LAPORAN

MESIN LEARNING

Nama : Muhammad Nasih (2341720009)

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

TAHUN 2025

TUGAS STUDI KASUS PEMBELAJARAN MESIN

Clustering dan Approximate Nearest Neighbor (ANN)

LINK GITHUB :

https://github.com/muhnasih/MESIN-LEARNING

LINK COLLAB :

https://colab.research.google.com/drive/1jXD_0NxgHVBGJEnSOmzZnB0bsmUaKToI?usp=sharing

Deskripsi Umum:

Tugas kali ini adalah mengerjakan **studi kasus analisis data dan clustering** menggunakan *unsupervised learning* dengan langkah-langkah sebagai berikut:

1. **Preprocessing data**
    - Tangani *missing values* (imputasi mean/median/modus sesuai jenis data)
    - Normalisasi atau standarisasi data
    - Buat minimal satu fitur baru hasil kombinasi fitur lama
2. **Clustering**
    - Terapkan **K-Means** dan **DBSCAN**
    - Bandingkan hasil clustering menggunakan:
        - **Silhouette Score**
        - **Davies–Bouldin Index**
3. **Approximate Nearest Neighbor (ANN)**
    - Gunakan **Annoy** untuk mencari tetangga terdekat dari beberapa *query points* hasil clustering
    - Tampilkan *output* berupa:
        - Index *query point*
        - Daftar tetangga terdekat yang ditemukan
        - Nilai jarak kemiripan

Tugas 3 — Heart Disease Dataset

Untuk mahasiswa dengan nomor absen 3, 6, 10, dst.

- **Dataset:** Heart Disease Dataset (UCI)

- **Deskripsi:** Dataset medis untuk melihat pengelompokan pasien berdasarkan fitur kesehatan seperti tekanan darah, kolesterol, umur, dan lain-lain.
- **Langkah tambahan:**
    - Tangani nilai kosong (jika ada).
    - Buat fitur gabungan seperti “CholAge = cholesterol × age”.

Output yang Diharapkan

Untuk setiap studi kasus, laporan dan notebook Colab harus memuat:

1. Penjelasan singkat dataset (jumlah sampel, fitur, tipe data).

```
=== INFORMASI DATASET ===
Jumlah sampel (baris): 1025
Jumlah fitur (kolom): 14

Tipe data setiap kolom:
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object

Cek Missing Values:
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

5 Data Teratas:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

Statistik Deskriptif:

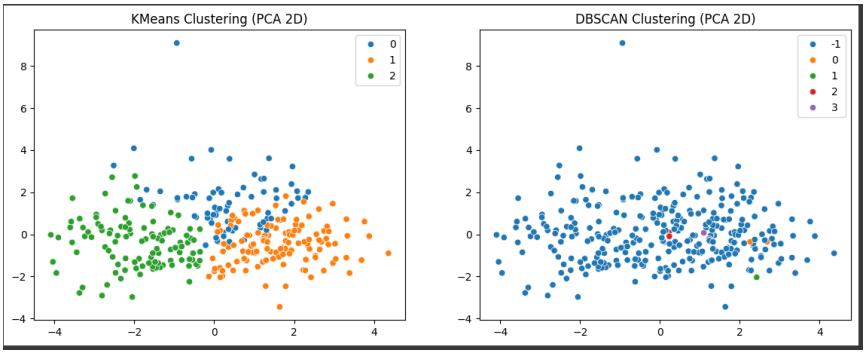
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146	2.323902	0.513171
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798	0.620660	0.500070
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

2. Proses preprocessing (missing values, normalisasi, pembuatan fitur baru).

```
=== PREPROCESSING DATA ===
Missing values tiap kolom:
age                   0
sex                   0
cp                    0
trestbps              0
chol                  0
fbs                   0
restecg               0
thalach               0
exang                 0
oldpeak               0
slope                 0
ca                    0
thal                  0
target                0
chol_trest_ratio      0
heart_rate_age_diff   0
dtype: int64

Fitur baru berhasil ditambahkan (jika kolom tersedia).

Data berhasil dinormalisasi.
Shape data sebelum: (1025, 15)
Shape data sesudah scaling: (1025, 15)
```

3. Hasil clustering KMeans dan DBSCAN, lengkap dengan:
    - Nilai Silhouette dan Davies–Bouldin
    - Visualisasi 2D (PCA/TSNE opsional)

```
=== CLUSTERING KMEANS & DBSCAN ===

--- KMeans ---
Jumlah Cluster: 3
Silhouette Score: 0.1224
Davies-Bouldin Score: 2.2558

--- DBSCAN ---
Cluster Ditemukan (tanpa -1 noise): 4
Silhouette Score: -0.1669740242604239
Davies-Bouldin Score: 1.223970063339141
```



4. Implementasi Annoy:
    - Pemilihan 3–5 titik query secara acak

- Output index dan tetangga terdekat dengan nilai jaraknya

```
Building Annoy index with dim=15, n_trees=50, metric=euclidean ...
Annoy index built.
Annoy index saved to annoy_heart_index.ann

Selected 3 random query indices: [771, 136, 612]

=== Query index: 771 ===
Query vector (first 8 dims): [-1.0404 -1.5117 -0.9158  0.3649 -0.1939 -0.4189 -1.004   0.1255]
Returned neighbors (count=10):
  01. idx=  95  distance=0.000000
  02. idx= 771  distance=0.000000
  03. idx= 953  distance=0.000000
  04. idx= 355  distance=0.271880
  05. idx= 408  distance=0.271880
  06. idx= 640  distance=0.271880
  07. idx= 320  distance=2.614772
  08. idx= 635  distance=2.614772
  09. idx= 954  distance=2.614772
  10. idx= 280  distance=2.733851
```

```
=== Query index: 136 ===
Query vector (first 8 dims): [ 0.0624 -1.5117  0.0559  0.0222  1.8616 -0.4189  0.8913  0.7343]
Returned neighbors (count=10):
  01. idx=  44  distance=0.000000
  02. idx= 136  distance=0.000000
  03. idx= 263  distance=0.000000
  04. idx= 302  distance=0.000000
  05. idx= 288  distance=1.716506
  06. idx= 602  distance=1.716506
  07. idx= 637  distance=1.716506
  08. idx= 233  distance=2.083118
  09. idx= 617  distance=2.083118
  10. idx= 654  distance=2.083118

=== Query index: 612 ===
Query vector (first 8 dims): [ 0.3932 -1.5117 -0.9158  2.1926 -0.4072  2.3873 -1.004  -0.1354]
Returned neighbors (count=10):
  01. idx= 135  distance=0.000000
  02. idx= 264  distance=0.000000
  03. idx= 612  distance=0.000000
  04. idx= 819  distance=0.000000
  05. idx=  47  distance=4.263064
  06. idx= 229  distance=4.263064
  07. idx= 452  distance=4.263064
  08. idx= 944  distance=4.263064
  09. idx= 175  distance=4.368378
  10. idx= 294  distance=4.368378
```

Pengumpulan: notebook ipynb link dimasukkan pada laporan tugas berformat pdf. Dikumpulkan pada LMS

1. Tulis kesimpulan singkat:
    a. Perbedaan hasil KMeans dan DBSCAN, mana yang lebih baik diantara kedua model ini dan jelaskan jawaban anda

Aspek	KMeans	DBSCAN
Prinsip kerja	Membagi data ke dalam jumlah cluster tertentu (k).	Mencari cluster berdasarkan kepadatan data (density).
Bentuk cluster	Cenderung membentuk cluster berbentuk bulat (spherical).	Mampu membentuk cluster tidak beraturan & mendeteksi noise.
Parameter utama	Jumlah cluster (k).	eps (radius) & min_samples.
Outlier/Noise	Setiap data masuk ke cluster tertentu.	Dapat mengidentifikasi data sebagai noise (-1).

    b. Nilai metrik terbaik (Silhouette, DBI).

- Silhouette Score tertinggi menunjukkan pemisahan cluster terbaik dan jarak antar titik dalam cluster yang rapat.

- Davies–Bouldin Index (DBI) terendah menunjukkan cluster lebih kompak dan berjauhan satu sama lain.

```
=== CLUSTERING KMEANS & DBSCAN ===

--- KMeans ---
Jumlah Cluster: 3
Silhouette Score: 0.1224
Davies-Bouldin Score: 2.2558

--- DBSCAN ---
Cluster Ditemukan (tanpa -1 noise): 4
Silhouette Score: -0.1669740242604239
Davies-Bouldin Score: 1.223970063339141
```

c. Hasil query Annoy: apakah tetangga yang ditemukan termasuk dalam cluster yang sama? Jelaskan jawaban anda.

Jika sebagian besar tetangga terdekat dari titik query berada dalam cluster yang sama, berarti:

- Hasil clustering konsisten dengan struktur ruang vektor Annoy.
- Annoy berhasil menemukan titik-titik yang memang memiliki kemiripan fitur.

Jika banyak tetangga berada di cluster berbeda, maka:

- Bisa jadi cluster kurang terpisah dengan baik.
- Atau dimensi jarak Annoy berbeda dengan struktur cluster yang dibentuk.