

Tugas 11: LAPORAN PRAKTIKUM MANDIRI

Syauqi Rabbani - 0110224208 ^{1*}

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

² Sistem Informasi, STT Terpadu Nurul Fikri, Depok

*E-mail: syauqi.rabbani36@gmail.com

1. Load Dataset

```
[35] 0d
▶ import pandas as pd
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
import folium
df = pd.read_csv(path + 'ebola_data_db_format.csv')
print(df.head())

...
   Indicator Country      Date \
0  Cumulative number of confirmed, probable and s...  Guinea  2015-03-10
1  Cumulative number of confirmed Ebola cases  Guinea  2015-03-10
2  Cumulative number of probable Ebola cases  Guinea  2015-03-10
3  Cumulative number of suspected Ebola cases  Guinea  2015-03-10
4  Cumulative number of confirmed, probable and s...  Guinea  2015-03-10

      value
0  3285.0
1  2871.0
2  392.0
3  22.0
4  2170.0
```

Pada bagian ini, program memuat dataset *ebola_data_db_format.csv* menggunakan fungsi `read_csv()` dari library pandas. Data tersebut kemudian ditampilkan menggunakan `df.head()` untuk memastikan bahwa dataset berhasil dibaca dengan benar. Dataset ini berisi informasi negara, tanggal, indikator, dan jumlah kasus Ebola, tetapi belum memiliki koordinat geografis sehingga belum dapat langsung digunakan untuk analisis berbasis lokasi.

2. Menambahkan GPS berdasarkan negara

```
[38] ✓
  country_gps = {
    "Guinea": {"latitude": 9.9456, "longitude": -9.6966},
    "Liberia": {"latitude": 6.4281, "longitude": -9.4295},
    "Sierra Leone": {"latitude": 8.4606, "longitude": -11.7799},
    "Nigeria": {"latitude": 9.0820, "longitude": 8.6753}
  }

  # Filter DataFrame to include only countries present in country_gps
  df = df[df['Country'].isin(country_gps.keys())].copy()

  df["latitude"] = df["Country"].map(lambda x: country_gps[x]["latitude"])
  df["longitude"] = df["Country"].map(lambda x: country_gps[x]["longitude"])
```

Karena dataset asli tidak memiliki informasi latitude dan longitude, program menambahkan koordinat GPS secara manual menggunakan dictionary yang berisi nilai koordinat untuk empat negara yang terdampak Ebola. Melalui fungsi `map()`, setiap baris data akan mendapatkan nilai latitude dan longitude sesuai negara masing-masing. Penambahan koordinat ini sangat penting karena akan menjadi dasar bagi proses clustering maupun visualisasi lokasi pada peta Afrika.

3. Proses DBSCAN clustering

```
[39] ✓ 0 d
  coords = df[["latitude", "longitude"]].values
  coords_scaled = StandardScaler().fit_transform(coords)
  |
  dbscan = DBSCAN(eps=0.3, min_samples=5)
  df["Cluster"] = dbscan.fit_predict(coords_scaled)

  print(df.head())
  ...
  Indicator Country Date \
0 Cumulative number of confirmed, probable and s... Guinea 2015-03-10
1 Cumulative number of confirmed Ebola cases Guinea 2015-03-10
2 Cumulative number of probable Ebola cases Guinea 2015-03-10
3 Cumulative number of suspected Ebola cases Guinea 2015-03-10
4 Cumulative number of confirmed, probable and s... Guinea 2015-03-10

  value latitude longitude Cluster
0 3285.0 9.9456 -9.6966 0
1 2871.0 9.9456 -9.6966 0
2 392.0 9.9456 -9.6966 0
3 22.0 9.9456 -9.6966 0
4 2170.0 9.9456 -9.6966 0
```

Setelah koordinat GPS tersedia, program mengekstrak latitude dan longitude sebagai fitur untuk analisis. Data koordinat tersebut kemudian dinormalisasi menggunakan `StandardScaler()` agar memiliki skala yang sama dan tidak memengaruhi performa DBSCAN. Setelah distandarisasi, DBSCAN diterapkan dengan parameter `eps=0.3` dan `min_samples=5`. DBSCAN akan mengelompokkan titik berdasarkan kedekatan geografis. Hasil cluster kemudian

disimpan dalam kolom baru bernama “Cluster”. Cluster yang bernilai -1 menunjukkan titik yang dianggap *noise* atau tidak masuk dalam kelompok mana pun.

4. Menampilkan Peta Afrika Menggunakan GeoJSON

```
[40] ① africa_geojson_url = "https://raw.githubusercontent.com/codeforgermany/click_that_hood/main/public/data/africa.geojson"
    # Pusat Afrika
    africa_map = folium.Map(location=[7, 21], zoom_start=4)

    # Tambahkan layer Afrika
    folium.GeoJson(
        africa_geojson_url,
        name="Africa",
        style_function=lambda feature: [
            "fillColor": "#c2d6d6",
            "color": "black",
            "weight": 1,
            "fillOpacity": 0.25
        ]
    ).add_to(africa_map)
...
... <folium.features.GeoJson at 0x78f7197fe5d0>
```

Pada poin ini, program membuat peta Afrika menggunakan library Folium dan file GeoJSON yang berisi bentuk geografis benua Afrika. GeoJSON tersebut diambil dari sumber open-source dan menampilkan batas negara secara akurat. Folium kemudian memuat file GeoJSON tersebut dan menampilkannya sebagai layer peta. Langkah ini memastikan bahwa visualisasi yang dibuat merupakan peta Afrika asli, bukan peta kosong atau peta dunia yang dipotong.

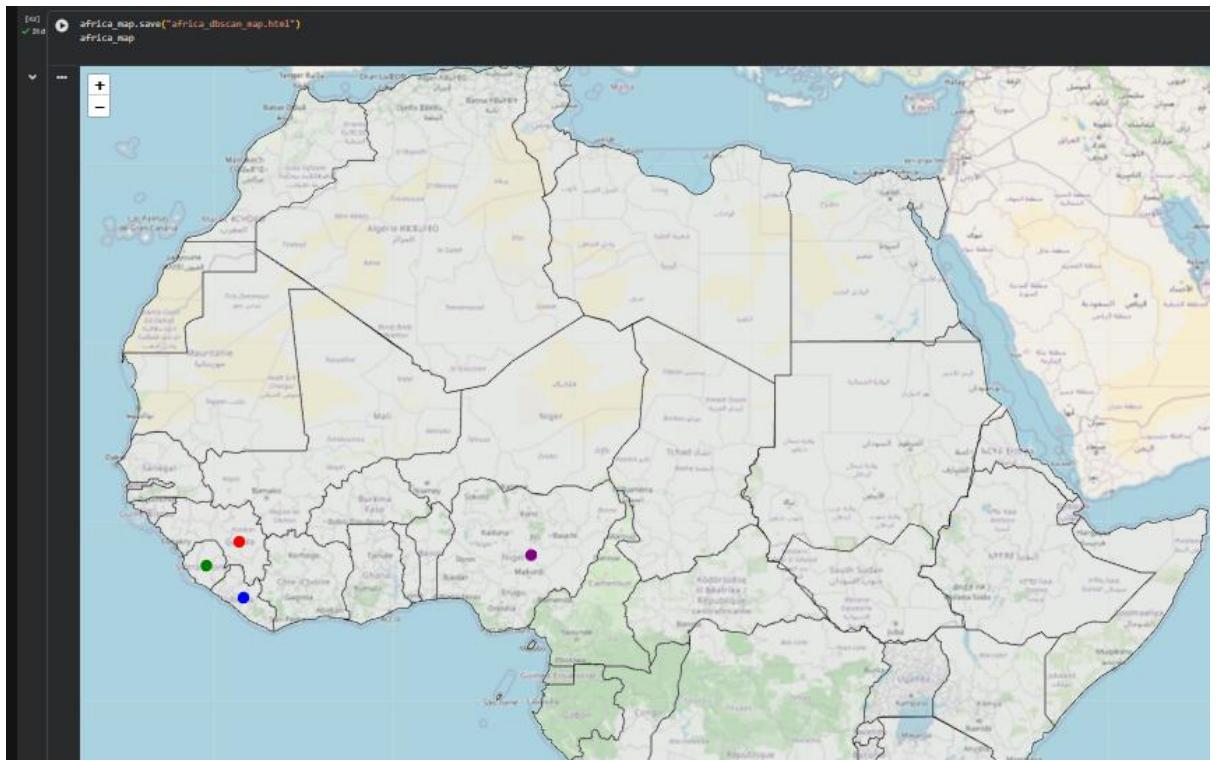
5. Menambahkan Marker Hasil Clustering ke Peta

```
[41] ✓   colors = {
        -1: "gray",
        0: "red",
        1: "blue",
        2: "green",
        3: "purple",
        4: "orange"
    }

    for _, row in df.iterrows():
        folium.CircleMarker([
            location=[row["latitude"], row["longitude"]],
            radius=7,
            color=colors.get(row["Cluster"], "black"),
            fill=True,
            fill_color=colors.get(row["Cluster"], "black"),
            fill_opacity=0.9,
            popup=(
                f"<b>Country:</b> {row['Country']}<br>" 
                f"<b>Date:</b> {row['Date']}<br>" 
                f"<b>Cases:</b> {row['value']}<br>" 
                f"<b>Cluster:</b> {row['Cluster']}"
            )
        ]).add_to(africa_map)
```

Setelah peta Afrika berhasil ditampilkan, program menambahkan marker lingkaran untuk setiap titik data pada peta. Warna marker ditentukan oleh hasil clustering DBSCAN sehingga setiap cluster dapat dibedakan secara visual. Selain itu, setiap marker dilengkapi dengan *popup* yang menampilkan informasi seperti negara, tanggal, jumlah kasus, dan nomor cluster. Dengan begitu, pengguna dapat mengeklik setiap marker untuk melihat detail data Ebola di lokasi tersebut secara interaktif.

6. Menyimpan dan Menampilkan Peta



Pada bagian terakhir, peta yang telah diberi marker cluster disimpan dalam bentuk file HTML menggunakan fungsi `save()`. File HTML ini dapat dibuka di browser untuk melihat peta secara penuh, termasuk fitur zoom dan interaksi marker. Selain disimpan, peta juga ditampilkan langsung pada notebook agar pengguna dapat melihat hasilnya segera tanpa membuka file terpisah.

LINK GITHUB :

https://github.com/Syauqi366/SyauqiRabbani_MachineLearning/blob/main/praktikumma_ndiri11.ipynb