

# Fondasi Koding & Analisis Data

untuk SMK dan umum



2025

# Kelas 10 Semester 1

## Fondasi Koding & Analisis Data

Aolia Ikhwanudin  
Onno W. Purbo

**Institut Teknologi Tangerang Selatan (ITTS)**  
**OnnoCenter**  
**2025**

# Lisensi & Catatan Karya

Karya ini dilisensikan di bawah lisensi **Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**.

Artinya, **siapa pun bebas untuk menggunakan, menyalin, membagikan, dan mengadaptasi** materi ini, **dengan syarat**:

- **Memberikan atribusi yang sesuai** (menyebut sumber asli),
- **Menyertakan lisensi yang sama** jika dimodifikasi atau dikembangkan lebih lanjut.

Lihat detail lisensi di sini <https://creativecommons.org/licenses/by-nd/4.0/legalcode.id>

Desain Cover: Irwan Siswanto

## Disclaimer

Materi pembelajaran ini dibuat dengan **dana swadaya masyarakat Indonesia** dan **kontribusi sukarela dari para dosen di Institut Teknologi Tangerang Selatan (ITTS)**.

Karya ini didistribusikan secara bebas untuk mendukung pendidikan digital dan kecerdasan buatan di kalangan pelajar Indonesia.

Kami **memohon maaf** jika terdapat kekurangan dalam isi maupun penyajian materi ini. Kritik dan saran sangat kami harapkan untuk terus menyempurnakan karya serupa di masa depan.

# Kata Pengantar

Buku ini disusun sebagai wujud nyata kepedulian sivitas akademika Institut Teknologi Tangerang Selatan (ITTS) terhadap perkembangan pendidikan kejuruan di Indonesia, khususnya bagi siswa Sekolah Menengah Kejuruan (SMK). Di tengah kemajuan teknologi yang begitu pesat, kami terdorong untuk menghadirkan materi pembelajaran **yang menantang, aplikatif, dan relevan dengan dunia industri digital**, agar siswa dapat mencapai **kemampuan maksimalnya** dalam memahami dan membangun sistem *Artificial Intelligence (AI)*.

Perlu kami sampaikan bahwa **materi dalam buku ini menuntut tingkat pemahaman teknis yang cukup tinggi**, terutama karena pembahasannya **sangat kental dengan praktik *coding* menggunakan bahasa Python** untuk membuat dan melatih model AI secara mandiri. Oleh karena itu, **tidak semua siswa, sekolah, maupun guru mungkin dapat langsung menyerap seluruh isi buku ini secara optimal**.

Bagi pembaca yang masih merasa kesulitan mengikuti langkah-langkah teknis yang disajikan, **kami sangat menyarankan untuk terlebih dahulu mempelajari buku “AI untuk SMA”**, yang disusun dengan pendekatan yang lebih **sederhana dan bertahap** sebelum melanjutkan ke tingkat SMK ini.

Penyusunan dan penerbitan buku ini dilakukan secara mandiri dengan semangat berbagi pengetahuan terbuka. Kami mengucapkan terima kasih kepada berbagai pihak yang telah mendukung dan memberi masukan dalam proses penulisan. Kritik dan saran yang membangun sangat kami harapkan agar edisi berikutnya dapat semakin baik dan mudah diakses oleh seluruh siswa Indonesia.

Kami berharap buku ini dapat menjadi **jembatan antara teori dan praktik AI**, sekaligus menginspirasi siswa SMK untuk **berpikir kreatif, logis, dan bertanggung jawab dalam menggunakan teknologi**. Dengan rendah hati, kami mohon doa dan dukungan agar karya sederhana ini dapat memberi manfaat luas dan menjadi amal jariyah bagi semua pihak yang terlibat.

Tangerang Selatan, Oktober 2025

**Penulis**

# Daftar Isi

|                                                                 |          |
|-----------------------------------------------------------------|----------|
| <b>Disclaimer.....</b>                                          | <b>2</b> |
| <b>Kata Pengantar.....</b>                                      | <b>3</b> |
| <b>Daftar Isi.....</b>                                          | <b>4</b> |
| <b>BAB 1: Pengenalan Coding dan AI di SMK.....</b>              | <b>8</b> |
| Tujuan Pembelajaran.....                                        | 8        |
| Peta Konsep Sederhana.....                                      | 8        |
| Apersepsi: Siapa yang Suka Main Game?.....                      | 9        |
| Penjelasan Teori: Ngobrol Dengan Komputer.....                  | 9        |
| Kenalan Singkat dengan Colab.....                               | 9        |
| Coding Itu Apa? (Halo Dunia!).....                              | 10       |
| Data Tabel dengan Pandas.....                                   | 11       |
| AI Itu Apa? (Mini Machine Learning dengan Keras).....           | 12       |
| Pseudocode: Cara AI “Belajar”.....                              | 13       |
| Simpan & Bagikan.....                                           | 13       |
| Tips Anti-Error.....                                            | 14       |
| Ringkasannya.....                                               | 14       |
| Kasus Ilustrasi: Dari Hobi Sampai Sekolah.....                  | 15       |
| Soal Latihan dan Pembahasan.....                                | 17       |
| Fun Facts!.....                                                 | 20       |
| Tips Belajar dan Menghafal Cepat.....                           | 20       |
| Aktivitas Siswa.....                                            | 21       |
| Rangkuman Singkat.....                                          | 26       |
| Latihan Soal.....                                               | 27       |
| Tugas Proyek (5 Soal).....                                      | 31       |
| Tujuan Pembelajaran.....                                        | 37       |
| Peta Konsep Sederhana.....                                      | 37       |
| Apersepsi: Kenapa Handphone Kamu Sering Lemot?.....             | 37       |
| Penjelasan Konsep (Teori).....                                  | 38       |
| Apa itu Computational Thinking?.....                            | 38       |
| Empat Pilar, dengan contoh yang dekat.....                      | 38       |
| Dekomposisi (Decomposition) — “Bagi yang besar jadi kecil”..... | 38       |
| Pengenalan Pola (Pattern Recognition) — “Cari yang sama”.....   | 38       |
| Abstraksi (Abstraction) — “Fokus ke inti”.....                  | 39       |
| Algoritma (Algorithm) — “Langkah jelas, berurutan”.....         | 39       |
| Dari konsep ke kode: jalur open source.....                     | 39       |
| Tips cepat biar makin nempel.....                               | 40       |
| Mini glosarium (biar mantap).....                               | 40       |
| Contoh Kasus atau Ilustrasi.....                                | 41       |
| Soal Latihan dan Pembahasan.....                                | 44       |
| Fun Facts.....                                                  | 47       |
| Tips Belajar atau Menghafal Cepat.....                          | 48       |
| Aktivitas Siswa.....                                            | 48       |

|                                                           |           |
|-----------------------------------------------------------|-----------|
| Rangkuman.....                                            | 53        |
| Latihan Soal.....                                         | 53        |
| Tugas Proyek (5 Soal).....                                | 57        |
| <b>BAB 3: Python Dasar.....</b>                           | <b>62</b> |
| Tujuan Pembelajaran.....                                  | 62        |
| Peta Konsep Sederhana.....                                | 62        |
| Apersepsi: Misteri Angka Hilang.....                      | 62        |
| Penjelasan Teori: Python itu Gampang!.....                | 62        |
| Kenapa Harus Python?.....                                 | 62        |
| Python = Resep Masakan.....                               | 63        |
| Contoh Sederhana.....                                     | 63        |
| Python di Kehidupan Sehari-Hari.....                      | 63        |
| Python untuk AI dan Machine Learning.....                 | 63        |
| Belajar Python di Google Colab.....                       | 64        |
| Rangkuman Singkat.....                                    | 64        |
| Kasus Ilustrasi: Python di Kehidupan Sehari-hari.....     | 64        |
| Soal Latihan & Pembahasan.....                            | 68        |
| Fun Facts.....                                            | 70        |
| Tips Belajar & Menghafal Cepat.....                       | 70        |
| Aktivitas Siswa: Python for Fun!.....                     | 71        |
| Rangkuman.....                                            | 76        |
| Latihan Soal.....                                         | 76        |
| Tugas Proyek (5 Soal).....                                | 80        |
| <b>BAB 4: Struktur Kontrol.....</b>                       | <b>84</b> |
| Tujuan Pembelajaran: Jadi Sutradara Kode.....             | 84        |
| Peta Konsep Sempel: Jalan-jalan di Dunia Kode.....        | 84        |
| Apersepsi: Mesin Fotokopi yang Bingung.....               | 84        |
| Penjelasan Teori: Ngobrolin If dan Loop.....              | 84        |
| Apa itu Struktur Kontrol? (ngerti cepat).....             | 84        |
| Bahan Pokok: Kondisi & Operator (fondasi dulu).....       | 85        |
| Percabangan: If–Elif–Else (ambil keputusan).....          | 85        |
| Perulangan: for vs while (jalan berulang).....            | 86        |
| Pattern Penting: Validasi & Menu (dipakai tiap hari)..... | 86        |
| Kesalahan Umum & Antitips (supaya tidak terjebak).....    | 87        |
| Ringkasan.....                                            | 88        |
| Mini Tantangan (coba dalam 3 menit).....                  | 88        |
| Contoh Kasus atau Ilustrasi.....                          | 88        |
| Contoh Soal dan Pembahasan.....                           | 92        |
| Fun Facts.....                                            | 97        |
| Tips Belajar & Menghafal Cepat.....                       | 97        |
| Aktivitas Siswa.....                                      | 97        |
| Catatan Akhir.....                                        | 101       |
| Rangkuman.....                                            | 102       |
| Latihan Soal.....                                         | 102       |

|                                                        |            |
|--------------------------------------------------------|------------|
| Tugas Proyek (5 Soal).....                             | 107        |
| <b>BAB 5: Pengolahan Data Dasar dengan Pandas.....</b> | <b>112</b> |
| Tujuan Pembelajaran.....                               | 113        |
| Peta Konsep Sederhana.....                             | 113        |
| Apersepsi: Bayangin Kalian Buka Kafe!.....             | 113        |
| Penjelasan Teori.....                                  | 114        |
| Kenapa Penting Belajar Pandas?.....                    | 114        |
| Ilustrasi Sederhana.....                               | 114        |
| Teori Super Gampang.....                               | 114        |
| Analogi Salad.....                                     | 115        |
| Aktivitas dengan Pandas.....                           | 115        |
| Rangkuman.....                                         | 115        |
| Contoh Kasus atau Ilustrasi.....                       | 115        |
| Contoh Soal dan Pembahasan.....                        | 117        |
| Fun Facts!.....                                        | 119        |
| Tips Belajar & Menghafal Cepat.....                    | 119        |
| Aktivitas Siswa.....                                   | 120        |
| Rangkuman.....                                         | 123        |
| Latihan Soal.....                                      | 123        |
| Tugas Proyek (5 Soal).....                             | 127        |
| <b>BAB 6: Visualisasi Data dengan matplotlib.....</b>  | <b>130</b> |
| Tujuan Pembelajaran.....                               | 130        |
| Peta Konsep Sederhana.....                             | 130        |
| Apersepsi: Tebak Gambar!.....                          | 130        |
| Penjelasan Teori.....                                  | 131        |
| Apa itu matplotlib?.....                               | 131        |
| Kenapa Visualisasi Itu Penting?.....                   | 131        |
| Contoh Perintah Dasar matplotlib.....                  | 131        |
| Analogi Seru.....                                      | 132        |
| Contoh Kasus atau Ilustrasi.....                       | 132        |
| Contoh Soal dan Pembahasan.....                        | 133        |
| Fun Facts!.....                                        | 136        |
| Tips Belajar & Menghafal Cepat.....                    | 136        |
| Aktivitas Siswa.....                                   | 136        |
| Rangkuman.....                                         | 141        |
| Latihan Soal.....                                      | 141        |
| Tugas Proyek (5 Soal).....                             | 146        |
| <b>BAB 7: Pengenalan Orange Data Mining.....</b>       | <b>151</b> |
| Tujuan Pembelajaran.....                               | 151        |
| Peta Konsep Sederhana.....                             | 151        |
| Apersepsi: Kamu Mau Bikin Tim Futsal Juara!.....       | 151        |
| Penjelasan Teori.....                                  | 152        |
| Apa Itu Orange Data Mining?.....                       | 152        |
| Cara Instalasi Orange Data Mining.....                 | 152        |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| 1. Install di Windows.....                                          | 152        |
| 2. Install di Linux (Ubuntu / Debian).....                          | 152        |
| 3. Install di macOS.....                                            | 152        |
| 4. Install via Anaconda (Cross-Platform).....                       | 152        |
| Kenapa Keren?.....                                                  | 153        |
| Konsep Dasar Orange.....                                            | 153        |
| Contoh Kasus atau Ilustrasi.....                                    | 154        |
| Contoh Soal dan Pembahasan.....                                     | 155        |
| Fun Facts!.....                                                     | 159        |
| Tips Belajar & Menghafal Cepat.....                                 | 159        |
| Aktivitas Siswa.....                                                | 159        |
| Rangkuman.....                                                      | 163        |
| Latihan Soal.....                                                   | 163        |
| Tugas Proyek (5 Soal).....                                          | 166        |
| <b>BAB 8: Mini Proyek – Analisis Dataset Sederhana.....</b>         | <b>171</b> |
| Tujuan Pembelajaran.....                                            | 171        |
| Peta Konsep Sederhana.....                                          | 171        |
| Apersepsi: Kamu Punya Warung Kopi Terhits!.....                     | 171        |
| Penjelasan Teori.....                                               | 171        |
| Kenapa Harus Belajar Mini Proyek?.....                              | 171        |
| Alur Mini Proyek Itu Simple.....                                    | 171        |
| Analogi Seru.....                                                   | 172        |
| Contoh Kasus Sederhana.....                                         | 172        |
| Contoh Kasus atau Ilustrasi.....                                    | 173        |
| Contoh Soal dan Pembahasan.....                                     | 175        |
| Fun Facts!.....                                                     | 180        |
| Tips Belajar & Menghafal Cepat.....                                 | 180        |
| Aktivitas Siswa.....                                                | 181        |
| Rangkuman.....                                                      | 183        |
| Latihan Soal.....                                                   | 184        |
| Tugas Proyek (5 Soal).....                                          | 188        |
| <b>Institut Teknologi Tangerang Selatan (ITTS) dan Penulis.....</b> | <b>191</b> |

# BAB 1: Pengenalan Coding dan AI di SMK

Halo adik-adik semua, apa kabar? Semoga selalu semangat ya!

Di bab ini, kita akan belajar tentang hal-hal yang mungkin terdengar keren dan rumit, tapi sebenarnya gampang banget, yaitu **koding** dan **AI** atau **Kecerdasan Buatan**. Jangan khawatir, kita akan belajar sambil main dan santai, kok. Siap? Yuk, kita mulai petualangan kita!

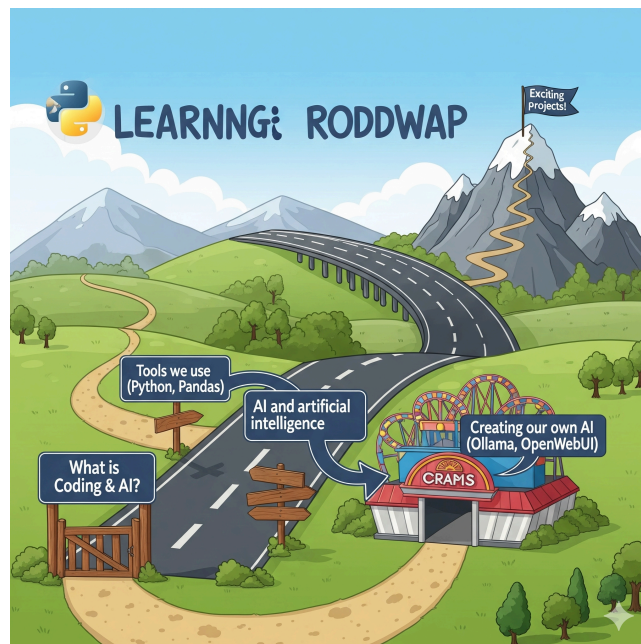
## Tujuan Pembelajaran

Setelah kalian selesai membaca bab ini, kalian akan bisa:

- Paham apa itu coding dan AI.
- Tahu apa saja yang bisa dibuat dengan coding dan AI, misalnya bikin game atau aplikasi.
- Bisa menjelaskan kenapa coding dan AI itu penting banget di zaman sekarang.
- Bisa pakai tool-tool keren buat mulai belajar coding dan AI.

## Peta Konsep Sederhana

Biar nggak bingung, coba bayangkan ini seperti peta jalan kita:



- **Pintu Gerbang:** Apa itu Coding & AI?
- **Jalan Utama:** tool-tool yang kita pakai (Python, Pandas).
- **Jalan Tol:** AI dan kecerdasan buatan.
- **Taman Hiburan:** Bikin AI sendiri (Ollama, OpenWebUI).
- **Puncak Gunung:** Proyek seru!

## Apersepsi: Siapa yang Suka Main Game?

Pasti banyak yang suka main game, kan? Nah, coba bayangkan kalian lagi main **Mobile Legends** atau **Free Fire**. Karakter-karakter di game itu bisa jalan, nembak, atau pakai *skill* karena ada **koding** di belakangnya. Coding itu seperti "otak" yang ngasih perintah ke game.

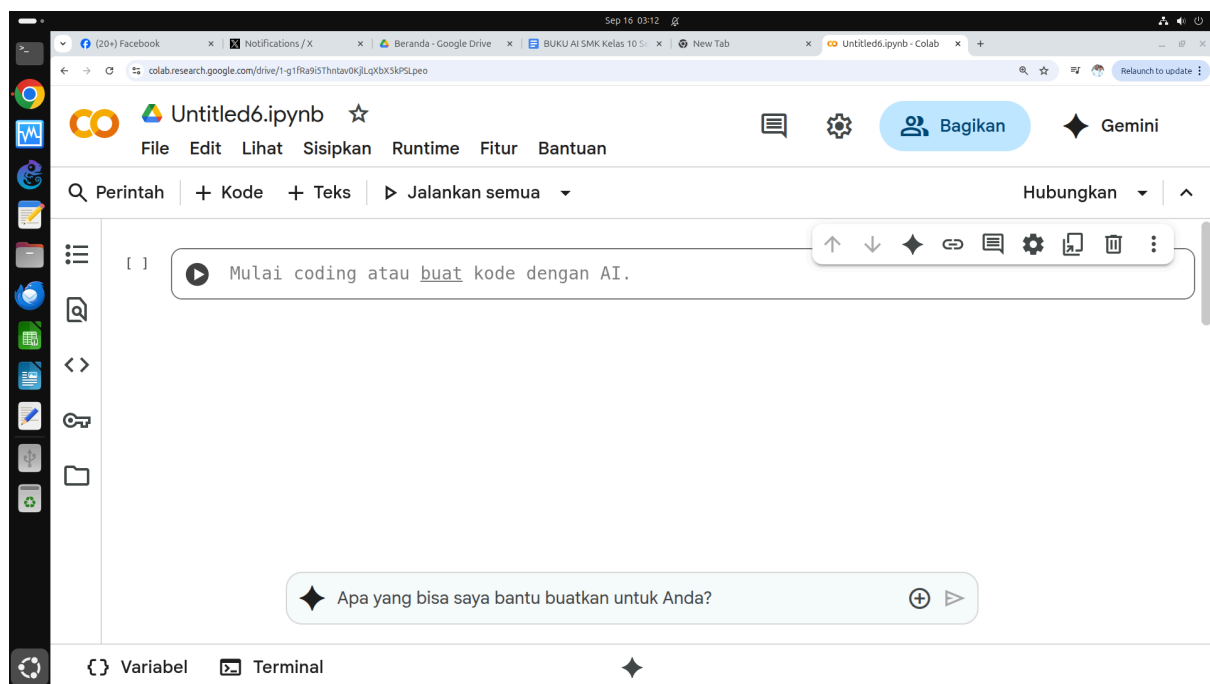
Terus, kalau kalian pakai aplikasi TikTok, kenapa video yang muncul di "**For You Page**" itu kok bisa pas banget sama kesukaan kalian? Itu berkat **AI**, guys! AI itu kayak teman pintar yang tahu apa yang kalian suka, terus ngasih rekomendasi video yang pas. Keren, kan? Jadi, coding itu bikin sesuatu, dan AI itu bikin sesuatu itu jadi lebih pintar.

## Penjelasan Teori: Ngobrol Dengan Komputer

### Kenalan Singkat dengan Colab

- **Apa itu Colab?** Tempat ngetik kode *Python* di web.
- **Apa yang bisa?** Jalankan kode, simpan ke Drive, pakai *GPU*, pakai *library open source*.
- **Istilah penting:** *notebook*, *cell*, *runtime*, *GPU*, *dataset*.

### Langkah cepat mulai:



1. Buka [colab.research.google.com](https://colab.research.google.com).
2. Pilih **New Notebook**.
3. Tekan **+ Code** buat tambah *cell*.
4. Ketik kode. Tekan **Shift+Enter** buat jalanin.

5. Kalau butuh *GPU*: **Runtime** → **Change runtime type** → **Hardware accelerator: GPU** → **Save**.

## Coding Itu Apa? (Halo Dunia!)

Coding itu “ngomong” ke komputer. Kita kasih perintah, komputer jalankan.  
Di Colab, ini gampang banget.

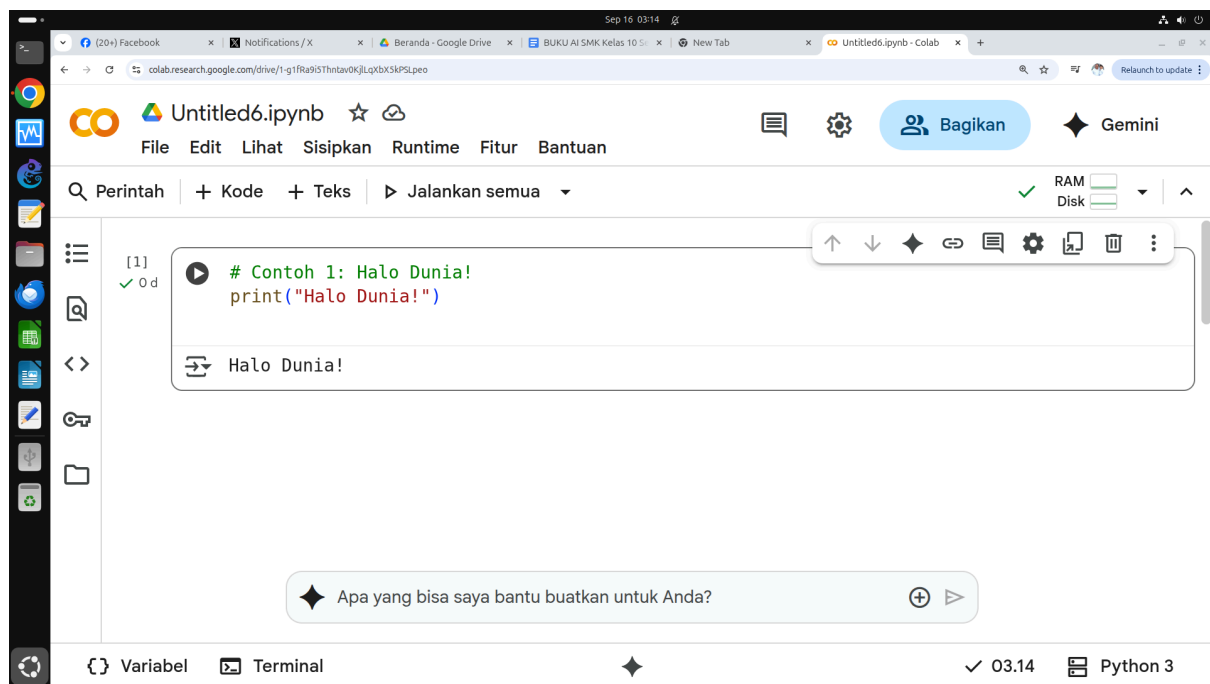
### Tujuan:

- Kenal *cell* kode.
- Cetak teks.

### Langkah:

- Tambah *cell*.
- Ketik kode di bawah.
- Jalankan.

```
# Contoh 1: Halo Dunia!  
print("Halo Dunia!")
```



### Tantangan mini:

- Ubah teks jadi namamu.
- Cetak dua baris.
- Tambahkan *emoji*.

## Data Tabel dengan *Pandas*

Kita olah data seperti *Excel*, tapi lewat kode. Cepat. Rapi. Ulang-ulang bisa.

### Tujuan:

- Baca data kecil.
- Hitung rata-rata.
- Cari nilai terbesar.

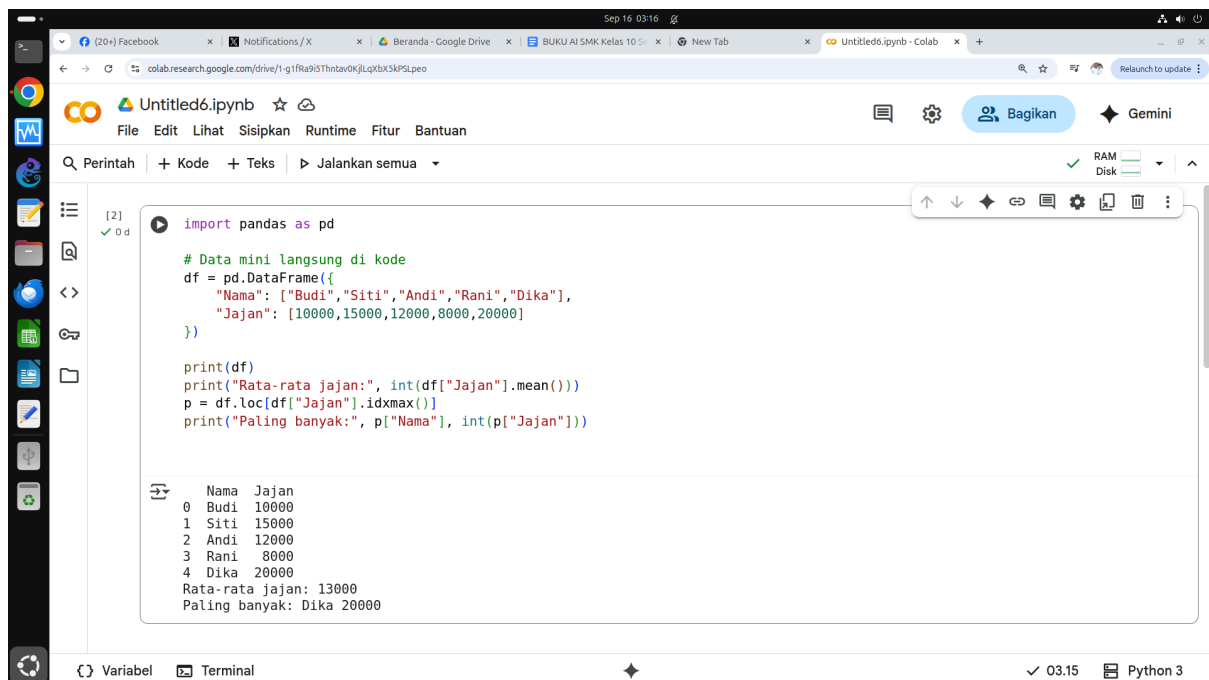
### Langkah:

1. Buat *cell* baru.
2. Jalankan kode.

```
import pandas as pd

# Data mini langsung di kode
df = pd.DataFrame({
    "Nama": ["Budi", "Siti", "Andi", "Rani", "Dika"],
    "Jajan": [10000, 15000, 12000, 8000, 20000]
})

print(df)
print("Rata-rata jajan:", int(df["Jajan"].mean()))
p = df.loc[df["Jajan"].idxmax()]
print("Paling banyak:", p["Nama"], int(p["Jajan"]))
```



```
import pandas as pd

# Data mini langsung di kode
df = pd.DataFrame({
    "Nama": ["Budi", "Siti", "Andi", "Rani", "Dika"],
    "Jajan": [10000, 15000, 12000, 8000, 20000]
})

print(df)
print("Rata-rata jajan:", int(df["Jajan"].mean()))
p = df.loc[df["Jajan"].idxmax()]
print("Paling banyak:", p["Nama"], int(p["Jajan"]))
```

|   | Nama | Jajan |
|---|------|-------|
| 0 | Budi | 10000 |
| 1 | Siti | 15000 |
| 2 | Andi | 12000 |
| 3 | Rani | 8000  |
| 4 | Dika | 20000 |

Rata-rata jajan: 13000  
Paling banyak: Dika 20000

### Tantangan mini:

- Urutkan dari jajan terbesar.
- Simpan ke file:

```
df.sort_values("Jajan",ascending=False).to_csv("ringkasan.csv",
index=False)
```

- Coba *upload* CSV kalian sendiri (klik ikon folder di kiri → *upload*).

### AI Itu Apa? (Mini *Machine Learning* dengan *Keras*)

AI belajar dari *dataset*. Kita coba *neural network* mini: menebak **Lulus (1)** atau **Tidak (0)** dari dua fitur: **Nilai Tugas** dan **Kehadiran**.

#### Tujuan:

- Lihat alur AI: data → model → latih (*train*) → prediksi.
- Pakai **Keras** di Colab.

#### Langkah:

1. Aktifkan *GPU* (opsional).
2. Jalankan kode.

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

# Data mini: [nilai_tugas, kehadiran], label: 1=lulus, 0=tidak
X =
np.array([[90,95],[85,90],[70,80],[60,60],[50,70],[95,85],[40,50
],[88,92]], dtype="float32")
y = np.array([1,1,1,0,0,1,0,1], dtype="float32")
# Bangun model sederhana
model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])

model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
model.fit(X, y, epochs=200, verbose=0)

# Uji dua data baru
uji = np.array([[75,85],[55,60]], dtype="float32")
pred = model.predict(uji, verbose=0).flatten()
print("Probabilitas:", pred)
print("Label akhir:", (pred >= 0.5).astype(int))
```

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

# Data mini: [nilai_tugas, kehadiran], label: 1=lulus, 0=tidak
X = np.array([[90,95],[85,90],[70,80],[60,60],[50,70],[95,85],[40,50],[88,92]], dtype="float32")
y = np.array([1,1,1,0,0,1,0,1], dtype="float32")
# Bangun model sederhana
model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])

model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
model.fit(X, y, epochs=200, verbose=0)

# Uji dua data baru
uji = np.array([[75,85],[55,60]], dtype="float32")
pred = model.predict(uji, verbose=0).flatten()
print("Probabilitas:", pred)
print("Label akhir:", (pred >= 0.5).astype(int))
```

Probabilitas: [0.00012563 0.00154356]  
Label akhir: [0 0]

### Tantangan mini:

- Tambah fitur **UTS** dan **UAS**.
- Ubah neuron dari 4 jadi 8.
- Bandingkan hasilnya.

### Pseudocode: Cara AI “Belajar”

*Pseudocode* itu resep bahasa manusia. Biar gampang paham sebelum bikin kode.

### Contoh: AI kenali kucing vs anjing

```
kumpulkan banyak gambar (dataset)
bagi jadi data_latih dan data_uji
bangun model (neural network) sederhana
latih model pada data_latih
uji model pada data_uji
jika akurasi bagus:
    pakai model untuk gambar baru
else:
    perbaiki data atau model, ulangi
```

### Hubungan ke Colab:

- Taruh *dataset* kecil di Drive.
- Baca di Colab.
- Latih model.
- Simpan model.

### Simpan & Bagikan

- **Simpan otomatis** ke Drive (judul *notebook* di kiri atas, ganti nama).
- **Unduh:** File → Download `.ipynb`.
- **Hubungkan ke Drive** (opsional):

```
from google.colab import drive
drive.mount('/content/drive') # ikuti instruksi yang muncul
```

## Tips Anti-Error

- Jalankan *cell* **dari atas ke bawah**.
- Kalau *runtime* bermasalah: **Runtime** → **Restart runtime**.
- Saat *import* gagal: `!pip install nama_library` (mis. `!pip install pandas`).
- Simpan kerjaanmu sering-sering.

## Ringkasannya

- **Coding** = “ngobrol” ke komputer pakai *Python*.
- **Colab** = tempat latihan cepat, gratis, tanpa install.
- **Data** pakai **Pandas**.
- **AI** mini pakai *machine learning* dengan **Keras** dan *neural network*.
- Semua tool **open source**. Belajar bisa mulai **sekarang**.

Semangat bereksperimen! Mulai dari `print("Halo Dunia!")`, lanjut data kecil, lalu AI mini. Pelan-pelan tapi konsisten

# Kasus Ilustrasi: Dari Hobi Sampai Sekolah

## Kasus 1: Sepak Bola: Prediksi Pemenang

Bayangkan kalian lagi suka Liga Champions. Pengen tahu siapa yang bakal menang? Kita bisa pakai data gol, tendangan, atau kartu merah. Dengan *Python + Pandas*, kita bisa olah datanya.

### Ilustrasi:

- Data: Skor pertandingan sebelumnya.
- Target: Prediksi pemenang.

### Pseudocode:

```
data = baca_file("hasil_pertandingan.csv")
hitung_rata2_gol(data)
jika rata2_gol_timA > rata2_gol_timB:
    print("Tim A kemungkinan menang")
else:
    print("Tim B kemungkinan menang")
```

## Kasus 2: TikTok: Rekomendasi Lagu

Kalian suka dengerin lagu di TikTok? Pengen AI yang kasih rekomendasi lagu sesuai mood kalian?

Gunakan *Pandas* untuk baca data lagu favorit teman-teman.

### Ilustrasi:

- Data: Lagu + jumlah “like”.
- Target: Rekomendasi lagu teratas.

### Pseudocode:

```
data = baca_file("lagu_teman.csv")
urutkan data berdasarkan jumlah_like
tampilkan 3 lagu_teratas
```

### Kasus 3: Jualan Makanan: Cek Stok Otomatis

Bayangin kalian buka usaha minuman kekinian. Ribet kalau stok bahan habis tiba-tiba. Nah, Python bisa jadi asisten stok.

#### Ilustrasi:

- Data: Stok gula, teh, susu.
- Target: Peringatan stok menipis.

#### Pseudocode:

```
stok = {"gula": 2, "teh": 5, "susu": 1}
untuk bahan, jumlah dalam stok:
    jika jumlah < 2:
        print(bahan, "hampir habis!")
```

### Kasus 4: Game Favorit: Diskon Karakter

Suka main Valorant atau Genshin Impact? Kalian bisa bikin bot sederhana yang cek kapan karakter favorit diskon.

#### Ilustrasi:

- Data: Tanggal promo karakter.
- Target: Notifikasi saat diskon.

#### Pseudocode:

```
promo = {"Ganyu": "15 Sept", "Viper": "20 Sept"}
hari_ini = "15 Sept"

jika hari_ini dalam promo.values():
    print("Hari ini ada promo karakter!")
```

### Kasus 5: Tugas Sekolah: AI Pencari Jawaban

Banyak tugas sekolah bikin pusing? Kalian bisa bikin *RAG (Retrieval-Augmented Generation)* lokal pakai **Ollama + OpenWebUI**. AI ini bisa cari jawaban di file dokumen kalian.

#### Ilustrasi:

- Data: File catatan pelajaran.
- Target: AI jawab pertanyaan dari catatan.

### Pseudocode:

```
dokumen = index_dokumen("catatan_sekolah/")
pertanyaan = input("Tulis pertanyaanmu: ")
jawaban = cari_jawaban(dokumen, pertanyaan)
print(jawaban)
```

## Kenapa Ini Seru?

- **Sederhana:** Kita mulai dari data kecil, bisa diketik sendiri.
- **Nyata:** Semua kasus ada di kehidupan sehari-hari (game, TikTok, jualan).
- **Gratis:** Semua pakai *open source tools* (Python, Pandas, Keras).
- **Menantang:** Bikin kalian mikir, “Eh ternyata aku bisa bikin AI sendiri ya!”

## Soal Latihan dan Pembahasan

### Soal 1: Halo Dunia!

#### Soal:

Kalian ingin membuat komputer menampilkan teks *Halo Dunia!* di layar menggunakan **Python**. Bagaimana perintahnya?

#### Pseudocode:

```
print("Halo Dunia!")
```

#### Jawaban & Pembahasan:

- `print()` adalah perintah di Python untuk menampilkan tulisan.
- Jika kita jalankan, layar akan menampilkan: **Halo Dunia!**
- Gampang banget, kan? Seperti ngobrol sama komputer.

### Soal 2: Hitung Rata-Rata Nilai

#### Soal:

Ada 3 nilai ujian: 80, 90, 70. Buat program Python untuk menghitung rata-ratanya.

#### Pseudocode:

```
nilai = [80, 90, 70]
rata2 = sum(nilai) / len(nilai)
print(rata2)
```

#### Jawaban & Pembahasan:

- `sum(nilai)` menjumlahkan semua angka.
- `len(nilai)` menghitung berapa jumlah data.
- Jadi:  $(80 + 90 + 70) / 3 = 80.0$
- Python membuat hitung rata-rata jadi mudah.

### Soal 3: Data Tabel dengan *Pandas*

#### Soal:

Kalian punya data jumlah jajan teman:Gunakan *Pandas* untuk menampilkan siapa yang jajan paling banyak.

| Nama | Jajan |
|------|-------|
| Budi | 10000 |
| Siti | 15000 |
| Andi | 12000 |

#### Pseudocode:

```
import pandas as pd

data = {"Nama": ["Budi", "Siti", "Andi"],
        "Jajan": [10000, 15000, 12000]}
df = pd.DataFrame(data)

print(df.loc[df["Jajan"].idxmax()])
```

#### Jawaban & Pembahasan:

- `DataFrame` adalah tabel di *Pandas*.
- `idxmax()` mencari nilai terbesar.
- Hasilnya: **Siti** dengan jajan 15000.

## Soal 4: AI Tebak Hewan

### Soal:

Kalian ingin bikin AI sederhana yang bisa tebak apakah gambar itu *kucing* atau *anjing*. Apa konsep AI yang digunakan?

### Pseudocode (alur sederhana):

```
gambar = input("Masukkan gambar: ")

jika gambar punya kumis panjang:
    print("Kucing")
else:
    print("Anjing")
```

### Jawaban & Pembahasan:

- Program ini belajar dari banyak contoh gambar.
- Konsepnya adalah *Machine Learning* dan *Neural Network*.
- AI belajar pola (misalnya kumis panjang → kucing).
- Jadi ini termasuk **AI**, bukan sekadar coding manual.

## Soal 5: Cek Stok Makanan

### Soal:

Kalian jualan minuman. Stok gula = 1, teh = 5, susu = 0. Buat program Python untuk cek stok yang hampir habis (<2).

### Pseudocode:

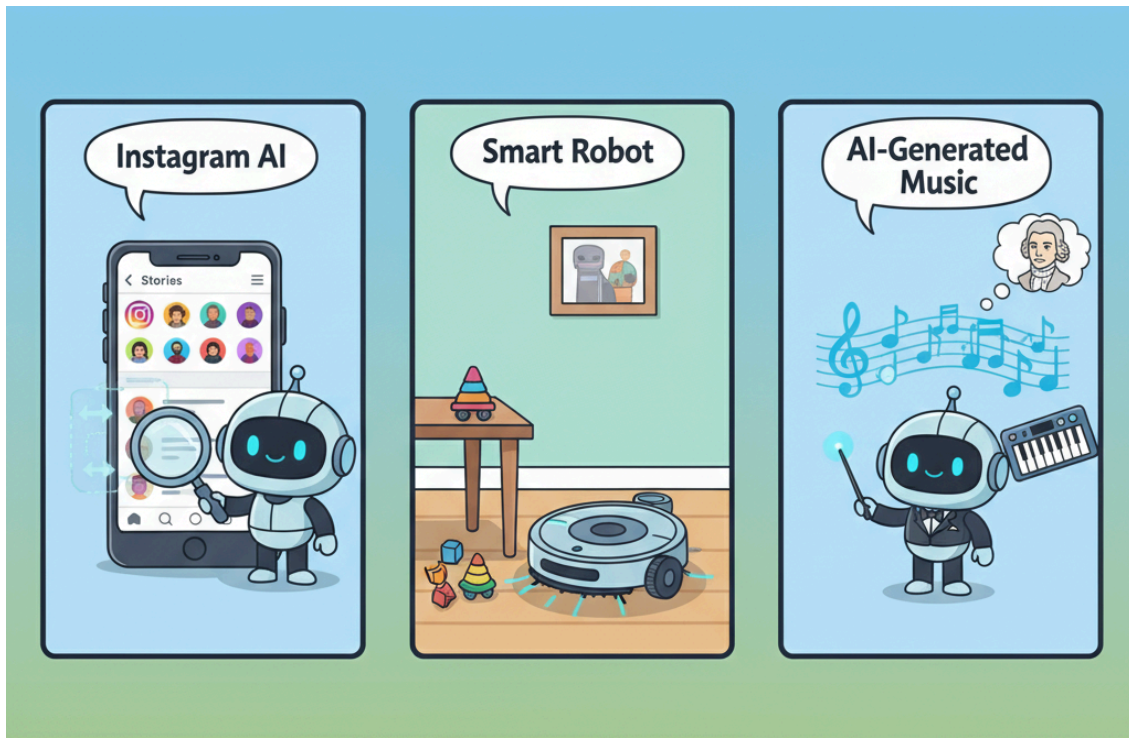
```
stok = {"gula": 1, "teh": 5, "susu": 0}

for bahan, jumlah in stok.items():
    if jumlah < 2:
        print(bahan, "hampir habis!")
```

### Jawaban & Pembahasan:

- `stok.items()` melihat data satu per satu.
- `if jumlah < 2` berarti kalau stok kurang dari 2.
- Hasil: **gula hampir habis!, susu hampir habis!**
- Dengan Python, usaha kecil bisa lebih rapi.

## Fun Facts!



- **Punya Instagram?** Algoritma yang mengatur siapa yang kalian lihat di Instagram Stories itu pakai AI, lho!
- **Robot Cerdas:** Robot yang bisa bersihkan rumah, seperti Roomba, juga pakai AI untuk navigasi dan menghindari rintangan.
- **Musik Buatan AI:** Ada AI yang bisa menciptakan musik baru, bahkan dengan gaya seperti komposer terkenal, lho!

## Tips Belajar dan Menghafal Cepat

- **Jangan cuma baca!** Coba langsung praktik. Coding itu kayak main game, harus sering latihan biar jago.
- **Bikin catatan sendiri.** Catat apa yang kalian anggap penting pakai bahasa kalian sendiri.
- **Ajak teman.** Belajar bareng lebih seru! Kalian bisa saling tanya jawab atau diskusi kalau ada yang nggak ngerti.
- **Lihat video.** Banyak video tutorial coding dan AI yang keren di YouTube.

## Aktivitas Siswa

### Aktivitas 1: Ngobrol dengan Komputer (Dasar Python)

Kita bikin program yang menyapa dan menghitung umur.

Tujuan: paham *input-output*, variabel, dan *print*.

#### Langkah:

- Buka editor: VS Code / Thonny (gratis).
- Buat file `halo.py`.
- Ketik kode.
- Jalankan.

#### Pseudocode:

```
minta nama dari pengguna
minta tahun_lahir
umur = tahun_sekarang - tahun_lahir
tampilkan "Halo, {nama}"
tampilkan "Umurmu {umur} tahun"
```

#### Contoh Python:

```
# halo.py
nama = input("Siapa namamu? ")
tahun = int(input("Tahun lahirmu? "))
tahun_sekarang = 2025
umur = tahun_sekarang - tahun
print("Halo,", nama)
print("Umurmu", umur, "tahun")
```

#### Tantangan:

- Ubah supaya tahun sekarang otomatis dari sistem (pakai `datetime`).
- Tambah *emoji* di output 😊

### Aktivitas 2: Statistik Jajan Kelas (Pandas)

Kita olah data kecil seperti di *Excel*, tapi pakai **Pandas**.

Tujuan: buat tabel, hitung rata-rata, cari yang terbesar.

#### Langkah:

Buat file `jajan.csv` berisi:

```
Nama,Jajan
Budi,10000
Siti,15000
```

Andi,12000  
Rani,8000  
Dika,20000

Pasang *library* **pandas**: `pip install pandas`.  
Jalankan skrip.

#### Pseudocode:

```
baca file csv ke tabel  
hitung rata_rata kolom Jajan  
cari baris dengan Jajan terbesar  
tampilkan hasil
```

#### Contoh Python:

```
import pandas as pd  
  
df = pd.read_csv("jajan.csv")  
rata = df["Jajan"].mean()  
terbanyak = df.loc[df["Jajan"].idxmax()]  
  
print("Rata-rata jajan:", int(rata))  
print("Paling banyak:", terbanyak["Nama"],  
      int(terbanyak["Jajan"]))
```

#### Tantangan:

- Urutkan dari terbesar ke terkecil.
- Simpan hasil ke `ringkasan.csv`.

### Aktivitas 3: Deteksi Nilai Lulus (Mini *Machine Learning* dengan Keras)

Kita latih *model* kecil untuk menebak **Lulus** (1) atau **Tidak** (0) dari dua fitur: **Nilai Tugas** dan **Kehadiran**.

Tujuan: kenal *dataset*, *training*, *predict*, *Neural Network* sederhana.

#### Langkah:

- Pasang **TensorFlow/Keras**: `pip install tensorflow` (CPU).
- Siapkan data kecil di kode.
- Latih *model*.
- Coba prediksi data baru.

#### Pseudocode:

```
buat dataset kecil (fitur, label)  
bangun model neural network sederhana  
latih model  
uji prediksi pada data baru
```

tampilkan hasil

### Contoh Python (Keras NN mini):

```
# nn_lulus.py
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

# Data sederhana: [nilai_tugas, kehadiran], label: 1=lulus,
0=tidak
X = np.array([
    [90, 95], [85, 90], [70, 80], [60, 60],
    [50, 70], [95, 85], [40, 50], [88, 92]
], dtype="float32")
y = np.array([1, 1, 1, 0, 0, 1, 0, 1], dtype="float32")

# Bangun *neural network* kecil
model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])

model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
model.fit(X, y, epochs=200, verbose=0)

# Coba prediksi
uji = np.array([[75, 85], [55, 60]], dtype="float32")
pred = model.predict(uji, verbose=0)
print("Prediksi probabilitas lulus:", pred.flatten())
print("Label akhir (>=0.5 jadi 1):", (pred.flatten() >=
0.5).astype(int))
```

### Penjelasan singkat:

- Ini *Machine Learning* dengan *Neural Network*.
- *Training* membuat *model* belajar pola.
- Output  $\geq 0.5 \rightarrow$  **Lulus (1)**.

### Tantangan:

- Tambah fitur **UTS** dan **UAS**.
- Bandingkan akurasi dengan *Decision Tree* (*scikit-learn*).

## Aktivitas 4: Warung Pintar: Cek Stok & Notifikasi (Python + Pandas)

Kita cek stok bahan. Kalau stok  $< 2$ , kasih peringatan.

Tujuan: *loop*, *if*, baca CSV, logika bisnis.

### Langkah:

Buat `stok.csv`:

```
Bahan,Qty
gula,1
teh,5
susu,0
sirup,3
kopi,1
```

Jalankan skrip.

### Pseudocode:

```
baca stok.csv ke tabel
untuk tiap baris:
    jika Qty < 2: catat sebagai hampir_habis
tampilkan daftar hampir_habis
```

### Contoh Python:

```
import pandas as pd

df = pd.read_csv("stok.csv")
hampir_habis = df[df["Qty"] < 2]

if len(hampir_habis) > 0:
    print("Peringatan! Bahan hampir habis:")
    for _, row in hampir_habis.iterrows():
        print("-", row["Bahan"], "sisanya", row["Qty"])
else:
    print("Semua stok aman.")
```

### Tantangan:

- Simpan daftar peringatan ke `peringatan.txt`.
- Tambah fitur: hitung total nilai stok (butuh kolom harga).

## Aktivitas 5: Asisten Dokumen Lokal (Mini RAG dengan Ollama)

Kita tanya jawaban dari catatan sendiri. *RAG = Retrieval-Augmented Generation*.  
Tujuan: konsep cari dokumen → kirim ke *model* lokal (semua *open source*).

### Langkah Setup (ringkas):

- Pasang **Ollama** (Linux/Mac/Win).
- Jalankan *model* lokal, misal `llama3` (gratis).
- Simpan catatan di folder `catatan/` (txt/pdf → ubah ke txt dulu).

- Buat skrip pencari teks sederhana + panggilan API Ollama.

### Pseudocode:

```
kumpulkan semua file .txt di folder
cari paragraf yang mirip dengan pertanyaan (cocokkan kata kunci)
gabungkan paragraf terbaik
kirim ke API ollama bersama pertanyaan
tampilkan jawaban
```

### Contoh Python (sederhana, *keyword match* + panggil Ollama REST):

```
# rag_mini.py
import os, re, json
import requests

def load_docs(folder="catatan"):
    docs = []
    for root, _, files in os.walk(folder):
        for f in files:
            if f.lower().endswith(".txt"):
                path = os.path.join(root, f)
                with open(path, "r", encoding="utf-8",
errors="ignore") as fh:
                    docs.append((path, fh.read()))
    return docs

def retrieve(query, docs, top_k=3):
    # Sangat sederhana: skor = jumlah kata kunci yang cocok
    terms = re.findall(r"\w+", query.lower())
    scored = []
    for path, text in docs:
        text_low = text.lower()
        score = sum(t in text_low for t in terms)
        scored.append((score, path, text))
    scored.sort(reverse=True, key=lambda x: x[0])
    return scored[:top_k]

def ask_ollama(model, prompt):
    url = "http://localhost:11434/api/generate"
    payload = {"model": model, "prompt": prompt}
    r = requests.post(url, json=payload, timeout=120)
    r.raise_for_status()
    # Ollama stream, ambil bagian "response"
    out = ""
    for line in r.text.splitlines():
        try:
            obj = json.loads(line)
            out += obj.get("response", "")
        except:
            pass
    return out.strip()
```

```

if __name__ == "__main__":
    pertanyaan = input("Tanya apa? ")
    docs = load_docs("catatan")
    top = retrieve(pertanyaan, docs)

    konteks = "\n\n".join([f"[{i+1}] {p}\n{text[:700]}" for i,
        (_, p, text) in enumerate(top)])
    prompt = f"""
Jawab singkat dan jelas. Gunakan konteks jika relevan.

Pertanyaan: {pertanyaan}

Konteks:
{konteks}
"""
    jawaban = ask_ollama("llama3", prompt)
    print("\n=== Jawaban Asisten Lokal ===\n", jawaban)

```

#### Catatan:

- Ini *RAG* sederhana (berbasis kata kunci).
- Tingkatkan nanti: pakai *embeddings* (misal **sentence-transformers**), *FAISS* (juga *open source*).

#### Tantangan:

- Batasi jawaban maksimal 5 kalimat.
- Tambah *highlight* sumber dokumen terbaik.

## Rangkuman Singkat

**Coding itu bahasa untuk komputer.** Kita memberi perintah. Komputer menjalankan. Mudah dan seru. Kita pakai *open source* seperti *Python* dan *Pandas*. Dengan itu, kita bisa hitung data, bikin grafik, dan mengatur file. **AI itu teman pintar.** AI belajar dari *dataset*. Ia pakai *machine learning*, *Keras*, dan *neural network*. Hasilnya? Komputer bisa mengenali gambar, menebak nilai, atau memberi rekomendasi.

**Semua toolnya gratis dan terbuka.** Itu keunggulan *open source*. *Python*, *Pandas*, *Keras*, sampai *Ollama* dan *OpenWebUI* bisa dipakai tanpa biaya. Banyak komunitas yang bantu. Dokumentasi jelas. **Belajar jadi cepat.** Kalian bisa mulai dari contoh kecil. Lalu naik level pelan-pelan. Dari *print("Halo Dunia!")* ke model AI mini. Dari tabel sederhana ke analisis kelas.

**Belajar terbaik adalah praktik.** Kerjakan proyek kecil tiap hari. Campur teori + latihan. Uji, salah, *debug*, perbaiki. **Bangun kebiasaan main kode.** Catat ide. Coba hal baru. Pakai *dataset* sendiri. Jadikan tugas sekolah sebagai *project-based learning*. Dengan *machine learning*, *Keras*, dan *neural network*, kalian siap membuat karya nyata. Siap, tim SMK?

# Latihan Soal

## A. Benar atau Salah (5 Soal)

Kalimatnya singkat. Pilih “Benar” atau “Salah”.

1. *Coding* adalah cara kita memberi perintah ke komputer.  
Jawaban: Benar
2. *AI* selalu butuh perintah langkah-per-langkah tanpa belajar dari data.  
Jawaban: Salah
3. **Python** bisa menampilkan teks dengan perintah *print*.  
Jawaban: Benar
4. **Pandas** dipakai untuk menggambar 3D game.  
Jawaban: Salah
5. *Open source* artinya kode dapat dipakai dan dimodifikasi bebas.  
Jawaban: Benar

## B. Pilihan Ganda (10 Soal)

Pilih A/B/C/D.

1. Perintah Python untuk menampilkan “Halo Dunia!” adalah...  
A. say("Halo Dunia!")  
B. echo("Halo Dunia!")  
C. print("Halo Dunia!")  
D. out("Halo Dunia!")  
Jawaban: C
2. *Library* Python untuk tabel data adalah...  
A. numpy  
B. pandas  
C. flask  
D. pygame  
Jawaban: B
3. *AI* belajar dari...  
A. wallpaper  
B. *dataset*  
C. password  
D. captcha  
Jawaban: B

4. *Neural network* termasuk bagian dari...
- A. *machine learning*
  - B. desain UI
  - C. kompresi zip
  - D. jaringan komputer
- Jawaban: A
5. tool yang dapat dijalankan lokal dan *open source* untuk *chat model* adalah...
- A. Photoshop
  - B. Ollama
  - C. Excel
  - D. Premiere
- Jawaban: B
6. Untuk membaca CSV di **Pandas**, kita pakai...
- A. `pd.read_text()`
  - B. `pd.read_file()`
  - C. `pd.read_csv()`
  - D. `pd.csv_open()`
- Jawaban: C
7. Aktivasi umum di *neural network* untuk klasifikasi biner pada output adalah...
- A. `relu`
  - B. `tanh`
  - C. `softmax`
  - D. `sigmoid`
- Jawaban: D
8. Baris Python yang benar untuk menyimpan rata-rata ke variabel `r`:
- A. `r = mean(nilai)`
  - B. `r = nilai.mean()`
  - C. `mean = r(nilai)`
  - D. `r = avg(nilai)`
- Jawaban: B
9. Yang **bukan** alasan memakai *open source* adalah...
- A. gratis untuk belajar
  - B. bisa dimodifikasi
  - C. komunitas besar
  - D. kode tertutup dan rahasia
- Jawaban: D

10. Potongan kode berikut akan menampilkan apa?

```
x = [80, 90, 70]
print(sum(x) / len(x))
```

- A. 70
- B. 80.0
- C. 240
- D. Error

Jawaban: B

### C. Isian Singkat (5 Soal)

Isi dengan kata/angka/kode yang tepat.

1. Perintah Python untuk mencetak teks ke layar adalah \_\_\_\_\_.

Jawaban: `print()`

2. *AI* belajar dari kumpulan data yang disebut \_\_\_\_\_.

Jawaban: *dataset*

3. *Framework open source* untuk membangun *neural network* di Python adalah \_\_\_\_\_.

Jawaban: Keras

4. *Library* Python untuk tabel dan kolom data adalah \_\_\_\_\_.

Jawaban: Pandas

5. Hasil dari `sum([2, 3, 5])` adalah \_\_\_\_\_.

Jawaban: 10

### D. Soal Explorasi (3 Soal)

Tulis jawaban di kertas/kode singkat. Ada kunci contoh jawaban.

**Soal 1:**

#### **Susun Data Jajan Mini dan Cari Terbesar**

Buat tabel kecil berisi 4 siswa dan kolom "Jajan". Tampilkan siapa yang jajan paling banyak dengan **Pandas**.

### Kunci jawaban (contoh):

```
import pandas as pd
df = pd.DataFrame({"Nama":["Ani","Budi","Citra","Dio"],
                  "Jajan":[8000,12000,7000,15000]})
p = df.loc[df["Jajan"].idxmax()]
print(p["Nama"], int(p["Jajan"]))
```

### Soal 2:

#### Pseudocode Alarm Stok

Buat *pseudocode* untuk cek stok bahan. Jika kurang dari 2, tampilkan “hampir habis”.

### Kunci jawaban (contoh):

```
baca tabel stok
untuk setiap baris:
    jika Qty < 2:
        tampilkan nama_bahan + " hampir habis"
```

### Soal 3:

#### Klasifikasi Mini: Lulus/Tidak dengan Keras

Buat *model* kecil dengan 2 fitur: nilai\_tugas dan kehadiran. Cetak prediksi untuk dua data baru.

### Kunci jawaban (contoh):

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

X = np.array([[90,95],[70,80],[60,60],[85,90]], dtype="float32")
y = np.array([1,1,0,1], dtype="float32")

model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam", loss="binary_crossentropy")
model.fit(X, y, epochs=200, verbose=0)

uji = np.array([[75,85],[55,60]], dtype="float32")
print((model.predict(uji, verbose=0).flatten() >=
0.5).astype(int))
```

# Tugas Proyek (5 Soal)

## Proyek 1: Kartu Nama Digital (Dasar Python)

Bikin program yang menyapa, hitung umur, dan cetak “kartu nama” sederhana.  
Fokus: *input*, *output*, variabel, logika.

### Langkah/Tahap:

- Buat file `kartu_nama.py`.
- Minta nama dan tahun lahir.
- Hitung umur.
- Cetak kartu nama di layar.

### Pseudocode:

```
minta nama
minta tahun_lahir
umur = tahun_sekarang - tahun_lahir
tampilkan garis
tampilkan "Nama: {nama}"
tampilkan "Umur: {umur}"
tampilkan hobi (opsional)
```

### Contoh Python:

```
nama = input("Namamu? ")
tahun = int(input("Tahun lahir? "))
from datetime import datetime
umur = datetime.now().year - tahun

print("="*20)
print("Nama :", nama)
print("Umur :", umur, "tahun")
print("="*20)
```

### Kunci Jawaban (apa yang benar):

- Program meminta *input*.
- Menghitung umur dengan benar.
- Menampilkan format rapi.
- Bonus: validasi jika tahun > tahun sekarang.

## Proyek 2: Statistik Jajan Kelas (Pandas Dasar)

Kita olah data jajan teman-teman. Cari rata-rata dan siapa paling banyak.  
Fokus: *dataset*, tabel, ringkasan.

### Langkah/Tahap:

Buat `jajan.csv`:

```
Nama,Jajan
Budi,10000
Siti,15000
Andi,12000
Rani,8000
Dika,20000
```

- Install **pandas**: `pip install pandas`.
- Baca CSV, hitung rata-rata, cari terbesar.

### Pseudocode:

```
baca csv ke tabel
rata = mean kolom Jajan
max_row = baris dengan Jajan terbesar
tampilkan rata dan nama max_row
```

### Contoh Python:

```
import pandas as pd
df = pd.read_csv("jajan.csv")

rata = df["Jajan"].mean()
terbanyak = df.loc[df["Jajan"].idxmax()]

print("Rata-rata:", int(rata))
print("Paling banyak:", terbanyak["Nama"],
      int(terbanyak["Jajan"]))
```

### Kunci Jawaban:

- Output menampilkan **rata-rata** dan **nama dengan jajan terbesar**.
- Bonus: urutkan menurun dan simpan ke `ringkasan.csv`.

### Proyek 3: Prediksi Lulus Mini (Keras + Neural Network)

Latih *neural network* kecil untuk tebak **Lulus (1)** atau **Tidak (0)** dari dua fitur: **Nilai Tugas** dan **Kehadiran**.

Fokus: *machine learning*, *Keras*, *training*, *predict*.

#### Langkah/Tahap:

- Install **TensorFlow/Keras**: `pip install tensorflow`.
- Siapkan data kecil di kode.
- Bangun model sederhana.
- Latih dan uji prediksi.

#### Pseudocode:

```
X = data fitur
y = label (0/1)
bangun model: Dense -> Dense(sigmoid)
compile (adam, binary_crossentropy)
fit beberapa epoch
prediksi contoh baru
```

#### Contoh Python:

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

X =
np.array([[90,95],[85,90],[70,80],[60,60],[50,70],[95,85],[40,50],
[88,92]], dtype="float32")
y = np.array([1,1,1,0,0,1,0,1], dtype="float32")

model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
model.fit(X, y, epochs=200, verbose=0)

uji = np.array([[75,85],[55,60]], dtype="float32")
pred = model.predict(uji, verbose=0).flatten()
label = (pred >= 0.5).astype(int)
print("Prob:", pred)
print("Label:", label)
```

#### Kunci Jawaban:

- Model *compile* dan *fit* tanpa error.
- Prediksi keluar sebagai probabilitas 0–1.

- Konversi  $\geq 0.5 \rightarrow 1$  (Lulus).

## Proyek 4: Alarm Stok Kantin (Python + Pandas)

Cek stok bahan. Jika  $< 2$ , kasih alarm.

Fokus: *if*, *loop*, baca CSV.

### Langkah/Tahap:

Buat `stok.csv`:

```
Bahan,Qty
gula,1
teh,5
susu,0
sirup,3
kopi,1
```

- Baca CSV, filter yang kurang.

### Pseudocode:

```
baca csv
hampir_habis = baris Qty < 2
jika ada:
    tampilkan daftar
lainnya:
    tampilkan "aman"
```

### Contoh Python:

```
import pandas as pd
df = pd.read_csv("stok.csv")
kritikal = df[df["Qty"] < 2]

if len(kritikal) > 0:
    print("ALARM! Hampir habis:")
    for _, r in kritikal.iterrows():
        print("-", r["Bahan"], "sisanya", r["Qty"])
else:
    print("Stok aman.")
```

### Kunci Jawaban:

- Program menampilkan **daftar bahan kritikal**.
- Bonus: simpan ke `peringatan.txt` dan hitung total nilai stok (butuh kolom Harga).

## Proyek 5: Klasifikasi Mood Sederhana (Keras—Teks Mini)

Model menebak *mood* kalimat: **senang (1)** atau **tidak (0)**.

Fokus: *feature engineering* sederhana, *neural network* mini.

### Langkah/Tahap:

- Buat fitur angka: berapa kali kata “senang”, “bahagia”, “seru”.
- Data kecil, latih *neural network* 1 *hidden layer*.
- Uji 2 kalimat baru.

### Pseudocode:

```
buat fungsi fitur(teks):  
    hitung kata positif  
X = fitur dari daftar kalimat latih  
y = label 0/1  
bangun model Dense->Dense(sigmoid)  
latih, lalu prediksi kalimat baru
```

### Contoh Python:

```
import numpy as np  
from tensorflow import keras  
from tensorflow.keras import layers  
  
positif = ["senang", "bahagia", "seru", "keren", "asyik"]  
  
def fitur(teks):  
    t = teks.lower().split()  
    hit = sum(w in t for w in positif)  
    panjang = len(t)  
    prop = hit / max(1, panjang)  
    return [hit, panjang, prop]  
  
kalimat = [  
    "hari ini seru dan asyik",          # 1  
    "tugas banyak tapi keren",         # 1  
    "aku sedih lelah capek",           # 0  
    "biasa saja tidak seru",           # 0  
    "kegiatan sekolah sangat bahagia", # 1  
    "cuaca buruk dan membosankan"     # 0  
]  
y = np.array([1,1,0,0,1,0], dtype="float32")  
X = np.array([fitur(k) for k in kalimat], dtype="float32")  
  
model = keras.Sequential([  
    layers.Input(shape=(3,)),  
    layers.Dense(6, activation="relu"),  
    layers.Dense(1, activation="sigmoid")  
])
```

```

model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
model.fit(X, y, epochs=300, verbose=0)

uji = ["kelas hari ini asyik dan keren", "aku capek tidak
semangat"]
Xu = np.array([fitur(k) for k in uji], dtype="float32")
pred = model.predict(Xu, verbose=0).flatten()
print(list(zip(uji, (pred>=0.5).astype(int), pred)))

```

#### Kunci Jawaban:

- Fitur dihitung benar (hit kata positif, panjang, proporsi).
- Model belajar dan memberi label **1 = senang, 0 = tidak**.
- Catatan: ini *demo* sederhana; nyata-nya butuh *dataset* besar.

#### Catatan Umum

- Semua tool yang dipakai **open source**: *Python*, **Pandas**, **TensorFlow/Keras**.
- Istilah penting: *open source*, *dataset*, *machine learning*, *neural network*, *input*, *output*, *feature*.
- **Tips**: mulai kecil, sering coba, *debug*, lalu tingkatkan. Mau saya jadikan versi **lembar kerja (worksheet)** siap cetak, atau **notebook Jupyter** siap jalan?

## BAB 2: Dasar Computational Thinking

Gimana kabarnya? Semoga selalu semangat ya! Hari ini kita mau belajar sesuatu yang seru banget, namanya **Computational Thinking** atau **Berpikir Komputasi**. Jangan panik dulu denger namanya yang keren, ini tuh sebenarnya gampang banget dan kita sering pakai tanpa sadar lho! Yuk, kita mulai! ✨

### Tujuan Pembelajaran

Setelah belajar bareng, kamu bakal bisa:

- **Paham apa itu Computational Thinking.**
- **Menggunakan Computational Thinking buat memecah masalah sehari-hari.**
- **Tahu kalau berpikir komputasi itu penting buat masa depan kamu.**

### Peta Konsep Sederhana

Bayangin lagi main game atau masak. Ada langkah-langkahnya kan? Nah, Computational Thinking juga punya 4 langkah utama yang harus kita lewati:

1. **Dekomposisi (Decomposition):** Memecah masalah besar jadi bagian-bagian kecil.
2. **Pengenalan Pola (Pattern Recognition):** Mencari pola atau kesamaan dari bagian-bagian kecil itu.
3. **Abstraksi (Abstraction):** Menyaring informasi penting dan buang yang gak penting.
4. **Algoritma (Algoritma):** Membuat langkah-langkah penyelesaian yang jelas.

### Apersepsi: Kenapa Handphone Kamu Sering Lemot?

Pernah gak lagi asyik main game, tiba-tiba handphone kamu macet? 😞 Kenapa ya?

Nah, bayangin handphone itu kayak kamu yang lagi mengerjakan banyak tugas. Kalau kamu mengerjakan semua tugas sekaligus, pasti pusing dan gak fokus kan? Begitu juga handphone. Dia butuh "berpikir" buat ngurusin semua aplikasi yang kamu buka. Kalau aplikasinya kebanyakan, dia bingung dan akhirnya jadi lemot.

Nah, dengan Computational Thinking, kita bakal ajarin "cara berpikir" yang lebih rapi biar gak lemot. Yuk, kita mulai!

## Penjelasan Konsep (Teori)

### Apa itu *Computational Thinking*?

Bayangkan kamu lagi main game atau masak. Ada langkah. Ada urutan. Ada strategi. Itulah *computational thinking*:

- Cara berpikir untuk memecah masalah.
- Cari pola.
- Fokus yang penting.
- Tulis langkah jelas.

Empat pilar utamanya: **Dekomposisi, Pengenalan Pola, Abstraksi, Algoritma.**

Kenapa penting?

- Bikin kamu cepat paham masalah.
- Ngoding jadi rapi.
- Mudah *debugging*.
- Siap ke dunia *AI* dan data.

### Empat Pilar, dengan contoh yang dekat

Dekomposisi (Decomposition) — “Bagi yang besar jadi kecil”

Masalah besar bikin pusing. Pecah jadi bagian kecil. Contoh: **nasi goreng**.

- Siapkan bahan.
- Panaskan wajan.
- Tumis bumbu.
- Masukkan nasi.
- Bumbui. Sajikan.

Pengenalan Pola (Pattern Recognition) — “Cari yang sama”

Lihat kesamaan antar kasus. Nasi goreng mirip mie goreng: sama-sama wajan, minyak, tumis bumbu.

Di data, pola adalah teman baik *AI*. *Neural network* belajar dari pola.

Contoh mini dengan **Pandas** (pola jajan teman):

```
import pandas as pd
df = pd.DataFrame({"Nama": ["A", "B", "C", "D"],
                  "Jajan": [8000, 12000, 7000]})
print("Rata-rata:", df["Jajan"].mean())
print("Terbanyak:", df.loc[df["Jajan"].idxmax(), "Nama"])
```

## Abstraksi (*Abstraction*) — “Fokus ke inti”

Ambil yang penting. Buang gangguan.

Robot siram tanaman butuh sensor tanah, pompa air, alur nyala-mati. Warna robot? Stiker? Tidak penting untuk fungsi inti.

*Checklist cepat:*

- Penting: tujuan, input, output, batasan.
- Tidak penting: dekorasi, gaya, efek visual (awal belajar).

## Algoritma (Algorithm) — “Langkah jelas, berurutan”

Algoritma itu resep. Harus spesifik,urut, dan bisa diulang.

*Pseudocode beli minum di kantin:*

```
masuk kantin
lihat menu
pilih minuman
bayar
ambil minuman
```

*Contoh **Python** sederhana:*

```
pilihan = ["Teh", "Jus", "Susu"]
for i, p in enumerate(pilihan, 1):
    print(i, p)
idx = int(input("Pilih nomor: "))
print("Kamu memilih:", pilihan[idx-1])
```

## Dari konsep ke kode: jalur *open source*

Kamu bisa mulai dari empat pilar, lalu lanjut ke tool gratis:

- **Python**: bahasa yang mudah dibaca.
- **Pandas**: olah tabel dan data.
- *Machine learning* + **Keras**: bangun *neural network* mini.
- Semua *open source*, banyak contoh, komunitas besar.

*Alur berpikir → ke praktik AI:*

1. **Dekomposisi**: pecah proyek (kumpul data, bersih, latih model, uji).
2. **Pola**: cari fitur yang terkait tujuan (mis. kata “senang” untuk mood).
3. **Abstraksi**: pilih fitur penting saja.
4. **Algoritma**: tulis langkah, lalu kodingkan.

*Contoh super mini model (konsep, bukan fokus akurasi):*

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

# Fitur sederhana: [nilai_tugas, kehadiran]
X = np.array([[90,95],[70,80],[60,60],[85,90]], dtype="float32")
y = np.array([1,1,0,1], dtype="float32") # 1=lulus, 0=tidak

model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam", loss="binary_crossentropy")
model.fit(X, y, epochs=200, verbose=0)

uji = np.array([[75,85],[55,60]], dtype="float32")
print((model.predict(uji, verbose=0).flatten() >= 0.5).astype(int))
```

Empat pilar tetap memandu: pecah tugas, melihat pola data, ambil fitur inti, susun langkah latihan-prediksi.

## Tips cepat biar makin nempel

- Tulis masalahmu. Pecah jadi 3–7 langkah kecil.
- Cari pola: apa yang berulang? apa yang sama?
- Coret yang tidak penting. Fokus tujuan.
- Tulis *algorithm* pendek. Uji. Perbaiki. Ulangi.
- Simpan contoh kecil di folder. Bangun *toolbox* belajarmu.

## Mini glosarium (biar mantap)

- *Computational thinking*: cara berpikir untuk memecah, mengenali pola, menyaring inti, dan menyusun langkah.
- *Decomposition*: membagi masalah besar jadi bagian kecil.
- *Pattern recognition*: menemukan kesamaan/pola.
- *Abstraction*: fokus pada inti, abaikan yang tidak relevan.
- *Algorithm*: urutan langkah yang jelas.
- *Open source*: kode terbuka, bebas dipakai dan diubah.
- *Machine learning, Keras, neural network*: tool dan teknik untuk membuat program yang belajar dari data.

Intinya: **pikir rapi** → **tulis langkah** → **uji di kode**. Dengan empat pilar *computational thinking*, kamu siap melangkah dari konsep ke praktik di **Python**, **Pandas**, dan dunia *AI*—semuanya *open source*!

## Contoh Kasus atau Ilustrasi

### Kasus 1: Beli Baju Online: Cepat & Pas

Kamu mau beli baju. Pilihan banyak. Bingung. Kita rapikan cara berpikir.

- **Dekomposisi:** toko → model → ukuran → warna → harga.
- **Pola:** toko A cenderung kasual; toko B cenderung formal.
- **Abstraksi:** yang penting *fit* (ukuran) + harga; abaikan detail hiasan.
- **Algoritma:** filter → urutkan → pilih.

#### Pseudocode:

```
masukan daftar_baju
filter ukuran == "M"
filter warna == "hitam"
urutkan berdasarkan harga naik
ambil 3 teratas
tampilkan
```

#### Python (opsional):

```
baju = [
    {"model": "kaos", "uk": "M", "warna": "hitam", "harga": 79000},
    {"model": "kemeja", "uk": "L", "warna": "hitam", "harga": 129000},
    {"model": "hoodie", "uk": "M", "warna": "hitam", "harga": 159000},
]
import pandas as pd
df = pd.DataFrame(baju)
print(df[(df.uk=="M") & (df.warna=="hitam")].sort_values("harga").
head(3))
```

### Kasus 2: Tugas Kelompok: Anti Drama, On-time

Tim kamu kebagian proyek presentasi. Waktu mepet.

- **Dekomposisi:** riset → slide → desain → latihan → presentasi.
- **Pola:** semua butuh tenggat dan *checkpoint*.
- **Abstraksi:** fokus siapa-mengerjakan-apa + deadline; abaikan dekor dulu.
- **Algoritma:** bagi tugas → atur deadline → cek progres → gabung.

### Pseudocode:

```
anggota = ["A", "B", "C"]
tugas = {"A": "riser", "B": "slide", "C": "desain"}
deadline = {"riser": "Rabu", "slide": "Kamis", "desain": "Kamis"}
setiap hari:
    cek progres tugas
    jika ada terlambat:
        bantu atau geser prioritas
Jumat: gabung semua, latihan
```

### Python (opsional):

```
import pandas as pd
df = pd.DataFrame([
    ["A", "riser", "Rabu", "progres 80%"],
    ["B", "slide", "Kamis", "progres 60%"],
    ["C", "desain", "Kamis", "progres 50%"],
], columns=["Nama", "Tugas", "Deadline", "Progres"])
print(df)
```

## Kasus 3: Cari File Hilang di Laptop: Kilat

File tugas "Tugas\_SMK\_Matematika\_NamaKamu" hilang. Panik? Tenang.

- **Dekomposisi:** cek Documents → Downloads → Desktop → *search*.
- **Pola:** nama file tugas punya pola "Tugas\_SMK\_\*".
- **Abstraksi:** yang penting ketemu file; lokasinya nomor dua.
- **Algoritma:** telusuri folder → cari nama pola → temukan terbaru.

### Pseudocode:

```
pola = "Tugas_SMK"
hasil = cari_semua_file_yang_mengandung(pola)
jika hasil kosong:
    tampilkan "tidak ditemukan"
lainnya:
    urutkan hasil berdasarkan tanggal_modifikasi turun
    tampilkan file_terbaru
```

### Python (opsional):

```
import glob, os
files = glob.glob("**/*Tugas_SMK*", recursive=True)
if not files:
    print("Tidak ditemukan")
else:
    files.sort(key=lambda f: os.path.getmtime(f), reverse=True)
    print("Terbaru:", files[0])
```

## Kasus 4: Konten Sosial: Siapkan Video yang “Nempel”

Mau agar video kamu ramai ditonton.

- **Dekomposisi:** ide → naskah → rekam → edit → unggah.
- **Pola:** *hook* 3–5 detik, teks pendek, musik tren.
- **Abstraksi:** fokus *hook* + pesan inti; efek mewah nomor belakang.
- **Algoritma:** pilih ide → tulis 5 kalimat → rekam → potong singkat → tambah teks.

### Pseudocode:

```
ide = pilih_tren_teratas()
hook = buat_kalimat_pembuka(<= 5 detik)
rekam_video(hook + isi + ajakan)
edit_potong_pendek()
tambahkan_teks_singkat()
unggah_dengan_hashtag_tren()
```

### Pandas (opsional, cek pola hashtag sederhana):

```
import pandas as pd
df =
pd.DataFrame({"hashtag":["#belajar", "#smk", "#tips", "#belajar", "#smk"]})
print(df["hashtag"].value_counts())
```

## Kasus 5: Debugging Python: Temukan & Perbaiki Cepat

Koding kamu error. Jangan emosi. Kita sistematis.

- **Dekomposisi:** baca pesan error → temukan baris → cek variabel → jalankan ulang.
- **Pola:** sering salah ketik nama variabel atau *indentation*.
- **Abstraksi:** fokus di bagian yang error; gaya tampilan abaikan dulu.
- **Algoritma:** isolasi baris bermasalah → perbaiki → tes unit kecil.

### Pseudocode:

```
jalankan program
jika error:
    catat jenis_error dan baris
    buka baris tersebut
    cek: nama_variabel, tanda_kutip, tanda_kurung, indentasi
    perbaiki
    jalankan ulang bagian kecil (unit test)
jika sudah jalan:
    lanjutkan ke modul berikutnya
```

## Upgrade ke AI — opsional untuk eksplorasi

- Dari Kasus 1: latih *model* rekomendasi sederhana (*machine learning*) dari *dataset* preferensi.
- Dari Kasus 4: klasifikasi “judul bagus” vs “biasa” pakai **Keras** dan *neural network* kecil (fitur: panjang teks, jumlah kata kunci).
- Semua bisa dikerjakan di **Python**; olah data dengan **Pandas**; *model* pakai **Keras**. Semua *open source*.

### Pseudocode AI mini (umum):

```
kumpulkan dataset
bagi data_latih dan data_uji
bangun model neural network sederhana
latih model pada data_latih
ukur akurasi di data_uji
jika akurasi cukup:
    pakai model untuk prediksi baru
else:
    perbaiki fitur atau tambah data
```

**Inti latihan:** pikir rapi (*decomposition*), cari *pattern*, ambil inti (*abstraction*), tulis langkah (*algorithm*). Lalu praktekkan di **Python/Pandas**. Kalau siap, naik kelas ke *machine learning* dengan **Keras** dan *neural network*. Semua serba *open source*!

## Soal Latihan dan Pembahasan

### Soal 1: Dekomposisi: Nasi Goreng Anti-Gagal

#### Soal:

Urutkan langkah membuat nasi goreng agar hasilnya enak. Ini daftar acak:

1. Masukkan nasi.
2. Sajikan.
3. Tumis bumbu.
4. Panaskan wajan.
5. Masukkan telur dan orak-arik.

#### Pseudocode (jawaban):

```
panaskan wajan
tumis bumbu
masukkan telur -> orak arik
masukkan nasi -> aduk rata
bumbui -> koreksi rasa
sajikan
```

**Jawaban singkat:**  $4 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow (\text{bumbui}) \rightarrow 2$ .

**Pembahasan ringkas:**

- Ini contoh **dekomposisi**: masalah besar dipecah jadi langkah kecil.
- Urutan penting. Kalau telurnya belakangan, bumbunya tidak rata.
- *Algorithm* yang jelas mencegah “nasi gosong”.

## Soal 2: Pengenalan Pola: Camilan Favorit Kelas

**Soal:**

Kamu kumpulkan data camilan favorit teman (5 orang). Tentukan camilan yang paling populer.

**Pseudocode:**

```
data = daftar camilan
hitung frekuensi tiap camilan
temukan camilan dengan frekuensi terbesar
tampilkan nama camilan + jumlahnya
```

**Python (opsional):**

```
import pandas as pd

df = pd.DataFrame({"Nama": ["Ani", "Budi", "Cici", "Dio", "Eka"],
                  "Camilan": ["Roti", "Ciki", "Roti", "Coklat", "Roti"]})
pop = df["Camilan"].value_counts().idxmax()
jumlah = df["Camilan"].value_counts().max()
print(pop, jumlah)
```

**Jawaban singkat:** Roti, 3 orang.

**Pembahasan ringkas:**

- Ini **pengenalan pola** (*pattern recognition*).
- Kita melihat pola yang sering muncul.
- **Pandas** memudahkan hitung frekuensi data tabel.

## Soal 3: Abstraksi: Robot Penyiram Tanaman

**Soal:**

Pilih komponen yang **wajib** untuk robot penyiram sederhana.

Pilihan: sensor kelembaban, pompa air, stiker lucu, warna robot, kabel, baterai.

**Pseudocode (jawaban):**

```
wajib = {sensor_kelembaban, pompa_air, kabel, baterai}
opsional = {stiker, warna}
tampilkan wajib
```

**Jawaban singkat:** Sensor kelembaban, pompa air, kabel, baterai.

**Pembahasan ringkas:**

- Ini **abstraksi**: fokus pada inti fungsi.
- Warna dan stiker tidak mempengaruhi penyiraman.
- Dengan *abstraction*, desain cepat dan efektif.

#### Soal 4: Algoritma: Beli Minum di Kantin

**Soal:**

Buat *algorithm* singkat untuk membeli minuman di kantin dengan antrian.

**Pseudocode (jawaban):**

```
masuk kantin
jika ada antrian:
    berdiri di belakang antrian
tunggu giliran
lihat menu -> pilih minuman
bayar -> ambil minuman
keluar kantin
```

**Jawaban singkat:** Langkah-langkah di atas (dengan percabangan antrian).

**Pembahasan ringkas:**

- Ini **algoritma**: urut, jelas, dan bisa diulang.
- Ada *branching* ("jika ada antrian").
- Tanpa langkah, kamu bisa memotong antrian (salah proses).

#### Soal 5: Debugging & Pola Error: Rata-Rata Nilai

**Soal:**

Kode berikut error. Perbaiki agar menampilkan rata-rata nilai.

```
nilai = [80, 90, 70]
totl = sum(nilai)
rata = total / len(nilai)
print("Rata:", rata)
```

**Pseudocode (cek & perbaiki):**

```
jalankan program
jika error -> baca pesan error
cek nama variabel yang konsisten
perbaiki penulisan 'total' -> 'totl' atau sebaliknya
jalankan ulang
```

**Python (jawaban benar):**

```
nilai = [80, 90, 70]
total = sum(nilai)
rata = total / len(nilai)
print("Rata:", rata)
```

**Jawaban singkat:** Perbaiki nama variabel agar konsisten (**total**). Output: 80.0.

**Pembahasan ringkas:**

- Ini *debugging* dengan **pola error** umum: salah ketik variabel.
- Baca pesan error → temukan baris → perbaiki → jalankan ulang.
- Langkah ini bagian dari *computational thinking*.

## Bonus Mini AI (opsional untuk penasaran)

**Konsep:** Klasifikasi *lulus/tidak* pakai *neural network* kecil (fitur: nilai tugas, kehadiran).

**Pseudocode:**

```
kumpulkan dataset kecil
bangun model dengan Keras
latih model
prediksi dua data baru
```

Tools yang dipakai *open source*: **Python**, **Pandas**, **Keras**. Saat siap, kamu bisa tingkatkan ke *machine learning* yang lebih nyata dengan *dataset* lebih besar.

## Inti belajar hari ini:

- Pecah masalah (**dekomposisi**).
- Cari yang berulang (**pola**).
- Ambil yang inti (**abstraksi**).
- Tulis langkah rapi (**algoritma**).

Lalu praktekkan di **Python**. Kalau lancar, lanjut eksplor AI dengan **Keras** dan *neural network*.

## Fun Facts

- **Computational Thinking** sudah ada sejak tahun 1980-an, lho! Tapi baru populer lagi belakangan ini.
- Kamu bisa pakai **Python** (gratis) buat bikin program, dari yang sederhana sampai yang canggih. Banyak perusahaan besar kayak Google dan Netflix pakai Python!
- **Ollama** itu seperti ChatGPT versi gratis yang bisa kamu jalankan di laptop sendiri. Keren kan?

## Tips Belajar atau Menghafal Cepat

1. **Pecah Materi:** Bagi materi yang susah jadi bagian-bagian kecil.
2. **Cari Pola:** Temukan konsep yang sering diulang-ulang.
3. **Buang yang Gak Penting:** Fokus pada inti materinya aja.
4. **Buat Algoritma:** Bikin langkah-langkah buat belajar atau ngerjain tugas.

## Aktivitas Siswa

### Aktivitas 1: Algoritma Kantin: Beli Minum Tanpa Drama

Tujuan: latih *decomposition* + *algorithm*.

#### Langkah:

- Tentukan tujuan: beli es teh.
- Pecah masalah: lihat menu → pilih → bayar → ambil.
- Tambah cabang: kalau ada antrian, tunggu.

#### Pseudocode:

```
masuk kantin
jika ada antrian: tunggu
lihat menu
pilih minuman
bayar
ambil minuman
keluar
```

#### Python:

```
menu = ["Es Teh", "Jus Jeruk", "Susu Coklat"]
print("Menu:")
for i, m in enumerate(menu, 1):
    print(i, m)

antrian = int(input("Jumlah orang di antrian? "))
if antrian > 0:
    print("Tunggu giliran...")

pilih = int(input("Pilih nomor minuman: "))
print("Kamu membeli:", menu[pilih-1])
print("Bayar -> Ambil -> Selesai!")
```

#### Jawaban (contoh output):

```
Kamu membeli: Es Teh
Bayar -> Ambil -> Selesai!
```

## Aktivitas 2: Cari Pola: Camilan Favorit dengan Pandas

Tujuan: latih *pattern recognition* di data tabel.

### Langkah:

- Buat data kecil (Nama, Camilan).
- Hitung frekuensi camilan.
- Temukan paling populer.

### Pseudocode:

```
baca data
hitung kemunculan tiap camilan
ambil camilan dengan frekuensi terbesar
tampilkan hasil
```

### Python:

```
import pandas as pd

df = pd.DataFrame({
    "Nama": ["Ani", "Budi", "Cici", "Dio", "Eka", "Fajar", "Gita"],
    "Camilan":
    ["Roti", "Ciki", "Roti", "Coklat", "Roti", "Ciki", "Roti"]
})
hit = df["Camilan"].value_counts()
populer = hit.idxmax()
print("Frekuensi:\n", hit)
print("Paling populer:", populer, "(", hit.max(), "siswa )")
```

### Jawaban (contoh output):

```
Paling populer: Roti ( 4 siswa )
```

## Aktivitas 3: Abstraksi & Alarm Stok Kantin

Tujuan: fokus pada yang penting (*abstraction*) + logika if.

### Langkah:

- Data stok: Bahan, Qty.
- Anggap *wajib* cek Qty saja (warna/merk abaikan).
- Jika Qty < 2 → “Hampir habis”.

### Pseudocode:

```
baca stok
```

```
untuk setiap bahan:
    jika qty < 2: catat sebagai hampir habis
tampilkan daftar hampir habis
```

### Python:

```
import pandas as pd
stok = pd.DataFrame({
    "Bahan": ["gula", "teh", "susu", "sirup", "kopi"],
    "Qty":   [1,      5,      0,      3,      1]
})
kritikal = stok[stok["Qty"] < 2]
if len(kritikal) == 0:
    print("Semua stok aman")
else:
    print("Hampir habis:")
    for _, r in kritikal.iterrows():
        print("-", r["Bahan"], "sisia", r["Qty"])
```

### Jawaban (contoh output):

```
Hampir habis:
- gula sisa 1
- susu sisa 0
- kopi sisa 1
```

## Aktivitas 4: Mini *Machine Learning*: Prediksi Lulus (Keras)

Tujuan: kenalan *machine learning*, **Keras**, *neural network*.

### Langkah:

- Fitur: [nilai\_tugas, kehadiran]. Label: 1=lulus, 0=tidak.
- Bangun *neural network* kecil.
- Latih → Prediksi data baru.

### Pseudocode:

```
siapkan X (fitur) dan y (label)
buat model: Dense(4,"relu") -> Dense(1,"sigmoid")
compile (adam, binary_crossentropy)
fit beberapa epoch
prediksi contoh baru
```

### Python (jalankan di Colab/lingkungan dengan TensorFlow):

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

X =
np.array([[90,95],[85,90],[70,80],[60,60],[50,70],[95,85],[40,50],[88,
92]], dtype="float32")
```

```

y = np.array([1,1,1,0,0,1,0,1], dtype="float32")

model = keras.Sequential([
    layers.Input(shape=(2,)),
    layers.Dense(4, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
model.fit(X, y, epochs=200, verbose=0)

uji = np.array([[75,85],[55,60]], dtype="float32")
prob = model.predict(uji, verbose=0).flatten()
pred = (prob >= 0.5).astype(int)
print("Prob:", prob)
print("Pred:", pred) # 1=lulus, 0=tidak

```

### Jawaban (contoh output):

```

Pred: [1 0]
Artinya data [75,85] → lulus, [55,60] → tidak (contoh).

```

## Aktivitas 5: Klasifikasi Mood Kalimat Mini (Keras)

Tujuan: *pattern recognition* pada teks + *neural network* mini.

### Langkah:

- Buat fitur sederhana: hitung kata positif.
- Bangun model kecil.
- Prediksi 2 kalimat baru.

### Pseudocode:

```

def fitur(teks): hitung_kata_positif, panjang, proporsi
X = fitur(kalimat_latih)
y = label 0/1
model Dense -> Dense(sigmoid)
latih
prediksi kalimat uji

```

### Python:

```

import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

positif = {"senang","bahagia","seru","keren","asyik"}

def fitur(teks):
    t = teks.lower().split()
    hit = sum(1 for w in t if w in positif)

```

```

    panjang = len(t)
    prop = hit / max(1, panjang)
    return [hit, panjang, prop]

kalimat = [
    "hari ini seru dan asyik",      # 1
    "tugas banyak tapi keren",     # 1
    "aku sedih lelah capek",       # 0
    "biasa saja tidak seru",       # 0
    "sekolah terasa bahagia",      # 1
    "cuaca buruk membosankan"     # 0
]
y = np.array([1,1,0,0,1,0], dtype="float32")
X = np.array([fitur(k) for k in kalimat], dtype="float32")

model = keras.Sequential([
    layers.Input(shape=(3,)),
    layers.Dense(6, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
model.compile(optimizer="adam", loss="binary_crossentropy")
model.fit(X, y, epochs=300, verbose=0)

uji = ["kelas hari ini asyik dan keren", "aku capek tidak
semangat"]
Xu = np.array([fitur(k) for k in uji], dtype="float32")
pred = (model.predict(Xu, verbose=0).flatten() >=
0.5).astype(int)
print(list(zip(uji, pred))) # 1=senang, 0=tidak

```

#### Jawaban (contoh output):

```

[('kelas hari ini asyik dan keren', 1), ('aku capek tidak
semangat', 0)]

```

### Ringkas & Mantap

- **Dekomposisi:** pecah tugas jadi kecil-kecil.
- **Pola:** lihat yang berulang di data.
- **Abstraksi:** fokus inti, singkirkan yang tidak penting.
- **Algoritma:** tulis langkah jelas.
- Lanjutkan ke *AI* mini dengan **Keras** dan *neural network*. Semua **open source**. Gas!

## Rangkuman

Berpikir komputasi itu cara pikir yang rapi dan terstruktur. Kita memecah masalah, fokus ke inti, lalu susun langkah. **Tujuannya: memudahkan hidup dan belajar.** Inilah yang disebut *computational thinking*. Dengan ini, kamu bisa **paham cepat, hemat waktu, dan lebih percaya diri** saat ngoding. Semua bisa dimulai dari tool *open source* seperti **Python** dan **Pandas**. Gratis. Mudah dicoba. Seru!

Empat pilar utamanya jelas. **Dekomposisi** (*decomposition*): bagi masalah besar jadi kecil. **Pengenalan Pola** (*pattern recognition*): cari kesamaan yang berulang. **Abstraksi** (*abstraction*): ambil yang penting, singkirkan yang ganggu. **Algoritma** (*algorithm*): tulis langkah yang urut dan bisa diulang. Contohnya sederhana: masak nasi goreng, beli minum di kantin, atau cari file di laptop. **Intinya: pecah → lihat pola → fokus inti → jalankan langkah.**

Setelah paham dasar, kamu bisa naik level. Olah data dengan **Pandas**. Lalu coba AI sederhana pakai *machine learning*, **Keras**, dan *neural network*. Semua **open source**, jadi kamu bebas belajar dan bereksperimen. **Kuncinya: praktik rutin, uji kecil, perbaiki, ulangi.** Mulai dari `print("Halo Dunia!")`, lanjut analisis data kecil, lalu buat model mini. Pelan tapi pasti, skill-mu akan naik kelas!

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. **Dekomposisi** adalah memecah masalah besar jadi bagian kecil.  
Jawaban: Benar
2. **Abstraksi** artinya menambahkan detail agar program makin panjang.  
Jawaban: Salah
3. **Algoritma** adalah langkah-langkah terurut untuk menyelesaikan masalah.  
Jawaban: Benar
4. AI selalu butuh data untuk belajar.  
Jawaban: Benar
5. **Pandas** dipakai untuk membuat animasi 3D.  
Jawaban: Salah

## B. Pilihan Ganda (10 Soal)

1. Yang **bukan** bagian *computational thinking* adalah...  
A. Dekomposisi B. Pengenalan Pola C. Abstraksi D. Dekorasi

Jawaban: D

2. Perintah Python untuk menulis teks ke layar adalah...  
A. say() B. echo() C. print() D. out()

Jawaban: C

3. Kegunaan utama **Pandas** adalah...  
A. Menggambar 3D B. Olah data tabel C. Main game D. Desain poster

Jawaban: B

4. Output kode berikut?

```
print(2 + 3 * 2)
```

A. 10 B. 8 C. 12 D. 7

Jawaban: B

5. Contoh **abstraksi** saat bikin robot penyiram tanaman adalah...  
A. Warna robot B. Nama robot C. Sensor kelembaban D. Stiker robot

Jawaban: C

6. Struktur data yang cocok untuk **antrian UKS** (masuk dulu → keluar dulu) adalah...  
A. Stack B. Queue C. Set D. Dictionary

Jawaban: B

7. Di AI, **neural network** adalah...  
A. Jenis kabel B. Model belajar dari data C. Aplikasi gambar D. Sistem operasi

Jawaban: B

8. Yang termasuk *open source* untuk *machine learning* adalah...  
A. Keras B. Excel C. Photoshop D. PowerPoint

Jawaban: A

9. Urutan *algoritma nasi goreng* yang benar adalah...  
A. Sajikan → Tumis bumbu → Masukkan nasi  
B. Tumis bumbu → Masukkan nasi → Sajikan  
C. Masukkan nasi → Sajikan → Tumis bumbu  
D. Sajikan → Masukkan nasi → Tumis bumbu

Jawaban: B

10. Hasil kode berikut?

```
nilai = 80
if nilai >= 75:
    print("Lulus")
```

```
else:  
    print("Remedial")
```

A. Remedial B. Error C. Lulus D. Tidak tampil

Jawaban: C

### C. Isian Singkat (5 Soal)

1. Sebutkan 4 langkah utama *computational thinking* secara urut.

Jawaban: Dekomposisi, Pengenalan Pola, Abstraksi, Algoritma

2. Perintah Python untuk menampilkan **Halo Dunia!** adalah \_\_\_\_\_.

Jawaban: `print("Halo Dunia!")`

3. **Abstraksi** berarti fokus pada \_\_\_\_\_ dan abaikan yang tidak penting.

Jawaban: hal penting

4. *Library* Python untuk olah data tabel seperti Excel adalah \_\_\_\_\_.

Jawaban: Pandas

5. Lengkapi kode agar menjumlahkan dua angka:

```
a = 7  
b = 5  
hasil = _____  
print(hasil)
```

Jawaban: `a + b`

### D. Soal Explorasi (3 Soal)

Tuliskan jawabanmu singkat, jelas, dan boleh pakai *pseudocode* / Python mini.

#### Soal 1:

**Rancang pengecek *deadline* tugas.**

Input: tanggal hari ini dan daftar tugas (nama, tanggal selesai). Keluarkan daftar tugas yang **jatuh tempo  $\leq 2$  hari**.

Kunci jawaban (contoh):

```
untuk setiap tugas:  
    selisih = tanggal_tugas - hari_ini  
    jika selisih <= 2 hari:  
        tambahkan ke daftar_mendesak  
tampilkan daftar_mendesak  
  
from datetime import date  
hari_ini = date(2025, 9, 14)
```

```
tugas = [("Matematika","2025-09-15"), ("IPA","2025-09-20"),
("Bhs","2025-09-14")]
from datetime import datetime
mendesak = [n for (n,t) in tugas
              if (datetime.fromisoformat(t).date() -
hari_ini).days <= 2]
print(mendesak)
```

## Soal 2:

### Buat penyaring rekomendasi video sederhana.

Input: daftar video (judul, durasi, tag), dan minat pengguna (misal: "game", durasi ≤ 5 menit). Keluarkan video yang cocok.

Kunci jawaban (contoh):

```
untuk setiap video:
    jika tag berisi minat DAN durasi <= batas:
        tambahkan ke rekomendasi
tampilkan rekomendasi

video = [
    ("Tips Aim",4,{"game","fps"}),
    ("Belajar Masak",6,{"food"}),
    ("Match Highlights",5,{"game","moba"})
]
minat, batas = "game", 5
rekom = [j for (j,d,t) in video if (minat in t and d <= batas)]
print(rekom)
```

## Soal 3:

### Hitung statistik sederhana jam belajar kelas.

Input: daftar nama dan jam belajar per hari. Tampilkan **rata-rata**, **maksimum**, dan **siapa yang paling rajin**.

Kunci jawaban (contoh):

```
rata = jumlah_semua_jam / banyak_siswa
maks = nilai_jam_tertinggi
nama_terajin = nama dengan jam = maks
tampilkan rata, maks, nama_terajin

import pandas as pd
df =
pd.DataFrame({"Nama":["Andi","Budi","Citra"],"Jam":[2,3,1]})
print(df["Jam"].mean(), df["Jam"].max(),
df.loc[df["Jam"].idxmax(),"Nama"])
```

Latihan ini melatih **dekomposisi**, **pola**, **abstraksi**, dan **algoritma**. Semua contoh pakai tool *open source*: **Python** dan **Pandas**, serta konsep *AI* seperti *neural network* secara ringan. Selamat mengerjakan!

## Tugas Proyek (5 Soal)

### Proyek 1: Detektor Kemiripan Teks (Anti-Copas Mini)

**Ide:** Bandingkan dua esai. Hitung seberapa mirip. Pakai *Jaccard similarity*.

**Fokus CT:** **Abstraksi** (ambil kata penting), **Pola** (kata yang sama), **Algoritma** (hitung skor).

#### Langkah:

- Ambil 2 teks pendek ( $\leq 150$  kata).
- Ubah ke huruf kecil. Buang tanda baca.
- Pecah jadi *set* kata.
- Hitung  $Jaccard = (irisannya) / (gabungannya)$ .
- Tampilkan skor 0–1 (semakin dekat 1, semakin mirip).

#### Pseudocode:

```
teks1, teks2 -> lower()
kata1 = set(tokenisasi(teks1))
kata2 = set(tokenisasi(teks2))
irisan = len(kata1  $\cap$  kata2)
gabungan = len(kata1  $\cup$  kata2)
skor = irisan / gabungan
tampilkan skor
```

#### Python:

```
import re

def tokenize(s):
    s = s.lower()
    s = re.sub(r'^a-z0-9\s', ' ', s)
    return set(s.split())

t1 = "Belajar Python itu seru dan mudah dipahami."
t2 = "Python itu asyik, belajar terasa mudah dan seru."
k1, k2 = tokenize(t1), tokenize(t2)
skor = len(k1 & k2) / len(k1 | k2)
print("Skor kemiripan:", round(skor, 3))
```

#### Kunci Jawaban (contoh):

Skor sekitar **0.6–0.8** (cukup mirip).

## Proyek 2: Simulasi Antrian UKS (First In First Out)

**Ide:** Buat *queue* untuk siswa yang antri ke UKS. Yang datang duluan dilayani dulu.

**Fokus CT:** **Dekomposisi** (datang/dilayani), **Algoritma** (FIFO), **Pola** (urutan masuk=urutan keluar).

### Langkah:

- Buat daftar antrian kosong.
- Tambah nama saat “datang”.
- Saat “layani”, pop dari depan.
- Cetak antrian setelah tiap aksi.

### Pseudocode:

```
antrian = []
fungsi datang(nama): antrian.append(nama)
fungsi layani():
    jika antrian tidak kosong: print(pop_depan)
    else: print("kosong")
```

### Python:

```
from collections import deque
q = deque()
def datang(n): q.append(n)
def layani():
    if q: print("Dilayani:", q.popleft())
    else: print("Antrian kosong")

datang("Ayu"); datang("Bima"); datang("Cici")
print("Antrian:", list(q))
layani(); print("Antrian:", list(q))
```

### Kunci Jawaban (contoh):

Urutan dilayani: **Ayu** → **Bima** → **Cici**.

## Proyek 3: Perencana Menu Sehat & Hemat (Greedy Sederhana)

**Ide:** Pilih menu harian agar **kalori mendekati target** dan **biaya ≤ budget**. Pakai *greedy*.

**Fokus CT:** **Abstraksi** (pakai kalori & harga saja), **Algoritma** (*greedy* pilih rasio terbaik), **Pola** (harga per kalori).

### Langkah:

- Buat tabel menu: Nama, Kalori, Harga.
- Hitung rasio *kalori/harga*.
- Urutkan menurun. Ambil dari atas sampai mendekati target kalori dan tidak lewat budget.

- Cetak pilihan.

#### Pseudocode:

```

input target_kalori, budget
urutkan menu berdasarkan (kalori/harga) turun
total_kalori = 0; total_biaya = 0; pilihan = []
untuk setiap item:
    jika total_biaya + harga <= budget dan total_kalori + kalori
    <= target_kalori:
        ambil item -> update total
tampilkan pilihan, total_kalori, total_biaya

```

#### Python (Pandas):

```

import pandas as pd

df = pd.DataFrame({
    "Menu":["Nasi","Ayam","Tahu","Sayur","Telur"],
    "Kalori":[300, 250, 120, 60, 150],
    "Harga":[5000, 8000, 3000, 2000, 4000]
})
df["rasio"] = df["Kalori"] / df["Harga"]
df = df.sort_values("rasio", ascending=False)

target, budget = 700, 12000
pilih, tkal, tbi = [], 0, 0
for _, r in df.iterrows():
    if tbi + r.Harga <= budget and tkal + r.Kalori <= target:
        pilih.append(r.Menu); tkal += r.Kalori; tbi += r.Harga
print("Pilihan:", pilih, "| Kalori:", tkal, "| Biaya:", tbi)

```

#### Kunci Jawaban (contoh):

Pilihan bisa **Nasi + Tahu + Sayur** (kalori ~480–540, biaya  $\leq$  budget). Hasil bisa berbeda sesuai data.

### Proyek 4: Prediksi Nilai dari Jam Belajar (*Regression Keras*)

**Ide:** Hubungkan jam belajar dengan nilai ulangan. Latih *neural network* kecil regresi.

**Fokus CT:** **Pola** (hubungan jam–nilai), **Abstraksi** (fitur = jam), **Algoritma** (latih  $\rightarrow$  uji).

#### Langkah:

- Buat *dataset* kecil: (jam\_belajar, nilai).
- Bagi jadi *train/test* sederhana.
- Model: Dense(8, *relu*)  $\rightarrow$  Dense(1) (*regression*).
- Latih beberapa *epoch*.
- Prediksi nilai untuk jam\_belajar baru.

### Pseudocode:

```
X = jam_belajar; y = nilai
model: Dense(8, 'relu')->Dense(1)
compile: adam, mse
fit: epochs kecil
prediksi: jam_baru -> nilai_perkiraan
```

### Python (Keras)

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

X = np.array([[1],[2],[3],[4],[5],[6]], dtype="float32")
y = np.array([55,60,68,75,82,90], dtype="float32")

model = keras.Sequential([
    layers.Input(shape=(1,)),
    layers.Dense(8, activation="relu"),
    layers.Dense(1)
])
model.compile(optimizer="adam", loss="mse")
model.fit(X, y, epochs=300, verbose=0)

jam_baru = np.array([[2.5],[5.5]], dtype="float32")
pred = model.predict(jam_baru, verbose=0).flatten()
print("Prediksi nilai:", [round(float(p),1) for p in pred])
```

### Kunci Jawaban (contoh):

Untuk **2.5 jam**  $\approx$  **64–70**, untuk **5.5 jam**  $\approx$  **86–90** (perkiraan, tergantung training).

## Proyek 5: Lampu Lalu Lintas Adaptif (*Rule & State Machine*)

**Ide:** Atur lama lampu hijau berdasar jumlah kendaraan. Lebih ramai  $\rightarrow$  lebih lama hijau.

**Fokus CT:** **Dekomposisi** (sensor  $\rightarrow$  keputusan  $\rightarrow$  waktu), **Algoritma** (aturan), **Abstraksi** (hanya hitung mobil).

### Langkah:

- Input: jumlah\_mobil\_utara, jumlah\_mobil\_timur.
- Aturan: dasar hijau 15 detik. Tambah 1 detik per 2 mobil (maks 40 dtk).
- Ganti *state*: HIJAU\_UTARA  $\rightarrow$  KUNING  $\rightarrow$  HIJAU\_TIMUR  $\rightarrow$  KUNING  $\rightarrow$  ulang.
- Cetak durasi tiap *state*.

### Pseudocode:

```
fungsi durasi_hijau(jumlah):
```

```

d = 15 + floor(jumlah/2)
jika d > 40: d = 40
kembalikan d

loop selamanya:
    tU = durasi_hijau(mobil_utara)
    state = HIJAU_UTARA selama tU
    state = KUNING 3 dtk
    tT = durasi_hijau(mobil_timur)
    state = HIJAU_TIMUR selama tT
    state = KUNING 3 dtk

```

### Python:

```

import math

def durasi_hijau(jml):
    d = 15 + math.floor(jml/2)
    return min(d, 40)

mobil_utara, mobil_timur = 18, 6
tU, tT = durasi_hijau(mobil_utara), durasi_hijau(mobil_timur)
siklus = [
    ("HIJAU_UTARA", tU),
    ("KUNING", 3),
    ("HIJAU_TIMUR", tT),
    ("KUNING", 3),
]
print("Siklus:")
for nama, dtk in siklus:
    print(f"{nama}: {dtk} detik")

```

### Kunci Jawaban (contoh):

- Jika Utara=18, Timur=6 → HIJAU\_UTARA 24 dtk, KUNING 3, HIJAU\_TIMUR 18 dtk, KUNING 3.

### Penutup Singkat

- **Semua proyek beda** dari aktivitas sebelumnya, tapi tetap pakai pilar **dekomposisi–pola–abstraksi–algoritma**.
- Tools yang dipakai **100% open source: Python, Pandas, Keras** (*neural network*).
- Kerjakan langkahnya pelan-pelan. Uji kecil. Perbaiki. *Repeat*. Gas terus, tim!

## BAB 3: Python Dasar

Gimana kabarnya? Semoga selalu semangat ya! Hari ini kita mau belajar sesuatu yang seru banget, namanya **Python**. Ada yang pernah dengar? Keren deh kalau sudah, tapi kalau belum, jangan khawatir! Karena kita akan belajar dari nol sampai kalian bisa bikin program sendiri. Siap? Yuk, kita mulai petualangan kita! 🚀

### Tujuan Pembelajaran

Setelah belajar bareng, kita harap kalian bisa:

- **Paham banget** apa itu Python dan gunanya buat apa aja.
- **Bisa nulis** kode Python yang super sederhana.
- **Ngerti** cara kerja komputer, terutama saat baca kode Python.
- **Jago** memecahkan masalah kecil dengan kode Python.

### Peta Konsep Sederhana

Biar nggak bingung, bayangin aja **Python** itu kayak resep masakan.

- **Bahan-bahan (Data):** Angka, tulisan, benar/salah.
- **tool Masak (Variabel):** Wadah buat nyimpen bahan.
- **Langkah-langkah (Fungsi):** Cara bikinnya.
- **Hasil Masakan (Program):** Jadi deh!

### Apersepsi: Misteri Angka Hilang

Bayangin kalian lagi main game. Tiba-tiba, ada kode rahasia yang butuh kalian pecahin. Kodenya:  $5 + 3 = ?$  Gampang banget kan? Tapi kalau kodenya:  $(15 * 8) - (4 / 2) = ?$  Nah, mulai pusing kan? **Python** itu kayak kalkulator super canggih yang bisa ngerjain soal-soal itu dalam sekejap. Bahkan, bisa ngelakuin hal-hal yang lebih gila lagi!

### Penjelasan Teori: Python itu Gampang!

Hari ini kita kenalan lebih dalam sama **Python**. Jangan bayangin ular ya 🐍, tapi bayangin sebuah *bahasa pemrograman open source* yang bisa dipakai gratis, gampang dipelajari, dan bisa dipakai buat bikin hal-hal keren.

Python itu ibarat bahasa Indonesia-nya komputer. Kalau kita ngobrol sama teman pakai kata-kata, kita ngobrol sama komputer pakai *kode Python*.

### Kenapa Harus Python?

- **Gratis & open source** → Nggak perlu bayar, semua orang bisa pakai.
- **Mudah dibaca** → Sintaks Python mirip bahasa Inggris.

- **Serba bisa** → Bisa buat *website*, *game*, aplikasi Android, robot pabrik, sampai *machine learning*.
- **Dipakai industri besar** → Google, Instagram, Spotify, dan YouTube pakai Python.

## Python = Resep Masakan

Bayangin kalian lagi masak mie instan.

- **Bahan (data)** → angka, teks, benar/salah.
- **Wadah (variabel)** → tempat simpan bahan.
- **Langkah (fungsi)** → instruksi cara masak.
- **Hasil (program)** → mie instan siap disantap.

Nah, coding itu mirip banget sama masak. Python bikin “resep” jadi gampang dipahami komputer.

## Contoh Sederhana

```
pensil = 10
print("Jumlah pensilku adalah:", pensil)
```

Output di layar:

```
Jumlah pensilku adalah: 10
```

Variabel `pensil` itu cuma wadah buat nyimpen angka **10**.

## Python di Kehidupan Sehari-Hari

- **Anak RPL** → bikin game balapan, Python atur skor dan kecepatan mobil.
- **Anak Multimedia** → atur animasi atau efek video dengan Python.
- **Anak TKJ** → bikin bot cek jaringan internet.
- **Anak Akuntansi** → hitung laporan keuangan pakai *Pandas*.
- **Anak Warkop** → bikin kasir mini buat ngitung tagihan kopi teman-teman.

## Python untuk AI dan *Machine Learning*

Python bukan cuma buat hitung-hitungan, tapi juga buat bikin komputer belajar sendiri:

- *Machine Learning* → komputer belajar dari data.
- *Neural Network* → tiruan cara kerja otak manusia.
- *Keras* → pustaka *open source* buat bikin model AI lebih cepat.

Bayangin kalian bikin program yang bisa ngenalin wajah teman dari foto. Itu bisa banget pakai Python!

## Belajar Python di Google Colab

Kabar baiknya, kalian nggak perlu install apa pun di laptop.

- **Google Colab** = ruang kelas online untuk coding Python.
- Bisa langsung dipakai lewat browser.
- Gratis, asal punya akun Gmail.
- Mendukung banyak pustaka keren: *Pandas*, *Numpy*, *Keras*, *TensorFlow*.
- Cocok banget buat latihan coding, bikin proyek data, sampai uji coba *machine learning*.

## Rangkuman Singkat

- Python = bahasa pemrograman *open source* yang mudah dipelajari.
- Bisa dipakai buat banyak hal: dari game, data, sampai kecerdasan buatan.
- Python mirip resep masakan → ada bahan, wadah, langkah, dan hasil.
- Belajar Python bukan soal hafalan, tapi soal berani mencoba dan memecahkan masalah.

## Kasus Ilustrasi: Python di Kehidupan Sehari-hari

### Kasus 1: Kasir Warung Kopi

#### Ilustrasi:

Kamu jaga kasir. Ada pesan **kopi** dan **roti**. Kadang ada **diskon**. Komputer harus hitung total cepat. Tanpa salah.

#### Tujuan:

Hitung total belanja. Terapkan diskon kalau belanja  $\geq 50.000$ .

**Konsep:** *variable*, operator, *if-else*.

#### Pseudocode:

```
SET harga_kopi = 15000
SET harga_roti = 10000
INPUT jumlah_kopi
INPUT jumlah_roti

SET subtotal = (harga_kopi * jumlah_kopi) + (harga_roti *
jumlah_roti)

IF subtotal >= 50000 THEN
    SET diskon = 0.1 * subtotal
ELSE
    SET diskon = 0
END IF
```

```

SET total = subtotal - diskon

PRINT "Subtotal:", subtotal
PRINT "Diskon:", diskon
PRINT "Total Bayar:", total

```

**Catatan open source:**

Nanti bisa dibuat di **Python**. Kalau mau ekspor struk ke CSV, pakai *library* **CSV** bawaan.

## Kasus 2: Nilai Rata-Rata Kelas dengan *Pandas*

**Ilustrasi:**

Kamu ketua kelas. Punya daftar nilai di file **CSV**. Kamu ingin tahu **rata-rata**, **nilai tertinggi**, dan **siapa yang remedial**.

**Tujuan:**

Baca data. Hitung statistik dasar. Tandai siswa < 70.

**Konsep:** *dataset*, *Pandas*, agregasi.

**Pseudocode:**

```

IMPORT pandas AS pd

READ file "nilai_kelas.csv" INTO df

SET rata2 = MEAN(df["nilai"])
SET maks  = MAX(df["nilai"])
SET min   = MIN(df["nilai"])

FILTER remedial = ROWS WHERE df["nilai"] < 70

PRINT "Rata-rata:", rata2
PRINT "Nilai Tertinggi:", maks
PRINT "Nilai Terendah:", min
PRINT "Daftar Remedial:", remedial["nama"]

```

**Catatan open source:**

**Pandas** itu *open source*. Cocok untuk analisis data di **Python**.

## Kasus 3: Game Tebak Angka Sederhana

**Ilustrasi:**

Komputer punya **angka rahasia**. Kamu menebak. Komputer memberi **petunjuk**: lebih besar atau lebih kecil. Seru!

**Tujuan:**

Latihan *loop*, *if-else*, dan *input-output*.

**Konsep:** *while loop*, logika perbandingan.

**Pseudocode:**

```
SET angka_rahasia = 27
SET tebakan = -1

WHILE tebakan != angka_rahasia DO
    INPUT tebakan
    IF tebakan < angka_rahasia THEN
        PRINT "Terlalu kecil!"
    ELSE IF tebakan > angka_rahasia THEN
        PRINT "Terlalu besar!"
    ELSE
        PRINT "Benar! Kamu menang!"
    END IF
END WHILE
```

**Catatan open source:**

Bisa dijalankan di **Python REPL** atau **VS Code** (editor *open source*-friendly).

## Kasus 4: Cek Kategori BMI (Awal *Machine Learning*)

**Ilustrasi:**

Kamu ingin tahu kategori **BMI: Kurus/Normal/Gemuk**. Ini langkah awal sebelum *machine learning*. Nanti datanya bisa dilatih jadi model.

**Tujuan:**

Hitung BMI. Klasifikasikan sederhana.

**Konsep:** rumus, *if-elif-else*. Gambaran menuju *machine learning*.

**Pseudocode:**

```
INPUT berat_kg
INPUT tinggi_m

SET bmi = berat_kg / (tinggi_m * tinggi_m)

IF bmi < 18.5 THEN
    SET kategori = "Kurus"
ELSE IF bmi < 25 THEN
    SET kategori = "Normal"
ELSE
    SET kategori = "Gemuk"
END IF

PRINT "BMI:", bmi
PRINT "Kategori:", kategori
```

**Catatan open source (lanjutan):**

Kalau mau naik level ke *machine learning*, nanti kumpulkan *dataset* BMI. Bisa latih model di **Python**. Untuk *deep learning*, pakai **Keras** (*open source*) bikin *neural network* sederhana. (Kasus 5 di bawah kasih *pseudocode* ide dasarnya.)

**Kasus 5: Neural Network Mini untuk Prediksi Lulus/Remedial****Ilustrasi:**

Kamu punya data nilai tugas, kuis, ujian. Kamu ingin **prediksi**: *Lulus* atau *Remedial*. Ini contoh *AI* mini.

**Tujuan:**

Tunjukkan alur dasar *neural network* dengan **Keras**. (Masih *pseudocode*, biar mudah.)

**Konsep:** *dataset, split, train-test, neural network, epoch*.

**Pseudocode:**

```
IMPORT pandas AS pd
IMPORT keras (layers, models)

# 1) Baca data
READ "nilai_siswa.csv" INTO df

# 2) Pisah fitur dan label
SET X = df[["tugas", "kuis", "ujian"]]
SET y = df["label_lulus"]      # 1 = Lulus, 0 = Remedial

# 3) Bagi data
SPLIT X, y INTO X_train, X_test, y_train, y_test WITH ratio
80:20

# 4) Bangun model NN sederhana
MODEL = Sequential()
ADD Dense 8 units, activation "relu", input_shape = jumlah_fitur
ADD Dense 4 units, activation "relu"
ADD Dense 1 unit, activation "sigmoid"

# 5) Kompilasi
COMPILE MODEL WITH optimizer "adam", loss "binary_crossentropy",
metrics ["accuracy"]

# 6) Latih
FIT MODEL ON X_train, y_train FOR epochs = 20, batch_size = 16

# 7) Evaluasi
EVALUATE MODEL ON X_test, y_test
PRINT "Akurasi:", accuracy

# 8) Prediksi data baru
INPUT nilai_tugas, nilai_kuis, nilai_ujian
```

```

SET pred = MODEL.PREDICT([nilai_tugas, nilai_kuis, nilai_ujian])

IF pred >= 0.5 THEN
    PRINT "Prediksi: Lulus"
ELSE
    PRINT "Prediksi: Remedial"
END IF

```

### Catatan *open source*:

**Keras** berjalan di atas *backend open source* (seperti **TensorFlow**). Semua bisa dipakai gratis.

## Ringkasan

- Mulai dari logika dasar: *variable*, operator, *if-else*, *loop*.
- Lanjut ke data: baca file CSV pakai **Pandas**.
- Naik level: *machine learning* dan *neural network* dengan **Keras**.
- Semua *open source*. Semua bisa dicoba di **Python**.

## Soal Latihan & Pembahasan

### Soal 1

Apa output dari kode berikut?

```
print("Belajar Python itu seru!")
```

#### Jawaban:

Belajar Python itu seru!

#### Pembahasan:

Fungsi *print()* dipakai untuk menampilkan teks. Teks harus diapit tanda kutip ("..." atau '...').

### Soal 2

Perhatikan kode berikut:

```

a = 8
b = 4
c = a * b
print(c)

```

Berapakah hasil yang ditampilkan?

**Jawaban:**

32

**Pembahasan:**

Variabel **a** berisi angka 8, **b** berisi 4. Operasi **a \* b** artinya  $8 \times 4 = 32$ .

### Soal 3

Ada yang salah dari kode ini. Apa ya?

```
print(Saya suka Python)
```

**Jawaban:**

Tulisan **Saya suka Python** harus pakai tanda kutip.

**Pembahasan:**

Python tidak bisa membaca teks tanpa tanda kutip. Seharusnya:

```
print("Saya suka Python")
```

### Soal 4

Jika:

```
nama = "Ani"  
umur = 16
```

Bagaimana cara menampilkan: **Ani berumur 16 tahun ?**

**Jawaban:**

```
print(nama, "berumur", umur, "tahun")
```

**Pembahasan:**

Kita bisa gabungkan teks dan variabel di dalam *print()*. Python otomatis memisahkan dengan spasi.

### Soal 5

Apa hasil dari kode ini?

```
angka1 = 50  
angka2 = 20  
hasil = angka1 / angka2  
print(hasil)
```

**Jawaban:**

2.5

**Pembahasan:**

Python selalu menampilkan hasil pembagian dalam bentuk *float* (desimal), walaupun hasilnya pecahan sederhana.

**Catatan Tambahan**

- Python itu bisa dipakai untuk hal keren seperti *machine learning*.
- Dengan *Pandas*, kalian bisa olah data seperti Excel tapi lebih cepat.
- Kalau mau bikin AI, bisa pakai *Keras* dan *neural network* — semua gratis dan open source!

**Tips Belajar**

- Jangan takut salah, coba terus!
- Praktek langsung di **Google Colab** biar gampang tanpa install apa-apa.
- Kerja bareng teman, biar lebih seru dan saling belajar.

**Fun Facts**

- Nama **Python** diambil dari acara komedi *Monty Python's Flying Circus*.
- Banyak teknologi populer (Instagram, YouTube, Spotify) dibangun pakai Python.
- Python bisa otomatisasi tugas, misalnya ganti nama 1000 file hanya dalam beberapa detik.

**Tips Belajar & Menghafal Cepat**

1. **Jangan takut salah!** Coding itu kayak main game, harus coba-coba. Kalau salah, berarti kalian lagi belajar.
2. **Coba praktek langsung!** Jangan cuma baca. Buka komputer, ketik kodenya, dan lihat apa yang terjadi.
3. **Cari teman belajar.** Belajar bareng lebih seru dan bisa saling tanya jawab.
4. **Bikin catatan kecil.** Catat hal-hal penting pakai bahasa kalian sendiri biar gampang diingat.

# Aktivitas Siswa: Python for Fun!

## Aktivitas 1: Salam & Perkenalan Otomatis

Kita buat program yang menyapa nama dan hobi kalian. Sederhana tapi berguna.

**Tujuan:** latihan *input*, *output*, variabel.

**Langkah:**

- Minta nama.
- Minta hobi.
- Cetak kalimat sapaan.

**Pseudocode:**

```
INPUT nama
INPUT hobi
PRINT "Halo", nama
PRINT "Hobimu:", hobi
```

**Python (jawaban):**

```
nama = input("Masukkan namamu: ")
hobi = input("Masukkan hobimu: ")
print("Halo", nama + "!")
print("Hobimu:", hobi)
```

## Aktivitas 2: Kasir Mini Diskon 10%

Hitung total belanja dua barang. Kalau  $\geq 50.000$ , diskon 10%. Tampilkan kembalian.

**Tujuan:** operator aritmetika, *if-else*.

**Langkah:**

- Simpan harga barang.
- Input jumlah.
- Hitung subtotal.
- Terapkan diskon jika perlu.
- Minta uang bayar.
- Hitung kembalian.

**Pseudocode:**

```
SET harga_a = 15000
SET harga_b = 10000
INPUT jml_a
```

```

INPUT jml_b
subtotal = harga_a*jml_a + harga_b*jml_b

IF subtotal >= 50000 THEN
    diskon = 0.10 * subtotal
ELSE
    diskon = 0
END IF

total = subtotal - diskon
INPUT bayar
kembalian = bayar - total

PRINT subtotal, diskon, total, kembalian

```

### Python (jawaban):

```

harga_a, harga_b = 15000, 10000
jml_a = int(input("Jumlah A: "))
jml_b = int(input("Jumlah B: "))
subtotal = harga_a*jml_a + harga_b*jml_b
diskon = 0.10*subtotal if subtotal >= 50000 else 0
total = subtotal - diskon
print("Subtotal:", subtotal)
print("Diskon:", diskon)
print("Total:", total)
bayar = int(input("Uang bayar: "))
kembalian = bayar - total
print("Kembalian:", kembalian)

```

## Aktivitas 3: Nilai Kelas dengan Pandas

Baca file CSV sederhana, hitung rata-rata, min, max, dan siapa yang perlu remedial (< 70).

**Tujuan:** kenalan *dataset* dan *Pandas (open source)*.

### Langkah:

- Siapkan `nilai_kelas.csv` dengan kolom: `nama`, `nilai`.
- Baca CSV.
- Hitung statistik.
- Tampilkan daftar remedial.

### Contoh isi CSV (buat sendiri):

```

nama,nilai
Ana,85
Budi,67
Cici,92
Doni,58

```

### Pseudocode:

```
IMPORT pandas AS pd
df = READ_CSV("nilai_kelas.csv")
rata2 = MEAN(df["nilai"])
maks  = MAX(df["nilai"])
min   = MIN(df["nilai"])
remed = FILTER df WHERE nilai < 70
PRINT rata2, maks, min, remed["nama"]
```

### Python (jawaban):

```
import pandas as pd

df = pd.read_csv("nilai_kelas.csv")
rata2 = df["nilai"].mean()
maks = df["nilai"].max()
minn = df["nilai"].min()
remed = df[df["nilai"] < 70]["nama"].tolist()

print("Rata-rata:", rata2)
print("Tertinggi:", maks)
print("Terendah :", minn)
print("Remedial :", remed)
```

## Aktivitas 4: Tabel Konversi Suhu dengan loop

Buat tabel Celcius → Fahrenheit untuk rentang nilai.

**Tujuan:** *loop*, format *output*.

### Langkah:

- Input awal, akhir, dan langkah.
- Ulangi dari awal sampai akhir.
- Konversi:  $F = (C * 9/5) + 32$ .
- Cetak rapi.

### Pseudocode:

```
INPUT awal
INPUT akhir
INPUT langkah
FOR c FROM awal TO akhir STEP langkah DO
    f = (c*9/5) + 32
    PRINT c, f
END FOR
```

### Python (jawaban):

```

awal = int(input("Celcius awal: "))
akhir = int(input("Celcius akhir: "))
langkah = int(input("Langkah: "))

print("C\tF")
for c in range(awal, akhir + 1, langkah):
    f = (c * 9/5) + 32
    print(f"{c}\t{f:.1f}")

```

## Aktivitas 5: Mini ML—Prediksi Lulus/Remedial dengan Keras

Kita coba *machine learning* kecil. Pakai *neural network* dari Keras (semua *open source*). Data contoh dibuat kecil biar mudah.

**Tujuan:** lihat alur *ML*: data → bagi train/test → latih → uji → prediksi.

**Langkah:**

- Buat *dataset* kecil: fitur tugas, kuis, ujian, label lulus(1)/remedial(0).
- Bagi data 80:20.
- Bangun model sederhana.
- Latih beberapa *epoch*.
- Uji akurasi.
- Prediksi satu contoh baru.

**Pseudocode:**

```

IMPORT pandas, numpy, keras
DATA = buat dataframe kecil
SPLIT DATA jadi X, y
TRAIN_TEST_SPLIT 80:20
MODEL = Dense(8,"relu") -> Dense(4,"relu") -> Dense(1,"sigmoid")
COMPILE adam, binary_crossentropy, accuracy
FIT epochs=50, batch=8
EVALUATE X_test, y_test
PREDICT nilai_baru
PRINT akurasi, prediksi

```

**Python (jawaban—contoh minimal):**

```

# Pastikan sudah install: pip install tensorflow pandas numpy
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# 1) Dataset kecil (contoh)
data = {

```

```

        "tugas": [80, 60, 75, 90, 55, 65, 95, 70, 50, 85, 78, 62],
        "kuis":  [75, 58, 70, 88, 60, 64, 92, 68, 55, 82, 74, 61],
        "ujian": [78, 55, 72, 91, 57, 66, 94, 69, 52, 86, 76, 63],
    }
    df = pd.DataFrame(data)
    # Label sederhana: lulus = 1 jika (rata-rata >= 70), else 0
    df["label"] = ((df[["tugas", "kuis", "ujian"]].mean(axis=1)) >=
    70).astype(int)

    X = df[["tugas", "kuis", "ujian"]].values
    y = df["label"].values

    # 2) Split data
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42, stratify=y
    )

    # 3) Model NN sederhana
    model = Sequential([
        Dense(8, activation="relu", input_shape=(3,)),
        Dense(4, activation="relu"),
        Dense(1, activation="sigmoid")
    ])

    model.compile(optimizer="adam", loss="binary_crossentropy",
    metrics=["accuracy"])

    # 4) Latih
    model.fit(X_train, y_train, epochs=50, batch_size=8, verbose=0)

    # 5) Evaluasi
    loss, acc = model.evaluate(X_test, y_test, verbose=0)
    print("Akurasi:", round(float(acc), 3))

    # 6) Prediksi contoh baru
    nilai_baru = np.array([[72, 68, 75]]) # tugas, kuis, ujian
    pred = model.predict(nilai_baru, verbose=0)[0][0]
    print("Skor prediksi:", round(float(pred), 3))
    print("Prediksi:", "Lulus" if pred >= 0.5 else "Remedial")

```

#### Catatan:

- Ini *demo* kecil. Untuk kelas besar, pakai *dataset* lebih banyak.
- Semua tool yang dipakai adalah *open source*.

### Penutup singkat

- Mulai dari *input-output*, lanjut *if-else*, *loop*, lalu data pakai **Pandas**.
- Terakhir, coba *machine learning* mini dengan **Keras** dan *neural network*.
- Semua bebas dipakai. Semua *open source*.
- Coba, ulangi, eksperimen. Itu kunci jago coding.

## Rangkuman

Python itu bahasa untuk “ngobrol” dengan komputer. **Mudah dibaca** dan **gratis karena open source**. Kita pakai **variabel** sebagai wadah data, dan **fungsi** sebagai langkah kerja. Kodenya mirip bahasa sehari-hari. Hasilnya jelas di layar. Ini bikin pemula cepat paham. *Goal*-nya sederhana: **pahami perintah, coba, lihat hasil**.

Python dipakai di banyak hal. **Ngolah data** pakai *Pandas*. **Bangun kecerdasan buatan** dengan *machine learning*, *Keras*, dan *neural network*. Semua juga *open source*. Kita belajar dari dasar: **input–output, operator, percabangan, dan loop**. Lalu naik kelas ke *dataset*, analisis, dan prediksi. **Langkah kecil, tapi konsisten**.

Kunci suksesnya: **praktek rutin**. Tulis kode pendek. Uji. Perbaiki. Ulangi. Catat hal penting. Diskusi dengan teman. Jangan takut error—itu guru terbaik. Dengan kebiasaan ini, kalian siap bikin proyek nyata: kasir mini, konversi suhu, hingga model *AI* sederhana. **Python itu tool. Kalian yang berkarya**.

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. Python adalah bahasa pemrograman *open source*.  
Jawaban: Benar
2. Variabel di Python dipakai untuk menyimpan data.  
Jawaban: Benar
3. Fungsi `print()` dipakai untuk membaca file.  
Jawaban: Salah
4. *Pandas* dipakai untuk analisis data tabel.  
Jawaban: Benar
5. *Neural network* selalu dipakai untuk menghitung penjumlahan sederhana.  
Jawaban: Salah

### B. Pilihan Ganda (10 Soal)

1. Perintah untuk menampilkan teks di layar adalah...  
A. `show()` B. `echo()` C. `print()` D. `out()`  
Jawaban: C
2. Manakah yang benar untuk membuat variabel?  
A. `10 = usia` B. `usia = 10` C. `var usia 10` D. `usia:10`

Jawaban: B

3. Tipe data untuk teks adalah...  
A. `int` B. `float` C. `str` D. `bool`

Jawaban: C

4. Hasil dari kode berikut apa?

```
a=5; b=2; print(a*b)
```

A. 2 B. 5 C. 7 D. 10

Jawaban: D

5. Pandas biasa disingkat impor sebagai...

- A. `import pandas as ps`  
B. `import pandas as pd`  
C. `import pandas as pn`  
D. `import pandas as pa`

Jawaban: B

6. Rumus konversi Celcius ke Fahrenheit adalah...

- A.  $F = (C * 5/9) + 32$   
B.  $F = (C * 9/5) + 32$   
C.  $F = C + 273$   
D.  $F = C * 1.8 - 32$

Jawaban: B

7. *Machine learning* dipakai untuk...

- A. Mengatur resolusi layar  
B. Membuat komputer belajar dari data  
C. Mengganti tema editor  
D. Menyusun folder otomatis

Jawaban: B

8. *Keras* adalah pustaka untuk...

- A. Menggambar GUI  
B. *Web scraping*  
C. *Deep learning*  
D. Enkripsi file

Jawaban: C

9. Output kode berikut apa?

```
x="Python"; print(x, "mudah")
```

- A. Pythonmudah  
B. Python mudah  
C. mudah Python  
D. x mudah

Jawaban: B

10. Operator pembagian di Python adalah...

A. : B. // C. / D. %

Jawaban: C

### C. Isian Singkat (5 Soal)

1. Lengkapi agar mencetak: **Halo, dunia!**

Jawaban:

```
print("Halo, dunia!")
```

2. Isi variabel agar total = 15.

Jawaban:

```
a=5; b=10; total=a+b
```

3. Tulis satu baris untuk membuat variabel **nama** berisi teks kalian.

Jawaban:

```
nama = "Budi"
```

4. Lengkapi impor *Pandas* standar.

Jawaban:

```
import pandas as pd
```

5. Tulis satu baris *if* yang mencetak "Lulus" jika **nilai**  $\geq 70$ .

Jawaban:

```
if nilai >= 70: print("Lulus")
```

### D. Soal Explorasi (3 Soal)

Tuliskan jawaban dalam 1–3 paragraf pendek per soal. Boleh menyertakan kode pendek.

#### Soal 1:

Kamu diminta membuat program kas kecil kantin sekolah. Program menerima tiga harga barang dan jumlahnya, lalu menampilkan total. Tambahkan diskon 10% jika total  $\geq 50.000$ . Tulis langkahmu dan kode singkat.

**Kunci jawaban eksplorasi (contoh):**

**Langkah:** buat variabel harga & jumlah, hitung subtotal, cek syarat diskon, tampilkan total.

**Kode:**

```
h1,j1=15000,2; h2,j2=10000,1; h3,j3=8000,3
subtotal=h1*j1+h2*j2+h3*j3
diskon=0.1*subtotal if subtotal>=50000 else 0
total=subtotal-diskon
print(total)
```

### Soal 2:

Kamu punya data nilai siswa dalam list, misalnya [80, 65, 90, 72]. Buat ringkasan: rata-rata, nilai tertinggi, nilai terendah. Jika perlu, tunjukkan cara pakai *Pandas*.

**Kunci jawaban eksplorasi (contoh):**

**Langkah:** simpan data, hitung mean, max, min. Versi murni Python dan versi *Pandas*.

**Kode:**

```
nilai=[80,65,90,72]
print(sum(nilai)/len(nilai), max(nilai), min(nilai))

import pandas as pd
s=pd.Series(nilai)
print(s.mean(), s.max(), s.min())
```

### Soal 3:

Buat demo AI mini: klasifikasi "Lulus/Remedial" berdasarkan rata-rata nilai *tugas*, *kuis*, *ujian*. Tidak perlu dataset besar. Tulis ide langkah dan contoh *neural network* mini dengan *Keras*.

**Kunci jawaban eksplorasi (contoh):**

**Langkah:** siapkan data kecil, labelkan  $\geq 70$  sebagai 1, bagi train/test, bangun model sederhana, latih, prediksi.

**Kode:**

```
import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

X=np.array([[80,75,78],[60,58,55],[90,88,91],[68,70,69]])
y=(X.mean(axis=1)>=70).astype(int)

model=Sequential([Dense(8,activation="relu",input_shape=(3,)),
                  Dense(1,activation="sigmoid")])
model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
model.fit(X,y,epochs=50,verbose=0)
pred=model.predict(np.array([[72,68,75]]),verbose=0)[0][0]
print("Lulus" if pred>=0.5 else "Remedial")
```

Selesai. Latihan ini menutup dasar **variabel**, **print()**, operasi, *Pandas* untuk data, serta sentuhan *machine learning* dengan *Keras* dan *neural network*.

## Tugas Proyek (5 Soal)

### Proyek 1: Daftar Kontak Sederhana

**Tugas:** Buat program untuk menyimpan nama dan nomor HP teman, lalu tampilkan semua.

**Langkah:**

- Buat dua variabel: nama dan nomor.
- Simpan dalam *list*.
- Cetak daftar kontak.

**Pseudocode:**

```
SET kontak = []
INPUT nama
INPUT nomor
APPEND (nama, nomor) TO kontak
PRINT kontak
```

**Python (jawaban):**

```
kontak = []
nama = input("Masukkan nama: ")
nomor = input("Masukkan nomor HP: ")
kontak.append((nama, nomor))
print("Daftar Kontak:", kontak)
```

### Proyek 2: Hitung Rata-Rata Nilai Kelas

**Tugas:** Gunakan *Pandas* untuk hitung rata-rata nilai siswa.

**Langkah:**

- Buat file CSV sederhana (nama,nilai).
- Baca file dengan *Pandas*.
- Hitung rata-rata.

**Pseudocode:**

```
IMPORT pandas
READ nilai.csv INTO df
rata2 = MEAN(df["nilai"])
PRINT rata2
```

### Python (jawaban):

```
import pandas as pd

df = pd.DataFrame({
    "nama": ["Ana", "Budi", "Cici"],
    "nilai": [80, 65, 90]
})

print("Rata-rata kelas:", df["nilai"].mean())
```

## Proyek 3: Game Mini “Gunting Batu Kertas”

**Tugas:** Buat game sederhana antara komputer vs pemain.

### Langkah:

- Simpan pilihan: gunting, batu, kertas.
- Minta input pemain.
- Komputer pilih random.
- Tentukan siapa pemenang.

### Pseudocode:

```
CHOICES = ["gunting", "batu", "kertas"]
INPUT pilihan_pemain
SET pilihan_komputer = RANDOM dari CHOICES
IF pemain == komputer THEN seri
ELSE tentukan pemenang
PRINT hasil
```

### Python (jawaban):

```
import random

choices = ["gunting", "batu", "kertas"]
pemain = input("Pilih gunting/batu/kertas: ")
komputer = random.choice(choices)

print("Komputer pilih:", komputer)

if pemain == komputer:
    print("Seri!")
elif (pemain == "batu" and komputer == "gunting") or \
     (pemain == "gunting" and komputer == "kertas") or \
     (pemain == "kertas" and komputer == "batu"):
    print("Kamu Menang!")
else:
    print("Kamu Kalah!")
```

## Proyek 4: Prediksi Cuaca Mini

**Tugas:** Buat *machine learning* mini dengan *if-else* untuk menebak “Hujan” atau “Cerah”.

**Langkah:**

- Simpan data suhu dan kelembaban.
- Jika suhu rendah & kelembaban tinggi → Hujan.
- Jika suhu tinggi & kelembaban rendah → Cerah.

**Pseudocode:**

```
INPUT suhu
INPUT kelembapan
IF suhu < 25 AND kelembapan > 70 THEN "Hujan"
ELSE "Cerah"
```

**Python (jawaban):**

```
suhu = int(input("Masukkan suhu (°C): "))
kelembapan = int(input("Masukkan kelembapan (%): "))

if suhu < 25 and kelembapan > 70:
    print("Prediksi: Hujan 🌧️")
else:
    print("Prediksi: Cerah ☀️")
```

## Proyek 5: Neural Network Mini “Lulus atau Remedial”

**Tugas:** Gunakan *Keras* buat model sederhana prediksi kelulusan.

**Langkah:**

- Buat dataset kecil.
- Tentukan label: lulus = 1, remedial = 0.
- Buat model *neural network*.
- Latih dan prediksi.

**Pseudocode:**

```
IMPORT keras
DATA = nilai siswa
LABEL = lulus/remedial
SPLIT train-test
MODEL = Dense layer (relu, sigmoid)
COMPILE adam, loss bce
FIT data
PREDICT siswa baru
```

**Python (jawaban sederhana):**

```

import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# data sederhana: [tugas, ujian]
X = np.array([[80, 85], [60, 55], [90, 92], [50, 60]])
y = np.array([1, 0, 1, 0]) # 1=lulus, 0=remedial

model = Sequential([
    Dense(4, activation="relu", input_shape=(2,)),
    Dense(1, activation="sigmoid")
])

model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
model.fit(X, y, epochs=50, verbose=0)

pred = model.predict(np.array([[70, 75]]), verbose=0)
print("Prediksi:", "Lulus" if pred[0][0] > 0.5 else "Remedial")

```

### **Kesimpulan:**

5 proyek ini melatih kalian dari dasar (variabel, input-output, *if-else*), ke level data (*Pandas*), sampai sentuhan AI (*Keras*). Semua bisa dicoba di laptop sekolah dengan Python *open source*.

## BAB 4: Struktur Kontrol

Kita bakal main-main sama dunia **Struktur Kontrol** di bab ini. Anggap aja kita lagi jadi sutradara yang mengatur alur cerita film. Kalau di film ada adegan "jika dia menang, dia dapat hadiah," nah, di dunia koding juga ada yang kayak gitu. Penasaran, kan? Yuk, langsung gas!

### Tujuan Pembelajaran: Jadi Sutradara Kode

Di akhir bab ini, kalian bakal jadi **sutradara kode** yang jago. Kalian bisa:

- Membuat program yang bisa **membuat keputusan** sendiri.
- Mengatur program untuk melakukan sesuatu **berulang kali**.
- Menemukan **bug** (kesalahan) di kode dengan cepat.
- Membangun logika berpikir yang keren buat jadi programmer andal.

### Peta Konsep Sempel: Jalan-jalan di Dunia Kode

Bayangin ini peta jalan kalian:

- **Struktur Kontrol**
  - **Percabangan (If-Else)**
    - Kalo gini, lakuin ini. Kalo gitu, lakuin yang lain.
  - **Perulangan (Loop)**
    - Lakuin ini **terus-terusan** sampai syaratnya tercapai.

### Apersepsi: Mesin Fotokopi yang Bingung

Bayangin kamu punya mesin fotokopi canggih. Tapi, dia cuma bisa fotokopi. Kalo kamu mau fotokopi 10 lembar, kamu harus pencet tombolnya 10 kali. Capek, kan? 😩

Nah, itu karena dia nggak punya **struktur kontrol perulangan**. Dengan perulangan, kamu cuma perlu bilang, "Fotokopi 10 kali," dan dia langsung kerja sendiri. Begitu juga sama program. Tanpa struktur kontrol, program kita kayak mesin fotokopi yang bingung.

### Penjelasan Teori: Ngobrolin If dan Loop

Apa itu Struktur Kontrol? (ngerti cepat)

- **Struktur Kontrol** = aturan untuk **memilih** dan **mengulang** tindakan.
- **Percabangan (if-else)**: "Kalau uang cukup, beli es krim. Kalau tidak, beli permen."

- **Perulangan (*loop*)**: “Lakukan hal ini berulang sampai target tercapai.”
- Dipakai di mana-mana: **game**, **kasir**, **absen kelas**, analisis data *Pandas*, bahkan melatih *neural network* di *Keras*.

Contoh super singkat:

```
uang = 8000
if uang >= 5000:
    print("Beli es krim 🍦")
else:
    print("Beli permen 🍬")

for i in range(3):
    print("Latihan ke-", i+1)
```

## Bahan Pokok: Kondisi & Operator (fondasi dulu)

- **Tipe Boolean**: **True** atau **False**.
- **Operator perbandingan**: **==**, **!=**, **>**, **<**, **>=**, **<=**.
- **Operator logika**: **and**, **or**, **not**.

Pola umum:

```
if (umur >= 17) and (punya_sim):
    print("Boleh bawa motor")
```

Kunci: **kondisi harus bisa dinilai benar/salah**.

## Percabangan: *If–Elif–Else* (ambil keputusan)

**Gunanya**: memilih satu jalan yang paling cocok.

- **If**: cek syarat utama.
- **Elif**: pilihan lain.
- **Else**: cadangan ketika semua salah.

Contoh (paket internet):

```
kuota = 1200 # MB
if kuota >= 1024:
    print("Stream HD 🎬")
elif kuota >= 300:
    print("Stream SD 📺")
else:
    print("Hemat kuota dulu 📶")
```

**Tips cepat**:

- **Jaga indentasi** (spasi/tab) di Python.

- Buat **kondisi jelas** dan **singkat**.
- Boleh *ternary* (ringkas satu baris):

```
status = "Lulus" if nilai >= 75 else "Tidak Lulus"
```

## Perulangan: *for* vs *while* (jalan berulang)

**for** → jumlah langkah sudah jelas.

**while** → ulangi selama syarat masih benar.

Contoh **for** (cetak 1–5):

```
for angka in range(1, 6):
    print(angka)
```

Contoh **while** (hitungan mundur):

```
n = 5
while n > 0:
    print("Mulai dalam", n)
    n -= 1
print("GO!")
```

**Kontrol loop:**

- **break** = keluar loop.
- **continue** = lewati langkah saat ini, lanjut berikutnya.

```
for x in range(1, 6):
    if x == 3:
        continue
    if x == 5:
        break
    print(x)  # cetak: 1, 2, 4
```

## Pattern Penting: Validasi & Menu (dipakai tiap hari)

**Validasi input** (biar program gak “bengong”):

```
nilai = -1
while not (0 <= nilai <= 100):
    nilai = int(input("Masukkan nilai 0-100: "))
print("Nilai diterima:", nilai)
```

**Menu sederhana:**

```
while True:
    print("1.Hitung  2.Konversi  3.Keluar")
    pilih = input("Pilih: ")
    if pilih == "1":
        print("Mode Hitung")
    elif pilih == "2":
```

```

        print("Mode Konversi")
    elif pilih == "3":
        print("Bye!")
        break
    else:
        print("Pilihan tidak valid")

```

## Struktur Kontrol + *Pandas* (filter data cepat)

*Pandas* itu *open source* buat data tabel. Kita gabungkan **kondisi** dengan **filter**.

```

import pandas as pd
df = pd.DataFrame({"nama": ["Ana", "Budi", "Cici"],
                  "nilai": [85, 64, 91]})
lulus = df[df["nilai"] >= 75]      # filter dengan kondisi
remed = df[df["nilai"] < 75]
print("Lulus:\n", lulus)
print("Remed:\n", remed)

```

- **Intinya:** Percabangan dipakai untuk **logika**, *Pandas* dipakai untuk **data**.

## Struktur Kontrol + *Machine Learning* (intuisi dulu)

Di *machine learning*, loop dipakai saat **training** (berulang per *epoch*). Percabangan dipakai untuk **cek akurasi**, **simpanan model**, atau **hentikan dini**.

Contoh *intuisi* di *Keras* (sangat ringkas):

```

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([Dense(8, activation="relu",
                        input_shape=(3,)),
                    Dense(1, activation="sigmoid")])
model.compile(optimizer="adam", loss="binary_crossentropy",
              metrics=["accuracy"])

# loop training ada di dalam .fit()
hist = model.fit(X_train, y_train, epochs=20, verbose=0)

# percabangan sederhana setelah training
loss, acc = model.evaluate(X_test, y_test, verbose=0)
print("Model oke") if acc >= 0.8 else print("Perlu perbaikan")

```

- **Garis besar:** *loop* mengulang belajar, *if* memutuskan langkah berikutnya.
- Semua tool ini **open source**.

## Kesalahan Umum & Antitips (supaya tidak terjebak)

- **Lupa update variabel di *while*** → loop tak berakhir.

- **Kondisi kabur** → pecah jadi kondisi kecil, jelas.
- **Indentasi berantakan** → konsisten 4 spasi.
- **Terlalu banyak `elif`** → pertimbangkan struktur data (misal `dict`) atau *mapping*.

## Ringkasan

- **If-Elif-Else** untuk **keputusan**.
- **For-While** untuk **pengulangan**.
- Gunakan **operator** perbandingan dan logika.
- Gabungkan dengan *Pandas* untuk filter data.
- Dalam *machine learning* dengan *Keras*, *loop* terjadi saat training, *if* untuk evaluasi dan aksi lanjutan.
- Semuanya **open source**. Kalian bebas belajar, mencoba, dan berkarya.

## Mini Tantangan (coba dalam 3 menit)

- Buat menu: 1) Cek lulus (`nilai >= 75`), 2) Cetak 5 kali “Semangat!”, 3) Keluar.
- Tambah validasi: jika input bukan 1–3, ulangi tanya.
- Bonus: simpan jumlah percobaan *input* yang salah, cetak di akhir.

Ingat: kalian **sutradara kode**. Atur alur, pilih adegan, ulangi yang perlu. Dengan **Struktur Kontrol**, program jadi **cerdas** dan **rapi**. 💪

## Contoh Kasus atau Ilustrasi

### Kasus 1: Kantin Diskon & Member

#### Ilustrasi:

Kamu belanja di kantin. Ada diskon berbeda. Kalau total besar dan kamu *member*, diskonnya lebih besar. Program harus memilih dengan *if-elif-else*.

#### Belajar apa:

- Percabangan *if-elif-else*.
- Operator logika: `and`.
- Hitung total bayar.

#### Pseudocode:

```
INPUT total_belanja
INPUT is_member    # True/False

IF total_belanja >= 50000 AND is_member == True THEN
    diskon = 0.15 * total_belanja
ELIF total_belanja >= 50000 THEN
    diskon = 0.10 * total_belanja
ELIF total_belanja >= 30000 THEN
```

```

        diskon = 0.05 * total_belanja
ELSE
    diskon = 0
END IF

total_bayar = total_belanja - diskon
PRINT "Diskon:", diskon
PRINT "Total Bayar:", total_bayar

```

## Kasus 2: Absen Kelas & Remedial

### Ilustrasi:

Wali kelas ingin menandai **Lulus** atau **Remedial** untuk tiap siswa. Kita *loop* data nilai. Aturan sederhana: nilai  $\geq 75 \rightarrow$  Lulus.

### Belajar apa:

- *For loop* untuk daftar.
- Percabangan *if-else*.
- Ringkasan data (mirip *Pandas* filter).

### Pseudocode:

```

INPUT daftar_nama
INPUT daftar_nilai  # paralel dengan daftar_nama

lulus = []
remed = []

FOR i FROM 0 TO LENGTH(daftar_nama)-1 DO
    IF daftar_nilai[i] >= 75 THEN
        APPEND daftar_nama[i] TO lulus
    ELSE
        APPEND daftar_nama[i] TO remed
    END IF
END FOR

PRINT "Lulus:", lulus
PRINT "Remedial:", remed
PRINT "Jumlah Lulus:", LENGTH(lulus)
PRINT "Jumlah Remedial:", LENGTH(remed)

```

Catatan: di *Python*, ini bisa dibantu *Pandas* (semua *open source*) untuk memfilter lebih cepat.

### Kasus 3: Game Koin: Hati-hati Bom

#### Ilustrasi:

Kamu kumpulkan koin sampai 100. Kalau ketemu **bom**, berhenti. Kalau ketemu **rintangan kecil**, lewati saja. Ini latihan *while loop*, *break*, dan *continue*.

#### Belajar apa:

- *While loop*.
- *break* untuk keluar.
- *continue* untuk lewati satu langkah.

#### Pseudocode:

```
koin = 0
posisi = 0

WHILE koin < 100 DO
    event = CEK_EVENT(posisi) # "koin", "rintangan", "bom",
    "kosong"

    IF event == "bom" THEN
        PRINT "Game Over"
        BREAK
    ELIF event == "rintangan" THEN
        posisi = posisi + 1
        CONTINUE
    ELIF event == "koin" THEN
        koin = koin + 10
    END IF

    posisi = posisi + 1
    PRINT "Koin:", koin
END WHILE

IF koin >= 100 THEN
    PRINT "Menang!"
END IF
```

### Kasus 4: Parkir Cerdas: Slot Terbatas

#### Ilustrasi:

Lahan parkir punya **slot\_mobil** dan **slot\_motor**. Kendaraan datang berbaris. Kalau slot sesuai masih ada, izinkan masuk. Kalau slot habis, tolak. Ini contoh *nested if* di dalam *loop*.

#### Belajar apa:

- *For/While loop* untuk antrean.
- Percabangan bersarang (*nested if*).

- Logika kapasitas.

#### Pseudocode:

```

INPUT antrian_kendaraan # contoh:
["motor","mobil","motor",...]
INPUT slot_motor
INPUT slot_mobil

FOR setiap jenis IN antrian_kendaraan DO
  IF jenis == "motor" THEN
    IF slot_motor > 0 THEN
      slot_motor = slot_motor - 1
      PRINT "Motor masuk. Sisa slot motor:", slot_motor
    ELSE
      PRINT "Parkir motor penuh"
    END IF
  ELIF jenis == "mobil" THEN
    IF slot_mobil > 0 THEN
      slot_mobil = slot_mobil - 1
      PRINT "Mobil masuk. Sisa slot mobil:", slot_mobil
    ELSE
      PRINT "Parkir mobil penuh"
    END IF
  ELSE
    PRINT "Jenis tidak dikenal. Lewati."
  END IF
END FOR

PRINT "Sisa slot motor:", slot_motor
PRINT "Sisa slot mobil:", slot_mobil

```

### Kasus 5: Mini ML Training Loop dengan Keras (Konsep)

#### Ilustrasi:

Kamu melatih *neural network* kecil pakai *Keras*. Latihan berjalan dalam banyak *epoch*. Kalau akurasi makin baik, **simpan model**. Kalau tidak membaik 3 kali berturut-turut, **hentikan** (*early stopping*). Ini contoh struktur kontrol di *machine learning*.

#### Belajar apa:

- *Loop* untuk *epoch*.
- *If-else* untuk keputusan simpan/berhenti.
- Konsep *open source* AI workflow.

#### Pseudocode:

```

best_acc = 0
tidak_membaik = 0
max_epoch = 20
batas_gagal = 3

```

```

FOR epoch FROM 1 TO max_epoch DO
    LATIH_MODEL_SESATU_EPOCH()          # model.fit(..., epochs=1)
    (loss, acc) = EVALUASI_VALIDASI()    # model.evaluate(...)

    PRINT "Epoch:", epoch, "Acc:", acc

    IF acc > best_acc THEN
        best_acc = acc
        SIMPAN_MODEL("model_terbaik.h5")
        tidak_membalik = 0
    ELSE
        tidak_membalik = tidak_membalik + 1
    END IF

    IF tidak_membalik >= batas_gagal THEN
        PRINT "Early stopping: akurasi tidak membalik"
        BREAK
    END IF
END FOR

PRINT "Akurasi terbaik:", best_acc

```

Di implementasi nyata, *loop* ini terjadi otomatis di `.fit()`, tapi pola kontrol seperti ini sering dipakai untuk *callback* dan keputusan setelah pelatihan. Semua toolnya *open source*.

## Penutup singkat

- **Percabangan** *if-elif-else* = ambil keputusan.
- **Perulangan** *for/while* = ulangi tugas sampai target.
- Pola ini dipakai di kantin, absen, parkir, game, sampai *machine learning* dengan *Keras* dan *neural network*.
- Terus coba. *Debug*. Ulangi. Itulah cara jadi **sutradara kode** yang andal.

## Contoh Soal dan Pembahasan

### Soal 1: Kantin Cerdas: Diskon + Voucher

#### Ilustrasi:

Kamu belanja di kantin. Ada diskon bertingkat. Ada *member*. Ada kode voucher. Program harus memutuskan dengan adil.

#### Tujuan:

- Melatih *if-elif-else* dan operator *and/or*.
- Hitung total bersih.

**Konsep yang dilatih:** *percabangan*, operator logika, aritmetika.

**Pseudocode:**

```
INPUT total_belanja
INPUT is_member      # True/False
INPUT kode_voucher   # teks atau kosong

diskon = 0

IF total_belanja >= 50000 AND is_member == True THEN
    diskon = 0.15 * total_belanja
ELIF total_belanja >= 50000 THEN
    diskon = 0.10 * total_belanja
ELIF total_belanja >= 30000 THEN
    diskon = 0.05 * total_belanja
END IF

IF kode_voucher == "HEMAT5" OR kode_voucher == "SMK5" THEN
    diskon = diskon + 0.05 * total_belanja
END IF

total_bayar = total_belanja - diskon
PRINT "Diskon:", diskon
PRINT "Total Bayar:", total_bayar
```

## Soal 2: Mesin Fotokopi Pintar (Anti Capek)

**Ilustrasi:**

Kamu mau fotokopi 20 lembar. Kadang kertas habis. Kadang mesin panas. Pakai *loop* biar otomatis. Gunakan *continue* dan *break* buat kondisi khusus.

**Tujuan:**

- Memakai *while loop*.
- Menghentikan proses saat darurat.

**Konsep yang dilatih:** *loop*, *break*, *continue*.

**Pseudocode:**

```
INPUT target_lembar    # misal 20
lembar = 0
kertas = 50
panas = False

WHILE lembar < target_lembar DO
    IF kertas == 0 THEN
        PRINT "Kertas habis. Isi ulang!"
        BREAK
    END IF
    lembar = lembar + 1
    IF panas == True THEN
        PRINT "Mesin panas. Tunggu sebentar!"
        CONTINUE
    END IF
    kertas = kertas - 1
END WHILE
```

```

IF panas == True THEN
    PRINT "Mesin panas. Tunggu 1 menit."
    # simulasi pendinginan, lalu lanjut
    panas = False
    CONTINUE
END IF

PRINT "Fotokopi 1 lembar..."
lembar = lembar + 1
kertas = kertas - 1

IF lembar % 10 == 0 THEN
    panas = True
END IF
END WHILE

PRINT "Selesai:", lembar, "lembar"

```

### Soal 3: Rekap Nilai Kelas dengan *Pandas* (Filter Cepat)

#### Ilustrasi:

Kamu punya data nilai di CSV. Kamu ingin daftar **Lulus** ( $\geq 75$ ) dan **Remedial**. *Pandas* itu *open source* dan enak buat tabel.

#### Tujuan:

- Memakai logika *if* untuk label.
- Menyaring data seperti "*if di data*".

**Konsep yang dilatih:** *if-else*, *loop* baris, filter *Pandas*.

#### Pseudocode:

```

IMPORT pandas AS pd
df = READ_CSV("nilai.csv")      # kolom: nama, nilai

# Tambah kolom status pakai kondisi
FOR setiap baris IN df DO
    IF baris.nilai >= 75 THEN
        baris.status = "Lulus"
    ELSE
        baris.status = "Remedial"
    END IF
END FOR

lulus = FILTER df WHERE status == "Lulus"
remed = FILTER df WHERE status == "Remedial"

PRINT "Jumlah Lulus:", COUNT(lulus)
PRINT "Jumlah Remedial:", COUNT(remed)
SAVE_CSV(df, "rekap_nilai.csv")

```

Catatan: di *Python* asli, filter ini bisa 1 baris dengan `df[df["nilai"]>=75]`. Keren dan cepat.

#### Soal 4: Lampu Lalu Lintas Cerdas (Respon Ambulans)

##### Ilustrasi:

Lampu berputar: Merah → Kuning → Hijau. Tapi kalau ada ambulans, lampu harus jadi hijau segera. Ini kombinasi *loop* + *if*.

##### Tujuan:

- Siklus *loop* tanpa henti.
- Percabangan prioritas darurat.

**Konsep yang dilatih:** *while True*, *if*, urutan kontrol.

##### Pseudocode:

```
WHILE True DO
  IF AMBULANS_DATANG() == True THEN
    SET_LAMPU("Hijau", 8)  # ijinan lewat
    CONTINUE
  END IF

  SET_LAMPU("Merah", 5)
  IF AMBULANS_DATANG() == True THEN
    SET_LAMPU("Hijau", 8)
    CONTINUE
  END IF

  SET_LAMPU("Kuning", 2)
  SET_LAMPU("Hijau", 5)
END WHILE
```

Fungsi `SET_LAMPU(warna, detik)` membalik warna dan menunggu *detik*. Cek darurat di titik aman.

#### Soal 5: Mini ML Training Loop dengan Keras: Early Stopping

##### Ilustrasi:

Kamu melatih *neural network* kecil pakai Keras (semua *open source*). Kalau akurasi validasi membaik, **simpan model**. Kalau jelek terus 3 kali, **berhenti** cepat.

##### Tujuan:

- Paham *loop* pada pelatihan.
- Keputusan dengan *if* setelah tiap *epoch*.

**Konsep yang dilatih:** *loop*, *if-else*, *early stopping*, *checkpoint*.

### Pseudocode:

```
best_acc = 0
gagal_beruntun = 0
batas_gagal = 3
max_epoch = 25

INISIALISASI_MODEL_KERAS()

FOR epoch FROM 1 TO max_epoch DO
    LATIH_SE_EPOCH() # model.fit(..., epochs=1)
    (loss, acc) = EVALUASI_VALIDASI() # model.evaluate(...)

    IF acc > best_acc THEN
        best_acc = acc
        SIMPAN_MODEL("best_model.h5")
        gagal_beruntun = 0
    ELSE
        gagal_beruntun = gagal_beruntun + 1
    END IF

    PRINT "Epoch", epoch, "Akurasi:", acc, "Terbaik:", best_acc

    IF gagal_beruntun >= batas_gagal THEN
        PRINT "Early stopping: tidak membaik 3 kali."
        BREAK
    END IF
END FOR

PRINT "Akurasi terbaik:", best_acc
```

Intinya: *loop* mengulang belajar, *if* memutuskan aksi. Ini pola umum di *machine learning* dengan *Keras* dan *neural network*.

### Penutup singkat

- **If–Else** untuk memilih. **Loop** untuk mengulang.
- Dipakai di kantin, mesin fotokopi, rekap nilai *Pandas*, lampu lalu lintas, hingga latihan *AI* mini.
- Semua toolnya *open source*. Coba, gagal, perbaiki, ulangi. Itulah jalan jadi **sutradara kode** yang jago.

## Fun Facts

- Awalnya, bahasa pemrograman kuno tidak punya **struktur kontrol** semudah sekarang. Programmer harus pakai `goto`, yang bikin kode jadi berantakan kayak benang kusut!
- **Ternary Operator** di Python itu versi super ringkas dari `if-else`. Cuma satu baris, lho! Contoh: `status = "Lulus" if nilai >= 75 else "Tidak Lulus"`

## Tips Belajar & Menghafal Cepat

- **Analogi Keren:** Hubungkan konsep `if-else` dengan "jika-maka" di kehidupan sehari-hari. Hubungkan `loop` dengan kegiatan berulang (nyiram bunga, olahraga).
- **Nulis Kode di Kertas:** Sebelum ngetik di komputer, coba buat alur logikanya dulu di kertas. Ini melatih otakmu berpikir secara terstruktur.
- **Diskusi Bareng:** Ajak temanmu main tebak-tebakan logika. "Kalau aku jajan 5 ribu, tapi uangku 10 ribu, sisa berapa? Pakai `if-else` ya!"
- **Challenge Kecil:** Coba buat program sederhana. Misalnya, program yang nanya namamu terus nge-print "Halo (namamu), selamat datang di program pertamaku!"
- **Main Game:** Banyak game edukasi yang bisa melatih logika pemrograman. Coba cari game tentang **block programming** atau teka-teki logika.

## Aktivitas Siswa

### Aktivitas 1: ATM Mini: Menu Interaktif

#### Ilustrasi:

Kamu punya ATM mini. Bisa **cek saldo**, **setor**, **tarik**, dan **keluar**. Kamu atur alur dengan `if-elif-else` dan `loop`.

**Belajar:** `while loop`, `if-elif-else`, `break`.

#### Langkah:

- Tampilkan menu.
- Baca pilihan.
- Jalankan aksi.
- Ulangi sampai pilih "keluar".

#### Pseudocode:

```
saldo = 0
WHILE True DO
    TAMPILKAN menu
    INPUT pilih
```

```

IF pilih == "1" THEN PRINT saldo
ELSEIF pilih == "2" THEN INPUT setor; saldo = saldo + setor
ELSEIF pilih == "3" THEN INPUT tarik;
    IF tarik <= saldo THEN saldo = saldo - tarik
    ELSE PRINT "Saldo kurang"
ELSEIF pilih == "4" THEN PRINT "Keluar"; BREAK
ELSE PRINT "Pilihan tidak valid"
END WHILE

```

### Python (jawaban):

```

saldo = 0
while True:
    print("\n=== ATM Mini ===")
    print("1. Cek Saldo\n2. Setor\n3. Tarik\n4. Keluar")
    pilih = input("Pilih menu: ")

    if pilih == "1":
        print("Saldo:", saldo)
    elif pilih == "2":
        setor = int(input("Jumlah setor: "))
        saldo += setor
        print("Setor sukses. Saldo:", saldo)
    elif pilih == "3":
        tarik = int(input("Jumlah tarik: "))
        if tarik <= saldo:
            saldo -= tarik
            print("Tarik sukses. Saldo:", saldo)
        else:
            print("Saldo kurang.")
    elif pilih == "4":
        print("Keluar. Terima kasih.")
        break
    else:
        print("Pilihan tidak valid.")

```

## Aktivitas 2: Password Checker: Aman atau Tidak

### Ilustrasi:

Buat *checker* yang minta password berulang sampai **valid**. Aturan: minimal 6 karakter dan punya angka.

**Belajar:** *while loop, if-else, continue.*

### Langkah:

- Minta password.
- Cek panjang.
- Cek ada angka.
- Kalau belum valid, minta ulang.

**Pseudocode:**

```
WHILE True DO
    INPUT pw
    IF LENGTH(pw) < 6 THEN PRINT "Terlalu pendek"; CONTINUE
    IF TIDAK_ADA_ANGKA(pw) THEN PRINT "Harus ada angka";
CONTINUE
    PRINT "Password valid"
    BREAK
END WHILE
```

**Python (jawaban):**

```
while True:
    pw = input("Masukkan password: ")
    if len(pw) < 6:
        print("Terlalu pendek.")
        continue
    if not any(ch.isdigit() for ch in pw):
        print("Harus mengandung angka.")
        continue
    print("Password valid ✅")
    break
```

**Aktivitas 3: Rekap Nilai dengan Pandas: Lulus/Remedial****Ilustrasi:**

Kamu punya *dataset* nilai. Tandai **Lulus** ( $\geq 75$ ) atau **Remedial**. Tampilkan ringkasannya. *Pandas* itu *open source*.

**Belajar:** logika kondisi pada data, *loop* baris (konsep), filter *Pandas*.

**Langkah:**

- Siapkan CSV: *nama, nilai*.
- Baca dengan *Pandas*.
- Buat kolom *status* memakai kondisi.
- Hitung jumlah tiap status.

**Pseudocode:**

```
df = READ_CSV("nilai.csv")
FOR tiap_baris IN df DO
    IF nilai >= 75 THEN status = "Lulus" ELSE status =
"Remedial"
END FOR
PRINT hitung status
SAVE_CSV(df, "rekap.csv")
```

**Python (jawaban):**

```
import pandas as pd

# contoh data langsung (boleh ganti baca CSV:
pd.read_csv("nilai.csv"))
df = pd.DataFrame({
    "nama": ["Ana", "Budi", "Cici", "Doni", "Eka"],
    "nilai": [85, 64, 91, 72, 77]
})
df["status"] = df["nilai"].apply(lambda n: "Lulus" if n >= 75
else "Remedial")
print(df)
print("\nRingkasan:")
print(df["status"].value_counts())
# df.to_csv("rekap.csv", index=False)
```

## Aktivitas 4: Jadwal Kelas & Istirahat: Cetak Otomatis

### Ilustrasi:

Kamu cetak jadwal 1–8. Jam ke-4 dan 7 adalah **Istirahat**. Lainnya **Belajar**. Gunakan *loop* dan *if*.

**Belajar:** *for loop, if–elif–else*.

### Langkah:

- *Loop* dari 1 ke 8.
- Jika 4 atau 7 → “Istirahat”.
- Selain itu → “Belajar”.
- Cetak rapi.

### Pseudocode:

```
FOR jam FROM 1 TO 8 DO
    IF jam == 4 OR jam == 7 THEN
        PRINT jam, "Istirahat"
    ELSE
        PRINT jam, "Belajar"
    END IF
END FOR
```

### Python (jawaban):

```
for jam in range(1, 9):
    if jam in (4, 7):
        print(f"Jam {jam}: Istirahat 🍽️")
    else:
        print(f"Jam {jam}: Belajar 📖")
```

## Aktivitas 5: *Mini ML* Sentuhan Struktur Kontrol: Prediksi Lulus

### Ilustrasi:

Kita bikin *neural network* kecil pakai *Keras* (semua *open source*). Data mini: tugas, kuis, ujian. Setelah latih, pakai *if* untuk memutuskan **Lulus** atau **Remedial**.

**Belajar:** *loop* pelatihan (di dalam `.fit()`), *if* untuk keputusan prediksi. Konsep *machine learning*, *Keras*, *neural network*.

### Langkah:

- Siapkan data kecil.
- Bangun model *Keras*.
- Latih singkat.
- Prediksi satu siswa.
- Jika skor  $\geq 0.5 \rightarrow$  "Lulus", else "Remedial".

### Pseudocode:

```
BANGUN_MODEL_DENSE()  
COMPILE adam, binary_crossentropy  
FIT data sebentar  
pred = PREDICT(data_baru)  
IF pred >= 0.5 THEN "Lulus" ELSE "Remedial"
```

### Python (jawaban, contoh minimal):

```
# Pastikan sudah install: pip install tensorflow numpy  
import numpy as np  
from tensorflow.keras import Sequential  
from tensorflow.keras.layers import Dense  
  
# data mini: [tugas, kuis, ujian]  
X = np.array([[80,75,78],[60,58,55],[90,88,91],[68,70,69]],  
dtype=float)  
y = (X.mean(axis=1) >= 70).astype(int) # label: 1 lulus, 0  
remedial  
  
model = Sequential([  
    Dense(8, activation="relu", input_shape=(3,)),  
    Dense(1, activation="sigmoid")  
)  
model.compile(optimizer="adam", loss="binary_crossentropy",  
metrics=["accuracy"])  
model.fit(X, y, epochs=50, verbose=0)  
  
baru = np.array([[72,68,75]], dtype=float)  
pred = model.predict(baru, verbose=0)[0][0]  
print("Skor:", round(float(pred), 3))  
print("Prediksi:", "Lulus" if pred >= 0.5 else "Remedial")
```

## Catatan Akhir

- **Struktur kontrol** itu senjata utama: *if-else* untuk **keputusan**, *loop* untuk **ulang tugas**.
- Bisa dipakai di **menu ATM**, **validasi password**, **rekap data dengan Pandas**, **jadwal kelas**, hingga **demo machine learning** dengan *Keras* dan *neural network*.
- Coba, ulangi, dan eksperimen. Itulah cara jadi **sutradara kode** yang jago.

## Rangkuman

Struktur kontrol itu sutradara di program. Ia memilih adegan dan mengatur ulang-ulangnya. **Dua tokoh utama: *if-else* (memilih) dan *loop* (mengulang)**. Dengan *if-else*, program **membuat keputusan**. Dengan *loop*, program **bekerja berulang** sampai syarat tercapai. Bahasa yang kita pakai **Python**. Mudah dibaca. Gratis karena *open source*. Kalian cukup tulis logika sederhana, komputer yang eksekusi.

Di dunia nyata, konsep ini dipakai di mana-mana. **Kantin**: diskon kalau belanja besar. **Absen**: panggil nama satu per satu dengan *loop*. **Parkir**: penuh atau masih ada, pakai *if*. Untuk data, kita bisa pakai *Pandas* (juga *open source*) untuk **filter data seperti “if di tabel”**. Kalau mau naik level, sentuhkan ke *machine learning* dengan *Keras* dan *neural network*: **training itu berulang (*loop*)**, **keputusan simpan model pakai *if***. Semua toolnya **gratis dan terbuka**.

Cara belajar yang mantap itu sederhana. **Mulai kecil → coba → lihat hasil → perbaiki**. Jaga **indentasi** biar rapi. Hindari *infinite loop* dengan **syarat yang jelas**. Catat pola umum: **cek kondisi → lakukan aksi → ulang jika perlu**. Kolaborasi bareng teman, uji kasus nyata, dan eksplor *open source* tools. **Kuasai *if-else* dan *loop***, maka kode kalian jadi **lebih cerdas, rapi, dan siap dipakai proyek beneran**.

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. *If-else* dipakai untuk membuat program memilih satu aksi dari beberapa kondisi.  
Jawaban: Benar
2. *For loop* dengan `range(5)` akan mengulang sebanyak 5 kali (0 sampai 4).  
Jawaban: Benar
3. *While loop* tidak butuh syarat berhenti, karena akan otomatis berhenti sendiri.  
Jawaban: Salah
4. Indentasi (spasi/tab) di Python tidak berpengaruh pada blok kode.  
Jawaban: Salah

5. *Pandas* adalah pustaka *open source* untuk mengolah data tabel.

Jawaban: Benar

## B. Pilihan Ganda (10 Soal)

1. Output kode berikut?

```
uang = 4000
if uang >= 5000:
    print("Es krim")
else:
    print("Permen")
```

- A. Es krim
- B. Permen
- C. Tidak cetak apa-apa
- D. Error

Jawaban: B

2. Output?

```
for i in range(1,4):
    print(i, end=" ")
```

- A. 0 1 2
- B. 1 2 3
- C. 1 2 3 4
- D. 2 3 4

Jawaban: B

3. Nilai **total**?

```
total = 0
for i in range(1,6):
    total += i
print(total)
```

- A. 10
- B. 12
- C. 15
- D. 20

Jawaban: C

4. Berapa kali "Hi" tercetak?

```
x = 0
while x < 3:
    print("Hi")
    x += 1
```

- A. 1
- B. 2
- C. 3
- D. 4

Jawaban: C

5. Output?

```
nilai = 74
status = "Lulus" if nilai >= 75 else "Tidak Lulus"
print(status)
```

- A. Lulus
- B. Tidak Lulus
- C. None
- D. Error

Jawaban: B

6. Output?

```
for n in range(1,6):
    if n == 3:
        break
    print(n, end=" ")
```

- A. 1 2 3 4 5
- B. 1 2
- C. 3 4 5
- D. 2 3

Jawaban: B

7. Output?

```
for i in range(1,6):
    if i % 2 == 0:
        continue
    print(i, end=" ")
```

- A. 1 2 3 4 5
- B. 2 4
- C. 1 3 5
- D. 3 5

Jawaban: C

8. Sintaks `elif` yang benar adalah...

- A. `else if x > 5:`
- B. `elif x > 5:`
- C. `elseif x > 5:`
- D. `else: if x > 5`

Jawaban: B

9. Hasil ekspresi?

```
(5 > 3 and 2 < 1) or (not False)
```

- A. True
- B. False
- C. 0
- D. Error

Jawaban: A

10. Pilih baris *Pandas* yang memberi label "Lulus" jika `nilai >= 75`.

- A. `df["status"] = "Lulus" if df["nilai"] >= 75 else "Remedial"`
- B. `df.apply(lambda n: "Lulus" if n >= 75 else "Remedial")`
- C. `df["status"] = df["nilai"].apply(lambda n: "Lulus" if n >= 75 else "Remedial")`
- D. `df["status"] == "Lulus" if df["nilai"] >= 75`

Jawaban: C

### C. Isian Singkat (5 Soal)

1. Lengkapi agar mencetak 0 1 2 3 4:

```
for i in _____:  
    print(i)
```

Jawaban: `range(5)`

2. Isi kata kunci yang hilang:

```
if nilai >= 75:  
    print("Lulus")  
_____  
    print("Tidak Lulus")
```

Jawaban: `else`

3. Biar *while* tidak *infinite loop*, lengkapi:

```
x = 0  
while x < 4:  
    print(x)  
    _____
```

Jawaban: `x += 1`

4. Lengkapi kondisi diskon 10% jika total > 50000:

```
if _____:
```

```
diskon = total * 0.1
```

Jawaban: `total > 50000`

5. Lengkapi agar mencetak angka ganjil 1..9:

```
for i in range(1, 10):  
    if i % 2 == 0:  
        _____  
    print(i)
```

Jawaban: `continue`

## D. Soal Explorasi (3 Soal)

Soal 1:

**Rancang *pseudocode* dan Python pendek untuk “Alarm Belajar”:** Ulangi hitung menit dari 1 sampai 30. Tiap kelipatan 10 cetak “Istirahat 1 menit”, lainnya “Belajar”.  
Contoh Jawaban:

```
FOR m = 1..30  
    IF m % 10 == 0 THEN PRINT "Istirahat 1 menit"  
    ELSE PRINT "Belajar"  
END FOR  
  
for m in range(1, 31):  
    print("Istirahat 1 menit" if m % 10 == 0 else "Belajar")
```

Soal 2:

**Buat ringkasan suhu kelas dengan *Pandas*:** Dari list `[28, 31, 35, 29, 33]`, beri label: `<30 Aman`, `30–34 Waspada`, `>=35 Bahaya`. Cetak jumlah tiap label.  
Contoh Jawaban:

```
import pandas as pd  
suhu = [28, 31, 35, 29, 33]  
df = pd.DataFrame({"suhu": suhu})  
df["status"] = df["suhu"].apply(lambda x: "Bahaya" if x>=35 else  
    ("Waspada" if x>=30 else "Aman"))  
print(df["status"].value_counts())
```

Soal 3:

**Mini “AI” dengan *Keras* (opsional, sangat sederhana):** Data kecil `[[80, 70], [60, 50], [90, 88]]` (tugas, kuis) dengan label lulus=1 jika rata-rata  $\geq 70$ , lainnya 0. Latih cepat dan prediksi satu sampel `[75, 72]`.  
Contoh Jawaban:

```
# pip install tensorflow
```

```

import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

X = np.array([[80,70],[60,50],[90,88]], float)
y = (X.mean(axis=1) >= 70).astype(int)
model = Sequential([Dense(4, activation="relu",
input_shape=(2,)),
                    Dense(1, activation="sigmoid")])
model.compile(optimizer="adam", loss="binary_crossentropy")
model.fit(X, y, epochs=50, verbose=0)
pred = float(model.predict(np.array([[75,72]]),
verbose=0)[0][0])
print("Prediksi:", "Lulus" if pred >= 0.5 else "Remedial")

```

Selamat latihan! Kuasai **struktur kontrol** (*if-else*, *loop*, *break*, *continue*), lalu sambungkan dengan data pakai *Pandas* dan sentuhan *AI* mini pakai *Keras/neural network* bila perlu.

## Tugas Proyek (5 Soal)

### Proyek 1: Kalkulator Ongkir Nusantara

**Tujuan:** hitung ongkir dengan aturan bertingkat. Latih *if-elif-else*, *loop*, dan *break*.

#### Aturan singkat:

- Tarif dasar: Rp10.000/kg + Rp200/km.
- *Layanan*: **reg** (×1.0), **express** (×1.5).
- *Member*: potongan 5%.
- *Berat* > 5 kg → biaya tambahan 10% dari tarif.

#### Langkah:

- Ulangi input paket sampai pengguna ketik berat **0**.
- Hitung tarif sesuai aturan.
- Tampilkan total, lalu lanjut ke paket berikutnya.

#### Pseudocode:

```

WHILE True
    INPUT berat
    IF berat == 0 THEN BREAK
    INPUT jarak
    INPUT layanan    # reg/express
    INPUT member     # y/n

    tarif = (10000*berat) + (200*jarak)
    IF berat > 5 THEN tarif = tarif * 1.10
    IF layanan == "express" THEN tarif = tarif * 1.5

```

```

IF member == "y" THEN tarif = tarif * 0.95

PRINT tarif
END WHILE

```

### Python (jawaban):

```

while True:
    berat = float(input("Berat kg (0 untuk selesai): "))
    if berat == 0:
        print("Selesai.")
        break
    jarak = int(input("Jarak km: "))
    layanan = input("Layanan (reg/express): ").lower()
    member = input("Member? (y/n): ").lower()

    tarif = (10000 * berat) + (200 * jarak)
    if berat > 5:
        tarif *= 1.10
    if layanan == "express":
        tarif *= 1.5
    if member == "y":
        tarif *= 0.95

    print("Total ongkir: Rp", int(tarif))

```

## Proyek 2: Simulasi Gacha Skin Game

**Tujuan:** *loop*, *break*, *random*, hitung hasil *pull*.

**Peluang:** *Legendary* 5%, *Rare* 25%, *Common* 70%.

**Aturan:** lakukan 10 *pull* atau berhenti lebih cepat jika dapat *Legendary*.

### Langkah:

- Ulangi *pull* maksimum 10 kali.
- Acak hasil sesuai peluang.
- Jika *Legendary* muncul, hentikan.
- Cetak ringkasan jumlah tiap raritas.

### Pseudocode:

```

legend=0; rare=0; common=0
FOR i = 1..10
    r = RANDOM(0..1)
    IF r < 0.05 THEN legend+=1; PRINT "Legendary!"; BREAK
    ELIF r < 0.30 THEN rare+=1
    ELSE common+=1
END FOR
PRINT legend, rare, common

```

### Python (jawaban):

```

import random
legend = rare = common = 0

for _ in range(10):
    r = random.random()
    if r < 0.05:
        legend += 1
        print("Dapet Legendary! 🏆")
        break
    elif r < 0.30:
        rare += 1
    else:
        common += 1

print(f"Ringkasan → Legendary:{legend}, Rare:{rare},
Common:{common}")

```

### Proyek 3: Sensor Suhu Kelas dengan *Pandas*

**Tujuan:** klasifikasi *Aman/Waspada/Bahaya* dari daftar suhu. Latih *if* pada data dan ringkasan cepat dengan *Pandas*.

#### Aturan label:

- < 30°C → **Aman**
- 30–34°C → **Waspada**
- ≥ 35°C → **Bahaya** (jika ≥2 kali, sarankan matikan AC/kipas lebih besar)

#### Langkah:

- Buat daftar suhu (atau baca dari CSV).
- Ubah ke *DataFrame*.
- Buat kolom *status* dengan kondisi.
- Hitung jumlah tiap status.
- Jika *Bahaya* ≥ 2, cetak peringatan.

#### Pseudocode:

```

df = DataFrame(suhu_list)
FOR tiap suhu
    IF suhu >= 35 THEN status="Bahaya"
    ELIF suhu >= 30 THEN status="Waspada"
    ELSE status="Aman"
END FOR
hitung = COUNT_BY(status)
IF hitung["Bahaya"] >= 2 THEN PRINT "Matikan AC/kurangi panas!"
PRINT hitung

```

#### Python (jawaban):

```

import pandas as pd

suhu = [28, 31, 33, 29, 36, 35, 32]
df = pd.DataFrame({"suhu": suhu})
def label(x):
    if x >= 35: return "Bahaya"
    elif x >= 30: return "Waspada"
    else: return "Aman"

df["status"] = df["suhu"].apply(label)
print(df)
ringkas = df["status"].value_counts()
print("\nRingkasan:\n", ringkas)
if ringkas.get("Bahaya", 0) >= 2:
    print("PERINGATAN: Turunkan suhu ruangan sekarang!")

```

## Proyek 4: Cetak Pola “Kotak + X”

**Tujuan:** *nested loop* dan *if* bersyarat. Cetak pola ukuran  $n \times n$ : garis tepi + diagonal X.

**Langkah:**

- Minta  $n$  (misal 7).
- Loop baris dan kolom.
- Tulis `*` untuk tepi atau  $i==j$  atau  $i+j==n-1$ .
- Lainnya spasi.

**Pseudocode:**

```

INPUT n
FOR i=0..n-1
    baris = ""
    FOR j=0..n-1
        IF i==0 OR i==n-1 OR j==0 OR j==n-1 OR i==j OR i+j==n-1
            baris += "*"
        ELSE
            baris += " "
    END FOR
    PRINT baris
END FOR

```

**Python (jawaban):**

```

n = int(input("Ukuran n: "))
for i in range(n):
    row = ""
    for j in range(n):
        if i in (0, n-1) or j in (0, n-1) or i == j or i + j ==
n - 1:
            row += "*"

```

```

else:
    row += " "
print(row)

```

## Proyek 5: Detektor Chat “Spam” Sederhana

**Tujuan:** aturan *rule-based* pakai *if-else*, *loop*, *continue*, *break*.

**Aturan:**

- *Spam* jika panjang > 200 **atau** mengandung kata terlarang **atau** !!! berulang.
- Ketik **STOP** untuk selesai.
- Cetak ringkasan jumlah *spam* dan *oke*.

**Langkah:**

- Siapkan daftar kata terlarang.
- Baca pesan berulang.
- Cek aturan. Tandai *Spam/Oke*.
- Simpan hitungan.
- Selesai saat **STOP**.

**Pseudocode:**

```

spam=0; oke=0
banned = ["gratis", "klik", "hadiah"]
WHILE True
    msg = INPUT()
    IF msg == "STOP" THEN BREAK
    IF len(msg)>200 OR "!!!" in msg OR
    ADA_KATA_TERLARANG(msg,banned)
        PRINT "Spam"; spam+=1; CONTINUE
    PRINT "Oke"; oke+=1
END WHILE
PRINT spam, oke

```

**Python (jawaban):**

```

banned = {"gratis", "klik", "hadiah"}
spam = oke = 0

while True:
    msg = input("Pesan (STOP untuk selesai): ")
    if msg == "STOP":
        break
    lower = msg.lower()
    ada_banned = any(k in lower for k in banned)
    if len(msg) > 200 or "!!!" in msg or ada_banned:
        print("Spam 🚫")
        spam += 1
    else:
        oke += 1

```

```

        continue
    print("Oke )
    oke += 1

print(f"Ringkasan → Spam:{spam}, Oke:{oke}")

```

### Opsional naik level (AI open source):

Coba latih *neural network* mungil dengan *Keras* untuk menilai *spam/oke* berbasis fitur sederhana (panjang teks dan jumlah tanda seru). Ini hanya untuk rasa *machine learning*—struktur kontrol tetap dipakai untuk evaluasi hasil.

```

# pip install tensorflow numpy
import numpy as np
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# Fitur sederhana: [panjang, jumlah_!]
X = np.array([[20,0],[250,5],[80,0],[230,3],[60,1],[15,0]],
              float)
y = np.array([0,1,0,1,0,0]) # 1=spam, 0=oke

model = Sequential([Dense(8, activation="relu",
                           input_shape=(2,)),
                    Dense(1, activation="sigmoid")])
model.compile(optimizer="adam", loss="binary_crossentropy",
              metrics=["accuracy"])
model.fit(X, y, epochs=50, verbose=0)

contoh = np.array([[180, 4]], float)
prob = float(model.predict(contoh, verbose=0)[0][0])
print("Probabilitas spam:", round(prob, 3))
print("Keputusan:", "Spam" if prob >= 0.5 else "Oke")

```

### Tips singkat

- **Baca soal pelan.** Pecah jadi langkah kecil.
- Pakai *print* untuk *debug*.
- Jaga **indentasi**.
- Uji *edge case* (input nol, kosong, huruf kapital).

Selamat berkarya! **Kuasai *if-else* dan *loop***, lalu tambahkan *Pandas* atau *Keras* saat butuh. Kalian siap jadi **sutradara kode**.

## BAB 5: Pengolahan Data Dasar dengan Pandas

Sudah siap untuk petualangan baru di dunia data? Kali ini kita akan belajar tentang **Pandas**, yang merupakan 'tool sakti' untuk mengolah data. Kenapa penting? Karena di dunia kerja nanti, data adalah segalanya, dan Pandas akan membantu kita mengubah data mentah menjadi informasi yang berharga.

## Tujuan Pembelajaran

Setelah selesai bab ini, kalian bakal bisa:

- **Membaca** data dari berbagai format file (seperti CSV atau Excel).
- **Menganalisis** data untuk menemukan informasi penting, seperti nilai rata-rata atau total.
- **Membersihkan** data yang berantakan atau ada yang hilang.
- **Menggabungkan** data dari beberapa sumber menjadi satu.

## Peta Konsep Sederhana

Bayangkan kalian mau membuat salad.

- **Bahan Baku:** Ini adalah **data** kalian, yang bisa berupa file **CSV** (mirip daftar belanja) atau **Excel** (seperti tabel bahan-bahan).
- **Tool Masak:** Ini adalah **Pandas**. Dia memiliki **DataFrame** (mangkuk besar tempat salad) dan **Series** (satu buah saja, misalnya satu tomat).
- **Proses Masak:** Ini adalah perintah-perintah Pandas, seperti `df.head()` (mengintip isi mangkuk), `df.describe()` (menganalisis kandungan nutrisi), atau `df.dropna()` (membuang sayuran yang layu).
- **Hasil Masak:** Ini adalah data yang sudah bersih dan siap disajikan.

## Apersepsi: Bayangin Kalian Buka Kafe!

Setiap hari kalian catat:

- Minuman apa yang dibeli?
- Berapa harganya?
- Jam berapa pembeli datang?

Di akhir bulan, tumpukan catatan kalian berantakan. Mana ada beberapa yang ketumpahan kopi! Nah, kalau kalian pakai **Pandas**, semua catatan itu bisa diubah jadi tabel digital yang rapi. Cukup ketik beberapa baris kode, kalian bisa langsung tahu:

- Total pendapatan bulan ini.
- Minuman apa yang paling laku?
- Waktu-waktu apa saja yang paling ramai?

Tuh, keren kan?

## Penjelasan Teori

### Kenapa Penting Belajar Pandas?

- Dunia kerja sekarang butuh orang yang bisa **olah data**.
- Data itu ibarat *emas digital*.
- Dengan *Pandas*, data yang tadinya berantakan bisa disulap jadi **informasi berharga**.
- Semua ini bisa kita lakukan dengan **kode singkat**.

### Ilustrasi Sederhana

Coba bayangkan kalian lagi buka **kafe kecil**.

Setiap hari kalian catat:

- Nama minuman
- Harga
- Jam pembeli datang

Setelah sebulan, catatan kalian jadi berantakan. Ada yang ketumpahan kopi, ada angka yang hilang. Nah, dengan *Pandas*, semua catatan itu bisa diubah jadi **tabel digital yang rapi**. Hanya dengan beberapa baris kode, kalian bisa tahu:

- Total pendapatan
- Menu paling laris
- Jam paling ramai

Seru kan?

### Teori Super Gampang

*Pandas* punya 2 struktur data utama:

1. **Series**
  - Isinya seperti **satu kolom data**.
  - Contoh: daftar nama teman → `['Andi', 'Budi', 'Caca']`.
2. **DataFrame**
  - Isinya seperti **tabel lengkap** (baris + kolom).
  - Contoh: tabel nilai rapor kalian.

| Nama | Matematika | Bahasa Indonesia |
|------|------------|------------------|
|------|------------|------------------|

|      |    |    |
|------|----|----|
| Andi | 90 | 85 |
| Budi | 80 | 92 |
| Caca | 95 | 88 |

## Analogi Salad

- **Bahan baku** = data mentah (file CSV atau Excel).
- **Mangkuk besar** = *DataFrame*.
- **Tomat, timun, selada** = *Series*.
- **Resep masak** = perintah Python (`df.head()`, `df.dropna()`, dll.).
- **Salad siap makan** = data yang sudah bersih.

## Aktivitas dengan Pandas

1. **Jelajah Data** → cari file CSV dari internet tentang film/game favoritmu.
2. **Bersih-Bersih Data** → hilangkan nilai kosong, ganti dengan rata-rata.
3. **Siapa Juara?** → urutkan nilai siswa dari tertinggi ke terendah.
4. **Kepo Data** → hitung berapa genre musik unik di playlist.
5. **Simpan Hasil** → export data jadi file Excel baru.

## Rangkuman

- *Pandas* = alat sakti Python untuk olah data.
- Punya 2 senjata utama: **Series** dan **DataFrame**.
- Bisa membaca, membersihkan, menganalisis, sampai menyimpan ulang data.
- Dipakai di dunia nyata: bisnis, riset, sampai AI.

## Contoh Kasus atau Ilustrasi

### Kasus 1: Analisis Game Online

Bayangin kalian punya data pemain game online: nama, skor, dan level. Kalian ingin tahu **5 pemain teratas**.

#### Ilustrasi:

- Data = tabel nama pemain
- Pandas = alat sortir otomatis
- Output = daftar "Top 5 Player"

#### Pseudocode:

```
load data pemain
sortir data berdasarkan skor dari tertinggi
tampilkan 5 baris pertama
```

## Kasus 2: Playlist Musik

Kalian punya playlist besar: judul lagu, artis, dan genre. Kalian ingin tahu **genre musik apa yang paling sering diputar**.

### Ilustrasi:

- Data = daftar lagu (mirip playlist di Spotify/YouTube)
- Pandas = hitung jumlah genre
- Output = genre paling populer

### Pseudocode:

```
load data musik
hitung jumlah lagu berdasarkan genre
urutkan hasil dari yang terbanyak
```

## Kasus 3: Jadwal Piket Kelas

Ada jadwal piket: nama, hari, tugas. Tugasnya sering lupa dikerjakan. Dengan Pandas, kalian bisa tahu **siapa yang belum pernah piket**.

### Ilustrasi:

- Data = tabel jadwal
- Pandas = filter siswa yang jumlah piketnya = 0
- Output = daftar siswa yang bolos piket

### Pseudocode:

```
load data piket
hitung jumlah piket per siswa
tampilkan siswa dengan jumlah piket = 0
```

## Kasus 4: Data Penjualan Kantin

Kalian punya data jualan gorengan, es teh, dan roti. Kalian ingin tahu **makanan apa yang paling laku**.

### Ilustrasi:

- Data = catatan penjualan harian
- Pandas = hitung jumlah terjual tiap makanan
- Output = makanan terlaris

### Pseudocode:

```
load data kantin
kelompokkan data berdasarkan nama makanan
jumlahkan total penjualan tiap makanan
urutkan dari yang terbanyak
```

## Kasus 5: Data Kehadiran Siswa

Kalian punya catatan kehadiran harian: nama, tanggal, status (hadir/izin/alpa). Dengan Pandas, kalian bisa tahu **siapa yang paling sering tidak hadir**.

### Ilustrasi:

- Data = tabel absensi
- Pandas = filter data "alpa"
- Output = daftar siswa dengan absen terbanyak

### Pseudocode:

```
load data kehadiran
filter hanya baris dengan status = "alpa"
hitung jumlah alpa per siswa
tampilkan siswa dengan jumlah alpa terbanyak
```

Dengan lima kasus ini, kalian bisa lihat sendiri betapa kerennya *Pandas*. Dari game, musik, piket, kantin, sampai absensi, semuanya bisa dianalisis dengan cepat dan rapi.

## Contoh Soal dan Pembahasan

### Soal 1: Melihat Data Awal

Kalian punya DataFrame nilai siswa. Bagaimana cara menampilkan **5 baris pertama**?

### Pseudocode:

```
load data nilai
tampilkan 5 baris pertama
```

### Jawaban:

```
df_nilai.head()
```

### Pembahasan:

`head()` dipakai untuk mengintip isi data. Biasanya default menampilkan 5 baris pertama. Ini seperti membuka halaman depan buku catatan nilai.

## Soal 2: Hitung Rata-Rata

Bagaimana cara menghitung **rata-rata nilai Matematika**?

**Pseudocode:**

```
load data nilai
ambil kolom matematika
hitung rata-rata
```

**Jawaban:**

```
df_nilai['Matematika'].mean()
```

**Pembahasan:**

`mean()` adalah fungsi untuk menghitung rata-rata. Kalau datanya 90, 80, 100 → hasilnya 90.

## Soal 3: Filter Data

Tampilkan semua siswa yang punya **nilai Fisika di atas 85**.

**Pseudocode:**

```
load data nilai
cari baris dengan fisika > 85
tampilkan hasil
```

**Jawaban:**

```
df_nilai[df_nilai['Fisika'] > 85]
```

**Pembahasan:**

Kode ini memilih hanya baris dengan nilai Fisika lebih besar dari 85. Mirip seperti kita menyaring siswa yang nilainya bagus.

## Soal 4: Hapus Kolom

Bagaimana cara menghapus kolom **Fisika** dari DataFrame?

**Pseudocode:**

```
load data nilai
hapus kolom fisika
```

**Jawaban:**

```
df_nilai.drop(columns=['Fisika'])
```

**Pembahasan:**

`drop(columns=...)` artinya kita buang kolom tertentu. Setelah ini, DataFrame hanya punya kolom lain seperti Nama dan Matematika.

### Soal 5: Isi Data Hilang

Ada beberapa data kosong di kolom Fisika. Bagaimana cara menggantinya dengan nilai 0?

**Pseudocode:**

```
load data nilai
cek data kosong
isi dengan 0
```

**Jawaban:**

```
df_nilai.fillna(0)
```

**Pembahasan:**

`fillna(0)` dipakai untuk mengganti data kosong (NaN) dengan angka 0. Jadi tabel lebih rapi dan tidak ada nilai kosong.

Dengan lima soal ini, kalian sudah bisa:

- Melihat data,
- Menghitung rata-rata,
- Memfilter,
- Menghapus kolom,
- Mengisi data kosong.

Itu semua adalah **langkah dasar analisis data** dengan *Pandas* di Python.

### Fun Facts!

- Nama **Pandas** diambil dari "PANel DAtaS", sebuah istilah di bidang ilmu statistik.
- Pandas diciptakan oleh seorang yang bekerja di bidang keuangan, jadi jangan heran kalau dia jago banget menghitung angka.
- **Ollama** dan **OpenWebUI** itu bisa dibilang "otak" dan "tampilan" buat bikin **AI** di komputer kalian. Nah, **RAG lokal** itu bikin AI-nya makin pintar dengan data yang ada di komputer kita sendiri. Keren kan?!

### Tips Belajar & Menghafal Cepat

- **Lakukan Langsung!** Tidak ada gunanya hanya membaca. Langsung buka Jupyter Notebook dan coba ketik kode-kode di atas.
- **Gunakan Contoh Pribadi!** Buat data tentang hobi kalian, misalnya game, sepak bola, atau film. Belajarnya jadi lebih seru!
- **Tanya ke AI!** Kalian bisa bertanya ke **Ollama** atau **OpenWebUI** (yang sudah kalian instal di komputer) kalau ada kode yang membingungkan.

## Aktivitas Siswa

### Aktivitas 1: Jelajah Data

Bayangin kamu punya data peringkat game. Kita mau lihat 10 data teratas.

#### Langkah:

1. Cari file CSV (misalnya `game.csv`).
2. Impor file ke Pandas.
3. Tampilkan 10 baris pertama.

#### Pseudocode:

```
load data dari CSV
tampilkan 10 baris pertama
```

#### Python Script:

```
import pandas as pd

df_game = pd.read_csv("game.csv")
print(df_game.head(10))
```

### Aktivitas 2: Bersih-Bersih Data

Kadang ada data kosong. Misalnya nilai matematika siswa ada yang hilang. Kita isi dengan nilai rata-rata.

#### Langkah:

1. Hitung jumlah data kosong.
2. Cari rata-rata nilai.
3. Ganti data kosong dengan rata-rata.

#### Pseudocode:

```
cek data kosong
hitung rata-rata nilai
isi data kosong dengan rata-rata
```

### Python Script:

```
import pandas as pd

df_nilai = pd.read_csv("nilai.csv")
print(df_nilai.isnull().sum())
mean_math = df_nilai['Matematika'].mean()
df_nilai['Matematika'] =
df_nilai['Matematika'].fillna(mean_math)
print(df_nilai)
```

## Aktivitas 3: Siapa Juara?

Dari data nilai, kita cari siapa siswa dengan total nilai tertinggi.

### Langkah:

1. Tambahkan kolom **Total**.
2. Jumlahkan semua nilai per siswa.
3. Urutkan dari yang tertinggi.

### Pseudocode:

```
buat kolom total
jumlahkan semua nilai per siswa
urutkan data berdasarkan total
```

### Python Script:

```
import pandas as pd

df_nilai = pd.read_csv("nilai.csv")
df_nilai['Total'] = df_nilai['Matematika'] + df_nilai['Fisika'] +
df_nilai['Kimia']
df_sorted = df_nilai.sort_values(by='Total', ascending=False)
print(df_sorted)
```

## Aktivitas 4: Kepo Data

Kamu punya playlist musik. Cari tahu genre apa yang paling banyak.

### Langkah:

1. Pilih kolom **Genre**.
2. Hitung jumlah setiap genre.
3. Tampilkan hasil.

### Pseudocode:

```
ambil kolom genre
```

```
hitung jumlah tiap genre  
tampilkan hasil
```

### Python Script:

```
import pandas as pd  
  
df_music = pd.read_csv("playlist.csv")  
print(df_music['Genre'].value_counts())
```

## Aktivitas 5: Simpan Data

Setelah data kamu bersih dan rapi, simpan lagi ke file Excel.

### Langkah:

1. Pastikan data sudah rapi.
2. Gunakan fungsi `to_excel`.
3. Cek file hasilnya.

### Pseudocode:

```
pastikan data sudah rapi  
simpan data ke file excel
```

### Python Script:

```
import pandas as pd  
  
df_nilai = pd.read_csv("nilai.csv")  
df_nilai.to_excel("nilai_baru.xlsx", index=False)
```

Dengan 5 aktivitas ini, kalian sudah bisa:

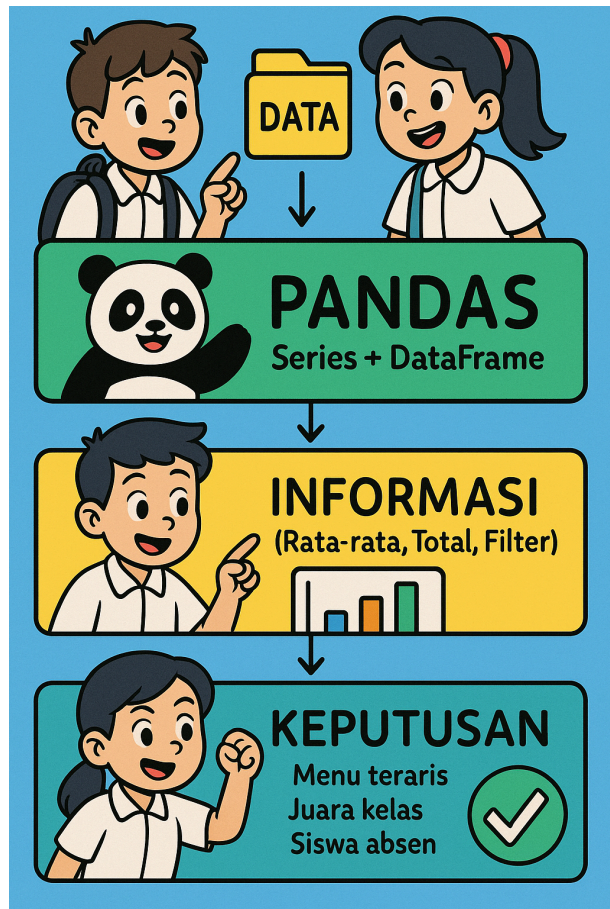
- Membaca data,
- Membersihkan data,
- Mengurutkan,
- Menganalisis,
- Menyimpan ulang hasil.

Semua pakai **Python + Pandas** yang open source, tanpa bayar lisensi.

## Rangkuman

Di bab ini kita belajar tentang **Pandas**, sebuah *library open source* di **Python** yang bisa dipakai untuk mengolah data. Dengan *Pandas*, kita bisa membaca file **CSV** atau **Excel**, lalu mengubah data yang berantakan menjadi rapi dan mudah dipahami. Dua senjata utamanya adalah **Series** (mirip satu kolom data) dan **DataFrame** (mirip tabel lengkap). Jadi, Pandas itu ibarat **mangkuk besar salad** yang bisa menampung banyak bahan lalu kita olah sesuai keinginan.

Kita juga melihat banyak contoh seru: dari **analisis game online**, **playlist musik**, **jadwal piket kelas**, **penjualan kantin**, sampai **absensi siswa**. Semua itu bisa dihitung dengan cepat hanya dengan beberapa baris kode *Python*. Misalnya mencari **nilai rata-rata**, **filter siswa terbaik**, atau **makanan paling laku**. Dengan Pandas, pekerjaan yang biasanya lama bisa selesai dengan singkat. Ini membuat kita paham bahwa **data adalah harta berharga** di dunia digital.



Belajar *Pandas* bukan hanya untuk anak IT. **Setiap profesi** yang berhubungan dengan data akan sangat terbantu. Mulai dari pedagang online, analis keuangan, peneliti, sampai pelatih e-sport bisa memanfaatkannya. Pandas juga sering dipakai sebelum masuk ke tahap *machine learning* dengan *Keras* atau *neural network*. Jadi, menguasai Pandas adalah langkah awal untuk jadi **data scientist masa depan**. Ingat, semakin cepat kita belajar *open source tools*, semakin besar peluang kita di dunia kerja.

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. Pandas hanya bisa membaca file dengan format CSV.  
**Jawaban: Salah**
2. DataFrame mirip dengan tabel yang punya baris dan kolom.  
**Jawaban: Benar**

3. Fungsi `df.head()` digunakan untuk menampilkan 5 baris pertama dari DataFrame.

**Jawaban: Benar**

4. Perintah `df.dropna()` digunakan untuk menambahkan baris baru ke DataFrame.

**Jawaban: Salah**

5. Series di Pandas mirip dengan satu kolom data saja.

**Jawaban: Benar**

## B. Pilihan Ganda (10 Soal)

1. Apa fungsi utama library Pandas?

- a. Membuat grafik
- b. Mengolah data
- c. Mengirim email
- d. Membuat game

**Jawaban: b**

2. Struktur data utama Pandas ada berapa?

- a. 1
- b. 2
- c. 3
- d. 4

**Jawaban: b**

3. Perintah apa yang digunakan untuk membaca file CSV di Pandas?

- a. `pd.open()`
- b. `pd.read_csv()`
- c. `pd.loadfile()`
- d. `pd.csv_open()`

**Jawaban: b**

4. Perintah `df.describe()` digunakan untuk...

- a. Menampilkan statistik deskriptif data
- b. Menghapus data kosong
- c. Mengurutkan data
- d. Menyimpan data ke Excel

**Jawaban: a**

5. Jika ingin menyimpan DataFrame ke file Excel, perintah yang benar adalah:

- a. `df.save_excel()`
- b. `df.to_excel()`
- c. `df.write_excel()`
- d. `df.excel_save()`

**Jawaban: b**

6. Apa hasil dari kode berikut?

```
import pandas as pd
df = pd.DataFrame({'Nilai': [80, 90, 100]})
print(df['Nilai'].mean())
```

- a. 80
- b. 90
- c. 100
- d. Error

**Jawaban: b**

7. Fungsi `df.drop(columns=['Fisika'])` artinya...

- a. Menghapus semua baris DataFrame
- b. Menghapus kolom Fisika
- c. Menambahkan kolom baru Fisika
- d. Menyalin kolom Fisika

**Jawaban: b**

8. Untuk menampilkan 10 baris pertama DataFrame digunakan perintah...

- a. `df.show(10)`
- b. `df.head(10)`
- c. `df.display(10)`
- d. `df.top(10)`

**Jawaban: b**

9. Jika ada nilai kosong di DataFrame, fungsi yang bisa dipakai adalah...

- a. `df.fillna()`
- b. `df.replace()`
- c. `df.empty()`
- d. `df.add()`

**Jawaban: a**

10. Apa nama library Python untuk analisis data yang dibahas di bab ini?

- a. Matplotlib
- b. NumPy
- c. Pandas
- d. TensorFlow

**Jawaban: c**

### C. Isian Singkat (5 Soal)

1. Fungsi `df.head()` secara default menampilkan ... baris pertama.

**Jawaban: 5**

2. Series di Pandas hanya berisi satu ... data.

**Jawaban: kolom**

3. Jika ingin mengetahui rata-rata nilai matematika, digunakan fungsi ...

**Jawaban: mean()**

4. Perintah `df.to_excel("data.xlsx", index=False)` akan menyimpan DataFrame ke format ...

**Jawaban: Excel**

5. DataFrame bisa dianalogikan sebagai ... besar yang menampung banyak bahan salad.

**Jawaban: mangkuk**

#### D. Soal Eksplorasi (3 Soal)

##### Soal 1:

###### Analisis Penjualan Kantin

Kamu punya data penjualan kantin berisi kolom: *makanan*, *jumlah*, *harga*. Buat program sederhana dengan Pandas untuk menghitung total penjualan tiap makanan, lalu tampilkan makanan paling laku.

**Jawaban Eksplorasi:**

```
import pandas as pd
df = pd.read_csv("kantin.csv")
df['total'] = df['jumlah'] * df['harga']
print(df.groupby('makanan')['total'].sum().sort_values(ascending=False))
```

##### Soal 2:

###### Analisis Playlist Musik

Kamu punya playlist berisi: *judul*, *artis*, *genre*. Tugasmu adalah mencari genre musik apa yang paling sering diputar.

**Jawaban Eksplorasi:**

```
import pandas as pd
df = pd.read_csv("playlist.csv")
print(df['genre'].value_counts().head(1))
```

##### Soal 3:

###### Analisis Kehadiran Kelas

Kamu punya data kehadiran berisi: *nama*, *tanggal*, *status* (hadir/izin/alpa). Buat kode untuk mencari siswa dengan jumlah *alpa* terbanyak.

**Jawaban Eksplorasi:**

```
import pandas as pd
df = pd.read_csv("absensi.csv")
df_alpa = df[df['status'] == "alpa"]
print(df_alpa['nama'].value_counts().head(1))
```

## Tugas Proyek (5 Soal)

### Proyek 1: Analisis Data Cuaca

Kalian diberi data cuaca: *tanggal*, *suhu*, *kelembapan*. Buat analisis rata-rata suhu per minggu.

**Langkah:**

1. Buka file *cuaca.csv*.
2. Kelompokkan data berdasarkan minggu.
3. Hitung rata-rata suhu tiap minggu.

**Pseudocode:**

```
load data cuaca
kelompokkan per minggu
hitung rata-rata suhu
```

**Python:**

```
import pandas as pd
df = pd.read_csv("cuaca.csv")
df['tanggal'] = pd.to_datetime(df['tanggal'])
df['minggu'] = df['tanggal'].dt.isocalendar().week
print(df.groupby('minggu')['suhu'].mean())
```

### Proyek 2: Analisis Kehadiran Ekstrakurikuler

Kalian punya data: *nama*, *ekskul*, *hadir*. Cari siswa dengan tingkat kehadiran tertinggi di ekstrakurikuler Basket.

**Langkah:**

1. Filter data dengan ekstrakurikuler = "Basket".
2. Hitung jumlah hadir tiap siswa.
3. Urutkan dari yang terbanyak.

**Pseudocode:**

```
filter ekstrakurikuler basket
hitung hadir per siswa
urutkan dari terbanyak
```

**Python:**

```
df = pd.read_csv("ekskul.csv")
df_basket = df[df['ekskul'] == "Basket"]
print(df_basket.groupby('nama')['hadir'].sum().sort_values(ascending=False))
```

### Proyek 3: Analisis Penjualan Online

Ada data penjualan: **produk**, **jumlah**, **harga**. Hitung total omzet per produk.

**Langkah:**

1. Buka file **penjualan.csv**.
2. Buat kolom **total** = jumlah × harga.
3. Kelompokkan berdasarkan produk.

**Pseudocode:**

```
load data penjualan
buat kolom total = jumlah*harga
kelompokkan per produk
jumlahkan total
```

**Python:**

```
df = pd.read_csv("penjualan.csv")
df['total'] = df['jumlah'] * df['harga']
print(df.groupby('produk')['total'].sum())
```

### Proyek 4: Analisis Waktu Belajar

Data: **nama**, **mapel**, **jam\_belajar**. Cari siswa dengan rata-rata jam belajar tertinggi.

**Langkah:**

1. Impor data belajar.
2. Hitung rata-rata jam belajar per siswa.
3. Cari yang tertinggi.

**Pseudocode:**

```
load data belajar
```

hitung rata-rata jam belajar per siswa  
tampilkan siswa dengan nilai tertinggi

**Python:**

```
df = pd.read_csv("belajar.csv")
print(df.groupby('nama')['jam_belajar'].mean().sort_values(ascending=False).head(1))
```

## Proyek 5: Analisis Data Film

Kalian punya data: **judul**, **genre**, **rating**. Cari genre dengan rata-rata rating tertinggi.

**Langkah:**

1. Impor data film.
2. Kelompokkan berdasarkan genre.
3. Hitung rata-rata rating tiap genre.
4. Urutkan dari yang tertinggi.

**Pseudocode:**

```
load data film
kelompokkan berdasarkan genre
hitung rata-rata rating
urutkan dari tertinggi
```

**Python:**

```
df = pd.read_csv("film.csv")
print(df.groupby('genre')['rating'].mean().sort_values(ascending=False))
```

Dengan 5 proyek ini, kalian bisa mencoba **analisis nyata** di berbagai bidang: cuaca, ekskul, jualan online, belajar, sampai film favorit. Semua pakai *Python + Pandas* yang open source.

# BAB 6: Visualisasi Data dengan matplotlib

Siapa buat jadi ahli visualisasi data? Kalau di bab sebelumnya kita sudah jago mengolah data pakai **Pandas**, sekarang waktunya kita "melukis" data itu jadi grafik dan diagram yang keren.

## Tujuan Pembelajaran

Setelah belajar bab ini, kalian bakal bisa:

- **Membuat** berbagai macam grafik dan diagram, dari yang paling sederhana sampai yang paling kompleks.
- **Menganalisis** data dengan lebih cepat dan mudah lewat grafik.
- **Menceritakan** sebuah "kisah" di balik data, cuma dengan melihat gambarnya.
- **Menghubungkan** visualisasi data dengan dunia nyata (hobi, game, dan proyek keren kalian!).

## Peta Konsep Sederhana

Bayangin kita mau buat poster keren buat acara sekolah.

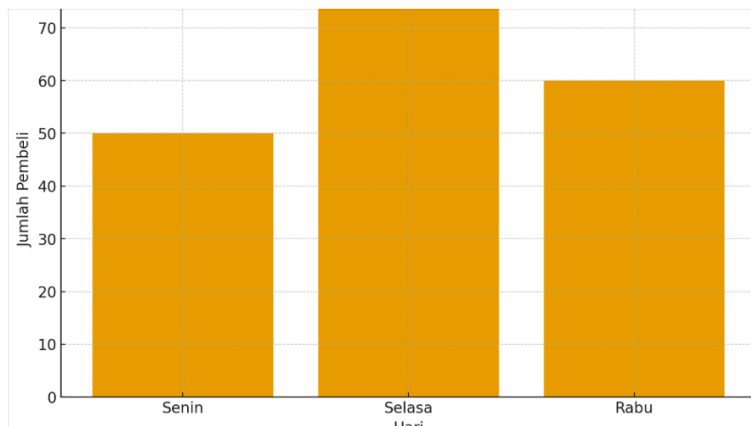
- **Bahan Baku:** Ini data mentah kita (udah diolah pakai Pandas). Ibaratnya, ini adalah daftar nama, nilai, dan kegiatan siswa.
- **Tool Gambar:** Ini si **matplotlib**. Dia punya banyak "kuas" dan "pensil" buat bikin grafik, seperti `plt.plot()` buat garis, `plt.bar()` buat batang, dan `plt.pie()` buat lingkaran.
- **Proses Menggambar:** Ini adalah langkah-langkah kita, seperti `plt.title()` buat judul, `plt.xlabel()` buat label sumbu X, dan `plt.show()` buat nampilin gambarnya.
- **Hasil Gambar:** Ini adalah grafik yang indah dan mudah dimengerti. Semua orang bisa langsung tahu pesan dari data kita.

## Apersepsi: Tebak Gambar!

Pernah main game "Tebak Gambar"? Kalau ada gambar orang sedang makan, kamu langsung tahu kan? Nah, data juga gitu! Daripada baca ribuan angka, mending kita lihat gambarnya.

**Challenge kecil:** Mana yang lebih gampang dimengerti?

- **A:** `['Hari', 'Senin', 'Selasa', 'Rabu'] ['Jumlah Pembeli', 50, 75, 60]`
- **B:**



Pasti lebih gampang lihat **B** kan? Kamu langsung tahu kalau hari Selasa paling ramai pembeli. Itulah kekuatan visualisasi data!

## Penjelasan Teori

### Apa itu matplotlib?

- *matplotlib* adalah **library open source di Python**.
- Dia adalah “kuas digital” untuk melukis data.
- Kita bisa bikin:
  - **Diagram batang** → cocok untuk membandingkan kategori.
  - **Diagram garis** → cocok untuk melihat tren dari waktu ke waktu.
  - **Diagram lingkaran** → cocok untuk tahu proporsi dari keseluruhan.

Contoh:

- Bandingkan jumlah siswa di jurusan TKJ, RPL, MM → pakai **diagram batang**.
- Lihat pertumbuhan subscribers YouTube dari bulan ke bulan → pakai **diagram garis**.
- Hitung persentase uang jajan: makan, pulsa, nabung → pakai **diagram lingkaran**.

### Kenapa Visualisasi Itu Penting?

- **Lebih gampang dimengerti**: sekali lihat grafik, kita langsung tahu cerita datanya.
- **Lebih cepat dianalisis**: tren naik-turun kelihatan jelas.
- **Lebih keren dipresentasikan**: laporanmu jadi hidup, bukan sekedar angka.

Ingat: setiap data punya cerita. Visualisasi adalah cara kita **menceritakan data lewat gambar**.

### Contoh Perintah Dasar *matplotlib*

- `plt.plot()` → bikin garis
- `plt.bar()` → bikin batang
- `plt.pie()` → bikin lingkaran

- `plt.title()` → kasih judul
- `plt.xlabel()` & `plt.ylabel()` → kasih label sumbu
- `plt.show()` → memunculkan grafik

Sederhana kan? Tinggal olah data di *Pandas*, terus kita “gambar” pakai *matplotlib*.

## Analogi Seru

Bayangin kamu main game **Tebak Gambar**.

Kalau ada gambar orang lagi makan, kamu langsung paham artinya.

Nah, visualisasi data juga gitu → lebih cepat dipahami daripada lihat tabel angka panjang.

Jadi, **visualisasi data = seni + sains**. Kita pakai *matplotlib* untuk bikin grafik yang indah, informatif, dan mudah dimengerti. Ini bukan cuma untuk belajar, tapi juga bisa dipakai di dunia kerja, riset, bahkan hobi kamu (analisis game, channel YouTube, atau keuangan pribadi).

## Contoh Kasus atau Ilustrasi

### Kasus 1: Rank Game MLBB

Kamu ingin tahu jumlah pemain di tiap rank: Warrior, Elite, Master.

Kalau cuma angka, ribet. Tapi kalau pakai grafik batang, langsung kelihatan rank mana yang paling ramai!

#### Pseudocode:

```
load data pemain
pilih kolom rank dan jumlah
buat diagram batang
tampilkan grafik
```

### Kasus 2: Pertumbuhan Subscribers YouTube

Kamu punya data subscribers channel favorit dari Januari sampai Juni.

Kalau ditampilkan dalam grafik garis, tren naik-turun akan kelihatan jelas.

#### Pseudocode:

```
load data subscribers
buat diagram garis dengan bulan di sumbu X
jumlah subscribers di sumbu Y
tampilkan grafik
```

### Kasus 3: Pengeluaran Uang Jajan

Data: makan 40%, transport 30%, game 20%, nabung 10%.  
Kalau ditampilkan dalam diagram lingkaran, langsung terlihat mana yang paling besar porsinya.

**Pseudocode:**

```
buat list kategori pengeluaran
buat list jumlah pengeluaran
buat diagram lingkaran
tampilkan grafik
```

## Kasus 4: Tingkat Kehadiran Sekolah

Data: jumlah siswa hadir tiap hari dalam seminggu.  
Dengan grafik batang, bisa ketahuan hari apa yang paling banyak bolos (misalnya Jumat).

**Pseudocode:**

```
load data kehadiran
buat diagram batang (hari vs jumlah hadir)
beri judul dan label
tampilkan grafik
```

## Kasus 5: Penjualan Makanan di Kantin

Data: gorengan 120, bakso 80, es teh 150.  
Kalau divisualisasikan, langsung kelihatan kalau es teh paling laris.

**Pseudocode:**

```
load data penjualan
buat diagram batang (makanan vs jumlah terjual)
beri warna berbeda untuk tiap makanan
tampilkan grafik
```

Dengan 5 contoh ini, kalian bisa lihat betapa mudahnya mengubah angka jadi **cerita visual**.  
Data nggak lagi bikin pusing, tapi justru jadi seru untuk dipelajari.

## Contoh Soal dan Pembahasan

### Soal 1: Diagram Batang Penjualan Makanan

Data kantin sekolah: gorengan = 120, bakso = 80, es teh = 150.  
Buatlah **diagram batang** untuk menampilkan data tersebut!

**Pseudocode:**

```
import matplotlib.pyplot as plt
```

```
makanan = ['Gorengan', 'Bakso', 'Es Teh']  
jumlah = [120, 80, 150]  
buat diagram batang dari (makanan, jumlah)  
tampilkan grafik
```

**Jawaban & Pembahasan:**

Gunakan `plt.bar()` untuk membuat grafik batang. Grafik akan menunjukkan kalau es teh paling laris.

**Soal 2: Diagram Garis Pertumbuhan Subscribers**

Data channel YouTube: Januari=100, Februari=200, Maret=400, April=600.  
Buat **diagram garis** untuk melihat tren pertumbuhan!

**Pseudocode:**

```
bulan = ['Jan', 'Feb', 'Mar', 'Apr']
subs = [100, 200, 400, 600]
buat diagram garis dari (bulan, subs)
tampilkan grafik
```

**Jawaban & Pembahasan:**

Gunakan `plt.plot()`. Hasil grafik menunjukkan tren naik terus, artinya channel makin populer.

**Soal 3: Diagram Lingkaran Pengeluaran Uang Jajan**

Data: makan=40%, transport=30%, game=20%, nabung=10%.

Buat **diagram lingkaran** untuk menampilkan persentase!

**Pseudocode:**

```
kategori = ['Makan', 'Transport', 'Game', 'Nabung']
persentase = [40, 30, 20, 10]
buat diagram lingkaran dari (persentase, kategori)
tampilkan grafik
```

**Jawaban & Pembahasan:**

Gunakan `plt.pie()`. Hasil grafik memperlihatkan pengeluaran terbesar untuk makan.

**Soal 4: Diagram Batang Kehadiran Siswa**

Data jumlah siswa hadir dalam seminggu: Senin=30, Selasa=32, Rabu=28, Kamis=29, Jumat=25.

Buatlah grafik batang untuk menampilkan data kehadiran!

**Pseudocode:**

```
hari = ['Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat']
hadir = [30, 32, 28, 29, 25]
buat diagram batang dari (hari, hadir)
tampilkan grafik
```

**Jawaban & Pembahasan:**

Gunakan `plt.bar()`. Dari grafik, terlihat Jumat paling banyak bolos.

**Soal 5: Tambahkan Judul dan Label Grafik**

Ambil data penjualan makanan di soal nomor 1. Tambahkan **judul**, **label sumbu X**, dan **label sumbu Y** ke grafik batang.

**Pseudocode:**

```
buat diagram batang dari (makanan, jumlah)
tambahkan judul "Data Penjualan Kantin"
tambahkan label sumbu X = "Jenis Makanan"
tambahkan label sumbu Y = "Jumlah Terjual"
tampilkan grafik
```

### Jawaban & Pembahasan:

Tambahkan `plt.title()`, `plt.xlabel()`, dan `plt.ylabel()`. Hasil grafik lebih informatif dan enak dilihat.

Dengan soal-soal ini, kalian belajar step by step: bikin grafik, kasih judul, sampai analisis hasilnya. Data jadi lebih gampang dipahami, dan grafiknya bisa dipakai buat laporan atau presentasi.

## Fun Facts!

- **Orange Data Mining** itu software visualisasi data yang gampang banget dipakai. Kamu tinggal drag-and-drop aja, nggak perlu coding!
- **matplotlib** itu kayak bapaknya visualisasi data di Python. Banyak library lain (seperti **seaborn**) yang dibuat di atasnya.
- **Ollama**, **OpenWebUI**, dan **RAG Lokal** juga bisa bantu kamu bikin visualisasi data. Kamu bisa minta AI-nya untuk membuatkan kode visualisasi data dari data yang kamu punya.

## Tips Belajar & Menghafal Cepat

- **Pikirkan Cerita di Balik Data!** Setiap data punya cerita. Bayangin apa yang mau kamu sampaikan sebelum bikin grafiknya.
- **Gunakan Contoh Sehari-hari!** Visualisasikan data tentang hal-hal yang kamu suka, misalnya statistik game, atau data tim sepak bola favoritmu.
- **Buat Mind Map!** Buat peta konsep visual tentang setiap jenis grafik dan kapan harus menggunakannya.

## Aktivitas Siswa

### Aktivitas 1: Grafik Pertumbuhan Subscribers YouTube

Bayangkan kamu punya data subscribers channel favorit dari Januari–April. Yuk, buat grafik garisnya!

**Langkah:**

1. Buat data bulan dan jumlah subscribers.
2. Gunakan `plt.plot()` untuk bikin grafik garis.
3. Tambahkan judul dan label.
4. Tampilkan hasil grafik.

**Pseudocode:**

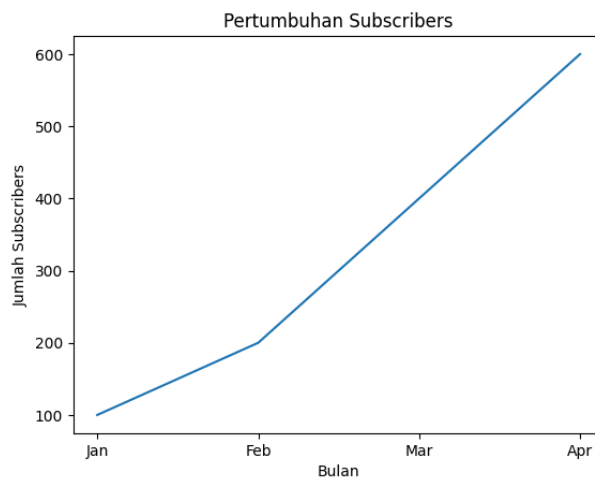
```
buat list bulan
buat list jumlah_subs
gambar garis dari (bulan, jumlah_subs)
beri judul dan label
tampilkan grafik
```

**Python:**

```
import matplotlib.pyplot as plt

bulan = ['Jan', 'Feb', 'Mar', 'Apr']
subs = [100, 200, 400, 600]

plt.plot(bulan, subs)
plt.title("Pertumbuhan Subscribers")
plt.xlabel("Bulan")
plt.ylabel("Jumlah Subscribers")
plt.show()
```



## Aktivitas 2: Diagram Lingkaran Uang Jajan

Catat pengeluaran seminggu: makan, transport, game, nabung. Bikin diagram lingkaran!

**Langkah:**

1. Buat kategori pengeluaran dan jumlahnya.
2. Gunakan `plt.pie()`.
3. Tambahkan label kategori.

#### 4. Tampilkan grafik.

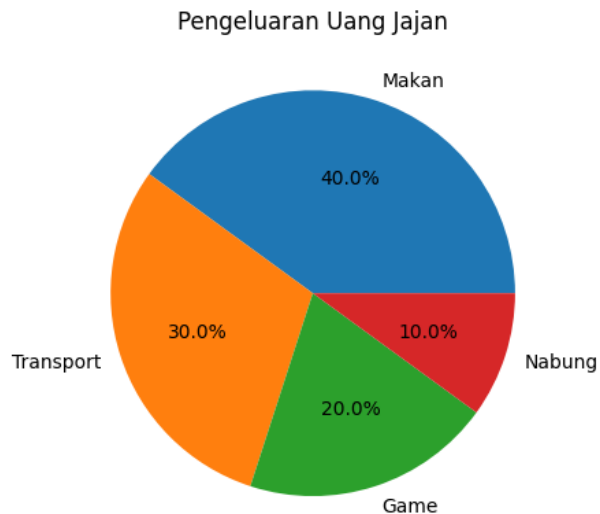
##### Pseudocode:

```
buat list kategori
buat list pengeluaran
gambar pie chart dari data
tampilkan grafik
```

##### Python:

```
kategori = ['Makan', 'Transport', 'Game', 'Nabung']
pengeluaran = [40, 30, 20, 10]

plt.pie(pengeluaran, labels=kategori, autopct='%1.1f%%')
plt.title("Pengeluaran Uang Jajan")
plt.show()
```



### Aktivitas 3: Grafik Batang Kehadiran Siswa

Data siswa hadir per hari: Senin=30, Selasa=32, Rabu=28, Kamis=29, Jumat=25.

##### Langkah:

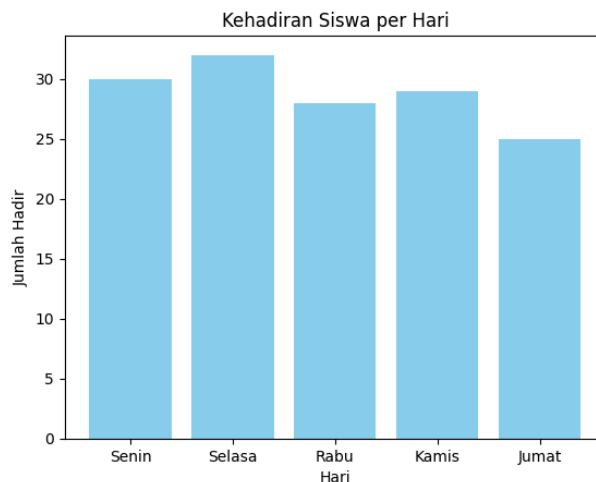
1. Buat list hari dan jumlah hadir.
2. Gunakan `plt.bar()`.
3. Tambahkan judul dan label.
4. Tampilkan grafik.

##### Pseudocode:

```
buat list hari
buat list hadir
gambar bar chart dari (hari, hadir)
beri judul dan label
tampilkan grafik
```

### Python:

```
hari = ['Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat']  
hadir = [30, 32, 28, 29, 25]  
  
plt.bar(hari, hadir, color='skyblue')  
plt.title("Kehadiran Siswa per Hari")  
plt.xlabel("Hari")  
plt.ylabel("Jumlah Hadir")  
plt.show()
```



## Aktivitas 4: Histogram Nilai Siswa

Bayangkan nilai ujian 100 siswa. Yuk, buat histogram untuk melihat sebaran nilai!

### Langkah:

1. Buat data nilai acak.
2. Gunakan `plt.hist()` untuk bikin histogram.
3. Tambahkan judul dan label.
4. Tampilkan hasil grafik.

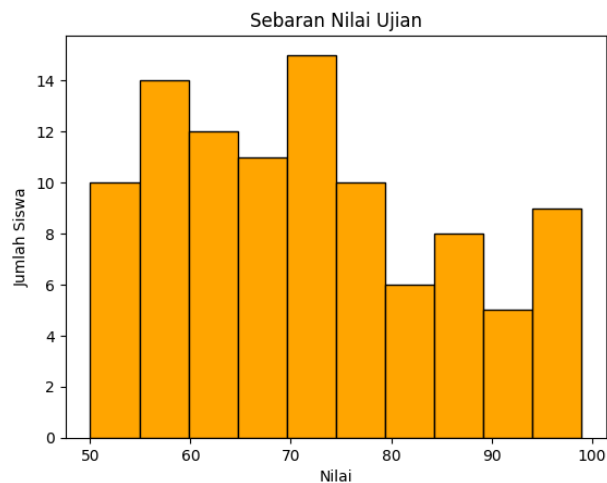
### Pseudocode:

```
buat data nilai acak  
gambar histogram  
beri judul dan label  
tampilkan grafik
```

### Python:

```
import numpy as np  
  
nilai = np.random.randint(50, 100, 100)  
  
plt.hist(nilai, bins=10, color='orange', edgecolor='black')  
plt.title("Sebaran Nilai Ujian")  
plt.xlabel("Nilai")
```

```
plt.ylabel("Jumlah Siswa")
plt.show()
```



## Aktivitas 5: Grafik Cuaca Harian

Data suhu 7 hari: [30, 32, 31, 29, 28, 30, 33]. Visualisasikan dalam grafik garis!

### Langkah:

1. Buat data hari dan suhu.
2. Gunakan `plt.plot()`.
3. Tambahkan judul dan label.
4. Tampilkan hasil grafik.

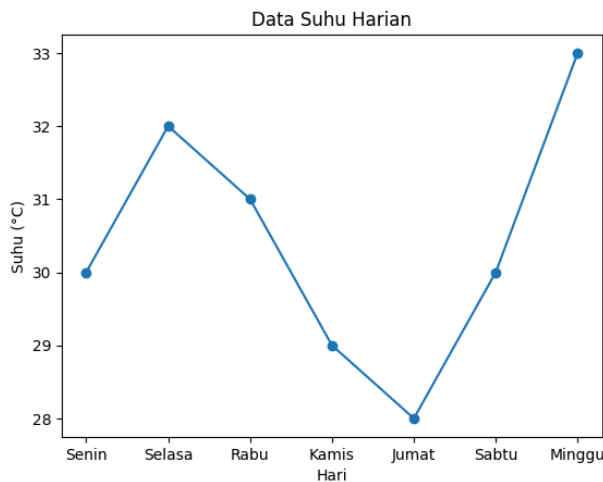
### Pseudocode:

```
buat list hari
buat list suhu
gambar line chart dari (hari, suhu)
beri judul dan label
tampilkan grafik
```

### Python:

```
hari = ['Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu']
suhu = [30, 32, 31, 29, 28, 30, 33]

plt.plot(hari, suhu, marker='o')
plt.title("Data Suhu Harian")
plt.xlabel("Hari")
plt.ylabel("Suhu (°C)")
plt.show()
```



Dengan 5 aktivitas ini, kalian belajar **step by step** membuat berbagai macam grafik: garis, batang, lingkaran, histogram, bahkan cuaca. Semua pakai *open source tools* seperti Python, Pandas, dan *matplotlib*.

## Rangkuman

Visualisasi data adalah **cara seru untuk mengubah angka jadi gambar**. Dengan *matplotlib*, kita bisa bikin grafik batang, garis, lingkaran, sampai histogram. Grafik ini bikin kita cepat paham data tanpa harus baca ribuan angka. Jadi, **data yang tadinya membosankan bisa jadi menarik**.

Alat ini 100% **open source** dan bisa dipakai semua orang. Di Python, cukup beberapa baris kode sederhana, kita sudah bisa bikin grafik yang keren. Misalnya pakai `plt.bar()` untuk batang, `plt.plot()` untuk garis, dan `plt.pie()` untuk lingkaran. **Setiap grafik punya cerita**: tren, perbandingan, atau proporsi. Tinggal pilih sesuai kebutuhan.

Skill visualisasi data ini **sangat penting di dunia kerja**. Perusahaan, guru, bahkan gamer e-sports butuh cara cepat memahami data. Dengan *matplotlib*, kita bisa bikin laporan jadi jelas dan menarik. Selain itu, ini juga bisa jadi hobi, misalnya bikin grafik pertumbuhan channel YouTube atau analisis uang jajan. **Jadi, jangan lupa terus berlatih!** Semakin sering kalian mencoba, semakin mudah kalian jago.

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. *matplotlib* adalah library Python yang digunakan untuk mengolah database.

**Jawaban:** Salah

2. Fungsi `plt.bar()` digunakan untuk membuat diagram batang.

**Jawaban:** Benar

3. Grafik garis cocok dipakai untuk melihat tren waktu ke waktu.

**Jawaban:** Benar

4. `plt.pie()` digunakan untuk membuat diagram histogram.

**Jawaban:** Salah

5. Dengan `plt.show()`, grafik bisa ditampilkan ke layar.

**Jawaban:** Benar

## B. Pilihan Ganda (10 Soal)

1. Fungsi untuk membuat diagram lingkaran di *matplotlib* adalah ...

- a) `plt.plot()`
- b) `plt.hist()`
- c) `plt.pie()`
- d) `plt.bar()`

**Jawaban:** c

2. Library Python yang populer untuk visualisasi data adalah ...

- a) numpy
- b) pandas
- c) matplotlib
- d) keras

**Jawaban:** c

3. Apa fungsi `plt.title()`?

- a) Menambahkan judul grafik
- b) Menambah label sumbu X
- c) Menambah label sumbu Y
- d) Menyimpan grafik

**Jawaban:** a

4. Jika ingin menampilkan data nilai ujian siswa dalam bentuk batang, kita gunakan ...

- a) `plt.plot()`
- b) `plt.bar()`
- c) `plt.pie()`
- d) `plt.scatter()`

**Jawaban:** b

5. Untuk membuat label sumbu X digunakan perintah ...

- a) `plt.xlabel()`
- b) `plt.ylabel()`
- c) `plt.title()`
- d) `plt.legend()`

**Jawaban: a**

6. Diagram lingkaran cocok untuk ...
- a) Tren pertumbuhan harga
  - b) Proporsi pengeluaran bulanan
  - c) Perbandingan nilai antar siswa
  - d) Menampilkan judul grafik

**Jawaban: b**

7. Jika kita ingin melihat distribusi data nilai, fungsi yang dipakai adalah ...
- a) `plt.bar()`
  - b) `plt.hist()`
  - c) `plt.pie()`
  - d) `plt.plot()`

**Jawaban: b**

8. Kode `plt.show()` digunakan untuk ...
- a) Menyimpan data
  - b) Menampilkan grafik
  - c) Menghitung rata-rata
  - d) Menghapus grafik

**Jawaban: b**

9. `plt.plot()` digunakan untuk membuat ...
- a) Diagram garis
  - b) Diagram lingkaran
  - c) Diagram batang
  - d) Histogram

**Jawaban: a**

10. Agar grafik mudah dibaca, kita bisa menambahkan ...
- a) Warna, judul, label sumbu
  - b) Database, server, koneksi
  - c) Neural network, node, layer
  - d) AI, machine learning, training

**Jawaban: a**

### C. Isian Singkat (5 Soal)

1. Fungsi untuk menambahkan label pada sumbu Y adalah \_\_\_\_\_.

**Jawaban: `plt.ylabel()`**

2. Jika ingin membuat histogram, gunakan fungsi \_\_\_\_\_.

**Jawaban: `plt.hist()`**

3. Untuk menampilkan grafik ke layar, gunakan perintah .....  
**Jawaban:** plt.show()
4. Data dalam bentuk tabel biasanya bisa diolah dulu dengan library .....  
**Jawaban:** pandas
5. Grafik yang cocok untuk melihat pertumbuhan subscribers YouTube adalah .....  
**Jawaban:** Diagram Garis (plt.plot())

## D. Soal Eksplorasi (3 Soal)

### Soal 1:

**Buatlah program sederhana untuk menampilkan grafik jumlah penonton bioskop selama 5 hari (Senin–Jumat). Data: [120, 150, 170, 100, 200]. Gunakan diagram batang.**

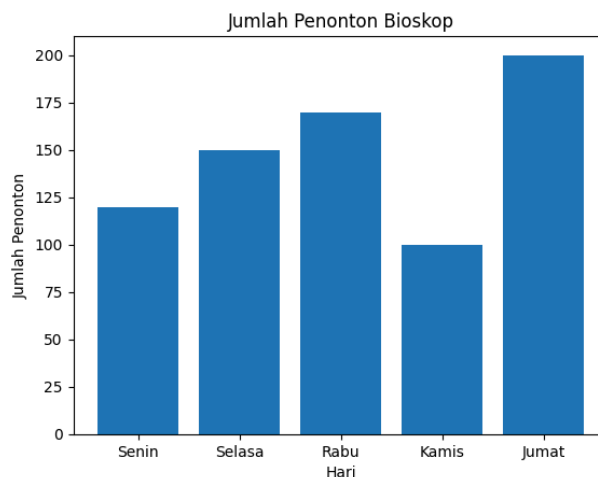
**Kunci Jawaban (contoh siswa):**

```
buat list hari
buat list jumlah_penonton
buat diagram batang
tampilkan grafik
```

Python:

```
import matplotlib.pyplot as plt
hari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat"]
penonton = [120, 150, 170, 100, 200]

plt.bar(hari, penonton)
plt.title("Jumlah Penonton Bioskop")
plt.xlabel("Hari")
plt.ylabel("Jumlah Penonton")
plt.show()
```



### Soal 2:

**Buat grafik garis yang menampilkan perubahan suhu harian selama 7 hari.  
Data suhu: [30, 31, 29, 32, 33, 31, 30].**

**Kunci Jawaban (contoh siswa):**

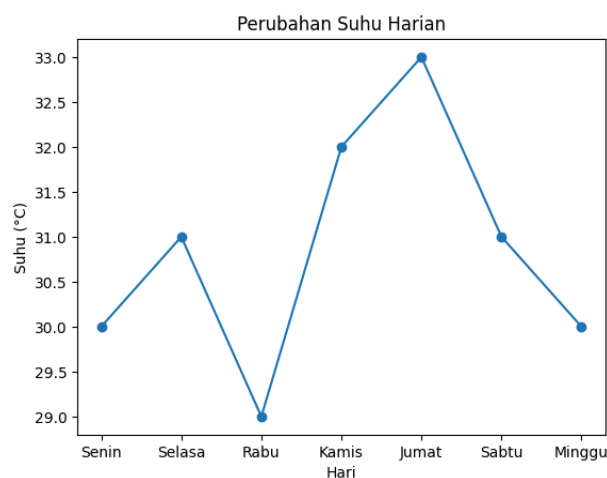
Pseudocode:

```
buat list hari
buat list suhu
buat diagram garis
tampilkan grafik
```

Python:

```
hari =
["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"]
suhu = [30, 31, 29, 32, 33, 31, 30]

plt.plot(hari, suhu, marker="o")
plt.title("Perubahan Suhu Harian")
plt.xlabel("Hari")
plt.ylabel("Suhu (°C)")
plt.show()
```



### Soal 3:

**Catat pengeluaranmu selama seminggu: Makan = 100 ribu, Transport = 50 ribu, Game = 30 ribu, Tabungan = 20 ribu. Buat diagram lingkaran.**

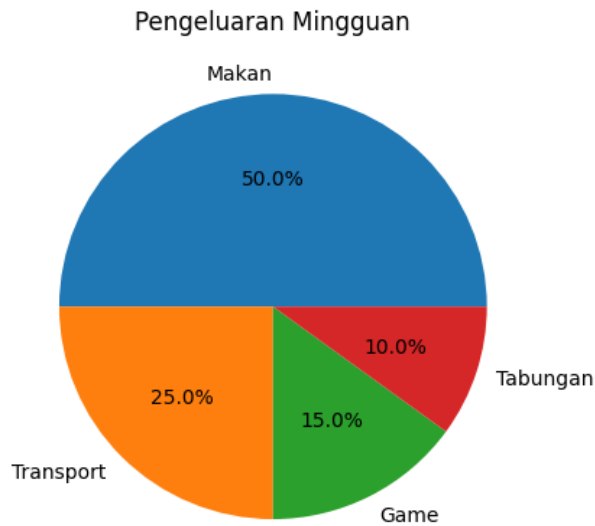
**Kunci Jawaban (contoh siswa):**

```
buat list kategori
buat list jumlah
buat diagram lingkaran
tampilkan grafik
```

Python:

```
aktivitas = ["Makan", "Transport", "Game", "Tabungan"]
uang = [100, 50, 30, 20]

plt.pie(uang, labels=aktivitas, autopct="%1.1f%%")
plt.title("Pengeluaran Mingguan")
plt.show()
```



## Tugas Proyek (5 Soal)

### Proyek 1: Grafik Hasil Voting Film Favorit

Buatlah grafik batang untuk menampilkan hasil voting film favorit di kelas (misalnya: Marvel, DC, Anime).

#### Pseudocode:

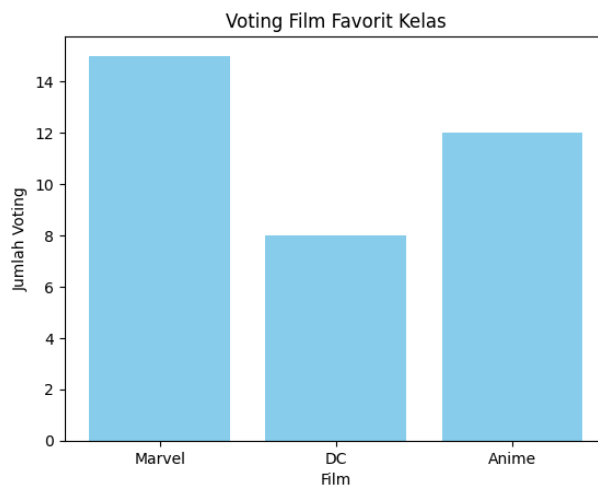
```
buat list nama_film
buat list jumlah_voting
buat diagram batang (film vs voting)
beri judul dan label sumbu
tampilkan grafik
```

### Contoh Python:

```
import matplotlib.pyplot as plt

film = ["Marvel", "DC", "Anime"]
voting = [15, 8, 12]

plt.bar(film, voting, color="skyblue")
plt.title("Voting Film Favorit Kelas")
plt.xlabel("Film")
plt.ylabel("Jumlah Voting")
plt.show()
```



## Proyek 2: Tren Nilai Matematika Selama 6 Bulan

Visualisasikan nilai ulangan Matematika seorang siswa selama 6 bulan dalam grafik garis.

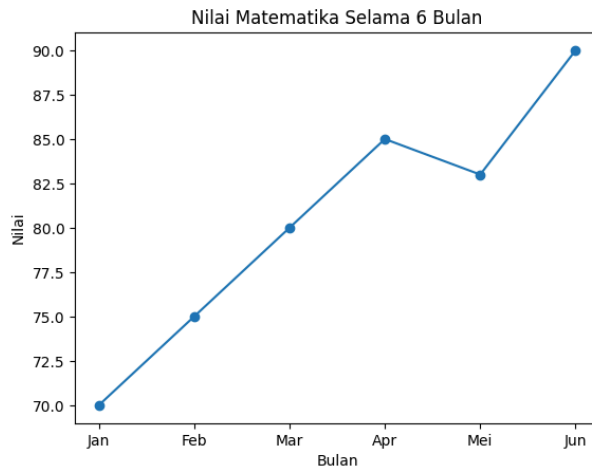
### Pseudocode:

```
buat list bulan
buat list nilai
buat diagram garis (bulan vs nilai)
beri judul, label
tampilkan grafik
```

### Contoh Python:

```
bulan = ["Jan", "Feb", "Mar", "Apr", "Mei", "Jun"]
nilai = [70, 75, 80, 85, 83, 90]

plt.plot(bulan, nilai, marker="o")
plt.title("Nilai Matematika Selama 6 Bulan")
plt.xlabel("Bulan")
plt.ylabel("Nilai")
plt.show()
```



### Proyek 3: Persentase Aktivitas Harian

Buat diagram lingkaran untuk menggambarkan aktivitas harianmu (belajar, tidur, main game, makan).

#### Pseudocode:

```

buat list aktivitas
buat list durasi
buat diagram lingkaran
tambah label
tampilkan grafik

```

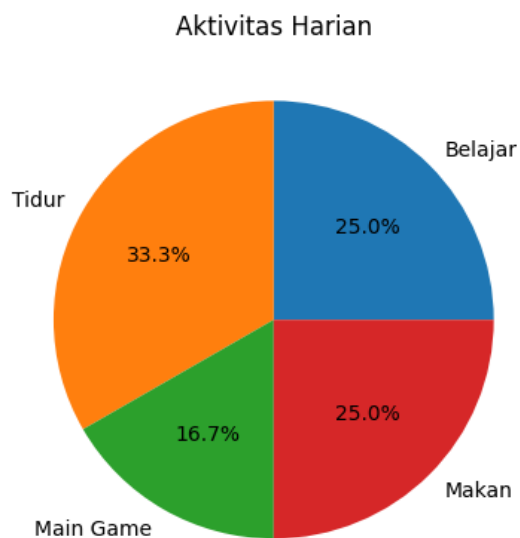
#### Contoh Python:

```

aktivitas = ["Belajar", "Tidur", "Main Game", "Makan"]
durasi = [6, 8, 4, 6]

plt.pie(durasi, labels=aktivitas, autopct="%1.1f%%")
plt.title("Aktivitas Harian")
plt.show()

```



## Proyek 4: Grafik Penjualan Minuman di Kantin

Kamu punya data penjualan minuman (Teh, Kopi, Susu, Jus). Buat grafik batang untuk menunjukkan mana yang paling laku.

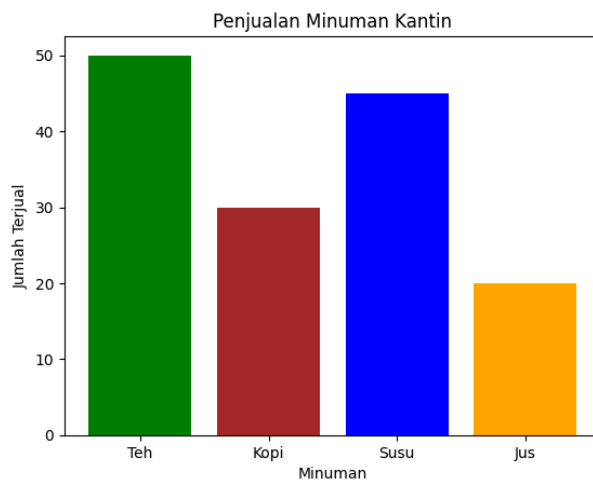
### Pseudocode:

```
buat list minuman
buat list jumlah_terjual
buat diagram batang
beri warna berbeda
tampilkan grafik
```

### Contoh Python:

```
minuman = ["Teh", "Kopi", "Susu", "Jus"]
penjualan = [50, 30, 45, 20]

plt.bar(minuman, penjualan, color=["green", "brown", "blue",
"orange"])
plt.title("Penjualan Minuman Kantin")
plt.xlabel("Minuman")
plt.ylabel("Jumlah Terjual")
plt.show()
```



## Proyek 5: Analisis Cuaca Harian

Buat grafik garis untuk menunjukkan perubahan suhu selama 7 hari terakhir.

### Pseudocode:

```
buat list hari
buat list suhu
buat diagram garis
tambahkan marker
tampilkan grafik
```

### Contoh Python:

```
hari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu",  
        "Minggu"]  
suhu = [30, 32, 31, 33, 29, 28, 30]  
  
plt.plot(hari, suhu, marker="o", linestyle="--", color="red")  
plt.title("Perubahan Suhu Selama 7 Hari")  
plt.xlabel("Hari")  
plt.ylabel("Suhu (°C)")  
plt.show()
```



Semua proyek ini **mudah tapi tetap menantang** karena setiap siswa bisa pakai data yang berbeda: film favorit, nilai, aktivitas harian, penjualan, sampai cuaca. Tujuannya agar siswa **merasakan langsung bagaimana data bisa bercerita lewat gambar**.

# BAB 7: Pengenalan Orange Data Mining

Halo, teman-teman SMK! 🚀 Siap-siap, karena kali ini kita mau kenalan sama sebuah tool super canggih yang bikin mengolah data jadi kayak main game! Nama toolnya **Orange Data Mining**. Kalian yang suka main game strategi, pasti suka banget pakai ini.

## Tujuan Pembelajaran

Setelah sesi ini, kalian bakal bisa:

- **Mengenal** apa itu Orange Data Mining dan kenapa dia keren banget.
- **Membuat alur kerja** sederhana untuk analisis data.
- **Membaca** data dari berbagai sumber.
- **Melihat visualisasi** data dengan cara yang gampang.
- **Menyimpan** hasil analisis kalian.

## Peta Konsep Sederhana

Bayangin kalian mau bikin video tutorial di YouTube.

- **Ide Video (Data)**: Kalian punya data tentang game favorit. Misalnya, daftar hero, win rate, dan jumlah pemain.
- **Tool Editing (Orange Data Mining)**: Ini studio editing kalian. Di sini ada banyak 'widget' atau 'kotak' yang fungsinya beda-beda, kayak potong video, kasih efek suara, atau tambahkan teks.
- **Proses Editing (Alur Kerja)**: Kalian tarik garis dari satu kotak ke kotak lain. Kotak pertama buat ambil data, kotak kedua buat lihat datanya, kotak ketiga buat bikin grafik keren, dan seterusnya.
- **Video Final (Hasil Analisis)**: Videonya siap di-upload! Kalian bisa lihat hero mana yang paling sering menang, atau kapan pemain paling banyak main.

## Apersepsi: Kamu Mau Bikin Tim Futsal Juara!

Kamu mau bikin tim futsal sekolah yang nggak terkalahkan. Tapi kamu bingung, siapa yang harus jadi penyerang, gelandang, atau kiper?

- **Cara Lama**: Kamu coba-coba semua pemain di berbagai posisi. Hasilnya? Kalah terus.
- **Cara Canggih (Orange DM)**: Kamu kumpulkan data semua temanmu: berapa kali cetak gol (goal), berapa kali nge-pass (assist), dan berapa kali menahan bola (defensive skill). Masukin semua datanya ke **Orange Data Mining**, lalu bikin alurnya. Orange bakal bantu kamu 'melihat' mana pemain yang paling cocok buat tiap posisi. Hasilnya? Tim kamu juara! Gampang, kan?

# Penjelasan Teori

## Apa Itu Orange Data Mining?

- **Software gratis & open source** untuk analisis data.
- Bisa dipakai tanpa harus jago *coding*.
- Tampilan visual, tinggal seret-seret *widget*.
- Cocok banget buat pemula, tapi tetap dipakai juga untuk riset serius.

## Cara Instalasi Orange Data Mining

Nah, sebelum main-main sama Orange, tentu kita harus install dulu. Ada beberapa cara:

### 1. Install di Windows

- Kunjungi website resminya: <https://orangedatamining.com>
- Download installer **Orange for Windows**.
- Klik dua kali file installer → ikuti petunjuk → selesai.
- Cari aplikasi **Orange** di menu Start → jalankan.

### 2. Install di Linux (Ubuntu / Debian)

Buka terminal, lalu ketik:

```
sudo apt update
sudo apt install python3-pip -y
pip3 install orange3
```

Setelah selesai, jalankan dengan perintah:

```
orange-canvas
```

### 3. Install di macOS

- Pastikan Python 3 sudah ada.
- Buka terminal, lalu ketik:

```
pip3 install orange3
```

- Jalankan dengan perintah:

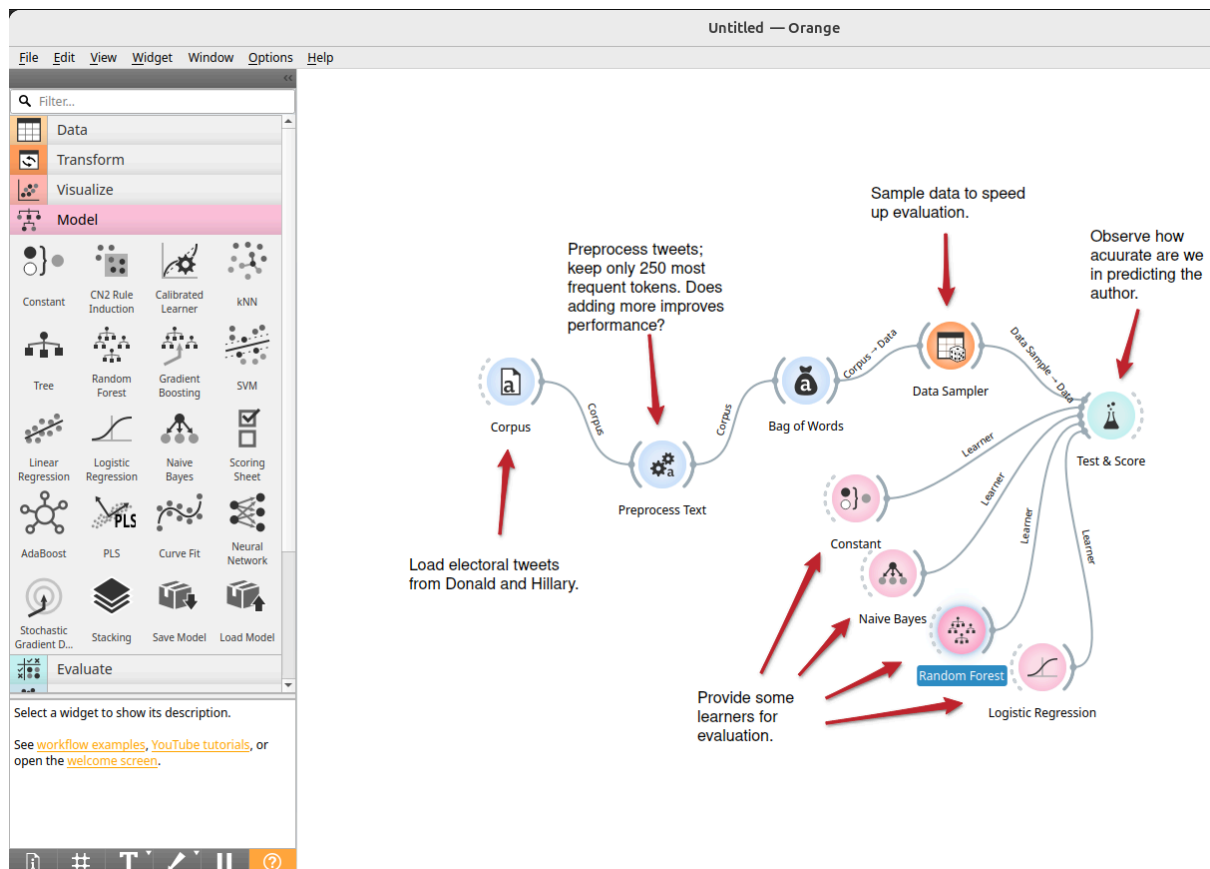
```
orange-canvas
```

### 4. Install via Anaconda (Cross-Platform)

Kalau kalian pakai **Anaconda** (open source juga), caranya lebih gampang:

```
conda install -c conda-forge orange3
```

Setelah itu, Orange siap dipakai. Tinggal buka aplikasinya, dan mulai tarik-tarik widget!



## Kenapa Keren?

- **Visual & Interaktif** → Belajar data kayak main puzzle.
- **Cepat** → Nggak perlu ngetik ribuan baris *Python*.
- **Fleksibel** → Bisa dipakai dari hal simpel (lihat tabel data) sampai hal rumit (machine learning).
- **Terhubung dengan Python & Pandas** → Kalau sudah mahir, bisa lanjutkan ke *coding*.

## Konsep Dasar Orange

Bayangin kalian lagi bikin video YouTube:

1. **Data (Ide Video)**: misalnya daftar hero Mobile Legends, win rate, jumlah pemain.
2. **Widget (Alat Editing)**: kotak-kotak ajaib buat ambil data, lihat data, bikin grafik.
3. **Workflow (Proses Editing)**: nyambungin kotak satu ke kotak lain.
4. **Hasil Analisis (Video Final)**: grafik dan laporan siap dipakai.

## Contoh Kasus atau Ilustrasi

### Kasus 1. Analisis Game Mobile (Win Rate Hero)

Bayangin kalian punya data hero Mobile Legends: nama hero, win rate (%), dan jumlah pemain.

**Ilustrasi:**

Kalian mau tahu hero mana yang paling kuat (*overpower*).

**Pseudocode:**

```
Ambil data_hero dari file
Tampilkan data_hero dalam tabel
Buat grafik sebaran win_rate
Tandai hero dengan win_rate > 60%
Cetak daftar hero overpower
```

### Kasus 2: Penilaian Resep Makanan

Kalian punya 500 resep kue. Ada bahan utama (coklat, keju, dll) dan rating kepuasan siswa.

**Ilustrasi:**

Mau tahu apakah kue coklat lebih populer dari kue keju.

**Pseudocode:**

```
Ambil data_resep dari file
Pisahkan data berdasarkan bahan utama
Hitung rata-rata rating tiap kelompok
Bandingkan rating coklat vs keju
Cetak bahan dengan rating tertinggi
```

### Kasus 3: Survei Kepuasan Siswa di Kantin

Ada data survei kantin: nama makanan, jumlah suara siswa, dan kepuasan (1-5).

**Ilustrasi:**

Kalian ingin tahu jajanan apa yang paling disukai siswa.

**Pseudocode:**

```
Ambil data_kantin dari file
Tampilkan dalam tabel
Buat grafik batang jumlah_suara per makanan
Cari makanan dengan suara terbanyak
Cetak makanan favorit siswa
```

## Kasus 4: Prediksi Cuaca Sederhana

Kalian punya data: suhu, kelembaban, dan curah hujan.

### Ilustrasi:

Ingin prediksi apakah besok hujan atau tidak.

### Pseudocode:

```
Ambil data_cuaca dari file
Bagi data jadi training_set dan testing_set
Latih model sederhana untuk prediksi_hujan
Uji model dengan testing_set
Cetak hasil prediksi besok
```

## Kasus 5: Data Kehadiran Guru

Ada data kehadiran guru: nama guru, jumlah hari hadir, jumlah hari absen.

### Ilustrasi:

Kalian ingin tahu guru mana yang paling rajin.

### Pseudocode:

```
Ambil data_kehadiran dari file
Hitung rata-rata hadir tiap guru
Urutkan data dari hadir tertinggi ke rendah
Tampilkan grafik batang
Cetak guru dengan kehadiran terbaik
```

## Kesimpulan

Dengan *Orange Data Mining*, kasus-kasus di atas bisa diselesaikan hanya dengan **drag & drop widget**, bukan nulis kode panjang. Tapi *pseudocode* ini membantu kalian paham logikanya.

## Contoh Soal dan Pembahasan

### Soal 1: Membaca File CSV

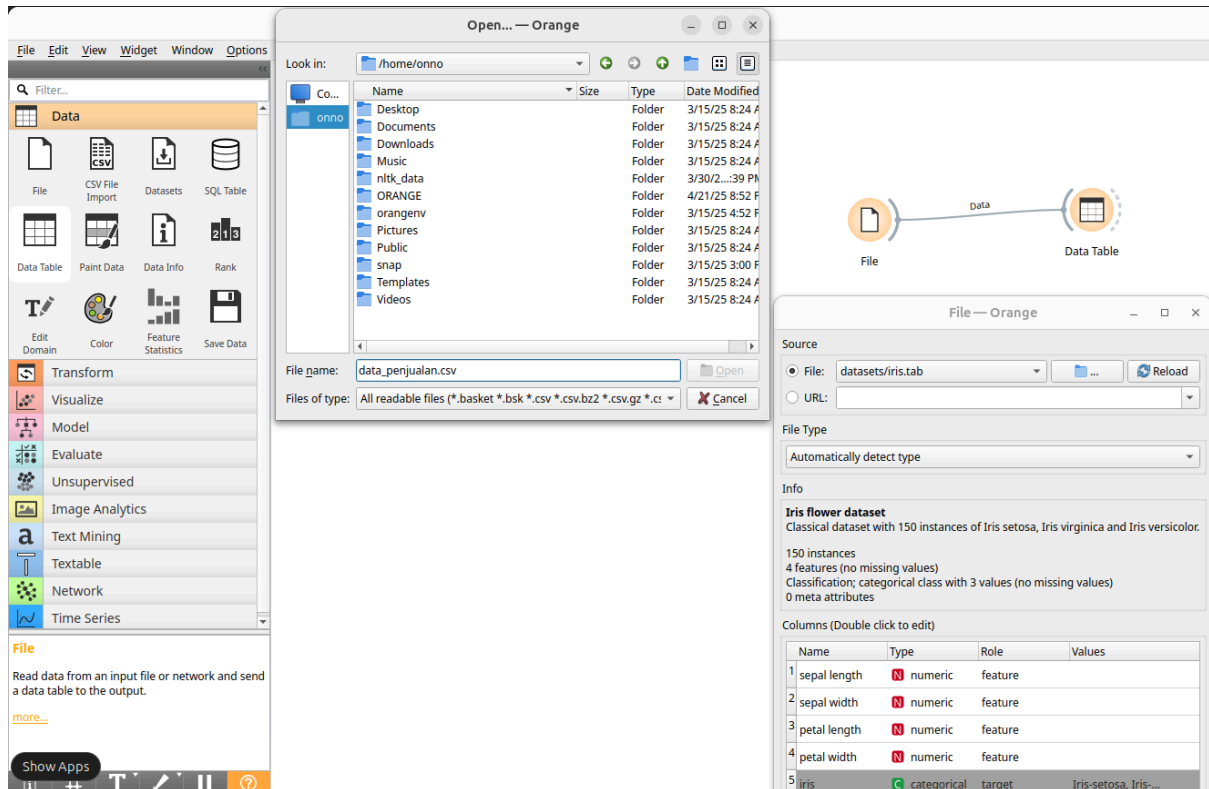
#### Soal:

Kamu punya data `data_penjualan.csv`. Bagaimana langkah sederhana di *Orange Data Mining* untuk membaca file itu?

### Pseudocode Jawaban:

Gunakan widget File  
Pilih file: data\_penjualan.csv  
Tampilkan isi file di Data Table

### Workflow Orange:



### Pembahasan:

Widget **File** berfungsi untuk ambil data dari file eksternal. Supaya bisa dilihat lebih jelas, biasanya dihubungkan ke **Data Table**. Jadi kamu bisa cek apakah datanya sudah masuk dengan benar.

## Soal 2: Melihat Hubungan Antar Data

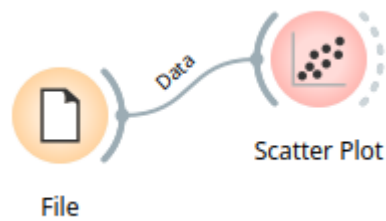
### Soal:

Kamu ingin tahu hubungan antara **harga** dan **jumlah\_terjual**. Widget apa yang paling cocok?

### Pseudocode Jawaban:

Ambil data dari File  
Hubungkan ke Scatter Plot  
X = harga  
Y = jumlah\_terjual

### Workflow Orange:



### Pembahasan:

Widget **Scatter Plot** menampilkan hubungan dua variabel dalam bentuk titik-titik. Dari sini kamu bisa lihat apakah harga yang lebih tinggi membuat penjualan turun atau naik.

## Soal 3: Menggabungkan Dua Data

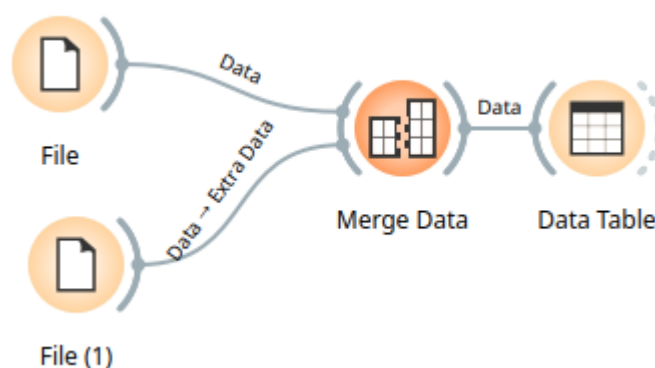
### Soal:

Ada dua file: `data_januari.csv` dan `data_februari.csv`. Bagaimana menggabungkannya?

### Pseudocode Jawaban:

```
Gunakan widget File untuk data_januari.csv  
Gunakan widget File untuk data_februari.csv  
Hubungkan keduanya ke widget Merge Data
```

### Workflow Orange:



### Pembahasan:

Widget **Merge Data** berguna untuk menyatukan dua sumber data. Cocok kalau kalian punya data per bulan dan ingin analisis keseluruhan.

## Soal 4: Menyimpan Data yang Sudah Diolah

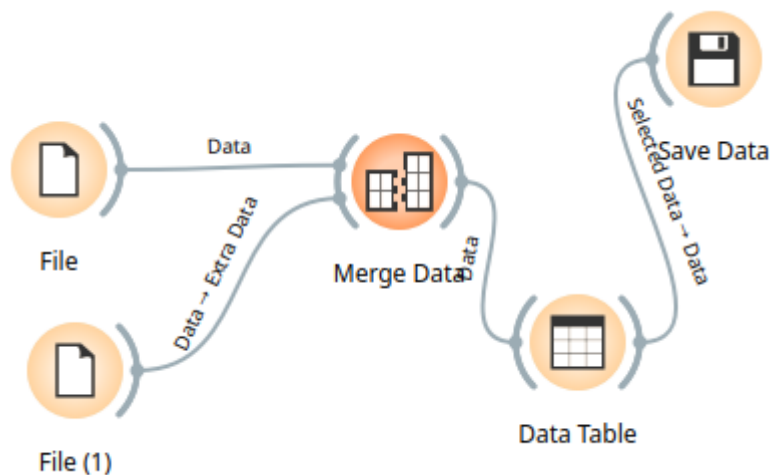
### Soal:

Setelah membersihkan data, bagaimana menyimpan hasilnya ke file baru?

### Pseudocode Jawaban:

```
Gunakan widget Save Data  
Pilih nama file baru (misalnya: data_bersih.csv)  
Simpan hasil data
```

### Workflow Orange:



### Pembahasan:

Widget **Save Data** dipakai untuk menyimpan hasil kerja. Jadi data yang sudah diproses bisa digunakan lagi tanpa harus mengulangi langkah sebelumnya.

## Soal 5: Prediksi Sederhana

### Soal:

Kamu punya data cuaca: suhu, kelembaban, curah hujan. Kamu ingin tahu apakah besok hujan atau tidak. Bagaimana langkahnya?

### Pseudocode Jawaban:

```
Ambil data_cuaca dari File  
Bagi data jadi training_set dan testing_set  
Latih model dengan Training Set  
Uji model dengan Testing Set  
Cetak hasil prediksi: Hujan / Tidak Hujan
```

### Pembahasan:

Di Orange, kamu bisa gunakan widget seperti **Logistic Regression** atau **Tree** untuk prediksi. Walau sederhana, konsepnya sama dengan *machine learning*: data dilatih → diuji → diprediksi.

## Kesimpulan

- Soal di atas melatih cara **membaca data, melihat hubungan antar data, menggabungkan data, menyimpan hasil, dan membuat prediksi sederhana.**
- Semua bisa dilakukan di *Orange Data Mining* hanya dengan **drag & drop widget**, tapi logikanya bisa dituliskan dengan *pseudocode*.
- Jadi, belajar data bisa semudah main puzzle, tapi hasilnya setara ilmu data science beneran!

## Fun Facts!

- Orange Data Mining dikembangkan di **Slovenia**, sebuah negara kecil di Eropa yang keren banget!
- Walaupun tampilannya simple, Orange bisa dipakai untuk proyek yang super rumit, lho! Bahkan buat penelitian ilmiah.
- **Orange** bisa dihubungkan ke **Python** dan **Pandas**! Jadi, kalau ada yang kurang, kalian bisa tambahkan pakai kode Python. Keren, kan?

## Tips Belajar & Menghafal Cepat

- **Mulai dari yang Paling Sempel:** Jangan langsung coba proyek yang rumit. Mulai dari baca data, lalu ditampilkan dalam tabel. Kalau sudah bisa, baru coba bikin grafik.
- **"Challenge" Diri Sendiri:** Coba tantang dirimu, "Aku mau bikin grafik yang bisa membedakan cowok dan cewek di data kehadiran!"
- **Hubungkan dengan Hobi:** Kalau kamu suka game, coba analisis data game. Kalau kamu suka masak, coba analisis data resep.

## Aktivitas Siswa

### Aktivitas 1: Membaca Data Game Mobile

**Tujuan:** Melihat data daftar hero dan win rate.

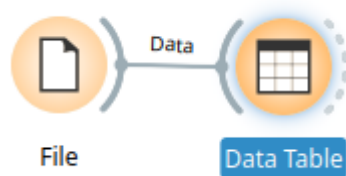
**Langkah:**

1. Buka Orange → seret **widget File**.
2. Pilih file **data\_game.csv**.
3. Hubungkan ke **Data Table** untuk melihat isi.

### Pseudocode:

```
Ambil file data_game.csv  
Tampilkan data di tabel
```

### Workflow Orange:



### Contoh Python:

```
import pandas as pd  
  
data = pd.read_csv("data_game.csv")  
print(data.head())
```

## Aktivitas 2: Membuat Grafik Penjualan Kantin

**Tujuan:** Menampilkan grafik makanan paling laris.

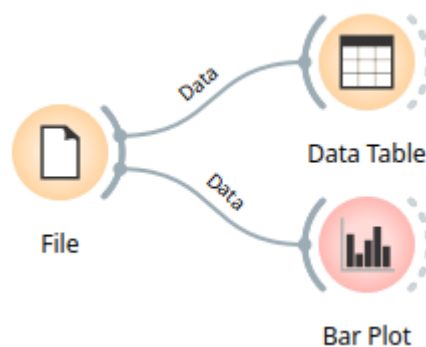
### Langkah:

1. Ambil data **kantin.csv** dengan widget **File**.
2. Hubungkan ke **Bar Plot**.
3. Sumbu X = nama\_makanan, Sumbu Y = jumlah\_terjual.

### Pseudocode:

```
Ambil data_kantin.csv  
Buat grafik batang jumlah_terjual per makanan
```

### Workflow Orange:



### Contoh Python:

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv("kantin.csv")
data.plot(x="nama_makanan", y="jumlah_terjual", kind="bar")
plt.show()
```

## Aktivitas 3: Analisis Data Kehadiran Guru

**Tujuan:** Cari guru yang paling rajin hadir.

### Langkah:

1. Ambil data **kehadiran.csv** dengan widget **File**.
2. Hubungkan ke **Data Table**.
3. Tambahkan **Bar Plot** untuk menampilkan jumlah hadir per guru.

### Pseudocode:

```
Ambil data_kehadiran.csv
Hitung jumlah hadir tiap guru
Tampilkan grafik batang
```

### Contoh Python:

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv("kehadiran.csv")
data.groupby("guru")["hadir"].sum().plot(kind="bar")
plt.show()
```

## Aktivitas 4: Prediksi Cuaca Sederhana

**Tujuan:** Membuat prediksi apakah besok hujan atau tidak.

### Langkah:

1. Ambil data **cuaca.csv** (suhu, kelembaban, curah\_hujan).
2. Gunakan **Tree** atau **Logistic Regression** di Orange.
3. Hubungkan ke **Test & Score** → lihat akurasi.

### Pseudocode:

```
Ambil data_cuaca.csv
Pisahkan training_set dan testing_set
Latih model prediksi
```

Uji model dengan `testing_set`

### Contoh Python (sederhana):

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

data = pd.read_csv("cuaca.csv")
X = data[["suhu", "kelembaban", "curah_hujan"]]
y = data["hujan"]

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
print("Prediksi:", model.predict(X_test))
```

## Aktivitas 5: Menggabungkan Data Penjualan

**Tujuan:** Menggabungkan dua file penjualan bulanan.

### Langkah:

1. Buka **File** → pilih `penjualan_januari.csv`.
2. Buka **File** kedua → pilih `penjualan_februari.csv`.
3. Gunakan **Merge Data** untuk menggabungkan.
4. Tampilkan hasil di **Data Table**.

### Pseudocode:

```
Ambil data_januari.csv
Ambil data_februari.csv
Gabungkan keduanya
Tampilkan hasil
```

### Contoh Python:

```
import pandas as pd

jan = pd.read_csv("penjualan_januari.csv")
feb = pd.read_csv("penjualan_februari.csv")
gabung = pd.concat([jan, feb])
print(gabung.head())
```

## Rangkuman

- Aktivitas di atas melatih kalian **membaca data, membuat grafik, menghitung, menggabungkan data, dan prediksi sederhana**.
- Semua bisa dilakukan di **Orange Data Mining** dengan *drag & drop widget*.

- Kalau mau lebih dalam, kalian bisa cek hasilnya juga di **Python** dengan library open source seperti *pandas*, *matplotlib*, atau *scikit-learn*.

## Rangkuman

Belajar **Orange Data Mining** itu seperti main puzzle. Kita cukup tarik dan hubungkan *widget* tanpa harus nulis kode panjang. Dengan Orange, kita bisa **membaca data, membersihkan data, dan membuat visualisasi** dengan cepat. Tampilan yang **visual dan interaktif** membuat data lebih mudah dipahami, bahkan buat pemula yang baru kenal dunia data.

Selain itu, Orange adalah **software gratis dan open source**. Jadi siapa pun bisa pakai tanpa harus bayar. Kelebihan lainnya, Orange bisa digabung dengan **Python, Pandas, Keras, bahkan neural network**. Artinya, kalau kita ingin naik level, kita bisa mulai dari yang simpel dulu di Orange, lalu lanjut ke *machine learning* yang lebih rumit dengan kode Python.

Penguasaan Orange bisa jadi **bekal berharga di masa depan**. Banyak perusahaan butuh orang yang bisa **mengubah data menjadi informasi**. Mulai dari analisis game, penjualan kantin, prediksi cuaca, sampai data kesehatan bisa dipelajari dengan Orange. Jadi, belajar Orange bukan hanya seru, tapi juga memberi **skill mahal** yang bisa dipakai di dunia nyata.

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. Orange Data Mining adalah software berbayar.  
**Jawaban:** Salah
2. Widget *File* digunakan untuk membaca data dari CSV.  
**Jawaban:** Benar
3. Scatter Plot digunakan untuk melihat hubungan antar dua variabel.  
**Jawaban:** Benar
4. Orange Data Mining tidak bisa terhubung dengan Python.  
**Jawaban:** Salah
5. Workflow di Orange mirip dengan menyambungkan kabel listrik.  
**Jawaban:** Benar

## B. Pilihan Ganda (10 Soal)

1. Orange Data Mining termasuk software:

- a. Open source
- b. Berbayar
- c. Game online
- d. Media sosial

**Jawaban:** a

2. Widget untuk melihat data dalam bentuk tabel adalah:

- a. File
- b. Scatter Plot
- c. Data Table
- d. Save Data

**Jawaban:** c

3. Jika ingin membuat grafik batang, kita gunakan widget:

- a. Bar Plot
- b. Pie Chart
- c. Line Plot
- d. File

**Jawaban:** a

4. Fungsi utama widget *Save Data* adalah:

- a. Membaca file
- b. Menyimpan hasil data yang sudah diolah
- c. Membuat grafik garis
- d. Menghapus data

**Jawaban:** b

5. Orange Data Mining dikembangkan di:

- a. Jepang
- b. Amerika
- c. Slovenia
- d. Indonesia

**Jawaban:** c

6. Untuk melihat tren nilai ujian dari waktu ke waktu, widget terbaik adalah:

- a. Line Plot
- b. Scatter Plot
- c. Data Table
- d. Merge Data

**Jawaban:** a

7. Widget *Merge Data* digunakan untuk:

- a. Menghapus data
- b. Menggabungkan dua dataset
- c. Menyimpan data

d. Menampilkan grafik

**Jawaban:** b

8. Orange bisa dihubungkan dengan:

- a. Python dan Pandas
- b. Excel dan PowerPoint
- c. WhatsApp dan Instagram
- d. TikTok dan YouTube

**Jawaban:** a

9. Untuk memvisualisasikan perbandingan jumlah siswa cowok dan cewek, bisa pakai:

- a. Pie Chart
- b. Data Table
- c. Save Data
- d. File

**Jawaban:** a

10. Dalam Orange, setiap *widget* digambarkan sebagai:

- a. Kotak
- b. Garis
- c. Angka
- d. Huruf

**Jawaban:** a

### C. Isian Singkat (5 Soal)

1. Widget yang digunakan untuk membaca file data adalah \_\_\_\_\_.

**Jawaban:** File

2. Untuk membuat grafik titik-titik hubungan dua variabel digunakan widget \_\_\_\_\_.

**Jawaban:** Scatter Plot

3. Orange Data Mining adalah software \_\_\_\_\_ dan \_\_\_\_\_.

**Jawaban:** gratis, open source

4. Workflow di Orange dibentuk dengan cara \_\_\_\_\_ antar widget.

**Jawaban:** menghubungkan

5. Widget yang digunakan untuk menyimpan hasil data adalah \_\_\_\_\_.

**Jawaban:** Save Data

### D. Soal Eksplorasi (3 Soal)

#### Soal 1

Bayangkan kamu punya data `nilai_siswa.csv` dengan kolom: `nama`, `matematika`, `bahasa`, `ipa`. Buatlah alur kerja di Orange untuk menampilkan grafik batang nilai siswa di tiap mata pelajaran.

**Contoh Jawaban:**

**Pseudocode:**

```
Ambil file nilai_siswa.csv
Tampilkan data dalam Data Table
Buat Bar Plot nilai tiap siswa
```

## Soal 2

Kamu ingin tahu hubungan antara `lama_tidur` siswa dengan `nilai_rata_rata`. Gunakan Orange untuk mencari apakah ada pola tertentu.

**Contoh Jawaban:**

**Pseudocode:**

```
Ambil file tidur.csv
Buat Scatter Plot (X = lama_tidur, Y = nilai_rata_rata)
Amati pola sebaran titik
```

## Soal 3

Sekolahmu mengadakan survei `jajanan_kantin.csv` dengan kolom `nama_makanan` dan `jumlah_vote`. Buat alur kerja untuk menemukan makanan favorit.

**Contoh Jawaban:**

**Pseudocode:**

```
Ambil file jajanan_kantin.csv
Tampilkan data dalam Data Table
Buat Pie Chart jumlah_vote per makanan
Cari makanan dengan vote terbanyak
```

Soal-soal ini bisa langsung dipraktekkan di **Orange Data Mining** dengan *drag & drop*, atau dicoba di **Python** dengan *pandas* dan *matplotlib*.

## Tugas Proyek (5 Soal)

### Proyek 1: Analisis Nilai Ujian

**Soal:**

Kamu punya data nilai ujian siswa (Matematika, Bahasa, IPA). Buat visualisasi siapa yang nilainya tertinggi di setiap mata pelajaran.

**Langkah:**

1. Buka **File** → pilih **nilai\_siswa.csv**.
2. Hubungkan ke **Data Table** untuk cek data.
3. Gunakan **Bar Plot** untuk tampilkan nilai tiap siswa.

**Pseudocode:**

```
Ambil file nilai_siswa.csv  
Tampilkan data dalam tabel  
Buat grafik batang nilai tiap siswa  
Tandai siswa dengan nilai tertinggi
```

**Python:**

```
import pandas as pd  
import matplotlib.pyplot as plt  
  
data = pd.read_csv("nilai_siswa.csv")  
data.plot(x="nama", y=["matematika", "bahasa", "ipa"], kind="bar")  
plt.show()
```

## Proyek 2: Analisis Waktu Tidur Siswa

### Soal:

Kamu punya data lama tidur siswa per hari. Buat grafik hubungan antara lama tidur dengan nilai rata-rata.

### Langkah:

1. Impor data `tidur.csv`.
2. Gunakan **Scatter Plot**.
3. X = lama\_tidur, Y = nilai\_rata\_rata.

### Pseudocode:

```
Ambil file tidur.csv
Buat grafik scatter lama_tidur vs nilai_rata_rata
Cari pola hubungan
```

### Python:

```
data = pd.read_csv("tidur.csv")
data.plot.scatter(x="lama_tidur", y="nilai_rata_rata")
plt.show()
```

## Proyek 3: Analisis Musik Favorit

### Soal:

Data berisi genre musik favorit siswa dan jumlah pendengar. Tampilkan musik paling populer.

### Langkah:

1. Impor data `musik.csv`.
2. Gunakan **Pie Chart** untuk visualisasi.
3. Lihat genre dengan persentase terbesar.

### Pseudocode:

```
Ambil file musik.csv
Buat diagram pie jumlah pendengar per genre
Cari genre dengan pendengar terbanyak
```

### Python:

```
data = pd.read_csv("musik.csv")
data.groupby("genre")["pendengar"].sum().plot(kind="pie",
autopct='%1.1f%%')
plt.show()
```

## Proyek 4: Prediksi Kehadiran Siswa

### Soal:

Kamu punya data **kehadiran.csv** (jumlah hadir, izin, alpa). Buat model sederhana untuk prediksi apakah siswa **rajin** atau **tidak**.

### Langkah:

1. Impor data **kehadiran.csv**.
2. Tentukan variabel X = hadir, izin, alpa; Y = kategori (rajin/tidak).
3. Gunakan **Tree** atau **Logistic Regression** di Orange.
4. Uji akurasi dengan **Test & Score**.

### Pseudocode:

```
Ambil file kehadiran.csv
Bagi data menjadi training_set dan testing_set
Latih model prediksi kategori
Uji akurasi model
```

### Python:

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier

data = pd.read_csv("kehadiran.csv")
X = data[["hadir", "izin", "alpa"]]
y = data["kategori"]

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
print("Prediksi:", model.predict(X_test))
```

## Proyek 5: Analisis Waktu Belajar

### Soal:

Kamu punya data lama belajar harian siswa dan nilai ujian. Buat analisis apakah waktu belajar lebih lama = nilai lebih tinggi.

### Langkah:

1. Impor **belajar.csv**.
2. Gunakan **Scatter Plot** (X = jam\_belajar, Y = nilai).
3. Tambahkan **Linear Regression** untuk lihat tren.

### Pseudocode:

```
Ambil file belajar.csv
```

Buat scatter plot jam\_belajar vs nilai  
Tambahkan garis tren (linear regression)

### Python:

```
from sklearn.linear_model import LinearRegression
import numpy as np

data = pd.read_csv("belajar.csv")
X = data[["jam_belajar"]]
y = data["nilai"]

model = LinearRegression()
model.fit(X, y)

import matplotlib.pyplot as plt
plt.scatter(X, y)
plt.plot(X, model.predict(X), color="red")
plt.show()
```

## Kesimpulan

Lima proyek ini mengajak kalian mengolah data dari dunia nyata: nilai, tidur, musik, kehadiran, hingga belajar. Semua bisa dilakukan dengan **Orange Data Mining** secara visual, atau dicoba juga dengan **Python + Pandas + Machine Learning open source**.

# BAB 8: Mini Proyek – Analisis Dataset Sederhana

Halo, teman-teman SMK! 🌟 Setelah kita belajar banyak tentang Pandas dan Orange Data Mining, sekarang saatnya kita gabungkan semua ilmu itu dalam sebuah **mini proyek!**

## Tujuan Pembelajaran

- **Mempraktikkan** semua materi yang sudah dipelajari.
- **Menganalisis** data dari awal sampai akhir.
- **Menggunakan** Pandas untuk mengolah data dan **Orange** untuk memvisualisasikannya.
- **Menyelesaikan** sebuah proyek nyata.

## Peta Konsep Sederhana

Bayangin kita mau merapikan kamar yang berantakan! 🧹

- **Kamar Berantakan (Dataset):** Ini data mentah yang belum diapa-apain.
- **Ambil Sampah & Rapikan Baju (Pandas):** Kita pakai Pandas buat bersihin data yang kotor (data hilang, data salah).
- **Susun Barang di Rak (Orange Data Mining):** Kita pakai Orange buat susun data jadi grafik-grafik yang rapi dan gampang dilihat.
- **Kamar Bersih (Laporan Akhir):** Kamar (data) kita udah rapi, dan kita bisa tahu barang apa aja yang ada di sana.

## Apersepsi: Kamu Punya Warung Kopi Terhits!

Kamu punya warung kopi yang ramai banget. Tiap hari, kamu catat penjualan: rasa kopi, jumlah terjual, dan harga.

- **Tantangan:** Kamu mau tahu, rasa kopi apa yang paling laku? Dan hari apa aja yang penjualannya paling ramai?
- **Solusi Canggih:** Kita bakal pakai data penjualan warung kopimu untuk menjawab pertanyaan itu. Gampang kan? Ini mirip kayak yang dilakukan data analyst di perusahaan-perusahaan besar, lho!

## Penjelasan Teori

### Kenapa Harus Belajar Mini Proyek?

- Biar ilmu Python dan Pandas kalian kepake nyata.
- Biar tahu gimana *data* bisa bantu bikin keputusan.
- Biar kalian ngerasain kerjaan anak IT beneran.
- Biar kalian punya karya buat ditunjuhin ke guru atau calon bos.

### Alur Mini Proyek Itu Simple

Seorang *data scientist* biasanya lewat jalur ini:

1. **Pengumpulan Data**

Kalian ambil data dari mana aja. Bisa bikin sendiri di Excel, bisa juga cari di internet.  
Contoh: data penjualan warung kopi, data cuaca, data nilai siswa.

2. **Pembersihan Data (*Data Cleaning*)**

Kadang data ada yang kosong, salah, atau doble. Di sinilah Pandas beraksi.  
Contoh: `df.fillna(0)` untuk isi data kosong jadi 0.

3. **Analisis Data (*Data Analysis*)**

Cari pola dari data. Hitung total, rata-rata, atau item paling laku.  
Contoh: `df['jumlah_terjual'].sum()` untuk tahu jumlah kopi yang terjual.

4. **Visualisasi Data (*Data Visualization*)**

Bikin grafik biar gampang dilihat. Kita bisa pakai Orange Data Mining.  
Contoh: *Bar Plot* untuk bandingin penjualan Latte vs Cappuccino.

5. **Interpretasi Hasil**

Dari grafik, kalian bikin kesimpulan.  
Contoh: “Latte paling laku di hari Senin.”

## Analogi Seru

- Dataset = kamar berantakan.
- Pandas = kalian yang beresin baju, buang sampah, rapikan rak.
- Orange = kalian yang hias kamar biar cakep dilihat.
- Hasil akhir = kamar rapi + bisa dipamerin ke orang tua.

## Contoh Kasus Sederhana

Misalnya kalian punya **warung kopi**.

Setiap hari kalian catat penjualan: *tanggal, rasa kopi, jumlah terjual, harga*.

Kalian bisa analisis:

- Kopi apa yang paling laku?
- Hari apa penjualan paling ramai?
- Berapa total uang masuk minggu ini?

Nah, ini sama kayak perusahaan besar loh! Bedanya mereka pakai jutaan data, kalau kita cukup puluhan baris dulu.

## Contoh Kasus atau Ilustrasi

### Kasus 1: Analisis Penjualan E-commerce

Kamu punya data penjualan toko online. Mau tahu barang apa yang paling laku.

#### Ilustrasi:

- Data = nama barang, jumlah terjual, harga
- Tujuan = cari barang dengan jumlah terjual terbanyak

#### Pseudocode:

```
data = baca_csv("penjualan.csv")
bersihkan(data)
total_per_barang = group_by(data, "nama_barang",
sum("jumlah_terjual"))
barang_terlaris = cari_max(total_per_barang)
tampilkan(barang_terlaris)
```

### Kasus 2: Survei Kepuasan Makanan di Kantin

Kamu bikin survei makanan di sekolah. Mau tahu menu apa yang perlu diperbaiki.

#### Ilustrasi:

- Data = nama menu, nilai kepuasan siswa (1-5)
- Tujuan = cari menu dengan nilai terendah

#### Pseudocode:

```
data = baca_csv("survei_kantin.csv")
bersihkan(data)
rata_rata_per_menu = group_by(data, "nama_menu",
mean("nilai_kepuasan"))
menu_terendah = cari_min(rata_rata_per_menu)
tampilkan(menu_terendah)
```

### Kasus 3: Analisis Performa Tim Olahraga

Kamu catat hasil pertandingan futsal. Mau tahu tim paling kuat.

#### Ilustrasi:

- Data = nama tim, jumlah menang, jumlah kalah
- Tujuan = cari tim dengan menang terbanyak

#### Pseudocode:

```

data = baca_csv("pertandingan.csv")
bersihkan(data)
total_menang = group_by(data, "tim", sum("menang"))
tim_terkuat = cari_max(total_menang)
tampilkan(tim_terkuat)

```

## Kasus 4: Tren Penggunaan Media Sosial

Kamu kumpulin data postingan Instagram kelasmu. Mau tahu jenis postingan paling disukai.

### Ilustrasi:

- Data = jenis postingan (foto/video), jumlah like
- Tujuan = cari jenis postingan paling populer

### Pseudocode:

```

data = baca_csv("postingan_ig.csv")
bersihkan(data)
total_like = group_by(data, "jenis_postingan",
sum("jumlah_like"))
paling_populer = cari_max(total_like)
tampilkan(paling_populer)

```

## Kasus 5: Pola Belanja Game Online

Kamu punya data item yang dibeli pemain game. Mau tahu item favorit.

### Ilustrasi:

- Data = nama item, level pemain, jumlah dibeli
- Tujuan = cari item yang paling banyak dibeli oleh pemain level 10

### Pseudocode:

```

data = baca_csv("belanja_game.csv")
bersihkan(data)
data_level10 = filter(data, level == 10)
total_item = group_by(data_level10, "nama_item",
sum("jumlah_dibeli"))
item_favorit = cari_max(total_item)
tampilkan(item_favorit)

```

## Kesimpulan

Mini proyek ini bikin kalian paham:

- Cara ngambil data → bersihin → analisis → bikin grafik → simpulkan.

- Semua bisa dilakukan dengan *open source tools* seperti **Python, Pandas, Orange Data Mining**.
- Mirip kerja detektif: cari petunjuk, lalu pecahkan misteri data.

Kalau kalian konsisten latihan, kalian bisa bilang:

*"Ini proyek saya waktu SMK, saya analisis data beneran!"*

Dan itu bakal keren banget buat masa depan kalian.

## Contoh Soal dan Pembahasan

### Soal 1: Kopi Terlaris (GroupBy + Sum)

**Konteks singkat:** Kamu jual 3 rasa kopi: Latte, Cappuccino, Espresso.

**Tugas:** Tentukan rasa kopi **paling laku** (total tertinggi).

**Data kecil:**

| rasa_kopi  | jumlah_terjual |
|------------|----------------|
| Latte      | 10             |
| Cappuccino | 8              |
| Espresso   | 12             |
| Latte      | 7              |

**Pseudocode:**

```
data = baca_csv("penjualan_kopi.csv")
ringkas = group_by(data, "rasa_kopi", sum("jumlah_terjual"))
kopi_terlaris = cari_idxmax(ringkas)
tampilkan(kopi_terlaris)
```

**Python (Colab):**

```
# Soal 1: Kopi terlaris
import pandas as pd

# Boleh ganti ini dengan pd.read_csv("penjualan_kopi.csv")
df = pd.DataFrame({
    "rasa_kopi": ["Latte", "Cappuccino", "Espresso", "Latte"],
    "jumlah_terjual": [10, 8, 12, 7]
})

ringkas = df.groupby("rasa_kopi")["jumlah_terjual"].sum()
kopi_terlaris = ringkas.idxmax()

print("Ringkas per rasa:\n", ringkas)
print("Kopi terlaris adalah:", kopi_terlaris)
```

**Jawaban:** Latte (total 17).

**Pembahasan:** Hitung total per rasa. Latte 10+7=17, Espresso 12, Cappuccino 8.

Jadi **Latte** paling laku.

*Insight:* inilah analisis dasar pakai *Pandas groupby*.

## Soal 2: Rata-rata Nilai & *Cleaning* (Missing + Duplikat)

**Konteks singkat:** Nilai siswa ada yang kosong dan ada duplikat.

**Tugas:** Bandingkan **rata-rata** sebelum dan sesudah *cleaning*.

**Data kecil:**

| nama | nilai         |
|------|---------------|
| Adit | 80            |
| Bima | (kosong)      |
| Cici | 70            |
| Deni | 90            |
| Adit | 80 (duplikat) |

**Pseudocode:**

```
data = baca_csv("nilai_siswa.csv")
rata_sebelum = mean(nilai tanpa NaN)
data = hapus_duplikat(data)
data = isi_kosong(data, kolom="nilai", nilai=0)
rata_sesudah = mean(nilai)
laporkan(rata_sebelum, rata_sesudah)
```

**Python (Colab):**

```
# Soal 2: Cleaning + rata-rata
import pandas as pd
import numpy as np

df = pd.DataFrame({
    "nama": ["Adit", "Bima", "Cici", "Deni", "Adit"],
    "nilai": [80, np.nan, 70, 90, 80] # duplikat baris Adit,
    dan 1 NaN
})

# rata-rata sebelum (drop NaN)
rata_sebelum = df["nilai"].dropna().mean()

# cleaning
df_bersih = df.drop_duplicates()
df_bersih["nilai"] = df_bersih["nilai"].fillna(0)

rata_sesudah = df_bersih["nilai"].mean()

print("Rata-rata sebelum cleaning:", rata_sebelum)
print("Rata-rata sesudah cleaning:", rata_sesudah)
print("\nData bersih:\n", df_bersih)
```

**Jawaban contoh:** sebelum ~80, sesudah lebih rendah karena NaN diisi 0.

**Pembahasan:** *Cleaning* memengaruhi hasil. Isi kosong dengan 0 menurunkan rata-rata. Pilih strategi *cleaning* sesuai konteks.

### Soal 3: Jajanan Terlaris & Total Uang (Feature Engineering)

**Konteks singkat:** Kantin sekolah.

**Tugas:** Buat kolom **total** = **harga** × **jumlah\_terjual**. Cari jajanan **dengan total terbesar**.

**Data kecil:**

| nama_jajanan | harga | jumlah_terjual |
|--------------|-------|----------------|
| Roti Coklat  | 5000  | 40             |
| Risol        | 3000  | 70             |
| Donat        | 4000  | 50             |

**Pseudocode:**

```
data = baca_csv("kantin.csv")
data.total = harga * jumlah_terjual
jajanan_top = cari_idxmax(data.total)
tampilkan(baris(jajanan_top))
```

**Python (Colab):**

```
# Soal 3: Total uang dan jajanan terlaris
import pandas as pd

df = pd.DataFrame({
    "nama_jajanan": ["Roti Coklat", "Risol", "Donat"],
    "harga": [5000, 3000, 4000],
    "jumlah_terjual": [40, 70, 50]
})

df["total"] = df["harga"] * df["jumlah_terjual"]
top = df.loc[df["total"].idxmax()]

print(df)
print("\nJajanan total terbesar:")
print(top.to_string(index=False))
```

**Jawaban:** Risol (3000×70=210.000).

**Pembahasan:** Ini disebut *feature engineering* sederhana. Lebih bermakna dari sekadar “jumlah terjual”.

### Soal 4: Pilih Grafik yang Tepat (Bar vs Line) + Plot

**Konteks singkat:** Kamu sudah ringkas total penjualan per **rasa\_kopi**:  
Latte 17, Cappuccino 8, Espresso 12.

**Tugas:** Pilih grafik paling cocok untuk **kategori** dan gambar grafiknya.

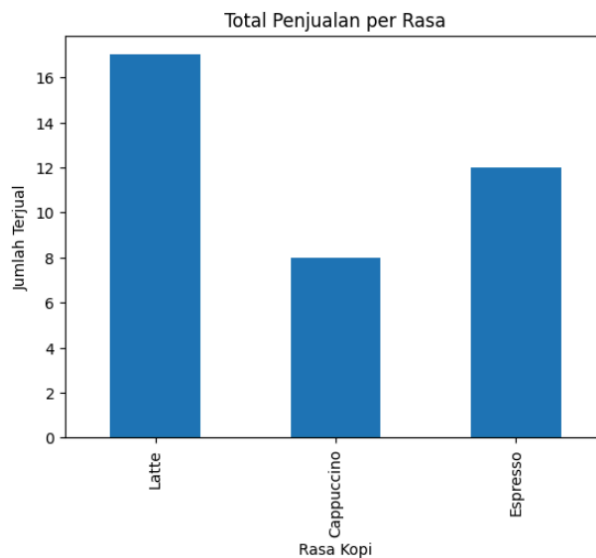
**Pseudocode:**

```
ringkas = baca_csv("ringkas.csv")
jika data_kategori:
    pakai_bar_plot(ringkas)
jika data_waktu:
    pakai_line_plot(ringkas_waktu)
```

**Python (Colab):**

```
# Soal 4: Visualisasi Bar untuk kategori
import pandas as pd
import matplotlib.pyplot as plt

ringkas = pd.Series({"Latte":17, "Cappuccino":8, "Espresso":12})
ax = ringkas.plot(kind="bar", title="Total Penjualan per Rasa")
ax.set_xlabel("Rasa Kopi")
ax.set_ylabel("Jumlah Terjual")
plt.show()
```



**Jawaban:** Bar Plot untuk kategori.

**Pembahasan:** Bar Plot paling jelas membandingkan kategori. Line Plot untuk tren waktu.

## Soal 5: Prediksi Sederhana (Regresi Linear Mini + Bonus Keras)

**Konteks singkat:** Tabel penjualan harian. Terlihat naik stabil.

**Tugas:** Prediksi **hari\_ke = 8**.

**Data kecil:**

| hari_ke | jumlah_terjual |
|---------|----------------|
| 1       | 8              |
| 2       | 10             |
| 3       | 12             |
| 4       | 14             |
| 5       | 16             |

### **Pseudocode:**

```

data = baca_csv("penjualan_harian.csv")
X = kolom("hari_ke")
y = kolom("jumlah_terjual")
model = latih_regresi_linear(X, y)
pred = prediksi(model, 8)
tampilkan(pred)

```

### **Python (Colab, scikit-learn):**

```

# Soal 5: Prediksi linear sederhana
import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.DataFrame({
    "hari_ke": [1,2,3,4,5],
    "jumlah_terjual": [8,10,12,14,16]
})

X = df[["hari_ke"]]
y = df["jumlah_terjual"]

m = LinearRegression().fit(X, y)
pred_8 = m.predict([[8]])[0]

print("Koefisien:", m.coef_[0], "| Intersep:", m.intercept_)
print("Prediksi jumlah_terjual saat hari_ke=8:", round(pred_8,
2))

Bonus neural network mini (Keras, opsional):
# Bonus: Model NN 1-lapisan untuk data kecil (opsional)
# Keras sudah tersedia di Colab
import numpy as np
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

df = pd.DataFrame({
    "hari_ke": [1,2,3,4,5],
    "jumlah_terjual": [8,10,12,14,16]
})

X = df[["hari_ke"]].values.astype("float32")

```

```

y = df["jumlah_terjual"].values.astype("float32")

model = Sequential([Dense(1, input_shape=(1,))])
model.compile(optimizer="adam", loss="mse")
model.fit(X, y, epochs=200, verbose=0)

pred_8_nn = model.predict(np.array([[8.0]], dtype="float32"),
verbose=0)[0][0]
print("Prediksi (Keras) hari_ke=8:", round(float(pred_8_nn), 2))

```

**Jawaban:  $\approx 22$ .**

**Pembahasan:** Pola linear (+2 per hari). *Regresi linear* cukup. *Neural network* juga bisa, tapi berlebihan untuk data kecil. Pilih yang sederhana dulu.

### Tips kilat

- Mulai kecil. Satu *dataset* dulu.
- Rapiin data = *data cleaning* (hapus duplikat, isi kosong).
- Kategori → **Bar Plot**, waktu → **Line Plot**.
- *Machine learning* dipakai jika pola tidak jelas. *Neural network* untuk pola lebih rumit.

### Catatan Colab

- Biasanya Colab sudah punya **pandas, matplotlib, scikit-learn, tensorflow/keras**. Kalau perlu, bisa install manual:

```
!pip install pandas matplotlib scikit-learn tensorflow
```

### Fun Facts!

- Analisis data itu kayak kerja detektif! Kalian harus mencari 'petunjuk' dari data untuk menemukan 'kebenaran'. 🕵️
- **Ollama** dan **OpenWebUI** bisa bantu kalian bikin AI pribadi. Kalian bisa tanya-tanya tentang proyek ini, loh!
- **RAG lokal** itu bikin AI kalian bisa 'ingat' dan 'belajar' dari dokumen di komputer kalian. Jadi, dia bakal pintar sesuai kebutuhan kalian.

### Tips Belajar & Menghafal Cepat

- **Proyek Kecil Dulu:** Jangan langsung ambil proyek yang besar. Mulai dari satu hal dulu, misalnya cuma hitung rata-rata, lalu lanjut ke visualisasi.
- **Tiru, Modifikasi, Lalu Buat Sendiri:** Ambil contoh proyek dari internet, tiru, modifikasi, lalu coba buat proyek dengan ide kalian sendiri.
- **Ngobrol Sama Teman:** Kerjakan proyek ini bareng teman-teman. Diskusi bisa membuka ide-ide baru yang nggak terpikirkan.

## Aktivitas Siswa

### Aktivitas 1: Mengumpulkan Dataset

**Tantangan:** Buat dataset sederhana penjualan kopi.

**Langkah-langkah:**

1. Buka Excel atau Google Sheets.
2. Buat tabel dengan kolom: *tanggal*, *rasa\_kopi*, *jumlah\_terjual*.
3. Isi dengan minimal 10 baris data.
4. Simpan file sebagai **penjualan\_kopi.csv**.

**Pseudocode:**

```
buat_tabel(["tanggal", "rasa_kopi", "jumlah_terjual"])
isi_tabel(10_baris)
simpan_csv("penjualan_kopi.csv")
```

**Python Script:**

```
import pandas as pd

# Buat data sederhana
data = {
    "tanggal": ["2025-09-01", "2025-09-01", "2025-09-02"],
    "rasa_kopi": ["Latte", "Cappuccino", "Espresso"],
    "jumlah_terjual": [10, 8, 12]
}

df = pd.DataFrame(data)
df.to_csv("penjualan_kopi.csv", index=False)
print("Dataset berhasil dibuat!")
```

### Aktivitas 2: Membaca Dataset

**Tantangan:** Baca file **penjualan\_kopi.csv** dengan Pandas.

**Langkah-langkah:**

1. Gunakan fungsi **read\_csv**.
2. Tampilkan 5 baris pertama data.

**Pseudocode:**

```
data = baca_csv("penjualan_kopi.csv")
```

```
tampilkan_head(data, 5)
```

### Python Script:

```
import pandas as pd

df = pd.read_csv("penjualan_kopi.csv")
print(df.head())
```

## Aktivitas 3: Membersihkan Data

**Tantangan:** Cek apakah ada data kosong, lalu isi dengan 0.

### Langkah-langkah:

1. Gunakan `isnull()` untuk cek data kosong.
2. Kalau ada, isi dengan 0 pakai `fillna(0)`.

### Pseudocode:

```
cek = data.isnull()
jika ada_data_kosong:
    data = isi_nilai_kosong(data, 0)
```

### Python Script:

```
print(df.isnull().sum()) # cek data kosong
df = df.fillna(0)        # isi dengan 0
print("Data sudah dibersihkan!")
```

## Aktivitas 4: Analisis Data Sederhana

**Tantangan:** Hitung kopi yang paling laku.

### Langkah-langkah:

1. Kelompokkan data berdasarkan `rasa_kopi`.
2. Jumlahkan kolom `jumlah_terjual`.
3. Cari kopi dengan angka tertinggi.

### Pseudocode:

```
total_per_rasa = group_by(data, "rasa_kopi", sum("jumlah_terjual"))
```

```
kopi_terlaris = cari_max(total_per_rasa)
tampilkan(kopi_terlaris)
```

### Python Script:

```
df_total = df.groupby("rasa_kopi")["jumlah_terjual"].sum()
kopi_terlaris = df_total.idxmax()
print("Kopi terlaris adalah:", kopi_terlaris)
```

## Aktivitas 5: Visualisasi Data

**Tantangan:** Buat grafik penjualan kopi.

### Langkah-langkah:

1. Simpan hasil `df_total` ke file baru.
2. Buka file di Orange Data Mining.
3. Pakai widget *Bar Plot* untuk bikin grafik.

### Pseudocode:

```
simpan_csv(total_per_rasa, "hasil_penjualan.csv")
buka_orange("hasil_penjualan.csv")
buat_grafik_bar("rasa_kopi", "jumlah_terjual")
```

### Python Script (simpan data):

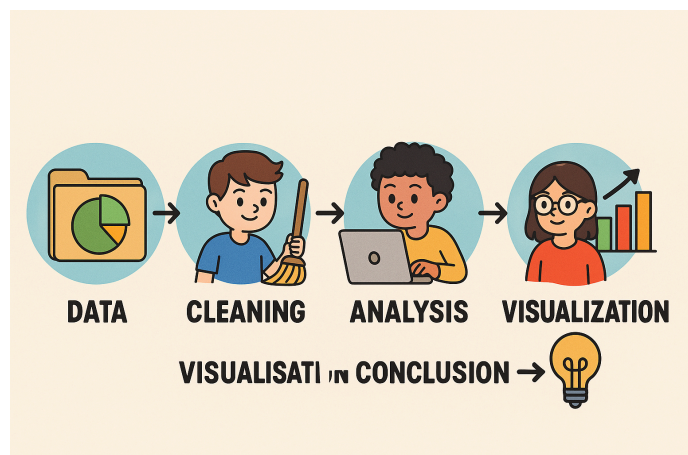
```
df_total.to_csv("hasil_penjualan.csv")
print("Data siap untuk divisualisasikan di Orange!")
```

## Kesimpulan

- Kalian sudah belajar bikin dataset, baca data, bersihin data, analisis, dan visualisasi.
- Semua pakai *open source* tools: **Python + Pandas + Orange**.
- Aktivitas ini bikin kalian siap jadi analis data beneran.

## Rangkuman

Belajar dari mini proyek ini, kita paham bahwa **data mentah** itu ibarat kamar berantakan. Dengan **Python** dan **Pandas**, kita bisa membersihkan dan merapikan data. Lalu, dengan **Orange Data Mining**, kita bisa menyusunnya jadi grafik yang rapi. Semua langkah itu penting supaya kita bisa ambil kesimpulan dari data.



Inilah dasar kerja seorang **data analyst** atau *data scientist*.

Mini proyek ini juga melatih kalian untuk **berpikir kritis**. Mulai dari mengumpulkan data, membersihkan, menganalisis, hingga memvisualisasikan. Tidak semua langkah selalu lancar, kadang harus bolak-balik. Tapi dari situlah kalian belajar bahwa proses analisis data itu mirip seperti jadi **detektif digital**, mencari petunjuk dan menemukan pola. Dengan latihan ini, kalian jadi lebih percaya diri untuk menghadapi dataset yang lebih besar di masa depan.

Yang paling keren, semua teknologi yang dipakai itu **open source**. Kalian bebas belajar dan memakainya tanpa biaya: *Python, Pandas, Orange, machine learning, keras, neural network*. Dengan skill ini, kalian bisa bikin proyek nyata dan menjadikannya **portofolio**. Bayangkan kalian bisa bilang: "Ini hasil analisis saya waktu SMK!" Itu pasti bikin guru bangga dan calon perusahaan kagum. 🚀

## Latihan Soal

### A. Benar atau Salah (5 Soal)

1. `pd.read_csv("data.csv")` dipakai untuk membaca file CSV.  
Jawaban: Benar
2. `df.fillna(0)` dipakai untuk menghapus kolom.  
Jawaban: Salah
3. *Bar Plot* di Orange cocok untuk membandingkan total per kategori.  
Jawaban: Benar
4. `df.groupby("menu")["jumlah"].sum()` menghitung total per menu.  
Jawaban: Benar
5. *Data Cleaning* tidak penting kalau datanya sedikit.  
Jawaban: Salah

### B. Pilihan Ganda (10 Soal)

1. Fungsi untuk melihat 5 baris pertama?  
A. `df.tail()` B. `df.top()` C. `df.head()` D. `df.peak()`  
Jawaban: C
2. Mengganti nilai kosong jadi 0?  
A. `df.dropna(0)` B. `df.fillna(0)` C. `df.zero()` D. `df.na(0)`  
Jawaban: B

3. Menghitung total penjualan seluruh baris kolom `jumlah_terjual`?  
 A. `df.sum()` B. `sum(df)` C. `df['jumlah_terjual'].sum()` D. `df.len()`  
 Jawaban: C
4. Widget Orange untuk sebaran dua angka (suhu vs kelembaban)?  
 A. *Box Plot* B. *Scatter Plot* C. *Line Plot* D. *Pie Chart*  
 Jawaban: B
5. Perintah memilih baris `rasa_kopi == "Latte"`?  
 A. `df("Latte")` B. `df[df=="Latte"]` C. `df['Latte']` D. `df[df['rasa_kopi']=="Latte"]`  
 Jawaban: D
6. Simpan ringkasan ke CSV baru?  
 A. `df.save_csv()` B. `df.to_csv("ringkas.csv")` C. `pd.save("ringkas.csv")` D. `csv.write(df)`  
 Jawaban: B
7. Library Python untuk *machine learning* sederhana (regresi linear)?  
 A. `matplotlib` B. `pandas` C. `sklearn` D. `keras.datasets`  
 Jawaban: C
8. Langkah yang tepat dalam mini proyek?  
 A. Visualisasi → Cleaning → Interpretasi  
 B. Cleaning → Visualisasi → Pengumpulan  
 C. Pengumpulan → Cleaning → Analisis → Visualisasi → Interpretasi  
 D. Analisis → Pengumpulan → Cleaning  
 Jawaban: C
9. Untuk mencari kategori terlaris dari `df_total` (Series hasil groupby)?  
 A. `df_total.max()`  
 B. `df_total.idxmax()`  
 C. `df_total.argmax()`  
 D. `df_total.big()`  
 Jawaban: B
10. *Neural network* sederhana di Keras fokus pada?  
 A. Menggambar grafik  
 B. Menyimpan file  
 C. Memprediksi pola dari data  
 D. Menghapus *missing values*  
 Jawaban: C

### C. Isian Singkat (5 Soal)

1. Perintah impor Pandas adalah `_____ as pd`.  
Jawaban: pandas
2. Perintah untuk cek jumlah nilai kosong per kolom: `df._____().sum()`  
Jawaban: isnull
3. Lengkapi: `df.groupby("menu")["jumlah_terjual"]._____()` untuk total.  
Jawaban: sum
4. Lengkapi kode memilih tanggal tertentu: `df[df["tanggal"] == "2025-09-01"]` kolom yang dibandingkan adalah \_\_\_\_\_.  
Jawaban: tanggal
5. Simpan DataFrame `df_hasil` ke file `hasil.csv` tanpa indeks:  
`df_hasil.to_csv("hasil.csv", _____=False)`  
Jawaban: index

### D. Soal Explorasi (3 Soal)

Tuliskan jawaban dalam 1–3 paragraf pendek. Sertakan *pseudocode* dan jika perlu kode Python singkat.

#### Soal 1:

##### Optimasi Grafik Penjualan Kopi

Kamu punya `penjualan_kopi.csv` (tanggal, rasa\_kopi, jumlah\_terjual). Buat ringkasan total per rasa. Bandingkan *Bar Plot* vs *Line Plot* untuk data ini. Grafik mana paling jelas untuk kategori? Jelaskan pilihanmu.

Contoh Jawaban (ringkas):

*Pseudocode:*

```
data = baca_csv("penjualan_kopi.csv")
ringkas = group_by(data, "rasa_kopi", sum("jumlah_terjual"))
simpan_csv(ringkas, "ringkas.csv")
buat_grafik_bar(ringkas)
```

Alasan: *Bar Plot* lebih jelas untuk kategori `rasa_kopi`. *Line Plot* cocok untuk tren waktu. Di sini fokus pada perbandingan rasa, jadi *Bar Plot* dipilih.

Python:

```
import pandas as pd
df = pd.read_csv("penjualan_kopi.csv")
tot = df.groupby("rasa_kopi")["jumlah_terjual"].sum()
tot.to_csv("ringkas.csv")
```

## Soal 2:

### Cek Kualitas Data Sebelum Analisis

Kamu menemukan nilai kosong dan duplikat pada `nilai_siswa.csv` (nama, jam\_belajar, nilai). Tunjukkan langkah cepat *data cleaning* dan dampaknya pada hasil analisis rata-rata nilai.

Contoh Jawaban (ringkas):

*Pseudocode:*

```
data = baca_csv("nilai_siswa.csv")
hapus_duplikat(data)
isi_kosong(data, 0)
rata = mean(data["nilai"])
laporkan(rata)
```

Dampak: Tanpa *cleaning*, rata-rata bias. Setelah isi kosong/hapus duplikat, rata-rata lebih akurat.

Python:

```
import pandas as pd
df = pd.read_csv("nilai_siswa.csv")
df = df.drop_duplicates()
df = df.fillna(0)
print(df["nilai"].mean())
```

## Soal 3:

### Mini *Machine Learning* Prediksi Penjualan Harian

Pakai `penjualan_harian.csv` (hari\_ke, jumlah\_terjual). Bangun model regresi linear untuk prediksi hari\_ke=8. Jelaskan kenapa model sederhana sudah cukup untuk data kecil.

Contoh Jawaban (ringkas):

*Pseudocode:*

```
data = baca_csv("penjualan_harian.csv")
X = kolom("hari_ke")
y = kolom("jumlah_terjual")
model = regresi_linear_latih(X, y)
pred = prediksi(model, 8)
laporkan(pred)
```

Alasan: Data kecil, pola linear sederhana, cepat, mudah dipahami. *Overfitting* lebih kecil dibanding *neural network* besar.

Python:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
df = pd.read_csv("penjualan_harian.csv")
X = df[["hari_ke"]]
y = df["jumlah_terjual"]
m = LinearRegression().fit(X, y)
print(m.predict([[8]]))
```

## Tugas Proyek (5 Soal)

### Proyek 1: Analisis Kehadiran Siswa

**Tantangan:** Buat dataset kecil tentang kehadiran siswa di kelas (nama, jumlah hadir, jumlah izin, jumlah alfa). Cari tahu siapa siswa paling rajin hadir.

#### Pseudocode Jawaban:

```
Baca dataset ke Pandas
Hitung persentase hadir tiap siswa
Cari siswa dengan persentase tertinggi
```

#### Python Sederhana:

```
import pandas as pd
df = pd.read_csv("kehadiran.csv")
df["persentase_hadir"] = (df["hadir"] /
(df["hadir"]+df["izin"]+df["alfa"])) * 100
print(df.sort_values("persentase_hadir",
ascending=False).head(1))
```

### Proyek 2: Analisis Penjualan Jajanan di Kantin

**Tantangan:** Buat dataset menu kantin (nama\_jajanan, harga, jumlah\_terjual). Cari jajanan paling laku dan hitung total uang masuk.

#### Pseudocode Jawaban:

```
Baca dataset ke Pandas
Hitung total penjualan per menu = harga * jumlah_terjual
Cari menu dengan total penjualan terbesar
```

#### Python Sederhana:

```
df = pd.read_csv("kantin.csv")
df["total"] = df["harga"] * df["jumlah_terjual"]
print(df.sort_values("total", ascending=False).head(1))
```

## Proyek 3: Prediksi Nilai Siswa dengan Machine Learning

**Tantangan:** Buat dataset nilai siswa (jam\_belajar, nilai). Gunakan model sederhana untuk memprediksi nilai berdasarkan jam belajar.

### Pseudocode Jawaban:

```
Baca dataset ke Pandas
Pisahkan data fitur (jam_belajar) dan target (nilai)
Latih model regresi linear
Prediksi nilai jika jam belajar = 5
```

### Python Sederhana:

```
from sklearn.linear_model import LinearRegression
import numpy as np

df = pd.read_csv("nilai.csv")
X = df[["jam_belajar"]]
y = df["nilai"]

model = LinearRegression()
model.fit(X, y)
print(model.predict([[5]]))
```

## Proyek 4: Analisis Genre Musik Favorit

**Tantangan:** Buat dataset (nama\_siswa, genre\_musik). Cari genre musik yang paling banyak dipilih.

### Pseudocode Jawaban:

```
Baca dataset ke Pandas
Hitung jumlah siswa per genre
Tampilkan genre dengan jumlah terbanyak
```

### Python Sederhana:

```
df = pd.read_csv("musik.csv")
print(df["genre_musik"].value_counts().head(1))
```

## Proyek 5: Analisis Data Cuaca

**Tantangan:** Buat dataset cuaca sederhana (tanggal, suhu, kelembaban). Cari hari dengan suhu tertinggi dan buat grafik sederhana.

### Pseudocode Jawaban:

```
Baca dataset ke Pandas
Cari data dengan suhu maksimum
```

Visualisasikan suhu harian dengan plot

### Python Sederhana:

```
import matplotlib.pyplot as plt

df = pd.read_csv("cuaca.csv")
print(df.loc[df["suhu"].idxmax()])
df["suhu"].plot(kind="line")
plt.show()
```

### Penutup

- **Proyek ini seru karena kalian belajar sambil praktek nyata.**
- Semua pakai alat *open source*, jadi gratis dan bisa dikembangkan lagi.
- Hasil proyek ini bisa kalian pakai buat **portofolio pribadi**.

# Institut Teknologi Tangerang Selatan (ITTS) dan Penulis.

Para penulis adalah **dosen aktif di Institut Teknologi Tangerang Selatan (ITTS)**, sebuah **institusi pendidikan tinggi berbasis teknologi** yang berperan aktif dalam pengembangan **solusi digital** dan *kecerdasan buatan (AI)*. Tidak hanya mengajar, mereka juga merupakan **praktisi dan inovator** yang menghasilkan **karya-karya nyata di bidang teknologi mandiri**.

Berlokasi di **Komplek Komersial BSD, Serpong Utara, Tangerang Selatan**, ITTS hadir sebagai **pusat inovasi** yang mengintegrasikan **pembelajaran, penelitian, dan pengembangan teknologi secara menyeluruh**. Fokus utama ITTS mencakup bidang strategis seperti *AI, Machine Learning, Deep Learning*, jaringan, *cloud*, keamanan siber, pemrograman, sistem informasi, dan multimedia.

**ITTS menawarkan lingkungan akademik yang dinamis dan kolaboratif**. Setiap minggu, diselenggarakan berbagai *workshop, demo teknologi*, seminar, hingga *webinar*—mayoritas **gratis dan tersedia secara terbuka di YouTube**. Ini menciptakan **kultur belajar yang aktif dan relevan dengan perkembangan industri**.

**Fasilitas ITTS menunjang kegiatan riset dan inovasi secara optimal**. Lab komputer dilengkapi *GPU RTX 4060* untuk eksperimen *AI*, serta **laboratorium teknologi dengan puluhan server, perangkat IoT, sistem komunikasi radio, dan infrastruktur jaringan jarak jauh**. Dari ekosistem ini lahir **produk-produk mandiri** seperti **ChatGPT versi lokal** untuk institusi pendidikan dan **sistem perpustakaan digital yang dapat diakses tanpa koneksi internet**.

**Daya saing ITTS** juga tercermin dari karakter dosen dan mahasiswa yang tidak hanya akademik, tetapi juga **produktif secara praktis**. Mahasiswa didorong aktif menjadi **narasumber, penulis, dan peneliti** sejak dini. Banyak diantaranya telah menghasilkan **karya buku, artikel ilmiah, serta terlibat langsung dalam proyek-proyek teknologi**.

**Kolaborasi strategis dengan berbagai mitra industri**—seperti *IDCloudHost, Ubuntu, XecureIT, Dinas KOMINFO Tangerang Selatan*, berbagai *ISP*, dan komunitas IT nasional—menjadi **jembatan nyata antara kampus dan dunia profesional**. Hasilnya, **lebih dari 75% mahasiswa ITTS telah bekerja bahkan sebelum lulus, dan hampir seluruh lulusan langsung terserap industri**.

**ITTS bukan hanya tempat belajar, tapi tempat mencipta**. Dosen sebagai **penulis dan praktisi**, mahasiswa sebagai **peneliti dan kontributor**—semuanya berperan **membangun ekosistem teknologi yang mandiri dan berdaya saing tinggi**.

## Institut Teknologi Tangerang Selatan (ITTS)

Komplek Komersial BSD

Jl. Raya Serpong No. Kav. 9, Lengkong Karya, Serpong Utara

Kota Tangerang Selatan, Banten 15331

☎ (+62) 877-7277-1775

✉ [info@itts.ac.id](mailto:info@itts.ac.id)

🌐 [www.itts.ac.id](http://www.itts.ac.id)