



# Laporan Visualisasi Data dengan Streamlit



## Identitas

- **Nama:** Micka Mayulia Utama
- **NIM:** 10231053
- **Mata Kuliah:** Adminstrasi Basis Data
- **Semester:** 5



## Step by Step Tugas Visualisasi

1. []
2. [📄 File vs.py]

```
import matplotlib.pyplot as plt
import pandas as pd
import streamlit as st

st.title("Visualisasi Data Interaktif")

# Menambahkan gambar dan teks
st.image("server.jpg", caption="Identifikasi server dalam mendukung analisis efisiensi")
st.markdown("""Dataset ini merepresentasikan performa operasional beberapa server di berbagai wilayah Indonesia, mencakup jumlah data yang diproses, durasi downtime, tingkat efisiensi, serta koordinat lokasi (lintang dan bujur). Visualisasi dibuat untuk membantu mengidentifikasi pola kinerja, membandingkan performa antar lokasi, dan memahami persebaran geografis server dalam mendukung analisis efisiensi serta potensi risiko operasional secara lebih intuitif dan terstruktur.
""")

# Dataset
data = pd.DataFrame({
    "Server_Location": ["Jakarta Data Hub", "Bandung AI Center", "Surabaya Compute",
                        "Bali Cloud Node", "Medan Storage Unit", "Makassar Edge Server",
                        "Yogyakarta AI Lab", "Semarang Processing", "Balikpapan Backup",
                        "Pontianak Mini Server"],
    "Data_Processed_TB": [220, 180, 260, 140, 110, 160, 95, 150, 70, 125],
    "Downtime_Hours": [4, 6, 3, 5, 8, 4, 7, 6, 9, 5],
    "Efficiency_%": [92, 88, 95, 85, 80, 90, 83, 87, 75, 86],
    "lat": [-6.2, -6.9, -7.2, -8.6, 3.5, -5.1, -7.8, -6.9, -1.2, -0.02],
```

```
    "lon": [106.8, 107.6, 112.7, 115.2, 98.6, 119.4, 110.4, 110.4, 116.8, 109.3]
})
```

```
st.subheader("Data")
st.dataframe(data)
```

```
# Dropdown
```

```
tipe = st.selectbox("Pilih jenis grafik:", ["Area", "Bar", "Line", "Map", "Pie"])
```

```
if tipe == "Bar":
```

```
    st.subheader("Kesehatan Server – Downtime vs Efficiency")
```

```
    fig, ax = plt.subplots(figsize=(8,4))
```

```
    ax.bar(data["Server_Location"], data["Downtime_Hours"])
```

```
    ax.set_ylabel("Downtime (hrs)")
```

```
    ax.tick_params(axis='x', rotation=45)
```

```
    st.pyplot(fig)
```

```
    st.markdown("""
```

Visualisasi ini menampilkan **dua indikator utama kesehatan server**:

- **Batang (bar)** menunjukkan jumlah **downtime** di setiap lokasi server.
- **Garis (line)** menunjukkan tingkat **efisiensi** kinerja server.

Temuan awal menunjukkan bahwa server dengan downtime tinggi tidak selalu memiliki efisiensi terendah,

yang berarti ada faktor lain (misalnya beban kerja, peran cadangan, atau infrastruktur lokal)

yang turut memengaruhi performa.

Asumsi umum bahwa **downtime tinggi = server buruk** tidak selalu tepat.

Contohnya, Balikpapan memiliki downtime relatif tinggi namun memang berfungsi sebagai backup node,

sehingga frekuensi aktivitas dan prioritasnya berbeda dengan server utama seperti Surabaya atau Jakarta.

```
    """)
```

```
    st.write(data[["Server_Location", "Downtime_Hours", "Efficiency_%"]])
```

```
elif tipe == "Line":
```

```
    st.subheader("Line Chart – Data Processed per Server")
```

```
    line_data = data.set_index("Server_Location")[["Data_Processed_TB"]]
```

```
    st.line_chart(line_data)
```

```
    st.markdown("""
```

Line chart ini **tidak lagi menciptakan ilusi time-series palsu**.

Urutan server disusun berdasarkan jumlah data yang diproses, sehingga visualisasi

sekarang menunjukkan **pola kenaikan kapasitas** antar server, bukan sekadar perubahan acak.

Interpretasi yang lebih jujur:

- **Surabaya** dan **Jakarta** berada di kapasitas tertinggi → berpotensi

```
menjadi node utama
- Balikpapan dan Yogyakarta terendah → kemungkinan node cadangan atau lokal
    """
    st.write(line_data)
```

```
elif tipe == "Area":
    st.subheader("Pertumbuhan Data (*Cumulative*)")
    df_sorted = data.sort_values("Data_Processed_TB",
ascending=False).reset_index(drop=True)
    df_sorted["Cumulative_TB"] = df_sorted["Data_Processed_TB"].cumsum()
    fig, ax = plt.subplots(figsize=(8,4))
    ax.fill_between(range(len(df_sorted)), df_sorted["Cumulative_TB"],
alpha=0.4)
    ax.plot(range(len(df_sorted)), df_sorted["Cumulative_TB"], marker='o')
    ax.set_xticks(range(len(df_sorted)))
    ax.set_xticklabels(df_sorted["Server_Location"], rotation=45)
    st.pyplot(fig)
    st.markdown("""
```

Area chart menunjukkan akumulasi (\*cumulative\*) data yang diproses oleh setiap server setelah diurutkan berdasarkan nilai terbesar ke terkecil. Visualisasi ini memperlihatkan bagaimana kontribusi tiap server membangun total beban sistem. Grafik ini membantu menjawab pertanyaan penting:

- *seberapa besar ketergantungan sistem terhadap server-server teratas?*
- Jika kurva naik tajam di awal, maka sebagian besar beban ditopang oleh sedikit server utama.

Urutan dari nilai terbesar ke terkecil menciptakan kenaikan yang terlihat halus dan stabil.

Ini dapat memberi kesan bahwa distribusi beban sudah seimbang, padahal bisa jadi terdapat ketimpangan yang signifikan.

Server di posisi puncak berpotensi menjadi *single point of overload* atau bahkan *single point of failure* apabila tidak memiliki backup yang memadai. Ketergantungan ini tidak langsung terlihat hanya dari bentuk kurva yang tampak “rata-rata naik”. Grafik ini berguna untuk mengidentifikasi konsentrasi beban dan potensi risiko sistem, namun harus dibaca bersama data downtime dan efisiensi untuk mendapatkan gambaran ketahanan server yang lebih realistis.

```
""")

st.write(df_sorted[["Server_Location", "Data_Processed_TB", "Cumulative_TB"]])
```

```
elif tipe == "Pie":
    st.subheader("Distribusi Beban Data (Share)")
    fig, ax = plt.subplots()
    ax.pie(data["Data_Processed_TB"], labels=data["Server_Location"],
autopct="%1.1f%%", startangle=90)
    ax.set_title("Share of Data Processed")
    st.pyplot(fig)
    st.markdown("""

*Pie chart* menunjukkan persentase kontribusi masing-masing server terhadap total data yang diproses oleh sistem. Visualisasi ini membantu mengidentifikasi
```

server mana yang menangani beban terbesar. Jika beberapa irisan tampak mendominasi, itu berarti sebagian besar beban kerja hanya ditopang oleh sedikit server utama.

Besar irisan sering diasumsikan sebagai \*indikator\* “kinerja terbaik”. Padahal, nilai yang tinggi belum tentu berarti sistem lebih sehat, bisa juga menandakan server tersebut bekerja mendekati batas \*maksimum\*. \*Pie chart\* menyamarkan apakah server paling \*dominan\* berada dalam kondisi stabil atau justru mendekati \*overload\*. Visual ini juga tidak memperlihatkan faktor penting seperti \*downtime\* tinggi atau efisiensi rendah, sehingga berpotensi menimbulkan rasa aman yang salah.

Pie chart efektif untuk melihat distribusi beban secara umum, namun tidak cukup untuk menilai ketahanan dan stabilitas sistem. Agar analisis lebih \*valid\*, grafik ini sebaiknya dibaca bersamaan dengan:

- grafik \*downtime\* (keandalan)
- grafik \*efisiensi\* (kinerja)
- serta visual perbandingan seperti \*bar\* atau \*line chart\*.

```
"""  
    st.write(data[["Server_Location","Data_Processed_TB"]])  
  
elif tipe == "Map":  
    st.subheader("Peta Persebaran – warna = Efficiency% , ukuran = TB")  
    st.map(data[["lat","lon"]])  
    st.markdown("""
```

Peta menampilkan persebaran fisik lokasi server di berbagai wilayah Indonesia berdasarkan koordinat latitude dan longitude. Visualisasi ini membantu melihat pola distribusi geografis server, apakah tersebar merata atau justru terkonsentrasi di wilayah tertentu. Hal ini penting untuk menilai potensi risiko wilayah (regional risk) serta efisiensi akses data dari berbagai area.

Persebaran yang tampak merata sering diasumsikan berarti sistem lebih aman dan andal. Padahal, kedekatan geografis tidak otomatis menjamin adanya redundansi atau load balancing yang sehat. Jika beberapa server terkonsentrasi di satu region, maka bencana alam, gangguan listrik, atau gangguan jaringan regional dapat melumpuhkan sebagian besar sistem sekaligus. Risiko ini tidak terlihat hanya dari jumlah titik, tetapi dari pola pengelompokannya (\*clustering\*).


Map efektif untuk mengidentifikasi potensi titik lemah geografis dan ketimpangan distribusi lokasi server. Untuk analisis yang lebih valid, peta sebaiknya dikombinasikan dengan data beban kerja, downtime, dan efisiensi performa masing-masing server.

```
"""  
  
st.write(data[["Server_Location","Data_Processed_TB","Downtime_Hours","Efficien  
cy_%","lat","lon"]])
```

## 5. ERD / Relasi Tabel




## 6. Output & Screenshot Dashboard

 1. Screenshot Data Customers




 2. Screenshot Data Orders




 3. Screenshot Data Products




 4. Screenshot Detail Order




 5. Screenshot Visualisasi Jumlah Pelanggan




 6. Screenshot Visualisasi Usia Pelanggan



 7. Screenshot Visualisasi Produk Terlaris 1



 8. Screenshot Visualisasi Produk Terlaris 2

