

1. Import Library

```
from google.colab import drive  
drive.mount('/content/drive')  
  
Mounted at /content/drive  
  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.cluster import KMeans, DBSCAN
```

2. Load Dataset

```
df = pd.read_csv("/content/drive/MyDrive/praktikum/Praktikum  
11/Data/Kota.csv")  
# Deteksi kolom Latitude & Longitude  
print("5 Data Teratas:")  
display(df.head())
```

5 Data Teratas:

```

],\n      \\"semantic_type\\": \"\",\\n      \\"description\\": \"\"\n}\n  },\\n  {\n    \\"column\\": \"long\",\\n    \\"properties\\":\n    {\n      \\"dtype\\": \"number\",\\n      \\"std\\\":\n      0.6235061894801057,\n      \\"min\\\": 96.38105,\n      \\"max\\\":\n      97.81081,\n      \\"num_unique_values\\\": 5,\n      \\"samples\\\": [\n      97.79769\n    ],\\n    \\"semantic_type\\\": \"\",\\n    \\"description\\\": \"\"\n  }\n}],\\n  ]\n}","type":"dataframe"}
```

df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 514 entries, 0 to 513
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Unnamed: 0   514 non-null    int64  
 1   id          514 non-null    int64  
 2   foreign     514 non-null    int64  
 3   name        514 non-null    object  
 4   lat         514 non-null    float64 
 5   long        514 non-null    float64 
dtypes: float64(2), int64(3), object(1)
memory usage: 24.2+ KB
```

lat_cols = [c for c in df.columns if "lat" in c.lower()]
lon_cols = [c for c in df.columns if "lon" in c.lower() or "long" in c.lower()]

if len(lat_cols) == 0 or len(lon_cols) == 0:
 raise ValueError("Dataset wajib memiliki kolom Latitude dan Longitude")

lat = lat_cols[0]
lon = lon_cols[0]

print("Kolom Latitude : ", lat)
print("Kolom Longitude : ", lon)

Data untuk clustering
X = df[[lon, lat]]

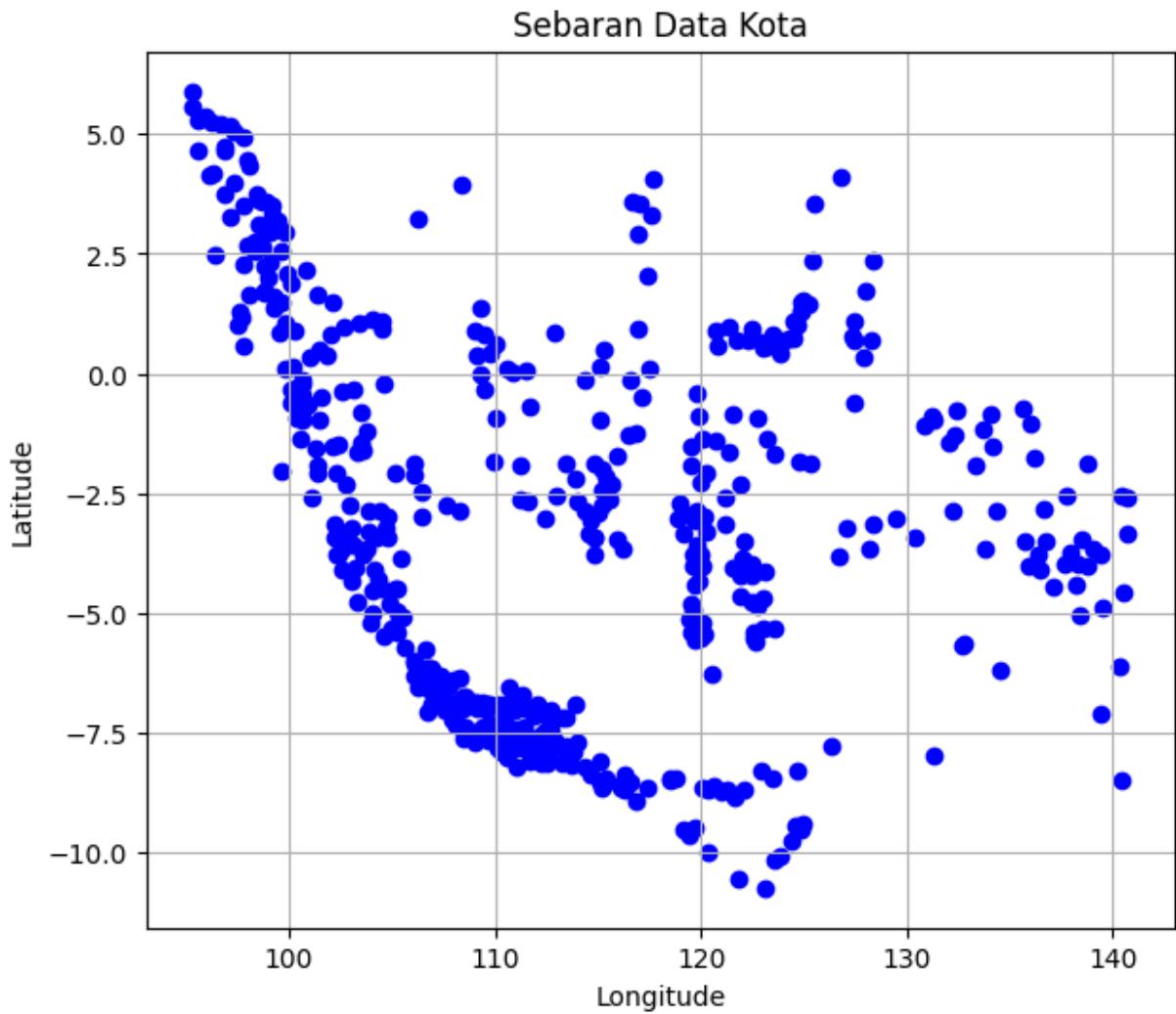
Kolom Latitude : lat
Kolom Longitude : long

3. Visualisasi

```

plt.figure(figsize=(7,6))
plt.scatter(df[lon], df[lat], c='blue')
plt.title("Sebaran Data Kota")
```

```
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.grid(True)
plt.show()
```



4. K-Means

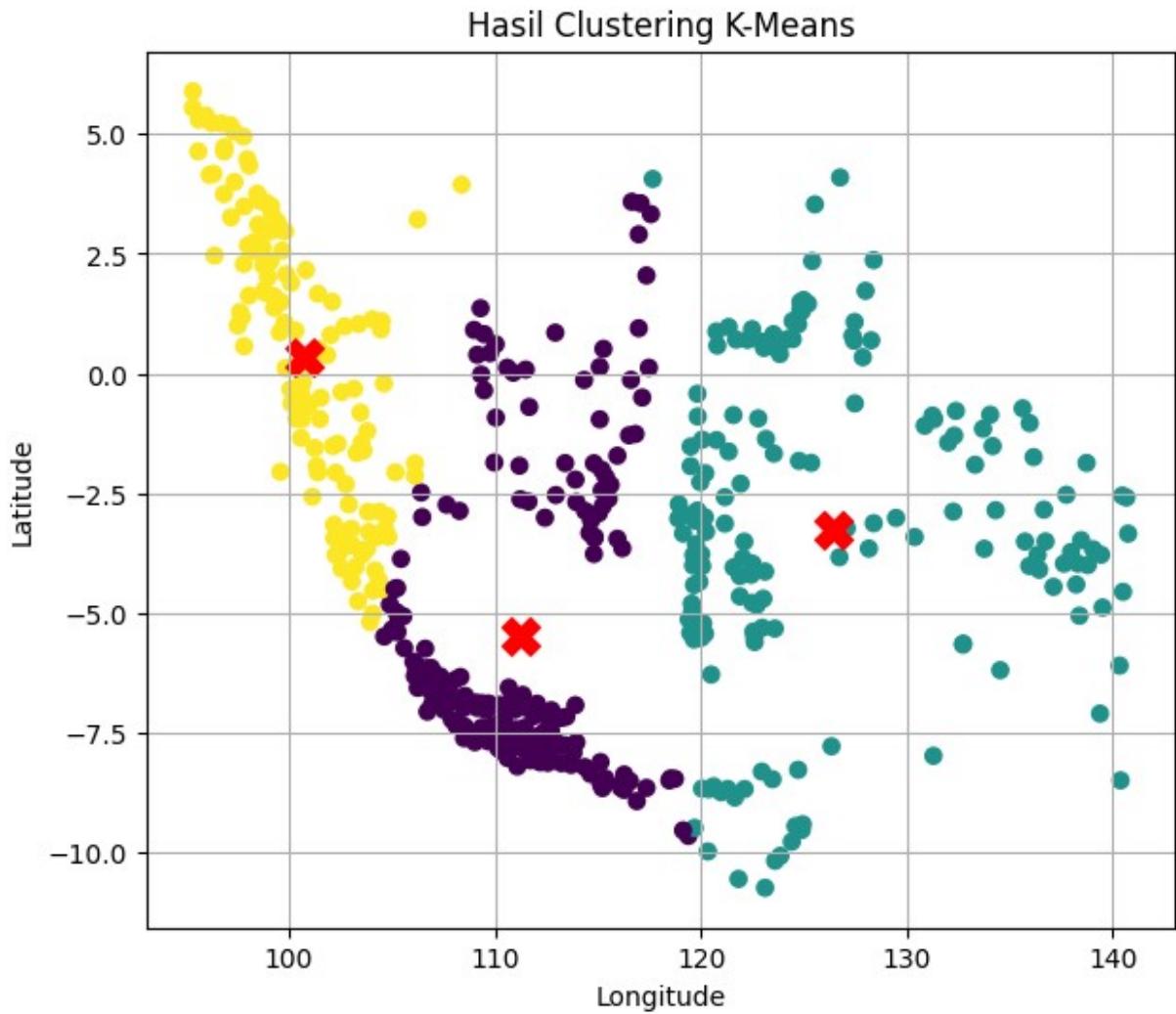
```
kmeans = KMeans(n_clusters=3, random_state=42)
df["Cluster_KMeans"] = kmeans.fit_predict(X)

plt.figure(figsize=(7,6))
plt.scatter(df[lon], df[lat], c=df["Cluster_KMeans"], cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:,0],
            kmeans.cluster_centers_[:,1],
            s=200, c='red', marker='X')
plt.title("Hasil Clustering K-Means")
```

```

plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.grid(True)
plt.show()

```



5. DB SCAN

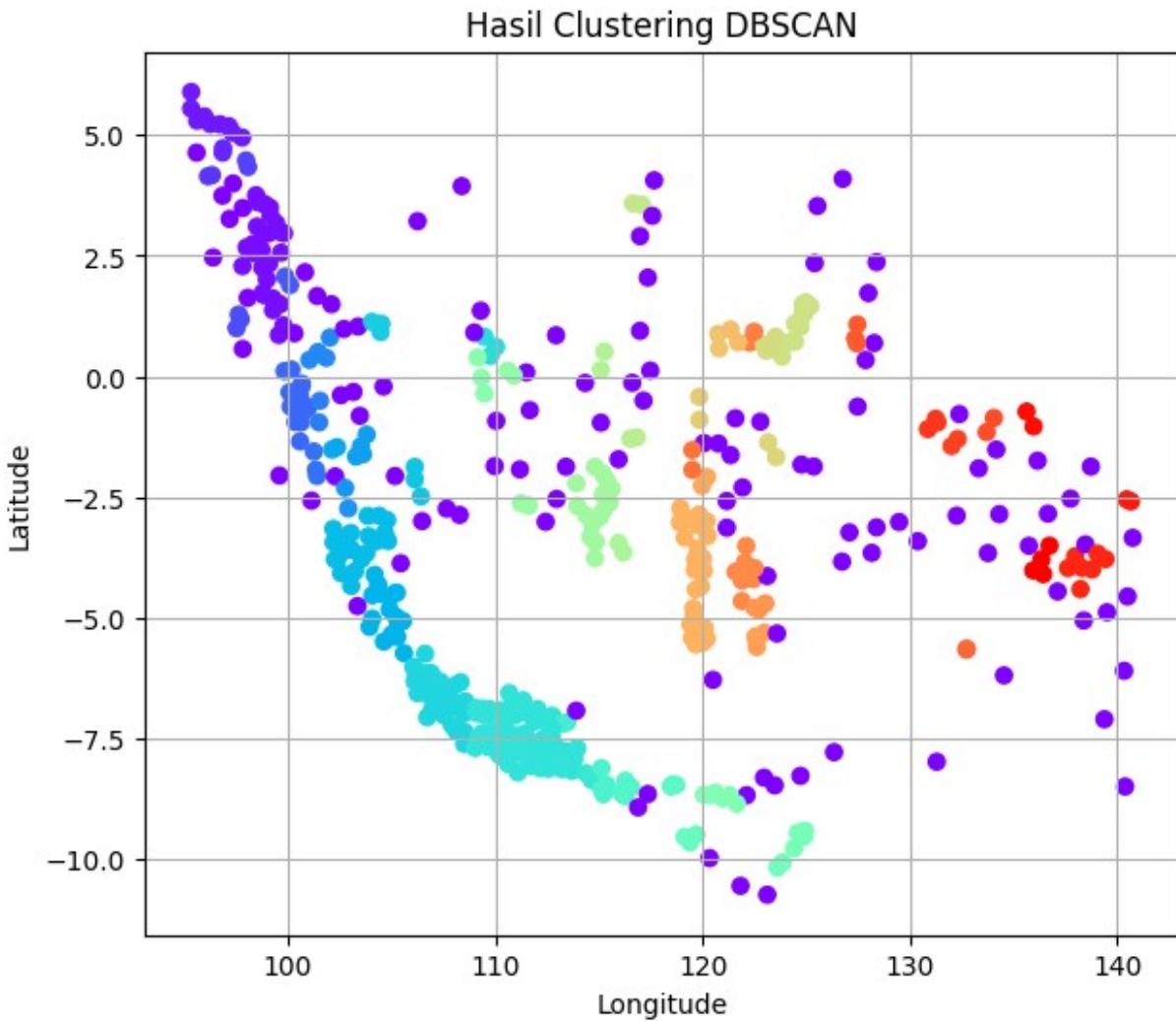
```

dbscan = DBSCAN(eps=0.5, min_samples=2)
df["Cluster_DBSCAN"] = dbscan.fit_predict(X)

plt.figure(figsize=(7,6))
plt.scatter(df[lon], df[lat], c=df["Cluster_DBSCAN"], cmap='rainbow')
plt.title("Hasil Clustering DBSCAN")
plt.xlabel("Longitude")
plt.ylabel("Latitude")

```

```
plt.grid(True)  
plt.show()
```



6. Hasil Akhir

```
print("DATASET SETELAH CLUSTERING:")
display(df)
```

DATASET SETELAH CLUSTERING:

```
{"summary": "{\n    \"name\": \"df\", \n    \"rows\": 514, \n    \"fields\": [\n        {\n            \"column\": \"Unnamed: 0\", \n            \"properties\": {\n                \"dtype\": \"number\", \n                \"std\": 148, \n                \"min\": 0, \n                \"max\": 513, \n                \"num_unique_values\": 514, \n                \"samples\": [\n                    304, \n                    497, \n                    440\n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\"\n            }\n        }\n    ]\n}
```

```

    },\n      {\n        \\"column\": \\"id\",\\n          \\"properties\": {\n          \\"dtype\": \\"number\",\\n            \\"std\": 2680,\\n              \\"min\":\n1101,\n            \\"max\": 9471,\\n              \\"num_unique_values\": 514,\n          \\"samples\": [\n            5313,\n              9416,\n              7502\n          ],\\n            \\"semantic_type\": \\"\",\\n              \\"description\": \\"\"\n        },\\n        {\n          \\"column\": \\"foreign\",\\n            \\"properties\": {\n              \\"dtype\": \\"number\",\\n                \\"std\":\n26,\n                \\"min\": 11,\n                \\"max\": 94,\n              \\"num_unique_values\": 34,\n              \\"samples\": [\n                36,\n                61,\n                74\n              ],\\n                \\"semantic_type\": \\"\",\\n                  \\"description\": \\"\"\n            },\\n            {\n              \\"column\":\n                \\"name\",\\n                  \\"properties\": {\n                    \\"dtype\": \\"string\",\\n                      \\"num_unique_values\": 514,\n                      \\"samples\": [\n                        \"KABUPATEN MANGGARAI\",\\n                          \"KABUPATEN YAHUKIMO\",\\n\n                        \"KABUPATEN GORONTALO\",\\n                          \\"semantic_type\": \\"\",\\n                            \\"description\": \\"\"\n                      ],\\n                        \\"column\": \\"lat\",\\n                          \\"properties\": {\n                            \\"dtype\":\n\"number\",\\n                              \\"std\": 3.879128622305856,\n                              \\"min\": -\n10.73864,\n                              \\"max\": 5.89285,\n                              \\"num_unique_values\":\n513,\n                              \\"samples\": [\n                                -10.73864,\n                                -\n4.88689,\n                                0.66354\n                              ],\\n                                \\"semantic_type\":\n\"\",\\n                                  \\"description\": \\"\"\n                            },\\n                            {\n                              \\"column\": \\"long\",\\n                                \\"properties\": {\n                                  \\"dtype\":\n\"number\",\\n                                    \\"std\": 11.009798659412843,\n                                    \\"min\":\n95.3208,\n                                    \\"max\": 140.76245,\n                                    \\"num_unique_values\":\n497,\n                                    \\"samples\": [\n                                      138.74361,\n                                      100.63241,\n                                      112.20187\n                                    ],\\n                                      \\"semantic_type\": \\"\",\\n                                        \\"description\": \\"\"\n                                  },\\n                                  {\n                                    \\"column\": \\"Cluster_KMeans\",\\n                                      \\"properties\": {\n                                        \\"dtype\": \\"int32\",\\n                                          \\"num_unique_values\": 3,\n                                          \\"samples\": [\n                                            2,\n                                            0,\n                                            1\n                                          ],\\n                                            \\"semantic_type\": \\"\",\\n                                              \\"description\": \\"\"\n                                        },\\n                                        {\n                                          \\"column\": \\"Cluster_DBSCAN\",\\n                                            \\"properties\": {\n                                              \\"dtype\":\n\"number\",\\n                                                \\"std\": 17,\n                                                \\"min\": -1,\n                                                \\"max\": 59,\n                                                \\"num_unique_values\": 61,\n                                                \\"samples\": [\n                                                  -1,\n                                                  4,\n                                                  45\n                                                ],\\n                                                \\"semantic_type\": \\"\",\\n                                                  \\"description\": \\"\"\n                                              }\n                                            }\n                                          }\n                                        }\n                                      }\n                                    }\n                                  }\n                                }\n                              }\n                            }\n                          }\n                        }\n                      }\n                    }\n                  }\n                }\n              }\n            }\n          }\n        }\n      }\n    }\n  }\n}\n\n", "type": "dataframe", "variable_name": "df"

```