

# **Tugas 1: Penerapan Algoritma DBSCAN untuk Analisis Densitas dan Pembentukan Cluster pada Dataset Koordinat GPS**

**Monika Septiana - 0110222127<sup>1\*</sup>**

<sup>1</sup> Teknik Informatika, STT Terpadu Nurul Fikri, Depok

\*E-mail: [moni22127ti@student.nurulfikri.ac.id](mailto:moni22127ti@student.nurulfikri.ac.id)

**Abstract.** Pada tugas ini dilakukan eksplorasi algoritma clustering DBSCAN (Density-Based Spatial Clustering of Applications with Noise) untuk menganalisis pola densitas menggunakan data koordinat GPS. Data GPS digunakan sebagai dasar untuk mengidentifikasi area dengan kepadatan tinggi (cluster) serta titik-titik yang dianggap sebagai noise atau berada di luar kelompok. Proses mencakup pembersihan data, visualisasi persebaran koordinat, penentuan parameter *eps* dan *min\_samples*, serta evaluasi hasil clustering menggunakan plot spasial. Hasil analisis menunjukkan bahwa DBSCAN mampu membentuk cluster berdasarkan kedekatan lokasi dan tingkat kepadatan, sehingga efektif digunakan pada dataset geografis tanpa perlu menentukan jumlah cluster di awal. Pendekatan ini relevan untuk berbagai kebutuhan, seperti pemetaan titik keramaian, analisis wilayah rawan, dan segmentasi area berbasis lokasi.

### ▼ 1. Import Library

```
!pip install haversine
import numpy as np
import pandas as pd
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt

Collecting haversine
  Downloading haversine-2.9.0-py2.py3-none-any.whl.metadata (5.8 kB)
  Downloading haversine-2.9.0-py2.py3-none-any.whl (7.7 kB)
Installing collected packages: haversine
Successfully installed haversine-2.9.0
```

### ▼ 2. Load Dataset

```
from google.colab import files
uploaded = files.upload()

import io
data = pd.read_csv(io.BytesIO(uploaded['beach-water-and-weather-sensor-locations-1.csv']))
data
```

Choose File: beach-water-and-weather-sensor-locations-1.csv(text/csv) - 892 bytes, last modified: 12/7/2025 - 100% done  
Saving beach-water-and-weather-sensor-locations-1.csv to beach-water-and-weather-sensor-locations-1.csv

index	Sensor Name	Sensor Type	Latitude	Longitude	Location	Actions
0	0	Calumet Beach	Water	41.714739	-87.527356 (41.71473900000002, -87.52735599999997)	
1	1	63rd Street Weather Station	Weather	41.780992	-87.572619 (41.78099199999998, -87.57261900000003)	
2	2	63rd Street Beach	Water	41.784561	-87.571453 (41.78456099999997, -87.57145300000005)	
3	3	Oak Street Weather Station	Weather	41.901997	-87.622817 (41.90199700000001, -87.62281699999998)	
4	4	Foster Weather Station	Weather	41.976464	-87.647525 (41.976464, -87.64752500000002)	
5	5	Montrose Beach	Water	41.969094	-87.638003 (41.96909399999998, -87.63800299999998)	
6	6	Osterman Beach	Water	41.987675	-87.651008 (41.98767500000003, -87.65100800000004)	
7	7	Ohio Street Beach	Water	41.894328	-87.613083 (41.89432800000002, -87.61308300000003)	
8	8	Rainbow Beach	Water	41.760147	-87.550081 (41.76014700000003, -87.55008100000006)	

Langkah berikutnya: [Buat kode dengan data](#) [New interactive sheet](#)

### ▼ 3. DBSCAN Clustering (menggunakan Haversine distance)

```
coords = data[['Latitude', 'Longitude']].to_numpy()

# Konversi jarak eps ke radian
kms_per_radian = 6371.0088
eps_km = 2 # radius 2 km, bisa disesuaikan
epsilon = eps_km / kms_per_radian

db = DBSCAN(
    eps=epsilon,
    min_samples=2,
    metric='haversine'
).fit(np.radians(coords))

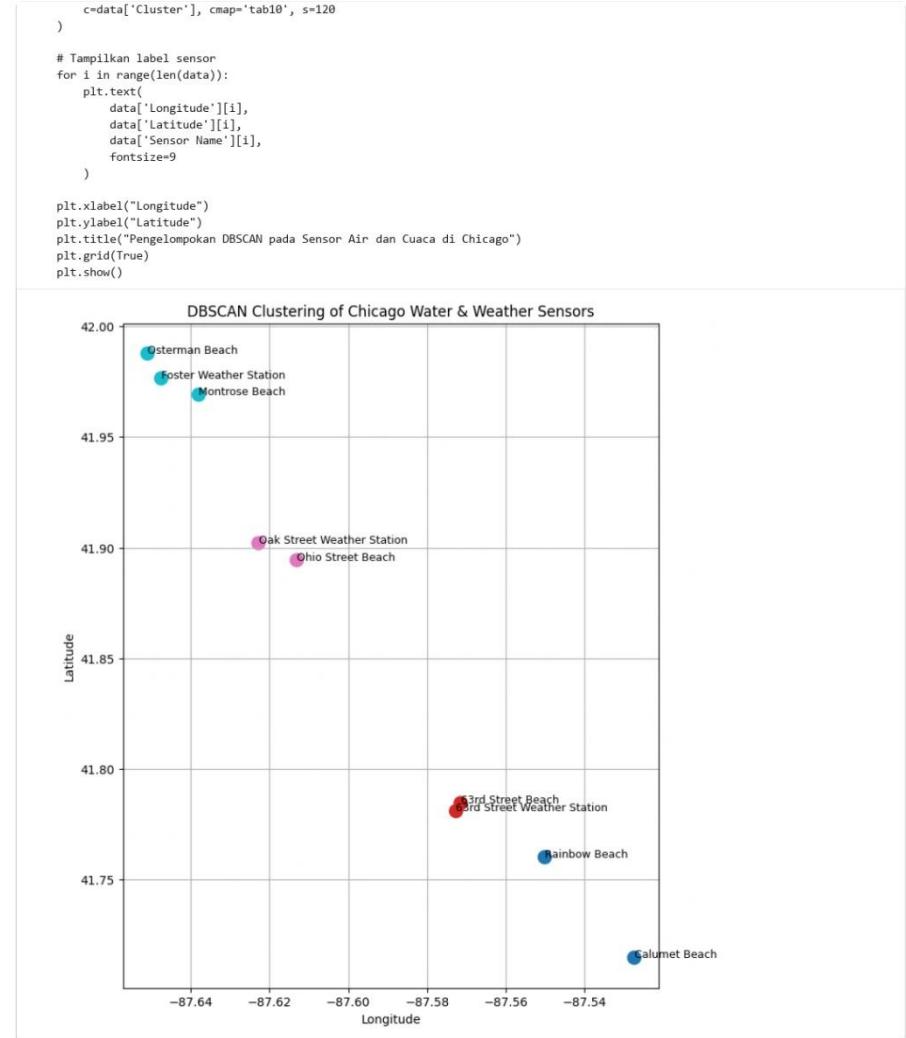
data['Cluster'] = db.labels_
data
```

index	Sensor Name	Sensor Type	Latitude	Longitude	Location	Cluster	Actions
0	0	Calumet Beach	Water	41.714739	-87.527356 (41.71473900000002, -87.52735599999997)	-1	
1	1	63rd Street Weather Station	Weather	41.780992	-87.572619 (41.78099199999998, -87.57261900000003)	0	
2	2	63rd Street Beach	Water	41.784561	-87.571453 (41.78456099999997, -87.57145300000005)	0	
3	3	Oak Street Weather Station	Weather	41.901997	-87.622817 (41.90199700000001, -87.62281699999998)	1	
4	4	Foster Weather Station	Weather	41.976464	-87.647525 (41.976464, -87.64752500000002)	2	
5	5	Montrose Beach	Water	41.969094	-87.638003 (41.96909399999998, -87.63800299999998)	2	
6	6	Osterman Beach	Water	41.987675	-87.651008 (41.98767500000003, -87.65100800000004)	2	
7	7	Ohio Street Beach	Water	41.894328	-87.613083 (41.89432800000002, -87.61308300000003)	1	
8	8	Rainbow Beach	Water	41.760147	-87.550081 (41.76014700000003, -87.55008100000006)	-1	

Langkah berikutnya: [Buat kode dengan data](#) [New interactive sheet](#)

### ▼ 4. Visualisasi Clustering

```
plt.figure(figsize=(8, 10))
plt.scatter(
    data['Longitude'], data['Latitude'],
```



## ▼ 5. Interpretasi Otomatis

```

clusters = data['Cluster'].unique()
print("Jumlah cluster ditemukan: {len(clusters) - (1 if -1 in clusters else 0)}")
print("\nDetail per cluster:\n")

for c in clusters:
    if c == -1:
        print("Cluster -1 (Noise):")
    else:
        print(f"Cluster {c}:")

    print(data[data['Cluster'] == c][['Sensor Name', 'Latitude', 'Longitude']])
    print()

Jumlah cluster ditemukan: 3

Detail per cluster:

Cluster -1 (Noise):
  Sensor Name  Latitude  Longitude
0  Calumet Beach  41.714739 -87.527356
8  Rainbow Beach  41.760147 -87.550081

Cluster 0:
  Sensor Name  Latitude  Longitude
1  63rd Street Weather Station  41.780992 -87.572619
2   63rd Street Beach  41.784561 -87.571453

Cluster 1:
  Sensor Name  Latitude  Longitude
3  Oak Street Weather Station  41.901997 -87.622817
7   Ohio Street Beach  41.894328 -87.613083

Cluster 2:

```

	Sensor Name	Latitude	Longitude
4	Foster Weather Station	41.976464	-87.647525
5	Montrose Beach	41.969094	-87.638003
6	Osterman Beach	41.987675	-87.651008

#### ▼ Interpretasi Hasil DBSCAN Clustering (Chicago Water & Weather Sensors)

Dengan parameter:

eps = 2 km

min\_samples = 2

metric = Haversine

DBSCAN menemukan 3 cluster dan 2 titik yang dianggap noise.

##### 1. Noise / Outlier (Cluster -1)

DBSCAN mengidentifikasi bahwa dua sensor ini tidak memiliki tetangga cukup dekat dalam radius 2 km:

Sensor	Lokasi	Keterangan
Calumet Beach	Selatan Chicago	Terletak sangat jauh dari pantai lain + terisolasi
Rainbow Beach	Chicago South (lebih ke pusat)	Jarak > 2 km dari 63rd Street Beach → tidak cukup rapat

##### Alasan menjadi noise:

Keduanya berada di area selatan yang lokasi pantainya jarang dan berjarak cukup jauh satu sama lain, sehingga tidak memenuhi densitas DBSCAN.

##### 2. Cluster 0 – South Chicago (Dekat 63rd Street)

###### Sensor

- 63rd Street Weather Station
- 63rd Street Beach

###### Interpretasi:

Sensor-sensor ini berada dalam jarak < 1 km dan merepresentasikan daerah pantai 63rd Street. Ini membentuk cluster yang kuat (dense region).

##### 3. Cluster 1 – Central Chicago Beaches (Downtown area)

###### Sensor

- Oak Street Weather Station
- Ohio Street Beach

###### Interpretasi:

Kedua sensor ini berada di area downtown Chicago, yang pantainya saling dekat (<1 km). Ini membentuk cluster tersendiri di area pusat kota.

##### 4. Cluster 2 – North Chicago Beaches

###### Sensor

- Foster Weather Station
- Montrose Beach
- Osterman Beach

###### Interpretasi:

Ketiga sensor ini berada di wilayah Chicago Utara, yang memang pantainya saling berdekatan secara fisik. Cluster ini menjadi cluster terbesar dan paling jelas densitasnya.

#### Kesimpulan

1. DBSCAN berhasil mengelompokkan sensor berdasarkan kedekatan geografis, meskipun jumlah cluster tidak ditentukan di awal.
2. Sistem pantai Chicago ternyata membentuk tiga zona geografis:
  - South → 63rd area
  - Central → downtown beaches
  - North → Montrose/Foster/Osterman
3. Sensor yang posisinya terlalu berjauhan (Calumet dan Rainbow) menjadi noise, karena tidak memenuhi kriteria densitas.

4. Dengan menggunakan Haversine distance, perhitungan jarak GPS menjadi akurat untuk clustering spasial.