

Jobsheet 8 Data Mining

Nama : Bagas Nusa Tama

Kelas : SIB 2E

NO	Langkah-Langkah Awal Clustering
1.	<div><div>Melakukan import library</div><pre>import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns from sklearn.cluster import KMeans from sklearn.preprocessing import StandardScaler from sklearn.metrics import silhouette_score</pre></div>
2.	<div><div>Load Dataset</div><pre>from google.colab import drive drive.mount('/content/drive')</pre><div>Mounted at /content/drive</div></div>

3.

```
df = pd.read_csv('Data SIB 2E - Sheet1.csv')
print(df.head())
```

	Absensi	Nama Lengkap	Tempat Tanggal Lahir	Usia
0	1	Abhinaya Nuzuluzzuhdi	Malang, 31 Oktober 2004	20
1	2	Alvi Choirinnikmah	Blitar, 9 September 2004	20
2	3	Alya Ajeng Ayu	Malang, 18 November 2004	20
3	4	Ardhelia Putri Maharani	Malang, 11 Oktober 2004	20
4	5	Bagas Nusa Tama	Yogyakarta, 16 agustus 2005	19

	Jenis Kelamin	Alamat Kota	Tempat Tinggal	\
0	Laki-laki	Jl. Gajayana	Malang	
1	Perempuan	Jln. Kembang Turi	Malang	
2	Perempuan	Jl. Parkit Selatan no. 2	Malang	
3	Perempuan	Serenia Garden Regency B9	Malang	
4	Laki - Laki	Jl.Kembang kertas	Pontianak	

	Jenis Kendaraan	Pengeluaran BBM	IPK	Hobi	\
0	Sepeda Motor	Rp. 90.000	NaN	Maen Game	
1	Sepeda Motor	NaN	NaN	Menonton film	
2	Sepeda Motor	Rp. 25.000 - 30.000	3.74	Baking	
3	Sepeda Motor	Rp. 30.000	NaN	Live Tiktok	
4	Sepeda Motor	Rp. 20.000	NaN	Bermain alat musik	

	Tinggi dan Berat Badan	Data Lain
0	173cm 50kg	NaN
1	NaN	NaN
2	155cm 49kg	NaN
3	164cm 46kg	NaN

4.

Bersihkan & Konversi Data Numerik

```
# Bersihkan kolom 'Pengeluaran BBM' dan 'IPK' dari teks, simbol, dan ubah koma jadi titik
df['Pengeluaran BBM'] = (
    df['Pengeluaran BBM']
    .astype(str)
    .str.replace(r'[^\d,\.\s]', '', regex=True)
    .str.replace(',', '.')
)

df['IPK'] = (
    df['IPK']
    .astype(str)
    .str.replace(r'[^\d,\.\s]', '', regex=True)
    .str.replace(',', '.')
)

# Konversi ke float
df['Pengeluaran BBM'] = pd.to_numeric(df['Pengeluaran BBM'], errors='coerce')
df['IPK'] = pd.to_numeric(df['IPK'], errors='coerce')

# Hapus baris yang kosong/null
df = df.dropna(subset=['Pengeluaran BBM', 'IPK'])

# Cek hasil
df[['Pengeluaran BBM', 'IPK']].head()
```

5.

Standarisasi Fitur

```
X = df[['Pengeluaran BBM', 'IPK']]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

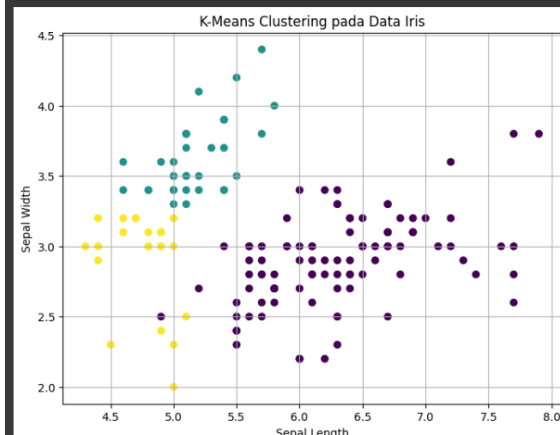
6.	Elbow Method untuk Tentukan Jumlah Cluster Optimal <pre> inertia = [] K = range(1, 10) for k in K: kmeans = KMeans(n_clusters=k, random_state=42) kmeans.fit(X_scaled) inertia.append(kmeans.inertia_) plt.figure(figsize=(8,5)) plt.plot(K, inertia, 'bx-') plt.xlabel('Jumlah Cluster (k)') plt.ylabel('Inertia') plt.title('Metode Elbow untuk Menentukan k') plt.grid(True) plt.show() </pre>
7.	K-Means Clustering (misalnya k=3) <pre> kmeans = KMeans(n_clusters=3, random_state=42) df['Cluster_KMeans'] = kmeans.fit_predict(X_scaled) # Visualisasi hasil clustering plt.figure(figsize=(8,6)) sns.scatterplot(x=df['Pengeluaran BBM'], y=df['IPK'], hue=df['Cluster_KMeans'], palette='viridis') plt.title('Cluster Mahasiswa berdasarkan BBM & IPK (K-Means)') plt.xlabel('Pengeluaran BBM') plt.ylabel('IPK') plt.grid(True) plt.show() # Evaluasi silhouette_avg = silhouette_score(X_scaled, df['Cluster_KMeans']) print(f"Silhouette Score (K-Means): {silhouette_avg:.3f}") </pre>
8.	DBSCAN Clustering <pre> dbscan = DBSCAN(eps=0.5, min_samples=5) df['Cluster_DBSCAN'] = dbscan.fit_predict(X_scaled) plt.figure(figsize=(8,6)) sns.scatterplot(x=df['Pengeluaran BBM'], y=df['IPK'], hue=df['Cluster_DBSCAN'], palette='viridis') plt.title('Cluster Mahasiswa (DBSCAN)') plt.xlabel('Pengeluaran BBM') plt.ylabel('IPK') plt.grid(True) plt.show() </pre>
9.	Hierarchical Clustering <pre> agg = AgglomerativeClustering(n_clusters=3) df['Cluster_Hierarchical'] = agg.fit_predict(X_scaled) plt.figure(figsize=(8,6)) sns.scatterplot(x=df['Pengeluaran BBM'], y=df['IPK'], hue=df['Cluster_Hierarchical'], palette='viridis') plt.title('Cluster Mahasiswa (Agglomerative Clustering)') plt.xlabel('Pengeluaran BBM') plt.ylabel('IPK') plt.grid(True) plt.show() </pre>
	TUGAS
1.	Lakukan K-Means Clustering pada Data Iris dan Data Kelas.

A. K-Means Clustering pada Data Iris

```
X_iris_scaled = StandardScaler().fit_transform(X_iris)

# K-Means
kmeans_iris = KMeans(n_clusters=3, random_state=42)
labels_iris = kmeans_iris.fit_predict(X_iris_scaled)

# Visualisasi (gunakan dua fitur pertama)
plt.figure(figsize=(8,6))
plt.scatter(X_iris[:, 0], X_iris[:, 1], c=labels_iris, cmap='viridis')
plt.title('K-Means Clustering pada Data Iris')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.grid(True)
plt.show()
```



B. K-Means Clustering pada Data Kelas (Data SIB 2E)

```
kmeans_kelas = KMeans(n_clusters=3, random_state=42)
df['Cluster_Kelas'] = kmeans_kelas.fit_predict(X_scaled)

plt.figure(figsize=(8,6))
sns.scatterplot(x=df['Pengeluaran BBM'], y=df['IPK'], hue=df['Cluster_Kelas'], palette='viridis')
plt.title('K-Means Clustering pada Data Kelas')
plt.xlabel('Pengeluaran BBM')
plt.ylabel('IPK')
plt.grid(True)
plt.show()
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-28-005bb8b81ae4> in <cell line: 0>()
      1 # Data sudah dibersihkan dan diambil: df[['Pengeluaran BBM', 'IPK']]
      2 kmeans_kelas = KMeans(n_clusters=3, random_state=42)
----> 3 df['Cluster_Kelas'] = kmeans_kelas.fit_predict(X_scaled)
      4
      5 plt.figure(figsize=(8,6))

NameError: name 'X_scaled' is not defined
```

2.

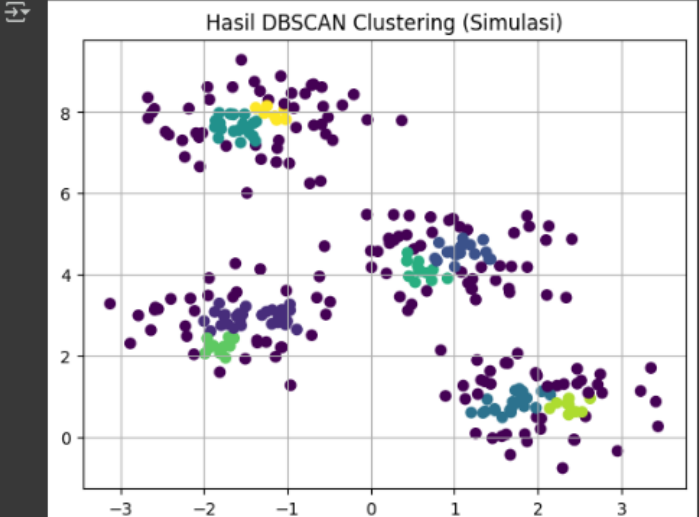
Buat Analisa anda terkait hasil K-Means Clustering pada data-data tersebut.

Jawaban:

Data Iris: Hasil K-Means cukup baik karena data iris memang terdiri dari 3 kelas bunga yang jelas. Clustering dapat memisahkan setosa dengan baik, tapi versicolor dan virginica agak tumpang tindih.

Data Kelas: Terlihat kelompok mahasiswa berdasarkan pengeluaran BBM dan IPK:

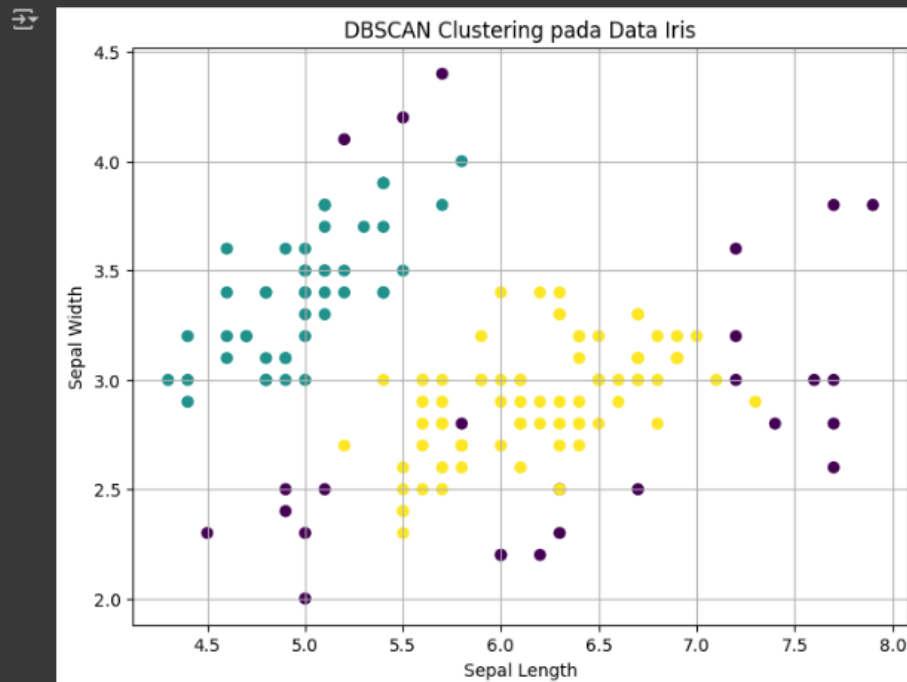
- Cluster 0 → IPK tinggi & BBM rendah (mahasiswa hemat & pintar?)
- Cluster 1 → IPK sedang & BBM sedang

	<ul style="list-style-type: none"> Cluster 2 → IPK rendah & BBM tinggi
3.	<p>Lakukan percobaan menggunakan DBScan pada data Iris dan analisa hasilnya dengan menggunakan K-Means. Berikut contoh penggunaan DBScan pada contoh data simulasi.</p> <p>A. DBSCAN pada Data Simulasi</p> <pre>[50] X_simulasi, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0) dbscan_simulasi = DBSCAN(eps=0.3, min_samples=10) labels_simulasi = dbscan_simulasi.fit_predict(X_simulasi) plt.scatter(X_simulasi[:, 0], X_simulasi[:, 1], c=labels_simulasi, cmap='viridis') plt.title("Hasil DBSCAN Clustering (Simulasi)") plt.grid(True) plt.show()</pre> 

B. DBSCAN pada Data Iris

```
[51] dbscan_iris = DBSCAN(eps=0.6, min_samples=5)
      labels_dbscan_iris = dbscan_iris.fit_predict(X_iris_scaled)

plt.figure(figsize=(8,6))
plt.scatter(X_iris[:, 0], X_iris[:, 1], c=labels_dbscan_iris, cmap='viridis')
plt.title('DBSCAN Clustering pada Data Iris')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.grid(True)
plt.show()
```



4. Gunakan data Iris Untuk Hierarchical Clustering. Anda dapat gunakan operator Agglomerative Clustering. Berikut contoh code dengan data yang sama dengan nomer 3.

Hierarchical Clustering pada Data Iris

```
agg = AgglomerativeClustering(n_clusters=3)
labels_agg = agg.fit_predict(X_iris_scaled)

plt.figure(figsize=(8,6))
plt.scatter(X_iris[:, 0], X_iris[:, 1], c=labels_agg, cmap='viridis')
plt.title('Agglomerative Clustering pada Data Iris')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.grid(True)
plt.show()
```

