

# Getting Started with Python

Samsung Innovation Campus 7





# Galuh Ramaditya

Mentor

*Data Engineer*

*LinkedIn:*

<https://www.linkedin.com/in/galuh-ramaditya/>

- **Data Engineer**

at Kredivo Group (2022–present)

- **Data Mediation**

at XL (2023–2024)

- **Data Engineer**

at SIRCLO (2021–2022)

# Pengenalan Data

# Masalah



Masalah umum yang dihadapi banyak bisnis, di mana informasi penting terabaikan dan peluang untuk berkembang terlewatkan.

# Solusi

## “Mengubah Data Menjadi Wawasan”

Data yang tepat:

- Laporan penjualan harian untuk mengetahui jenis kopi mana yang paling laris.
- Mengetahui jam-jam sibuk untuk mengoptimalkan jumlah staf.
- Menganalisis data promosi untuk melihat seberapa besar dampaknya terhadap penjualan.

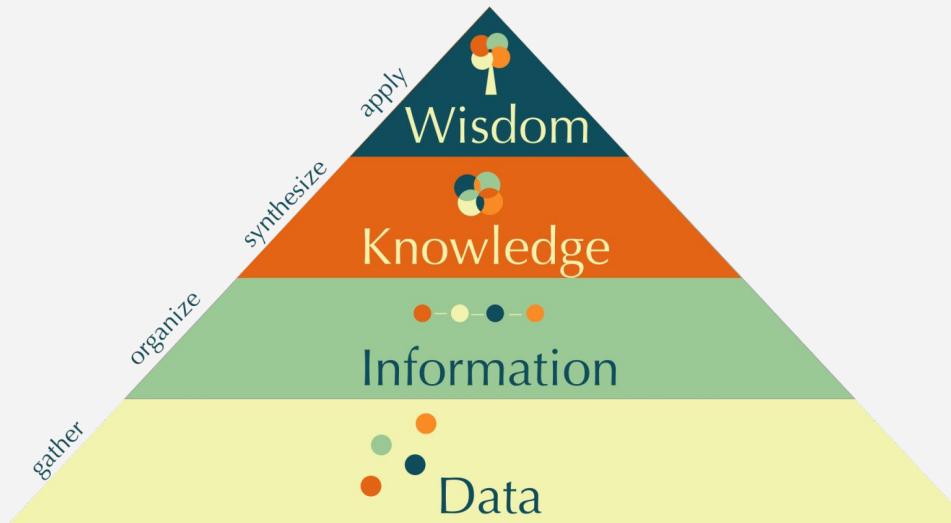
**Data yang terkelola dengan baik memberikan wawasan untuk membuat keputusan yang lebih strategis.**

# Pengertian

**Data** adalah kumpulan **fakta** mentah, angka, atau informasi yang belum diolah.



# DIKW Pyramid



## Data:

- Angka "50"
- nama "Cappuccino"
- harga "Rp25.000".

## Informasi:

"Penjualan Cappuccino hari ini adalah 50 gelas dengan total pendapatan Rp1.250.000."

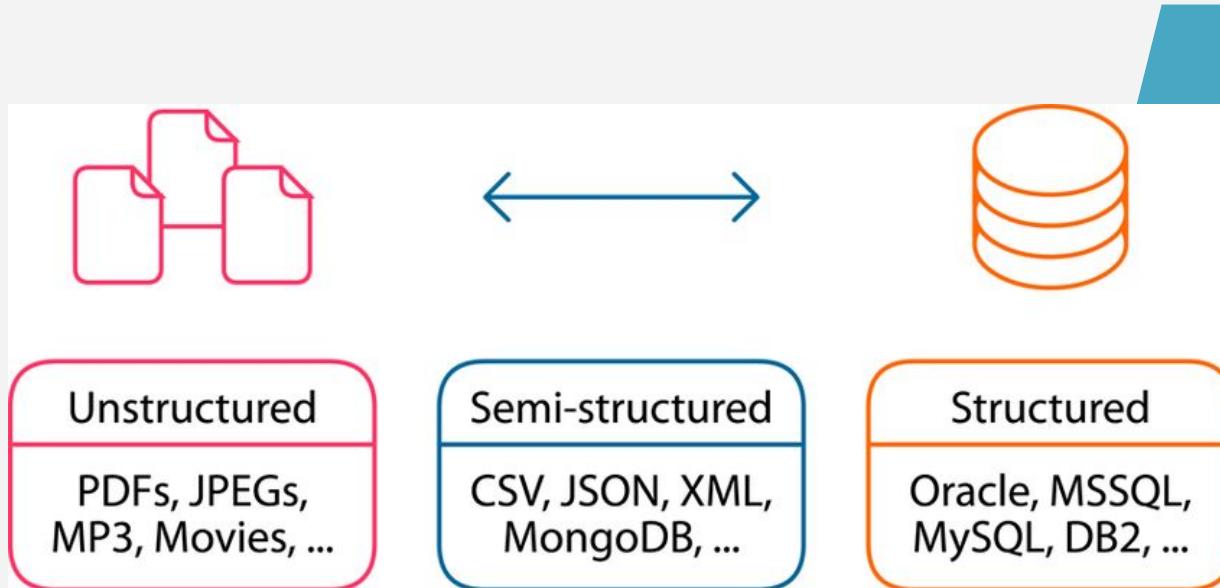
## Pengetahuan:

"Cappuccino adalah produk paling laris di toko ini dan berkontribusi besar pada pendapatan."

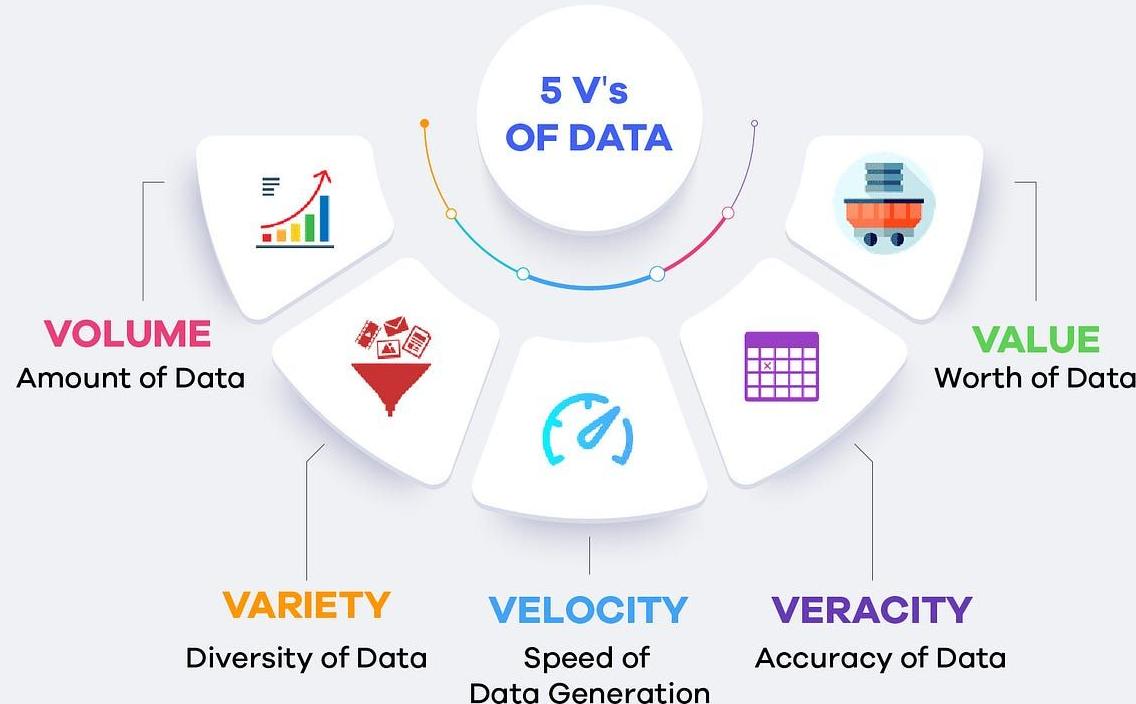
## Wawasan:

"Karena Cappuccino sangat populer, kita bisa membuat promosi bundling 'Cappuccino + Kue' untuk meningkatkan penjualan produk lain."

# Jenis



# Big Data

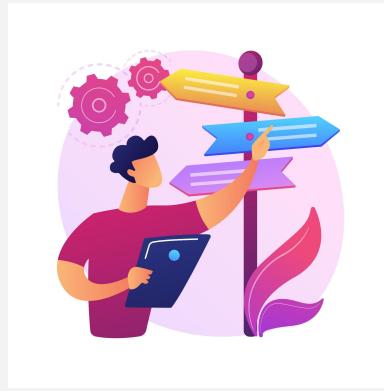


# Case Study: Coffe Shop

- **Volume**  
Seluruh data penjualan, stok, dan pelanggan dari 1000 cabang selama 5 tahun terakhir.
- **Velocity**  
Data pesanan yang masuk secara real-time dan notifikasi dari aplikasi pemesanan online yang terus-menerus mengalir.
- **Variety**  
Meliputi data transaksi (terstruktur), gambar produk dari kamera pengawas (tidak terstruktur), dan review pelanggan (teks tidak terstruktur).
- **Veracity**  
Memastikan semua data penjualan yang dicatat oleh sistem POS adalah akurat dan tidak ada duplikasi.
- **Value**  
Menganalisis data ini untuk menemukan tren, membuat rekomendasi produk yang dipersonalisasi, dan mengoptimalkan manajemen stok, yang semuanya meningkatkan keuntungan.

# Manfaat

Pengambilan Keputusan  
yang Lebih Baik



Efisiensi Operasional

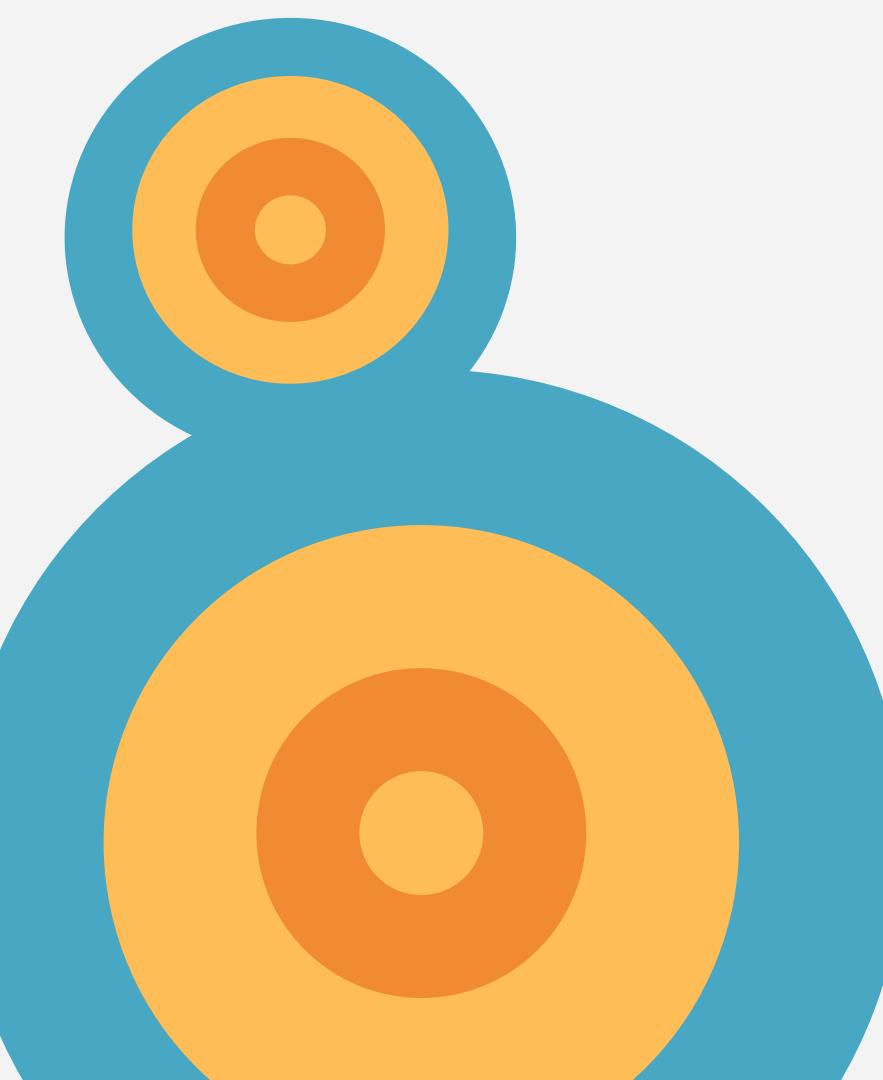


Prediksi Masa Depan



Personalisasi Layanan





Pertanyaan?

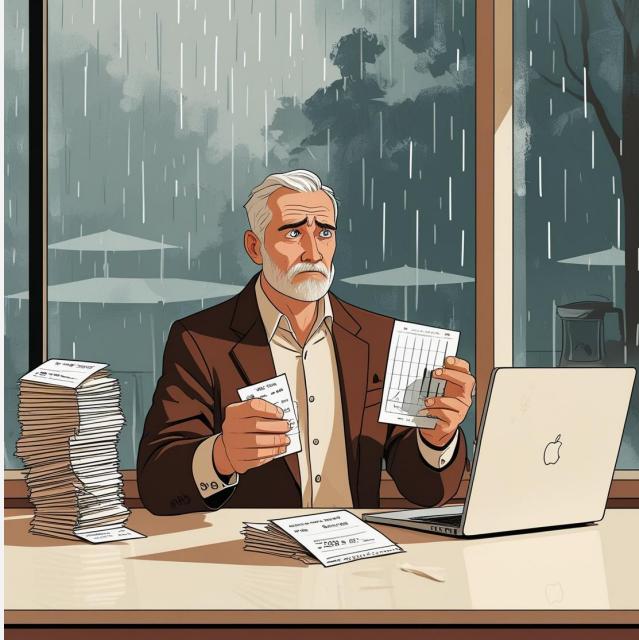


**It's Kahoot Time..**

# Pengantar Machine Learning



# Masalah



Bagaimana kita bisa membuat komputer memprediksi dan menemukan pola dari data secara otomatis, tanpa harus menulis setiap aturan satu per satu?

# Solusi

## Machine Learning (ML)

"Mesin" yang bisa belajar dari **data historis** untuk **menemukan pola-pola** yang rumit, dan **membuat prediksi atau keputusan** tanpa perlu diprogram secara eksplisit.

Kita bisa:

- **Mendeteksi Pola Kompleks**

Bisa menemukan hubungan yang sangat kompleks antara ratusan variabel.

- **Membuat Prediksi Otomatis**

Model yang secara otomatis memprediksi probabilitas seorang nasabah akan gagal bayar.

- **Mengurangi Human Error**

Keputusan yang dihasilkan oleh model ML lebih konsisten dan objektif.

# Machine Learning vs. Traditional Programming



## Traditional Programming

Dalam pemrograman tradisional, seorang *developer* harus menulis setiap aturan secara eksplisit.

- **Input:** Data transaksi.
- **Program:** Aturan yang ditulis oleh manusia.
- **Output:** Rekomendasi.

**Keterbatasan:** Aturan ini sangat kaku dan tidak bisa beradaptasi. Jika ada produk baru atau preferensi pelanggan berubah, Anda harus mengubah kode secara manual. Program ini hanya bisa merekomendasikan apa yang sudah Anda ketahui dan masukkan ke dalam kode.

**Contoh Kasus:** Jika Anda ingin merekomendasikan produk, Anda harus menulis logika seperti ini:

IF pelanggan membeli "Kopi Latte":

REKOMENDASIKAN "Donat".

ELSE IF pelanggan membeli "Cappuccino":

REKOMENDASIKAN "Croissant".

ELSE:

REKOMENDASIKAN "Menu Promo".

# Machine Learning vs. Traditional Programming



## Machine Learning

Dalam Machine Learning, prosesnya dibalik. Kita memberikan data dan membiarkan algoritma menemukan polanya.

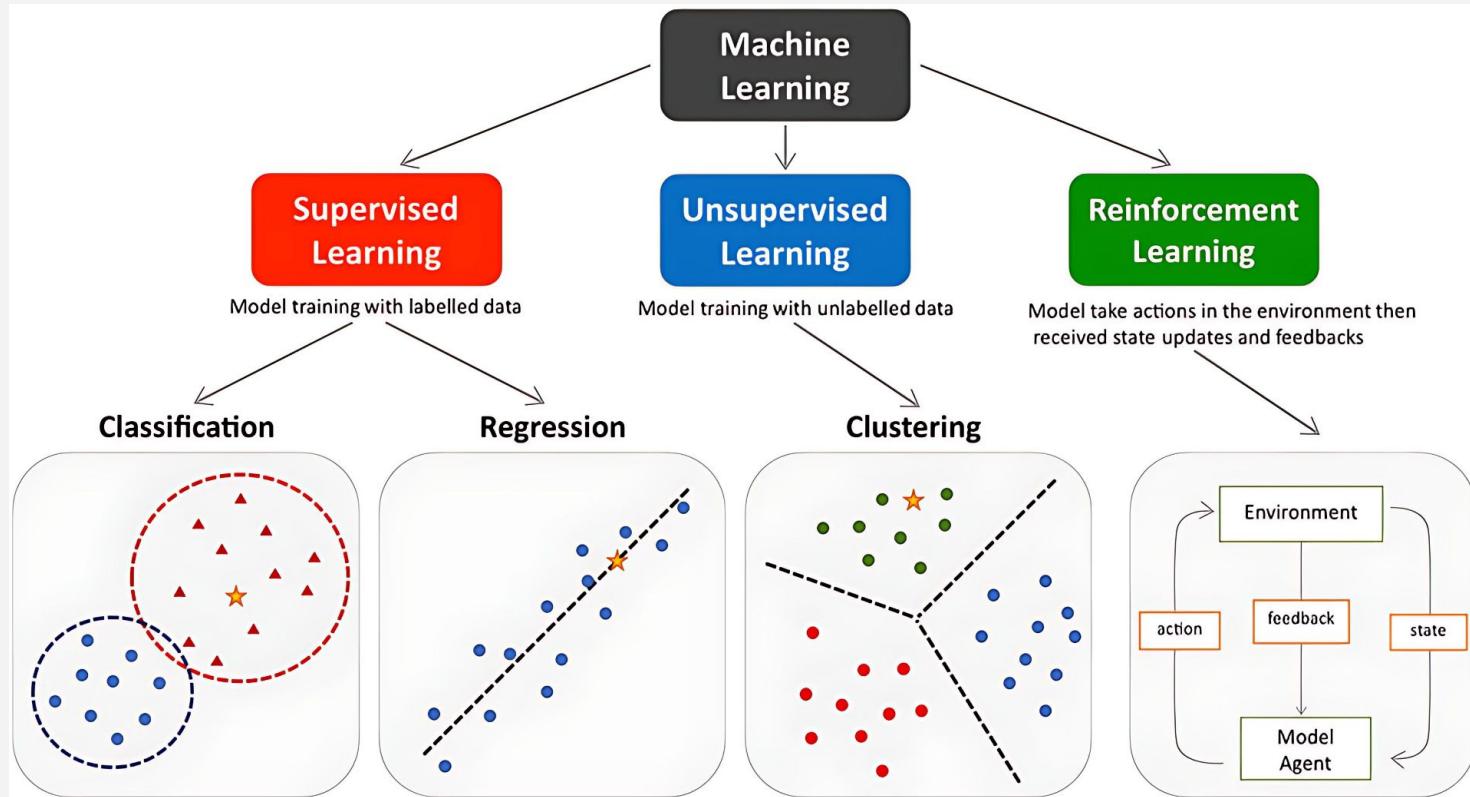
- **Input:** Data transaksi historis (pelanggan, produk yang dibeli bersamaan).
- **Program:** Model yang "belajar" dari data.
- **Output:** Aturan (pola atau model).

**Keunggulan:** Model ini secara otomatis menemukan pola yang mungkin tidak pernah terpikirkan oleh manusia, bahkan dari jutaan data. Model akan merekomendasikan "Muffin" jika ada pelanggan membeli "Teh Matcha" tanpa perlu Anda menulis aturan itu. Jika tren berubah, Anda hanya perlu melatih ulang model dengan data terbaru.

## Contoh Kasus:

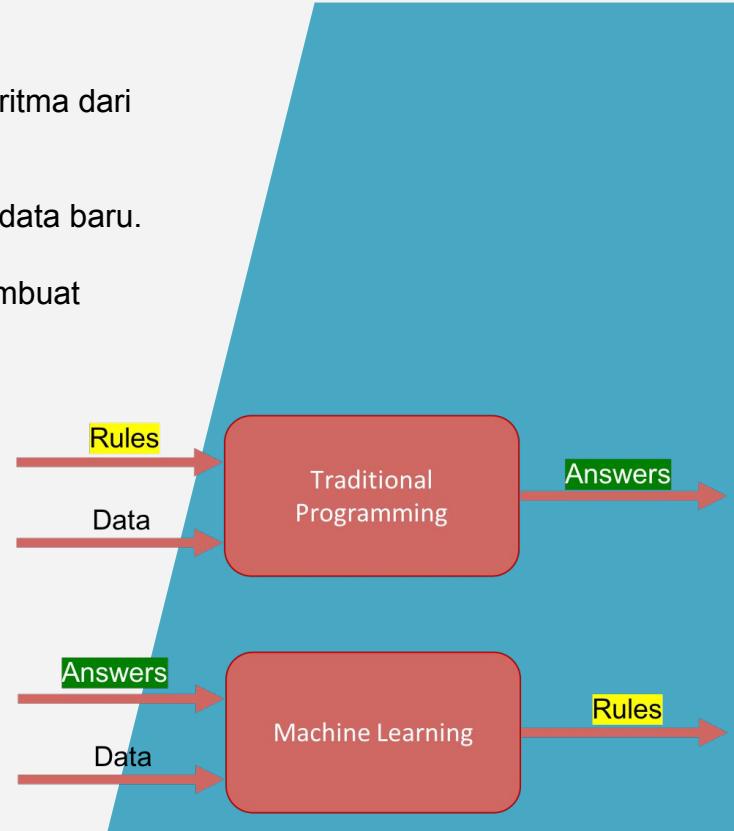
1. **Pengumpulan Data:** Kumpulkan ribuan data transaksi: [Pelanggan A, Beli Kopi Latte & Donat], [Pelanggan B, Beli Cappuccino & Croissant], dan seterusnya.
2. **Pelatihan Model:** Berikan data ini ke algoritma Machine Learning (misalnya, **Association Rule Mining**).
3. **Hasil Model:** Model akan menemukan pola seperti:
  - Kopi Latte -> Donat
  - Cappuccino -> Croissant
  - Teh Matcha -> Muffin

# Tipe Machine Learning

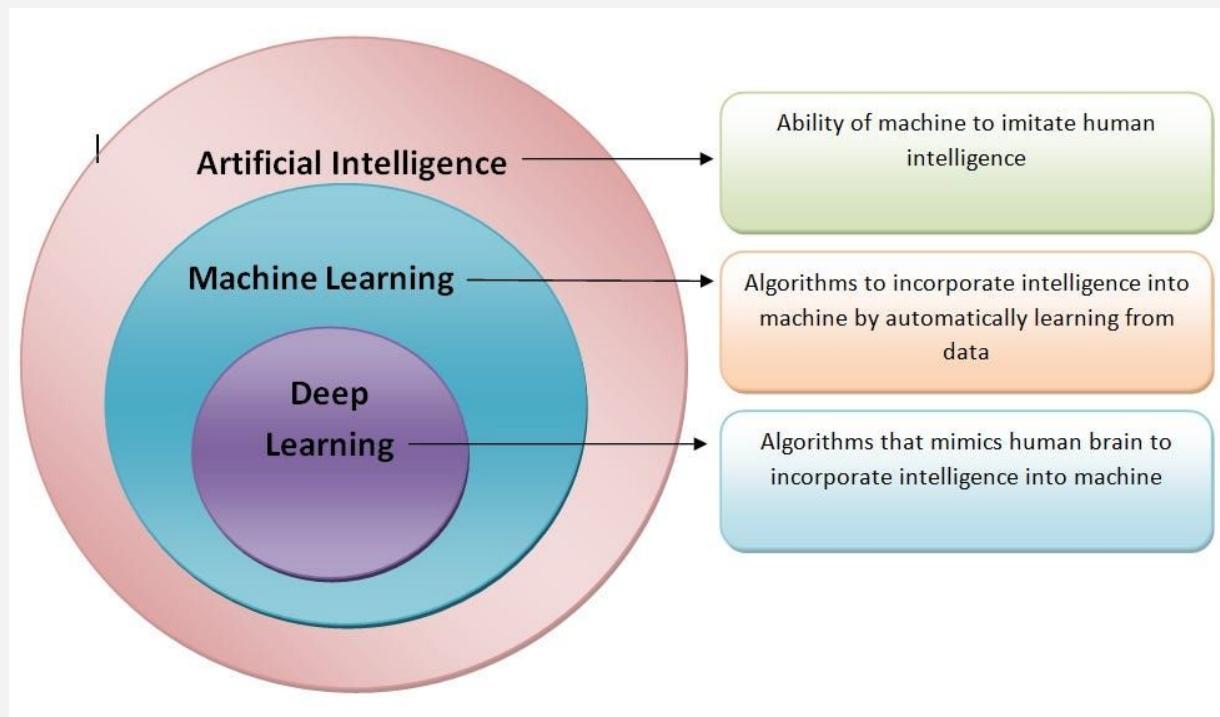


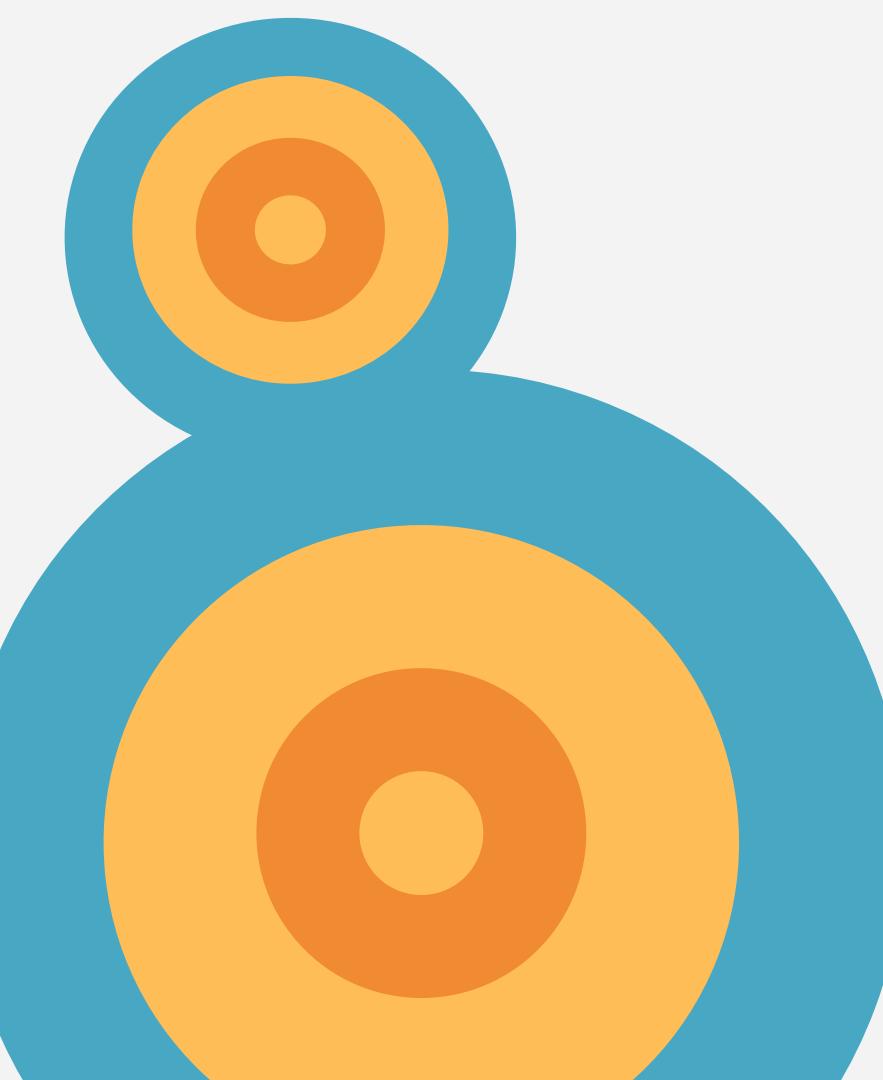
# Machine Learning vs. Traditional Programming

	Traditional Programming	Machine Learning
Logika	Aturan eksplisit ditulis oleh manusia.	Aturan ditemukan oleh algoritma dari data.
Adaptasi	Memerlukan perubahan kode manual.	Dapat dilatih ulang dengan data baru.
Tujuan	Menyelesaikan masalah terdefinisi jelas.	Mengidentifikasi pola & membuat prediksi.
Performa	Terbatas pada aturan yang sudah ditentukan.	Meningkat seiring dengan bertambahnya data.



# Posisi Machine Learning





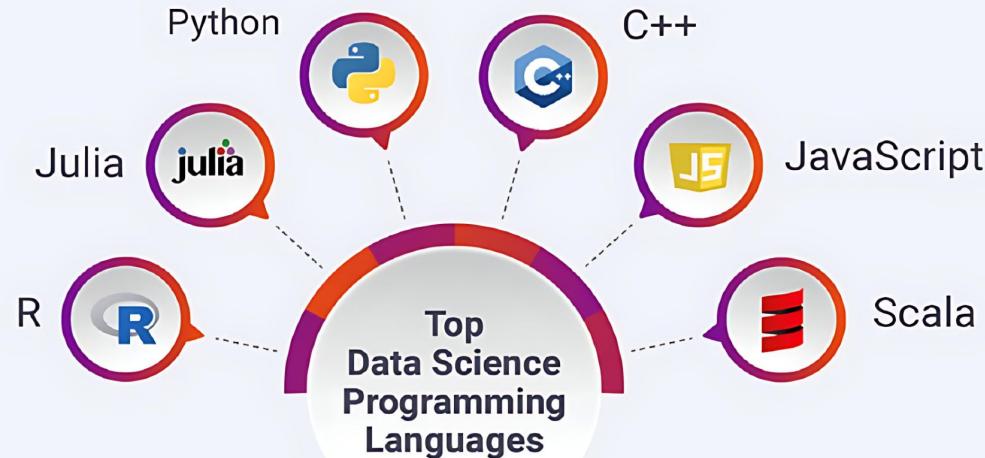
Pertanyaan?



**It's Kahoot Time..**

# Mengapa Python?

# Bahasa Pemrograman



“Kenapa Python?”

# Kenapa Python?

Python umum di dunia teknologi karena tiga pilar utama:

## 1. Kemudahan dan Kecepatan Pengembangan

Sintaks Python yang mudah dibaca dan ringkas mengurangi waktu yang diperlukan untuk menulis, membaca, dan memelihara kode.

## 1. Dukungan Komunitas dan Fleksibilitas

Python memiliki komunitas yang sangat aktif dan besar, yang berarti banyak sumber daya dan dukungan yang tersedia.

Fleksibilitasnya memungkinkan Python berintegrasi dengan bahasa lain (seperti C++ dan Java) untuk performa tinggi, dan yang paling penting.

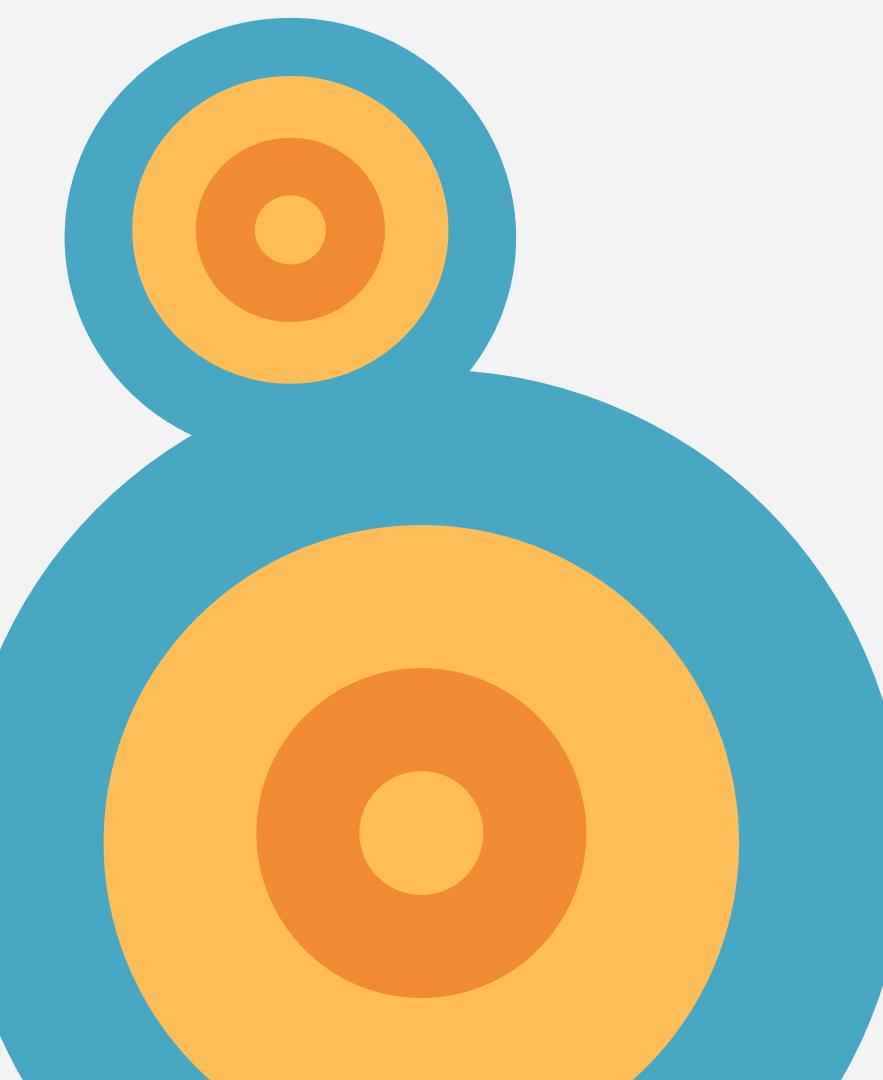
## 1. Ekosistem Library yang Luas

Ini adalah kekuatan terbesar Python. Untuk setiap masalah di bidang data, hampir pasti ada *library* Python yang bisa menyelesaiakannya.

- **Untuk AI & ML:** Ada **Scikit-learn** (untuk model ML klasik), **TensorFlow** dan **PyTorch** (untuk *deep learning* dan jaringan saraf tiruan / Neural Network), dan **NLTK** (untuk pemrosesan bahasa alami / NLP).
- **Untuk Data Analisis:** Ada **Pandas** (untuk manipulasi data) dan **NumPy** (untuk komputasi numerik).
- **Untuk Visualisasi:** Ada **Matplotlib** dan **Seaborn** (untuk membuat grafik yang informatif).

# Kenapa Python?





Pertanyaan?



**It's Kahoot Time..**

# Menjalankan Python



# Menjalankan Python

CLI

```
$ python
Python 3.x.x (default, Date)
[GCC x.x.x] on linux
Type "help", "copyright", "credits" or "license" for more information
>>> print("Hello, World!")
Hello, World!
>>> x = 5
>>> y = 10
>>> x + y
15
>>> exit()
$
```

Python Script

```
datatypes.py x
G: > Python > datatypes.py > ...
1  tup = ("Pi", "My", "Life", "Up")
2  sep = " "
3
4  print(sep.join(tup))
5
6

TERMINAL    PROBLEMS    OUTPUT    DEBUG CONSOLE    JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! http://go.microsoft.com/fwlink/?LinkID=136741&clcid=0x409

PS G:\Python> python datatypes.py
Pi My Life Up
PS G:\Python>
```

Notebook

jupyter tutorial Last Checkpoint: 3 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [1]: `import matplotlib.pyplot as plt  
import pandas as pd  
pd._version_`

Out[1]: '0.24.1'

**PyCon 2018: Using pandas for Better (and Worse) Data Science**

GitHub: <https://github.com/justmarkham/pycon-2018-tutorial>

**Dataset: Stanford Open Policing Project** ([video](#))

In [2]: `# ri stands for Rhode Island  
ri = pd.read_csv('police.csv')`

In [3]: `# what does each row represent?  
ri.head()`

Out[3]:

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2005-02-20	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	

# Notebook



Jupyter

The screenshot shows the Jupyter Notebook interface. At the top, there's a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a Logout button. Below the menu is a toolbar with icons for file operations like New, Open, Save, Run, and Cell Type. The main area has a title "PyCon 2018: Using pandas for Better (and Worse) Data Science" and a GitHub link. It contains two code cells:

```
In [1]: import matplotlib.pyplot as plt  
import pandas as pd  
pd.__version__  
Out[1]: '0.24.1'
```

Below the cells is a section titled "Dataset: Stanford Open Policing Project (video)" with a link. There are two more code cells:

```
In [2]: # ri stands for Rhode Island  
ri = pd.read_csv('police.csv')  
In [3]: # what does each row represent?  
ri.head()  
Out[3]:
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	
3	2005-02-20	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	

Google Colab

The screenshot shows the Google Colab interface. At the top, there's a browser header with the URL "colab.research.google.com/drive/17dhJ\_yv...". Below it is a toolbar with back, forward, search, and refresh buttons. The main area has a title "terminal.ipynb" and a "Comment" button. It contains two code cells:

```
[2] from datetime import datetime  
datetime.now().strftime("%Y-%m-%d %H:%M:%S")  
Out[2]: '2025-08-04 02:53:54'  
[3] from google.colab.output import serve_kernel_port_as_window  
serve_kernel_port_as_window(8000, anchor_text = "Open a terminal")  
Open a terminal
```

VSCode

The screenshot shows the VSCode interface. At the top, there's a browser header with the URL "confluent". Below it is a toolbar with back, forward, search, and refresh buttons. The main area has a title "test4.ipynb" and a "Comment" button. It contains two code cells:

```
[1] from datetime import datetime  
datetime.now().strftime("%Y-%m-%d %H:%M:%S")  
Out[1]: '2025-08-04 02:53:54'  
[2] from google.colab.output import serve_kernel_port_as_window  
serve_kernel_port_as_window(8000, anchor_text = "Open a terminal")  
Open a terminal
```

# Notebook

## Jupyter

1. Download  
<https://www.anaconda.com/download/success>
2. Install
3. Buka “Anacoda Navigator”
4. Pilih “Notebook” / “Jupyter Lab”

## Google Colab

1. Buka  
<https://colab.research.google.com/>
2. Pilih “New Notebook”

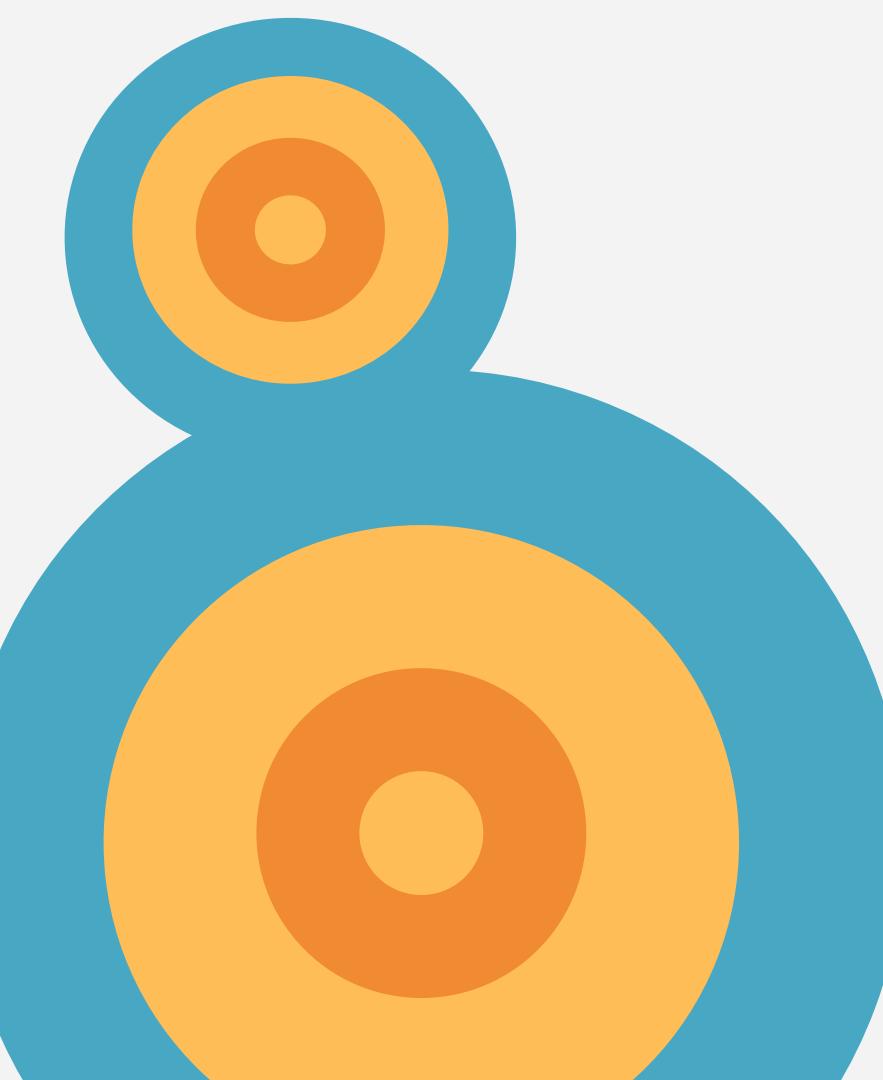
## VSCode

1. Download  
<https://www.anaconda.com/download/success>
2. Install
3. Pastikan bisa menjalankan “python” di terminal / cmd
4. Buka “Visual Studio Code”
5. Buka menu “Extention”
6. Install Extention “Python” & “Jupyter”
7. Buat file baru dengan filename “**nama\_file.ipynb**”

# Mencoba Python

```
print("Hi Jamkrindo!")
```

```
print(1 + 1)
```



Pertanyaan?

# Syntax, Indentation, dan Print Statements



# Syntax

# Konsep

**Syntax** adalah **aturan** yang menentukan **bagaimana** menulis kode agar valid.

Jika melanggar aturan ini, Python akan menampilkan *syntax error*.

Contoh aturan sederhana:

- gunakan tanda kutip (" atau "") untuk teks
- tanda kurung () saat memanggil fungsi seperti print()
- dll

# Praktik

Benar ✓

```
print("Ini adalah teks yang benar")
print('Teks ini juga benar')
```

Salah ✗

```
print("Teks ini salah") # Kutip pembuka & penutup berbeda
print(Teks ini juga salah) # Teks tidak diapit kutip
```

# Indentation

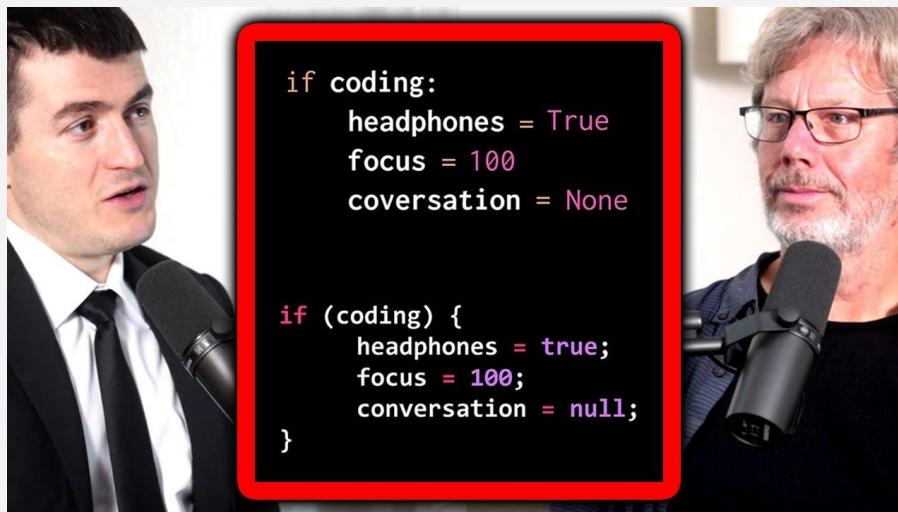
# Konsep

Indentation adalah spasi atau tab di awal baris kode.

Python menggunakan untuk mendefinisikan blok kode, seperti dalam if statement.

Umumnya, satu tingkat indentasi adalah empat spasi. Jika Anda salah melakukannya,

Python akan memunculkan IndentationError.



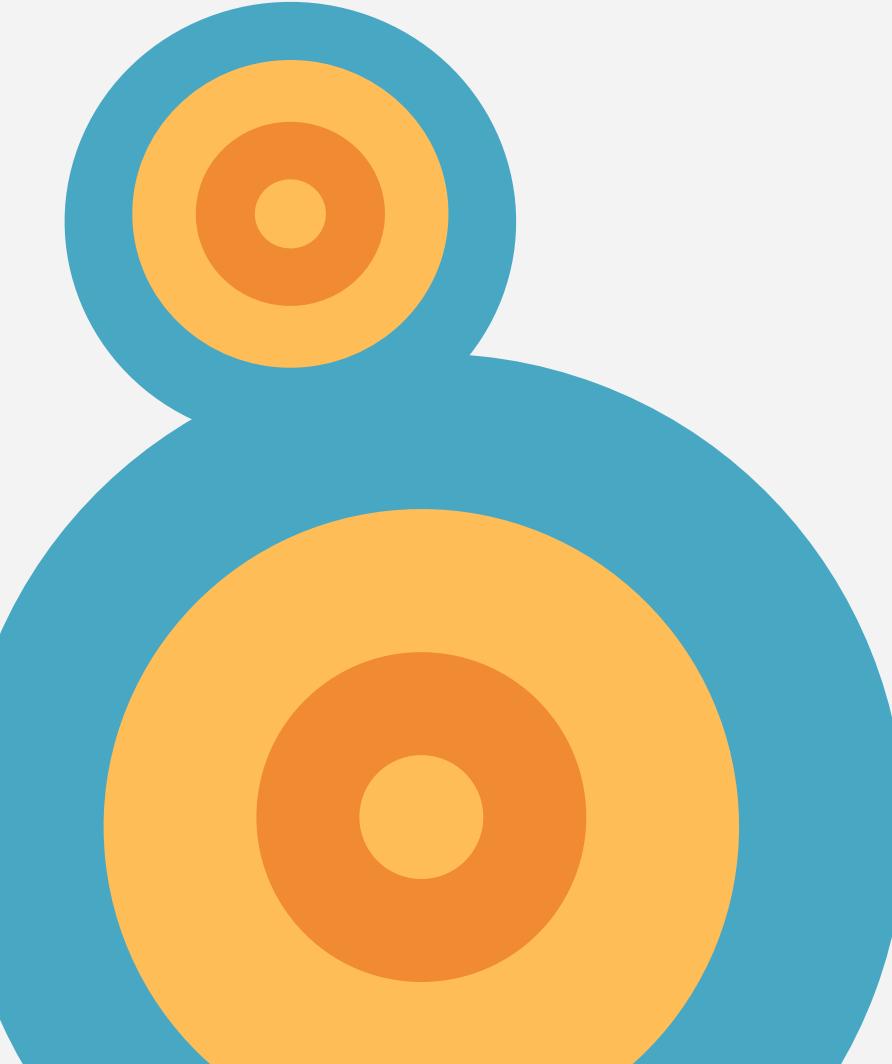
# Praktik

Benar 

```
if 5 > 2:  
    print("5 lebih besar dari 2")  
    print("Indentasi di sini sangat penting")
```

Salah 

```
if 5 > 2:  
    print("Baris ini salah karena tidak ada indentasi")
```



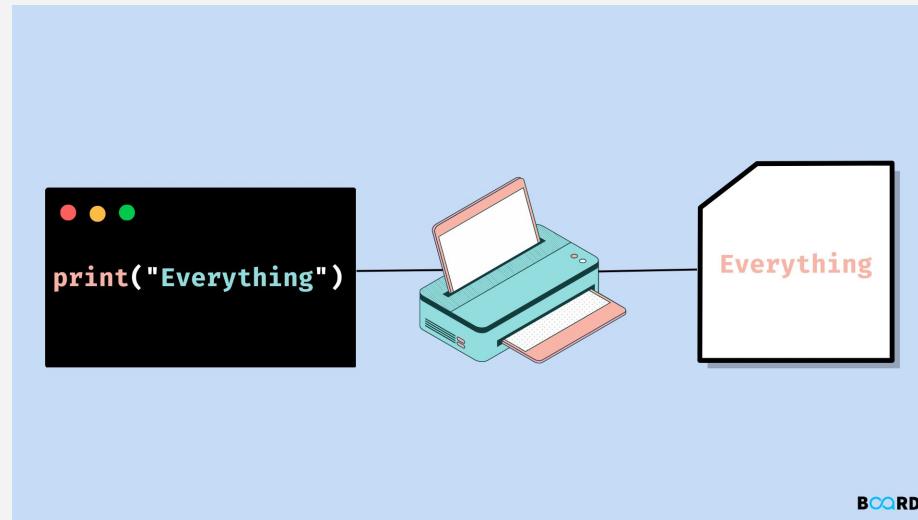
# Pertanyaan?



Print

# Konsep

Fungsi `print()` adalah perintah dasar untuk menampilkan output ke layar. Anda bisa menampilkan berbagai jenis data dan menggabungkannya.



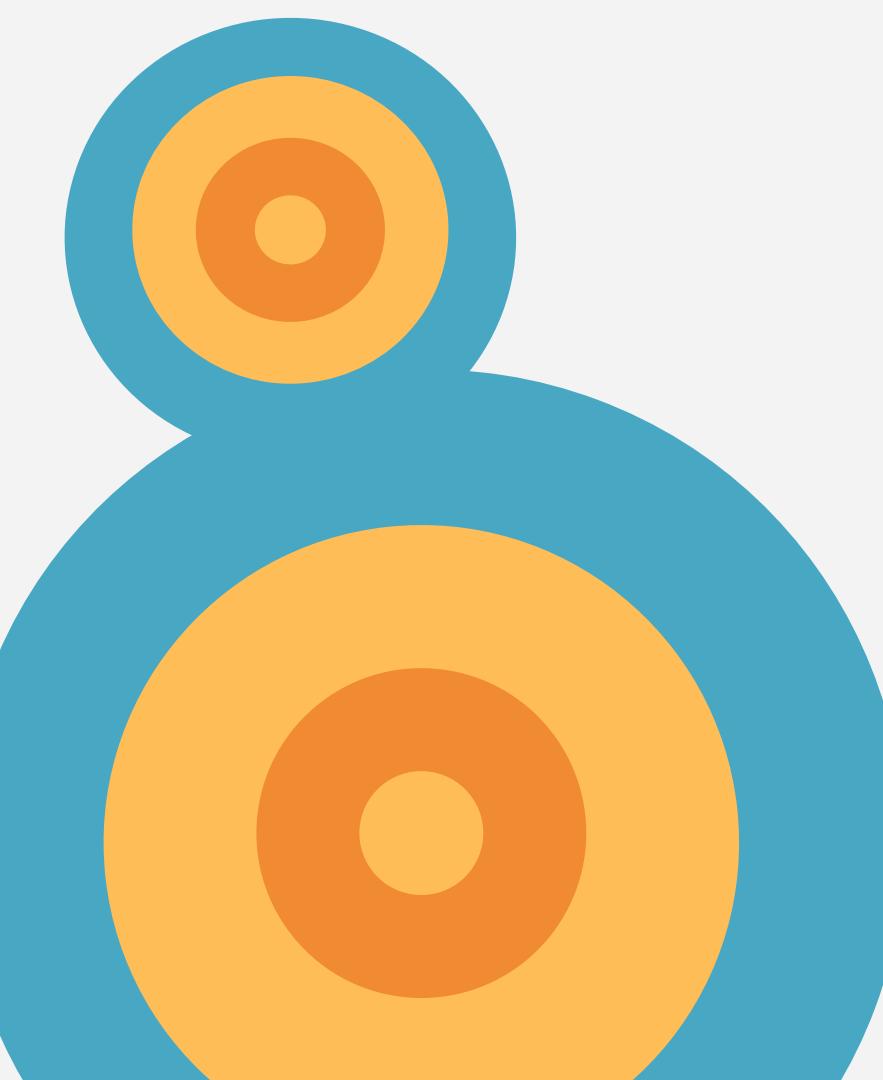
# Praktik

```
print("Selamat datang di Coffee Shop!")  
print('Analisis data hari ini:')
```

```
print(1500000)  
print(1_500_000)
```

```
print("Jumlah pesanan hari ini:", 150)  
print("Total pendapatan:", 1_500_000, "Rupiah")
```

```
# Contoh sep  
print("Kopi", "Latte", "Donat", sep="-")  
  
# Contoh end  
print("Laporan pendapatan", end=" ")  
print("sudah selesai.")
```



Pertanyaan?

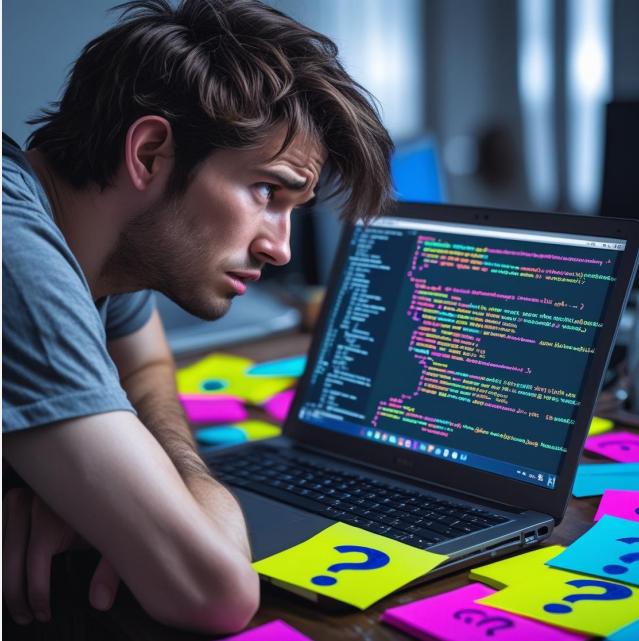


**It's Kahoot Time..**

# Komentar dan Keterbacaan Kode



# Masalah



Bagaimana kita bisa membuat kode kita mudah dipahami  
oleh diri sendiri dan orang lain?

# Solusi

## Komentar

**Apa pun yang ditulis** setelah simbol komentar ini dalam satu baris **akan diabaikan** oleh interpreter Python. Komentar tidak memengaruhi cara kerja program Anda.

Gaya penulisan komentar:

### 1. Komentar Sebaris (#)

Digunakan untuk penjelasan singkat di samping baris kode.

### 1. Komentar Blok (#)

Digunakan untuk menjelaskan satu blok kode. Biasanya ditempatkan di atas blok kode tersebut.

### 1. Docstring (""" """")

Ini adalah jenis khusus dari komentar blok yang digunakan untuk mendokumentasikan blok kode yang lebih besar.

# Praktik

```
# Ini adalah komentar satu baris yang menjelaskan kode di bawahnya.  
print("Halo, selamat datang!")
```

```
print("Rp.", 200_000) # Ini adalah komentar dalam satu baris
```

```
"""  
Ini adalah contoh komentar blok.  
Komentar ini bisa digunakan untuk  
penjelasan yang lebih panjang.  
"""  
print("Analisis data sedang dimulai...")
```

# Praktik

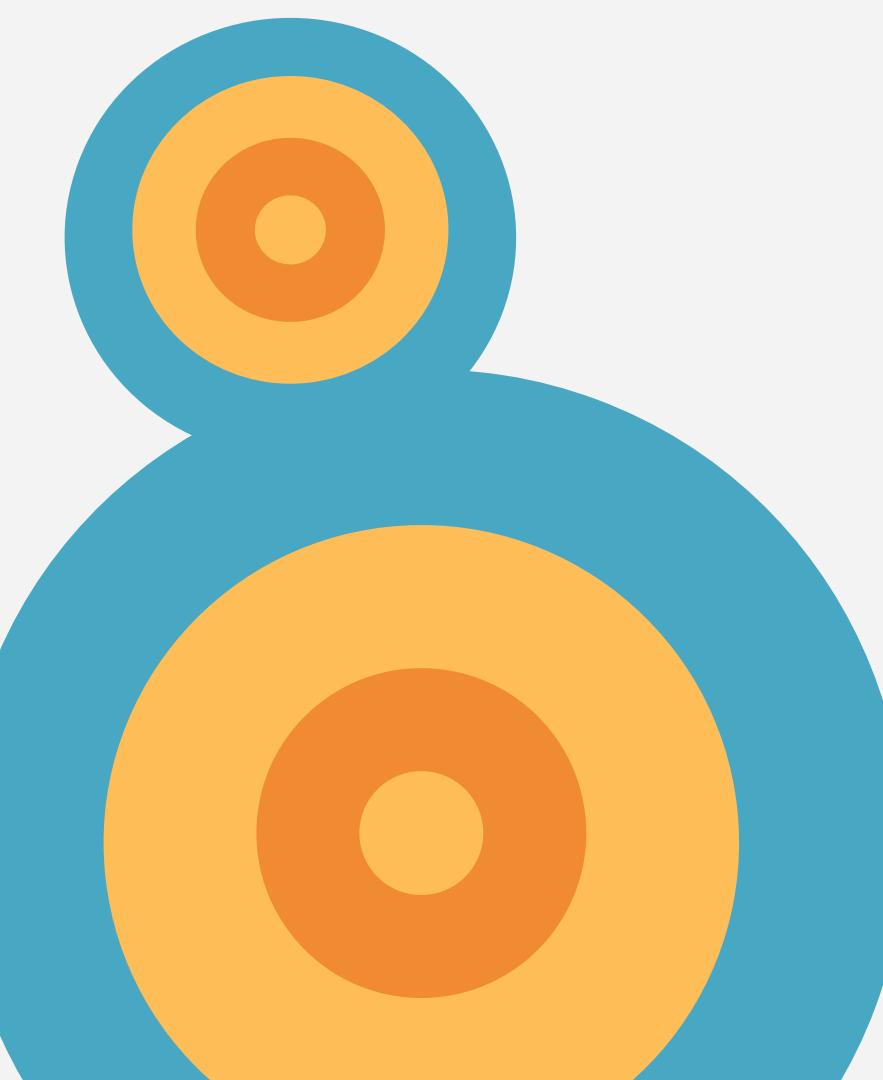
```
# =====
# Laporan Harian Sederhana
# =====

# Mencetak informasi pendapatan harian
print("Total pendapatan harian:")
print(1500000) # Angka ini adalah total pendapatan hari ini

# Mencetak informasi jumlah pesanan
print("Jumlah pesanan:")
print(150) # Angka ini adalah total pesanan yang masuk

"""
=====
Catatan Tambahan:
Laporan ini bisa digunakan untuk melihat performa harian
dan membuat keputusan promosi di masa depan.
=====
"""

print("Laporan selesai.")
```



Pertanyaan?



**It's Kahoot Time..**

# Exercise

## 1. Tugas Essay

**Petunjuk:** Jawablah pertanyaan-pertanyaan berikut dengan jelas dan ringkas. Fokus pada pemahaman konsep dan cara Anda berpikir.

- Studi Kasus Coffee Shop dan 5V Big Data:** Bayangkan Anda adalah manajer operasional di sebuah *coffee shop*. Jelaskan bagaimana Anda bisa menerapkan setiap dari **5V Big Data (Volume, Velocity, Variety, Veracity, Value)** untuk mengoptimalkan operasional dan meningkatkan keuntungan. Berikan contoh spesifik untuk setiap V.
- Machine Learning vs. Pemikiran Manusia:** Jelaskan dengan kata-kata Anda sendiri, mengapa Machine Learning dapat membantu pemilik *coffee shop* membuat keputusan yang lebih baik dibandingkan dengan hanya mengandalkan intuisi atau pengalaman pribadi? Berikan satu contoh masalah yang sulit dipecahkan dengan intuisi tetapi mudah dipecahkan oleh Machine Learning.
- Peran Python dan Keterbacaan Kode:** Mengapa Python menjadi "lingua franca" untuk Data Science? Berikan alasan-alasan yang kuat. Lalu, mengapa penting untuk menulis kode yang mudah dibaca (*readable code*)? Jelaskan bagaimana komentar dan gaya penulisan yang baik dapat membantu dalam proyek analisis data di *coffee shop*.

# Exercise

## 2. Tugas Praktik

**Petunjuk:** Gunakan Notebook untuk menyelesaikan semua tugas di bawah ini. Pastikan untuk menambahkan komentar (#) di setiap langkah untuk menjelaskan apa yang Anda lakukan.

### 1. Mencetak Informasi Sederhana:

- Gunakan perintah print() untuk mencetak pesan "Selamat Datang di Analisis Data Coffee Shop!".
- Cetak nama produk favorit Anda dari *coffee shop*.
- Cetak satu baris angka yang menunjukkan total penjualan harian (misal: 1500000).

### 2. Mencetak dengan Gaya yang Berbeda:

- Gunakan perintah print() untuk mencetak "Kopi", "Donat", dan "Croissant" dalam satu baris, dipisahkan oleh tanda koma. (Hint: Gunakan parameter sep).
- Cetak "Laporan penjualan" dan "sudah selesai" di baris yang sama. (Hint: Gunakan parameter end).

# Exercise

## 2. Tugas Praktik

### 3. Membuat Struktur Kode dengan Indentasi:

- Tulis sebuah if statement dengan kondisi `5000 > 1000`.
- Di dalamnya, gunakan indentasi yang benar (4 spasi) dan cetak dua pesan berbeda.
- Tambahkan komentar di setiap baris `print()` untuk menjelaskan outputnya.

### 4. Membuat Kesalahan (Belajar dari Error):

- Tulis sebuah `print()` statement yang sengaja Anda buat salah (misal: hilangkan tanda kutip atau kurung penutup).
- Jalankan kode tersebut dan salin pesan *error* yang muncul.
- Tambahkan komentar yang menjelaskan mengapa *error* tersebut terjadi.

### 5. Menggunakan Komentar & Keterbacaan:

- Tulis kode untuk mencetak laporan sederhana (misal: nama produk dan harganya).
- Di atas kode tersebut, gunakan komentar blok (#) untuk memberikan judul laporan.
- Tambahkan komentar sebaris di setiap baris cetakan untuk menjelaskan apa yang dicetak.
- Pastikan kode Anda memiliki spasi yang rapi dan mudah dibaca.