

1. Jelaskan cara kerja dari algoritma tersebut!

Jawab:

K-Means adalah algoritma clustering yang bertujuan membagi data ke dalam beberapa kelompok (clusters) berdasarkan kesamaan fitur. Algoritma ini bekerja dengan cara iteratif untuk meminimalkan jarak antara titik data dan pusat cluster (centroids). Berikut langkah kerjanya:

- Inisialisasi centroid dengan memilih secara acak k titik sebagai centroid awal dari k cluster.
- Untuk setiap titik data, hitung jarak antara titik tersebut dan setiap centroid. Titik akan ditugaskan ke centroid terdekatnya.
- Setelah semua titik ditugaskan, hitung ulang centroid dari setiap cluster sebagai rata-rata dari semua titik yang ditugaskan ke cluster tersebut.
- Ulangi langkah 2 dan 3 sampai centroid tidak berubah lagi atau batas maksimal iterasi tercapai.
- Untuk K-Means++, centroid awal dipilih sedemikian rupa agar lebih menyebar dengan baik.

4. Bandingkan hasil evaluasi model pada nomor 2 dan 3, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

Jawab:

Menggunakan Implementasi K-Means dari Scratch

```
import importlib
import kmeans
importlib.reload(kmeans)

from kmeans import KMeansFromScratch

kmeans_scratch = KMeansFromScratch(n_clusters=2, max_iter=100, init='kmeans++')
kmeans_scratch.fit(X)

scratch_labels = kmeans_scratch.labels

print("Centroids (Scratch):", kmeans_scratch.centroids, sep="\n")
print("\nLabels (Scratch):", scratch_labels, sep="\n")

# Silhouette Score
score = silhouette_score(X, kmeans_scratch.labels)
print("\nSilhouette Score (Scratch):", score)

✓ 0.0s
```

```
Centroids (Scratch):
[[1.27028455e+01 2.54455285e+00 2.33910569e+00 2.04081301e+01
 2.06211382e+00 3.92682927e-01 1.45406504e+00 4.85138211e+00
 9.08617886e-01 2.40821138e+00 5.65869919e+02]
 [1.36665455e+01 1.87072727e+00 2.42781818e+00 1.74527273e+01
 2.81618182e+00 2.92909091e-01 1.89690909e+00 5.52036364e+00
 1.06665455e+00 3.06672727e+00 1.15172727e+03]]

Labels (Scratch):
[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1
 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 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 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

Silhouette Score (Scratch): 0.6600722211123818

Menggunakan Implementasi K-Means dari Scikit-Learn

```
kmeans_sklearn = KMeans(n_clusters=2, init='k-means++', max_iter=100, random_state=42)
kmeans_sklearn.fit(X)

sklearn_labels = kmeans_sklearn.labels_
print("Centroids (Scikit-learn):", kmeans_sklearn.cluster_centers_, sep="\n")
print("\nLabels (Scikit-learn):", sklearn_labels, sep="\n")

# Silhouette Score
score = silhouette_score(X, sklearn_labels)
print("\nSilhouette Score (Scikit-learn):", score)
```

✓ 0.0s

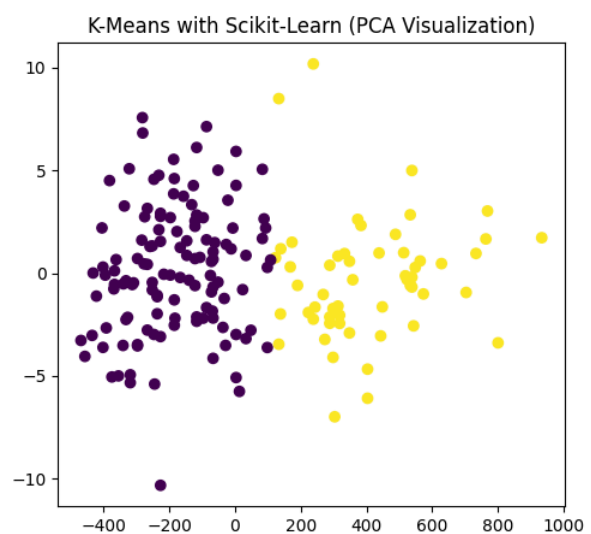
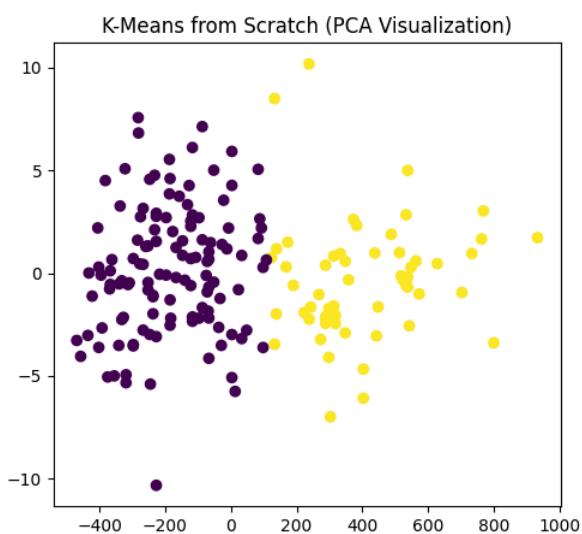
Centroids (Scikit-learn):

```
[[1.27028455e+01 2.54455285e+00 2.33910569e+00 2.04081301e+01
 2.06211382e+00 3.92682927e-01 1.45406504e+00 4.85138211e+00
 9.08617886e-01 2.40821138e+00 5.65869919e+02]
[1.36665455e+01 1.87072727e+00 2.42781818e+00 1.74527273e+01
 2.81618182e+00 2.92909091e-01 1.89690909e+00 5.52036364e+00
 1.06665455e+00 3.06672727e+00 1.15172727e+03]]
```

Labels (Scikit-learn):

```
[1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 1 1 1 1 1 1 1
 1 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 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 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

Silhouette Score (Scikit-learn): 0.6600722211123818



Tidak ada perbedaan dalam hasil evaluasi antara implementasi dari scratch dan dari Scikit-Learn.