PRAKTEK IMAGE MINING CLUSTERING

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

MATA KULIAH : DATA MINING II

Tunjukkan proses preprocessing data MNIST, jelaskan langkah-langkah yang perlu dilakukan (tunjukkan code dan outputnya)

Preprocessing data MNIST

1. Memasukkan data MNIST

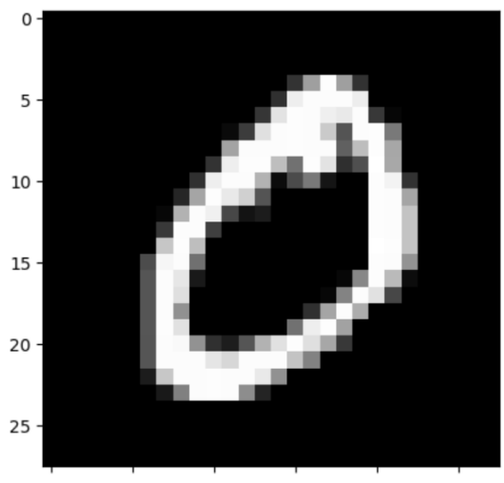
```
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
%matplotlib inline
```

```
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.mnist.load_data()

print("Training data shape:", X_train.shape, y_train.shape)
print("Test data shape:", X_test.shape, y_test.shape)
```

```
Training data shape: (60000, 28, 28) (60000,)
Test data shape: (10000, 28, 28) (10000,)
```

Data dimasukkan menggunakan kode tf.keras.datasets.mnist.load_data() yang akan mengambil dataset mnist dari library keras dalam tensorflow. Data dibagi menjadi data train dan data test. Berikut adalah salah satu contoh sampel datanya.

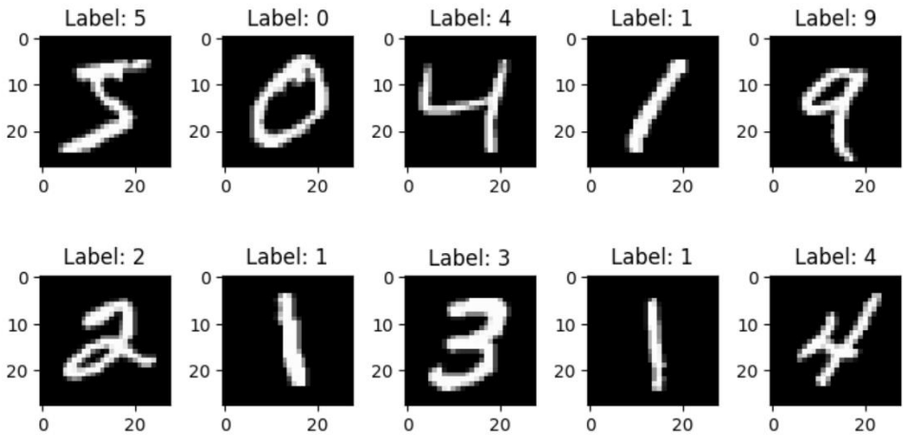
```
sample = 1
image = X_train[sample]

fig = plt.figure
plt.imshow(image, cmap='gray')
plt.show()
```



NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
MATA KULIAH : DATA MINING II

Berikut adalah 10 sampel data lainnya.

```
num = 10
images = X_train[:num]
labels = y_train[:num]

num_row = 2
num_col = 5

# plot images
fig, axes = plt.subplots(num_row, num_col, figsize=(1.5*num_col, 2*num_row))
for i in range(num):
    ax = axes[i//num_col, i%num_col]
    ax.imshow(images[i], cmap='gray')
    ax.set_title(f'Label: {labels[i]}')

plt.tight_layout()
plt.show()
```



Terlihat setiap gambar memiliki labelnya masing-masing.

2. Melakukan preprocessing dengan PCA

PCA dilakukan untuk mengubah bentuk data dari gambar 28x28 menjadi vektor 784

```
np.random.seed(48)

# Mengubah gambar 28x28 menjadi vektor 784
X_train_flattened = X_train.reshape(X_train.shape[0], -1) / 255.0
X_test_flattened = X_test.reshape(X_test.shape[0], -1) / 255.0
```

```
print("Training data shape:", X_train_flattened.shape)
print("Test data shape:", X_test_flattened.shape)
```

```
Training data shape: (60000, 784)
Test data shape: (300, 2)
```

NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
MATA KULIAH : DATA MINING II

Bentuk data diubah karena beberapa algoritma clustering membutuhkan data berbentuk vektor. Hal di atas disebut dengan ‘flattening’ yaitu mengubah image 28x28 yang berbentuk 2 dimensi menjadi suatu vektor 1 dimensi dengan 784 fitur.

3. Subset data

   Untuk clustering hierarchical dan dbscan akan dilakukan dengan menggunakan data yang disubset. Tujuannya adalah supaya pelaksanaan clustering tidak menggunakan terlalu banyak memori dan berjalan lebih cepat.

```
# Reduce the data size to avoid memory issues
subset_size = 1000
X_train_subset = X_train_pca[:subset_size]
```

Jelaskan tentang algoritma clustering yang kalian gunakan

K-Means Clustering

Algoritma kmeans dilakukan dengan menggunakan jumlah cluster sebanyak 3 yang didapat dari penghitungan silhouette score di bawah ini.

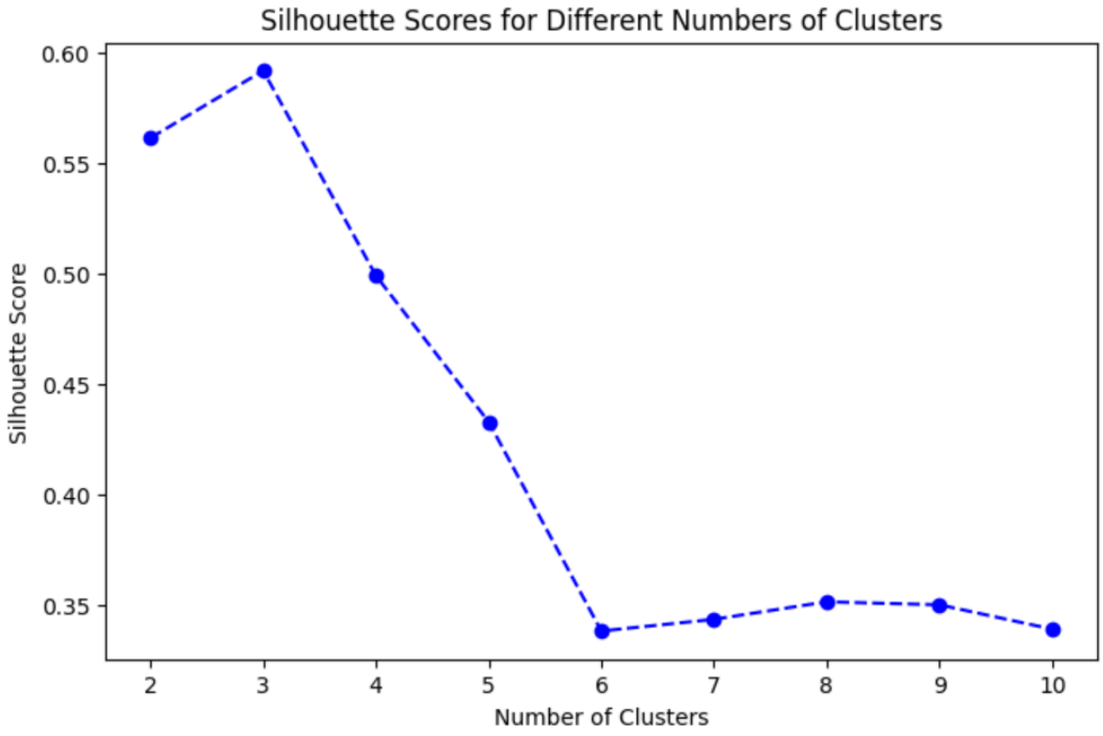
```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.metrics import silhouette_score

X_test_flattened, y_test = make_blobs(n_samples=300, cluster_std=1, random_state=12)

# Test different numbers of clusters (1 to 10)
silhouette_scores = []
num_clusters_range = range(2, 11)

for n_clusters in num_clusters_range:
    kmeans = KMeans(n_clusters=n_clusters, n_init=10, random_state=42)
    cluster_labels = kmeans.fit_predict(X_test_flattened)

    # Calculate silhouette score
    score = silhouette_score(X_test_flattened, cluster_labels)
    silhouette_scores.append(score)
    print(f"For n_clusters = {n_clusters}, the silhouette score is {score:.4f}")

# Plot the silhouette scores for each number of clusters
plt.figure(figsize=(8, 5))
plt.plot(num_clusters_range, silhouette_scores, marker='o', linestyle='--', color='b')
plt.title("Silhouette Scores for Different Numbers of Clusters")
plt.xlabel("Number of Clusters")
plt.ylabel("Silhouette Score")
plt.show()
```

PRAKTEK IMAGE MINING CLUSTERING

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

MATA KULIAH : DATA MINING II



Kemudian dilakukan kmeans clustering dengan kode di bawah.

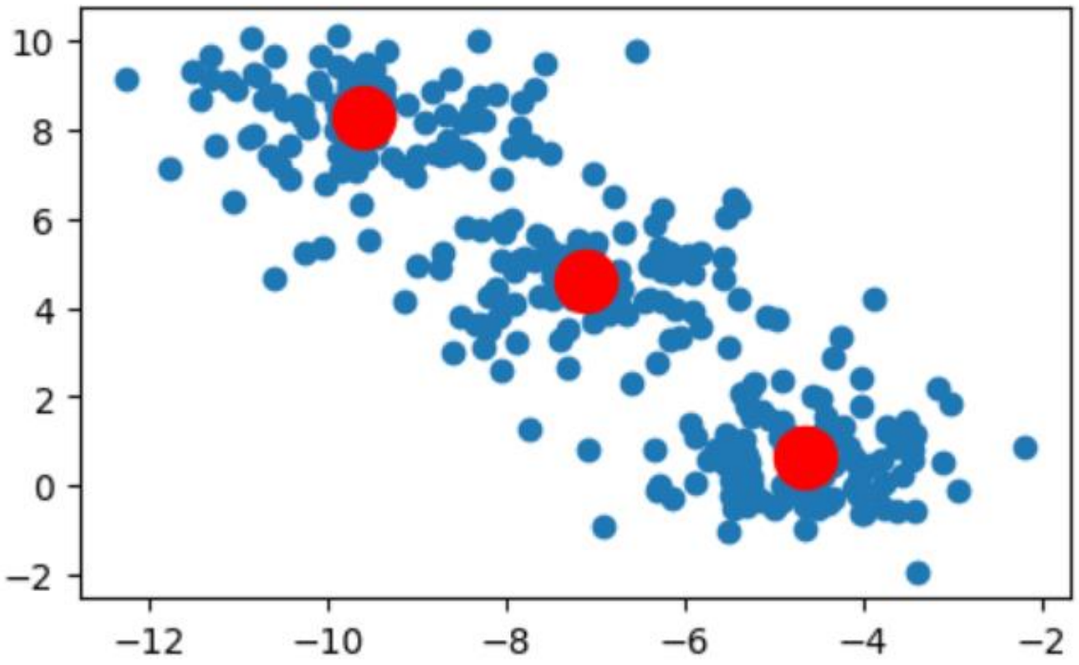
```
import matplotlib.pyplot as plt
import sklearn
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

kmeans = KMeans(n_clusters=3, n_init=10)
pred_y = kmeans.fit_predict(X_test_flattened)
X_test_flattened, y_test = make_blobs(n_samples=300, cluster_std=1, random_state=12)
plt.figure(figsize=(5,3))
plt.scatter(X_test_flattened[:, 0], X_test_flattened[:, 1])
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=300, c='red')
plt.show()
```

PRAKTEK IMAGE MINING CLUSTERING

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

MATA KULIAH : DATA MINING II



Setelah melakukan clustering akan dilakukan evaluasi dengan menggunakan Silhouette Score, Adjusted Rand Index, dan Normalized Mutual Info.

```
# Silhouette Score
from sklearn.metrics import silhouette_score

score = silhouette_score(X_test_flattened, pred_y)
print(f"Silhouette Score: {score:.4f}")
```

```
Silhouette Score: 0.5917
```

```
from sklearn.metrics import adjusted_rand_score, normalized_mutual_info_score

ari = adjusted_rand_score(y_test, pred_y)
nmi = normalized_mutual_info_score(y_test, pred_y)

print(f"Adjusted Rand Index: {ari:.4f}")
print(f"Normalized Mutual Info: {score:.4f}")
```

```
Adjusted Rand Index: 0.9605
Normalized Mutual Info: 0.5917
```

Silhouette score muncul dari rentang -1 hingga 1 di mana skor yang mendekati 1 adalah skor yang baik. Di sini hasil dari silhouette score menunjukkan nilai 0.5917 di mana nilai tersebut sudah cukup tinggi dalam pembagian clustering. Nilai Adjusted Rand Index (ARI) memiliki pembacaan

NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
MATA KULIAH : DATA MINING II

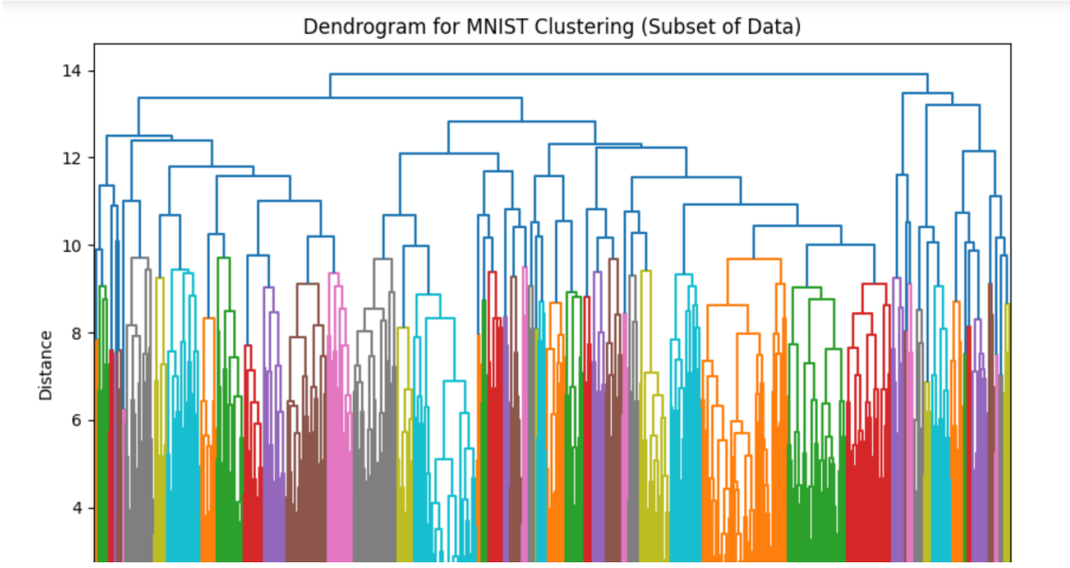
yang mirip dengan Silhouette Score di mana rentangnya adalah -1 hingga 1 dengan nilai 1 merupakan skor sempurna. Nilai yang muncul adalah 0.9605 di mana sangat mendekati 1 sehingga hasil dari clustering akurat dan mendekati label yang sebenarnya. Sedangkan nilai Normalized Mutual Information memiliki rentang nilai 0 hingga 1. Nilai 0.5917 menunjukkan nilai yang cukup dan tidak terlalu tinggi. Artinya adalah tiap cluster kemungkinan ada yang bertindihan.

Hierarchical Clustering

Untuk mencari nilai cluster yang tepat dalam hierarchical akan digunakan plot dendogram. Namun sebelumnya data akan dikecilkan atau disubset agar dendogram tidak memakan waktu lama.

```
# Reduce the data size to avoid memory issues
subset_size = 1000
X_train_subset = X_train_pca[:subset_size]
```

```
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

linked = linkage(X_train_subset, method='complete') # metode complete, average, atau ward

plt.figure(figsize=(10, 7))
dendrogram(linked)
plt.title('Dendrogram for MNIST Clustering (Subset of Data)')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```



PRAKTEK IMAGE MINING CLUSTERING

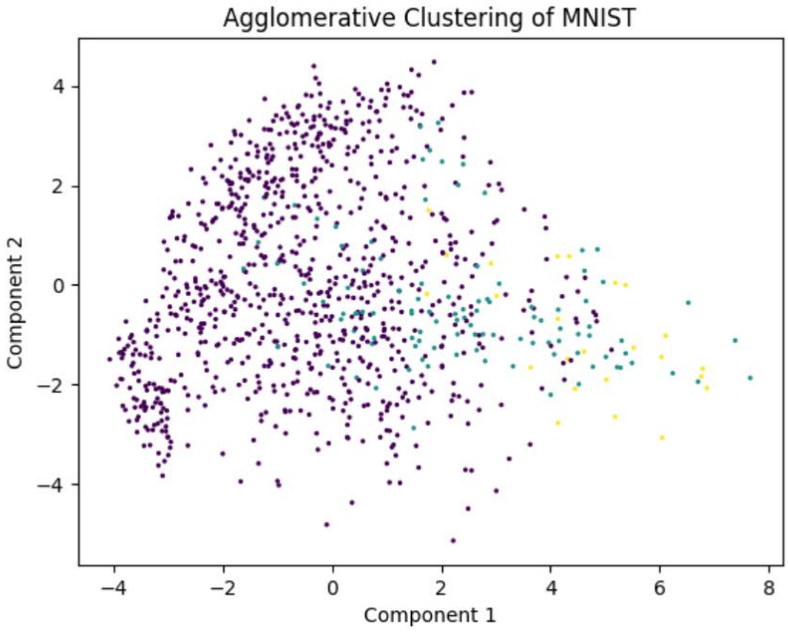
NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

MATA KULIAH : DATA MINING II

Dengan menggunakan metode linkage ‘complete’, akan diambil cluster sejumlah 3 cluster dengan cutoff di distance sekitar 13. Kemudian dilakukan clustering.

```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='complete')
y_cluster = cluster.fit_predict(X_train_subset)

plt.scatter(X_train_subset[:, 0], X_train_subset[:, 1], c=y_cluster, cmap='viridis', s=2)
plt.title('Agglomerative Clustering of MNIST (PCA Reduced Data)')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()
```



Dengan jumlah cluster 3 didapatkan nilai silhouette sebesar 0.1087. Nilai tersebut termasuk sangat-sangat kecil yang berarti clustering akan menjalankan tugas dengan buruk. Namun jika dibandingkan dengan jumlah cluster lain dan metode linkage lain, nilai ini sudah termasuk baik.

```
# Silhouette Score
from sklearn.metrics import silhouette_score

score = silhouette_score(X_train_subset, y_cluster)
print(f"Silhouette Score: {score:.4f}")
```

```
Silhouette Score: 0.1087
```

PRAKTEK IMAGE MINING CLUSTERING

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

MATA KULIAH : DATA MINING II

DBSCAN Clustering

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
import numpy as np
```

Library yang digunakan tercantum di atas. Kemudian akan dilakukan dbscan clustering. Nilai eps dan min_samples dapat diubah secara manual. Di sini saya melakukan percobaan perbandingan nilai-nilai tersebut secara manual dengan membandingkan jumlah cluster dan silhouette score yang muncul.

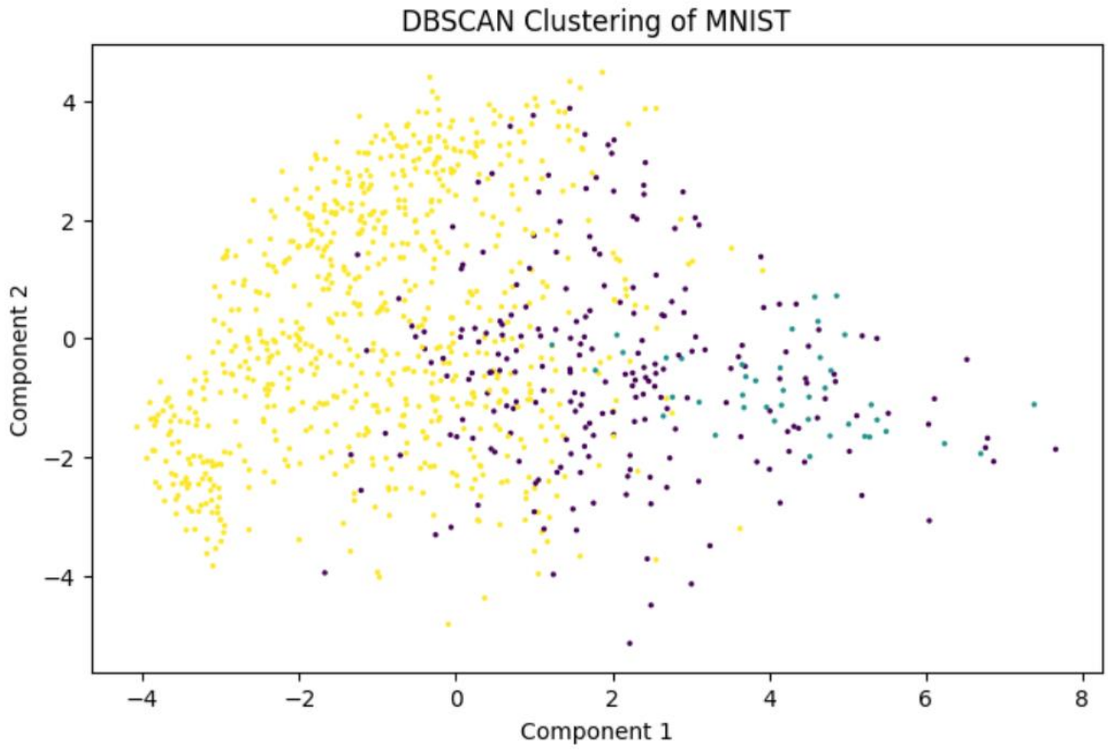
```
# Try different values of eps and min_samples
eps_values = [1,2,3,4,5,6,7,8,9,10]
min_samples_values = [5, 10, 20]

for eps in eps_values:
    for min_samples in min_samples_values:
        dbscan = DBSCAN(eps=eps, min_samples=min_samples)
        cluster_labels = dbscan.fit_predict(X_train_subset)

        # Count clusters and noise points
        n_clusters = len(set(cluster_labels)) - (1 if -1 in cluster_labels else 0)
        n_noise = list(cluster_labels).count(-1)

        if n_clusters > 1:
            print(f"eps: {eps}, min_samples: {min_samples}")
            print(f"Estimated number of clusters: {n_clusters}")
            print(f"Estimated number of noise points: {n_noise}")

            score = silhouette_score(X_train_subset, cluster_labels)
            print(f"Silhouette Score: {score:.4f}")
```

```
eps: 5, min_samples: 10
Estimated number of clusters: 5
Estimated number of noise points: 417
Silhouette Score: 0.0102
eps: 5, min_samples: 20
Estimated number of clusters: 3
Estimated number of noise points: 594
Silhouette Score: 0.0083
eps: 6, min_samples: 10
Estimated number of clusters: 2
Estimated number of noise points: 105
Silhouette Score: 0.0948
eps: 6, min_samples: 20
Estimated number of clusters: 2
Estimated number of noise points: 222
Silhouette Score: 0.0967
```

NAMA : AMALIKA ARI ANINDYA
NIM : 164221029
MATA KULIAH : DATA MINING II

Terlihat di atas bahwa nilai yang menghasilkan cluster lebih dari satu dan memiliki silhouette score tertinggi adalah eps=6 dan min_samples=20.

```
dbscan = DBSCAN(eps=6, min_samples=20)
dbscan_labels = dbscan.fit_predict(X_train_subset)

plt.figure(figsize=(8, 5))
plt.scatter(X_train_subset[:, 0], X_train_subset[:, 1], c=dbscan_labels, cmap='viridis', s=2)
plt.title('DBSCAN Clustering of MNIST')
plt.xlabel('Component 1')
plt.ylabel('Component 2')
plt.show()
```



Hasil DBSCAN menampilkan scatter plot dengan warna yang dianggap clusternya.

```
# Evaluate the clustering
n_clusters = len(set(dbscan_labels)) - (1 if -1 in dbscan_labels else 0)
n_noise = list(dbscan_labels).count(-1)

print(f"Estimated number of clusters: {n_clusters}")
print(f"Estimated number of noise points: {n_noise}")
```

```
Estimated number of clusters: 2
Estimated number of noise points: 222
```

PRAKTEK IMAGE MINING CLUSTERING

NAMA : AMALIKA ARI ANINDYA

NIM : 164221029

MATA KULIAH : DATA MINING II

Evaluasi dilakukan dengan menampilkan jumlah cluster dan noise points. Jumlah cluster yang dihasilkan adalah dua dan noise points-nya sebanyak 222. Noise points berarti titik yang tidak masuk ke dalam cluster manapun. Titik tersebut bisa jadi outliers.

```
# Silhouette Score
from sklearn.metrics import silhouette_score

score = silhouette_score(X_train_subset, dbscan_labels)
print(f"Silhouette Score: {score:.4f}")
```

```
Silhouette Score: 0.0967
```

Hasil dari silhouette score menunjukkan nilai 0.097 di mana clustering berlangsung dengan buruk namun jika dibandingkan dengan nilai epsilon dan minimum sample lain, nilai ini sudah cukup bagus.