

DBSCAN (Density-Based Spatial Clustering of Application with Noise)

Adinda Putri - 13523071

DBSCAN merupakan algoritma yang mengelompokkan data yang saling berdekatan dan menandai outliers sebagai noise berdasarkan densitasnya di ruang fitur. **DBSCAN** bekerja baik untuk arbitrary-shaped clusters, dan untuk mengidentifikasi serta meng-handle noise dan outliers.

Cara Kerja

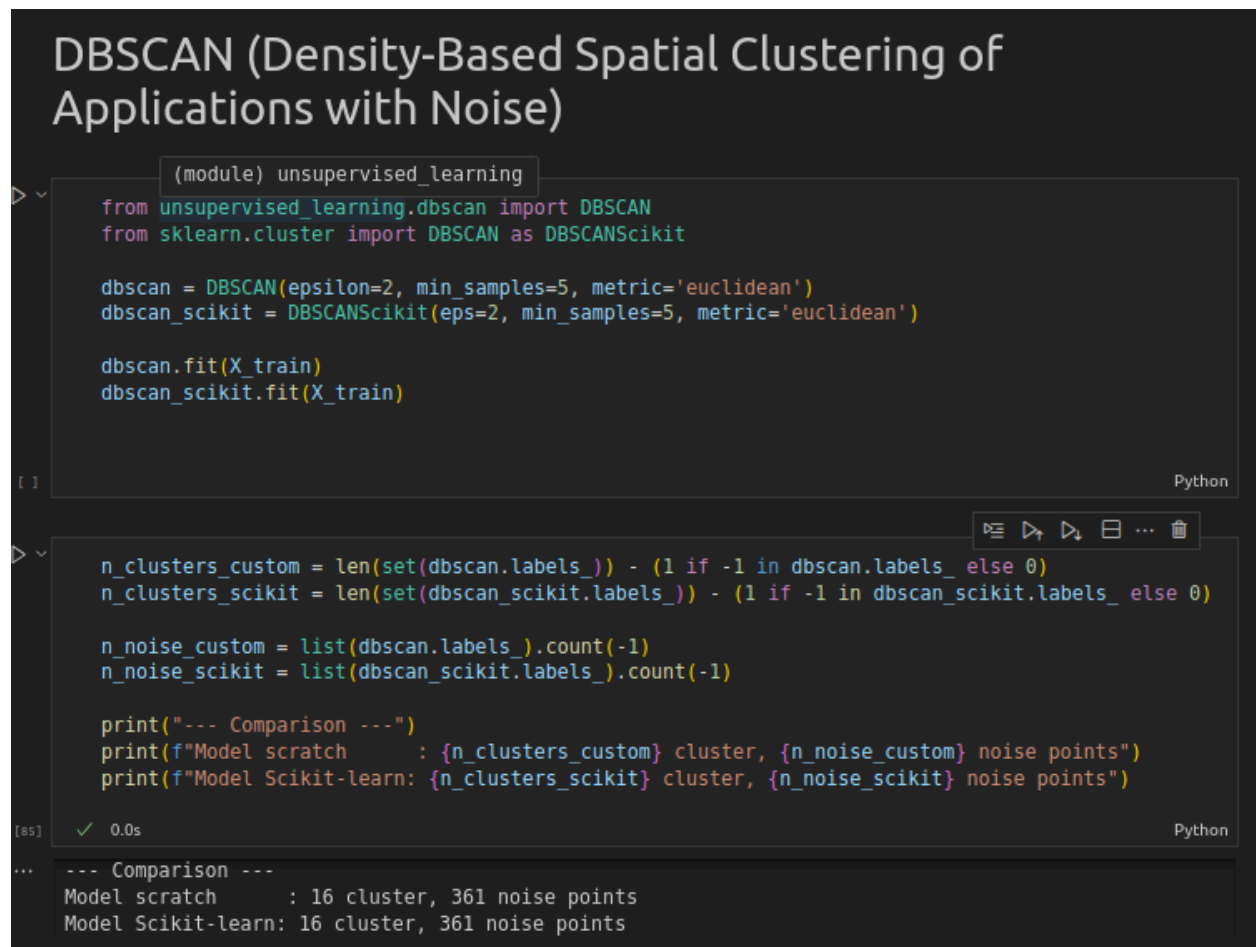
DBSCAN mengkategorikan data points menjadi tiga jenis yaitu:

1. **Core points**, yaitu data point yang memiliki jumlah tetangga yang cukup (minimal sebanyak **min_samples**) dalam radius tertentu (**epsilon**). Parameter **min_samples** dan **epsilon** bersifat pre-defined.
2. **Border points**, yaitu data point yang dekat dengan core point tetapi jumlah tetangga mereka tidak cukup (kurang dari **min_samples**) untuk dikategorikan menjadi core points
3. **Noise points**, yaitu data point yang tidak termasuk ke cluster manapun

Langkah-langkah dari algoritmanya adalah sebagai berikut.

1. **Identify Core Points:** Untuk setiap point pada data, hitung jumlah tetangganya dalam radius epsilon. Jika jumlah tetangganya lebih besar atau sama dengan **min_samples**, tandai point tersebut sebagai core point.
2. **Form Clusters:** Untuk setiap core point yang belum termasuk ke cluster manapun, buat cluster baru. Temukan semua density-connected points dalam radius epsilon dari core point tersebut secara rekursif dan tambahkan points tersebut ke cluster.
3. **Density Connectivity:** Point P dikatakan density-reachable dari Q jika P merupakan tetangga dari Q dalam radius epsilon dan keduanya merupakan core points. Point P dikatakan density-connected ke Q jika terdapat chain of points $P_1, P_2, P_3, \dots, P_n$, $P_1 = P$ and $P_n = Q$ sedemikian sehingga P_{i+1} directly density reachable dari P_i .
4. **Label Noise Points:** Setelah semua points diproses, setiap point yang tidak termasuk ke cluster manapun ditandai sebagai noise

Perbandingan model dari scratch dengan dari Scikit-Learn



The image shows a Jupyter Notebook with the following content:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

```
(module) unsupervised_learning
from unsupervised_learning.dbscan import DBSCAN
from sklearn.cluster import DBSCAN as DBSCANScikit

dbscan = DBSCAN(epsilon=2, min_samples=5, metric='euclidean')
dbscan_scikit = DBSCANScikit(eps=2, min_samples=5, metric='euclidean')

dbscan.fit(X_train)
dbscan_scikit.fit(X_train)
```

```
n_clusters_custom = len(set(dbscan.labels_)) - (1 if -1 in dbscan.labels_ else 0)
n_clusters_scikit = len(set(dbscan_scikit.labels_)) - (1 if -1 in dbscan_scikit.labels_ else 0)

n_noise_custom = list(dbscan.labels_).count(-1)
n_noise_scikit = list(dbscan_scikit.labels_).count(-1)

print("--- Comparison ---")
print(f"Model scratch      : {n_clusters_custom} cluster, {n_noise_custom} noise points")
print(f"Model Scikit-learn: {n_clusters_scikit} cluster, {n_noise_scikit} noise points")
```

```
... --- Comparison ---
Model scratch      : 16 cluster, 361 noise points
Model Scikit-learn: 16 cluster, 361 noise points
```

Gambar 1. Inisialisasi, Training, dan Perbandingan Masing-Masing Model

Sumber: Penulis

Hasil di atas menunjukkan bahwa model DBSCAN dari scratch menghasilkan output yang identik dengan model DBSCAN dari Scikit-Learn. Kedua model berhasil menemukan 16 cluster dan 361 noise points.

Referensi:

[1] *DBSCAN Clustering in ML – Density based clustering*, GeeksforGeeks. [Daring]. Tersedia: <https://www.geeksforgeeks.org/machine-learning/dbscan-clustering-in-ml-density-based-clustering/>. [Diakses: 2 September 2025].

[2] *Clustering with DBSCAN, Clearly Explained!!!*, YouTube (StatQuest). [Daring]. Tersedia: <https://www.youtube.com/live/RDZUdRSDOok>. [Diakses: 2 September 2025].

[3] *DBSCAN Clustering from Scratch*, Kaggle Notebook (phunghieu). [Daring]. Tersedia: <https://www.kaggle.com/code/phunghieu/dbscan-clustering-from-scratch>. [Diakses: 2 September 2025].