1. Cara kerja algoritma K-means dari scratch pada repository ini

a. Inisialisasi parameter dan metode inisialisasi
   - Algoritma dimulai dengan menginisialisasi jumlah cluster (n_clusters), jumlah iterasi maksimum (max_iter), dan metode inisialisasi centroid (init_method), yang dapat berupa **random** atau **kmeans++**.
   - Jika init_method diatur ke **random**, centroid awal dipilih secara acak dari titik data. Jika **kmeans++**, centroid pertama dipilih secara acak dan centroid berikutnya dipilih berdasarkan probabilitas proporsional terhadap jarak kuadrat terjauh dari centroid yang sudah dipilih.

b. Inisialisasi centroid
   - Dalam metode *Random Initialization*, centroid dipilih secara acak dari dataset dengan menggunakan permutasi acak indeks data.
   - Dalam metode *K-Means++ Initialization*, algoritma memilih centroid pertama secara acak dan selanjutnya memilih titik data baru dengan probabilitas yang lebih tinggi jika jauh dari centroid yang sudah ada.

c. Menghitung jarak antar titik data dan centroid
   Setelah centroid awal ditentukan, algoritma menghitung jarak antara setiap titik data dengan setiap centroid menggunakan jarak Euclidean. Jarak Euclidean dihitung sebagai :

$$\text{Euclidean Distance} = \sqrt{\sum (x_i - c_i)^2}$$

   dengan x_i adalah titik data dan c_i adalah centroid.

d. Menetapkan titik data ke cluster terdekat

   Setiap titik data kemudian ditetapkan ke cluster terdekat berdasarkan jarak yang telah dihitung. Proses ini dilakukan dengan memilih centroid yang memiliki jarak terpendek ke setiap titik data.

e. Menghitung ulang centroid

   Setelah semua titik data ditetapkan ke cluster, centroid baru dihitung sebagai rata-rata dari semua titik data yang termasuk dalam cluster tersebut. Proses ini menggeser posisi centroid lebih dekat ke titik data dalam cluster masing-masing.

f. Iterasi Hingga Konvergensi atau Mencapai Batas Iterasi

   Langkah 3 hingga 5 diulang hingga tidak ada perubahan signifikan dalam posisi centroid (konvergensi tercapai) atau hingga jumlah iterasi maksimum tercapai. Konvergensi

tercapai ketika centroid baru yang dihitung hampir sama atau identik dengan centroid sebelumnya.

g. Output Hasil Clustering

Algoritma menghasilkan label klaster untuk setiap titik data yang menunjukkan ke klaster mana setiap titik termasuk. Selain itu, posisi centroid terakhir juga disimpan sebagai bagian dari model.

4. Perbandingan hasil evaluasi model

Berdasarkan hasil evaluasi, model KMeansScratch dan KMeans_Sklearn menghasilkan hasil clustering yang sangat mirip. Kedua model menunjukkan visualisasi cluster yang hampir identik dengan posisi centroid akhir yang berdekatan. Silhouette Score untuk KMeansScratch adalah 0.2241, sementara untuk KMeans_Sklearn adalah 0.2235.

Perbedaan kecil dalam hasil ini mungkin disebabkan oleh perbedaan dalam implementasi algoritma K-Means itu sendiri. Implementasi internal di pustaka scikit-learn mungkin memiliki optimisasi tambahan atau penanganan numerik yang lebih efisien dibandingkan implementasi dari scratch. Selain itu, cara penanganan kondisi terminasi dalam model scikit-learn bisa lebih canggih, yang dapat mempengaruhi hasil akhir clustering secara halus.