

WEBINAR SERIES

# Mindset Shift: Dari SQL ke Python

Upgrade Skill Data Analyst & DBA: Dari Query ke Processing

# Paradigma: SQL vs Python

## SQL (Declarative)

"Apa datanya"

- Fokus ke table, join, agregasi
- Sangat cepat untuk operasi set-based
- Struktur kaku namun teroptimasi

## Python (Imperative)

"Bagaimana prosesnya"

- Bisa looping, kondisi kompleks
- Gabungkan banyak sumber data (API, PDF)
- Ekstensi kemampuan DBA, bukan pengganti



# Pekerjaan Sulit di SQL, Mudah di Python



## Text Processing

Pembersihan nama kompleks dengan Regex lanjutan & Fuzzy matching (Levenshtein).



## File Parsing

Membaca data tidak terstruktur dari PDF, Excel kotor, dan CSV tanpa skema.



## Integrasi API

Koneksi ke Dukcapil, OSS, atau internal service secara otomatis.



# Analogi Kerja

## SQL

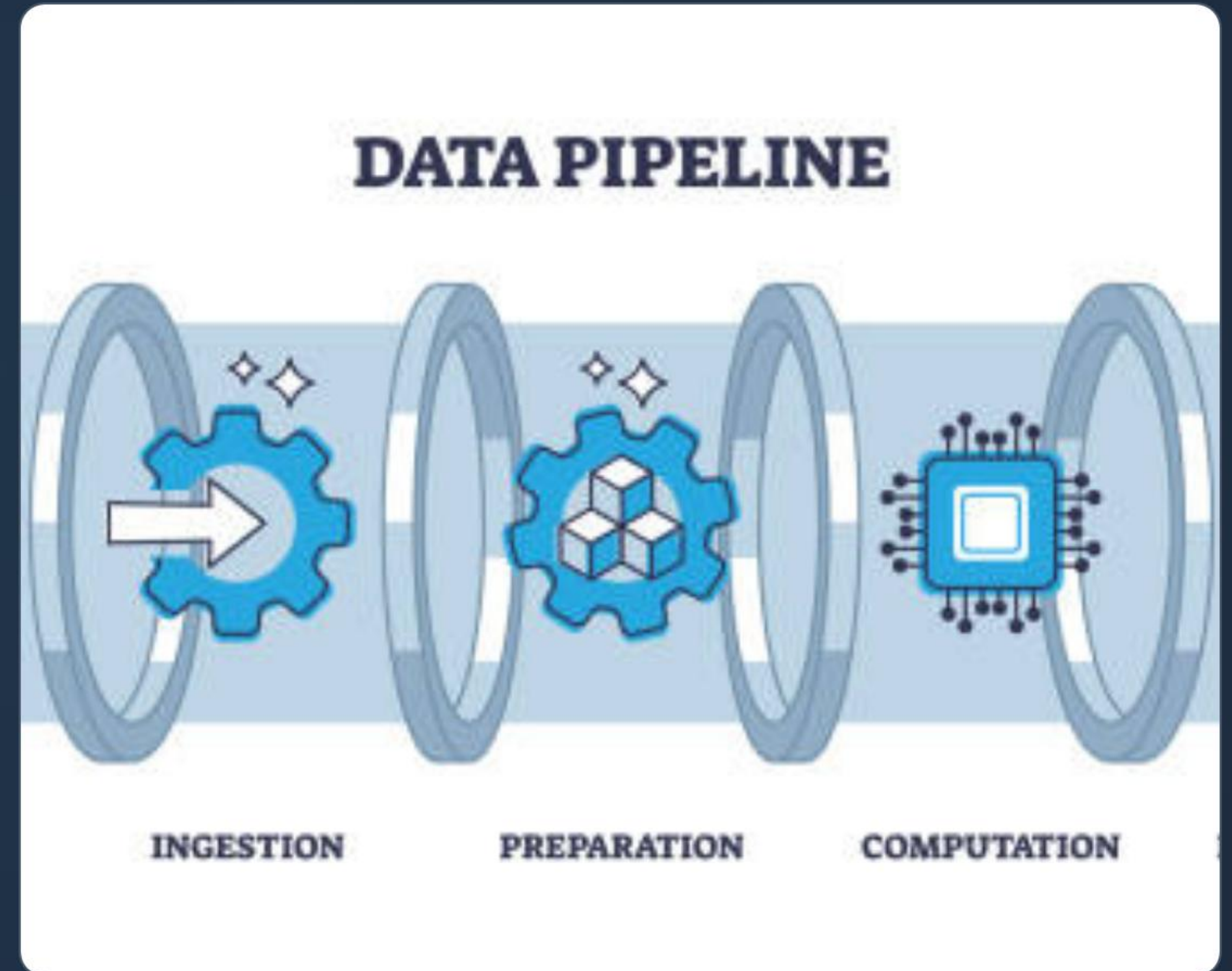
Seperti mesin pabrik yang sangat cepat dan presisi untuk mengolah bahan baku terstruktur.

## Python

Seperti "toolbox" raksasa. Bisa dipakai memperbaiki mesin, menyambung pipa, hingga mengecat dinding.

## Kombinasi

Data Pipeline Superpower.





# Mapping Konsep: SQL ke Python

Operasi	SQL	Python (Pandas)
Select	SELECT kolom	<code>df["kolom"]</code>
Filter	WHERE kondisi	<code>df[df.kolom == ...]</code>
Agregasi	GROUP BY	<code>df.groupby()</code>
Join	JOIN	<code>df.merge()</code>
Logic	CASE WHEN	<code>np.where()</code> / <code>apply()</code>
Unik	DISTINCT	<code>df.drop_duplicates()</code>



# Mindset Baru: Goal Oriented

---

**01**

**SQL**

Query the data

**02**

**Python**

Process the data

**03**

**Streamlit**

Present the data

*"Bukan jago sintaks, tapi jago menyelesaikan masalah."*



# Python 101 untuk Analyst

---

## Konsep Dasar

- **Variabel:** Tempat menyimpan nilai.
- **List/Dict:** Struktur data (mirip array/JSON).
- **Loop/Kondisi:** Logika pemrograman (for, if).
- **Function:** Rumus reusable.
- **Module:** Library tambahan.

```
# Contoh Sederhana data = { "nama": "Budi", "usia":  
21, "role": "Analyst" } if data["usia"] > 20:  
print(f"Senior: {data['nama']}")
```



# Package Wajib untuk DBA

---

**pandas**

Data Table & Manipulation

**sqlalchemy**

Koneksi Database (Postgres)

**openpyxl**

Baca/Tulis Excel

**Levenshtein**

Fuzzy Matching Nama

**streamlit**

User Interface Cepat



# Koneksi Python ke PostgreSQL

Menggunakan library modern **SQLAlchemy**.

Tidak perlu menulis kode koneksi berulang-ulang. Sekali setup engine, bisa dipakai untuk query apapun.



```
from sqlalchemy import create_engine import pandas
as pd # Setup koneksi engine = create_engine(
"postgresql://user:pass@localhost:5432/db" ) # Baca
langsung ke DataFrame df = pd.read_sql( "SELECT *
FROM dikti_mahasiswa", engine )
```



# Membaca Data dari File

---

```
import pandas as pd # Membaca Excel df_excel =  
pd.read_excel("data_mahasiswa.xlsx") # Membaca CSV  
df_csv = pd.read_csv("data_log.csv") # Preview data  
print(df_excel.head())
```

## Freedom of Format

Analyst tidak lagi bergantung pada tabel database yang kaku. Anda bisa menggabungkan data Excel dari HRD dengan data Transaksi dari Database dalam satu script.



# DataFrame Operation = SQL Operation

---

- Filter Data
- Group By & Aggregation
- Join / Merge Tables
- Fill Missing Values (FillNA)
- Sorting

```
# Contoh JOIN (Left Join) df_merge = df1.merge(  
df2, on="nik", how="left" ) # Contoh Group By  
df_summary = df_merge.groupby( "provinsi" )  
["jumlah"].sum()
```



# Power of Regex & Normalisasi

---

```
import re df["nama_bersih"] = ( df.nama
.str.upper() # Hapus simbol aneh .str.replace(r"
[^A-Z ]", "", regex=True) # Hapus spasi ganda
.str.replace(r"\s+", " ", regex=True) .str.strip()
)
```

## Manfaat Utama

Membersihkan data "kotor" yang sangat sulit dilakukan dengan SQL standar.

Contoh Input:

" budi Prakoso, S.Kom. "

Output:

"BUDI PRAKOSO SKOM"



# OOP Sekilas untuk DBA

## Class & Object

**Class:** Mirip definisi tabel (schema) + behavior (fungsi).

**Object:** Instance data (baris data).

Berguna untuk membuat "Reusable ETL Pipeline". Anda bisa membuat Class DataCleaner yang bisa dipakai di banyak proyek berbeda.





# Python untuk Pemadanan Data

---

## Fungsi Cleaning

Kita bisa membungkus logika pembersihan dalam satu fungsi sederhana yang bisa diaplikasikan ke jutaan baris data sekaligus.

```
def clean_nama(s): if not isinstance(s, str):  
    return "" # Hapus non-alpha, upper, trim spasi  
    return re.sub( r"\s+", " ", re.sub(r"^[A-Za-z ]",  
    "", s.upper())) ).strip()
```



# Fuzzy Matching

---

```
from fuzzywuzzy import fuzz a = "Aisyah Dwi" b = "A  
i s y a h D w i" # Simple Ratio score =  
fuzz.token_sort_ratio(a, b) print(score) # Output:  
100 (Match Sempurna)
```

## Solusi Typo Manusia

SQL hanya bisa = atau LIKE. Python bisa mengukur "jarak" kemiripan.

Sangat berguna untuk mencocokkan data input manual dengan database referensi (Dukcapil/Dikti).



# Pipeline Pemadanan: Dikti vs Dukcapil

---

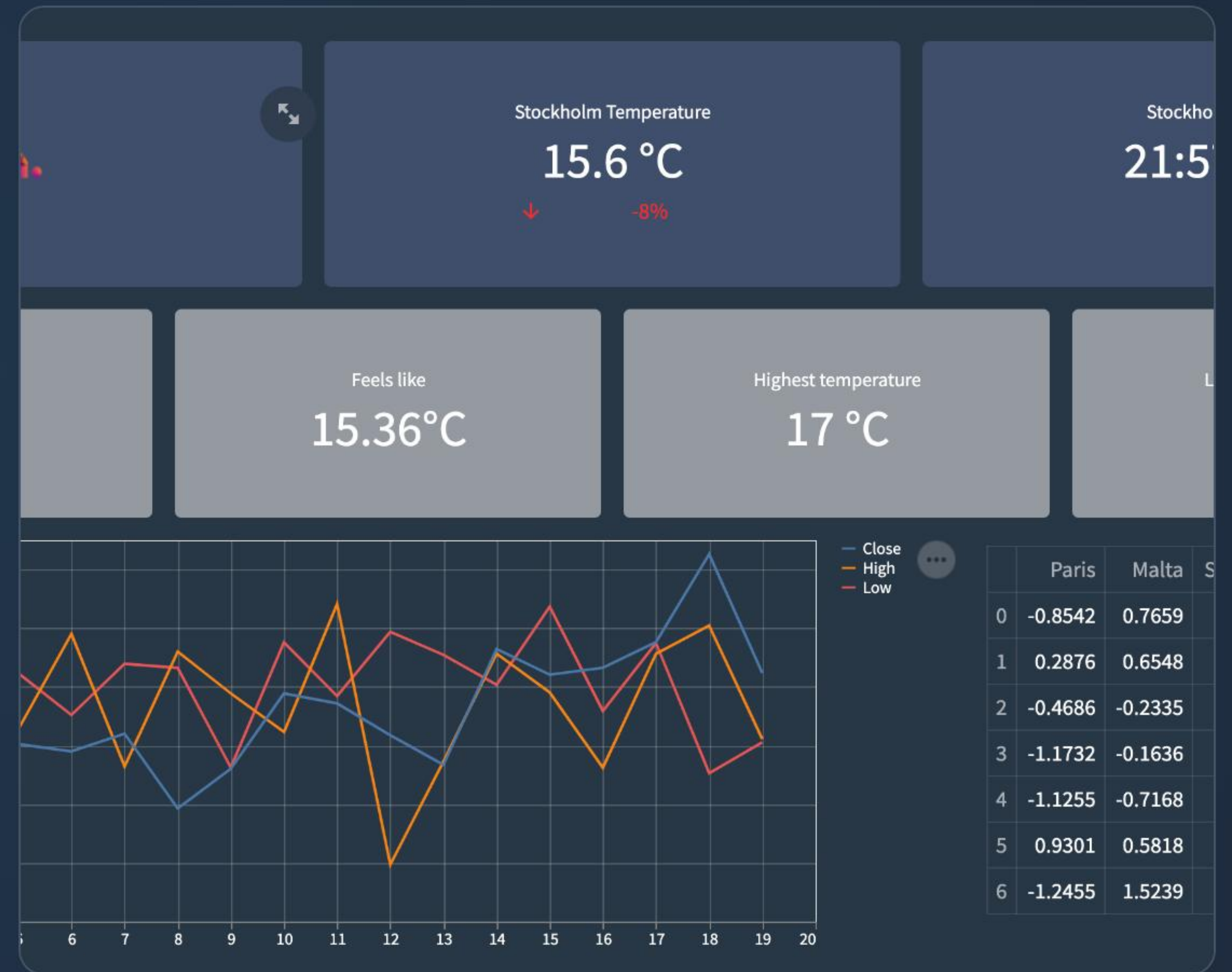




# Apa itu Streamlit?

Framework Python untuk membuat **Aplikasi Web Data** dalam hitungan menit.

- Tidak perlu HTML/CSS/JS.
- Murni Python.
- Cocok untuk dashboard validasi data.
- Tools monitoring ETL.





# Streamlit: Hello World

---

```
import streamlit as st
st.title("Pemeriksaan Data Dikti")
nama = st.text_input("Nama Mahasiswa")
if nama:
    st.write("Anda memasukkan:", nama)
```

## Instant UI

Hanya dengan beberapa baris kode, Anda sudah memiliki input form dan display text yang interaktif.



# Fitur Upload Excel

---

## Interaktifitas Data

Memungkinkan user non-teknis (tim bisnis/admin) untuk mengupload data mereka sendiri dan melihat hasil pemrosesan Python secara langsung.

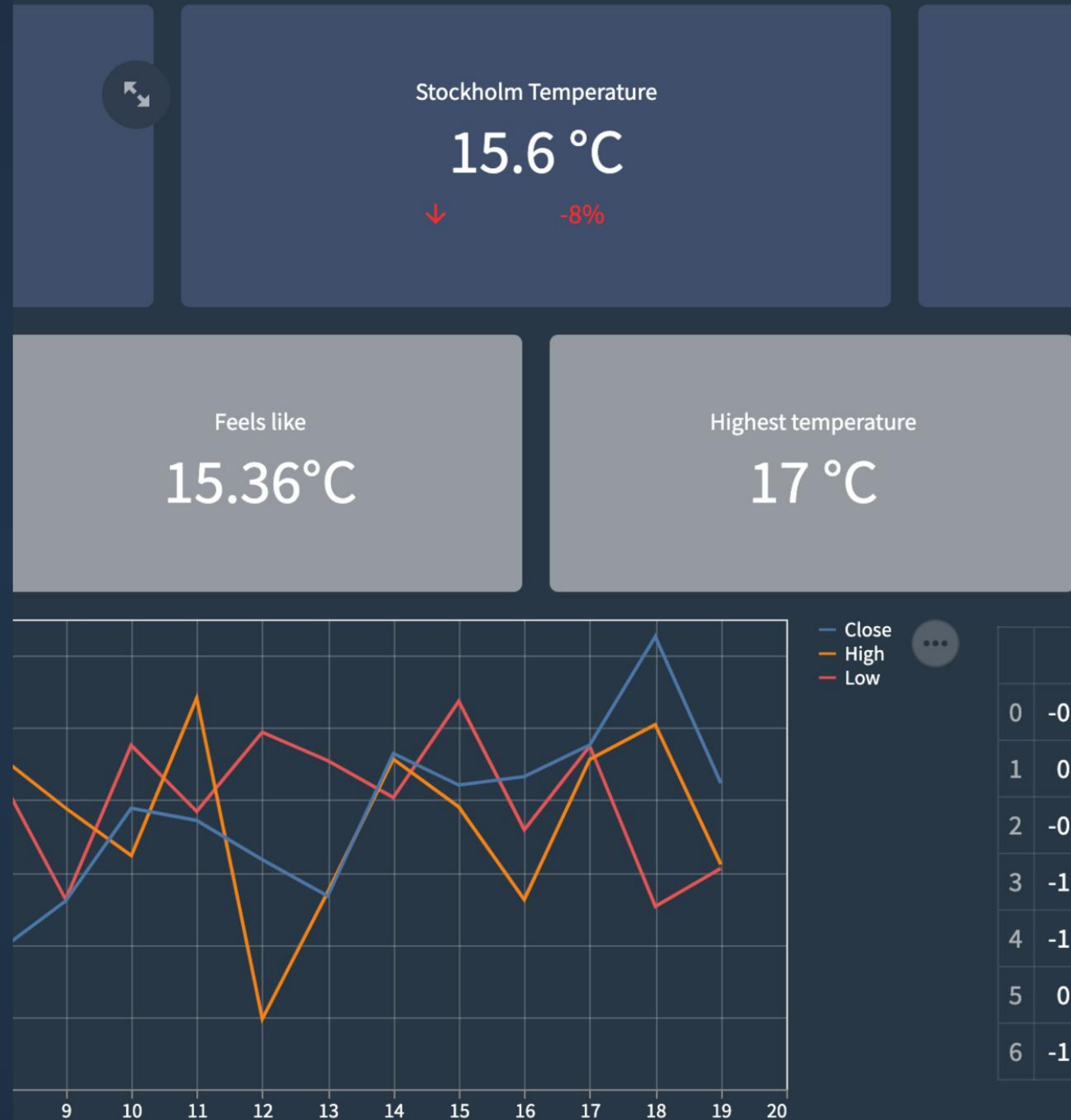
```
file = st.file_uploader("Upload Excel") if file: #  
    Baca file yg diupload user df = pd.read_excel(file)  
    # Tampilkan tabel interaktif st.dataframe(df)
```



# App Pemadanan Data

- Upload data Excel
- Cleaning otomatis di background
- Fuzzy match by Nama
- Highlight baris yang tidak match
- Export hasil ke Excel baru

**Analyst jadi "Developer Ringan".**





# Deployment Mudah

## Share Hasil Anda

Jangan biarkan script hanya jalan di laptop Anda. Deploy agar bisa dipakai satu kantor.

- Streamlit Cloud (Public/Private)
- Internal Server (Intranet)
- HuggingFace Spaces

```
$ streamlit run app.py
```





# Use Case 1: Dikti ↔ Dukcapil

---

## Masalah

- Nama berbeda format (gelar, singkatan).
- NIK invalid atau kosong.
- Input manual operator banyak typo.

## Solusi Python

- Fuzzy matching otomatis massal.
- Rule-based anomaly detection.
- Export hasil final siap pakai.



# Use Case 2: Cleansing Nama Perusahaan

---

## Tantangan Variasi Teks

Satu entitas bisa ditulis dengan puluhan cara berbeda:

- PT. Maju Jaya
- P.T. Maju Jaya
- Maju Jaya, PT
- Perseroan Terbatas Maju Jaya

## Pendekatan Python

- **Regex:** Normalisasi awalan/akhiran (PT, CV).
- **Tokenization:** Memecah kata kunci.
- **Dictionary:** Klasifikasi berbasis kamus baku.



# Use Case 3: Laporan Otomatis

---



## Manual

Query DB → Export Excel → Copy Paste  
Template → Email manual.



## Python Automation

Script jalan tiap jam 8 pagi → Query →  
Transform → Kirim PDF via Email/Telegram  
Bot.



# Use Case 4 & 5: Integrasi & Anomaly

## ETL Ringan

Ambil data API OSS, gabung data internal, load ke Postgres, lalu tampilkan di dashboard Streamlit. Semua dalam satu flow.

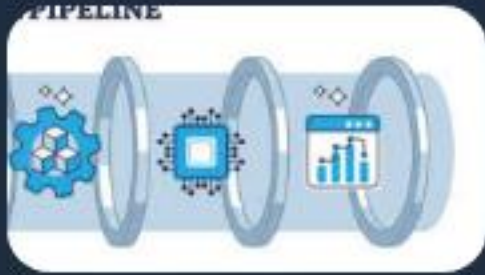
## Anomaly Detection

Deteksi usia tidak logis, nama vs tanggal lahir tidak match, atau inkonsistensi angka pelatihan. Rule dibuat sekali, dipakai selamanya.



# Image Sources

---



<https://media.istockphoto.com/id/2076339830/vector/data-pipeline-with-computing-file-preparation-process-stages-outline-diagram.jpg?s=612x612&w=0&k=20&c=8-9DcEB0LnyR2lqSYg2kwFgC0VxQjs9QdU91srOPzRI=>

Source: [www.istockphoto.com](https://www.istockphoto.com)

---



[https://static.vecteezy.com/system/resources/previews/026/173/370/non\\_2x/cloud-computing-concept-art-abstract-cloud-server-data-center-organize-technology-blue-light-room-ai-generated-photo.jpeg](https://static.vecteezy.com/system/resources/previews/026/173/370/non_2x/cloud-computing-concept-art-abstract-cloud-server-data-center-organize-technology-blue-light-room-ai-generated-photo.jpeg)

Source: [www.vecteezy.com](https://www.vecteezy.com)

---



<https://media.geeksforgeeks.org/wp-content/uploads/20230831083134/SQL-Vs-Python.png>

Source: [www.geeksforgeeks.org](https://www.geeksforgeeks.org)

---



[https://miro.medium.com/1\\*hbnJvvvd5jf4Acaod3swtA.png](https://miro.medium.com/1*hbnJvvvd5jf4Acaod3swtA.png)

Source: [medium.com](https://medium.com)