2
1	Gina	Zenius	Tik Tok	1	2	1
2	Hari	Ruangguru	Instagram	2	1	0
3	Indah	Zenius	YouTube	3	2	2
4	Rlan	Google Classroom	Tik Tok	4	0	1

- Melihat hasil *mapping label encoding*

Kode Program 4.10

```
01 #Melihat Mapping Label
02 le.fit(df['Aplikasi Belajar'])
03 mapping = dict(zip(le.classes_, le.transform(le.classes_)))
04 print(mapping)
05
```

- One Hot Encoding*

Salah satu teknik *encoding* yang paling populer adalah *one hot encoding*. Teknik ini cocok untuk data kategorikal nominal, seperti jenis kelamin atau warna. Teknik *one hot encoding* bertujuan untuk mengubah data kategorikal nominal menjadi bentuk *boolean (true/false)*, di mana setiap kategori unik akan diperluas menjadi kolom atau parameter baru. *One hot encoding* bekerja dengan membuat kolom baru untuk setiap kategori. Berikut langkah-langkah yang dilakukan pada teknik *one hot encoding*.

Langkah-Langkah *One Hot Encoding*



1. Identifikasi Kolom Kategori

Laki-laki

Perempuan

2. Temukan Semua Kategori Unik

Laki-laki

Olahraga

+

Musik

4. Isi Setiap Kolom dengan 0 dan 1

Nama	Hobi	Hobi Membaca
Andi	Musik	0
Budi	Membaca	1
Citra	Musik	0

3. Isi Setiap Kolom dengan 0 dan 1

Nama	Hobi	Hobi Membaca	Hobi Musik
Andi	Musik	0	1
Budi	Membaca	0	0
Citra	Musik	1	0

01
01

Gambar 4.3 Langkah Teknik *One Hot Encoding*



Aktivitas 4.4

Kasus Sehari-hari: Menentukan Warna Kaos Kelas (Aktivitas Individu)

a. Kasus

Ketua kelas ingin mendesain kaos untuk seragam kelas, dia ingin mengetahui warna favorit teman-temannya untuk mengambil keputusan penentuan warna desain kaos seragam kelas. Data survei yang diperoleh sebagai berikut.

Nama	Warna
Andi	merah

Nama	Warna
Budi	hijau
Cici	biru
Dedi	hijau

b. Pertanyaan Analisis

- 1) Dari *dataset* tersebut, variabel apa yang perlu di-*encode*? Jelaskan alasannya!

- 2) Jika kamu menggunakan *one hot encoding* pada kolom “warna”, berapa kolom baru yang terbentuk?

- 3) Apakah kolom “Nama” perlu di-*encode*? Mengapa?

c. Langkah-Langkah Praktik Python

Gunakan *library* Pandas dan `get_dummies` untuk *one hot Encoding*

Kode Program 4.11

Python

Menggunakan *Library* Pandas dan `get_dummies`

```

01 import pandas as pd
02
03 # Data awal
04 data = {'Nama': ['Andi', 'Budi', 'Cici', 'Dedi'], 'Warna':
05         ['Merah', 'Hijau', 'Biru', 'Hijau']}
06         df = pd.DataFrame(data)
07 # Melakukan One Hot Encoding

```

```

08 df_encoded = pd.get_dummies(df, columns=['Warna'])
09 # Mengubah hanya kolom hasil encoding (True/False) menjadi 1/0
10 for col in df_encoded.columns:
11     if col.startswith('Warna_'):
12         # Hanya kolom hasil encoding yang diubah
13         df_encoded[col] = df_encoded[col].astype(int)
14 # Menampilkan hasil
15 print(df_encoded)
16

```

Output Program

	Nama	Warna Biru	Warna Hijau	Warna Merah
0	Andi	0	0	1
1	Budi	0	1	0
2	Cici	1	0	0
3	Dedi	0	1	0

Setelah berhasil mengerjakan Aktivitas 4.4, kamu dapat berlatih dengan studi kasus di bidang lain untuk memperkuat pemahamanmu tentang *one hot encoding*.



Aktivitas 4.5

Data Bercerita: Analisis Produk Favorit Pelanggan dengan *One Hot Encoding* (Aktivitas Individu)

- a. Amati kasus berikut dengan cermat.

Toko *online* ABC ingin menganalisis kecenderungan pemilihan produk yang sering dibeli pelanggan. *Dataset* dikumpulkan melalui aplikasi kasir yang berisi informasi tentang ID pelanggan, kategori produk (elektronika, pakaian, makanan, dan lain-lain), dan frekuensi pembelian. Toko ini ingin menggunakan data tersebut untuk proses berikut.

 - 1) mengidentifikasi kategori produk yang paling populer;
 - 2) membuat rekomendasi produk yang dipersonalisasi; dan

3) mengoptimalkan strategi pemasaran dan penempatan iklan.

Inilah contoh/sampel *dataset* yang diperoleh.

ID Pelanggan	Kategori Produk	Frekuensi Pembelian
001	elektronik	5
002	pakaian	3
003	makanan	7
004	elektronik	2
005	pakaian	4
006	makanan	6

b. Lakukan langkah-langkah berikut untuk *one hot encoding*.

- 1) Dari *dataset* yang tersedia, tentukan jenis variabel yang dapat dilakukan *encode*!
- 2) Buatlah tabel untuk hasil *one hot encoding*-nya! Kamu dapat membuat tabel menggunakan Python, Spreadsheet, atau di lembar kertas.
- 3) Bagaimana hasil analisis data dari *dataset* sesuai dengan kebutuhan toko ABC?
 - a) Identifikasi kategori populer
 - b) Rekomendasi produk
 - c) Strategi *marketing* dan penempatan iklan

c. Langkah praktik menggunakan Python.

Untuk menjawab permasalahan Toko ABC, kamu dapat melakukan *one hot encoding* pada *dataset* yang diperoleh dengan perintah Python. *Library* yang akan digunakan adalah Pandas dan Scikit-learn dengan perintah:

```
pip install pandas scikit-learn
```

d. Implementasikan *one hot encoding* dan analisis data.

Kode Program 4.12

```
01 import pandas as pd
02 from sklearn.preprocessing import OneHotEncoder
03
04 # Data contoh
05 data = {
06     'ID Pelanggan': ['001', '002', '003', '004', '005', '006'],
07     'Kategori Produk': ['Elektronik', 'Pakaian', 'Makanan', 'Elektronik',
08     'Pakaian', 'Makanan'],
09     'Frekuensi Pembelian': [5, 3, 7, 2, 4, 6]
10 }
11 # Membuat DataFrame
12 df = pd.DataFrame(data)
13
14 # Menampilkan dataset awal
15 print("Dataset Awal:")
16 print(df)
17
18 # Membuat objek OneHotEncoder
19 encoder = OneHotEncoder(sparse_output=False)
20 # sparse_output=False untuk mendapatkan array biasa
21 # Melakukan encoding pada kolom 'Kategori Produk'
22 encoded_data = encoder.fit_transform(df[['Kategori Produk']])
23
24 # Membuat DataFrame dari hasil encoding
25 encoded_df = pd.DataFrame(encoded_data,
26     columns=encoder.get_feature_names_out(['Kategori Produk']))
27
28 # Menggabungkan DataFrame awal dengan hasil encoding
29 df_final = pd.concat([df, encoded_df], axis=1)
```

```

30 # Menghapus kolom 'Kategori Produk' asli karena sudah di-encode
31 df_final = df_final.drop('Kategori Produk', axis=1)
32
33 # Menampilkan dataset setelah One Hot Encoding
34 print("\nDataset Setelah One Hot Encoding:")
35 print(df_final)
36
37 # Analisis sederhana: Menghitung total frekuensi pembelian per kategori
38 total_frekuensi = df_final.groupby(['Kategori Produk_Elektronik',
39 'Kategori Produk_Pakaian', 'Kategori Produk_Makanan']).sum()
39 print("\nTotal Frekuensi Pembelian per Kategori:")
40 print(total_frekuensi)

```

Hasil Output Program

1) *Dataset awal*

Dataset Awal:

	ID Pelanggan	Kategori Produk	Frekuensi Pembelian
0	001	Elektronik	5
1	002	Pakaian	3
2	003	Makanan	7
3	004	Elektronik	2
4	005	Pakaian	4
5	006	Makanan	6

2) *Dataset setelah one hot encoding*

Dataset Setelah One Hot Encoding:

	ID Pelanggan	Frekuensi Pembelian	Kategori Produk_Elektronik \
0	001	5	1.0
1	002	3	0.0
2	003	7	0.0
3	004	2	1.0
4	005	4	0.0
5	006	6	0.0

	Kategori Produk_Makanan	Kategori Produk_Pakaian
0	0.0	0.0
1	0.0	1.0
2	1.0	0.0
3	0.0	0.0
4	0.0	1.0
5	1.0	0.0

e. Analisis total frekuensi pembelian per kategori.

Total Frekuensi Pembelian per Kategori:

			ID Pelanggan \
Kategori Produk_Elektronik	Kategori Produk_Pakaian	Kategori Produk_Makanan	
00	0.0	1.0	003006
	1.0	0.0	002005
1.0	0.0	0.0	001004
			Frekuensi Pembelian
Kategori Produk_Elektronik	Kategori Produk_Pakaian	Kategori Produk_Makanan	
0.0	0.0	1.0	13
	1.0	0.0	7
1.0	0.0	0.0	7

Hasil analisis:

- 1) [1.0, 0.0, 0.0] = elektronik → total pembelian = 5 (ID 001) + 2 (ID 004) = 7
- 1) [0.0, 1.0, 0.0] = pakaian → total = 3 (ID 002) + 4 (ID 005) = 7
- 2) [0.0, 0.0, 1.0] = makanan → total = 7 (ID 003) + 6 (ID 006) = 13

Terlihat hasil total frekuensi pembelian makanan = 13; elektronik = 7; dan pakaian = 7. Dapat disimpulkan bahwa produk makanan adalah kategori paling populer dengan frekuensi pembelian tertinggi. Kategori elektronik dan pakaian berada di peringkat kedua dengan frekuensi yang sama. Berdasarkan popularitas tersebut, Toko ABC direkomendasikan berfokus pada produk makanan karena minat dari konsumen yang tinggi. Untuk pelanggan, sistem dapat menyarankan paket *bundling* makanan populer atau diskon untuk pembelian makanan berulang.

Strategi *marketing* dan penempatan iklan untuk Toko ABC dapat direkomendasikan pada iklan makanan dengan promo “Beli 2, Gratis 1” atau pemberian diskon jika jumlah pembelian banyak. Untuk segmentasi pasar, sistem dapat mengirimkan *email marketing* rutin dan *flash sale* untuk para pelanggan. Untuk produk elektronik dan pakaian dapat menggunakan target pada momen tertentu, misalnya saat akhir tahun atau hari raya.

Perlu kamu ketahui!

Jika jumlah kategori kurang dari 10, maka *one hot encoding* adalah pilihan yang tepat (Aurélien Géron, 2019). *One hot encoding* cocok untuk kategori yang sedikit karena tidak menambah terlalu banyak dimensi data.

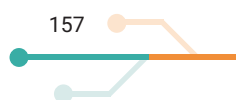
c. *Binary Encoding*

Selain kedua teknik di atas, kamu juga perlu mempelajari teknik *encoding* lainnya, yaitu *binary encoding*. Teknik ini merupakan kombinasi antara *label encoding* dan sistem bilangan biner (0 dan 1) yang setiap angkanya disebut dengan bit. Teknik *binary encoding* digunakan jika data memiliki banyak jenis kategori, seperti nama produk atau jenis pekerjaan.

Cara mengimplementasikan teknik *binary encoding*, yaitu setiap kategori akan diubah terlebih dahulu menjadi angka (1, 2, 3, dan seterusnya). Kemudian, angka tersebut akan dikonversi ke dalam bentuk bilangan biner, misalnya angka 3 menjadi 11, angka 4 menjadi 100, dan seterusnya. Teknik ini lebih efisien jika dibandingkan dengan *one hot encoding* karena tidak membuat banyak kolom.

Untuk mengingat kembali materi konversi bilangan biner, kerjakan latihan berikut.

- 1) Cobalah kamu konversikan bilangan desimal ini ke dalam bentuk biner: 2, 7, 9, 10, 15.
- 2) Presentasikan hasil konversi dari bilangan tersebut di depan kelas secara bergantian.

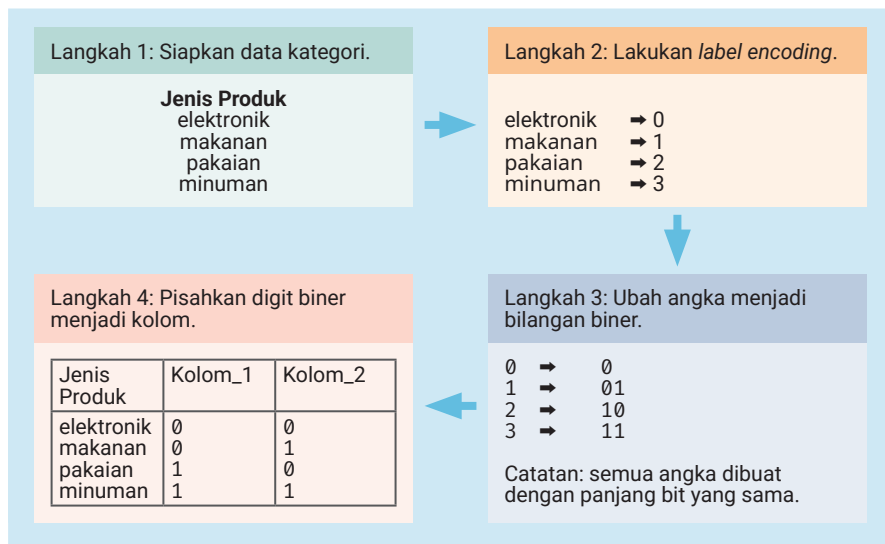


Agar kamu lebih memahami teknik *binary encoding*, perhatikan contoh berikut. Misalnya, sebuah kolom jenis produk dengan empat kategori.

Tabel 4.4 Contoh Teknik *Binary Encoding*

ID	Jenis Produk
1	Elektronik
2	Makanan
3	Pakaian
4	Minuman

Berikut langkah pengerjaan untuk teknik *binary encoding*.



Gambar 4.4 Langkah *Binary Encoding*

Sehingga, tabel hasil *binary encoding*-nya menjadi seperti berikut.

Tabel 4.5 Contoh Hasil Teknik *Binary Encoding*

Jenis Produk	Label	Biner	Kolom 1	Kolom 2
elektronik	0	00	0	0
makanan	1	01	0	1
pakaian	2	10	1	0
minuman	3	11	1	1



Aktivitas 4.6

Aktivitas 4.6 Mari Menganalisis (Aktivitas Individu)

- a. Amati kasus berikut dengan saksama.

Sebuah klinik kesehatan digital sedang mengembangkan sistem untuk mengklasifikasi pasien berdasarkan gejala penyakit yang dialami, seperti batuk, demam, mual, pusing, dan sesak napas. Sistem membutuhkan data dalam bentuk numerik biner untuk proses deteksi awal penyakit. Tugasmu adalah melakukan teknik *binary encoding* pada data gejala.

- b. Penyelesaian menggunakan Excel

ID Pasien	Gejala	Label	Biner	Gejala_0	Gejala_1	Gejala_2
A001	batuk	1	001	0	0	1
A002	demam	2	010	0	1	0
A003	mual	3	011	0	0	1
A004	pusing	4	100	1	1	0
A005	sesak nafas	5	101	1	0	1

Binary encoding mengubah data dari label menjadi bilangan biner. Karena kategori gejala unik ada 5, maka digunakan 3 bit ($2^3 = 8$ cukup untuk menampung 5 kategori).

- c. Menggunakan Phyton

Menginstal *library category_encoders* dengan perintah:

```
01 !pip install category_encoders
```

- d. Kemudian dilanjutkan dengan *binary encoding*.

Kode Program 4.13

```
01 import pandas as pd
02 import category_encoders as ce
03
```

```

04 # Data Gejala Pasien
05 data = pd.DataFrame({
06     'ID Pasien': ['A01', 'A02', 'A03', 'A04', 'A05'],
07     'Gejala': ['Batuk', 'Demam', 'Mual', 'Pusing', 'Sesak Nafas']
08 })
09
10 # Binary Encoding
11 encoder = ce.BinaryEncoder(cols=['Gejala'])
12 encoded = encoder.fit_transform(data[['Gejala']])
13 df_final = pd.concat([data, encoded], axis=1)
14
15 print("Data setelah Binary Encoding:")
16 print(df_final)
17

```

d. *Ordinal Encoding*

Ordinal encoding merupakan salah satu metode *encoding* untuk mengubah data kategorikal berdasarkan urutan atau tingkatan tertentu yang dimiliki data tersebut. Maksudnya adalah setiap nilai kategori memiliki posisi tertentu yang lebih tinggi atau lebih rendah. Berikut karakteristik yang dimiliki oleh *ordinal encoding*.

- 1) Data berupa Kategori yang memiliki tingkatan atau *ranking*.
- 2) Setiap kategori akan diberikan nilai berupa angka berdasarkan urutan.
- 3) Angka yang dihasilkan mewakili urutan, bukan nilai sebenarnya.

Contoh data:

- 1) Tingkat kepuasan pelanggan (rendah, sedang, tinggi)
- 2) Ukuran baju (S/M/L/XL)
- 3) Jenjang pendidikan (SD, SMP, SMA, S1, S2)

Ordinal encoding dapat membantu sistem pembelajaran mesin memahami bahwa kategori tersebut memiliki nilai yang meningkat atau menurun. Hal ini membuat prediksi atau analisis data menjadi lebih akurat dan bermakna.



Aktivitas 4.7

Studi Kasus: Menentukan Kualitas Tanah untuk Rekomendasi Jenis Tanaman (Aktivitas Individu)

a. Amatilah kasus berikut dengan cermat.

Sebuah kelompok tani di Desa Sukamakmur bekerja sama dengan balai penyuluh pertanian digital untuk merekomendasikan jenis tanaman yang sesuai dengan kondisi kualitas tanah. Klasifikasi tanah berupa data ordinal menjadi: Buruk, Cukup, Baik, dan Sangat Baik. Data ini akan digunakan dalam sistem prediksi berbasis *machine learning* untuk merekomendasikan jenis tanaman, seperti jagung, padi, atau sayuran. Berikut contoh *dataset* yang diperoleh.

Lokasi Lahan	Kualitas Tanah	Kelembaban (%)	Rekomendasi Tanaman
Lahan 1	Baik	70	padi
Lahan 2	Cukup	55	jagung
Lahan 3	Sangat Baik	80	padi
Lahan 4	Buruk	40	tidak direkomendasikan
Lahan 5	Cukup	60	sayuran

b. *Ordinal encoding* untuk kualitas tanah

Kualitas Tanah	Kode
Buruk	1
Cukup	2
Baik	3
Sangat Baik	4

c. Implementasi Phyton

Kode Program 4.14

```
01 import pandas as pd
02 from sklearn.preprocessing import OrdinalEncoder
03
04 # Membuat Dataset
05 data = pd.DataFrame({
06     'Kualitas Tanah': ['Baik', 'Cukup', 'Sangat Baik', 'Buruk', 'Cukup'],
07     'Kelembapan': [70, 55, 80, 40, 60],
08     'Rekomendasi Tanaman': ['Padi', 'Jagung', 'Padi', 'Tidak
Direkomendasikan', 'Sayuran']
09 })
10
11 # Ordinal Encoding
12 encoder = OrdinalEncoder(categories=[['Buruk', 'Cukup', 'Baik', 'Sangat
Baik']])
13 data['Kualitas Tanah (Angka)'] = encoder.fit_transform(data[['Kualitas
Tanah']])
14
15 print(data)
16
```

d. Penjelasan

Ordinal encoder dari *sklearn.preprocessing* digunakan untuk

- 1) mengubah data kategori berurutan (ordinal) menjadi angka (baris 02).
- 2) Membuat objek *encoder* dan mendefinisikan urutan kategori dari yang terendah ke tertinggi (baris 11–13).

e. Target Encoding

Target encoding adalah salah satu teknik pengolahan data yang digunakan untuk mengubah data kategori menjadi angka dengan cara yang lebih cerdas. Biasanya, ketika kita memiliki data yang berupa kategori, seperti warna atau jenis kelamin, kita perlu mengubahnya menjadi angka supaya

komputer bisa memahaminya. *Target encoding* mengubah kategori menjadi angka berdasarkan nilai rata-rata dari target atau hasil yang ingin kita prediksi. Misalnya, kamu mempunyai data dari budi daya ikan. Ada beberapa jenis pakan yang digunakan, yaitu pakan A, pakan B, dan pakan C. Tujuannya adalah memprediksi apakah ikan tumbuh dengan sehat atau tidak.

Nah, kamu dapat menggunakan *target encoding* untuk mengubah jenis pakan (yang berupa kategori) menjadi angka berdasarkan seberapa sering pakan itu menghasilkan ikan yang sehat.

- 1) Contoh data setelah dicek.
 - a) Pakan A menghasilkan 90% ikan sehat → dikodekan menjadi 0,9.
 - b) Pakan B menghasilkan 70% ikan sehat → dikodekan menjadi 0,7.
 - c) Pakan C menghasilkan 50% ikan sehat → dikodekan menjadi 0,5.

Jadi, daripada menuliskan “Pakan A”, “Pakan B”, atau “Pakan C”, kamu dapat menulis 0,9, 0,7, atau 0,5 agar komputer dapat mengenali hubungan antara pakan dan kesehatan ikan. Dengan begitu, komputer dapat memahami mana pakan ikan yang lebih efektif berdasarkan angka ini ketika membuat prediksi. Dengan cara ini, *target encoding* membantu komputer menangkap hubungan antara kategori dan hasilnya secara lebih tepat dibandingkan dengan cara mengubah kategori menjadi angka biasa.

- 2) Dari contoh di atas, jika kita asumsikan, maka *datasetnya* sebagai berikut.

Tabel 4.6 Contoh *Dataset* pada Teknik *Target Encoding*

No.	Jenis Pakan	Ikan Sehat (1 = Sehat, 0 = Tidak Sehat)
1	Pakan A	1
2	Pakan A	1
3	Pakan A	0
4	Pakan B	1
5	Pakan B	0

No.	Jenis Pakan	Ikan Sehat (1 = Sehat, 0 = Tidak Sehat)
6	Pakan B	0
7	Pakan C	1
8	Pakan C	0

3) Langkah-langkah menerapkan *target encoding*.

a) Kelompokkan data berdasarkan kategori jenis pakan.

- Pakan A: 3 Data $\rightarrow [1,1,0]$
- Pakan B: 3 Data $\rightarrow [1,0,0]$
- Pakan C: 2 Data $\rightarrow [1,0]$

b) Hitung rata-rata nilai target (ikan sehat/tidak sehat) untuk masing-masing pakan.

- Pakan A: $(1 + 1 + 0)/3 = \frac{2}{3} = 0,67$
- Pakan B: $(1 + 0 + 0)/3 = \frac{1}{3} = 0,33$
- Pakan C: $(1 + 0)/2 = \frac{1}{2} = 0,50$

c) Ganti Jenis pakan dengan nilai hasil *encoding*.

Tabel 4.7 Contoh Nilai Hasil *Encoding* pada Teknik *Target Encoding*

No.	Jenis Pakan	Target Encoding
1	Pakan A	0,67
2	Pakan A	0,67
3	Pakan A	0,67
4	Pakan B	0,33
5	Pakan B	0,33
6	Pakan B	0,33
7	Pakan C	0,50

No.	Jenis Pakan	Target Encoding
8	Pakan C	0,50

Dengan *target encoding*, kamu sudah berhasil mengubah jenis pakan (data kategori) menjadi angka berdasarkan pengaruhnya terhadap kesehatan ikan. Angka tersebut selanjutnya dapat digunakan sebagai input untuk model prediksi atau algoritma pembelajaran mesin agar dapat mengambil keputusan dengan lebih akurat.

4) Implementasi menggunakan Phyton.

Kode Program 4.15

```

01 import Pandal as pd
02 .
03 # 1. Membuat data
04 data = {
05     'Jenis_Pakan': ['Pakan A', 'Pakan A', 'Pakan A',
06                   'Pakan B', 'Pakan B', 'Pakan B',
07                   'Pakan C', 'Pakan C'],
08     'Ikan_Sehat': [1, 1, 0, 1, 0, 0, 1, 0]
09 }
10 .
11 df = pd.DataFrame(data)
12 .
13 # 2. Menampilkan data awal
14 print("Data Awal:")
15 print(df)
16 .
17 # 3. Melakukan Target Encoding
18 encoding_map = df.groupby('Jenis_Pakan')['Ikan_Sehat'].mean()
19 df['Pakan_Encoded'] = df['Jenis_Pakan'].map(encoding_map)
20 .
21 # 4. Menampilkan hasil encoding
22 print("\nData Setelah Target Encoding:")
23 print(df)
24 .

```

5) Penjelasan mengenai program.

Kelompokkan data berdasarkan jenis pakan menggunakan `groupby()` dan hitung rata-rata nilai target (Ikan_Sehat) per jenis pakan dengan `.mean()` (baris 18-19). Maka, hasil yang diperoleh:

- a) Pakan A $\rightarrow (1 + 1 + 0)/3 = 0.67$
- b) Pakan B $\rightarrow (1 + 0 + 0)/3 = 0.33$
- c) Pakan C $\rightarrow (1 + 0)/2 = 0.50$

Hasil ini disimpan di `encoding_map`.



Aktivitas 4.8

Data Menjadi Dokter: Menganalisis Efektivitas Obat Flu dengan Target Encoding (Aktivitas Individu)

a. Amati kasus berikut dengan saksama.

Sebuah klinik memiliki data rekam medis dari 8 pasien yang diberikan obat A, obat B, dan obat C untuk mengobati flu. Klinik mencatat pasien sembuh (1) atau pasien tidak sembuh (0) setelah mengonsumsi obat.

Untuk membuat sistem rekomendasi obat terbaik, dilakukan teknik *target encoding* terhadap kolom nama obat agar model atau analis dapat mengenali obat mana yang paling efektif berdasarkan data rekam medis. Berikut data rekam medis berdasarkan kasus tersebut.

Data Rekam Medis

No.	Nama Obat	Sembuh
1	Obat A	1
2	Obat A	1
3	Obat A	0
4	Obat B	1
5	Obat B	0
6	Obat B	0

No.	Nama Obat	Sembuh
7	Obat C	0
8	Obat C	1

b. Buatlah langkah-langkah analisis *target encoding*.

c. Hitunglah rata-rata tingkat kesembuhan pasien.

Rumus:

Rata-Rata = Total Pasien/Jumlah Pasien Sembuh

Nama Obat	Jumlah Pasien	Pasien Sembuh	Rata-Rata Kesembuhan
Obat A			
Obat B			
Obat C			

d. Berikan rekomendasi dari hasil analisis.

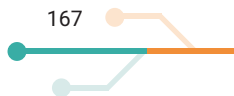
Berdasarkan rata-rata tersebut, tentukanlah hal-hal berikut.

1) Obat mana yang memiliki tingkat kesembuhan yang tinggi?

2) Obat mana yang sebaiknya direkomendasikan untuk pasien?

e. Refleksi

Tuliskan simpulanmu berdasarkan analisis kasus tersebut tentang manfaat *target encoding* dalam pengambilan keputusan!



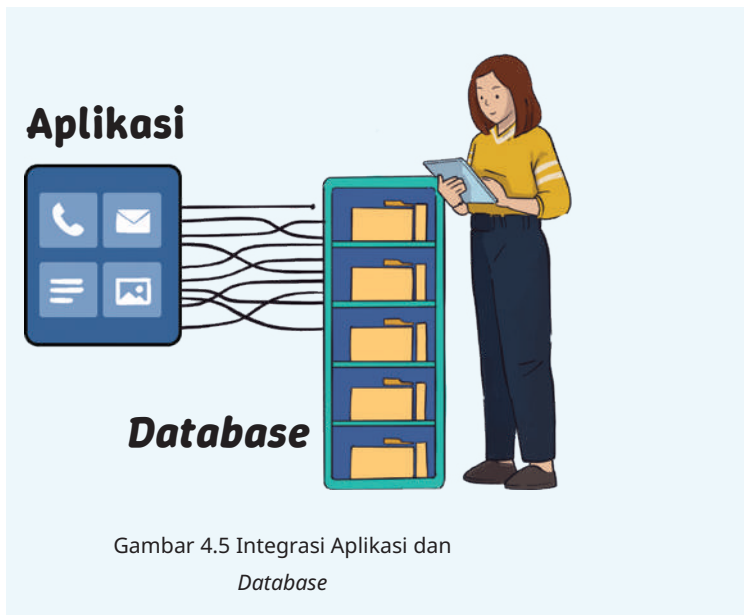
C. Integrasi Aplikasi dan Database

Integrasi aplikasi dengan basis data adalah proses menghubungkan aplikasi dengan sistem penyimpanan data yang disebut dengan *database*. Hal ini bertujuan agar aplikasi dapat menyimpan, mengambil, meng-*update*, atau menghapus data secara dinamis. Kamu dapat membayangkan *database* seperti lemari penyimpanan yang diatur dan teratur. Sedangkan, aplikasi adalah orang yang menggunakan lemari tersebut untuk menyimpan dan mengambil barang dari rak lemari tersebut.

Dengan adanya integrasi aplikasi dan *database* ini, keduanya dapat “berkomunikasi” ke *database* untuk melakukan operasi-operasi sesuai dengan kebutuhan.

Penting untuk menghubungkan aplikasi dengan *database* dalam proses menyimpan dan mengambil data. Mengapa hal ini penting? Karena aplikasi membutuhkan tempat untuk penyimpanan dan mengelola data agar dapat berfungsi dengan baik. Misalnya, aplikasi di sebuah toko *online* membutuhkan penyimpanan data produk, pelanggan, dan transaksi. Data-data ini akan disimpan dalam *database* dengan rapi dan akan digunakan oleh aplikasi untuk berbagai operasi dalam menampilkan informasi. Dengan adanya integrasi aplikasi dan *database*, maka aplikasi akan bekerja dengan pintar dan efisien, contohnya untuk menampilkan stok barang yang selalu *update* dan mencatat transaksi-transaksi pembelian. Berikut contoh penerapan integrasi aplikasi dan *database*.

- Aplikasi toko *online* yang menyimpan data produk, pelanggan, dan transaksi.
- Aplikasi absensi siswa yang menyimpan data kehadiran.



Gambar 4.5 Integrasi Aplikasi dan Database

1. Konsep *Database* dan Relasional

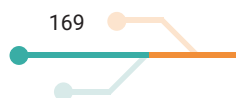
Pernahkah kamu menyimpan daftar nomor *handphone* di kontak? atau daftar nama siswa di sekolah? Kira-kira, seperti itulah yang disebut sebagai contoh penggunaan *database*. *Database* adalah tempat menyimpan data atau informasi secara rapi, sehingga mudah untuk dicari, ditambah, diubah, dan dihapus kapan saja. Silakan kamu cari lagi contoh penggunaan *database* dalam kehidupan sehari-hari.

Pengetahuan ini tentu sudah tidak asing lagi bagimu. Di kelas X, kamu sudah mempelajari tentang dasar-dasar *database* dan juga cara membuat serta mengelola data menggunakan perangkat lunak, seperti MySQL atau SQLite. Konsep dasar seperti tabel, *field*, dan *record*. Operasi dasar dengan perintah dasar SQL, di antaranya:

- **SELECT** : menampilkan data
- **INSERT INTO** : menambahkan data
- **UPDATE** : mengubah data
- **DELETE** : menghapus data
- **JOIN** : menggabungkan tabel

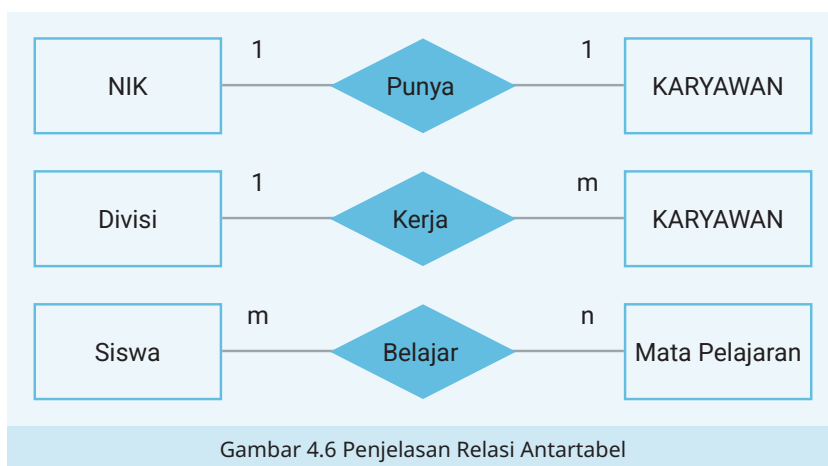
Nah, pada pembelajaran kali ini, kamu akan melanjutkan pemahaman tersebut dengan menggali lebih dalam tentang model data dan bagaimana relasi antartabel berperan penting dalam menyusun sistem *database* yang efisien dan terstruktur. Sekarang kita lanjutkan dengan model data yang merupakan cara untuk menyusun data dengan tujuan agar data lebih mudah dikelola. Dapat kamu bayangkan bahwa model data adalah seperti cara kita menyusun rak buku di perpustakaan. Terdapat tiga model data yang perlu kamu ketahui.

- a. Model Hierarki, yaitu data disusun secara bertingkat seperti pohon.
- b. Model Jaringan, yaitu data saling terhubung seperti jaring.
- c. Model Relasional, yaitu data disimpan dalam bentuk tabel yang saling terhubung.



Dalam model relasional, hubungan antara satu tabel dengan tabel yang lain disebut sebagai relasi antartabel. Adapun jenis relasi antartabel terdiri dari:

- 1) *One to One*: Satu data di tabel A hanya berhubungan dengan satu data di tabel B, misalnya satu siswa memiliki satu kartu identitas.
- 2) *One to Many*: Satu data di tabel A dapat berhubungan dengan banyak data di tabel B, misalnya satu guru mengajar banyak kelas.
- 3) *Many to Many*: Banyak data di tabel A dapat berhubungan dengan banyak data di tabel B (biasanya memerlukan tabel relasi tambahan), misalnya siswa bisa mengikuti banyak ekstrakurikuler dan setiap ekstrakurikuler diikuti oleh banyak siswa.



Dengan memahami model relasional dan relasi antartabel, kamu dapat merancang sistem *database* yang efisien untuk berbagai kebutuhan nyata, seperti sistem peminjaman buku, manajemen kehadiran, atau bahkan pengelolaan data penjualan di toko *online*. *Database* merupakan jantung dari banyak aplikasi. Aplikasi mengirim permintaan ke *database* untuk menyimpan data input pengguna, menampilkan informasi yang diminta, serta meng-*update* dan menghapus data sesuai kebutuhan.



Aktivitas 4.9

Mendesain Relasi Sistem Rawat Inap di Rumah Sakit (Aktivitas Kelompok)

a. Amati kasus berikut dengan saksama.

Bayangkan jika kalian menjadi bagian dari tim IT di sebuah rumah sakit yang ingin membangun sistem rawat inap. Sistem ini memiliki fitur mencatat informasi pasien, dokter, dan kamar rawat inap serta keterkaitannya. Kalian akan bekerja dengan tiga tabel utama berikut.

- 1) Tabel Pasien (**id_pasien** (Primary Key), nama, usia, diagnosis)
- 2) Tabel Dokter (**id_dokter** (Primary Key), nama_dokter, spesialis)
- 3) Tabel Rawat Inap (**id_rawat**, **id_pasien** (Foreign Key), **id_dokter** (Foreign Key), tanggal_masuk, kamar)

Adapun ketentuan yang diminta oleh rumah sakit, yaitu:

- 1) Seorang pasien bisa dirawat lebih dari satu kali di rumah sakit.
- 2) Seorang dokter bisa menangani banyak pasien yang dirawat.
- 3) Setiap data rawat inap hanya berkaitan dengan satu pasien dan satu dokter.

b. Bentuklah sebuah kelompok yang terdiri atas 4–5 orang untuk melakukan tahapan aktivitas berikut.

1) Langkah 1: Identifikasi relasi antartabel.

- a) Pasien dan rawat inap: _____
- b) Dokter dan rawat inap: _____
- c) Apakah dibutuhkan tabel relasi tambahan untuk sistem ini? Jelaskan alasannya! _____

2) Langkah 2: Gambar skema relasi tabel.

Gambarlah skema sederhana (diagram) yang menunjukkan hubungan antara ketiga tabel. Tandai *Primary Key* (PK) dan *Foreign Key* (FK). Gunakan panah atau simbol relasi (*One to Many*) untuk menunjukkan arah hubungan.

3) Langkah 3: Jalankan perintah SQL untuk membuat tabel, pengisian data, dan *Query* menggunakan SQL editor yang telah kalian pelajari.

Membuat Tabel

```
CREATE TABLE pasien (  
    id_pasien INTEGER PRIMARY KEY,  
    nama TEXT,  
    usia INTEGER,  
    diagnosa TEXT  
);  
  
CREATE TABLE dokter (  
    id_dokter INTEGER PRIMARY KEY,  
    nama_dokter TEXT,  
    spesialis TEXT  
);  
  
CREATE TABLE rawat_inap (  
    id_rawat INTEGER PRIMARY KEY,  
    id_pasien INTEGER,  
    id_dokter INTEGER,  
    tanggal_masuk DATE,  
    kamar TEXT,  
    FOREIGN KEY (id_pasien) REFERENCES pasien(id_pasien),  
    FOREIGN KEY (id_dokter) REFERENCES dokter(id_dokter)  
);
```

Menambahkan data masing-masing (2–3 data)

```
SELECT pasien.nama, dokter.nama_dokter, rawat_inap.kamar  
FROM rawat_inap  
JOIN pasien ON rawat_inap.id_pasien = pasien.id_pasien  
JOIN dokter ON rawat_inap.id_dokter = dokter.id_dokter;
```

- 4) Langkah 4: Lakukan refleksi dan diskusi.
Berdiskusilah dengan teman sekelompok kalian untuk menjawab pertanyaan-pertanyaan berikut.
 - a) Mengapa kalian perlu membuat relasi antartabel?

- b) Apa kelebihan menyimpan data pasien dan rawat inap dalam sistem relasional?
- c) Bagaimana sistem ini membantu perawat atau petugas administrasi rumah sakit?

Pada Aktivitas 4.9 ini, kalian akan diberikan aspek penilaian dengan ketentuan sebagai berikut.

- a. Identifikasi dan Penjelasan Relasi Tabel (25%)
- b. Pembuatan Struktur dan Diagram ERD (25%)
- c. Penulisan dan Eksekusi *Query* SQL (30%)
- d. Refleksi dan Pemahaman Kontekstual (20%)

2. Koneksi Database

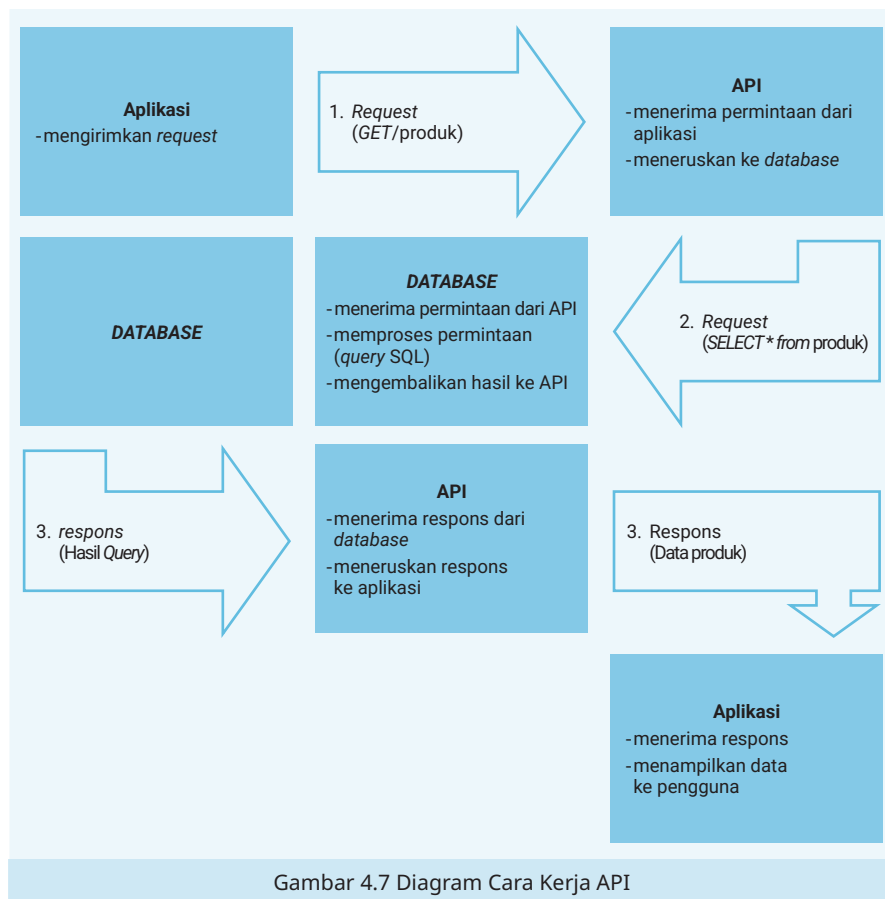
Setelah memahami pentingnya integrasi aplikasi dan *database*, sekarang kita lanjutkan untuk membahas dua cara yang digunakan untuk komunikasi antara *database* dan aplikasi, yaitu penggunaan API dan koneksi *database*.

a. Penggunaan API

API memungkinkan aplikasi untuk berkomunikasi dengan *database*. Apa itu API? API singkatan dari *Application Programming Interface*, yaitu antarmuka yang dapat menghubungkan aplikasi untuk berkomunikasi dengan sistem lain, termasuk *database*. Dengan kata lain, API adalah perantara aplikasi dengan *database*. Kamu dapat bayangkan API sebagai seorang penerjemah yang membantu dua orang yang berkomunikasi dengan perbedaan bahasa.

Dalam konteks aplikasi dan *database*, API memungkinkan aplikasi untuk mengirimkan permintaan (*request*) ke *database* dan menerima respons (*response*) yang berisi data yang dibutuhkan. Misalnya, ketika kamu menggunakan aplikasi *e-commerce*, API membantu aplikasi tersebut mengambil data produk dari *database* untuk ditampilkan di layarmu. Proses ini dapat terjadi dengan cepat dan efisien, sehingga kamu sebagai pengguna dapat melihat informasi yang diinginkan tanpa perlu mengetahui cara kerja di balik layar. Dengan API, komunikasi

antara aplikasi dan *database* menjadi lebih terstruktur dan mudah dikelola. Tentunya, dengan cara seperti ini aplikasi menjadi lebih ringan.



Berikut penjelasan mengenai diagram cara kerja API.

- 1) Aplikasi mengirimkan permintaan/*request* ke API atau langsung ke *database*. Contohnya, aplikasi *e-commerce* mengirimkan permintaan untuk mengambil data daftar produk. *Request* dapat berupa *select* (mengambil data), *insert* (menyimpan data), *update* (meng-update data), atau *delete* (menghapus data).
- 2) API atau *database* memproses permintaan aplikasi. Jika API digunakan, maka API akan meneruskan permintaan ke *database*. Jika langsung ke *database*, maka aplikasi mengirimkan *query SQL* ke *database*.

- 3) Hasil pemrosesan akan dikembalikan ke aplikasi dengan sebutan *respons*. *Database* mengembalikan hasil *respons* ke API dan kemudian diteruskan ke aplikasi. Aplikasi menerima *respons* dan menampilkan informasi ke pengguna. Sebagai contoh, aplikasi *e-commerce* menampilkan data daftar produk yang sesuai dengan pencarian.

b. Koneksi *Database*

Koneksi *database* adalah bagaimana kamu dapat menghubungkan aplikasi dengan *database* menggunakan *library* tertentu. Kamu tidak dapat mengelola data di *database* tanpa koneksi ini. Koneksi *database* ibarat “pipa” yang menghubungkan aplikasi yang kamu gunakan dengan *database* (gudang penyimpanan data). Tanpa pipa ini, aplikasi tidak bisa menyimpan, membaca, atau mengubah data. Ibaratnya, seperti kamu mencoba mengisi bak air tanpa selang. Ada berbagai cara membangun “pipa” ini, tergantung bahasa pemrograman dan *database* yang dipakai. Pipa ini harus dipasang dengan benar. Jika tidak, dapat menimbulkan kebocoran (*error*) atau malah memperlambat. Berikut contoh penggunaan koneksi dengan benar.

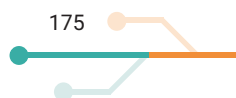
- 1) Pastikan koneksi tertutup setelah dipakai, seperti kamu mematikan keran air yang sudah tidak digunakan (jangan dibiarkan terbuka).
- 2) Ganti koneksi manual dengan *connection pool* (seperti menyediakan beberapa pipa cadangan agar tidak *delay*). Tujuannya agar aplikasi tidak lambat saat diakses banyak pengguna.
- 3) Hindari *SQL injection* (jangan sampai ada orang iseng yang dapat menyisipkan perintah jahat melalui input).

Adapun contoh dari teknologi koneksi *database*, di antaranya:

- 1) *JDBC (Java Database Connectivity)*

Teknologi ini digunakan untuk aplikasi yang dibangun dengan bahasa pemrograman Java.

Contoh: Jika kamu membuat aplikasi desktop dengan Java, *JDBC* akan membantu aplikasi tersebut terhubung ke *database* *MySQL* atau *PostgreSQL*.



2) ODBC (*Open Database Connectivity*)

ODBC dapat digunakan oleh berbagai bahasa pemrograman, seperti C++, Python, atau PHP.

Contoh: Aplikasi berbasis Windows yang terhubung ke *database* Microsoft Access menggunakan ODBC.

3) Mysql-Connector-Python

Library yang memungkinkan kita berinteraksi dengan *database* MySQL secara langsung menggunakan *query* SQL atau dikenal dengan istilah *direct SQL*.

4) ORM (*Object-Relational Mapping*)

ORM adalah teknik yang memungkinkan kamu mengelola *database* menggunakan objek Python, tanpa perlu menulis banyak *query* SQL secara manual.

Ada beberapa metode berbeda untuk integrasi aplikasi kita ke *database*, masing-masing dengan keunikan tersendiri. Misalnya, JDBC dibuat khusus untuk aplikasi berbasis Java, sementara ODBC dapat digunakan oleh berbagai bahasa pemrograman. Ada juga ORM yang lebih praktis karena memungkinkan kita mengelola data, seperti bekerja dengan objek biasa tanpa perlu menulis perintah SQL yang rumit. Pilihan teknologi ini akan memengaruhi struktur program, kecepatan, dan kemudahan pengembangan.

3. Implementasi CRUD dalam Aplikasi

Pembahasan ini merupakan tahapan membuat aplikasi dengan operasi *create*, *read*, *update*, dan *delete* yang terhubung ke *database*. Implementasi CRUD salah satunya diterapkan pada penggunaan Python dan MySQL. Contoh penerapan integrasi aplikasi dengan *database* menggunakan Python dan MySQL, yaitu bagaimana aplikasi Python terhubung ke *database* MySQL menggunakan *library* *mysql-connector-python*.

Instal *library*.

Untuk memastikan *library* `mysql-connector-python` sudah terinstal, kamu dapat menggunakan perintah berikut.

```
pip install mysql-connector-python
```

Buat koneksi ke *database*.

```
import mysql.connector

# Membuat koneksi ke database
db = mysql.connector.connect(
    host="localhost", # Alamat server database
    user="root", # Username database
    password="", # Password database
    database="sekolah" # Nama database
)

# Membuat cursor untuk menjalankan query
cursor = db.cursor()
```

Ambil data dari *database*.

```
# Query untuk mengambil semua data dari tabel siswa
cursor.execute("SELECT * FROM siswa")

# Mengambil hasil query
hasil = cursor.fetchall()

# Menampilkan data
for row in hasil:
    print(row)
```



Simpan data ke database.

```
# Query untuk menambahkan data ke tabel siswa
query = "INSERT INTO siswa (nama, kelas) VALUES (%s, %s)"
nilai = ("Andi", "XI-1")

# Menjalankan query
cursor.execute(query, nilai)

# Menyimpan perubahan ke database
db.commit()

print("Data berhasil ditambahkan!")
```

Tutup koneksi.

```
cursor.close()
db.close()
```

Setelah memahami dasar koneksi ke MySQL, berikut adalah contoh CRUD yang menggunakan SQLite. Mengapa SQLite? Karena SQLite sangat praktis untuk latihan mandiri dan tidak memerlukan server *database* eksternal, sehingga cocok untuk latihan di komputer lokal di laboratorium sekolahmu. Berikut contoh program CRUD dengan SQLite.

Pengembang aplikasi sedang membuat *prototype* stok opname Toko Bintang. Data barang yang disimpan adalah ID produk, nama produk, harga, dan stok. Aplikasi ini memiliki fitur:

- a. menambahkan produk baru;
- b. melihat daftar produk;
- c. mengubah produk; dan
- d. menghapus produk.

Koneksi ke Database

Kode Program 4.16

```
01 import sqlite3
02
03 # --- Inisialisasi dan koneksi database ---
04 conn = sqlite3.connect('toko_bintang.db')
05 cursor = conn.cursor()
06
```

Penjelasan Perintah

`sqlite3.connect()` = membuat/ membuka file *database* toko_bintang.db.
`cursor` = digunakan untuk menjalankan perintah SQL.

Membuat Tabel

Kode Program 4.17

```
07 # --- Membuat tabel jika belum ada ---
08 cursor.execute('''
09     CREATE TABLE IF NOT EXISTS produk (
10         id INTEGER PRIMARY KEY AUTOINCREMENT,
11         nama_produk TEXT NOT NULL,
12         harga INTEGER NOT NULL,
13         stok INTEGER NOT NULL
14     )
15 ''')
16 conn.commit()
17
```

Penjelasan Perintah

Membuat tabel dengan nama produk dengan kolom:

- `id` : sebagai *primary key* untuk nomor unik, *auto increment*
- `nama_produk` : nama barang
- `harga` : harga satuan setiap produk.
- `stok` : jumlah stok produk

`CREATE TABLE IF NOT EXISTS`: perintah agar tidak membuat tabel baru lagi jika tabel yang dimaksud sudah ada.

`conn.commit()`: menyimpan perubahan ke *database*

Fungsi `CREATE` (Input data produk)

Kode Program 4.18

```
18 # --- Fungsi CRUD ---
19 def input_produk():
20     nama = input("Nama Produk: ")
21     harga = int(input("Harga Produk: "))
22     stok = int(input("Stok Produk: "))
23     cursor.execute("INSERT INTO produk (nama_produk, harga, stok)
VALUES (?, ?, ?)", (nama, harga, stok))
24     conn.commit()
25     print("Produk berhasil ditambahkan.\n")
26
```

Penjelasan Perintah

Mengambil data yang ditambahkan pengguna untuk disisipkan pada tabel produk menggunakan perintah `INSERT INTO`. Penggunaan “?” bertujuan untuk menghindari SQL *injection* dan disimpan dengan perintah `conn.commit()`

Fungsi `READ` (Menampilkan data produk)

Kode Program 4.19

```
27 def read_produk():
28     cursor.execute("SELECT * FROM produk")
29     data = cursor.fetchall()
30     print("\n--- Daftar Produk ---")
31     for row in data:
32         print(f"ID: {row[0]} | Nama: {row[1]} | Harga: {row[2]} | Stok:
{row[3]}")
33     print()
34
```

Penjelasan Perintah

Untuk memilih data gunakan perintah *select*, untuk mengambil semua data dari tabel produk dapat menggunakan perintah `SELECT * FROM produk`.

`fetchall()`: perintah untuk mengambil seluruh isi *query*. Perintah untuk menampilkan data produk satu per satu menggunakan pengulangan/*loop for*.

Fungsi *UPDATE* (Memperbarui data produk)

Kode Program 4.20

```
35 def update_produk():
36     id_produk = int(input("ID Produk yang ingin diubah: "))
37     nama = input("Nama Baru: ")
38     harga = int(input("Harga Baru: "))
39     stok = int(input("Stok Baru: "))
40     cursor.execute("UPDATE produk SET nama_produk=?, harga=?,
41     stok=? WHERE id=?", (nama, harga, stok, id_produk))
42     conn.commit()
43     print("Produk berhasil diperbarui.\n")
43
```

Penjelasan Perintah

Untuk memperbarui data produk dibutuhkan **id** sebagai *primary key* dan jika data ditemukan, maka dapat dilakukan *update*. Perintahnya `UPDATE ... SET ... WHERE id=?`

Fungsi *DELETE* (Menghapus data produk)

Kode Program 4.21

```
44 def delete_produk():
45     id_produk = int(input("ID Produk yang ingin dihapus: "))
46     cursor.execute("DELETE FROM produk WHERE id=?", (id_produk,))
47     conn.commit()
48     print("Produk berhasil dihapus.\n")
49
```

Penjelasan Perintah

Menghapus data produk yang tersimpan di tabel berdasarkan **id** sebagai *primary key* dengan perintah **DELETE FROM...WHERE...**

Setelah selesai dengan fungsi untuk CRUD, maka untuk antarmuka dengan pengguna dibuatkan menu pilihan berikut.

Menu Aplikasi

Kode Program 4.22

```
50 # --- Menu Aplikasi ---
51 def menu():
52     while True:
53         print("=== Menu Toko Online ===")
54         print("1. Input Produk")
55         print("2. Lihat Produk")
56         print("3. Update Produk")
57         print("4. Hapus Produk")
58         print("5. Keluar")
59         pilihan = input("Pilih menu (1-5): ")
60
61         if pilihan == '1':
62             input_produk()
63         elif pilihan == '2':
64             read_produk()
65         elif pilihan == '3':
66             update_produk()
67         elif pilihan == '4':
68             delete_produk()
69         elif pilihan == '5':
70             print("Keluar dari program.")
71             break
72         else:
73             print("Pilihan tidak valid. Coba lagi.\n")
74
```

Penjelasan Program

Menu utama ini akan berjalan terus sampai pengguna memilih keluar. Pengguna dapat memilih menu dengan input angka pilihan untuk memanggil fungsi CRUD.

Menutup Program dan Koneksi

Kode Program 4.23

```
75 # --- Menjalankan Program ---
76 menu()
77
78 # --- Menutup koneksi database ---
79 conn.close()
80
```

Perintah terakhir yang ditambahkan untuk menutup koneksi *database* gunakan perintah `conn.close()` untuk keamanan dan menghindari kebocoran *resource*.

```
... === Menu Toko Online ===
  1. Input Produk
  2. Lihat Produk
  3. Update Produk
  4. Hapus Produk
  5. Keluar
  Pilih menu (1-5): 
```

Gambar 4.8 Tangkap Layar Program



Aktivitas 4.10

Membuat Aplikasi Pencatatan Transportasi Harian dengan *Encoding* dan *Database SQLite* (Aktivitas Individu)

- a. Perhatikan ilustrasi berikut.
Untuk mendukung gaya hidup sehat dan ramah lingkungan, sebuah komunitas membuat aplikasi yang akan digunakan untuk mencatat moda transportasi yang digunakan oleh warga kota setiap hari. Hal tersebut bertujuan untuk melihat tren penggunaan transportasi dan mendorong penggunaan moda yang lebih “hijau”. Data jenis transportasi perlu dilakukan *encoding* sebelum disimpan dalam *database* untuk dianalisis.
- b. Pembahasan
 - 1) Struktur data yang digunakan tabel catatan transportasi (ID, nama, tanggal, dan kode_transportasi).
 - 2) *Ordinal encoding* yang digunakan sebelum disimpan ke *database*.
 - 3) Fitur aplikasi: menambahkan data, menampilkan data (*decoding*), mengubah data, dan menghapus data.
 - 4) Daftar transportasi dan *encoding*
 - 5) jalan kaki: 0, sepeda: 1, angkot: 2, bus: 3, motor pribadi: 4, ojek *online*: 5, mobil pribadi: 6
 - 6) *Encoding* dilakukan sebelum data disimpan dan kode dikonversi kembali ke dalam *text* saat membaca data (*read*).
 - 7) *Study case* ini dapat digunakan sebagai praktik yang mencerminkan penerapan *encoding* kategorikal sebelum *machine learning*.
 - 8) Dapat dikembangkan lebih lanjut menjadi aplikasi versi GUI.
- c. *Listing* program lengkap

Kode Program 4.24

```
01 import sqlite3
02 from datetime import date
03
04 # --- Koneksi ke Database ---
05 conn = sqlite3.connect("transportasi.db")
06 cur = conn.cursor()
```

```

07
08 # --- Membuat Tabel ---
09 cur.execute("""
10 CREATE TABLE IF NOT EXISTS catatan_transportasi (
11     id INTEGER PRIMARY KEY AUTOINCREMENT,
12     nama TEXT,
13     tanggal TEXT,
14     kode_transportasi INTEGER
15 )
16 """)
17 conn.commit()
18
19 # --- Dictionary Encoding ---
20 transportasi_dict = {
21     "Jalan Kaki": 0,
22     "Sepeda": 1,
23     "Angkot": 2,
24     "Bus": 3,
25     "Motor Pribadi": 4,
26     "Ojek Online": 5,
27     "Mobil Pribadi": 6
28 }
29 kode_to_transportasi = {v: k for k, v in transportasi_dict.items()}
30
31 # --- Fungsi CRUD ---
32 def tambah_data():
33     nama = input("Masukkan nama: ")
34     tanggal = input("Masukkan tanggal (YYYY-MM-DD), kosongkan untuk hari ini: ")
35     if not tanggal:
36         tanggal = str(date.today())
37
38     print("Pilih transportasi:")
39     for i, key in enumerate(transportasi_dict.keys()):
40         print(f"{i}. {key}")
41     pilihan = int(input("Masukkan nomor pilihan: "))
42     transportasi = list(transportasi_dict.keys())[pilihan]

```

```

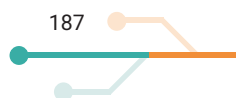
43     kode = transportasi_dict[transportasi]
44     .
45     cur.execute("INSERT INTO catatan_transportasi (nama, tanggal,
kode_transportasi) VALUES (?, ?, ?)",
46     .         (nama, tanggal, kode))
47     conn.commit()
48     print("Data berhasil ditambahkan.\n")
49     .
50     def tampilkan_data():
51         cur.execute("SELECT * FROM catatan_transportasi")
52         rows = cur.fetchall()
53         if not rows:
54             print("Belum ada data.\n")
55         else:
56             for row in rows:
57                 id, nama, tanggal, kode = row
58                 print(f"ID: {id}, Nama: {nama}, Tanggal: {tanggal},
Transportasi: {kode_to_transportasi.get(kode)}")
59                 print()
60         .
61     def ubah_transportasi():
62         tampilkan_data()
63         id = int(input("Masukkan ID data yang ingin diubah: "))
64         print("Pilih transportasi baru:")
65         for i, key in enumerate(transportasi_dict.keys()):
66             print(f"{i}. {key}")
67         pilihan = int(input("Masukkan nomor pilihan: "))
68         transportasi_baru = list(transportasi_dict.keys())[pilihan]
69         kode_baru = transportasi_dict[transportasi_baru]
70         .
71         cur.execute("UPDATE catatan_transportasi SET kode_transportasi = ?
WHERE id = ?", (kode_baru, id))
72         conn.commit()
73         print("Data berhasil diubah.\n")
74         .
75     def hapus_data():

```

```

76     tampilkan_data()
77     id = int(input("Masukkan ID data yang ingin dihapus: "))
78     cur.execute("DELETE FROM catatan_transportasi WHERE id = ?", (id,))
79     conn.commit()
80     print("Data berhasil dihapus.\n")
81
82 # --- Menu Utama ---
83 while True:
84     print("== Menu Aplikasi Transportasi Harian ==")
85     print("1. Tambah Data")
86     print("2. Tampilkan Data")
87     print("3. Ubah Transportasi")
88     print("4. Hapus Data")
89     print("5. Keluar")
90
91     pilihan = input("Pilih menu (1-5): ")
92
93     if pilihan == "1":
94         tambah_data()
95     elif pilihan == "2":
96         tampilkan_data()
97     elif pilihan == "3":
98         ubah_transportasi()
99     elif pilihan == "4":
100        hapus_data()
101     elif pilihan == "5":
102        print("Keluar dari program.")
103        break
104     else:
105        print("Pilihan tidak valid.\n")
106
107 # --- Tutup Koneksi ---
108 conn.close()
109

```



d. Hasil Tampilan

```
=== Menu Aplikasi Transportasi Harian ===  
1. Tambah Data  
2. Tampilkan Data  
3. Ubah Transportasi  
4. Hapus Data  
5. Keluar  
Pilih menu (1-5): 
```



Uji Kompetensi

A. Soal Pilihan Ganda

Pilihlah satu jawaban yang paling tepat!

1. Seorang pengembang ingin mengintegrasikan *database* MySQL ke dalam aplikasi Python. Modul manakah yang paling sesuai?
 - a. Pandas
 - b. Sqlite3
 - c. Pyodbc
 - d. Mysql-connector-python
 - e. Seaborn
2. Pada sebuah situasi ketika jumlah kategori dalam fitur sangat besar, misalnya nama produk. Teknik *encoding* mana yang dapat mengurangi jumlah fitur secara signifikan?
 - a. *Label Encoding*
 - b. *One Hot Encoding*
 - c. *Ordinal Encoding*
 - d. *Target Encoding*
 - e. *Binary Encoding*
3. Mengapa kita perlu melakukan *encoding* pada data kategorikal sebelum digunakan pada pemodelan *Machine Learning*?
 - a. Supaya lebih cepat diunggah ke internet.
 - b. Karena komputer hanya memahami angka.
 - c. Untuk mengurangi ukuran *file*.
 - d. Agar data dapat dicetak lebih mudah.
 - e. Supaya data dapat digunakan untuk presentasi.

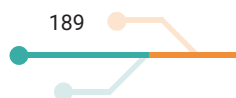
4. Jika suatu kolom data memiliki nilai 'Pagi', 'Siang', dan 'Malam', lalu kita ingin mengubahnya menjadi angka tanpa memperhatikan urutan, teknik *encoding* mana yang paling tepat?
 - a. *Ordinal Encoding*
 - b. *Label Encoding*
 - c. *Frequency Encoding*
 - d. *One Hot Encoding*
 - e. *Binary Encoding*

5. Berikut ini merupakan contoh hasil dari *One Hot Encoding* untuk data ['Merah', 'Hijau', 'Biru'] adalah
 - a. [1, 2, 3]
 - b. ['Merah' = 1, 'Hijau' = 2, 'Biru' = 3]
 - c. [Merah, Merah, Merah]
 - d. [1,0,0], [0,1,0], [0,0,1]
 - e. [Red, Green, Blue]

6. Saat menyiapkan *dataset* untuk analisis, apa langkah pertama yang sebaiknya dilakukan?
 - a. menghapus semua kolom
 - b. mengubah data ke dalam bentuk angka
 - c. memahami isi dan struktur data (*data exploration*)
 - d. menyimpan data ke dalam *database*
 - e. menampilkan data dalam grafik

7. Seorang siswa menemukan banyak nilai kosong (*null*) pada kolom di *dataset*. Apa dampak dari hal ini jika tidak ditangani?
 - a. Data menjadi lebih lengkap.
 - b. Komputer akan memperkirakan sendiri nilainya.
 - c. Proses analisis atau pelatihan model dapat mengalami kegagalan atau tidak akurat.
 - d. Data menjadi lebih cepat diproses.
 - e. Tidak ada pengaruh apapun.

8. Salah satu tujuan pra-pemrosesan data pada *dataset* adalah
 - a. menambahkan gambar ke dalam data
 - b. menyembunyikan nilai data agar tidak dapat dibaca
 - c. menyusun data agar lebih mudah dipahami dan diolah oleh komputer
 - d. menghapus semua data yang bernilai 0
 - e. menyimpan data langsung ke aplikasi



9. Teknik *encoding* yang menggunakan rata-rata nilai target untuk mengganti nilai kategorikal disebut . . .
 - a. *binary encoding*
 - b. *label encoding*
 - c. *target encoding*
 - d. *ordinal encoding*
 - e. *one hot encoding*
10. Mengapa *one hot encoding* dianggap lebih aman digunakan daripada *label encoding* untuk data kategorikal tanpa urutan?
 - a. Hasilnya berbentuk angka acak.
 - b. Lebih mudah diimplementasikan.
 - c. Membuat model salah menganggap ada hubungan urutan antarkategori.
 - d. Membuat data lebih kecil ukurannya.
 - e. Dapat digunakan tanpa komputer.

B. Uji Praktik Koding

Amati studi kasus berikut dengan saksama, lalu kerjakan tugasnya!

1. Sebuah aplikasi wisata mencatat jenis tujuan wisata yang sering dikunjungi penggunanya, seperti ‘pantai’, ‘gunung’, ‘kota tua’, ‘hutan’, dan ‘taman hiburan’. Jenis wisata tersebut merupakan data kategorikal, sehingga perlu dilakukan *encoding* menggunakan *binary encoding*.
 - a. Buatlah urutan abjad dari kategori wisata!
 - b. Beri representasi biner 3 bit untuk masing-masing kategori!
 - c. Ubahlah data kunjungan menjadi representasi biner!
2. Sebuah survei kuliner mencatat jenis makanan favorit di sebuah restoran dari lima kategori berikut: ‘sate’, ‘soto’, ‘bakso’, ‘rendang’, dan ‘gado-gado’. Agar dapat digunakan oleh sistem klasifikasi preferensi, jenis makanan dikonversi ke dalam *one hot encoding*.
 - a. Ubahlah data menjadi bentuk *one hot encoding*!
 - b. Buatlah tabel *encoding* untuk kolom makanan!
 - c. Buatlah koding program menggunakan Python!
3. Sebuah *startup* pendidikan *online* memiliki tiga aplikasi belajar, seperti ‘App A’, ‘App B’, dan ‘App C’. *Startup* ini ingin mengetahui aplikasi mana yang paling memuaskan pengguna berdasarkan tingkat kepuasan/rating aplikasi (1 = puas, 0 = tidak puas) menggunakan *target encoding*.

4. Berikut data dari kasus tersebut.

No.	Nama Aplikasi	Puas
1	App A	1
2	App A	1
3	App A	0
4	App B	1
5	App B	0
6	App B	0
7	App C	0
8	App C	1

Tugas

- Kelompokkan data berdasarkan aplikasi!
- Hitunglah rata-rata tingkat kepuasan setiap aplikasi!
 - App A = _____
 - App B = _____
 - App C = _____
- Lakukan *target encoding* pada kolom aplikasi!
- Analisis dan berikan rekomendasi dari hasil *encoding* berdasarkan nilai *encoding*! Aplikasi mana yang paling direkomendasikan ke pengguna baru?
- Lakukan praktik koding dengan Python!

No.	Aplikasi	Puas
1	App A	1
2	App A	1
3	App A	0
4	App B	1
5	App B	0
6	App B	0
7	App C	0
8	App C	1

Pengayaan

Melalui kegiatan pengayaan ini, kamu tidak hanya memahami konsep dasar, tetapi juga siap menjelajah lebih dalam bagaimana *encoding* dan *database* menjadi fondasi penting dalam sistem digital modern. Jadilah pembuat solusi data, bukan hanya pengguna. Kamu dapat berlatih membuat aplikasi sederhana untuk pencatatan kegiatan harian siswa yang menyimpan data ke *database*. Gunakan teknik *encoding* untuk data waktu, jenis kegiatan, dan lokasi. Aplikasi harus dapat menampilkan statistik kegiatan terbanyak. Kemudian, demonstrasikan aplikasimu di hadapan guru dan teman-teman di kelas.

Latihan untuk memperkaya keterampilan dari materi yang telah dipelajari dan tantangan baru, kalian dapat mengunjungi laman melalui QRCode



<https://buku.kemdikbud.go.id/s/KKA11-SuplemenBab4>



Refleksi

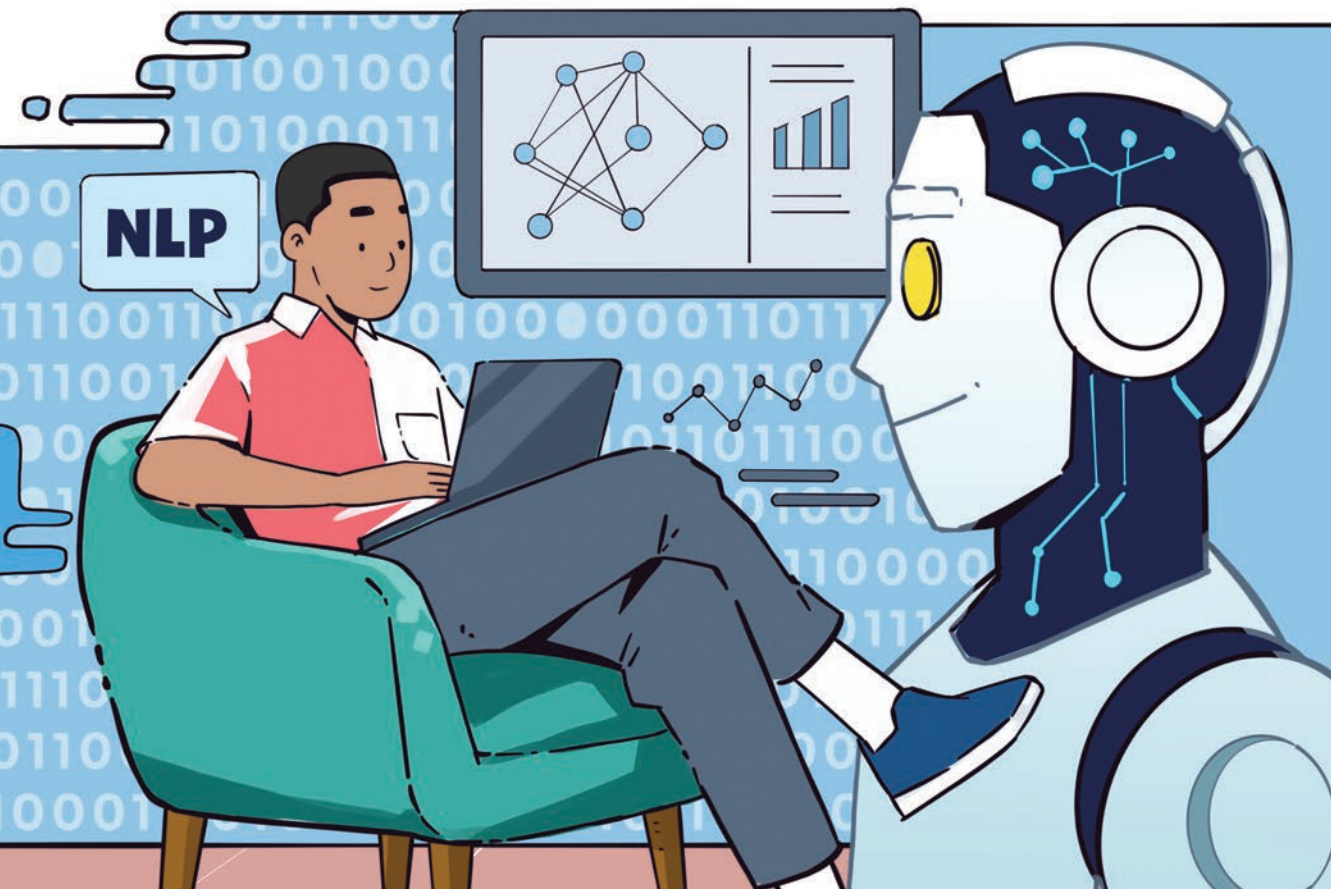
Setelah kamu mempelajari materi Bab 4 tentang “Data *Encoding* dan Integrasi Basis Data dalam Aplikasi”, silakan kamu identifikasi apa saja yang telah kamu peroleh.

1. Apa hal baru yang telah saya pelajari tentang jenis data dan proses *encoding*? Bagian mana dari materi Bab 4 ini yang paling saya kuasai? Mengapa?
2. Bagian mana yang masih membingungkan bagi saya? Apa yang bisa saya lakukan untuk memahaminya?
3. Apa manfaat memahami *encoding* data bagi kehidupan atau pekerjaan saya di masa depan?
4. Jika saya diminta menjelaskan kembali tentang label *encoding* atau *one hot encoding*, seberapa yakin saya bisa melakukannya?

KEMENTERIAN PENDIDIKAN DASAR DAN MENENGAH
REPUBLIK INDONESIA, 2025
Koding dan Kecerdasan Artifisial
untuk SMA/MA Kelas XI
Penulis: Ibnu Indarwati, Marwondo, Muhammad Faisal
ISBN 978-634-00-2055-7 (jil.2 PDF)

Bab 5

Dasar-Dasar *Machine Learning* dan *Natural Language Processing*



Pernahkah kamu memperhatikan bagaimana proses Kecerdasan Artifisial (KA) mengenal sebuah objek?



Tujuan Pembelajaran

Setelah mempelajari materi pada Bab 5, kamu diharapkan mampu memahami konsep dasar *Machine Learning* beserta jenis-jenis dan algoritmanya; serta memahami konsep dasar *Natural Language Processing* dan komponen yang terdapat di dalamnya.

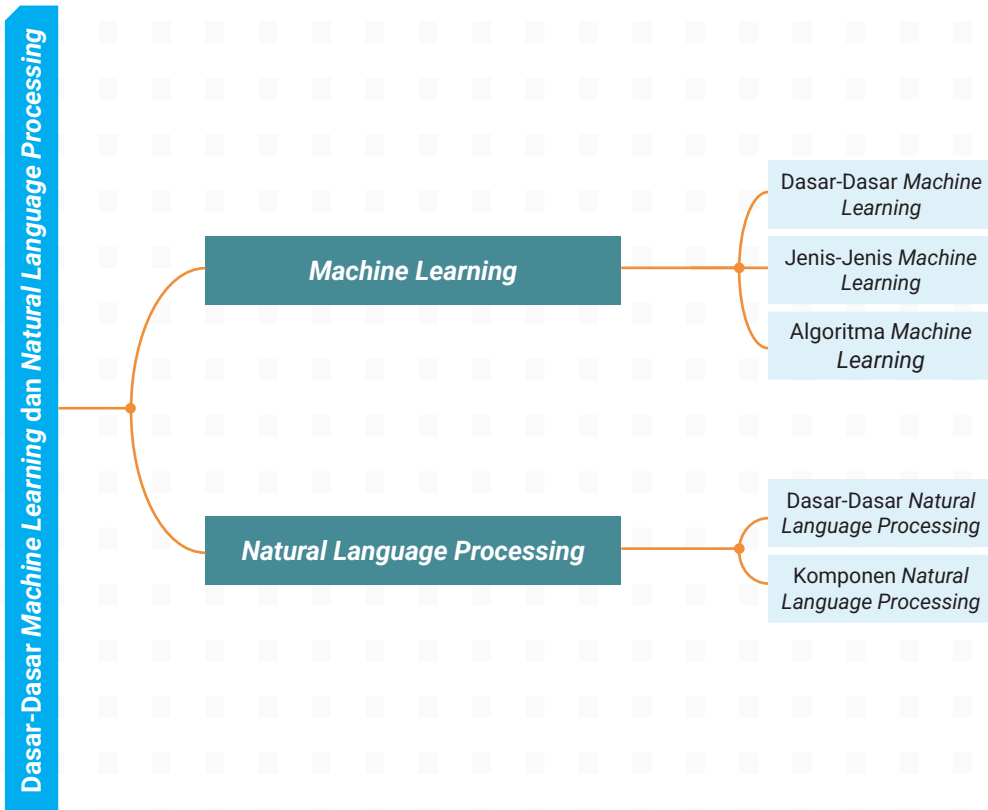


Kata Kunci

machine learning, natural language processing, supervised learning, unsupervised learning, reinforcement learning



Peta Materi





Siap-Siap Belajar

Kamu pasti telah merasakan bagaimana teknologi *Machine Learning* (ML) dan *Natural Language Processing* (NLP) membantumu dalam berbagai aspek kehidupan sehari-hari, mulai dari asisten virtual, seperti *Google Assistant* hingga sistem rekomendasi pada platform *streaming*. Teknologi ini tidak terpisahkan lagi dari kehidupan kita. ML dan NLP merupakan cabang dari Ilmu Komputer yang berkembang dengan pesat. Oleh sebab itu, sangat penting bagimu untuk memahami kedua teknologi tersebut, baik secara fakta, konsep, maupun prosedurnya. Dengan mempelajari dasar-dasar ML dan NLP, kamu akan memahami bagaimana teknologi ini mengerjakan tugas-tugas spesifiknya. Sehingga, ML dan NLP dapat membantumu dalam berbagai bidang, seperti analisis data, pengenalan gambar, pengenalan suara, dan sebagainya.



Cek Pemahaman Awal

Jawablah pernyataan berikut sesuai dengan pengetahuan dan pemahamanmu. Beri tanda centang (✓) pada kolom yang sesuai.

No.	Pernyataan	Benar	Salah
1	Algoritma adalah serangkaian instruksi yang sistematis dan logis untuk menyelesaikan masalah.		
2	Algoritma sangat penting dalam pengembangan perangkat lunak dan sistem komputer.		
3	<i>Machine Learning</i> (ML) adalah cara membuat komputer belajar.		
4	<i>Machine Learning</i> (ML) belajar melalui data.		
5	Saya sering menemukan produk kesukaan saya pada saat <i>browsing</i> .		
6	<i>Natural Language Processing</i> (NLP) merupakan teknologi yang memungkinkan komputer memahami bahasa manusia.		
7	Melalui teknologi NLP, komputer mampu mengetahui kata benda, kata sifat, dan keterangan dalam sebuah kalimat.		

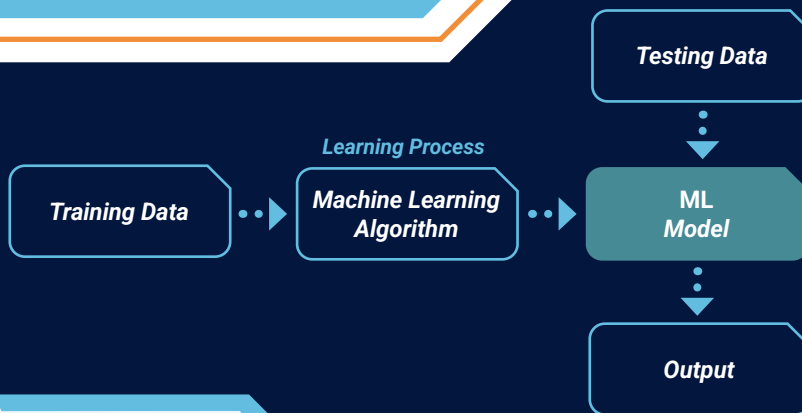
No.	Pernyataan	Benar	Salah
8	Saya sering menggunakan <i>machine translator</i> seperti Google Translator.		
9	Saya menggunakan fitur pengenalan wajah untuk membuka <i>smartphone</i> .		
10	Teknologi <i>Natural Language Processing</i> (NLP) dapat membantu kita dalam komunikasi dan otomatisasi.		

Kita semua mesti menyadari bahwa teknologi dunia bukan lagi sekadar fantasi, melainkan kenyataan yang sedang kita bangun saat ini. Kecerdasan Artifisial (KA) telah mengubah cara kita berinteraksi dengan teknologi, dari asisten virtual di ponsel pintar hingga mobil otonom yang melaju di jalan raya. Dalam materi ini, kita akan menjelajahi bagaimana *Machine learning* (ML) dan *Natural Processing Language* (NLP) dalam model Kecerdasan Artifisial (KA) dikembangkan dan diterapkan dalam berbagai bidang.

A. Dasar-Dasar *Machine learning* (ML)

Machine learning (ML) adalah cabang dari Kecerdasan Artifisial (KA) yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya seiring waktu tanpa perlu diprogram secara eksplisit. Konsep dasar *machine learning* berfokus pada pengembangan algoritma yang dapat menganalisis data, mengenali pola, dan membuat prediksi atau keputusan berdasarkan data tersebut.

Dalam *machine learning*, terdapat empat jenis pembelajaran utama, yaitu pembelajaran terbimbing (*supervised learning*), pembelajaran tidak terbimbing (*unsupervised learning*), pembelajaran semi-terbimbing (*semi-supervised learning*), dan pembelajaran penguatan (*reinforcement learning*). *Supervised learning* melibatkan penggunaan data berlabel untuk melatih model; *Unsupervised learning* bekerja dengan data tanpa label untuk menemukan pola tersembunyi; *Semi-supervised learning* merupakan kombinasi antara pembelajaran terbimbing dan tidak terbimbing untuk meningkatkan performanya; Sedangkan *reinforcement learning* merupakan jenis pembelajaran mesin yang terlibat interaksi luar atau lingkungan sebagai umpan balik. Contoh dari penerapan *machine learning* adalah robot, di mana robot dilatih untuk mengerjakan tugas-tugas tertentu.



Gambar 5.1 Skema Kerja *Machine Learning*

Machine learning membutuhkan data untuk belajar. Data awal yang digunakan untuk berlatih disebut dengan *data training*. Data menjadi objek penting untuk melatih mesin supaya menjadi cerdas. Data akan diproses menggunakan algoritma tertentu sesuai dengan fungsi atau tujuan *machine learning* dibangun. Setelah *data training* diproses, maka akan menghasilkan sebuah model yang disebut dengan *machine learning model*.

Machine learning model yang dihasilkan perlu diuji. Hal ini dimaksudkan untuk memeriksa apakah *machine learning* yang dibangun sudah sesuai dengan tujuan dan fungsinya. Untuk melakukan pengujian ini, digunakan data yang disebut dengan *data testing*. Data inilah yang digunakan untuk menganalisis dan mengevaluasi seberapa baik kinerja model yang telah dibangun.

Data testing dimasukkan ke dalam *machine learning model* dan menghasilkan *output*. *Output* yang dihasilkan akan dievaluasi, apakah sesuai dengan *data testing* atau tidak. Jika tidak sesuai, maka *machine learning model* perlu dilatih kembali.

Ingatlah!

- *Machine learning* selalu belajar dari data untuk menghasilkan pengetahuan.
- *Data testing* merupakan sebuah *dataset* yang berbeda dari *data training*.



Aktivitas 5.1

Mari Eksplorasi *Machine Learning* (Aktivitas Kelompok)

Diskusikanlah dengan teman kelompokmu tentang fakta-fakta teknologi *Machine Learning* dan perkembangannya. Kemudian, presentasikan hasil kerja kalian di hadapan guru dan kelompok lain. Berikut langkah-langkah untuk mengerjakan Aktivitas 5.1.

- Bentuklah kelompok yang terdiri atas 3–4 orang!
- Gunakan tabel berikut untuk melakukan eksplorasi!

Beri tanda centang (✓) pada kolom yang sesuai

No.	Nama Aplikasi/Sistem	Prediksi	Rekomendasi
1	Platform <i>marketplace</i> Shopee		
2	Platform <i>marketplace</i> Tokopedia		
3	Platform <i>marketplace</i> Bukalapak		
4	Google Map Direction		
5	Fitur cuaca pada <i>smartphone</i>		

- Analisis dan jabarkan hasil eksplorasi pada tabel (Mengapa aplikasi tersebut berjenis prediksi atau rekomendasi?) bersama kelompokmu!
- Buatlah presentasi yang menarik, kemudian presentasikan di depan guru dan kelompok lain!

1. Jenis-Jenis *Machine Learning*

Untuk memahami lebih dalam mengenai konsep dasar *machine learning*, kamu perlu mengenal jenis-jenis *machine learning* secara umum. Adapun jenis-jenis *machine learning* tersebut dapat kamu pahami pada penjelasan berikut.

a. *Supervised Learning*

Supervised learning merupakan jenis pembelajaran terbimbing dalam *machine learning*. Hal ini dapat kita ibaratkan seperti cara manusia dalam belajar di bawah bimbingan seorang guru. Model ini dilatih menggunakan data yang telah diberi label serta memiliki target yang jelas. Tujuan *supervised learning* adalah untuk memprediksi atau mengklasifikasi data baru berdasarkan pola yang dipelajari dari *data training* menggunakan algoritma klasifikasi dan regresi.

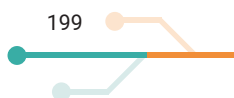
b. *Unsupervised Learning*

Unsupervised learning merupakan jenis pembelajaran tidak terbimbing dalam *machine learning* atau belajar tanpa guru. Data yang digunakan untuk melatih model ini tidak memiliki label dan target yang jelas. Tujuan dari *unsupervised learning* adalah untuk menemukan pola, struktur, atau hubungan dalam data menggunakan algoritma klusterisasi dan asosiasi.

c. *Semi-Supervised Learning*

Semi-supervised learning merupakan kombinasi dari tipe *supervised learning* dan *unsupervised learning*, yaitu jenis pembelajaran yang bertujuan untuk meningkatkan kinerja prediksi. Tipe ini memberikan solusi untuk menyelesaikan masalah. Di mana, data yang telah dilabeli sulit didapatkan, namun data yang tidak dilabeli tersedia dalam jumlah besar.

Salah satu keunggulan dari *machine learning* tipe *semi-supervised learning* adalah kemampuannya untuk memanfaatkan informasi yang terdapat dalam data yang tidak dilabeli. Meskipun data yang tidak dilabeli tidak bisa memberikan informasi tentang *output* yang diinginkan, tetapi data ini tetap bisa memberikan *insight* tentang struktur data dan hubungan antarfitur.



Dengan memahami konsep *semi-supervised learning* dan potensinya dalam memanfaatkan data yang dilabeli dan tidak dilabeli, kamu dapat mengembangkan model *machine learning* yang lebih efisien dan akurat. Dengan terus melakukan penelitian dan inovasi dalam bidang *semi-supervised learning*, kamu dapat mengoptimalkan penerapannya dalam berbagai bidang aplikasi dan menghasilkan solusi yang lebih canggih dan adaptif.

d. *Reinforcement Learning*

Algoritma *reinforcement learning* didefinisikan sebagai metode *machine learning* yang berkaitan dengan cara *software* mengambil tindakan di lingkungannya. Algoritma ini merupakan bagian dari metode *deep learning* yang akan memaksimalkan sebagian *reward*/aksi kumulatif.

Contoh penggunaan *reinforcement learning* adalah pada sektor manufaktur. Beberapa perusahaan manufaktur menggunakan robot dengan *reinforcement learning* untuk mengambil barang dari satu tempat ke tempat lain. Robot ini dilatih untuk menghafal objek dan melakukan pekerjaan dengan kecepatan dan presisi yang tinggi. Selain itu, robot-robot ini juga dapat digunakan untuk menyortir berjuta-juta produk di gudang *supermarket* atau *e-commerce*. Tujuan pemanfaatan robot ini adalah untuk menghindari *human error*, sehingga produk tersebut dapat dikirim ke konsumen yang tepat. Selain itu, penggunaan robot juga dapat mengurangi risiko cacat saat proses produksi.

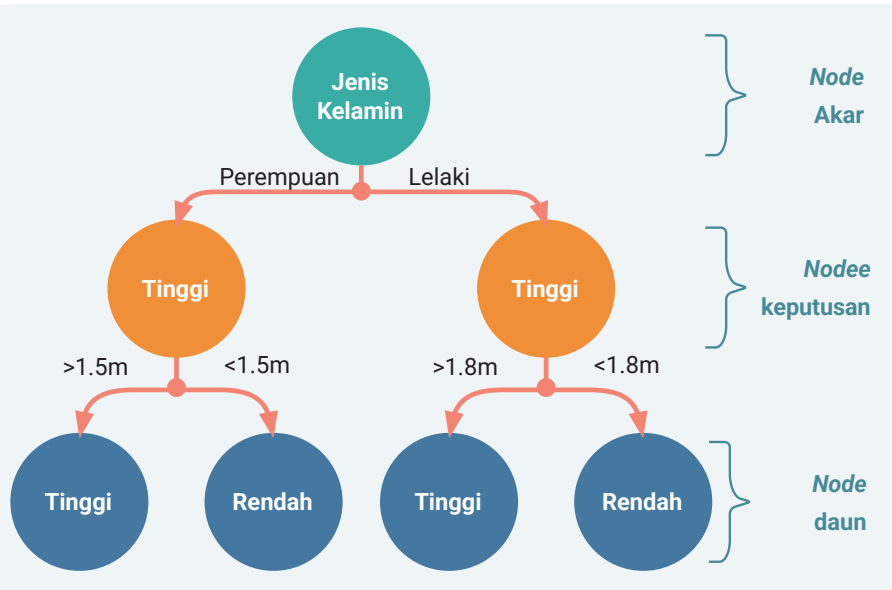
2. Algoritma *Machine Learning*

Dalam implementasi teknologi *machine learning*, penting bagi kita untuk memahami dan mendalami algoritma-algoritimanya. Berikut ini merupakan beberapa algoritma umum dan populer yang digunakan dalam *machine learning*.

a. *Decision Tree*

Decision tree atau bisa disebut juga dengan pohon keputusan merupakan algoritma yang mengacu kepada pola struktur pohon yang terdiri dari titik-titik atau *node* yang merepresentasikan keputusan. Setiap cabang merepresentasikan konsekuensi dari keputusan yang dibuat. Setiap *node* dalam *decision tree* merepresentasikan variabel dalam *dataset* yang memengaruhi keputusan dan konsekuensinya. Jika kamu

menguasai algoritma ini, maka kamu akan dapat membuat sistem cerdas untuk merekomendasikan produk atau layanan berdasarkan perilaku pelanggan (*preferences*). Dengan demikian, kamu akan mampu meningkatkan target pemasaran.



Gambar 5.2 Ilustrasi Algoritma *Decision Tree*

Kamu dapat mempelajari alur kerja dan contoh proyek *Decision Tree* pada *link* berikut.



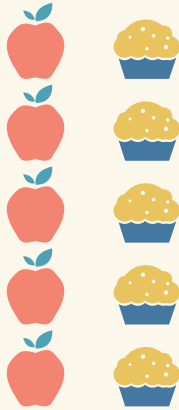
<https://buku.kemdikbud.go.id/s/KKA11-DecesionTree>

a. *K-Nearest Neighbour (KNN)*

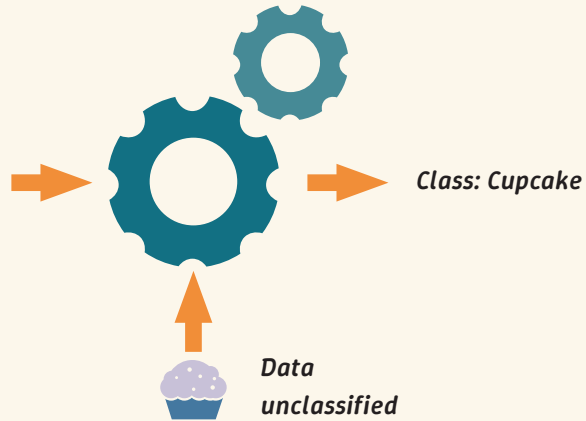
Algoritma ini bekerja untuk menentukan kategori suatu data berdasarkan kedekatannya dengan data lain yang sudah diklasifikasikan sebelumnya. Jika kamu menguasai algoritma ini, maka kamu akan mampu membuat sistem cerdas untuk klasifikasi gambar dan klasifikasi suara berdasarkan fitur-fitur yang melekat pada objek, seperti visual objek, warna, dan/atau teksturnya.

Training Dataset

Apples Cupcakes



Machine Learning Algorithm



Gambar 5.3 Ilustrasi Algoritma KNN

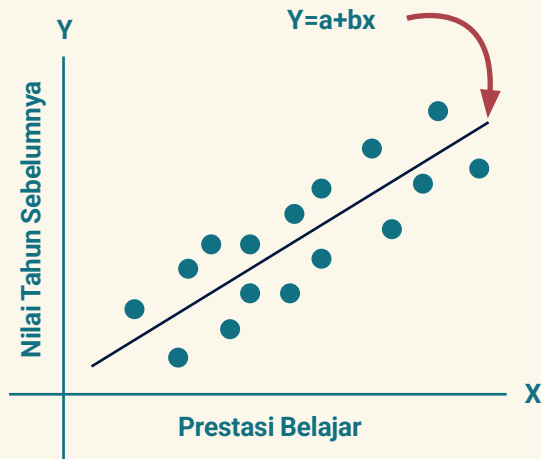
Kamu dapat mempelajari alur kerja dan contoh proyek *K-Nearest Neighbour* (KNN) pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-KNN>

b. Regresi Linier

Merupakan salah satu algoritma yang dapat digunakan untuk membuat sebuah model prediksi. Jika kamu menguasai algoritma ini, kamu akan mampu membangun sistem prediksi, seperti prediksi harga rumah berdasarkan luas tanah, jumlah kamar, jarak ke pusat bisnis, dan sebagainya.



Gambar 5.4 Ilustrasi Algoritma Regresi Linier

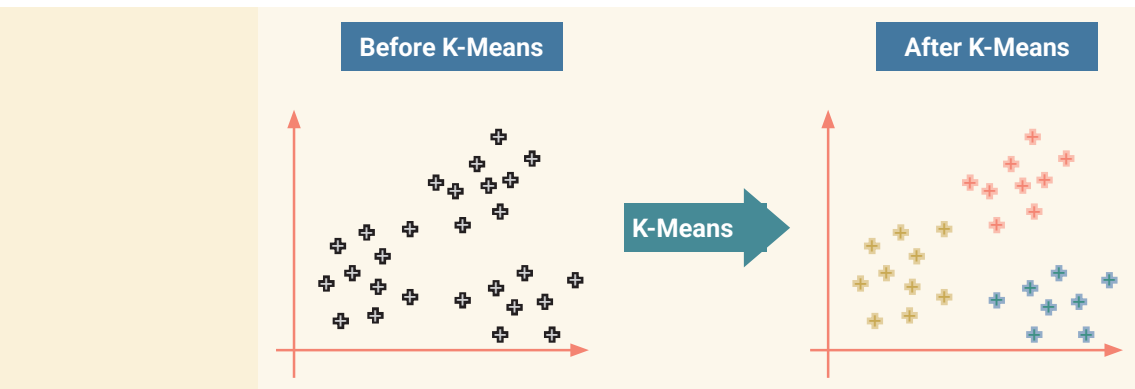
Kamu dapat mempelajari alur kerja dan contoh proyek Regresi Linier pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-RegresiLinier>


c. *K-Means*

Algoritma ini digunakan untuk mengelompokkan data ke beberapa kelompok berdasarkan kemiripan karakteristiknya. Tujuannya adalah membagi data menjadi sejumlah kelompok yang disebut dengan “*cluster*”, sehingga data dalam satu kelompok memiliki karakteristik yang mirip satu sama lain. Jika kamu menguasai algoritma ini, kamu akan mampu membangun sistem segmentasi pelanggan berdasarkan perilakunya, menganalisis pengguna media sosial berdasarkan kesamaan hobi, mengelompokkan data pasien berdasarkan gejala atau berdasarkan diagnosis, dan sebagainya.



Gambar 5.5 Ilustrasi Algoritma *K-Means*

Kamu dapat mempelajari alur kerja dan contoh proyek *K-Means* pada *link* berikut.


<https://buku.kemdikbud.go.id/s/KKA11-KMeans>

d. *Association Rule Model*

Algoritma ini digunakan untuk menemukan pola hubungan antara item-item dalam data. Misalnya, kamu ingin mengetahui pola perilaku pelanggan pada suatu platform tertentu setelah membeli barang A, kemudian pelanggan tersebut akan membeli lagi barang B. Dengan menguasai algoritma ini, kamu dapat membuat sistem cerdas yang mampu merekomendasikan barang sesuai perilaku pelanggan tersebut.



Gambar 5.6 Ilustrasi Algoritma *Association Rule Model*

Kamu dapat mempelajari alur kerja dan contoh proyek *Association Rule Model* pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-AssociationRuleModel>



Aktivitas 5.2

Mari Mendalami *Machine Learning* (Aktivitas Kelompok)

Berdiskusilah dengan teman kelompokmu tentang bagaimana konsep *Machine Learning* dapat membantu kalian dalam menyelesaikan masalah. Misalnya, sistem rekomendasi produk pada platform *marketplace* atau rekomendasi video di TikTok atau Youtube. Pilihlah satu aplikasi/sistem; tentukan algoritma apa yang digunakan; uraikan langkah kerjanya; kemudian presentasikan di hadapan guru dan kelompok lain. Adapun langkah-langkah untuk mengerjakan Aktivitas 5.2 adalah sebagai berikut.

- Bentuklah kelompok yang terdiri atas 3–4 orang!
- Gunakan tabel berikut untuk melakukan eksplorasi!

No.	Nama Aplikasi/Sistem	Prediksi/ Rekomendasi	Algoritma yang Digunakan
1	Platform <i>marketplace</i> Shopee		
2	Platform <i>marketplace</i> Tokopedia		
3	Platform <i>marketplace</i> Bukalapak		
4	Google Map Direction		
5	Fitur cuaca pada <i>smartphone</i>		

- c. Setelah mengetahui algoritma yang digunakan, silakan kamu berdiskusi bersama teman sekelompok untuk melakukan eksplorasi secara mendalam tentang pengertian, kelebihan dan kekurangan, formula inti yang digunakan, *dataset* yang digunakan, serta tahapan kerja algoritma yang dipilih!
- d. Presentasikan hasil kerja kelompokmu di depan kelas!
- e. Tuliskan kesimpulan tentang algoritma yang digunakan pada *machine learning*!

Setelah kamu dan kelompokmu selesai mengerjakan Aktivitas 5.2, kamu akan lebih memperdalam pemahaman tentang algoritma *machine learning* tersebut dengan mengimplementasikannya pada beberapa kasus yang ada di sekitarmu. Perhatikan dengan saksama instruksi pada Aktivitas 5.3 berikut, lalu kerjakan langkah-langkah kerjanya bersama teman sekelompokmu.



Aktivitas 5.3

Mari Implementasikan *Machine Learning* (Aktivitas Kelompok)

Pada Aktivitas 5.3 ini, kalian akan berlatih untuk mengimplementasikan algoritma *machine learning* pada kasus-kasus yang terjadi dalam kehidupan sehari-hari. Kerjakan aktivitas ini sesuai dengan petunjuk berikut.

a. Studi Kasus

Amati kedua studi kasus berikut dengan saksama!

Studi Kasus 1

Prediksi kedisiplinan siswa dengan jadwal tidur malam.

Studi Kasus 2

Klasterisasi kedisiplinan siswa dengan jarak tempuh ke sekolah.

b. Langkah Kerja

1) Mengumpulkan Data

Diskusikan dengan teman kelompok tentang data apa saja yang dibutuhkan untuk membangun proyek dimaksud!

- 2) **Membersihkan Data**
Pada tahap ini pastikan data yang telah dikumpulkan terhindar dari data ganda, data tidak lengkap (data hilang), dan data tidak konsisten! Selanjutnya, tentukan format dan tipe data agar mudah dalam menganalisisnya!
- 3) **Memilih Algoritma atau Model *Machine Learning* yang Tepat**
Diskusikan dengan teman kelompok untuk memilih algoritma atau model *machine learning* yang tepat berdasarkan kasus yang kalian pilih!
- 4) **Melatih Model dengan Data (*Training Model*)**
Lakukan *training model* terhadap data yang telah diperoleh!
- 5) **Membuat Prediksi atau Klasterisasi dan Evaluasi Model**
Buatlah prediksi atau klasterisasi dan evaluasi model terhadap data yang telah melalui proses *training model* tersebut!
- 6) **Presentasi di depan kelas.**
Presentasikan hasil kerja kelompokmu di hadapan guru dan kelompok lain!

B. *Natural Language Processing* (NLP)

Natural Language Processing (NLP) adalah teknologi yang memungkinkan komputer memaknai, memahami, dan memanipulasi bahasa manusia. NLP juga dipandang sebagai salah satu cabang ilmu dalam Kecerdasan Artifisial (KA) yang berfokus pada pengembangan sistem untuk memahami bahasa manusia.

1. *Dasar-Dasar Natural Language Processing* (NLP)

Natural Language Processing (NLP) bertujuan untuk membuat mesin lebih “mengerti” bahasa yang kita gunakan sehari-hari, sehingga komunikasi antara manusia dan mesin menjadi lebih baik. Beberapa tugas utama NLP adalah pemahaman teks, analisis sentimen, terjemahan bahasa, dan pengenalan suara. Melalui teknologi NLP, komputer dapat membaca teks, mendengarkan ucapan, dan menulis teks secara mandiri.

NLP kini digunakan di berbagai aplikasi yang kita jumpai sehari-hari, baik secara langsung maupun tidak langsung. Contoh yang paling umum adalah asisten virtual, penerjemahan bahasa, chatbot layanan pelanggan, dan analisis sentimen.

a. **Asisten Virtual**

Aplikasi yang digunakan sebagai asisten virtual, di antaranya Siri, Alexa, dan Google Assistant. Aplikasi-aplikasi tersebut menggunakan NLP untuk memahami perintah pengguna dan meresponsnya. Misalnya, saat kita mengatakan: “Setel alarm untuk pukul 7 pagi!”, NLP digunakan untuk memahami konteks waktu dan tindakan yang diminta.

b. **Penerjemahan Bahasa**

Aplikasi penerjemahan bahasa, seperti Google Translate menggunakan NLP untuk menerjemahkan teks dari satu bahasa ke bahasa lain. Aplikasi ini melibatkan pemahaman sintaks dan semantik dari bahasa sumber dan target agar maknanya tetap akurat.

c. **Chatbot Layanan Pelanggan**

Banyak perusahaan kini menggunakan chatbot berbasis NLP untuk menangani pertanyaan dan keluhan pelanggan. Chatbot ini mampu merespons berbagai permintaan pengguna, seperti: “Apa status pesanan saya?” atau “Saya ingin mengembalikan produk”.

d. **Analisis Sentimen di Media Sosial**

Perusahaan dapat menggunakan NLP untuk menganalisis sentimen pelanggan terhadap merek atau produk mereka. Dengan analisis sentimen, perusahaan dapat memahami apakah pelanggan mereka merasa puas, kecewa, atau memiliki saran perbaikan.



Aktivitas 5.4

Mari Diskusikan Dampak NLP (Aktivitas Kelompok)

Diskusikan dengan teman kelompokmu tentang fakta-fakta teknologi NLP dan perkembangannya sekaligus dampak positif dan dampak negatifnya secara umum. Kemudian, presentasikan hasil diskusi kalian di depan kelas. Berikut langkah-langkah pengerjaan Aktivitas 5.4.

- a. Bentuklah kelompok yang terdiri atas 3–4 orang!
- b. Gunakan tabel berikut ini untuk melakukan eksplorasi!

No.	Teknologi	Dampak Positif	Dampak Negatif
1	Mesin Penerjemah		
2	Mesin Pencari		
3	Asisten Virtual		

- c. Sebutkan beberapa contoh aplikasi yang pernah kalian gunakan!
- d. Uraikan penjelasannya dalam sebuah presentasi yang menarik, lalu presentasikan di hadapan guru dan kelompok lain!

Setelah kamu dan kelompokmu mengerjakan Aktivitas 5.4, tentunya kini kamu mengetahui dampak dari penggunaan teknologi NLP dalam kehidupan sehari-hari. Selanjutnya, kamu akan melakukan diskusi untuk memahami konsep dan prosedur NLP dalam bidang teknologi mesin penerjemah, mesin pencari, dan asisten virtual.



Aktivitas 5.5

Mari Diskusikan Konsep dan Prosedur NLP (Aktivitas Kelompok)

Berdiskusilah dengan teman kelompokmu tentang konsep dan prosedur NLP dalam bidang teknologi mesin penerjemah, mesin pencari, dan asisten virtual, kemudian presentasikan hasil diskusi kelompokmu di hadapan guru dan kelompok lain. Berikut langkah-langkah pengerjaan Aktivitas 5.5.

- a. Bentuklah kelompok yang terdiri atas 3–4 orang!

- b. Diskusikan dan eksplorasi konsep dan prosedur NLP yang melibatkan komponen-komponen NLP berikut!

No.	Nama Teknologi	Konsep dan Prosedur Kerja Teknologi
1	Mesin Penerjemah	
2	Mesin Pencari	
3	Asisten Virtual	

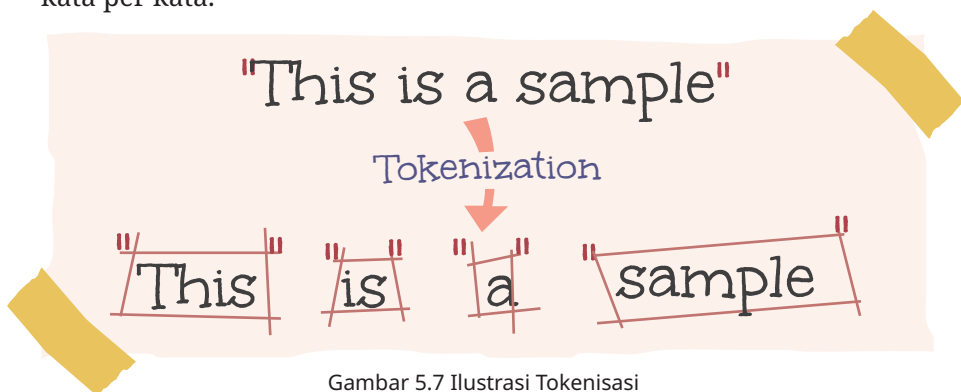
- c. Presentasikan hasil kerja kelompokmu di hadapan guru dan kelompok lain!

2. Komponen *Natural Language Processing* (NLP)

Untuk memahami bagaimana NLP bekerja, penting bagimu mengetahui terlebih dahulu komponen NLP berikut.

a. Tokenisasi

Komponen ini merupakan langkah pertama untuk memecah teks menjadi unit-unit kecil yang disebut dengan “token”, seperti kata atau frasa. Tokenisasi membantu dalam pemrosesan teks yang lebih lanjut. Misalnya, kamu memiliki sebuah kata “*This is a sample.*”, maka mesin akan melakukan pemisahan setiap kata supaya mudah diterjemahkan, sehingga menjadi “*this*”, “*is*”, “*a*”, “*sample*”. Komputer tidak dapat memahami kalimat yang utuh. Oleh karena itu, diperlukan *tokenisasi* kata per kata.



Gambar 5.7 Ilustrasi Tokenisasi

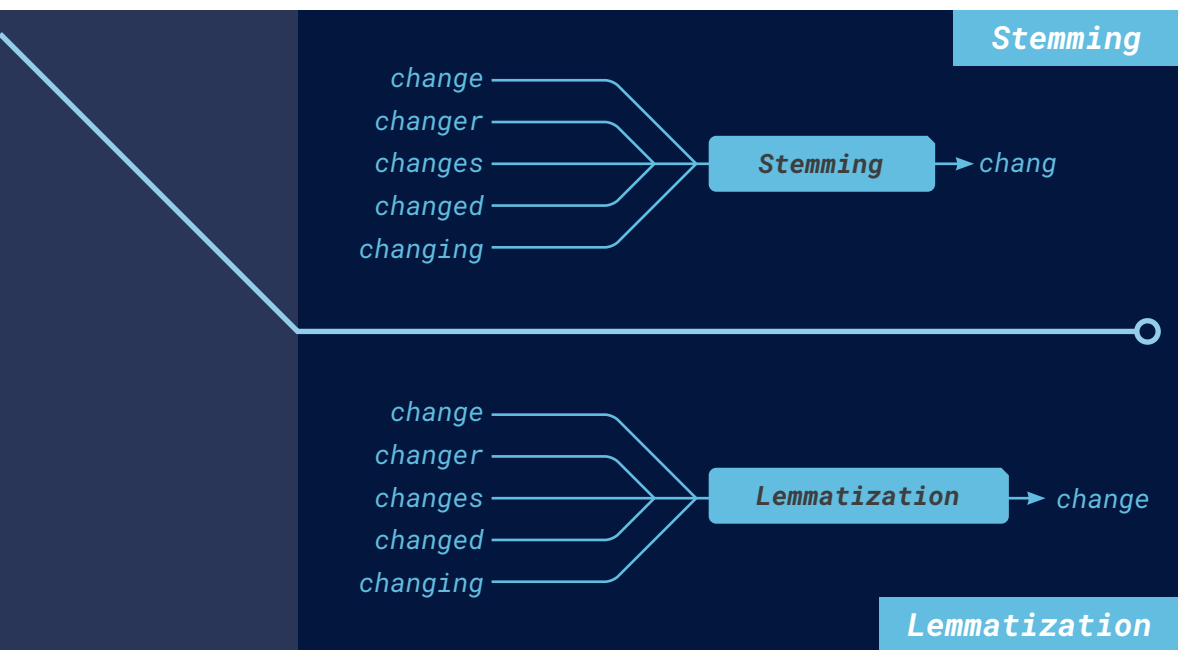
Kamu dapat mendalami dan mempraktikkan konsep Tokenisasi pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-KonsepTokenisasi>

b. *Stemming* dan *Lemmatization*

Stemming dan *lemmatization* adalah teknik untuk mengubah kata-kata ke bentuk dasarnya. *Stemming* menghilangkan imbuhan untuk mendapatkan akar kata, sedangkan *lemmatization* mengubah kata-kata menjadi bentuk dasarnya. Misalnya, “berlari,” “lari,” dan “lari-lari” akan diubah menjadi “lari”.



Gambar 5.8 Ilustrasi *Stemming* dan *Lemmatization*

Kamu dapat mendalami dan mempraktikkan konsep *Stemming* dan *Lemmatization* melalui *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-Stemming&Lemmatization>

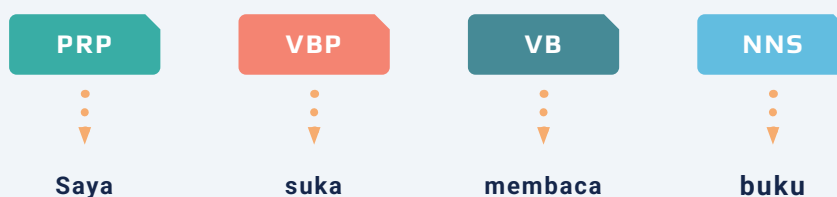
c. *Part of Speech (POS) Tagging*

Teknik membuat *tag* pada sebuah kata berdasarkan posisinya dalam kalimat. Misalnya, sebuah kata dikenali sebagai kata benda, kata sifat, kata kerja, atau kata keterangan. Teknik ini akan membantu NLP memahami makna dan struktur kalimat. Sebagai contoh, kamu memiliki sebuah kalimat, yaitu “Saya suka membaca buku.”, maka proses POS *tagging*-nya berupa:

- 1) “saya” label kata benda
- 2) “suka” label kata sifat
- 3) “membaca” label kata kerja
- 4) “buku” label kata benda

Pos Tagging

In NLP



Gambar 5.9 Ilustrasi *Part of Speech (POS) Tagging*

Kamu dapat mendalami dan mempraktikkan konsep *Part of Speech* (POS) *Tagging* pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-POSTagging>

d. *Named Entity Recognition* (NER)

Teknik ini digunakan untuk mengenali entitas dalam sebuah kalimat. Misalnya, nama orang atau nama tempat. Sebagai contoh, kamu memiliki sebuah kalimat “Presiden berjumpa dengan Sekjen PBB di Washington.”, maka yang menjadi entitas adalah Presiden sebagai entitas *person*, Sekjen PBB sebagai entitas *person/organisasi*, dan Washington sebagai entitas lokasi. Teknik ini sangat penting dalam aplikasi pencarian data dan ekstraksi informasi.



Gambar 5.10 Ilustrasi *Named Entity Recognition* (NER)

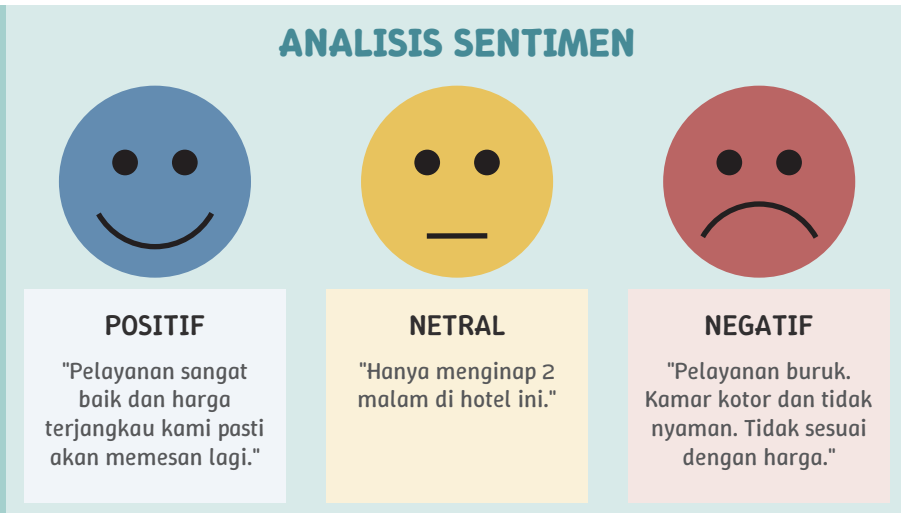
Kamu dapat mendalami dan mempraktikkan konsep *Named Entity Recognition* (NER) pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-NER>


e. Analisis Sentimen

Teknik ini mampu mengenali emosi atau maksud yang ada dalam sebuah teks/kalimat. Apakah emosi tersebut bermakna positif, netral, atau negatif. Teknik ini banyak digunakan dalam aplikasi analisis produk dan analisis media sosial berdasarkan komentar dari pengguna.



Gambar 5.11 Ilustrasi Analisis Sentimen

Kamu dapat mendalami dan mempraktikkan konsep Analisis Sentimen pada *link* berikut.



<https://buku.kemdikbud.go.id/s/KKA11-AnalisisSentimen>



Aktivitas 5.6

Mari Implementasikan NLP (Aktivitas Kelompok)

Kerjakan aktivitas ini dengan mengikuti langkah-langkah berikut.

- Diskusikan dengan teman kelompok (yang sudah dibentuk pada Aktivitas 5.5) tentang implementasi teknologi NLP. Misalnya, membuat chatbot sederhana menggunakan LLM tertentu yang bersifat *opensource*!

- b. Dokumentasikan langkah-langkah implementasi tersebut!
- c. Presentasikan hasil kerja kalian di hadapan guru dan kelompok lain!



Uji Kompetensi

Perhatikan dengan saksama ilustrasi dan instruksi pada setiap soal, lalu jawablah pertanyaannya.

1. Sebuah *decision tree* digunakan untuk memprediksi apakah seseorang akan membeli produk (Ya/Tidak) berdasarkan usia (≤ 30 , > 30) dan pendapatan (Rendah, Tinggi). Pohon keputusan memiliki aturan berikut.
 - a. Jika usia ≤ 30 dan pendapatan = Rendah \rightarrow “Tidak Beli”
 - b. Jika usia ≤ 30 dan pendapatan = Tinggi \rightarrow “Beli”
 - c. Jika usia > 30 \rightarrow “Beli”

Pertanyaan:

Apa prediksi untuk seseorang berusia 25 tahun dengan pendapatan tinggi? Jelaskan langkah-langkahnya!

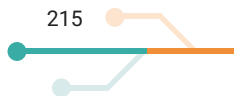
2. Diberikan data latih untuk prediksi harga rumah (dalam juta) berdasarkan luas (m^2) berikut.

Luas (m^2)	Harga
40	Rp200.000.000,00
60	Rp280.000.000,00
80	Rp360.000.000,00

Pertanyaan:

Jika model regresi linier menghasilkan persamaan: $\text{Harga} = 4 \times \text{Luas} + 40$, prediksikan harga rumah dengan luas $70 m^2$! Hitung juga *error* (selisih absolut) jika harga sebenarnya adalah Rp320.000.000,00!

3. Algoritma *K-Means* dengan $k = 2$ mengelompokkan data pelanggan berdasarkan usia dan pengeluaran bulanan (dalam juta).
 - a. *Cluster 1*: Rata-rata usia 25 tahun, pengeluaran 3 juta.
 - b. *Cluster 2*: Rata-rata usia 50 tahun, pengeluaran 8 juta.



Pertanyaan:

Termasuk ke dalam *cluster* manakah seorang pelanggan yang berusia 30 tahun dengan pengeluaran 5 juta? Jelaskan dengan menghitung jarak *euclidean* ke *centroid* masing-masing *cluster*!

4. *Market basket analysis* (MBA) menemukan *rule*:
{Roti, Selai} → {Susu} dengan *support* = 30%, *confidence* = 75%, dan *lift* = 1.5.

Pertanyaan:

- a. Apa arti *confidence* 75% dalam konteks ini?
- b. Mengapa *lift* = 1.5 menunjukkan bahwa aturan ini bernilai?

5. Kamu ingin memprediksi nilai ujian siswa (0–100) berdasarkan jam belajar, jumlah latihan soal, dan akses ke bimbil.

Pertanyaan:

Algoritma *supervised learning* apa yang paling tepat untuk digunakan? Berikan dua alasan mengapa algoritma tersebut dipilih!

6. Sebuah model *K-Means* divisualisasikan dengan *elbow method*, di mana “*elbow*” muncul di $k = 3$.

Pertanyaan:

- a. Apa tujuan *elbow method*?
- b. Mengapa $k = 3$ dipilih sebagai jumlah *cluster* optimal?

7. Tokenisasi adalah proses memecah teks menjadi unit-unit yang lebih kecil, seperti kata atau kalimat.

Teks:

“Belajar NLP itu menyenangkan. Ayo, kita eksplorasi lebih dalam!”

Pertanyaan:

Lakukan tokenisasi kata pada teks di atas! Berapa jumlah token yang dihasilkan?

8. POS *tagging* adalah pemberian label kelas kata (seperti *noun*, *verb*, *adjective*) pada setiap token.

Kalimat:

“Kucing itu berlari cepat di taman.”

Pertanyaan:

Beri label POS *tagging* untuk setiap kata dalam kalimat di atas!

9. *Named Entity Recognition* (NER) adalah identifikasi entitas, seperti nama orang, lokasi, atau organisasi dalam teks.

Teks:

“Presiden berkunjung ke Bandung pada 15 Januari 2024.”

Pertanyaan:

Identifikasi dan klasifikasikan entitas yang ada dalam teks di atas!

10. Analisis sentimen menentukan apakah suatu teks memiliki polaritas positif, negatif, atau netral.

Teks:

“Film ini sangat membosankan, alur ceritanya tidak jelas, dan aktingnya buruk.”

Pertanyaan:

Tentukan sentimen dari teks di atas beserta alasannya!

Pengayaan

Pelajari teknik menyelesaikan masalah melalui pendekatan berpikir komputasional (*computational thinking*) agar kamu mampu mengidentifikasi masalah dan mencari solusi yang efektif dan efisien. Perdalam materi *Machine Learning* dan NLP melalui proyek-proyek yang sederhana, namun mampu melatih logika dan keterampilan kodingmu. Berfokuslah pada kompetensi *data engineering* dan data analisis melalui pendalaman materi bahasa Python, algoritma, dan *Large Language Model* (LLM). *Data engineering* dan data analisis merupakan profesi yang sangat “mahal” saat ini dan masa yang akan datang.

Kamu dapat melatih dan memperdalam kompetensi Kecerdasan Artifisial (KA) serta membangun portofoliomu melalui platform belajar/kursus *online*, seperti:

1. <https://www.kaggle.com/>
2. <https://www.coursera.org/>
3. <https://bisa.ai/>

Selain contoh tersebut, masih banyak lagi platform-platform lain yang dapat membantumu meningkatkan kompetensi KA. Di samping itu, kamu juga dapat mengakses kode QR berikut untuk memperdalam kompetensi Kecerdasan Artifisialmu.



<https://buku.kemdikbud.go.id/s/KKA11-SuplemenBab5>



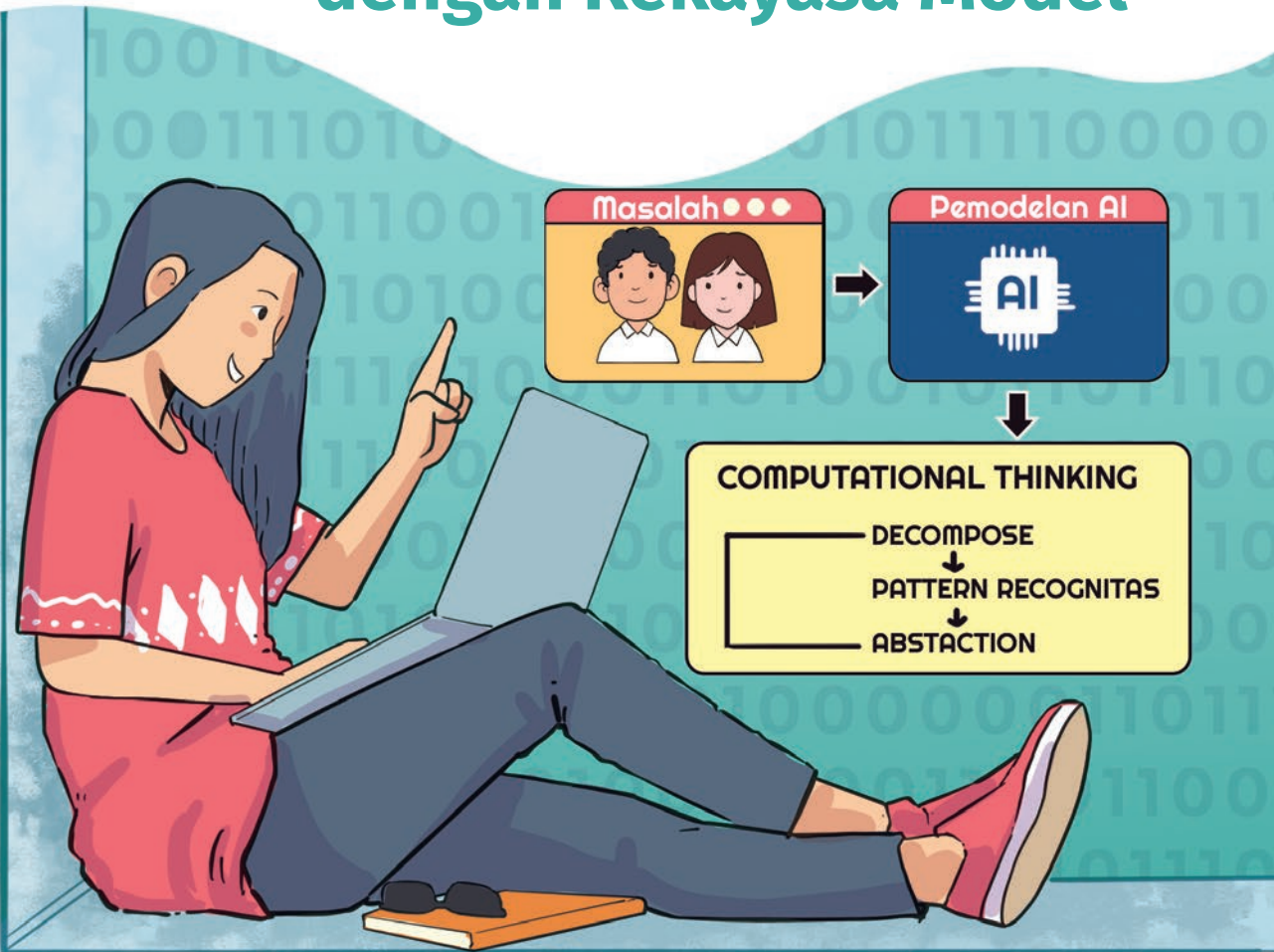
Refleksi

Jawablah pertanyaan berikut dengan jujur berdasarkan pengalamanmu selama mengikuti pembelajaran pada bab ini.

1. Apakah kamu tertarik ketika mengetahui bahwa mesin dapat mengenali kalimat (yang positif atau yang negatif) atau bisa menerjemahkan bahasa hanya dengan data yang dipelajarinya?
2. Apakah kamu tertarik mengetahui bahwa teknologi seperti ini digunakan di dunia nyata dan membuat kamu lebih semangat belajar?
3. Apakah kamu masih merasa kebingungan, bagaimana sebuah model *machine learning* benar-benar bisa “belajar” dari data?
4. Apakah kamu merasa bahwa materi tahapan kerja NLP perlu diulang?
5. Teknologi apa yang dapat kamu ciptakan atau kembangkan setelah menguasai dasar-dasar *machine learning* dan NLP?

Bab 6

Penyelesaian Masalah dengan Rekayasa Model



Di masa depan, masalah semakin kompleks, tetapi teknologi dan Kecerdasan Artfisial (KA) dapat menjadi solusi. Bagaimana manusia menggunakannya untuk menyelesaikan masalah?



Tujuan Pembelajaran

Setelah mempelajari Bab 6, kamu diharapkan mampu mengidentifikasi dan memecahkan masalah-masalah kompleks yang sering muncul di lingkungan sekitar dengan Koding dan KA; memahami tahapan-tahapan dalam proses rekayasa (*engineering process*), mulai dari memahami masalah, merancang, membangun, hingga menguji solusi teknologi; serta mengembangkan model Kecerdasan Artifisial (KA), yaitu teknologi yang bisa “belajar” dan “berpikir” seperti manusia agar kamu lebih siap menghadapi tantangan di dunia nyata.

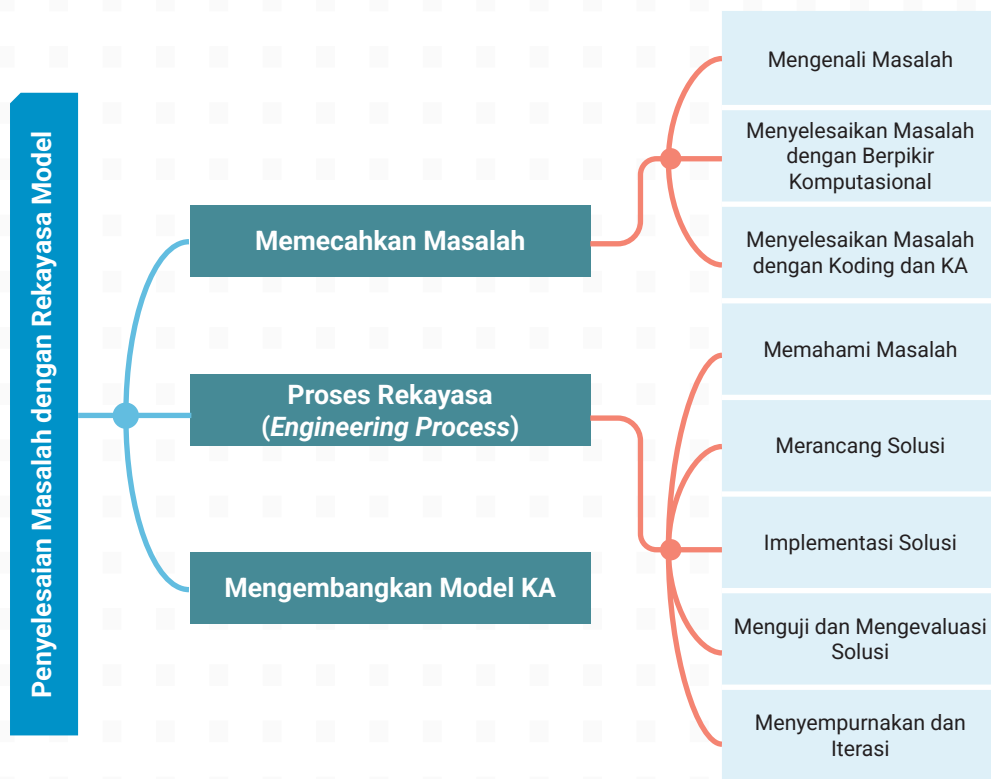


Kata Kunci

masalah, kompleks, berpikir sistematis, rekayasa, model



Peta Materi





Siap-Siap Belajar

Coba deh, kamu amati lingkungan tempat tinggalmu. Pernahkah kamu melihat masalah, seperti tumpukan sampah di jalan, air got yang pampat, atau kurangnya lampu penerangan di jalan? Semua itu adalah contoh masalah nyata yang sering kita temui di lingkungan sekitar.

Pada pembelajaran sebelumnya, kamu sudah mempelajari tentang pentingnya kolaborasi dan berpikir kritis dalam membuat konten digital. Kini pembelajaranmu akan berlanjut ke level berikutnya. Sekarang, coba kamu bayangkan, bagaimana jika kamu bisa membantu menyelesaikan masalah-masalah yang ada di lingkungan sekitarmu dengan cara yang cerdas dan kreatif dengan pendekatan ala *engineer* (observasi, rancang, uji coba, dan perbaikan). Pada pembelajaran ini, kamu akan diajak untuk menerapkan kemampuan tersebut dalam bentuk proyek nyata yang memadukan keterampilan analisis masalah, pemikiran sistematis (*engineering process*), dan kemampuan membuat model kecerdasan buatan sederhana.



Cek Pemahaman Awal

Petunjuk:

Jawablah pertanyaan berikut dengan jujur sesuai pemahamanmu saat ini. Kamu tidak perlu mencari jawaban dari buku atau internet. Latihan ini bukan untuk penilaian, tetapi untuk mengetahui sejauh mana kamu sudah mengenal topik ini.

A. Pertanyaan Pilihan Ganda

Pilihlah satu jawaban yang paling tepat!

1. Apa yang dimaksud dengan “masalah” dalam konteks rekayasa?
 - a. Sesuatu yang harus dihindari.
 - b. Aktivitas yang rutin dilakukan setiap hari.
 - c. Keadaan yang menyenangkan.
 - d. Perbedaan antara harapan dan kenyataan yang membutuhkan solusi.
 - e. Informasi yang tidak relevan.

2. Yang termasuk langkah awal dalam menyelesaikan masalah menggunakan proses rekayasa adalah
 - a. menulis kode program
 - b. menguji model
 - c. memahami masalah secara menyeluruh
 - d. mengumpulkan opini dari sosial media
 - e. menyusun laporan akhir
3. Berpikir komputasional tidak mencakup
 - a. dekomposisi
 - b. pengenalan pola
 - c. intuisi sosial
 - d. abstraksi
 - e. penyusunan algoritma
4. Dalam konteks KA (Kecerdasan Artifisial), data yang digunakan untuk melatih model disebut
 - a. informasi publik
 - b. *dataset*
 - c. metode eksperimen
 - d. lembar kerja
 - e. *database*
5. Salah satu contoh penggunaan KA dalam kehidupan sehari-hari adalah
 - a. mengerjakan PR sendiri
 - b. bermain bola
 - c. filter wajah di media sosial
 - d. belanja di pasar tradisional
 - e. menyiram tanaman dengan ember

B. Isian Singkat

Jawablah pertanyaan berikut secara singkat dan jelas!

1. Sebutkan satu masalah nyata di lingkungan sekitarmu yang ingin kamu selesaikan dengan teknologi!
2. Apa yang kamu ketahui tentang “rekayasa perangkat lunak”? (Jawaban boleh dalam 1–2 kalimat.)
3. Menurutmu, bagaimana teknologi dapat membantu manusia menyelesaikan masalah sosial (seperti masalah kemacetan atau masalah sampah)?

C. Pertanyaan Skala Pemahaman

Jawablah pertanyaan berikut sesuai dengan pengetahuan dan pemahamanmu! Berilah tanda centang (✓) pada kolom yang sesuai!

No.	Pernyataan/Pertanyaan	SP	P	KP	TP
1	Saya dapat menyebutkan minimal tiga permasalahan lingkungan utama yang terjadi di sekitar saya.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Saya memahami bahwa masalah dapat diselesaikan dengan proses berpikir sistematis.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Saya mengetahui langkah-langkah dalam proses rekayasa (<i>engineering process</i>).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Saya dapat menjelaskan konsep berpikir komputasional.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Saya pernah menggunakan atau mengetahui penggunaan Kecerdasan Artifisial (KA).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Saya mengetahui perbedaan antara solusi manual dan solusi berbasis teknologi.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	Saya tertarik untuk menyelesaikan masalah nyata menggunakan pendekatan teknologi dan KA.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	Saya percaya diri membuat solusi digital sederhana untuk permasalahan di sekitar saya.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	Saya pernah mendengar tentang model pembelajaran mesin (<i>machine learning</i>) atau model KA.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

No.	Pernyataan/Pertanyaan	SP	P	KP	TP
10	Saya pernah membangun model KA sederhana.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Keterangan:

SP : Sangat Paham

P : Paham

KP : Kurang Paham

TP : Tidak Paham

A. Memecahkan Masalah

Kamu pasti tahu, ada banyak masalah di masyarakat yang belum terselesaikan secara efektif, seperti akses pendidikan, kesehatan, transportasi, atau sampah. Di era digital, bentuk masalah masyarakat pun ikut berkembang. Penyebaran hoaks, *cyberbullying*, dan kecanduan media sosial menjadi persoalan baru yang membutuhkan pemikiran dan solusi kreatif (Nugroho dan Kristiadi, 2021).



Gambar 6.1 Berbagai Masalah di Sekitar Kita

Masalah bukan hanya urusan orang dewasa, kamu pun setiap hari menghadapi berbagai masalah dan tantangan. Tentunya, kamu membutuhkan

cara berpikir sistematis untuk menyelesaikan masalah-masalah tersebut, bukan hanya ide, tetapi juga proses teknis yang terukur. Sebenarnya, masalah-masalah ini dapat diselesaikan dengan pendekatan *Computational Thinking* (CT), yaitu cara berpikir logis dan sistematis untuk menyelesaikan masalah, mirip seperti cara kerja komputer.

1. Mengenali Masalah

Setiap masalah di masyarakat pasti mempunyai cara tersendiri untuk dikenali (diidentifikasi). Akan tetapi, secara umum ada beberapa langkah yang dapat kamu ikuti agar proses identifikasi masalah menjadi lebih terarah dan efektif. Yuk, kita bahas satu per satu langkah-langkah tersebut!



Gambar 6.2 Langkah-Langkah Identifikasi Masalah

- a. **Lihat dan Catat (Observasi dan Pengumpulan Data Awal)**
Langkah pertama adalah mengamati langsung apa yang terjadi di lingkungan sekitarmu. Coba lihat kondisi sekitar, misalnya bagaimana warga beraktivitas, apa saja yang menjadi perhatian mereka, adakah hal-hal yang tampak tidak berjalan dengan semestinya, atau apa yang membuat warga merasa tidak nyaman. Kamu bisa mencatat, memotret, atau merekam hasil pengamatanmu tersebut agar lebih mudah untuk mengingatnya.
- b. **Ajak Ngobrol Santai (Diskusi dan Wawancara)**
Untuk mengetahui lebih jelas dan akurat tentang permasalahan yang terjadi, ajaklah warga untuk ngobrol. Kegiatan tersebut bisa berupa obrolan santai, diskusi kelompok, ataupun wawancara perorangan. Tanyakan kepada mereka mengenai hal-hal umum, seperti: “Menurut Bapak/Ibu, apa yang menjadi masalah di lingkungan kita ini?” atau “Apa harapan Bapak/Ibu untuk lingkungan kita ke depannya?” Dari obrolan ini, kita bisa mendapatkan cerita-cerita dan sudut pandang yang beragam.
- c. **Periksa dan Pelajari Hasil (Analisis Data dan Identifikasi Pola)**
Apabila data sudah terkumpul, maka saatnya untuk dianalisis. Cermati, apakah ada pola atau tren tertentu. Misalnya, apakah banyak warga mengeluhkan hal yang sama? Siapa yang paling terdampak? Kamu dapat menggunakan data angka (seperti jumlah orang yang tidak punya pekerjaan) maupun cerita-cerita dari warga untuk membentuk gambaran yang lebih lengkap tentang masalah tersebut.
- d. **Tentukan Prioritas (Prioritaskan Masalah dan Tetapkan Kriteria)**
Terkadang, masalah yang ditemukan di masyarakat cukup banyak. Nah, kamu perlu menentukan masalah mana yang harus ditangani terlebih dahulu. Hal itu bisa dilihat dari seberapa mendesaknya masalah tersebut, seberapa besar dampaknya, dan apakah ada sumber daya yang cukup untuk menyelesaikannya. Misalnya, masalah yang berdampak kepada banyak orang dan dapat segera diselesaikan bisa dijadikan prioritas utama.
- e. **Tuliskan Masalah (Rumuskan Masalah secara Spesifik dan Terukur)**
Setelah mengetahui masalah mana yang ingin diselesaikan, kamu perlu merumuskannya dengan jelas. Hindari kalimat yang terlalu umum, seperti “masalah kebersihan”. Lebih baik, masalah tersebut dirumuskan

dengan “Kurangnya fasilitas pengelolaan sampah di lingkungan desa menyebabkan penumpukan sampah di area publik.” Kalimat seperti ini akan memudahkanmu ketika membuat solusi dan mengevaluasi hasilnya.

f. **Cek dan Ricek (Validasi dan Verifikasi)**

Langkah terakhir, jangan lupa untuk mengonfirmasi masalah yang sudah dirumuskan. Ajak tokoh masyarakat, perangkat desa, atau orang yang ahli di bidangnya untuk mengecek apakah masalah tersebut benar-benar penting dan relevan. Proses ini penting agar semua pihak merasa dilibatkan dan siap bekerja sama mencari solusi.

“Kita tidak dapat menyelesaikan masalah dengan pemikiran yang sama seperti yang kita gunakan saat kita membuatnya.”

- Albert Einstein



Aktivitas 6.1

Mengidentifikasi Masalah di Lingkungan Sekitar (Aktivitas Kelompok)

Melalui Aktivitas 6.1, kalian akan belajar mengidentifikasi masalah yang ada di lingkungan sekitar. Ikuti petunjuk berikut untuk pengerjaannya.

- Bentuklah kelompok yang terdiri atas 3–5 orang!
- Diskusikan dengan anggota kelompokmu tentang permasalahan yang terjadi di lingkungan sekolah, lingkungan tempat tinggal, dan lingkungan masyarakat lainnya!
- Lakukan tahapan-tahapan identifikasi masalah seperti yang sudah dijelaskan di atas!
- Presentasikan hasil identifikasi dan rumusan masalah tersebut kepada guru, tokoh masyarakat, atau pihak terkait!
- Mintalah umpan balik (Apakah masalah tersebut benar-benar penting, relevan, dan dapat diselesaikan?)!

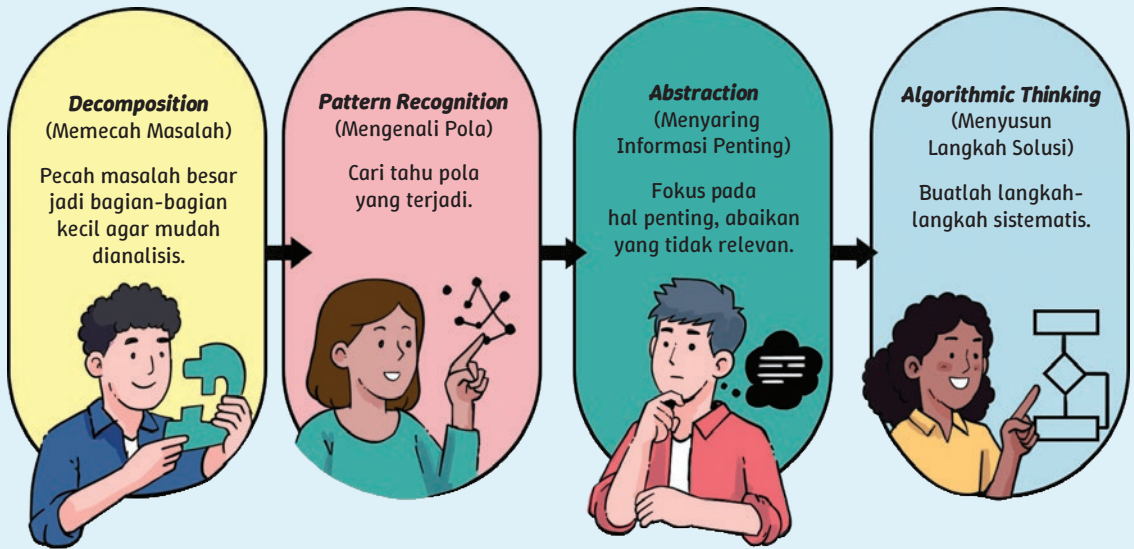
2. Menyelesaikan Masalah dengan Berpikir Komputasional

Kamu pasti pernah menghadapi situasi yang terasa tidak sesuai harapan, kan? Ketika terjadi perbedaan antara kenyataan dengan apa yang diharapkan, itulah yang dinamakan dengan masalah (Niam et al., 2024). Untuk menyelesaikannya, kamu dapat menggunakan cara berpikir komputasional (Syarif, 2019).

Pada materi sebelumnya, kamu sudah memahami tentang berpikir komputasional. Berpikir komputasional adalah kemampuan memecahkan masalah secara logis, sistematis, dan efisien dengan menggunakan konsep dasar ilmu komputer dan informatika. Keterampilan ini mencakup penalaran logis, kritis, dan kreatif berbasis data yang dapat diterapkan, baik secara mandiri maupun kolaboratif serta dapat dijalankan oleh manusia maupun mesin (Bebras Indonesia, n.d.; Tresnawati et al., n.d.; Kepka BSKAP No. 032 Tahun 2024, hlm. 239). Orientasi kritis dalam berpikir komputasional menekankan bahwa pentingnya menyelesaikan masalah tidak hanya secara sistematis, tetapi juga harus mempertimbangkan aspek etika, dampak sosial, dan keberlanjutan solusi.

Berpikir komputasional dan berpikir kritis saling berkaitan, karena keduanya membentuk cara berpikir logis dan mendalam untuk menghasilkan solusi yang efektif (Kemendikbud Ristek, 2024; Meitjing & Fuad, 2023). Google for Education mengidentifikasi bahwa terdapat empat pilar berpikir komputasional, yaitu mencakup dekomposisi, pengenalan pola, abstraksi, dan berpikir algoritma (Adi Mulyanto dan Yasya Rusyida, 2020).

Berpikir Komputasional



Gambar 6.3 Pilar Berpikir Komputasional

a. *Decomposition (Memecah Masalah)*

Pecahkan masalah besar menjadi bagian-bagian kecil agar mudah dianalisis. Contoh: Masalah sampah dipecah menjadi beberapa bagian, yaitu pengumpulan, pemilahan, dan pengangkutan.

b. *Pattern Recognition (Mengenali Pola)*

Cari tahu pola yang terjadi. Misalnya: Sampah paling banyak menumpuk di dekat kantin.

c. *Abstraction (Menyaring Informasi Penting)*

Fokuslah kepada hal penting dan abaikan yang tidak relevan. Contoh: Fokus pada jenis sampah yang paling sering ditemukan.

d. *Algorithmic Thinking (Menyusun Langkah Solusi)*

Buat langkah-langkah sistematis. Misalnya:

- 1) Sediakan tempat sampah organik dan anorganik.
- 2) Jadwalkan pengangkutan.
- 3) Buatlah tim piket.

Jika kamu terbiasa menggunakan pendekatan berpikir komputasional, kamu akan lebih siap menghadapi berbagai tantangan, tidak hanya pada Pelajaran Koding dan Kecerdasan Artifisial saja, tetapi juga dalam kehidupan sehari-hari.

“Saya belum pernah gagal. Saya baru saja menemukan 10.000 cara yang tidak berhasil.” - Thomas Alva Edison



Aktivitas 6.2

Menyusun Solusi dengan Pendekatan Berpikir Komputasional (Aktivitas Kelompok)

Pada Aktivitas 6.1 kalian telah berlatih cara mengidentifikasi masalah. Kini, kalian akan berlatih cara menyelesaikan masalah melalui pendekatan komputasional dengan langkah-langkah berikut.

- Diskusikanlah dengan anggota kelompok kalian tentang penyelesaian dari masalah (pada Aktivitas 6.1) melalui pendekatan berpikir komputasional dalam konteks yang konkret!
- Presentasikan hasil diskusi tersebut di hadapan guru dan kelompok lain!
- Mintalah umpan balik kepada guru dan anggota kelompok lain tentang ketepatan langkah-langkah penyelesaian masalah yang telah kalian presentasikan!

3. Menyelesaikan Masalah dengan Koding dan KA

Kamu telah mempelajari tentang cara mengenali masalah, lalu mencari solusinya dengan cara berpikir komputasional. Sekarang, saatnya bagimu untuk naik level, yaitu menyelesaikan masalah langsung dengan koding dan Kecerdasan Artifisial (KA).

Kecerdasan Artifisial dapat diibaratkan seperti “otak digital” yang bisa diajari untuk berpikir dan belajar, mirip seperti otak manusia. Nah, agar KA dapat bekerja dengan baik, kamu perlu memberinya data dan mengatur logikanya melalui koding atau pemrograman. Gabungan keduanya (koding dan KA) bisa menghasilkan solusi yang cerdas, efisien, dan dapat diandalkan. Lalu, bagaimana KA dan koding dapat membantumu menyelesaikan masalah?

a. **Mengumpulkan dan Mengalisis Data Melalui Kode**

Kamu bisa menulis kode program (misalnya Python atau Java) untuk mengumpulkan data dari media sosial, sensor, atau survei daring. Lalu, gunakan KA untuk menganalisis data tersebut untuk menemukan pola tersembunyi. Misalnya, kamu mau mengetahui apa masalah yang paling banyak dikeluhkan warga? Hal itu sangat memungkinkan untuk dapat dianalisis menggunakan koding dan model pembelajaran mesin (*machine learning*).

b. **Membuat Peta Masalah Melalui Visualisasi**

Dengan koding, kamu bisa membuat visualisasi data menggunakan *library*, seperti Matplotlib atau Seaborn (Scikit-learn, n.d.) untuk membuat peta interaktif yang menunjukkan lokasi-lokasi dengan masalah tertentu (misalnya titik rawan banjir atau kemacetan). Visualisasi ini dapat membantu orang-orang untuk memahami masalah dengan lebih cepat.

c. **Prediksi Masalah Sebelum Terjadi**

Dengan koding, kamu bisa membuat model KA untuk memprediksi kejadian di masa depan. Contohnya: “Apakah minggu depan berpotensi hujan deras di sekitar sekolah?” atau “Apakah siswa dengan jam belajar rendah cenderung mendapatkan nilai di bawah KKM?”. Model tersebut dapat dilatih dari data sebelumnya, lalu ditebak menggunakan kode program. Hal ini berguna untuk mengambil tindakan lebih awal.

d. **Otomatisasi Tugas Rutin**

Kamu bisa membuat chatbot sederhana dengan koding yang dilatih menggunakan KA untuk membantumu menjawab pertanyaan secara otomatis. Selain itu, kamu juga bisa membuat sistem laporan otomatis dari data yang masuk. Jadi, pekerjaan manual bisa di-*handle* oleh sistem dan kamu bisa fokus ke hal lain yang lebih penting. Semua itu dapat dilakukan dengan logika yang dibuat melalui koding, lalu ditambah dengan kemampuan KA, seperti *Natural Language Processing* (NLP).

e. **Solusi yang Disesuaikan**

KA dapat digunakan untuk memberi rekomendasi personal. Kamu bisa membuat aplikasi edukasi sederhana yang menggunakan KA untuk memberi materi belajar yang pas sesuai dengan kemampuan masing-masing siswa. Semua logika tersebut dapat diatur melalui kode program.

f. Melibatkan Banyak Orang Melalui Aplikasi

Dengan koding, kamu dapat membuat aplikasi partisipatif. Sehingga, ketika warga melaporkan masalah lingkungan, KA akan membantu mengelompokkan laporan-laporan tersebut secara otomatis agar mudah untuk ditindaklanjuti.

Coba kamu perhatikan contoh sederhana berikut. Misalnya, kamu menemukan masalah tentang banyak siswa yang susah untuk bangun pagi. Bagaimana solusinya? Buatlah aplikasi alarm pintar yang dapat mempelajari pola tidur siswa dan memberi saran waktu tidur yang ideal! Dengan koding dan KA, kamu bisa membuat sistem yang dapat memberi notifikasi otomatis dan terus belajar dari data pengguna.

Koding itu ibarat bahasa. Jika kamu bisa ‘berbicara’ dengan komputer, kamu bisa memintanya untuk membantumu menyelesaikan banyak hal. Kecerdasan buatan bukan hanya tentang teknologi canggih, tetapi tentang bagaimana menggunakannya untuk membuat hidup menjadi lebih baik (Dinata dan Fikry, 2014). Jika kamu dapat menggabungkan logika berpikir, koding, dan KA, maka kamu akan mempunyai kekuatan super untuk membuat perubahan nyata di sekitarmu.

Walaupun KA dan koding itu sangat keren, tetapi kamu harus tetap bijak dalam menggunakannya. Jangan sembarangan mengambil data orang, jangan membuat sistem yang bisa menyebarkan hoaks, dan pastikan tujuanmu menggunakannya adalah untuk membantu, bukan untuk merugikan.

Teknologi yang hebat bukanlah tentang kerumitannya, tetapi tentang bagaimana teknologi itu dapat digunakan untuk membantu orang lain. Jadi, apa yang harus kamu lakukan?

- a. Temukan masalah nyata di sekitarmu.
- b. Buatlah solusi dari masalah tersebut dengan berpikir komputasional.
- c. Kembangkan ide tersebut menjadi aplikasi atau sistem kecil menggunakan koding.
- d. Tambahkan fitur KA sederhana, seperti prediksi atau klasifikasi.
- e. Uji dan perbaiki berdasarkan data dan masukan.

"Teknologi hanyalah alat. Namun, untuk menjadikan anak-anak bisa saling bekerja sama dan termotivasi, guru adalah yang paling penting." - Bill Gates



Aktivitas 6.3

Merancang Proyek Solusi dengan Koding dan KA (Aktivitas Kelompok)

Ikuti langkah-langkah berikut untuk mengerjakan Aktivitas 6.3 ini.

- Bentuklah kelompok yang terdiri atas 3–5 orang!
- Bersama dengan anggota kelompok, persiapkan laporan tertulis menggunakan kriteria penulisan tertentu tentang masalah yang telah dipilih sebelumnya!
- Lakukan presentasi yang berfokus kepada penyelesaian masalah menggunakan pendekatan berpikir komputasional yang dimulai dari tahap identifikasi masalah!
- Gunakan teknologi Kecerdasan Artifisial sebagai alat bantu untuk menyelesaikan proyek tersebut!

B. Proses Rekayasa (*Engineering Process*)

Pernahkah kamu berpikir tentang bagaimana aplikasi keren seperti Google Maps dapat memberi tahumu jalan tercepat? Atau bagaimana kamera di ponsel bisa secara otomatis mengenali wajahmu? Semua itu tentu tidak terjadi secara tiba-tiba. Di balik berbagai solusi canggih tersebut, ada sebuah pendekatan sistematis yang disebut dengan *engineering process* atau proses rekayasa.

Istilah rekayasa berasal dari kata *engineering* dalam bahasa Inggris, yang berarti penerapan ilmu pengetahuan dan matematika untuk merancang, membangun, dan memelihara struktur atau sistem guna menyelesaikan masalah nyata (Suharyadi, 2017). Artinya, rekayasa bukan hanya tentang membuat alat atau mesin, tetapi lebih luas dari itu, yaitu mencakup cara berpikir logis dan terstruktur untuk menciptakan solusi yang efektif dan efisien.

Proses rekayasa (*engineering process*) adalah cara berpikir dan bekerja yang terstruktur untuk mengubah masalah menjadi solusi nyata. Proses ini membantu dalam merancang, membangun, dan mengevaluasi solusi secara



bertahap. Sebenarnya, apabila kamu pernah membuat proyek mini di sekolah atau melakukan eksperimen sederhana, maka kamu sudah menerapkan bagian dari proses ini. Menurut ITEEA (2020) dan Bybee (2013), proses rekayasa menjadi fondasi penting dalam dunia teknologi karena mendorong untuk berpikir kritis, kreatif, dan sistematis.

“Engineering is the application of scientific principles to design or develop structures, machines, apparatus, or manufacturing processes.”
— American Engineers’ Council for Professional Development —

Proses rekayasa ini tidak hanya digunakan di dunia digital saja, melainkan digunakan juga di berbagai bidang, di antaranya sebagai berikut.

1. Rekayasa sipil digunakan untuk merancang jembatan, gedung, dan jalan.
2. Rekayasa biomedis digunakan untuk menciptakan alat bantu kesehatan, seperti prostetik.
3. Rekayasa perangkat lunak digunakan untuk membangun sistem digital, seperti aplikasi dan *website*.

Untuk lebih memahami proses tersebut, kamu dapat membayangkan proses rekayasa itu seperti resep masakan. Nah, dalam dunia digital, ada versi khusus untuk resep tersebut yang dikenal sebagai rekayasa perangkat lunak (*software engineering*). Proses ini digunakan untuk membangun sistem dan aplikasi digital. Misalnya, saat kamu membuat aplikasi pengingat belajar, kamu akan melalui serangkaian langkah, yaitu menentukan fitur, merancang antarmuka, menulis kode program, lalu menguji apakah aplikasi itu berjalan sesuai harapan. Semua langkah tersebut merupakan bagian dari *software engineering*.

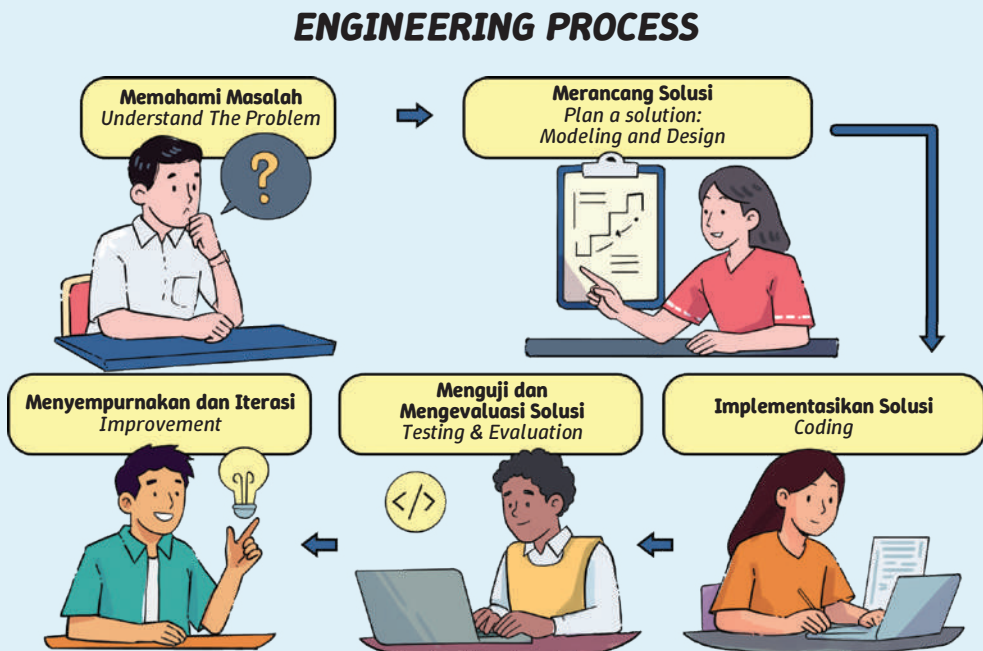
Membangun model KA juga tidak jauh berbeda, yaitu mengikuti langkah-langkah khas proses rekayasa. Dimulai dari memahami masalah yang ingin diselesaikan, lalu merancang solusi berupa model KA, membangunnya dengan cara melatih model tersebut menggunakan data, hingga akhirnya menguji dan mengevaluasi efektivitas solusi tersebut. Pendekatan sistematis ini menunjukkan bahwa KA bukan hanya sekadar teknologi canggih, tetapi juga merupakan hasil dari penerapan prinsip rekayasa.

Proses rekayasa adalah jembatan antara masalah dan solusi dengan teknologi. Kamu bisa menjadi arsiteknya di masa depan. Di bidang teknologi informasi,

prinsip rekayasa sangat penting karena dapat menjembatani ide dan inovasi dengan implementasi nyata yang bisa digunakan oleh banyak orang. Tanpa pendekatan rekayasa, ide-ide hebat akan sulit diwujudkan dalam bentuk nyata yang dapat digunakan dan diandalkan.

Lalu, apa itu rekayasa perangkat lunak dan bagaimana hal tersebut dapat menyelesaikan masalah melalui teknologi? Rekayasa perangkat lunak (*software engineering*) adalah proses sistematis dalam mengembangkan perangkat lunak, mulai dari memahami masalah, merancang solusi, menulis kode, menguji hasilnya, hingga melakukan penyempurnaan (Sommerville, 2016). Menurut Pressman dan Maxim (2020), *software engineering* tidak hanya berkutat seputar koding saja, tetapi juga melibatkan perencanaan, pengelolaan, dan evaluasi agar sistem yang dikembangkan bisa tepat guna dan dapat dipelihara dalam jangka panjang.

Pada pembelajaran ini, kamu akan mencoba menerapkan prinsip rekayasa untuk menyelesaikan masalah nyata melalui pembuatan proyek berbasis aplikasi dan Kecerdasan Artifisial.



Gambar 6.4 Proses Rekayasa (*Engineering Process*)

1. Memahami Masalah (*Understand The Problem*)

Sebelum sibuk membuat solusi, pastikan dahulu apa masalahnya. Tahap pertama dalam rekayasa adalah memahami masalah secara jelas dan spesifik. Ibaratnya seperti seorang dokter, sebelum memberi obat, tentu harus mengetahui terlebih dahulu apa penyakit pasiennya. Apabila hanya menebak-nebak, kemungkinan solusinya juga akan salah sasaran.

Jika pada bagian sebelumnya kamu sudah memahami cara mengidentifikasi masalah, sekarang waktunya kamu memahami analisis kebutuhan. Dalam disiplin rekayasa perangkat lunak, tahap ini disebut dengan *requirement elicitation* dan *requirement analysis*. Tahap ini bertujuan untuk mencari tahu siapa saja yang akan menggunakan? Apa saja yang sebenarnya dibutuhkan oleh pengguna? Apa batasan sistemnya? Dan apa tujuan akhirnya? Jadi, bukan asal membuat, tetapi membuat yang benar-benar dibutuhkan.

Nah, di tahap ini kamu akan melakukan wawancara dan menggali informasi. Misalnya, kamu dapat bertanya langsung kepada orang yang mengalami masalah. Orang tersebut boleh jadi guru, teman, warga sekitar, atau siapa pun. Catatlah kebutuhan dan kendalanya. Contohnya, jika kamu mau membuat aplikasi pengingat belajar, kamu perlu mengetahui siapa penggunanya? Fitur apa yang mereka butuhkan? Dan apa yang mereka tidak suka dari aplikasi serupa? Rumuskan tujuan yang jelas. Tujuan dari penggalian informasi ini harus dapat terukur. Bukan sekadar “agar lebih bagus”, tetapi harus terukur, misalnya “membantu siswa untuk mengatur waktu belajarnya dan agar tidak lupa mengerjakan tugas”.

Mengapa tahapan ini sangat penting? Berikut alasannya.

- a. Solusi menjadi lebih tepat guna karena kamu sangat memahami masalahnya.
- b. Tidak mudah berubah-ubah. Jika dari awal sudah jelas kebutuhannya, maka tidak perlu direvisi terus.
- c. Kualitas produk meningkat. Aplikasi atau solusi yang kamu buat pasti lebih sesuai dengan kebutuhan dan mudah digunakan.
- d. Hemat waktu dan biaya. Tidak perlu bolak-balik untuk perbaikan, sehingga menjadi lebih efisien.

Jadi, sebelum memulai untuk membangun sesuatu, pastikan terlebih dahulu kamu sudah benar-benar mengerti apa yang ingin diselesaikan. Jangan terburu-buru menggunakan coding, sebelum jelas masalahnya, ya! Menurut Fauzi (2018) dan Pressman dan Maxim (2020), keberhasilan proyek perangkat lunak sangat ditentukan oleh sejauh mana masalah dipahami dengan baik sejak awal.



Aktivitas 6.4

Mendeskripsikan Permasalahan dan Kebutuhan Sistem (Aktivitas Kelompok)

Ikuti petunjuk berikut untuk mengerjakan Aktivitas 4.6 ini.

- a. Identifikasi masalah yang ada di sekitar kalian (sekolah atau tempat tinggal kalian), misalnya siswa susah bangun pagi, banyak makanan sisa di kantin, dan sebagainya!
- b. Ambillah salah satu masalah yang akan dikembangkan menjadi perangkat lunak!
- c. Diskusikan dengan anggota kelompok, lalu buatlah uraian singkat untuk mendeskripsikan permasalahan tersebut!
- d. Gunakan 5W+1H untuk memudahkan kalian mendeskripsikan kebutuhan perangkat lunak tersebut!
 - 1) *What*: Deskripsikan perangkat lunak apa yang akan dibuat (Berisikan setidaknya fitur utama apa saja yang dimiliki oleh perangkat lunak tersebut)!
 - 2) *Why*: Deskripsikan permasalahan apa yang mendasari fitur-fitur tersebut harus ada dalam perangkat lunak yang akan dikembangkan!
 - 3) *How*: Deskripsikan bagaimana cara kerja setiap fitur yang akan ada pada perangkat lunak yang akan dikembangkan!
 - 4) *Who*: Uraikan siapa saja yang berkepentingan dengan perangkat lunak tersebut (masing-masing peran mempunyai tugas dan tanggung jawab seperti apa)!
 - 5) *Where*: Tentukan di mana perangkat lunak tersebut akan diimplementasikan!
 - 6) *When*: Kapan estimasi waktu untuk perangkat lunak tersebut rilis?



2. Merancang Solusi (*Plan A Solution: Modeling and Design*)

Setelah kamu benar-benar memahami masalah dan kebutuhan pengguna, kini waktunya untuk membuat rancangan solusi. Tapi tunggu dulu, jangan langsung loncat ke koding, ya! Dalam dunia rekayasa perangkat lunak, tahapan ini sangat krusial karena menjadi jembatan antara ide dan implementasi nyata.

Sebelum mulai mengetik kode program, kamu perlu menyusun rencana atau desain sistem yang akan dibuat. Ibarat membangun rumah, maka akan membutuhkan gambar denah terlebih dahulu. Nah, desain sistem ini adalah denah dari aplikasi atau program yang akan kamu buat.

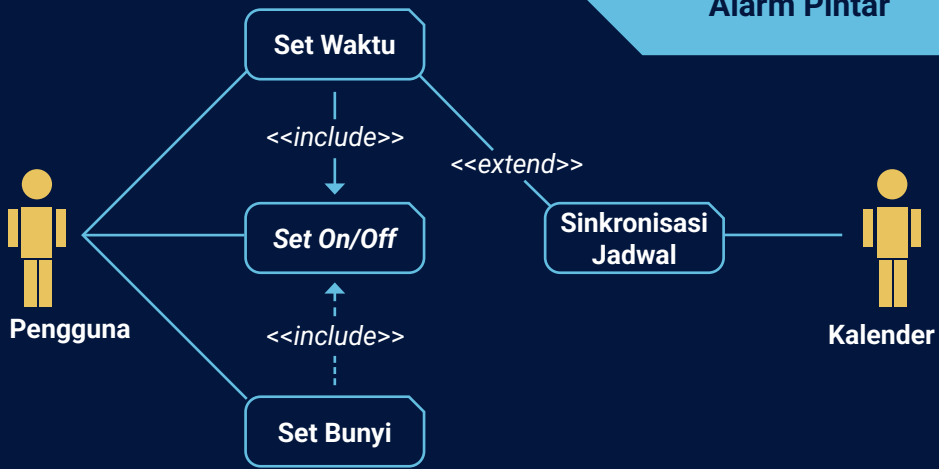
Menurut Pressman dan Maxim (2020), desain perangkat lunak adalah proses teknis yang berfokus pada menciptakan representasi struktur sistem, mulai dari bagaimana komponen-komponennya berinteraksi hingga bagaimana data mengalir di dalamnya. Dengan kata lain, desain sistem adalah “peta jalan” yang memandu dari ide awal menuju ke sistem digital yang nyata dan dapat digunakan.

Dalam praktik rekayasa perangkat lunak (*software engineering*), tahap desain ini biasanya merupakan hasil dari analisis kebutuhan (*requirement analysis*) dan menjadi dasar dari tahapan selanjutnya, yaitu implementasi (koding). Proses ini dijelaskan dalam berbagai model pengembangan perangkat lunak, seperti Waterfall, Agile, maupun Design Thinking. Dalam model Waterfall, desain dilakukan secara menyeluruh di awal. Sedangkan, pada Agile dan Design Thinking, desain bisa dilakukan secara iteratif, yaitu terus disesuaikan dengan masukan dan perubahan kebutuhan.

Menurut Sommerville (2016), desain sistem yang baik harus modular, *reusable*, dan *maintainable*. Hal ini berarti bahwa komponen-komponen sistem harus dapat dipakai ulang dan mudah diperbaiki jika ada perubahan kebutuhan di masa depan. Lalu, apa saja yang harus dirancang? Kamu seharusnya merancang hal-hal berikut terlebih dahulu.

a. Model Fungsional (*Functional Model*)

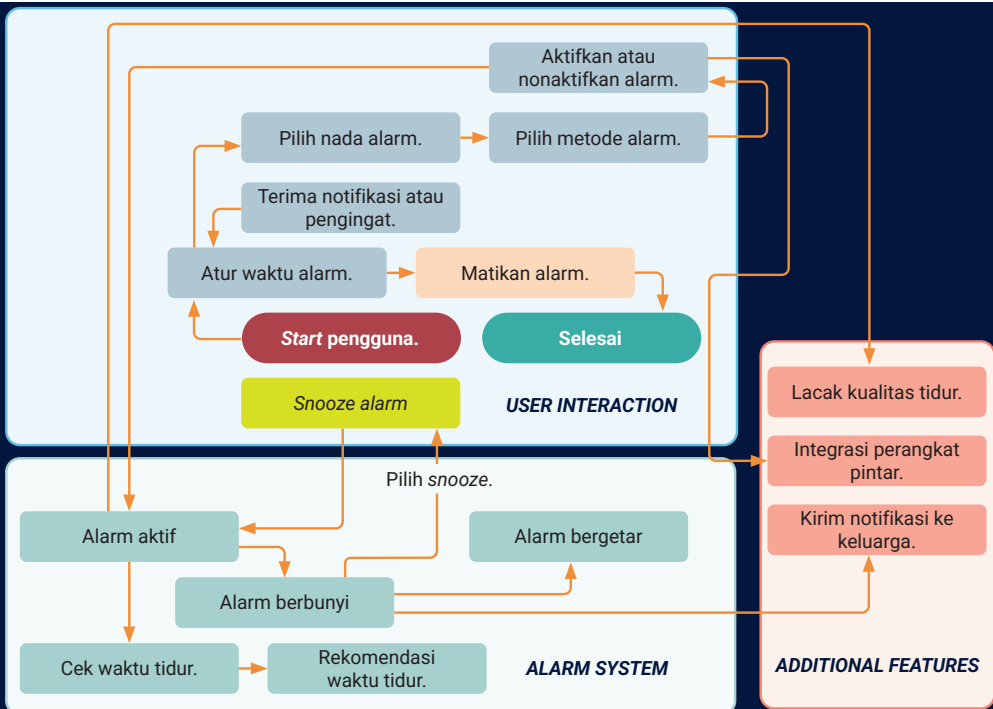
Model fungsional menjelaskan tentang apa yang dilakukan oleh sistem. Biasanya model ini direpresentasikan dalam bentuk *Use Case Diagram* atau *DFD (Data Flow Diagram)*.



Gambar 6.5 Contoh Use Case Diagram Alarm Pintar Bangun Pagi

b. Model Proses (Process Model)

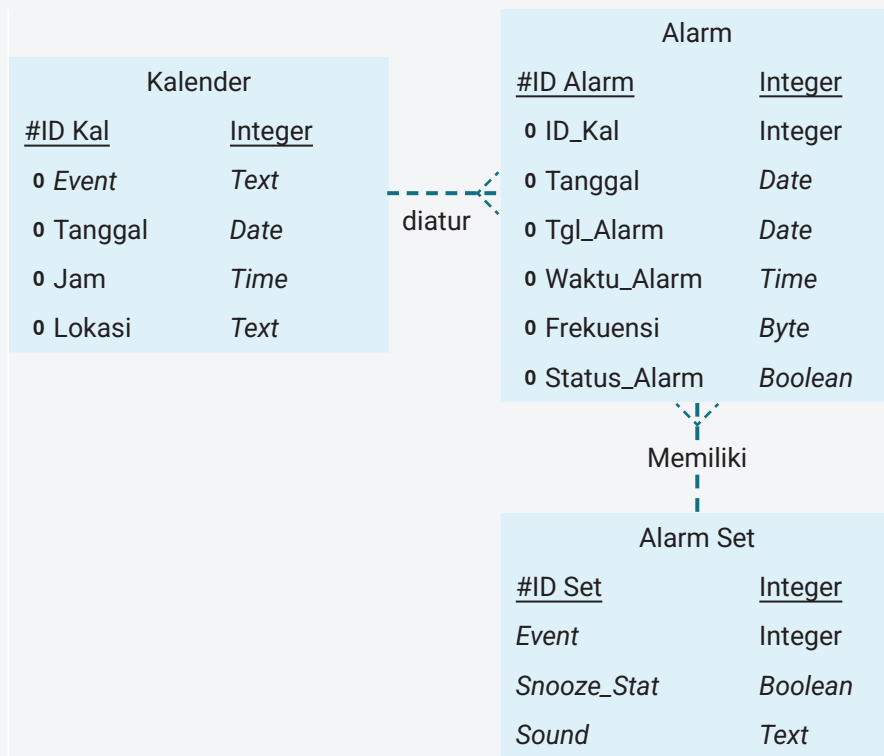
Model proses menggambarkan urutan aktivitas sistem dan interaksinya, misalnya dengan Flowchart atau Activity Diagram.



Gambar 6.6 Contoh Flowchart untuk Alarm Pintar Bangun Pagi

c. Model Data (*Data Model*)

Model data menggambarkan struktur data yang digunakan, seperti *Entity-Relationship Diagram (ERD)*.



Gambar 6.7 Contoh ERD untuk Alarm Pintar Bangun Pagi

d. Model Antarmuka (*UI/UX Mockup*)

Model antarmuka menunjukkan bagaimana tampilan aplikasi dan bagaimana pengguna berinteraksi dengannya.



Gambar 6.8 Contoh Antarmuka Alarm Pintar Bangun Pagi

Lalu, bagaimana memodelkannya? Kamu dapat menggunakan alat-alat bantu yang disebutkan pada tabel berikut. Kamu juga dapat memulainya dari cara manual, digital biasa, hingga alat berbasis Kecerdasan Artificisial (KA).

Tabel 6.1 Alat Bantu Praktik Rekayasa Perangkat Lunak

Tujuan	Alat Manual	Alat Digital	Alat Berbasis (KA)
Sketsa UI/UX	kertas dan pensil	Figma – https://www.figma.com Canva – https://www.canva.com Balsamiq – https://balsamiq.com	Uizard - https://uizard.io Visily – https://www.visily.ai
Diagram Proses/ <i>Flowchart</i>	<i>post-it</i> dan papan tulis	Lucidchart – https://www.lucidchart.com Draw.io – https://draw.io Visio – https://www.microsoft.com/en/microsoft-365/visio/	Eraser.io - https://www.eraser.io/diagramgpt
ERD/Struktur Data	kartu indeks	dbdiagram.io – https://dbdiagram.io MySQL Workbench – https://www.mysql.com/products/workbench/	DiagramGPT – https://diagramgpt.vercel.app
Kolaborasi Desain	diskusi kelompok	Miro – https://miro.com Notion – https://www.notion.so Trello – https://trello.com	Notion AI – https://www.notion.so/product/ai Miro AI – https://miro.com/ai/
Data Sampel (<i>Mock Data</i>)	tulisan manual	Mockaroo – https://mockaroo.com Excel, Google Sheets	Generatedata – https://www.generatedata.com

Sehingga, dengan model-model tersebut, kamu akan dapat:

- Memastikan solusi yang dirancang sesuai dengan kebutuhan pengguna;
- Meminimalkan risiko kesalahan saat pengembangan;
- Membagi pekerjaan secara sistematis jika dilakukan secara tim; serta
- Mempermudah proses evaluasi dan pengujian nantinya.



Aktivitas 6.5

Membuat Model dan Desain Solusi Perangkat Lunak (Aktivitas Individu)

Pada Aktivitas 6.4, kamu sudah memahami masalah dan menemukan kebutuhan-kebutuhan pengguna. Berdasarkan hasil tersebut, buatlah model dan desain perangkat lunak yang meliputi:

- a. Model Fungsional
- b. Model Proses
- c. Model Data
- d. Antarmuka

Paparkan hasilnya untuk mendapatkan umpan balik.

3. Implementasikan Solusi (Coding)

Setelah desain selesai dibuat, tahapan selanjutnya adalah mengubah desain tersebut menjadi bentuk nyata melalui proses koding atau pemrograman. Kamu harus menuliskan instruksi-instruksi logis dalam bentuk bahasa pemrograman untuk menciptakan solusi digital yang dapat dijalankan oleh komputer.

Koding adalah cara manusia “berbicara” dengan komputer. Melalui bahasa pemrograman, seperti Python, C, C#, Java maupun bahasa pemrograman lainnya, kamu dapat mengubah rancangan sistem menjadi aplikasi nyata yang bisa melakukan perhitungan, menampilkan data, bahkan membuat keputusan otomatis melalui Kecerdasan Artifisial (KA). Dalam konteks rekayasa perangkat lunak, tahapan ini disebut sebagai implementasi (*implementation*) dan sangat menentukan keberhasilan proyek secara teknis (Pressman dan Maxim, 2020).

Sebenarnya kamu sudah mempelajari implementasi pada materi sebelumnya, baik pemrograman terstruktur maupun pemrograman berorientasi objek. Agar koding yang kamu buat berjalan dengan baik, berikut beberapa teknik dasar dan lanjutan yang dapat kamu terapkan.

Tabel 6.2 Teknik Dasar dan Lanjutan dalam Koding

Teknik	Penjelasan Singkat	Contoh/Tools
<i>Modular Programming</i>	Membagi program menjadi bagian-bagian kecil (fungsi/modul) yang independen.	Fungsi <code>def</code> di Python
<i>Control Flow</i> (Alur Kendali)	Penggunaan <code>if-else</code> , <i>looping</i> (<code>for</code> , <code>while</code>) untuk mengontrol jalannya program.	Struktur percabangan & perulangan
<i>Error Handling</i>	Penanganan kesalahan program agar tidak <i>crash</i> .	<code>try-except</code> di Python
<i>Object-Oriented Programming</i> (OOP)	Membuat kode berdasarkan objek dan kelas. Cocok untuk aplikasi skala menengah dan besar.	<code>class</code> , <code>__init__</code>
<i>Use of Libraries</i>	Memanfaatkan pustaka (<i>library</i>) yang sudah tersedia agar efisien.	<code>scikit-learn</code> , <code>pandas</code> , <code>numpy</code>
<i>Version Control</i>	Menyimpan dan melacak perubahan kode. Cocok untuk kerja tim.	Git, GitHub
<i>Agile Development</i>	Menerapkan metode pengembangan bertahap, iteratif, dan fleksibel.	Sprint, Scrum Board

Berikut *tools* dan bahasa yang dapat digunakan dalam pengembangan perangkat lunak.

Tabel 6.3 *Tools* dan Bahasa dalam Pengembangan Perangkat Lunak

Bahasa/Tools	Kegunaan
Python	Bahasa populer untuk pemula dan proyek KA.
Streamlit	Untuk membuat aplikasi <i>web</i> interaktif dengan Python.
Jupyter/Google Colab	Untuk eksperimen data dan pemodelan KA interaktif.
VS Code/PyCharm	<i>Editor</i> kode untuk pengembangan aplikasi secara penuh.
Git dan GitHub	Untuk kolaborasi tim dan pengelolaan versi kode.



Lalu bagaimana tips koding yang efektif? Kamu dapat melakukan langkah-langkah berikut.

- Tulis komentar (`# komentar`) agar kode mudah dipahami.
- Gunakan nama variabel yang jelas dan bermakna.
- Uji fungsi secara bertahap (*unit testing*).
- Gunakan *debugging tools* untuk melacak error.
- Simpan perubahan kode secara rutin.



Aktivitas 6.6

Mengimplementasikan Prototipe Aplikasi Solusi (Aktivitas Individu)

Pada Aktivitas 6.5, kamu sudah membuat rancangan perangkat lunak. Sekarang, kamu akan mengimplementasikan prototipe aplikasi solusi dengan langkah-langkah berikut.

- Lakukan koding untuk setiap fungsional yang sudah dirancang!
- Buatlah prototipe aplikasi yang akan dikembangkan dengan menggunakan teknik koding yang telah dipelajari!
- Presentasikan hasilnya untuk mendapatkan umpan balik!

4. Menguji dan Mengevaluasi Solusi (*Testing and Evaluation*)

Setelah kamu selesai membuat aplikasi atau sistem berbasis Kecerdasan Artifisial (KA), kamu jangan langsung merasa puas dulu, ya! Sama halnya ketika kamu membuat kue, meskipun bentuknya sudah bagus, tetapi rasanya belum tentu enak. Nah, begitu juga dengan solusi teknologi, harus diuji dan dievaluasi terlebih dahulu untuk memastikan bahwa aplikasi tersebut benar-benar bekerja sesuai dengan harapan dan tidak ada kesalahan yang fatal.

Pengujian (*testing*) dan evaluasi (*evaluation*) adalah tahap penting dalam proses rekayasa. Dari tahapan pengujian ini dapat diketahui apakah solusi yang dibuat tersebut layak digunakan, perlu diperbaiki, atau bahkan harus dirombak total. Menurut Sommerville (2016), pengujian adalah proses sistematis untuk menemukan kesalahan dalam perangkat lunak, sedangkan evaluasi bertujuan untuk menilai kualitas dan kesesuaian solusi dengan kebutuhan pengguna.

Ada beberapa jenis pengujian yang dapat kamu lakukan, yaitu:

a. *Unit Testing*

Jenis pengujian ini digunakan untuk menguji bagian terkecil dari program (misalnya, satu fungsi atau modul) secara terpisah. Tujuan pengujian ini adalah untuk memastikan bahwa unit bekerja sesuai dengan spesifikasi dan bebas dari kesalahan logika. Kamu dapat menggunakan *white-box testing* (*glass-box testing*) untuk melakukannya. Teknik ini berfokus pada logika internal dari unit program dan digunakan untuk memverifikasi semua jalur logika yang memungkinkan. Teknik spesifik yang disarankan, di antaranya *basis path testing*, *condition testing*, *loop testing*, dan *data flow testing*.

b. *Integration Testing*

Pengujian ini digunakan untuk menguji bagaimana bagian-bagian program bekerja saat digabungkan. Jangan sampai ada modul yang saling “tidak nyambung”. Fokuslah pada pemeriksaan kesalahan dalam komunikasi data dan kontrol antarmodul. Kamu dapat menggunakan teknik berikut untuk pengujian tersebut.

- 1) *Top-Down Integration*, yaitu teknik untuk menguji modul utama lebih dahulu dan secara bertahap menambahkan modul bawah.
- 2) *Bottom-Up Integration*, yaitu teknik yang dimulai dari modul paling bawah lalu ke modul yang paling atas.
- 3) *Big-Bang Integration*, yaitu teknik yang menggabungkan semua modul sekaligus (berisiko tinggi jika *error* muncul).
- 4) *Sandwich Testing* merupakan gabungan dari pendekatan *top-down* dan *bottom-up*.

c. *Validation Testing*

Pengujian ini digunakan untuk mengonfirmasi bahwa perangkat lunak sesuai dengan kebutuhan dan harapan pengguna. Pengujian ini berfokus pada “Apakah kita membangun produk yang benar?” sesuai spesifikasi kebutuhan pengguna. Nah, untuk melakukan pengujian ini, kamu dapat menggunakan beberapa teknik berikut.



- 1) *Black-Box Testing*, yaitu teknik yang berfokus pada *input* dan *output* tanpa melihat isi kode program. Jenis teknik yang ada, di antaranya *equivalence partitioning*, *boundary value analysis*, *state-based testing*, dan *error guessing*.
- 2) *User Acceptance Testing (UAT)*, yaitu pengujian akhir oleh pengguna untuk memverifikasi bahwa perangkat lunak siap untuk digunakan.

d. *System Testing*

Pengujian ini dilakukan untuk menguji keseluruhan sistem sebagai satu kesatuan. Apakah semua fitur berjalan sesuai desain? Tujuannya adalah untuk memastikan bahwa perangkat lunak bekerja secara utuh dalam berbagai skenario. Kamu dapat menggunakan beberapa teknik berikut untuk melakukan pengujian jenis ini.

- 1) *Functional Testing*, yaitu teknik yang digunakan untuk menguji apakah fitur bekerja sesuai spesifikasi.
- 2) *Performance Testing*, yaitu teknik yang digunakan untuk menguji kecepatan, stabilitas, dan respons aplikasi, termasuk *stress testing*, *load testing*, dan *volume testing*.
- 3) *Security Testing*, yaitu teknik yang digunakan untuk memverifikasi kerentanan sistem.
- 4) *Recovery Testing*, yaitu teknik yang digunakan untuk menguji kemampuan sistem pulih dari *crash* atau gangguan.
- 5) *Compatibility Testing*, yaitu teknik yang berjalan pada berbagai OS, *browser*, atau perangkat.

e. *Acceptance Testing*

Pada pengujian ini pengguna terlebih dahulu akan diminta untuk mencoba aplikasinya, lalu mengumpulkan umpan balik. Hal ini penting karena pengguna biasanya memiliki perspektif berbeda dari pembuatnya. Pengujian ini berfungsi untuk memastikan apakah pengguna puas dengan fungsi dan kinerja sistem?



Gambar 6.9 Calon Pengguna Menguji Perangkat Lunak Sebelum digunakan

Kamu bisa menggunakan *alpha testing* (biasanya dilakukan oleh *user* di lingkungan pengembang) atau *beta testing* (biasanya dilakukan oleh *user* di lingkungan asli (lapangan)). Pengujian ini biasanya mengandalkan skenario penggunaan nyata, survei, dan umpan balik langsung.

Pressman dan Maxim (2020) juga menekankan bahwa pengujian bukan merupakan proses yang hanya dilakukan satu kali, melainkan proses iteratif dan berlapis, yaitu:

- a. Mulai dari *unit testing* oleh *programmer*;
- b. Dilanjutkan dengan *integration testing* oleh tim teknis.;
- c. Lalu *system testing* oleh QA; lalu
- d. Terakhir *acceptance testing* oleh pengguna akhir.

5. Menyempurnakan dan Iterasi (*Improvement*)

Perangkat lunak yang baik tidak berhenti di versi pertama. Harus ada penyempurnaan berdasarkan evaluasi, misalnya:

- a. Menambahkan fitur baru;
- b. Memperbaiki *bug*; atau
- c. Menyesuaikan tampilan antarmuka.

Dalam metode pengembangan modern seperti Agile atau Design Thinking, iterasi dilakukan secara terus-menerus agar solusi makin relevan (Purnama dan Hidayat, 2022).



Aktivitas 6.7

Menyempurnakan dan Merencanakan Pengembangan Lanjutan (Aktivitas Individu)

Dari hasil pengerjaan Aktivitas 6.6, ada umpan balik yang kamu dapatkan. Berdasarkan umpan balik tersebut, lakukanlah proses perbaikan. Setelah diperbaiki, buatlah rencana pengembangan aplikasi berikutnya. Misalnya, fitur apa yang akan ditambahkan, platform apa yang digunakan, dan sebagainya.

C. Mengembangkan Model Kecerdasan Artifisial

Kamu sudah belajar tentang cara memecahkan masalah, begitu juga dengan melakukan proses rekayasa. Pada materi sebelumnya, kamu juga sudah belajar tentang pembelajaran mesin dan bahasa alami. Sekarang, waktunya kamu mempelajari tentang cara membangun model untuk mengimplementasikannya.

1. Proses Pengembangan Model

Ternyata, Kecerdasan Artifisial (KA) kini sudah menjadi bagian dari kehidupan sehari-hari. Mungkin kamu tidak menyadarinya, tetapi rekomendasi film di Netflix, filter-filter lucu yang sering digunakan di Instagram, hingga mobil yang dapat menyetir sendiri adalah contoh dari penggunaan KA. Pernahkah kamu berpikir bagaimana cara membuat “otak” dari semua sistem cerdas ini? Nah, pada bagian ini, kamu akan belajar tentang proses penting yang berkaitan dengan hal tersebut, yaitu “Pengembangan Model”.

Coba deh kamu bayangkan, KA ini bagaikan seorang siswa yang sedang belajar sepertimu. Tujuan kamu belajar tentu agar kamu pintar, bukan? Nah, sama halnya dengan KA, perangkat ini pun perlu diajari terus-menerus supaya lebih pintar dalam mengenali pola, membuat prediksi yang akurat, bahkan mengambil keputusan. Proses belajar dan pelatihan inilah yang biasanya disebut sebagai “Pengembangan Model”.

Bagaimana sih cara mengajari KA? Caranya mirip seperti seorang kakak yang ingin mengajari adik kecilnya untuk membedakan buah apel dan buah jeruk. Untuk lebih jelasnya, perhatikan langkah-langkah berikut.

- a. Langkah awal dilakukan dengan menunjukkan berbagai contoh. Misalnya:

“Ini apel. Warnanya ada yang merah dan ada juga yang hijau. Bentuknya seperti ini, bulat dan halus.”

“Kalau yang ini jeruk. Warnanya oranye. Bentuknya juga sama, bulat, tapi ada teksturnya.”

Pada KA, langkah seperti ini merupakan bagian dari proses pengumpulan data.

- b. Langkah selanjutnya adalah mengoreksi adik ketika dia salah menentukan jenis buah yang tepat. Misalnya:

“Bukan, ini adalah jeruk.”

“Betul, yang ini apel.”

Pada KA, proses ini seperti melakukan pelatihan model untuk KA.

c. Langkah berikutnya, yaitu tes kemampuan. Misalnya:

“Kalau ini namanya buah apa?”

“Coba, yang ini buah apa?”

Pada KA, proses ini seperti melakukan evaluasi model untuk KA.



Gambar 6.10 Belajar Mengenali Apel dan Jeruk

Kurang lebih, pengembangan model KA itu mirip seperti analogi tersebut, namun bedanya pada KA yang diajari adalah komputer. Jadi, bukan hanya bisa membedakan mana jeruk dan apel, komputer bisa diajari untuk mengenal banyak sekali gambar, suara, dan bahkan menganalisis teks dalam jumlah yang begitu besar. Keren, kan?

Pengembangan model itu sangat penting, *lho!* Bayangkan saja, jika modelnya tidak dilatih dengan baik, maka bisa saja aplikasi pengenalan wajah dapat salah mengenali orang. Contoh lain adalah jika model tidak dilatih dengan model yang akurat, maka aplikasi prediksi cuaca bisa salah memperkirakan hujan. Kamu yang sudah berharap hari ini akan cerah, eh ternyata malah kebasahan karena hujan. Jadi, jangan disepelekan ya, proses ini!

Terdapat berbagai metode yang biasanya digunakan oleh para ahli untuk mengembangkan model KA, beberapa di antaranya adalah CRISP-DM (*Cross-Industry Standard Process for Data Mining*), KDD (*Knowledge Discovery in Databases*), dan OSEMN (*Obtain, Scrub, Explore, Model, and interpret*). Anggap saja semua metode ini seperti resep masakan yang berbeda-beda, tetapi memiliki tujuan yang sama, yaitu ingin membuat masakan yang enak. Begitupun halnya dengan pengembangan KA, terdapat berbagai metode, ada yang langkah-langkahnya sedikit lebih detail dan ada juga yang langkahnya lebih sederhana. Namun, metode-metode tersebut memiliki tujuan yang sama, yaitu untuk membuat model KA yang bagus.

Nah, pada bagian ini, kamu akan belajar menggunakan pendekatan yang lebih sederhana dan tentunya lebih mudah untuk dimengerti, sehingga cocok untuk kamu yang baru mulai belajar. Kira-kira, sebagai berikut tahapannya.

Tabel 6.4 Tahapan Pengembangan KA

Tahapan	Output yang Diharapkan
<i>Task</i>	Tujuan model, contoh: “memprediksi nilai ujian”.
<i>Dataset</i>	Data mentah, misalnya: data nilai dan absensi.
<i>Data Preparation</i>	<i>Dataset</i> siap pakai.
<i>Training Model</i>	Model awal yang bisa digunakan.
<i>Testing Model</i>	Evaluasi awal, misalnya: akurasi, eror, dan sebagainya.
<i>Evaluation</i>	Nilai evaluasi, misalnya: MAE dan akurasi.
<i>Tuning/Iterasi</i>	Model final.

Mari kita bahas satu per satu mengenai tahapan tersebut.

a. Menentukan Misi (*Task*)

Pada tahapan ini, kamu harus menentukan maksud pembuatan model KA terlebih dahulu. Sebagai contoh, pada tabel di atas tertulis bahwa model KA dibuat untuk memprediksi nilai ujian. Kamu juga dapat menentukan maksud pembuatan KA sesuai dengan kepentinganmu. Mungkin kamu ingin membuat model KA untuk memprediksi apa yang

akan dimasak oleh ibu besok berdasarkan menu masakan yang biasa dibuat ibu selama dua bulan ini. Menarik, kan?

Output yang diharapkan adalah tujuan model yang jelas. Misalnya, jika merujuk pada tabel, tujuannya, yaitu: “Model KA untuk memprediksi siswa yang berpotensi mendapatkan nilai di atas 80 pada ujian Matematika.” atau jika merujuk pada model KA masakan ibu, maka tujuannya adalah “Model KA untuk memprediksi masakan ibu saat makan malam keluarga”.

b. Mengumpulkan Semua Data (*Dataset*)

Masih ingatkah kamu bahwa model KA pun perlu belajar? Nah, KA itu belajar melalui data. Tentu saja data yang kamu gunakan tersebut harus relevan dengan misi/*task* yang sudah ditentukan tadi. Tidak mungkin kamu ingin membuat model KA untuk memprediksi nilai ujian, tetapi kamu menggunakan *dataset* skor bola. Semakin bagus kualitas data dan semakin banyak datanya, maka biasanya model KA yang dibuat pun akan semakin pintar. Misalnya, untuk memprediksi nilai ujian, maka data yang dipakai bisa berupa catatan nilai-nilai tugas, nilai kuis, persentase kehadiran siswa, dan keterangan lulus KKM atau tidak pada beberapa nilai sebelumnya.

Luaran (*output*) yang diharapkan adalah data mentah. Isi dari data mentah bisa bermacam-macam, bisa dari tabel di Excel atau kumpulan teks dan gambar bahkan suara. Dikatakan data mentah karena data-data yang diperoleh tersebut baru sekadar dikumpulkan saja dan belum jelas struktur datanya.

c. Membersihkan dan Merapikan (*Data Preparation*)

Data mentah itu bagaikan bahan masakan yang dibeli dari pasar. Bentuknya bermacam-macam dan ada yang harus dipotong dahulu, kemudian dicuci, lalu dibuang yang tidak perlunya, setelah itu baru dimasak. Sama halnya dengan data mentah, di mana pada tahap ini diperlukan proses membersihkan data, merapikan data, dan mengubah data menjadi format yang bisa “dimakan” oleh model KA. Kamu bisa bayangkan berapa lama waktu yang dibutuhkan pada tahap ini.

Terkadang, kamu akan menemukan data kosong, seperti nilai siswa yang tidak terisi. Dari sinilah kamu harus menentukan apakah data kosong tersebut mau dihapus atau diisi dengan nilai rata-rata? contoh lainnya yang sering ditemui pada data mentah adalah format tanggal yang berbeda-beda. Ada yang menggunakan format tanggal seperti ini: “13 Juli 2004”, ada juga yang menggunakan format berikut: “13/07/24”. Data-data yang tidak rapi ini-lah yang harus dibersihkan dan dirapikan terlebih dahulu. Sehingga, *output* dari tahapan ini diharapkan akan menjadi data yang rapi dan bersih serta siap digunakan untuk melatih model KA.

d. **Saatnya KA Belajar (*Training Model*)**

Jika data sudah rapi dan bersih, maka saatnya untuk mengajari model KA. Dalam hal ini, KA akan belajar untuk mencari pola-pola dari data yang kamu berikan. Caranya adalah dengan memilih algoritma atau metode belajar tertentu yang cocok untuk misi KA. Contoh: misinya adalah untuk klasifikasi. Maka, metode yang dapat digunakan adalah algoritma *Decision Tree*, *Naive Bayes*, atau *Support Vector Machine*. Contoh misi lainnya, yaitu prediksi angka. Maka, algoritma yang dapat digunakan adalah *Linear Regression*. Banyaknya algoritma/metode yang digunakan sama halnya seperti banyaknya cara yang kamu gunakan untuk belajar. Untuk pembelajaran yang sifatnya hafalan, kamu harus banyak membaca buku dan menghafal kata-kata penting. Untuk pembelajaran yang sifatnya bergerak, seperti pelajaran olahraga, kamu harus sering berlatih lari di tempat atau melatih kecepatan.

Untuk melatih KA, diperlukan alat bantu, karena dalam hal ini yang dilatih adalah komputer, sehingga harus menggunakan alat bantu untuk komunikasi dengan komputer dan untuk mengajari KA. *Tools* atau alat yang populer adalah Google Collaboratory atau Google Colab. Ada juga *tools* lain, seperti Jupyter Notebook, Spyder, Anaconda, bahkan perangkat lunak khusus *data science*, seperti RapidMiner atau KNIME. Namun, pembahasan pada buku ini akan fokus pada Google Collab yang cocok untuk digunakan oleh kamu yang baru belajar. *Library* yang paling umum digunakan adalah Scikit-learn, Pandas, dan NumPy.

Namun, masih banyak *library* lain yang bisa kamu gunakan sesuai dengan kebutuhan model yang akan kamu buat.

Sebagai informasi tambahan, biasanya sebelum mulai melatih model, data sudah dipisah menjadi dua bagian, yaitu sebagian besar data digunakan untuk latihan (*training data*) dan sebagian lagi digunakan untuk diuji (*testing data*). Luaran dari tahapan ini adalah model awal KA yang sudah belajar dari data. Model ini sudah dapat kamu gunakan untuk melakukan prediksi sederhana.

e. **Menguji Hasil Belajar (*Testing Model*)**

Sebagaimana halnya ketika kamu sudah belajar, kemudian kamu diuji dengan kuis untuk mengetahui sejauh mana pengetahuanmu. Begitu pula dengan KA yang harus diuji terlebih dahulu untuk mengetahui seberapa pintar KA tersebut setelah belajar. Cara menguji model KA dilakukan dengan memberikan data yang benar-benar baru yang tidak pernah dipakai sama sekali untuk belajar maupun latihan (data yang belum pernah disebutkan sebelumnya). Proses pengujian ini disebut dengan *testing data*. Sama halnya dengan kamu, apabila dites dengan soal-soal yang sama persis seperti yang pernah ada pada latihan, maka pasti nilai kamu akan bagus. Namun, belum tentu kamu benar-benar paham dengan materinya.

Luaran tahapan ini adalah evaluasi awal dari performa model KA. Dari hasil proses ini, kamu dapat melihat berapa banyak yang berhasil ditebak secara benar oleh model. Selain itu, kamu juga dapat melihat tingkat akurasi model (seberapa sering tebakannya benar) dan tingkat *error* (seberapa sering tebakannya salah).

f. **Memberi Rapor ke Model (*Evaluation*)**

Kini saatnya mengukur seberapa baik performa model secara lebih detail dengan menggunakan beberapa metrik tertentu. Jadi, metrik yang digunakan ini seperti halnya nilai-nilai di rapormu untuk menunjukkan kemampuanmu dalam berbagai aspek. Jenis metrik yang digunakan harus disesuaikan dengan misi model. Jika misimu adalah membuat klasifikasi untuk memprediksi kelulusan, maka kamu dapat menggunakan akurasi (persentase tebakan benar), presisi (hasil akurasi yang di atas KKM), *recall* (jumlah siswa yang berhasil ditebak di atas KKM), atau *F1-score* (gabungan presisi dan *recall*).

Selain itu, ada juga metrik lain yang dapat digunakan untuk memprediksi angka, yaitu MAE (*Mean Absolute Error*) yang dapat digunakan untuk mengukur rata-rata selisih antara tebakan model dengan nilai sebetulnya atau ada juga MSE (*Mean Squared Error*). Semakin kecil MAE atau MSE-nya, maka semakin bagus juga modelnya.

Sebagai contoh, sebuah model yang sudah dibuat memiliki akurasi 70%, hal ini berarti bahwa dari 100 siswa yang ada di data ujian, terdapat 70 siswa yang berhasil ditebak lulus oleh model. Akan tetapi, kamu juga harus mengecek apakah model KA tersebut benar-benar memiliki tebakan yang tepat ataukah justru tebakannya meleset, sehingga siswa yang seharusnya lulus malah tidak tertebak. Nah, itulah fungsi dari metrik-metrik yang telah dijelaskan di atas. Dengan menggunakan metrik tersebut, kamu akan mendapatkan nilai-nilai evaluasi untuk model, seperti akurasi = 70%, presisi = 70%, *recall* = 80%, dan MAE = 6.5.

g. Dipoles Lagi agar Semakin Keren (*Tuning/Iterasi*)

Jarang sekali sebuah model KA yang telah dibuat akan langsung sempurna pada percobaan pertama. Oleh karena itu, perlu dilakukan pemolesan lagi agar performanya semakin oke. Proses ini dilakukan berulang-ulang (*iterasi*) sebanyak yang kamu mau, sampai kamu puas dengan hasilnya. Dengan dilakukannya *iterasi*, kamu bisa mencoba untuk mengubah parameter model atau yang biasa disebut juga dengan *hyperparameter tuning*. Seperti halnya sebuah masakan, setiap algoritma pun memiliki ‘racikan’ sendiri-sendiri. Mungkin kamu pernah memasak nasi goreng, lalu masakanmu itu ternyata kurang kecap, kurang asin, kebanyakan telur, dan sebagainya. Maka, tentunya kamu akan menambahkan/mengurangi bumbu hingga hasilnya memuaskan. Nah, begitu pula dengan model KA yang kamu buat, pasti akan memerlukan pemolesan berkali-kali dalam proses penyempurnaannya.

Apabila algoritma yang kamu gunakan masih kurang memuaskan, maka kamu dapat mencoba untuk menggunakan algoritma lain yang lebih cocok. Selain itu, bisa dilakukan juga dengan menambah data karena terkadang model KA bisa menjadi kurang pintar ketika datanya kurang banyak atau kurang beragam. Kamu juga dapat melakukan rekayasa fitur atau yang biasa disebut dengan *feature engineering*. Hal ini dapat dilakukan dengan menambahkan fitur atau informasi baru

dari data yang sudah ada. Misalnya, pada fitur tanggal lahir, kamu dapat menambahkan fitur baru, yaitu “usia”.

Sebagai contoh, jika model KA yang kamu gunakan sekarang memprediksi hasil evaluasi dengan akurasi sebesar 43%, maka tentu saja hasil tersebut sangat kurang. Nah, kamu dapat mencoba untuk mengganti algoritmanya atau dapat mengatur-atur ulang parameter algoritma tersebut. Jika tidak, kamu dapat menambahkan lebih banyak lagi data dari data siswa pada tahun-tahun sebelumnya untuk dilatih, diuji, dan dievaluasi lagi. Begitu terus sampai hasilnya memuaskan, sehingga luarannya nanti dapat berupa mode final dengan performa yang sudah paling optimal serta sudah siap untuk digunakan sesuai misi awal.

2. *Library* untuk Pengembangan

Kita tadi sudah membahas mengenai pembuatan KA. Apakah kamu tahu bahwa dalam pembuatan KA, kamu tidak perlu membuat semuanya dari nol. Sama halnya seperti ketika kamu ingin membuat poster keren untuk acara sekolah atau presentasi di kelas. Pasti kamu lebih suka memilih untuk menggunakan templat yang sudah ada, sehingga kamu hanya tinggal mengubahnya sesuai dengan kemauanmu saja. Nah, di dunia koding juga terdapat konsep serupa yang dinamakan dengan *library*. Konsep ini pernah dibahas sepintas pada pembahasan sebelumnya. Kode-kode dalam *library* sudah dirancang, dioptimalkan, dan diuji oleh para ahli serta komunitas pengembang dari seluruh dunia, jadi kode-kode tersebut sudah terbukti andal dan efisien.

Jika kamu memperhatikan tahapan pengembangan model di atas, maka tahapan tersebut akan jauh lebih efisien apabila dilakukan dengan menggunakan *library*. Pada *library*, terdapat tahapan untuk mengolah dan menyiapkan data, melatih model KA, mengevaluasi performa model, dan masih banyak lagi tahapan lainnya. Setelah kamu mengenal secara umum tentang *library*, kini saatnya kamu mengenal ragam *library* andal yang sering digunakan di dunia.

a. Skicit-learn

Library yang satu ini dirancang khusus agar mudah digunakan, sehingga cocok sekali untuk pemula. *Library* jenis ini sangat lihai dalam membuat model-model KA yang menjadi dasar penting, seperti klasifikasi, regresi, dan sebagainya. Apalagi Skicit-learn ini biasanya sudah terpasang dan siap pakai di Google Colab dengan bahasa pemrograman Python. Contoh cara memanggil *library* ini adalah sebagai berikut.

```
from sklearn.linear_model import LinearRegression
```

Kode singkat di atas menunjukkan algoritma *Linear Regression* yang siap dipakai menggunakan *library* Skicit-learn

b. TensorFlow

Library yang dikembangkan oleh para ahli di Google ini siap untuk membantumu mengerjakan misi-misi KA yang lebih sulit/kompleks dan menantang. TensorFlow menjadi salah satu *library* andalan utama di dunia untuk membuat model-model KA yang sangat rumit, seperti *Deep Learning* (KA yang mampu belajar dari data yang sangat besar dan kompleks). *Library* ini juga sering digunakan dalam pengenalan gambar, pemrosesan bahasa alami (seperti pada chatbot), dan pengenalan suara (asisten virtual di HP). Contoh cara memanggil *library* ini adalah sebagai berikut.

```
import tensorflow as tf
```

Cara membacanya adalah aku mau menggunakan (*import*) *library* tensorflow, lalu memanggilnya dengan nama singkat 'tf'. Biasanya, banyak penggunaan panggilan lebih singkat agar lebih mudah digunakan. Ada pula jenis *library* lain yang umum digunakan, di antaranya dapat kamu pelajari pada tabel berikut.

Tabel 6.5 Jenis *Library* Lain yang Umum Digunakan

Library	Fungsi Utama	Catatan
PyTorch	alternatif TensorFlow	Banyak dipakai di dunia riset KA.
Pandas	untuk manipulasi data tabel	Sering digunakan bersama Scikit-learn.
Numpy	untuk perhitungan matematika vektor	<i>Library</i> dasar yang mendukung perhitungan KA.
Matplotlib/Seaborn	visualisasi data dan hasil model	Membantu pembuatan grafik dan <i>chart</i> .



Aktivitas 6.8

Mengeksplorasi *Library* Python untuk Kecerdasan Artifisial (Aktivitas Individu)

Pada aktivitas ini kamu akan mencoba eksplorasi *library*. Jangan lupa siapkan Google Colabnya terlebih dahulu, ya! Jika sudah, ikuti petunjuk berikut.

- Silakan tik perintah berikut!

Kode Program 6.1

```
01 import sklearn
02 import pandas as pd
03 import numpy as np
04
```

- Jika tidak muncul *error*, artinya *library* sudah dapat digunakan. Selanjutnya, tambahkan eksplorasi kecil: menampilkan versi dari *library*.

Kode Program 6.2

```
05 print(sklearn.__version__)
06 print(pd.__version__)
```

- Bagaimana hasilnya? Versi berapa *library* yang kamu gunakan?

3. Studi Kasus Sederhana

Sejauh ini kamu sudah mengenal berbagai tahapan, *tools* yang dipakai, serta *library* yang digunakan dalam pengembangan KA. Kini, kamu akan melakukan pengamatan terhadap beberapa model KA sederhana. Seperti halnya ketika kamu belajar teori memasak dan membaca buku resep, maka kini saatnya untukmu masuk ke dapur dan melihat cara pembuatannya. Dalam studi kasus sederhana, kamu akan berjumpa dengan tiga jenis masalah umum yang akan sering kamu temui dalam dunia KA.

a. Klasifikasi

Seperti halnya seorang petugas sortir, KA juga bertugas untuk mengamati suatu objek atau data, lalu memutuskan objek tersebut akan dimasukkan ke dalam kategori yang mana. Misalnya, “Data yang ini cocoknya dimasukkan ke dalam kategori A” atau “Seharusnya data yang seperti ini dimasukkan ke kelompok B”, dan sebagainya.

b. Prediksi

Dalam hal ini, KA menjadi semacam ‘peramal angka’ berdasarkan data-data sebelumnya. Nanti dari data-data tersebut, KA akan memperkirakan nilai numerik yang akan datang berikutnya.

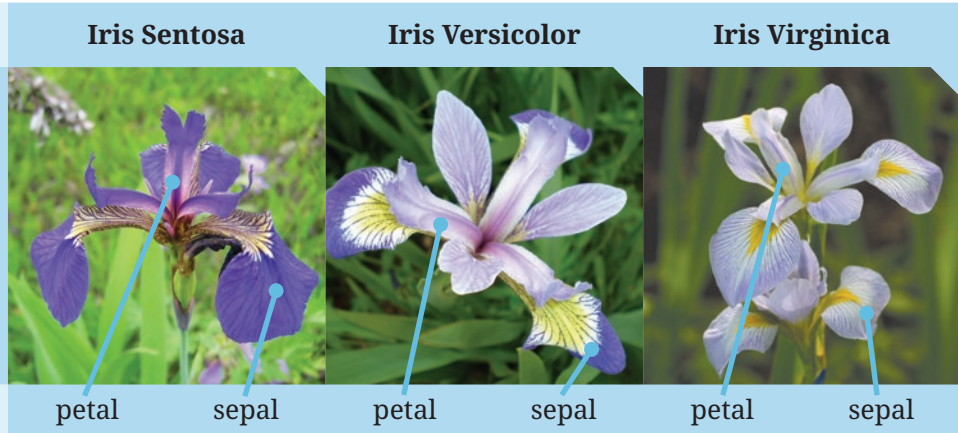
c. Klasterisasi (Pengelompokan)

Berbeda dengan klasifikasi, dalam klasterisasi KA, kategori tidak diberi tahu di awal. Tugasnya adalah mencari pola dan mengelompokkan data secara otomatis berdasarkan kemiripan karakteristik. Bayangkan KA sebagai pengamat yang melihat sekumpulan benda, lalu mengelompokkannya berdasarkan bentuk atau warna, tanpa mengetahui nama kelompok tersebut sebelumnya. Contoh penerapannya adalah mengelompokkan pelanggan berdasarkan kebiasaan belanja untuk keperluan pemasaran.

Dalam contoh kasus berikut, kamu hanya akan menggunakan dua jenis masalah umum, yaitu klasifikasi dan prediksi. Klasterisasi akan dibahas pada jenjang berikutnya. Semua kode program pada kasus berikut dilakukan menggunakan Google Colab dengan bahasa pemrograman Python. Sebagai permulaan, kamu akan melihat metode-metode yang paling dasar, seperti *Regresi Linear* untuk memprediksi nilai dan *Decision Tree* atau Regresi Logistik untuk melakukan klasifikasi.

Studi Kasus 1: Menebak Jenis Bunga Iris

Pada studi kasus ini, kamu memiliki misi untuk membantu seorang botanikus pemula. Dia kebingungan dalam membedakan tiga jenis bunga iris yang apabila dilihat sekilas terlihat sangat mirip. Apakah kamu bisa membantu botanikus ini dengan menggunakan KA?



Gambar 6.11 Perbedaan Bunga Iris

Sumber: 1) w.ru:Денис Анисимов (2005) 2) Jefficus (2007) 3) Eric Hunt (2018)

Misi utama yang harus kamu lakukan adalah membuat model KA yang bisa menentukan (mengklasifikasikan) jenis bunga iris dengan melihat ukuran panjang dan lebar kelopak (petal) serta daun pelindung (sepal) dari bunga tersebut. *Dataset* iris merupakan *dataset* yang akan digunakan. Data ini berisi 150 bunga iris yang masing-masing terdiri atas 50 bunga untuk setiap jenis. Nah, *dataset* ini sudah tersedia langsung di dalam *library* Skicit-learn. Sehingga, kamu tidak perlu repot-repot untuk mencarinya lagi.

Apabila misi sudah ada, data sudah siap, apa yang harus dilakukan selanjutnya? Yang harus dilakukan selanjutnya adalah memanggil data atau *load data* mengenai *dataset* bunga iris dari Skicit-learn. Lalu, pisahkan data menjadi data *training* (mengambil dari sebagian besar data yang sudah dimiliki) dan data *testing*.

Sebagaimana tahapan yang sudah kamu pelajari sebelumnya, sekarang saatnya untuk menggunakan model klasifikasi, misalnya kamu mau menggunakan *DecisionTreeClassifier*. Algoritma ini bekerja seperti membuat diagram alir untuk mengambil keputusan. Jangan lupa untuk mengerjakan

evaluasi model karena tahap penting untuk melihat seberapa banyak tebakan jenis bunga yang benar (akurasi).

Kode Program 6.3

```
01 from sklearn.datasets import load_iris
02 from sklearn.model_selection import train_test_split
03 from sklearn.tree import DecisionTreeClassifier
04 from sklearn.metrics import accuracy_score
05
06 # Memuat dataset Iris
07 iris = load_iris()
08 X = iris.data
09 y = iris.target
10
11 # Membagi data menjadi data latih dan data uji
12 X_train, X_test, y_train, y_test = train_test_split(X, y,
13 .....: test_size=0.3, random_state=42)
14
15 # Membuat model Decision Tree
16 model_bunga = DecisionTreeClassifier(random_state=42)
17
18 # Melatih model
19 model_bunga.fit(X_train, y_train)
20
21 # Melakukan prediksi
22 prediksi_jenis_bunga = model_bunga.predict(X_test)
23
24 # Menghitung akurasi
25 akurasi = accuracy_score(y_test, prediksi_jenis_bunga)
26 print("Akurasi model kita dalam menebak jenis bunga "
27 .....: "Iris adalah:", akurasi)
```

Berdasarkan kode tersebut, diharapkan dapat mencetak angka skor akurasi (biasanya antara 0 dan 1, di mana 1 memiliki arti sempurna) untuk menunjukkan seberapa baik model yang dibuat dalam mengenali bunga iris. Kira-kira, berdasarkan kode tersebut, berapa ya, hasil akurasinya?

Studi Kasus 2: Memprediksi Nilai Siswa dari Jam Belajar

Bayangkan, bagaimana seandainya kamu bisa memperkirakan nilai ujian kamu sendiri berdasarkan berapa lama kamu belajar? Ya, pada studi kasus ini, misi utamanya adalah membuat model KA yang bisa memperkirakan nilai ujian siswa (dari skala 0–100) berdasarkan jumlah jam belajar mereka. Untuk *dataset* yang digunakan adalah *dataset* manual, di mana data yang digunakan berupa data sederhana saja yang hanya terdiri dari kolom ‘jam_belajar’ dan ‘nilai_ujian’. *Library* yang akan digunakan adalah Pandas untuk membuat dan mengelola data dalam bentuk tabel.

Lalu, bagaimana cara membangun modelnya? Pertama, kamu tikkan data jam belajar dan nilai ujian tersebut. Kemudian kamu gunakan *LinearRegression* dari Skicit-learn untuk mencari hubungan garis lurus antara jam belajar dan nilai. Selanjutnya, kamu ‘ajari’ model dengan data yang sudah dibuat tadi. Lalu, coba tebak nilai untuk jumlah jam belajar yang baru.

Kode Program 6.4

```
01 import pandas as pd
02 from sklearn.linear_model import LinearRegression
03
04 # Data jam belajar dan nilai ujian
05 data_nilai = {
06     'jam_belajar': [1, 2, 3, 4, 3, 6, 5, 4],
07     'nilai_ujian': [50, 58, 65, 70, 78, 82, 88, 92]
08 }
09
10 # Membuat DataFrame
11 dataset_siswa = pd.DataFrame(data_nilai)
12
13 # Menyiapkan variabel fitur dan target
14 X_jam = dataset_siswa[['jam_belajar']]
15 y_nilai = dataset_siswa['nilai_ujian']
16
17 # Membuat dan melatih model regresi linier
18 model_nilai = LinearRegression()
19 model_nilai.fit(X_jam, y_nilai)
```

```

20
21 # Prediksi nilai ujian untuk jam belajar tertentu
22 jam_belajar_baru = [[2]]
23 prediksi_nilai_baru =
24 model_nilai.predict(jam_belajar_baru)
25
26 # Menampilkan hasil
27 print("Dataset yang kita gunakan:")
28 print(dataset_siswa)
29
30 print("\nPrediksi nilai untuk siswa yang belajar",
31       jam_belajar_baru[0][0],
32       "jam adalah sekitar:", prediksi_nilai_baru[0])
33
34 print("Koefisien (slope):", model_nilai.coef_[0])
35 print("Intersep:", model_nilai.intercept_)

```

Kode tersebut akan menampilkan *dataset* yang telah kamu buat dan hasil prediksinya selama 2 jam. Selain itu, kamu juga dapat melihat koefisien (kemiringan) garis dan intersep dari model regresinya. Selain menggunakan Regresi Linear, kamu juga dapat menggunakan jenis regresi lain, ya.



Aktivitas 6.9

Membangun Model KA Sederhana dari Studi Kasus (Aktivitas Individu)

Untuk mengerjakan aktivitas ini, silakan kamu ikuti langkah-langkah berikut.

- a. Implementasikan studi kasus 1 dan studi kasus 2!
- b. Dari studi kasus 1 dan 2, apa hasil yang kamu dapatkan?
 - 1) Apakah sesuai dengan hasil yang diharapkan?
 - 2) Bagaimana jika kamu mengganti beberapa angka di dalamnya, apa yang akan terjadi?
- c. Coba kamu buat model KA sendiri (bisa mengacu pada studi kasus 1 atau 2) secara sederhana saja!

4. Evaluasi Model

Masih ingatkah kamu dengan evaluasi model? Seperti yang telah dibahas sebelumnya bahwa evaluasi sangat penting untuk melihat seberapa akurat model KA yang telah dibuat? Bayangkan saja, saat diuji, model KA mungkin bisa menebak 90% benar. Akan tetapi, setelah dievaluasi ternyata ada kesalahan dalam menebak? Maka dari itulah, evaluasi ini penting agar kamu dapat mengetahui apakah model yang dibuat tersebut benar-benar cerdas atautkah hanya “beruntung” saja dalam menebak.

Beberapa jenis model sudah kamu pelajari sebelumnya, sekarang saatnya kamu mempelajari lebih dalam beberapa jenis model tersebut dan metrik evaluasi mana yang cocok untuk mengevaluasi KA yang telah dibuat. Berikut ini beberapa model evaluasi berdasarkan jenis modelnya.

a. Model Klasifikasi

Seperti pada studi kasus menebak jenis bunga iris, model klasifikasi ini bertugas untuk mengelompokkan data ke dalam kategori tertentu. Nah, untuk proses evaluasinya kamu dapat menggunakan metrik berikut.

Tabel 6.6 Metrik yang Digunakan pada Model Klasifikasi

Metrik	Penjelasan Sederhana
Akurasi	Persentase prediksi yang benar dari semua prediksi.
<i>Precision</i>	Dari semua yang diprediksi “positif”, beberapa yang benar-benar positif.
<i>Recall</i>	Dari semua data “positif”, beberapa yang berhasil ditemukan.
<i>F1 Score</i>	Rata-rata harmonis antara <i>precision</i> dan <i>recall</i> .
<i>Confusion Matrix</i>	Tabel 2x2 yang menunjukkan jumlah benar dan salah untuk setiap kategori.

Pada pembahasan ini, jenis model klasifikasi akan lebih fokus pada penggunaan akurasi dan *confusion matrix* untuk menilai model. Tetapi, jika kamu ingin menggunakan semuanya, maka akan lebih baik karena kamu bisa melihat ‘nilai rapor’ secara lebih luas lagi. Bagaimana cara untuk menuliskannya ke dalam kode di Python? Perhatikan contoh berikut.

Kode Program 6.5

```
01 from sklearn.metrics import accuracy_score, confusion_matrix
02
03 # Menghitung akurasi model
04 akurasi_model = accuracy_score(y_test, prediksi)
05 print("Akurasi Model Kita:", akurasi_model)
06
07 # Menghitung confusion matrix
08 matriks_kebingungan = confusion_matrix(y_test, prediksi)
09 print("Confusion Matrix Model Kita:\n", matriks_kebingungan)
```

b. Model Regresi (Prediksi)

Masih ingatkah kamu dengan Studi Kasus 2 mengenai prediksi nilai ujian berdasarkan jam belajar? Kira-kira untuk evaluasinya sama atau berbeda? Dalam kasus tersebut, model prediksi atau regresi bertugas untuk menebak nilai, maka metrik evaluasinya juga berbeda. Coba kamu perhatikan tabel berikut.

Tabel 6.7 Metrik yang Digunakan pada Model Regresi

Metrik	Penjelasan Sederhana
<i>MAE (Mean Absolute Error)</i>	Rata-rata selisih absolut antara hasil prediksi dan nilai sebenarnya.
<i>MSE (Mean Squared Error)</i>	Rata-rata selisih kuadrat (lebih peka terhadap kesalahan besar).
<i>RMSE</i>	Akar dari MSE (memberikan satuan yang sama seperti datanya).
<i>R² Score</i>	Seberapa baik model menjelaskan variasi data (skor antara 0 dan 1).

Ingatlah, semakin kecil nilai MAE, maka hasilnya akan lebih baik. Bagaimana cara untuk menuliskannya ke dalam kode di Python? Kamu dapat melihat contohnya pada kode program berikut dengan pemisalan kamu sudah mempunyai `y_test` (nilai sebenarnya dari data ujian) dan `prediksi` (hasil tebakan nilai dari model).

Kode Program 6.6

```
01 from sklearn.metrics import mean_absolute_error
02 from sklearn.metrics import mean_squared_error, r2_score
03
04 # Menghitung Mean Absolute Error (MAE)
05 mae_model = mean_absolute_error(y_test, prediksi)
06 print("MAE Model Kita:", mae_model)
07
08 # Menghitung Mean Squared Error (MSE)
09 mse_model = mean_squared_error(y_test, prediksi)
10 print("MSE Model Kita:", mse_model)
11
12 # Menghitung R2 Score
13 r2_model = r2_score(y_test, prediksi)
14 print("R2 Score Model Kita:", r2_model)
```

Terkadang, angka saja kurang untuk menceritakan hasil evaluasi, visualisasi juga dibutuhkan supaya lebih mudah memahami performa model. Pasti kamu juga merasa pusing ketika melihat angka-angka yang banyak. Belum lagi, jika kamu tidak mengetahui maksud di balik angka-angka tersebut. Maka dari itu, terciptalah beberapa *library* pada Python yang dapat membantumu untuk melihat hasil evaluasi menjadi lebih visual dan bergambar. Beberapa alat bantu evaluasi tersebut, di antaranya:

Tabel 6.8 Alat Bantu Evaluasi

Tools	Fungsi
Matplotlib	Membuat grafik akurasi, <i>error</i> , dan sebagainya.
Seaborn	Membuat <i>confusion matrix</i> dalam bentuk <i>heatmap</i> .
Pandas	Analisis tabular sederhana untuk <i>error/perbandingan</i> .

Selain *tools* pada tabel di atas, masih ada lagi beberapa *tools* lainnya yang lebih canggih untuk membantu tahap visualisasi dan evaluasi. Misalnya, ada TensorBoard atau Weka. Kedua *tools* ini biasa digunakan pada pemodelan *deep learning*.



Aktivitas 6.10

Mengevaluasi Model KA dan Visualisasi Hasilnya (Aktivitas Individu)

Kini kamu telah memahami lebih dalam tentang evaluasi model. Untuk menguji pemahamanmu tersebut, silakan kerjakan aktivitas berikut.

- Jalankan salah satu model yang menurutmu mudah untuk dilakukan!
- Coba hitung akurasinya (jika itu klasifikasi) atau MAE-nya (jika itu prediksi)!
- Buatkan visualisasinya!
- Silakan kamu ubah data yang ada di dalamnya dan lihat apakah hasil evaluasinya berubah!

5. Pengenalan Model Bahasa Besar (LLM)

Pernahkah kamu melihat KA yang dapat menulis cerita, mengarang puisi, menjawab pertanyaan sulit dengan lancar, atau bahkan mungkin kamu pernah mencoba berbincang dengan KA sebagaimana kamu mengobrol dengan manusia? Di balik kerennya itu semua, ternyata ada sebuah teknologi super canggih yang bernama *Large Language Model* atau LLM. Agar kamu memahami lebih dalam tentang LLM dan cara kerjanya, perhatikan penjelasan berikut.

LLM adalah program komputer yang super pintar, tujuannya adalah untuk memahami, menghasilkan, dan bekerja dengan bahasa manusia. Coba kamu bayangkan, untuk belajar saja, LLM membaca jutaan buku, miliaran artikel berita, percakapan internet, dan berbagai macam teks lainnya. Jika diibaratkan, LLM bisa membaca semua buku pelajaranmu dari kelas 1 SD sampai nanti kuliah untuk belajar, bahkan itupun masih terhitung bacaan yang sangat sedikit bagi LLM. Bukankah itu sangat keren?

Mungkin otakmu akan merasa pusing dan kewalahan jika disuruh untuk membaca sebanyak itu. Lalu, untuk apa LLM belajar banyak sekali? Berikut di antaranya beberapa alasan LLM mempelajari hal-hal tersebut.

- Makna kata dipelajari oleh LLM untuk mengetahui arti berbagai kata dari bacaan yang dibacanya.

- b. Tata bahasa digunakan LLM untuk memahami cara menyusun kalimat yang benar seperti layaknya manusia.
- c. Gaya bahasa dipelajari LLM untuk mengenali apakah sebuah tulisan itu formal, bercanda, santai, puitis, dan sebagainya.
- d. Konteks digunakan LLM untuk memahami apa maksud dari percakapan atau tulisan. Sehingga, LLM tidak hanya bisa membaca atau mengenali gaya bahasanya saja.

Kamu mungkin sering mendengar atau mungkin mencoba sendiri contoh LLM, seperti ChatGPT dari OpenAI atau layanan KA dari Google, seperti Gemini. Bagaimana kesan yang kamu peroleh ketika menggunakan KA tersebut? Apakah terasa seperti mengobrol dengan teman sendiri? LLM merupakan bagian penting dari cabang KA, yaitu Pemrosesan Bahasa Alami atau *Natural Language Processing* (NLP). Fokus utamanya adalah agar komputer dapat berinteraksi dan mengerti bahasa yang digunakan sehari-hari tanpa terkesan terlalu ‘robotik’.

Lalu, bagaimana cara kerja LLM? Pasti kamu penasaran dengan cara kerja otak pintar satu ini? Keberhasilan LLM terletak pada desain mesin yang disebut dengan *transformer*. Nah, arsitektur *transformer* ini seperti resep khusus yang digunakan agar LLM dapat melakukan hal-hal berikut.

a. Memperhatikan Konteks

Terkadang manusia saja masih suka merasa kebingungan ketika mencoba untuk memahami maksud dari kalimat yang dibaca, apalagi maksud soal pada kuis. LLM ini mencoba untuk memahami makna konteks secara akurat (tidak hanya melihat kata satu per satu).

b. Memprediksi Kata Berikutnya

Keren sekali, kan LLM ini? Selain bisa memahami maksud dari kalimat yang dibaca, LLM juga bisa menebak kata apa yang akan muncul selanjutnya. Rasanya seperti berbincang dengan teman yang punya satu pemikiran denganmu. Proses ‘menebak kata selanjutnya’ ini dilihat dari kata apa yang paling mungkin muncul selanjutnya dalam sebuah kalimat atau paragraf.



Arsitektur *transformer* ini pertama kalinya diperkenalkan dalam karya ilmiah yang berjudul “*Attention is All You Need*” oleh para peneliti di Google pada tahun 2017 dan sejak saat itulah konsep tersebut menjadi fondasi bagi hampir semua LLM canggih yang ada saat ini.

Jika dipikir-pikir, LLM ini seperti manusia ya, tetapi dengan kecerdasan yang tinggi. Bagaimana tidak, kemampuan LLM ini sangat luar biasa. Berikut pemaparan mengenai beberapa kemampuan yang dimiliki oleh LLM.

Tabel 6.9 Kemampuan LLM

Kemampuan	Contoh
Menjawab Pertanyaan	“Siapa presiden pertama Indonesia?” → “Ir. Soekarno”
Menulis Teks	“Tolong buat puisi tentang matahari.”
Menerjemahkan Bahasa	Menerjemahkan dari bahasa Inggris ke Indonesia.
Meringkas Artikel	Mengubah teks panjang menjadi lebih pendek.
Membantu <i>Coding</i>	“Tolong buat fungsi Python untuk menghitung luas segitiga.”

Bahkan beberapa LLM yang lebih canggih dari itu dapat melakukan hal-hal lain di luar teks. Misalnya, kamu bisa meminta KA untuk membuat gambar dengan deskripsi: “Buatkan gambar tentang sekolah jika berada di bawah laut!” atau LLM juga bisa mewakili idemu dalam membuat musik sederhana yang akan digunakan untuk tampil di panggung sekolah. Kemampuan LLM dalam ‘menciptakan’ sesuatu yang baru seperti ini, sering disebut sebagai KA Generatif.

Selain ChatGPT dan Gemini, masih banyak lagi model LLM yang mungkin baru kamu ketahui karena LLM ini dikembangkan oleh berbagai perusahaan dan organisasi riset. Agar kamu lebih mengenal model LLM tersebut, perhatikan tabel berikut.

Tabel 6.10 Model LLM

Nama Model	Pengembang Utama	Catatan Singkat
ChatGPT (GPT-3.5, GPT-4, dst.)	OpenAI	Sangat populer, bisa diakses melalui <i>web</i> , pandai mengobrol, menulis, dan sebagainya.
Gemini	Google DeepMind	Model canggih dari Google yang digunakan di berbagai produk Google dan memiliki kemampuan multimodal (teks, gambar, kode).
LLaMA & Llama Family	Meta (Facebook)	Model kuat yang juga tersedia untuk penelitian. Terdapat versi yang lebih “ringan” dan bisa dijalankan di perangkat sendiri (dengan syarat tertentu).
Claude	Anthropic	Fokus pada KA yang lebih aman dan membantu. Dikenal karena kemampuannya dalam percakapan panjang dan analisis.
BERT	Google	Salah satu model <i>transformer awal</i> yang revolusioner dan sangat berpengaruh dalam pemahaman bahasa (lebih ke fondasi daripada produk <i>chat</i> langsung).

Dunia LLM berkembang sangat cepat. Sehingga, bisa saja ketika kamu membaca buku ini, mungkin akan muncul model LLM terbaru yang tidak kalah kerennya dengan model LLM yang sedang dibahas saat ini.

Adakah di antara kalian yang suka menggunakan LLM untuk pelajaran sekolah? Seperti halnya ketika kamu menggunakan teknologi canggih yang lain, kamu juga harus bijak, bertanggung jawab, dan beretika ketika menggunakan LLM. Sebagai siswa, ada beberapa prinsip penting yang perlu kamu pegang ketika menggunakan LLM agar bisa menjadi teman diskusi yang berguna dan bukan sebagai alat penghancur bagi diri sendiri. Simaklah cara menggunakan LLM secara bijak berikut.

a. Gunakan LLM bukan untuk menyontek apalagi plagiat.

LLM memang bisa membantumu menjelaskan konsep atau memberimu ide. Tetapi, ingatlah, cukup gunakan LLM hanya untuk menjelaskan konsep dan memberi ide saja. Jangan sampai kamu menggunakannya untuk mengerjakan ujian atau tugas secara langsung sebagai hasil karyamu sendiri. Jika kamu melakukan hal tersebut, itu berarti kamu tidak jujur dan melanggar etika. Bukan hanya itu, hal tersebut juga akan membuatmu menjadi malas berpikir ketika ujian. Jadi, jika ada materi yang kurang kamu pahami, kamu bisa berdiskusi dengan KA, namun sebelum menuliskannya, kamu harus mencerna terlebih dahulu materi tersebut hingga kamu memahaminya.

b. Bersikaplah kritis terhadap jawaban.

“Tidak ada yang sempurna.”, slogan tersebut cocok untuk LLM. Sekerenkerennya LLM, tetap saja masih perlu belajar. Sehingga, bisa saja jawabannya salah, tidak akurat, bias, atau tidak lengkap. Jangan lupa untuk mengecek ulang informasi penting yang terdapat pada sumber lain yang terpercaya (Jangan langsung menerima informasi yang diberikan KA secara mentah-mentah).

c. Gunakan LLM sebagai alat bantu belajar, bukan sebagai pengganti proses belajar.

Prinsip ini hampir mirip dengan poin pertama. LLM dapat kamu jadikan sebagai teman diskusi dan teman belajar karena LLM dapat mempercepat pemahaman, mencari penjelasan alternatif, bahkan memberikan inspirasi. Akan tetapi, jangan sampai LLM membuatmu menjadi malas untuk berpikir. Berdiskusi dengan guru dan teman juga sangat diperlukan dalam proses belajarmu.

d. Jagalah privasi.

Prinsip ini merupakan hal sangat penting. Berhati-hatilah ketika memasukkan informasi pribadi atau rahasia ke dalam LLM. Mengapa begitu? Karena bisa jadi, informasi yang kamu simpan, dapat digunakan untuk pengembangan lebih lanjut (tetapi hal ini juga tergantung pada kebijakan masing-masing dari penyedia layanan).

Berdasarkan penjelasan dari prinsip-prinsip tersebut, semoga kamu bisa lebih bijak lagi dalam menggunakan LLM. Ingatlah bahwa LLM hanya dapat dijadikan sebagai teman yang membantu proses belajar, bukan sebagai pengganti otak manusia!



Aktivitas 6.11

Eksplorasi Langsung LLM (*Large Language Model*) (Aktivitas Individu dan Kelompok)

Pada Aktivitas 6.11 ini, kamu akan mengeksplorasi langsung LLM. Ikuti petunjuk berikut untuk pengerjaannya.

- a. Buka salah satu LLM (Kamu bisa memilih Gemini, ChatGPT, atau layanan LLM lainnya yang mau kamu coba!)
- b. Coba ajukan beberapa pertanyaan menarik, seperti “Apa saja fakta menarik tentang planet Venus?” atau “Berikan aku tiga ide yang jarang terpikirkan orang untuk mengurangi sampah plastik di sekolah!”
- c. Kemudian, kamu amati dan bandingkan jawaban-jawaban tersebut!
 - 1) Apakah jawabannya lebih mudah dipahami?
 - 2) Apakah lebih akurat?
 - 3) Jika kamu mencoba lebih dari satu LLM, apakah jawabannya mirip atau malah berbeda?
- d. Diskusikan hasil kerjamu dengan teman!
 - 1) Apakah temanmu mendapatkan jawaban yang sama?
 - 2) Bagaimana pengalamannya ketika menggunakan LLM?
 - 3) Hal menarik apa saja yang kamu temukan?
- e. Jangan lupa untuk mengeksplorasi semua hal menarik lainnya dan tetaplah ingat pada prinsip-prinsip yang telah kita bahas.



Uji Kompetensi

A. Analisis Kasus

Amati ilustrasi kasus berikut dengan saksama, lalu jawablah pertanyaan di bawahnya!

Seorang siswa membuat model untuk memprediksi nilai matematika berdasarkan jumlah jam belajar. Namun, setelah diuji, model tersebut memberikan hasil yang sangat tidak akurat.

Pertanyaan:

1. Apa saja kemungkinan penyebab model tersebut tidak akurat?
2. Sebutkan minimal dua langkah yang bisa dilakukan siswa untuk memperbaiki model tersebut!

B. Pilihan Ganda

Pilihlah salah satu jawaban yang paling tepat.

1. Pernyataan yang paling tepat tentang hubungan rekayasa dan model KA adalah . . .
 - a. Model KA menggantikan semua tahapan proses rekayasa.
 - b. Proses rekayasa tidak diperlukan dalam pembuatan model KA.
 - c. Model KA adalah bagian dari solusi yang dikembangkan melalui proses rekayasa.
 - d. Rekayasa hanya dibutuhkan untuk perangkat keras.
 - e. Model KA digunakan hanya setelah proses rekayasa selesai.
2. Akurasi model *DecisionTree* hanya 50%. Langkah pertama yang tepat adalah
 - a. mengganti algoritma ke *Random Forest*
 - b. menaikkan kompleksitas model tanpa validasi
 - c. memeriksa apakah data sudah dibersihkan dan relevan
 - d. menghapus data yang tidak sesuai
 - e. menambah data tanpa mengubah parameter

C. Proyek Mini

Buatlah desain aplikasi prediksi sisa makanan di kantin sekolah. Lalu, jelaskan hal-hal berikut!

1. Masalah yang diselesaikan
2. Data yang digunakan
3. Algoritma KA
4. Evaluasi model

D. Soal Uraian

1. Komparasi Metode Rekayasa
Bandingkan metode *Waterfall* dan *Agile* untuk proyek siswa. Manakah metode yang lebih sesuai? Jelaskan alasannya!
2. Refleksi Etika LLM
Apa risiko yang akan didapatkan jika siswa langsung menyalin jawaban dari ChatGPT?
3. Komunikasi dan Ketepatan Bahasa
Sebutkan dua alasan mengapa kita tetap harus berpikir kritis saat menggunakan LLM!

Rubrik Penilaian

Aspek yang Dinilai	Skor Maksimal
Analisis Kasus	20
Pilihan Ganda HOTS	10
Proyek Mini	25
Komparasi Metode Rekayasa	15
Refleksi Etika LLM	10
Komunikasi dan Ketepatan Bahasa	20
Total	100

Kriteria Skor Tambahan:

- Sangat Baik (100–85): Semua aspek dijawab lengkap, analisis dalam, dan solusi kreatif.
- Baik (84–70): Sebagian besar aspek dijawab tepat, ada beberapa kekurangan minor.

- Cukup (69–55): Jawaban masih umum, kurang mendalam, atau belum sepenuhnya sesuai konteks.
- Kurang (<55): Jawaban tidak mencerminkan pemahaman atau tidak lengkap.

Pengayaan

Untuk menambah pengetahuanmu mengenai materi pada Bab 6, kamu dapat berlatih dengan memindai kode QR atau pranala berikut.



<https://buku.kemdikbud.go.id/s/KKA11-SuplemenBab6>

Kamu juga bisa mencari tahu lebih banyak lagi materi serupa melalui beberapa media berikut.

1. Eksplorasi *Dataset* Publik

Kamu bisa mengeksplorasi *dataset* publik untuk latihan latihan, seperti pada *link* berikut.

- *Kaggle datasets*: <https://www.kaggle.com/datasets>
- *UCI Machine Learning Repository*: <https://archive.ics.uci.edu/ml/index.php>
- *Google Dataset Search*: <https://datasetsearch.research.google.com>

2. Algoritma dan Model KA

Pada tahap *training model*, kamu bisa mengeksplorasi lebih banyak lagi algoritma populer lainnya, seperti *K-Nearest Neighbors (KNN)*, *Random Forest*, dan masih banyak lagi melalui *link* berikut: <https://www.geeksforgeeks.org/ai-algorithms/>

3. Evaluasi dan *Tuning* Model

Untuk penjelasan mengenai tahap evaluasi, *tuning*, dan masih banyak lagi, bisa diakses pada *link* berikut: https://scikit-learn.org/stable/model_selection.html

4. *Library* Pemrograman dan KA

Yuk, pelajari lagi lebih dalam mengenai *library* melalui *link-link* berikut.

- Scikit-learn: <https://scikit-learn.org/>
- TensorFlow: <https://www.tensorflow.org/>
- PyTorch: <https://docs.pytorch.org/tutorials/>
- Panduan Modul Python: <https://docs.python.org/3/tutorial/modules.html>

5. Visualisasi dan Grafik

Kamu bisa mempelajari bagaimana cara memvisualisasikan data melalui *link* berikut.

- Matplotlib: <https://matplotlib.org/>
- Seaborn: <https://seaborn.pydata.org/>

6. LLM dan *Transformer*

Agar kamu lebih memahami materi tentang LLM, bacalah artikel yang berjudul “*Attention is All You Need (Transformer Paper)*” dengan mengakses *link* berikut: <https://arxiv.org/pdf/1706.03762>

7. Dokumentasi Teknis Tambahan

Buat kamu yang masih penasaran dan ingin belajar lebih dalam tentang dokumentasi teknis tambahan, kamu bisa mengunjungi *link* berikut.

- `train_test_split` – Scikit-learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- `DecisionTreeClassifier` – Scikit-learn: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>



Refleksi

Petunjuk:

Isilah refleksi berikut dengan jujur berdasarkan pengalamanmu selama mengikuti pembelajaran pada bab ini. Tidak ada jawaban benar atau salah, yang penting adalah proses berpikirmu sendiri.

1. Apa pemahaman barumu tentang proses rekayasa dalam pengembangan model Kecerdasan Artifisial (KA)?

Jelaskan bagian mana yang sebelumnya belum kamu ketahui dan kini kamu pahami lebih baik!

2. Menurutmu, mengapa proses seperti evaluasi model tidak boleh dilewatkan dalam pembuatan sistem berbasis KA? Apa dampak yang mungkin terjadi jika langkah ini diabaikan?

3. Bagaimana kamu bisa menggunakan pendekatan rekayasa dan KA untuk menyelesaikan masalah di sekolah atau lingkungan sekitarmu?

Berikan contoh nyata dari kehidupanmu!

4. Apa tantangan utama yang kamu rasakan saat mencoba memahami atau menerapkan model KA dan bagaimana kamu mengatasinya?

5. Setelah mengenal LLM seperti ChatGPT, apa pendapatmu tentang peran KA dalam pendidikan? Bagaimana cara menggunakan teknologi ini secara bijak dan bertanggung jawab?

6. Apakah kamu merasa bahwa proses berpikir komputasional (dekomposisi, abstraksi, dan sebagainya) membantumu dalam belajar? Jelaskan alasannya!

7. Jika diberi kesempatan untuk mengembangkan sebuah model KA, untuk masalah apa kamu ingin membuatnya? Jelaskan alasannya!

Glosarium

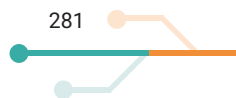
abstraksi	: Proses menyederhanakan permasalahan dengan menonjolkan hal-hal penting dan mengabaikan detail yang tidak relevan.
<i>acceptance testing</i> (uji penerimaan)	: Pengujian akhir oleh pengguna untuk memastikan sistem sesuai dengan kebutuhan.
<i>accuracy</i> (akurasi)	: Ukuran ketepatan hasil prediksi model terhadap data sebenarnya.
adaptasi keterampilan	: Upaya meningkatkan keterampilan untuk mempersiapkan diri terhadap perubahan.
adversarial	: Teknik menipu model KA dengan inputan data yang “membodohi”.
algoritma	: Langkah-langkah logis yang disusun untuk menyelesaikan suatu masalah.
analisis kebutuhan (<i>requirement analysis</i>)	: Proses memahami kebutuhan pengguna dan batasan sistem yang akan dibuat.
<i>Artificial Intelligence</i> (Kecerdasan Artifisial/ KA)	: Teknologi yang membuat sistem komputer mampu belajar, menalar, dan mengambil keputusan seperti manusia.
atribut	: Data atau variabel yang dimiliki oleh objek atau kelas untuk menyimpan informasi tentang karakteristiknya.
audio	: Suara yang direkam atau diproduksi untuk mendukung konten digital, misalnya <i>podcast</i> atau narasi video.
<i>big-bang integration</i>	: Cara menguji sistem dengan langsung menggabungkan semua bagiannya sekaligus.
<i>black-box testing</i>	: Pengujian perangkat lunak berdasarkan input dan <i>output</i> tanpa melihat isi kode di dalamnya.
<i>caption</i>	: Keterangan singkat yang menyertai konten digital, biasanya pada unggahan media sosial.
<i>class</i> (kelas)	: Rancangan atau cetak biru yang digunakan untuk membuat objek; berisi atribut dan metode.
<i>clustering</i> (klasterisasi)	: Proses mengelompokkan data tanpa label berdasarkan kesamaan karakteristik.
<i>coding</i> (koding)	: Kegiatan menulis instruksi dalam bahasa pemrograman untuk membuat aplikasi.
<i>computational thinking</i> (berpikir komputasional)	: Cara berpikir logis dan sistematis untuk memecahkan masalah menggunakan prinsip ilmu komputer.
<i>constructor</i> (<i>init</i>)	: Metode khusus dalam Python yang dijalankan secara otomatis ketika objek dibuat.

<i>continuous</i>	: Data numerik yang bisa mengambil nilai apa pun dalam rentang tertentu (biasanya bilangan desimal).
<i>Creative Commons (CC)</i>	: Lisensi terbuka yang memperbolehkan penggunaan karya dengan hak tertentu sesuai ketentuan.
CRISP-DM	: Model proses standar dalam pembuatan sistem berbasis data dan pembelajaran mesin.
data kategorikal	: Data yang dapat menggambarkan kategori atau kelompok data.
data nominal	: Data kategorikal yang tidak memiliki urutan.
data numerik	: Data angka yang dapat dihitung/dioperasikan.
data ordinal	: Data kategorikal yang memiliki urutan atau hierarki.
<i>data preparation</i> (persiapan data)	: Tahap membersihkan dan menyiapkan data sebelum digunakan untuk melatih model KA.
<i>dataset</i> (kumpulan data)	: Kumpulan data hasil inputan sebuah sistem; Data yang digunakan untuk melatih dan menguji model kecerdasan artifisial.
<i>debugging</i>	: Kegiatan mencari dan memperbaiki kesalahan dalam kode program.
<i>decision tree</i> (pohon keputusan)	: Algoritma pembelajaran mesin berbentuk cabang-cabang keputusan untuk membuat klasifikasi atau prediksi.
<i>decoding</i>	: Proses mengembalikan data ke bentuk semula seperti sebelum proses <i>encoding</i> .
<i>decomposition</i> (dekomposisi)	: Cara memecah masalah besar menjadi bagian-bagian kecil agar lebih mudah diselesaikan.
<i>discrete</i> (diskrit):	: Data numerik yang hanya bisa mengambil nilai tertentu (biasanya bilangan bulat).
<i>design thinking</i>	: Pendekatan kreatif untuk menemukan solusi dengan berfokus pada kebutuhan manusia.
diseminasi	: Proses penyebarluasan konten digital kepada audiens melalui berbagai platform.
<i>editing</i> (penyuntingan)	: Proses memperbaiki atau menyempurnakan teks, gambar, audio, atau video agar lebih jelas, menarik, dan berkualitas.
empati	: Kemampuan manusia memahami perasaan, kondisi, dan konteks orang lain yang tidak dimiliki oleh KA.

<i>encoding</i>	: Proses mengubah data dari satu format ke format lain yang lebih mudah diproses oleh komputer.
<i>engineering process</i> (proses rekayasa)	: Langkah-langkah sistematis untuk merancang dan membangun solusi dari suatu masalah.
<i>Entity-Relationship Diagram</i> (ERD)	: Diagram yang menunjukkan hubungan antarentitas dalam suatu sistem data.
enkapsulasi	: Proses melindungi data dengan menyembunyikan detail implementasi agar hanya dapat diakses melalui metode tertentu.
etika digital	: Prinsip moral dan aturan perilaku yang harus dipatuhi dalam menggunakan teknologi dan membuat konten digital.
<i>evaluation</i> (evaluasi model)	: Tahap menilai kinerja model menggunakan ukuran tertentu seperti akurasi atau galat.
<i>feature engineering</i>	: Proses memilih atau membuat fitur data agar model bekerja lebih baik.
<i>flowchart</i> (diagram alur)	: Gambar yang menunjukkan urutan langkah atau logika proses dalam program.
<i>functional model</i> (model fungsional)	: Rancangan yang menggambarkan fungsi utama dari sistem yang akan dibuat.
<i>Generative AI</i> (GenAI)	: Jenis kecerdasan artifisial yang dapat menghasilkan konten baru, seperti teks, gambar, audio, dan video.
hoaks	: Informasi palsu atau menyesatkan yang disebarakan untuk memengaruhi opini publik.
<i>human centered KA</i>	: Pendekatan yang menempatkan manusia sebagai inti dari desain pengembangan dan implementasi Kecerdasan Artifisial.
<i>hyperparameter tuning</i>	: Penyesuaian nilai parameter algoritma untuk meningkatkan kinerja model.
infografik	: Visualisasi data atau informasi dalam bentuk gambar agar mudah dipahami.
<i>inheritance</i> (pewarisan)	: Mekanisme yang memungkinkan suatu kelas mewarisi atribut dan metode dari kelas lain.
inklusivitas	: Keterlibatan semua kalangan masyarakat termasuk kelompok rentan misalnya penyandang disabilitas dan lansia.
<i>integration testing</i> (uji integrasi)	: Pengujian untuk memastikan bagian-bagian sistem bekerja bersama dengan baik.
instansiasi	: Proses pembuatan objek dari sebuah kelas.
<i>iteration</i> (iterasi)	: Pengulangan proses untuk memperbaiki atau menyempurnakan hasil.
KA otonom	: Sistem KA yang beroperasi secara mandiri dengan sedikit intervensi dari manusia.

kampanye digital	: Upaya terencana untuk menyampaikan pesan atau isu sosial melalui konten digital.
Kecerdasan Artifisial (KA)	: Cabang ilmu komputer yang berkaitan dengan otomatisasi perilaku cerdas yang bertujuan untuk menciptakan sistem yang dapat meniru atau bahkan melampaui kemampuan kognitif manusia dalam menyelesaikan tugas tertentu; Sistem yang dapat meniru perilaku cerdas manusia dengan bantuan data dan algoritma.
klasifikasi (<i>classification</i>)	: Proses mengelompokkan data ke dalam kategori tertentu menggunakan model KA.
kelas abstrak	: Kelas yang tidak dapat dibuat objeknya secara langsung dan berfungsi sebagai kerangka bagi kelas turunannya.
ketimpangan sosial	: ketidaksetaraan ekonomi dan sosial
kolaborasi	: Kerja sama antara manusia dengan manusia lain atau dengan KA untuk mencapai tujuan bersama.
koneksi <i>database</i>	: Cara menghubungkan aplikasi dengan <i>database</i> menggunakan <i>library</i> tertentu.
konten digital	: Materi berupa teks, gambar, audio, atau video yang dibuat dan disebarluaskan melalui media digital.
<i>library</i> (pustaka)	: Kumpulan kode siap pakai yang membantu mempercepat pembuatan program.
<i>linear regression</i> (regresi linear)	: Metode prediksi yang mencari hubungan linier antara variabel masukan dan keluaran.
lisensi	: Izin hukum yang mengatur hak penggunaan suatu karya.
MAE (<i>Mean Absolute Error</i>)	: Ukuran rata-rata selisih absolut antara nilai prediksi dan nilai sebenarnya.
<i>machine learning</i> (pembelajaran mesin)	: Cabang dari kecerdasan artifisial yang membuat sistem mampu belajar dari data.
<i>manual control</i>	: Sistem tradisional yang bergantung pada manusia tanpa bantuan KA.
membersihkan data (<i>data cleaning</i>)	: menghapus data kosong, memperbaiki penulisan, dan membuang duplikat
metode	: Fungsi yang didefinisikan di dalam kelas untuk menggambarkan perilaku atau aksi objek.
metrik evaluasi	: Ukuran kuantitatif yang digunakan untuk menilai seberapa baik kinerja model.
model artisan	: Pola kolaborasi di mana KA mendukung profesional (seperti guru atau dokter) dengan data dan wawasan, sementara keputusan akhir tetap pada manusia.
<i>model factory</i>	: Pola kolaborasi di mana KA menangani tugas berulang dan massal, misalnya entri data atau pemantauan sistem.

model KA (model Kecerdasan Artifisial)	: Representasi matematis yang digunakan untuk mengenali pola dan membuat prediksi.
<i>mockup</i> (rancangan awal antarmuka)	: Contoh tampilan awal aplikasi sebelum dikembangkan lebih lanjut.
<i>modular programming</i> (pemrograman modular)	: Cara membuat program dengan memecahnya menjadi bagian-bagian kecil yang saling terhubung.
objek	: Entitas yang dibuat dari kelas, memiliki atribut dan metode tertentu.
optimasi	: Proses membuat suatu pekerjaan lebih efisien dan efektif, misalnya pengaturan rute logistik oleh KA.
<i>overfitting</i>	: Kondisi ketika model terlalu menyesuaikan diri dengan data latihan sehingga kurang baik saat diuji dengan data baru.
<i>overriding</i>	: Proses mengganti atau menulis ulang metode dari kelas induk di dalam kelas anak.
<i>pattern recognition</i> (pengenalan pola)	: Kemampuan sistem untuk mengenali kesamaan dan pola dalam data.
plagiarisme	: Tindakan menggunakan karya orang lain tanpa izin atau tanpa menyebutkan sumber.
<i>podcast</i>	: Konten audio digital yang dapat diunduh atau diputarkan melalui internet.
polimorfisme	: Kemampuan objek dari kelas yang berbeda untuk merespons metode yang sama dengan cara yang berbeda.
<i>private</i>	: Atribut atau metode yang tidak dapat diakses langsung dari luar kelas (dalam Python ditandai dengan dua garis bawah).
<i>protected</i>	: Atribut atau metode yang hanya dapat diakses oleh kelas itu sendiri dan kelas turunannya (ditandai dengan satu garis bawah).
<i>prompt</i>	: Perintah atau instruksi yang diberikan kepada KA untuk menghasilkan keluaran tertentu.
<i>prompt engineering</i> (rekayasa <i>prompt</i>)	: Teknik menyusun perintah agar KA memberikan hasil sesuai kebutuhan.
prototipe (<i>prototype</i>)	: Versi awal sistem atau aplikasi yang dibuat untuk diuji sebelum dikembangkan penuh.
<i>public</i>	: Atribut atau metode yang dapat diakses dari luar kelas tanpa batasan.
Python	: Bahasa pemrograman populer yang banyak digunakan dalam pengembangan aplikasi dan kecerdasan artifisial.
refleksi	: Kegiatan meninjau kembali pengalaman belajar untuk mengambil makna dan pembelajaran.
<i>requirement elicitation</i>	: Proses menggali dan mengumpulkan kebutuhan pengguna.



Scikit-learn	: Pustaka Python yang digunakan untuk pembelajaran mesin dengan cara yang sederhana.
serangan evasi	: memodifikasi input agar AI salah prediksi
serangan <i>linkage</i>	: Gabungan data "anonim" disandingkan dengan sumber publik sehingga identitas aslinya bisa diketahui.
serangan <i>poisoning</i>	: merusak data latihan AI agar hasilnya bias/menyesatkan
<i>software engineering</i> (rekayasa perangkat lunak)	: Proses merancang, membangun, dan memelihara perangkat lunak secara sistematis.
<i>subclass</i> (kelas anak)	: Kelas yang mewarisi atribut dan metode dari kelas induk.
<i>superclass</i> (kelas induk)	: Kelas yang menjadi sumber pewarisan bagi kelas lainnya.
<i>system testing</i> (uji sistem)	: Pengujian menyeluruh untuk memastikan sistem berjalan sesuai fungsinya.
TensorFlow	: Pustaka sumber terbuka dari Google untuk membuat model pembelajaran mesin dan <i>deep learning</i> .
<i>testing</i> (pengujian)	: Proses memastikan program atau sistem berjalan sesuai harapan tanpa kesalahan.
<i>training model</i> (pelatihan model)	: Tahap melatih model KA agar dapat mengenali pola dari data.
<i>user acceptance testing</i> (UAT)	: Pengujian oleh pengguna untuk memastikan sistem memenuhi kebutuhan mereka.
<i>validation testing</i> (uji validasi)	: Pengujian untuk memastikan sistem sesuai dengan spesifikasi yang telah ditetapkan.
<i>video editing</i> (penyuntingan video)	: Proses mengatur, memotong, atau menambahkan efek pada rekaman video agar lebih baik.
<i>waterfall</i>	: Model pengembangan perangkat lunak yang dilakukan secara berurutan, dari analisis hingga pemeliharaan.

Daftar Pustaka

- Adib Moghaddam, Arshin. *Is Artificial Intelligence Racist?: The Ethics of AI and the Future of Humanity*. London: Bloomsbury Publishing Plc, 2023.
- Blake, Annabel. *How to Be Good at Human Things in a Post-AI World*. Future Minds Lab, 2019. <https://www.futuremindslab.com/blog/2019/6/12/how-to-be-good-at-human-things-in-a-post-ai-world>.
- Brown, Tim, dan Jocelyn Wyatt. *Design Thinking for Education and Social Innovation*. Stanford Social Innovation Review, 2022. 8, 30–35. <https://doi.org/10.1596/1020797X.12.1.29>
- Bybee, Rodger W. *The Case for STEM Education: Challenges and Opportunities*. Arlington, VA: NSTA Press, 2013.
- Deitel, Paul J., dan Harvey M. Deitel. *Python for Programmers*. Edisi ke-2. Boston: Pearson, 2025.
- Dinata, E. R., dan M. R. Fikry. *Pemrograman dan Aplikasi Kecerdasan Buatan*. Bandung: Informatika, 2014.
- Dunia Ilkom. *PYTHON 14. Enkapsulasi & Constructor*. YouTube. Diakses 30 Juni 2025. <https://www.youtube.com/watch?v=IhwgorxcGic>.
- Eck, David J. *Introduction to Programming Using Python*. New York: Hobart and William Smith Colleges, 2019.
- Educative.io. *Polymorphism in Python*. Diakses 30 Juni 2025. <https://www.educative.io/answers/what-is-polymorphism-in-python>.
- Fauzi, Ahmad. *Rekayasa Perangkat Lunak untuk Pendidikan*. Yogyakarta: Deepublish, 2018.
- Geeks for Geeks. *Encapsulation in Python*. Diakses 30 Juni 2025. <https://www.geeksforgeeks.org/encapsulation-in-python/>.
- Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.)*. New York: O'Reilly Media, 2019.
- Goodfellow, Ian, Yoshua Bengio, dan Aaron Courville. *Deep Learning and AI Models for Real-World Applications*. Cambridge, MA: MIT Press, 2024.
- Google Arts & Culture. *Say What You See*. 2024. <https://artsandculture.google.com/experiment/say-what-you-see/jwG3m7wQShZngw?hl=en>
- Google DeepMind. *Gemini Model Overview*. 2024. <https://deepmind.google/technologies/gemini>
- Hidayat, R. *Pemrograman Python Berbasis OOP untuk SMK dan MA*. Jakarta: Bumi Aksara, 2023.

- IBM Newsroom. *IBM for AI Youth–Indonesia*. 2024. <https://asean.newsroom.ibm.com/indonesia>.
- Kai-Fu Lee. *AI Superpowers: China, Silicon Valley, and the New World Order*. Boston: Houghton Mifflin Harcourt, 2018.
- Kartikawati, L., Kusriani, K., & Luthfi, E. T. *Algoritma K-Means pada Pengelompokan Pembelajaran Tatap Muka Terbatas Sesudah Vaksinasi COVID-19*. 2022. *Jurnal Eksplorasi Informatika*, 11(1), 20–28. <https://doi.org/10.30864/eksplorasi.v11i1.560>
- Kelas Terbuka. *Belajar Python OOP #08–Encapsulation*. YouTube. Diakses 30 Juni 2025. <https://www.youtube.com/watch?v=NDjX-8kak2g>.
- Kelas Terbuka. *Inheritance Python | OOP Python*. YouTube. Diakses 30 Juni 2025. <https://www.youtube.com/watch?v=WIEYbAK4SBM>.
- Kemendikbud Ristek. *Modul Ajar Informatika SMA–Computational Thinking dan Kecerdasan Buatan*. Jakarta: BSKAP, 2024.
- Kurniawan, D. *Membuat Aplikasi CRUD Python + SQLite3 + Tkinter*. Jakarta: Elex Media Komputindo, 2021.
- Luger, George F. dan William A. Stubblefield. *Artificial Intelligence Structures and Strategies for Complex Problem Solving 3rd Edition*. California: Addison Wesley Longman Inc, 1997.
- LIPI. *AI dan Media Digital: Potensi, Risiko, dan Regulasi*. Jakarta: LIPI Press, 2021.
- Lutz, Mark. *Learning Python. Edisi ke-5*. Sebastopol, CA: O'Reilly Media, 2013.
- Magisto. *Smart Video Editor*. 2024. <https://www.magisto.com/>
- Marwondo, dan Rini Melati. *Dasar-Dasar Pengembangan Perangkat Lunak dan Gim untuk SMK/MAK Kelas X*. Jakarta: Pusat Perbukuan, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, 2023.
- McKinsey & Company. *The State of AI–How Organizations are Rewiring to Capture Value*. 2025. <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai>
- Meitjing, D., dan M. N. Fuad. *Berpikir Kritis dan Komputasional dalam Kurikulum Merdeka*. Bandung: Remaja Rosdakarya, 2023.
- Meta AI. *LLaMA 3: Open Foundation Model for Language Understanding*. 2024. <https://ai.meta.com/llama>
- Microsoft Learn. *Fundamentals of Machine Learning with Python*. 2023. <https://learn.microsoft.com/en-us/training/paths/ml-python>
- Munir, Rinaldi, dan Lidya. *Algoritma dan Pemrograman dalam Bahasa Python*. Bandung: Informatika, 2020.

- Nugroho, R., dan J. Kristiadi. *Literasi Digital dan Problematika Sosial di Era Teknologi*. 2021. Jakarta: Prenada Media.
- OpenAI. *Best Practices for Prompt Engineering*. 2024. <https://platform.openai.com/docs/guides/gpt-best-practices>.
- Pressman, Roger S., dan Bruce R. Maxim. *Software Engineering: A Practitioner's Approach*. 9th ed. New York: McGraw-Hill Education, 2020.
- Programiz. *Python Inheritance*. Diakses 30 Juni 2025. <https://www.programiz.com/python-programming/inheritance>.
- Programiz. *Python OOP Tutorial for Beginners*. YouTube. Diakses 30 Juni 2025. https://www.youtube.com/watch?v=Ej_02ICOIgs.
- Purnama, Irfan, dan Taufik Hidayat. **Agile dan Design Thinking dalam Pengembangan Produk Digital**. Bandung: Pustaka Teknologi, 2022.
- Real Python. *Object-Oriented Programming (OOP) in Python 3*. Diakses 30 Juni 2025. <https://realpython.com/python3-object-oriented-programming/>.
- Risman, Budi. *Kecerdasan Buatan dalam Dunia Pendidikan*. Jakarta: Literasi Media Nusantara, 2020.
- Russell, Stuart. *Human Compatible: Artificial Intelligence and the Problem of Control*. New York: Viking, 2019.
- Scikit-learn. *Model Selection*. 2023. https://scikit-learn.org/stable/model_selection.html
- Shneiderman, B. *Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy*. New York: Oxford University Press, 2020.
- Sihombing, P., Nababan, M., Sinaga, C. V., Napitupulu, I. W., Rahman, A., & -, S. *Application of the K-Means Clustering Method for Performance Assessment Based on the Competency of Educators*. *Jurnal Sistem Informasi Dan Ilmu Komputer*. Prima (JUSIKOM PRIMA), 2023. 7(1), 50–60. <https://doi.org/10.34012/jurnalsisteminformasidanilmukomputerv7i1.3951>
- Simplilearn. *AI for Beginners*. 2023. YouTube. <https://www.youtube.com/watch?v=5x9g1uyfUuk>.
- Sommerville, Ian. *Software Engineering*. 10th ed. Boston: Pearson, 2016.
- Suharyadi. *Pengantar Teknik dan Rekayasa*. Jakarta: Erlangga, 2017.
- Suyanto, S. *Machine Learning: Menggunakan Python dan Scikit-Learn*. Yogyakarta: Andi Publisher, 2018.
- Syahputra, Y. H., & Hutagalung, J. *Superior Class to Improve Student Achievement Using the K-Means Algorithm*. *Sinkron*, 2022. 7(3), 891–899. <https://doi.org/10.33395/sinkron.v7i3.11458>

- Tech with Tim. "Polymorphism in Python." YouTube. Diakses 30 Juni 2025. <https://www.youtube.com/watch?v=0b69nFqBrfE>.
- Tresnawati, A., dkk. n.d. *Berpikir Komputasional dalam Pendidikan Abad 21*. Jakarta: Pustekom, 2014.
- Topol, Eric. *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. New York: Basic Books, 2019.
- UNESCO. *Recommendation on the Ethics of Artificial Intelligence*. 2023. <https://en.unesco.org/artificial-intelligence/ethics>.
- Vaswani, Ashish, et al. *Attention Is All You Need*. 2017. <https://arxiv.org/abs/1706.03762>.
- Wahana Komputer. *Pemrograman Python Tingkat Lanjut*. Yogyakarta: Andi, 2021.
- World Economic Forum. *How AI Is Changing Our Work*. 2023. YouTube. <https://www.youtube.com/watch?v=fLz8yffLk1A>.
- World Bank. *AI in Education*. 2023. YouTube. <https://www.youtube.com/watch?v=J4XXUJb3gsk>.
- YouTube. *Prompt Engineering for Beginners*. 2024. <https://www.youtube.com/watch?v=80UXKxllOaQ>.
- YouTube. *How to Talk to AI*. 2024. <https://www.youtube.com/watch?v=1eK8dzQk0A0>.

Daftar Sumber Gambar

Gambar 6.11 diunduh dari <https://commons.wikimedia.org/wiki/File:Irissetosa1.jpg> pada 18 September 2025, <https://commons.wikimedia.org/wiki/File:IrisVersicolor-FoxRoost-Newfoundland.jpg> pada 18 September 2025, https://commons.wikimedia.org/wiki/File:Iris_virginica_2.jpg pada 18 September 2025

A

abstrak 74, 118, 120, 122, 123, 126, 280
abstraksi 85, 222, 228, 276, 277
adversarial 65, 277
AI engineer 29
akurasi 68, 250, 253, 254, 255, 260, 263, 264, 265, 277, 279
akurat 11, 21, 22, 35, 36, 68, 70, 83, 133, 160, 165, 189, 200, 208, 226, 248, 249, 263, 267, 270, 271, 272
alat bantu 5, 6, 41, 233, 234, 241, 252, 265, 270
algoritma 3, 28, 37, 53, 60, 62, 75, 77, 78, 142, 165, 196, 197, 199, 200, 201, 202, 203, 204, 205, 206, 207, 216, 217, 222, 228, 252, 254, 255, 256, 272, 274, 277, 279, 280
analisis data 11, 35, 41, 42, 141, 142, 153, 154, 160, 195
aplikasi 12, 13, 14, 18, 28, 30, 32, 35, 46, 53, 54, 55, 56, 57, 58, 69, 75, 86, 99, 127, 130, 131, 132, 144, 146, 152, 168, 170, 173, 174, 175, 176, 178, 184, 188, 189, 190, 191, 192, 198, 200, 205, 208, 209, 213, 214, 231, 232, 233, 234, 235, 236, 238, 240, 242, 243, 244, 246, 247, 249, 272, 277, 280, 281
Artificial Intelligence 23, 25, 31, 126, 277, 283, 284, 285, 286, 292, 295, 296
asosiasi 199
atribut 85, 86, 87, 88, 89, 90, 93, 94, 96, 97, 99, 100, 101, 103, 104, 105, 106, 107, 108, 109, 110, 115, 118, 121, 122, 124, 277, 279, 281, 282
audio 2, 14, 17, 277, 278, 279, 280, 281
autentikasi 56

B

bahasa alami 248, 256
bahasa pemrograman 77, 89, 94, 98, 175, 176, 242, 256, 258, 277
basis data 168
bias 28, 36, 53, 54, 59, 60, 63, 66, 68, 69, 270, 282
binary encoding 157, 158, 159, 190
bot 34

C

cara berpikir 75, 225, 228, 230, 233
cetak biru 86, 94, 118, 277
Chatbot 30, 34, 35, 43, 75, 208, 214, 231, 256
content moderation 53
CSV 131, 133, 141
customer service 10, 34, 42

D

dampak positif 34, 35, 37, 208
data 3, 4, 5, 6, 8, 9, 11, 18, 28, 29, 30, 31, 33, 34, 35, 36, 37, 41, 42, 43, 49, 53, 54, 56, 57, 59, 60, 63, 65, 66, 67, 68, 69, 70, 71, 74, 75, 79, 83, 85, 86, 87, 89, 94, 98, 99, 100, 103, 121, 124, 127, 129, 130, 131, 132, 133, 134, 135, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 151, 152, 153, 154, 157, 159, 160, 161, 162, 163, 164, 165, 166, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 180, 181, 182, 184, 185, 186, 187, 188, 189, 190, 191, 192, 195, 196, 197, 199, 200, 201, 203, 204, 206, 207, 213, 215, 217, 218, 222, 226, 228, 230, 231, 232, 234, 238, 240, 242, 243, 245, 248, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 263, 264, 266, 272, 275, 277, 278, 279, 280, 281, 282
database 99, 130, 131, 132, 142, 168, 169, 170, 173, 174, 175, 176, 177, 178, 179, 180, 183, 184, 188, 189, 192, 222, 280
data cleaning 137, 280
data scientist 29, 127
dataset 63, 64, 130, 131, 133, 134, 137, 138, 139, 140, 141, 142, 143, 146, 147, 151, 153, 154, 155, 161, 189, 197, 200, 206, 222, 251, 259, 260, 261, 262, 274, 278
data training 197, 199, 259
decision tree 200, 215, 278
decoding 142, 184, 278
deduksi 66
definisi 86, 87, 111
dekomposisi 222, 228, 276, 278
desain 1, 6, 10, 35, 52, 53, 54, 55, 58, 150, 238, 242, 246, 267, 272, 279
desainer 9, 10
desain KA 58
diseminasi 18, 20, 278
diskrit 278

E

e-commerce 29, 173, 174, 175, 200
efisien 3, 4, 5, 6, 11, 29, 35, 36, 60, 83, 131, 157, 168, 169, 170, 173, 200, 217, 228, 230, 233, 236, 243, 255, 281
encoding 130, 131, 132, 138, 139, 142, 143, 145, 146, 148, 149, 151, 152, 153, 154, 155, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 184, 188, 189, 190, 191, 192, 278, 279
engineering 217, 220, 221, 223, 233, 234, 235, 238, 254, 279, 281, 282

enkapsulasi 83, 98, 99, 100, 121, 122, 124, 279
eror 53, 119, 121, 131, 133, 175, 244, 250
etika 9, 15, 16, 24, 26, 28, 59, 60, 61, 65, 69, 70, 71, 228, 270, 279
evaluasi 18, 53, 60, 207, 235, 241, 244, 247, 249, 250, 253, 254, 255, 260, 263, 265, 266, 274, 276, 279, 280
evasi 282

F

fitur 7, 12, 13, 30, 46, 52, 53, 55, 56, 57, 58, 68, 76, 88, 92, 99, 100, 125, 171, 178, 188, 196, 201, 232, 234, 237, 246, 247, 254, 255, 261, 279
framework 52

G

game 31, 69, 127, 131
gdown 140, 141

H

hak cipta 15, 22
human-centered 28, 29, 52, 55, 56, 57, 70, 72

I

ide 10, 11, 12, 17, 34, 54, 58, 76, 84, 85, 225, 232, 235, 238, 270, 271
identifikasi 66, 72, 84, 94, 192, 217, 225, 227, 233
implementasi 52, 65, 98, 118, 120, 126, 200, 214, 215, 235, 238, 242, 279
informasi 6, 32, 37, 42, 52, 56, 59, 65, 66, 84, 88, 113, 114, 115, 132, 142, 152, 168, 169, 170, 171, 173, 175, 199, 213, 222, 234, 236, 253, 254, 270, 271, 277, 279
inklusivitas 279
inovasi 40, 200, 235
instansiasi 87, 89, 96, 97, 279
instruksi 31, 77, 195, 206, 215, 242, 277, 281
integrasi 130, 168, 173, 176, 279
iterasi 17, 247, 254, 279
iteratif 238, 243, 247

K

kategorikal 133, 134, 135, 138, 139, 142, 144, 148, 149, 160, 184, 188, 190, 278
Kecerdasan Artifisial , 1, 2, 3, 4, 5, 11, 16, 23, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37, 40, 41, 42, 43, 45, 46, 52, 55, 56, 59, 60, 61, 64, 65, 69, 72, 75, 129, 132, 133, 134, 137, 142, 144, 193, 196, 207, 217, 219, 220, 222, 223, 230, 233, 235, 244, 248, 257, 276, 277, 279, 280, 281, 292, 294

kelas 3, 20, 59, 66, 70, 73, 74, 83, 85, 86, 87, 89, 90, 92, 93, 94, 95, 96, 97, 98, 100, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 117, 118, 119, 120, 121, 122, 123, 124, 126, 136, 144, 150, 157, 169, 170, 178, 192, 206, 207, 208, 216, 243, 255, 266, 277, 279, 280, 281, 282

kelas abstrak 118, 120, 122, 123, 126, 280

kelas dasar 103, 106, 107, 118

kelas induk 103, 104, 105, 108, 109, 111, 112, 118, 281, 282

kelas turunan 103, 104, 105, 108, 109, 112, 118, 119, 120, 122, 123

klasifikasi 139, 190, 199, 201, 232, 252, 253, 256, 258, 259, 263, 266, 278, 280

klasterisasi 207, 258, 277

kode 45, 62, 75, 76, 79, 82, 83, 94, 96, 97, 98, 100, 101, 103, 105, 106, 107, 108, 109, 110, 112, 115, 117, 118, 121, 123, 124, 125, 127, 138, 184, 185, 186, 222, 231, 234, 235, 238, 243, 244, 246, 255, 258, 260, 263, 264, 269, 274, 277, 278, 280

koding 77, 190, 191, 230, 231, 232, 237, 238, 242, 244, 255, 277

koefisien 262

kolaborasi 2, 3, 4, 5, 8, 10, 11, 16, 17, 19, 22, 26, 28, 49, 50, 53, 221, 243, 280

kolaboratif 6, 16, 21, 228

kompleksitas 49, 272

komputasional 217, 222, 223, 228, 230, 232, 233, 276, 277

konten 1, 2, 3, 4, 6, 7, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 26, 35, 36, 50, 71, 221, 277, 278, 279, 280

konten digital 2, 3, 4, 6, 11, 15, 16, 19, 26, 221, 277, 278, 279, 280

kreatif 2, 3, 4, 8, 11, 16, 29, 40, 54, 221, 224, 228, 234, 273, 278

kreativitas 2, 4, 5, 6, 9, 10, 22, 26, 34, 49, 50

kritis 29, 34, 37, 221, 228, 234, 270, 273

L

label encoding 142, 143, 145, 146, 148, 149, 157, 190, 192

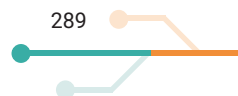
lemmatization 211

library 126, 144, 147, 151, 159, 175, 176, 177, 231, 243, 253, 255, 256, 257, 258, 259, 265, 275, 280

LLM 214, 217, 266, 267, 268, 269, 270, 271, 273, 275, 276

logika berpikir 36, 232

logis 22, 75, 77, 195, 225, 228, 233, 242, 277



M

Machine Learning (ML) 195
majemuk 104, 108, 109, 125
manual control 280
marketplace 131, 198, 205
memprediksi 46, 65, 133, 163, 199, 215, 216, 231, 250, 251, 253, 254, 255, 258, 272
mengevaluasi 19, 197, 227, 233, 234, 255, 263
mengoreksi 53, 248
mesin 5, 28, 29, 30, 31, 35, 40, 42, 49, 50, 71, 77, 85, 88, 90, 93, 99, 160, 165, 196, 197, 207, 209, 210, 218, 223, 228, 231, 233, 248, 267, 278, 280, 282
metode 16, 70, 85, 86, 89, 90, 93, 94, 97, 98, 99, 100, 101, 103, 104, 106, 108, 110, 111, 112, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 160, 176, 200, 222, 239, 243, 247, 250, 252, 258, 273, 277, 279, 280, 281, 282
model 5, 8, 9, 53, 64, 65, 66, 88, 90, 91, 92, 93, 113, 165, 166, 169, 170, 189, 190, 196, 197, 199, 200, 202, 207, 215, 216, 218, 220, 221, 222, 223, 224, 231, 234, 238, 241, 242, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 268, 269, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 285
model KA 65, 66, 223, 224, 231, 234, 249, 250, 251, 252, 253, 254, 255, 256, 258, 259, 261, 262, 263, 272, 276, 277, 278, 280, 281, 282
modular 83, 103, 126, 238, 281
moral 15, 64, 279
mounting 146

N

nama kelas 96, 104
Natural Language Processing (NLP) 195, 196, 207, 210, 231, 267
nominal 99, 149, 278
numerik 133, 135, 138, 142, 143, 159, 258, 278

P

Pandas 141, 147, 151, 153, 165, 188, 252, 257, 261, 265
pelanggaran privasi 36
pemrograman 41, 44, 74, 75, 77, 83, 85, 87, 89, 94, 98, 103, 111, 121, 123, 175, 176, 230, 242, 256, 258, 277, 281
perencanaan bisnis 34
pewarisan 74, 83, 103, 104, 105, 109, 111, 122, 125, 127, 279, 282
platform 13, 18, 19, 20, 23, 26, 54, 71, 195, 204, 205, 217, 247, 278, 285
poisoning 282
pola 6, 31, 43, 49, 62, 63, 86, 100, 196, 199, 200, 204, 222, 226, 228, 229, 231, 232, 248, 252, 258, 281, 282
polimorfisme 74, 83, 110, 111, 121, 122, 125, 127, 281
pranala 274

prediksi 64, 66, 160, 161, 163, 165, 196, 198, 199, 202, 207, 215, 232, 248, 249, 252, 253, 258, 260, 262, 263, 264, 265, 266, 272, 277, 278, 280, 281, 282
presisi 35, 200, 253, 254
primary key 179, 181, 182
privasi 15, 36, 56, 57, 59, 69, 70, 271
private 98, 99, 100, 101, 103, 124, 281
program 46, 73, 75, 76, 77, 78, 79, 80, 83, 85, 94, 96, 97, 98, 100, 101, 105, 106, 107, 108, 109, 110, 112, 115, 117, 118, 120, 122, 133, 134, 137, 138, 141, 152, 155, 166, 176, 178, 182, 183, 184, 187, 190, 222, 231, 234, 238, 243, 245, 246, 258, 264, 266, 278, 279, 280, 281, 282
properti 85
proses 2, 4, 5, 7, 10, 11, 12, 13, 14, 17, 18, 20, 26, 29, 42, 52, 71, 75, 96, 98, 99, 100, 124, 127, 130, 131, 133, 137, 138, 142, 143, 152, 159, 168, 192, 193, 200, 207, 212, 216, 220, 222, 223, 225, 233, 234, 235, 238, 239, 241, 242, 244, 247, 248, 249, 251, 253, 254, 263, 270, 271, 272, 276, 278, 279
protected 100, 281
proteksi 56, 65
proyek 16, 17, 22, 26, 70, 123, 127, 201, 202, 203, 204, 205, 206, 217, 221, 233, 234, 235, 237, 242, 243, 273
public 98, 99, 100, 101, 281

R

rancangan 86, 238, 242, 244, 281
real time 49, 64
regresi 199, 215, 256, 261, 262, 264, 280
reinforcement learning 194, 196, 200
rekan kerja 16
rekayasa 16, 220, 221, 222, 223, 233, 234, 235, 236, 238, 242, 244, 248, 254, 272, 276, 279, 281, 282
rekomendasi 6, 9, 10, 11, 20, 37, 46, 68, 133, 152, 166, 167, 191, 195, 198, 205, 231, 248
relasional 170, 173
requirement 236, 238, 277, 281
robot 29, 34, 40, 75, 110, 196, 200
rumusan 227
runtutan 78

S

scam 37
select 174, 181
self-checkout 30, 42
semi-supervised learning 199, 200
sharing 37, 140, 141
sistem 9, 11, 31, 35, 36, 37, 42, 53, 55, 56, 60, 61, 65, 66, 68, 69, 70, 71, 75, 88, 91, 92, 99, 103, 125, 126, 130, 133, 134, 137, 139, 142, 156, 157, 159, 160, 161, 166, 168, 169, 170, 171, 173, 190, 192, 195, 196, 201, 202, 203, 204, 205, 207, 231, 232, 233, 234, 235, 238, 239, 242, 244, 246, 248, 276, 277, 278, 279, 280, 281, 282

sistematis 77, 195, 220, 221, 223, 225, 228, 229, 233, 234, 235, 241, 244, 277, 279, 282
solusi 11, 22, 46, 48, 49, 50, 59, 61, 75, 76, 192, 199, 200, 217, 219, 220, 221, 223, 224, 227, 228, 230, 232, 233, 234, 235, 236, 238, 241, 242, 244, 247, 272, 273, 278, 279
sumber 40, 66, 89, 208, 226, 270, 281, 282
superintelligence 60
supervised learning 194, 196, 199, 200, 216

T

tabel 21, 43, 47, 93, 105, 131, 133, 153, 158, 169, 170, 171, 177, 178, 179, 180, 181, 182, 184, 190, 198, 205, 209, 241, 250, 251, 256, 257, 261, 264, 265, 268
target encoding 163, 164, 165, 166, 167, 190, 191
task 251
teknologi 2, 5, 6, 29, 30, 31, 32, 33, 35, 40, 41, 42, 45, 50, 53, 55, 58, 59, 61, 65, 69, 70, 72, 132, 175, 176, 195, 196, 198, 200, 207, 208, 209, 214, 218, 219, 220, 222, 223, 232, 233, 234, 235, 244, 266, 270, 276, 279
testing 197, 244, 245, 246, 247, 253, 259, 277, 279, 282
text generator 36
tokenisasi 210, 216
training 197, 199, 207, 253, 259, 274, 282, 284
tuning 254, 274, 279

U

unsupervised learning 194, 196, 199
update 168, 170, 174, 176, 181, 182

V

variabel 87, 88, 115, 121, 143, 151, 153, 200, 244, 261, 277, 280
video 1, 2, 3, 4, 6, 7, 8, 13, 14, 15, 18, 19, 21, 26, 58, 71, 127, 133, 205, 277, 278, 279, 280, 282
visualisasi data 231, 257

W

warna 10, 13, 63, 81, 86, 87, 88, 89, 90, 92, 93, 118, 120, 138, 143, 149, 150, 151, 162, 201, 258
wawancara 16, 47, 226, 236



Ibnu Indarwati, S.T., M.Kom.

PENULIS

Email:

ibnuindarwati19@guru.smk.
belajar.id

Instansi:

SMK Negeri 1 Purwakarta

Alamat Instansi:

Jl. Industri Km. 4 Babakancikao
Purwakarta

Pekerjaan/Profesi:

1. SMKN 1 Purwakarta (2018–saat ini)
2. SMKN 2 Kota Bogor (2014–2018)
3. SMKN 1 Purwakarta (2007–2014)

Pendidikan Terakhir:

1. STMIK Likmi Sistem Informasi (2014)
2. STMIK Bandung Teknik Informatika (2004)
3. SMU Negeri 3 Sragen (1999)
4. SMP Negeri Sambirejo Kabupaten Sragen (1996)
5. SD Negeri Kadipiro 1 Kabupaten Sragen (1993)

Bidang Keahlian:

1. Algoritma dan Struktur Data
2. Rekayasa Perangkat Lunak (*Software Engineering*)
3. Kecerdasan Artifisial (*Artificial Intelligence*)
4. *Data Science* dan *Big Data*
5. Jaringan dan Keamanan Komputer
(*Computer Networks & Cybersecurity*)

Rekam Jejak Karya Tulis:

1. *Buku Panduan Guru Dasar-Dasar Teknik Jaringan Komputer dan Telekomunikasi untuk SMK/MAK Kelas X* (2023)
2. *Dasar-Dasar Teknik Jaringan Komputer dan Telekomunikasi untuk SMK/MAK Kelas X* (2023)
3. *Administrasi Jaringan Komputer Kelas XI* (2021)
4. *Teknologi Jaringan Luas (WAN)* (2021)

Rekam Jejak Penelitian:

1. *Buku Panduan Guru Dasar-Dasar Teknik Jaringan Komputer dan Telekomunikasi untuk SMK/MAK Kelas X* (2023)
2. *Dasar-Dasar Teknik Jaringan Komputer dan Telekomunikasi untuk SMK/MAK Kelas X* (2023)
3. *Administrasi Jaringan Komputer Kelas XI* (2021)
4. *Teknologi Jaringan Luas (WAN)* (2021)

Marwondo, S.T., M.Kom.

Pekerjaan/Profesi:

Dosen Tetap Program Studi Informatika, Universitas Informatika dan Bisnis Indonesia (UNIBI), Bandung (2003–saat ini)

Pendidikan Terakhir:

S2 Sistem Informasi, Bidang Rekayasa Sistem. STMIK LIKMI, Bandung–Jawa Barat (2017)

Bidang Keahlian:

1. Rekayasa Perangkat Lunak
2. Keamanan Komputer
3. Kecerdasan Buatan
4. Multimedia dan Gim

Rekam Jejak Karya Tulis:

1. *Dasar-Dasar Pengembangan Perangkat Lunak dan Gim untuk SMK/MAK Kelas X*, Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi (2023)
2. *Buku Panduan Guru Dasar-Dasar Pengembangan Perangkat Lunak dan Gim untuk SMK/MAK Kelas X*. Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi (2023)
3. *Manajemen Retail di Era Pemasaran Online*. Bandung: UNIBI Press. (2021)

Rekam Jejak Penelitian:

1. *Creating NS2A Binding Affinity in Dengue Virus Using Hybrid Method for Antiviral Prediction* (2024)
2. *Leveraging Molecular Docking and Machine Learning to Predict NS2A Binding Affinity in Dengue Virus* (2024)
3. *Optimasi Penggunaan Teknologi dan Akses Digital untuk Pendidikan Lanjutan pada Kober Nurul Ikhlas* (2024)

Informasi Lain:

1. Scopus ID: [59748557400](#)
2. ORCID: [0009-0008-1726-6346](#)
3. Google Scholar ID: [gJ9HcDYAAAAJ](#)



PENULIS

Email:

m1214w@gmail.com

Instansi:

Universitas Informatika dan Bisnis Indonesia (UNIBI)

Alamat Instansi:

Jl. Soekarno-Hatta No. 643
Bandung, Jawa Barat



PENULIS

Email:
faisalpidie@gmail.com

Instansi:
SMK Negeri 3 Lhokseumawe

Alamat Instansi:
Jl. H. T. Markam, No. 36,
Lhokseumawe, Aceh

Muhammad Faisal, S.Pd., M.Kom.

Riwayat Pekerjaan/Profesi:

1. Guru Informatika,
SMK Negeri 3 Lhokseumawe (2014–saat ini)
2. Fasilitator KKA (2025)
3. Fasilitator Nasional Bidang TIK (2015)
4. Guru Informatika, SMA Negeri 1 Calang (2010–2014)

Riwayat Pendidikan:

1. Magister Teknologi Informasi,
Universitas Malikussaleh (2023)
2. Sarjana Pendidikan Teknik Informatika, Universitas
Negeri Padang (2009)

Bidang Keahlian:

1. Kecerdasan Artifisial
2. *Machine Learning*
3. *Data Science*

Rekam Jejak Penelitian:

*Information and Communication Technology Competencies
Clustering for students at Vocational High School Students
Using K-Means Clustering Algorithm, Ijesty, 2022*

Dr. Ir. Heri Nurdiyanto, M.TI., M.MT.



PENELAAH

Email:

herinurdiyanto@uny.ac.id

Instansi:

Universitas Negeri Yogyakarta

Alamat Instansi:

Jl. Colombo No.1, Karang
Malang, Caturtunggal,
Kec. Depok, Kabupaten
Sleman, Daerah Istimewa
Yogyakarta 55281

Riwayat Pekerjaan/Profesi:

1. Dosen di Universitas Negeri Yogyakarta (2023–saat ini)
2. Dosen di Universitas Dharma Wacana (2011–2023)

Riwayat Pendidikan:

1. Magister manajemen Teknologi,
Institut Informatika dan Bisnis (2024)
2. Pendidikan Teknologi Kejuruan,
Universitas Negeri Yogyakarta (2022)
3. Program Profesi Insinyur,
Universitas Negeri Yogyakarta (2022)
4. Magister Teknik Informatika,
Institut Informatika dan Bisnis (2014)
5. Teknik Informatika,
Universitas Teknologi Yogyakarta (2008)

Bidang Keahlian:

1. *Artificial Intelligence*
2. *Expert System*
3. *Software Engineering*
4. *Decision Support System*

Rekam Jejak Penelaahan/Review Buku:

1. *Reveiwer BTP SMK* (2022–saat ini)
2. *Supervisor BTP* (2023–saat ini)

Rekam Jejak Karya Tulis:

1. *Phyton Basic Programming* (2023)
2. *Cara Cepat Menguasai Desain Grafis Canva* (2023)
3. *Interaksi Manusia dan Komputer* (2023)
4. *Pemrograman Berorientasi Objek Menggunakan Bahasa Pemrograman Phyton* (2024)
5. *Komunikasi Data dan Jaringan* (2024)

Informasi Lain:

1. <https://scholar.google.com/citations?hl=id&user=RW8mObEAAAAJ>
2. <https://www.scopus.com/authid/detail.uri?authorId=57200089726>
3. <https://www.webofscience.com/wos/author/record/C-7376-2016>
4. <https://sciprofiles.com/profile/2395923>

Yuli Wahyuni, S.T., M.T.

Riwayat Pekerjaan/Profesi:

1. Supervisor Buku Pendamping SMK/MAK, Perbukuan Nasional (2023–saat ini)
2. *Reviewer* Jurnal, Jurnal Nasional (2022–saat ini)
3. Editor in Chief Jurnal Aplikasi Bisnis dan Komputer (Jubikom) Vokasi Unpak, Universitas Pakuan (2021–2025)
4. Unit Penjaminan Mutu Program Studi Bisnis Sekolah Vokasi, Universitas Pakuan (2021–2025)
5. Sekretaris Program Studi Teknik Komputer Sekolah Vokasi, Universitas Pakuan (2017–2021)

Riwayat Pendidikan:

1. S2 Teknik Elektro Spesifikasi Sistem Komunikasi dan Informatika, Universitas Brawijaya Malang (2008)
2. S1 Teknik Elektro Spesifikasi Elektronika, Institut Teknologi Nasional Malang (2003)

Bidang Keahlian:

1. *Machine Learning*
2. *Artificial Intelligence*
3. Sistem Informasi
4. Pemrograman Berbasis *Website*

Rekam Jejak Penelaahan/Review Buku:

1. *Journal of Social Science and Multidisciplinary Analysis pada Noble Cendekia Indonesia* (2024–saat ini)
2. Jurnal Pelayanan dan Pengabdian Masyarakat Indonesia (JPPMI), Sekolah Tinggi Ilmu Administrasi YAPPI Makassar (2023–saat ini)
3. Jurnal Pengabdian Masyarakat Indonesia (SEWAGATI), Sekolah Tinggi Ilmu Ekonomi Pariwisata Indonesia (2023–saat ini)
4. Jurnal Sains dan Teknologi Universitas Cenderawasih (SAINTEK FT) (2023–saat ini)

Rekam Jejak Karya Tulis:

1. *IoT Dalam Sterilisasi Medis Sebagai Inovasi Untuk Keamanan Pasien Sebelum Operasi* (2025)
2. *Alat Pemantau Kalori Harian Berbasis Sensor Akselerometer dan Arduino Nano* (2025)
3. *Optimasi Model Rekam Medik Jantung Berbasis IoT* (2024)
4. *Sistem Monitoring Network Menggunakan Zabbix Pada Linux Ubuntu* (2024)
5. *Buku Ajar Membuat Website Profile Eduwisata Kacang Koro Menggunakan PHP MySQLI* (2022)



PENELAAH

Email:
yuli_wahyuni@unpak.ac.id

Instansi:
Universitas Pakuan

Alamat Instansi:
Jl. Pakuan PO Box 452
Bogor 16143

Nana Maulana, S.Sn.

Bidang Keahlian:

1. Ilustrasi Buku Anak
2. Komik

Pekerjaan/Profesi:

1. *Owner* Bikindong Creative Studio (2017–saat ini)
2. Guru Seni dan Budaya SMA Bintara Depok (2015–2018)
3. Kartunis Penerbit Gerrmedia Komik (2011–2015)

Riwayat Pendidikan:

1. S1 : Universitas Indraprasta PGRI (2007–2011)
2. SMA : SMK Budi Utomo Depok (2001–2004)
3. SMP : SLTP Negeri 6 Depok (1998–2001)
4. SD : SD Al Khoeriyah (1992–1998)

Rekam Jejak Pembuatan Ilustrasi:

1. Komik Lumi Tidak Malu (2024)
2. Komik Serba 90-an (2015)

Informasi Lain:

1. Juara 1 Lomba Komik Kereta Cepat Internasional (2023)
2. Juara 3 Lomba Komik Bank Indonesia Net Festival (2021)
3. Juara 1 Lomba Komik Pembelajaran SD (2019)



ILUSTRATOR

Email:

kartunaung@gmail.com

Medsos:

@nananaung

Instansi:

Freelance

Alamat Instansi:

Depok, Jawa Barat





EDITOR

Email:
wonderfullwidi@gmail.com

Instansi:
SMPN 1 Cileunyi

Alamat Instansi:
Jl. Raya Cinunuk, Cimekar, Kec.
Cileunyi, Kab. Bandung

Nurhasanah Widianingsih, M.Pd.

Riwayat Pekerjaan/Profesi:

1. Guru Bahasa Indonesia di SMPN 1 Cileunyi (2023–saat ini)
2. Editor, Pusbuk–Kemdikbud (2021–saat ini)
3. Penilai BTP–Kemdikbud (2024)
4. Penulis Buku Pandik, PT. Duta Pustaka Indonesia (2024)
5. Guru Bahasa Indonesia di SMP Labschool UPI Cibiru (2019–2023)
6. Penulis Buku Bahasa Indonesia SD, CV. Eka Prima Mandiri (2023)
7. Penulis Modul Kurikulum Merdeka SMP, Pusmenjar–Kemdikbud (2020)
8. Penelaah Modul SMP Terbuka, Ditjen SMP–Kemdikbud (2020)

Riwayat Pendidikan:

S2 Pendidikan Bahasa Indonesia,
Universitas Pendidikan Indonesia (2020)

Rekam Jejak Penyuntingan:

1. *Buku Siswa dan Buku Guru Manajemen Logistik, SMK/MAK Kelas XI*, Pusbuk–Kemdikbud (2024)
2. *Buku Panduan Guru Pengembangan Komunikasi bagi Peserta Didik dengan Hambatan Pendengaran, SDLB, SMPLB, dan SMALB*, Pusbuk–Kemdikbud (2024)
3. *Buku Siswa dan Buku Guru Dasar-Dasar Teknik Konstruksi Kapal, SMK/MAK Kelas X Semester 1 dan 2*, Pusbuk–Kemdikbud (2022)
4. *Buku Panduan Guru Prakarya dan Kewirausahaan: Rekayasa, SMA/MA Kelas XI*, Pusbuk–Kemdikbud (2022)
5. *Buku Panduan Guru Prakarya dan Kewirausahaan: Rekayasa, SMA/MA Kelas XII*, Pusbuk–Kemdikbud (2022)

Rekam Jejak Karya Tulis:

1. *Pembelajaran Aktif dan Kolaboratif dalam Kurikulum Merdeka*, PT. Duta Pustaka Indonesia (2024)
2. *Belajar Tuntas (Beta) Bahasa Indonesia Kelas 1*, CV. Eka Prima Mandiri (2023)
3. *Belajar Tuntas (Beta) Bahasa Indonesia Kelas IV*, CV. Eka Prima Mandiri (2023)

Nadia Mahatmi



EDITOR VISUAL

Riwayat Pekerjaan/Profesi (10 Tahun Terakhir)

1. (Dosen Desain Komunikasi Visual, Universitas Multimedia Nusantara (2017–saat ini)
2. Dosen Desain Komunikasi Visual, Telkom University (2015–2017)

Riwayat Pendidikan dan Tahun Belajar

1. Magister Desain, Institut Teknologi Bandung (2012–2015)
2. Sarjana Desain Komunikasi Visual, Institut Teknologi Bandung (2005–2009)

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir)

1. *Board Game Design to Learn about User Persona in Entrepreneurship Programme in Kurikulum*
2. *Uji Coba Buklet Aktivitas Museum Bank Indonesia untuk Siswa Sekolah Menengah* (2021)
3. *Perancangan Board Game Kolaboratif. Studi Kasus: Legenda Gunung Tondoyan* (2021)
4. *Mascot Design for the Indonesian Pavilion at World Expo 2020* (2020)
5. *Activity Booklet Design for Museum Bank Indonesia for Middle School Students* (2020)

Informasi Lain



Email:

nadia.mahatmi@umn.ac.id

Instansi:

Universitas
Multimedia Nusantara

Alamat Instansi

Jl. Scientia Boulevard,
Gading Serpong, Kel. Curug
Sangereng, Kec. Kelapa Dua,
Kab. Tangerang, Prop. Banten
15810

Keahlian:

Ilustrasi



Mohamad Lutvi

DESAINER

Email:

lulutvi28@gmail.com

Bidang Keahlian:

1. Desain Grafis
2. Multimedia

Riwayat Pekerjaan/Profesi:

Pekerja Lepas:

1. Fotografer (2015–saat ini)
2. Desainer Grafis (2012–saat ini)
3. Musik: *Sound Designer, Composer & Arranger* (2010–saat ini)
4. *Web Designer* (2012–2018)

Riwayat Pendidikan:

1. D3 Teknik Elektro-Elektronika, Instrumentasi dan Kendali, Universitas Negeri Jakarta (2001–2004)
2. SMU Negeri 58 Jakarta (1998–2001)
3. SMP Negeri 208 Jakarta (1995–1998)
4. SD Negeri Ciracas 05 Jakarta Timur (1989–1995)

Portfolio:

<https://www.youtube.com/@LutviBinsimin>

