

LAMPIRAN

```

121 | from google.colab import files
✓ 44s | uploaded = files.upload()

141 | Choose File(s) Book5.xlsx
    | Book5.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 20582 bytes, last modified: 2/24/2026 - 100% done
    | Saving Book5.xlsx to Book5.xlsx

161 | import pandas as pd
✓ 1s | import numpy as np
    | import matplotlib.pyplot as plt
    | import seaborn as sns

    | from sklearn.preprocessing import StandardScaler
    | from sklearn.cluster import DBSCAN
    | from sklearn.neighbors import NearestNeighbors
    | from sklearn.decomposition import PCA
    | from sklearn.metrics import silhouette_score

171 | df = pd.read_excel("data.xlsx")
✓ 0s | print("Preview data:")
    | df.head()

181 | fitur = [
✓ 0s |     "akomodasi",
    |     "konsumsi",
    |     "transportasi",
    |     "belanja",
    |     "tiket",
    |     "lama_tinggal"
    | ]
    |
    | X = df[fitur]

191 | scaler = StandardScaler()
✓ 0s | X_scaled = scaler.fit_transform(X)

201 | minPts = len(fitur) + 1
✓ 0s | print("Nilai MinPts:", minPts)

    |
    | Nilai MinPts: 7

211 | neighbors = NearestNeighbors(n_neighbors=minPts)
✓ 0s | neighbors_fit = neighbors.fit(X_scaled)
    | distances, indices = neighbors_fit.kneighbors(X_scaled)
    |
    | distances = np.sort(distances[:, -1])

221 | plt.figure(figsize=(8,4))
✓ 0s | plt.plot(distances)
    | plt.xlabel("Data ke-")
    | plt.ylabel("Jarak")
    | plt.title("K-Distance Graph (Penentuan Epsilon)")
    | plt.grid()
    | plt.show()

```

```

113] eps_value = 2.5
✓ Oz
dbSCAN = DBSCAN(eps=eps_value, min_samples=minPts)
cluster_labels = dbSCAN.fit_predict(X_scaled)

df["cluster"] = cluster_labels

114]
✓ Oz
jumlah_cluster = len(set(cluster_labels)) - (1 if -1 in cluster_labels else 0)
jumlah_noise = list(cluster_labels).count(-1)

print("Jumlah cluster terbentuk:", jumlah_cluster)
print("Jumlah data noise:", jumlah_noise)

Show hidden output

115]
✓ Oz
ringkasan_cluster = df.groupby("cluster")[fitur].mean()
print(ringkasan_cluster)

Show hidden output

116]
✓ Oz
pca = PCA(n_components=2)
pca_result = pca.fit_transform(X_scaled)

df["pc1"] = pca_result[:, 0]
df["pc2"] = pca_result[:, 1]

117]
✓ Oz
plt.figure(figsize=(7,6))
sns.scatterplot(
    x="pc1",
    y="pc2",
    hue="cluster",
    palette="tab10",
    data=df,
    s=60
)
plt.title("Visualisasi Hasil Clustering Wisatawan Toraja (DBSCAN)")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.legend(title="Cluster")
plt.grid(True)
plt.show()

Show hidden output

118]
✓ Oz
mask = cluster_labels != -1
if 1 (variable) sil_score: Any > 1:
    sil_score = silhouette_score(X_scaled[mask], cluster_labels[mask])
    print("Silhouette Score:", sil_score)
else:
    print("Silhouette Score tidak dapat dihitung (cluster kurang dari 2)")

Show hidden output
Silhouette Score tidak dapat dihitung (cluster kurang dari 2)

119]
✓ Oz
df.to_excel("hasil_cluster_wisatawan_toraja.xlsx", index=False)
print("File hasil clustering berhasil disimpan.")

File hasil clustering berhasil disimpan.

```



