Praktikum & Tugas 11: Clustering Koordinat GPS Menggunakan Algoritma DBSCAN

Rizky Hilmiawan Anggoro[1] - 01102221401

[1] Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: rizk22140ti@student.nurulfikri.ac.id

Abstract. Penelitian ini bertujuan untuk menerapkan algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise) untuk mengelompokkan data koordinat geografis (lintang dan bujur) dari 148.061 data kota. Data koordinat dipilih sebagai fitur utama clustering. DBSCAN dijalankan dengan parameter eps 0.5 dan min_samples 10 menggunakan metrik Euclidean. Hasil analisis menunjukkan bahwa algoritma berhasil mengidentifikasi 391 klaster. Selain itu, 13.718 titik data diidentifikasi sebagai noise atau outliers (label -1). Klaster terbesar yang ditemukan adalah Cluster 1 dengan 66.252 kota, mengindikasikan adanya kepadatan geografis yang signifikan. Langkah terakhir adalah visualisasi hasil clustering menggunakan plot pencar untuk memahami distribusi spasial klaster dan noise dalam data koordinat GPS.

1. Pilih Koordinat GPS

Langkah pertama adalah memuat data kota dari file cities.csv ke dalam sebuah DataFrame menggunakan pustaka Pandas. Kemudian, kolom yang berisi data lintang (latitude) dan bujur (longitude) dipilih dan disimpan dalam variabel X. Data ini merepresentasikan koordinat geografis setiap kota yang akan dijadikan input untuk proses clustering.

```
# 1. Pilih Koordinat GPS
X = df[['latitude', 'longitude']]
```

2. Terapkan DBSCAN

Algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise) diterapkan pada data koordinat X. DBSCAN memerlukan dua parameter utama: eps (epsilon) yang menentukan radius maksimum di sekitar suatu titik untuk mencari tetangga, dan min_samples yang menentukan jumlah minimum titik dalam radius eps agar titik tersebut dianggap sebagai core point. Dalam kasus ini, digunakan eps=0.5 dan min_samples=10 dengan metrik jarak Euclidean. Hasil clustering (dbscan.fit_predict(X)) yang berupa label cluster untuk setiap titik ditambahkan sebagai kolom baru, dbscan_cluster, pada DataFrame.

```
# 2. Terapkan DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=10, metric='euclidean')
df['dbscan_cluster'] = dbscan.fit_predict(X)
```

3. Analisis Hasil Clustering:
Setelah DBSCAN dijalankan, dilakukan analisis terhadap hasilnya. Terdapat total 148.061 baris data (kota). Hasilnya menunjukkan bahwa terdapat 391 cluster yang ditemukan (ditandai dengan label lebih dari -1) dan 13.718 titik data yang diidentifikasi sebagai noise atau outliers (ditandai dengan label -1). Kemudian, diperlihatkan 5 cluster teratas berdasarkan jumlah kota anggotanya, dengan Cluster 1 sebagai yang terbesar memiliki 66.252 kota.

```
# 3. Analisis Hasil Clustering
n_clusters = len(set(df['dbscan_cluster'])) - (1 if -1 in df['dbscan_cluster'].values else 0)

n_noise = list(df['dbscan_cluster']).count(-1)

print(f"Total Baris Data: {len(df)}")
print("-" * 30)
print(f"Jumlah Cluster yang Ditemukan (Label > -1): {n_clusters}")
print(f"Jumlah Titik Noise/Outliers (Label -1): {n_noise}")
print("-" * 30)

# Menampilkan 5 Cluster Terbesar
cluster_counts = Counter(df['dbscan_cluster'])
if -1 in cluster_counts:
    del cluster_counts[-1]

print("Cluster Sizes (Top 5):")
for cluster_id, count in cluster_counts.most_common(5):
    print(f"Cluster {cluster_id}: {count} kota")
```

4. Visualisasi
Langkah terakhir adalah memvisualisasikan hasil clustering ini untuk melihat bagaimana titik-titik (kota) terdistribusi dan terkelompok berdasarkan koordinat lintang dan bujur. Karena jumlah data sangat besar, dilakukan pengambilan sampel acak sebanyak 50.000 data untuk membuat plot pencar. Plot tersebut menampilkan titik-titik koordinat di mana warna setiap titik merepresentasikan ID cluster-nya, termasuk noise (label -1).

```python
# 4. Visualisasi (Opsional)
plt.figure(figsize=(10, 8))
sample_df = df.sample(n=50000, random_state=42) if len(df) > 50000 else df
plt.scatter(sample_df['longitude'], sample_df['latitude'], c=sample_df['dbscan_cluster'], cmap='viridis', s=5, alpha=0.5)
plt.title('DBSCAN Clustering pada Koordinat GPS (Sampel Data)')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.colorbar(label='Cluster ID (-1 adalah Noise)')
plt.show()
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
['data', 'notebooks']
Total Baris Data: 148061
------------------------------
Jumlah Cluster yang Ditemukan (Label > -1): 391
Jumlah Titik Noise/Outliers (Label -1): 13718
------------------------------
Cluster Sizes (Top 5):
Cluster 1: 66252 kota
Cluster 88: 15673 kota
Cluster 54: 10448 kota
Cluster 245: 6434 kota
Cluster 12: 2835 kota
```