

PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

1. Tujuan

Tujuan dari kode ini adalah untuk melakukan clustering (pengelompokan) terhadap gambar digit tulisan tangan dari dataset MNIST menggunakan algoritma K-Means. Dataset MNIST sendiri berisi 70.000 gambar digit tulisan tangan (0–9) dengan ukuran 28x28 piksel dalam skala grayscale.

Dengan menggunakan K-Means, diharapkan:

- Mampu membentuk 10 cluster yang merepresentasikan digit 0–9.
- Mengamati bagaimana bentuk centroid tiap cluster yang menyerupai “rata-rata digit”.
- Mengevaluasi kualitas cluster dengan beberapa metrik (Silhouette Score, Adjusted Rand Index, dan Normalized Mutual Information).

2. Import Library

```
1 import tensorflow as tf
2 import matplotlib.pyplot as plt
3 from sklearn.cluster import KMeans
4 import numpy as np
5 %matplotlib inline
```

- TensorFlow → digunakan untuk memanggil dataset MNIST bawaan.
- Matplotlib → visualisasi gambar digit.
- Scikit-learn (KMeans) → algoritma clustering yang digunakan.
- NumPy → manipulasi data numerik.
- %matplotlib inline → agar setiap plot ditampilkan langsung dalam notebook.

3. Load Dataset MNIST

```
1 (X_train, y_train), (X_test, y_test) = tf.keras.datasets.mnist.load_data()
2
3 print("Training data shape:", X_train.shape, y_train.shape)
4 print("Test data shape:", X_test.shape, y_test.shape)
```

- Dataset dibagi menjadi training set (60.000 gambar) dan test set (10.000 gambar).
- X_train dan X_test berisi data gambar (28x28 piksel).
- y_train dan y_test berisi label angka sebenarnya (0–9).

PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

Output akan menampilkan dimensi data:

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 0s 0us/step
Training data shape: (60000, 28, 28) (60000,)
Test data shape: (10000, 28, 28) (10000,)
```

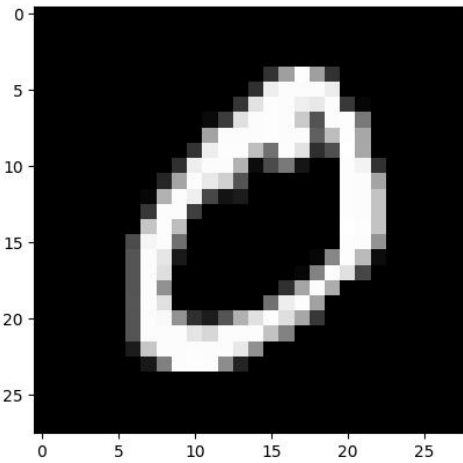
(60000, 28, 28) untuk gambar dan (60000,) untuk label.

4. Menampilkan Contoh Gambar

```
1 sample = 1
2 image = X_train[sample]
3
4 fig = plt.figure
5 plt.imshow(image, cmap='gray')
6 plt.show()
```

- Menampilkan salah satu contoh digit dari dataset.
- cmap='gray' digunakan agar gambar terlihat jelas dalam skala abu-abu.
- Tahap ini membantu memahami isi dataset secara visual.

Output:



PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

5. Mengambil Beberapa Contoh Gambar & Label

```
1 num = 10
2 images = X_train[:num]
3 labels = y_train[:num]
```

- Mengambil 10 gambar pertama beserta label aslinya.
- Tujuannya agar bisa divisualisasikan dan diperiksa apakah dataset sudah benar.

6. Visualisasi Beberapa Gambar dengan Label

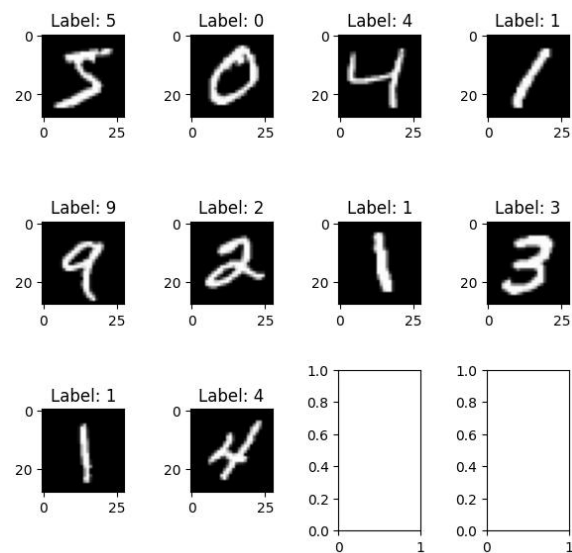
```
1 num_row = 3
2 num_col = 4
3
4 # plot images
5 fig, axes = plt.subplots(num_row, num_col, figsize=(1.5*num_col,2*num_row))
6 for i in range(num):
7     ax = axes[i//num_col, i%num_col]
8     ax.imshow(images[i], cmap='gray')
9     ax.set_title('Label: {}'.format(labels[i]))
10 plt.tight_layout()
11 plt.show()
```

- Membuat grid 3x4 untuk menampilkan gambar.
- Setiap gambar diberi judul sesuai label asli (y_train).
- Hasil plot menunjukkan digit 0–9 dalam format asli dataset.

PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

Output:



7. Preprocessing Data

```
1 x_train_flattened = X_train.reshape(X_train.shape[0], -1) / 255.0
2 x_test_flattened = X_test.reshape(X_test.shape[0], -1) / 255.0
```

- Data gambar diubah dari bentuk 28x28 piksel menjadi vektor 784 dimensi.
- Normalisasi dilakukan dengan membagi 255 sehingga nilai piksel berada di rentang [0, 1].
- Tujuan preprocessing ini:
 - Agar data lebih mudah diproses algoritma K-Means.
 - Mempercepat komputasi dan mencegah bias dari skala data besar.

8. Clustering dengan K-Means

```
1 num_clusters = 10
2
3 kmeans = KMeans(n_clusters=num_clusters, random_state=42)
4
5 kmeans.fit(x_train_flattened)
6
7 y_kmeans = kmeans.predict(x_test_flattened)
```

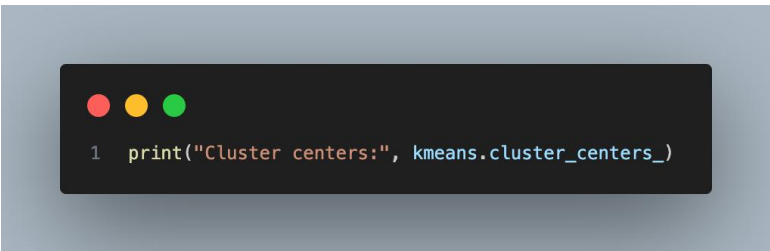
- Membuat model K-Means dengan jumlah cluster = 10 (sesuai jumlah digit).

PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

- fit(): melatih K-Means pada data training.
 - predict(): mengelompokkan data test ke dalam cluster yang sesuai.
- Tahap ini adalah inti dari proses unsupervised learning.

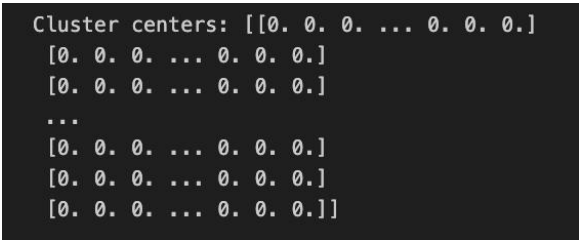
9. Menampilkan Pusat Cluster (Centroid)



```
1 print("Cluster centers:", kmeans.cluster_centers_)
```

- Setiap baris = 1 centroid (jumlahnya sama dengan num_clusters, yaitu 10).
- Setiap kolom = 1 fitur (jumlahnya 784 karena setiap gambar 28x28 di-flatten).
- Nilai-nilai di dalam array merepresentasikan rata-rata intensitas piksel untuk cluster tertentu.
- Karena terlalu panjang, NumPy secara default hanya menampilkan sebagian kecil saja dan mengganti sisanya dengan tanda ...

Output:



```
Cluster centers: [[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
```

10. Menampilkan Pusat Cluster Keseluruhan



```
1 import numpy as np
2 np.set_printoptions(threshold=np.inf)
3 print("Cluster centers:", kmeans.cluster_centers_)
```

- Sekarang seluruh isi array ditampilkan tanpa dipotong.
- Angka-angka seperti 8.39922380e-05 atau 2.28046167e-02 adalah nilai float hasil normalisasi piksel (range 0–1).
- Nilai mendekati 0.0 berarti pikselnya cenderung hitam, nilai mendekati 1.0 berarti pikselnya cenderung putih.

PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

Output:

```
Cluster centers: [[ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 8.39922380e-05  1.83914038e-04  6.91178885e-19 -2.50933511e-20
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 1.54583513e-20  5.79256814e-06  4.03307556e-04  1.00645871e-03]
```

11. Visualisasi Centroid sebagai Gambar

```
1 fig, axes = plt.subplots(1, num_clusters, figsize=(10, 3))
2 for i, ax in enumerate(axes):
3     ax.imshow(kmeans.cluster_centers_[i].reshape(28, 28), cmap='gray')
4     ax.axis('off')
5
6 plt.show()
```

- Centroid (vektor 784 dimensi) dikembalikan ke bentuk 28x28 agar bisa divisualisasikan.
- Setiap gambar centroid biasanya menyerupai digit (misalnya cluster tertentu menyerupai “0”, “1”, dsb).
- Visualisasi ini memberi gambaran interpretatif apakah clustering berhasil mendekati label asli.

Output:



12. Evaluasi dengan Silhouette Score

```
1 from sklearn.metrics import silhouette_score
2
3 # Calculate the silhouette score for the K-Means clustering on the test data
4 score = silhouette_score(x_test_flattened, y_kmeans)
5
6 print(f"Silhouette Score: {score:.4f}")
```

- Silhouette Score mengukur seberapa baik data dikelompokkan oleh algoritma clustering.
- Nilainya berkisar dari -1 sampai 1:
 - Mendekati 1 → cluster terpisah dengan baik.
 - Mendekati 0 → cluster tumpang tindih.
 - Mendekati -1 → cluster salah terklasifikasi.

PRAKTIKUM IMAGE CLUSTERING

NAMA : OKAN ATHALLAH MAREDITH
NIM : 164231088
MATA KULIAH : DATA MINING II

Output:

Silhouette Score: 0.0586

Nilai rendah, artinya cluster tidak terpisah dengan baik. Hal ini wajar karena digit tulisan tangan sangat bervariasi.

13. Evaluasi dengan Adjusted Rand Index (ARI) dan Normalized Mutual Information (NMI)

```
1 from sklearn.metrics import adjusted_rand_score, normalized_mutual_info_score
2
3 # Calculate the ARI and NMI scores
4 ari = adjusted_rand_score(y_test, y_kmeans)
5 nmi = normalized_mutual_info_score(y_test, y_kmeans)
6
7 print(f"Adjusted Rand Index: {ari:.4f}")
8 print(f"Normalized Mutual Information: {nmi:.4f}")
```

Adjusted Rand Index: 0.3667
Normalized Mutual Information: 0.4926

- Adjusted Rand Index (ARI)
 - Membandingkan hasil clustering dengan label asli, dengan koreksi terhadap kesamaan acak.
 - Nilai: 0.3667 → cluster cukup sesuai dengan label, tapi jauh dari sempurna.
- Normalized Mutual Information (NMI)
 - Mengukur keterkaitan informasi antara label asli dan hasil cluster.
 - Nilai: 0.4926 → ada korelasi sedang antara hasil clustering dan label asli.