CLUSTERING

Mulia Sulistiyono, M.Kom
muliasulistiyono@amikom.ac.id

Learning Objective

In this course you will:

A. Melakukan pemilahan data untuk digunakan sebagai input feature

B. Mempelajari algortima clustering K-Means, Hierachical Clustering, DBSCAN

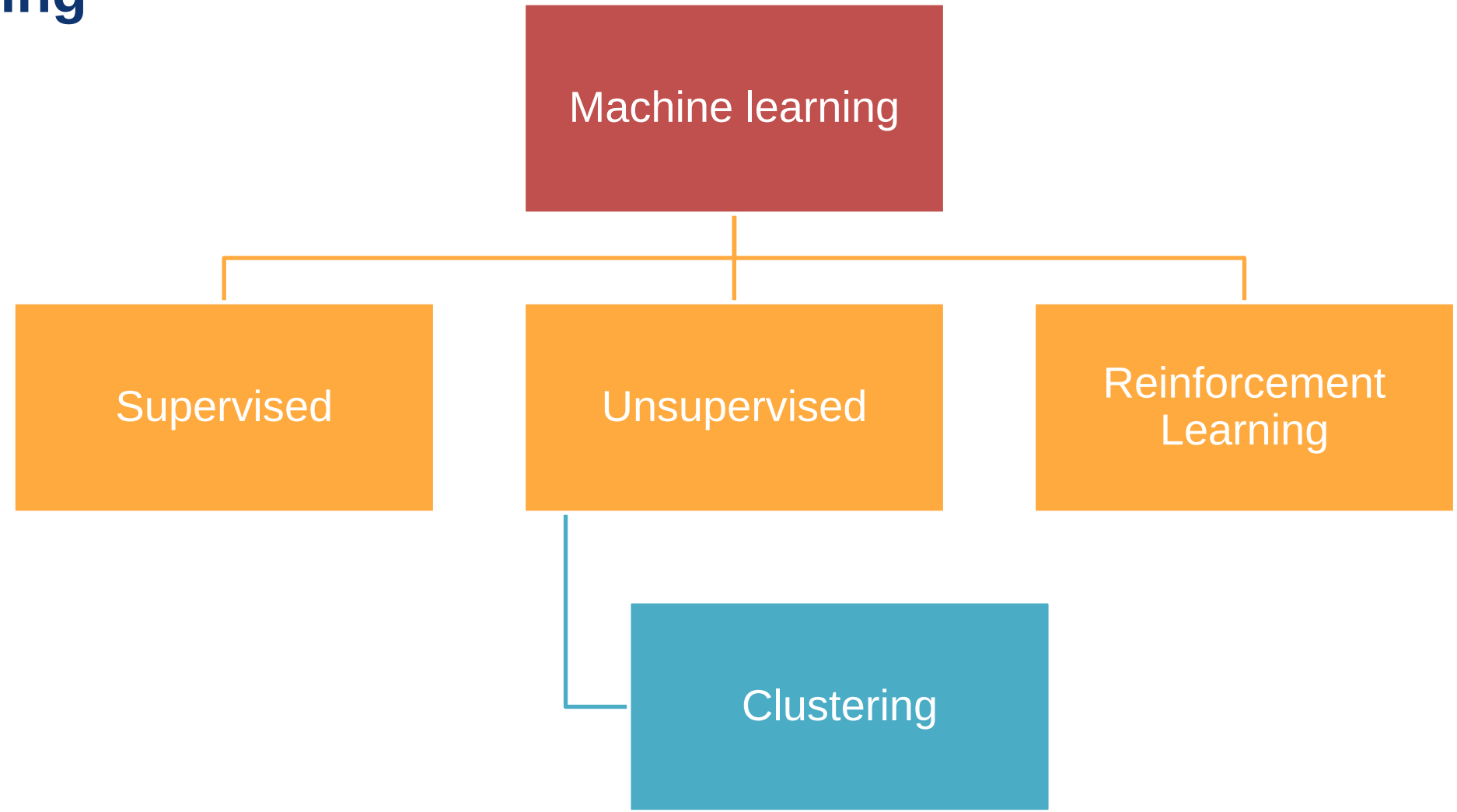
C. Mempelajari optimasi K-Means dengan metode Elbow untuk menentukan jumlah K yang tepat

D. Mempelajari optimasi Hierachical Clustering dengan Dendogram Diagram untuk menentukan jumlah K yang tepat

E. Mempelajari optimasi clustering untuk density data menggunakan metode DBSCAN

F. Praktek coding menggunakan python

Clustering


Machine learning
Supervised
Unsupervised
Reinforcement Learning
Clustering

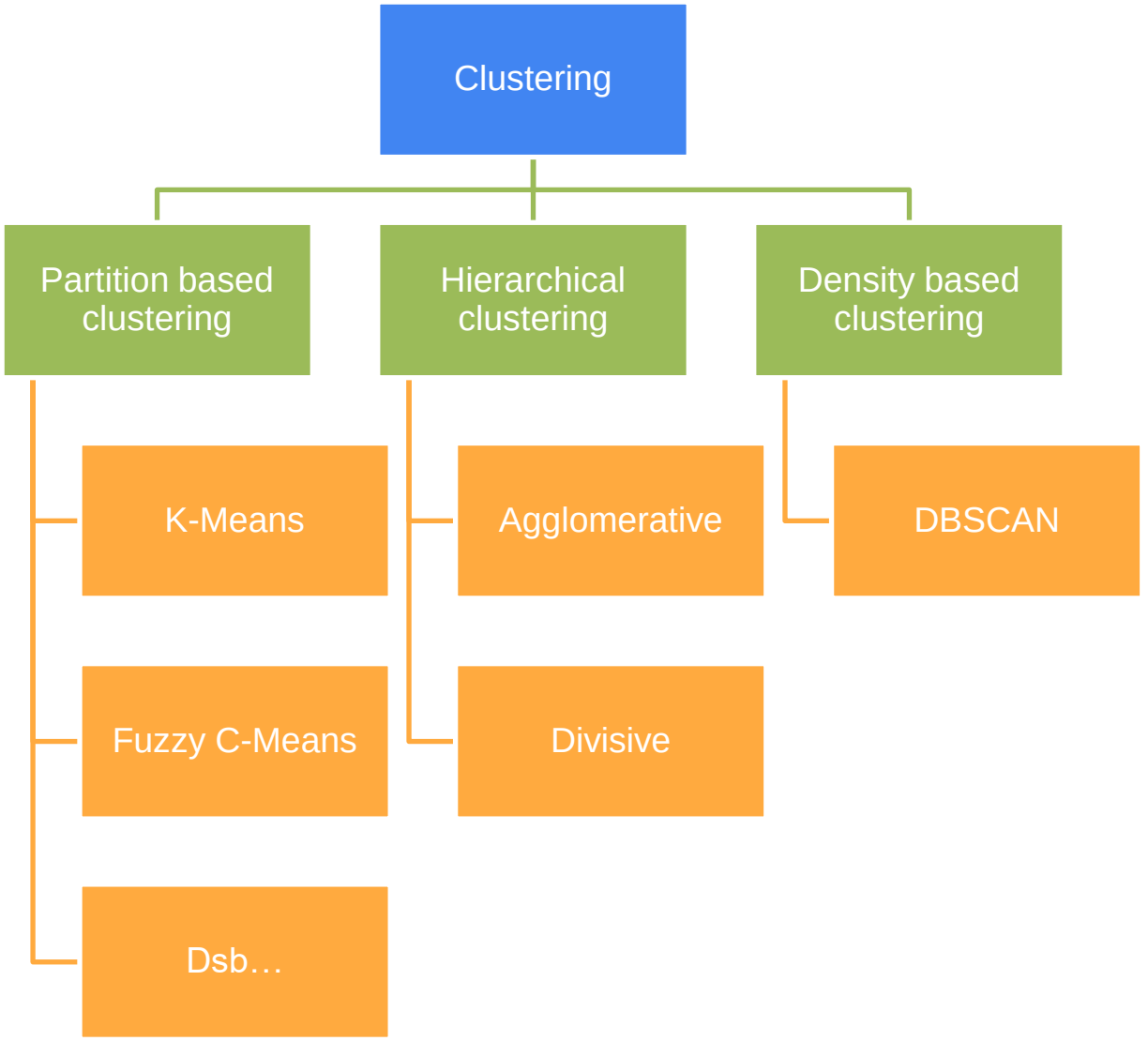
Clustering

Teknik clustering melakukan pembelajaran mesin tanpa ada supervisi untuk menyelesaikan suatu masalah

Clustering juga dapat dianalogikan sebagai tugas mengidentifikasi subkelompok dalam data sedemikian rupa sehingga titik data dalam subkelompok yang sama (cluster) sangat mirip sedangkan titik data dalam cluster yang berbeda sangat berbeda

Keputusan tentang ukuran kemiripan dapat ditentukan melalui jumlah centroid masing-masing kelompok dan jaraknya.

Clustering


Clustering
Partition based clustering
Hierarchical clustering
Density based clustering
K-Means
Fuzzy C-Means
Dsb…
Agglomerative
Divisive
DBSCAN

K-Means

Algoritma Kmeans adalah salah satu algoritma *clustering* yang bersifat iteratif yang mencoba untuk mempartisi dataset menjadi subkelompok non-overlapping berbeda yang ditentukan oleh K (cluster) di mana setiap titik data hanya dimiliki oleh satu kelompok.

K-Means mencoba membuat titik data intra-cluster semirip mungkin sambil dengan titik data yang lain pada satu cluster.

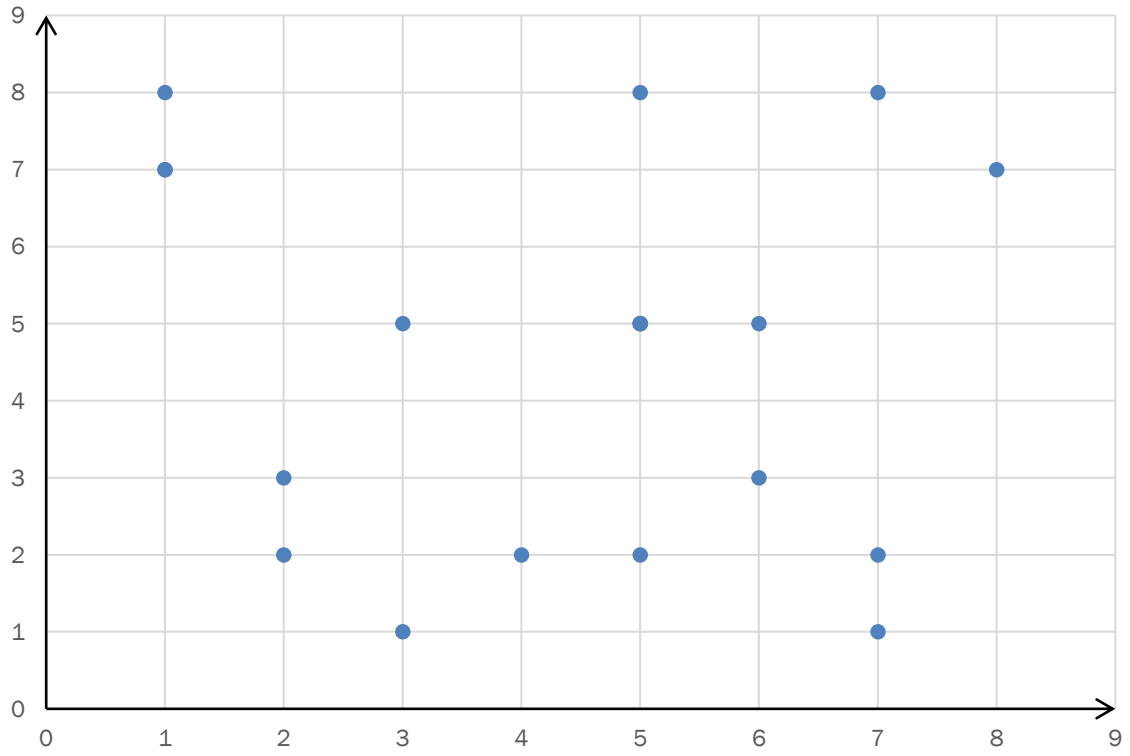
K-Means menetapkan poin data ke cluster sedemikian rupa sehingga jumlah jarak kuadrat antara titik data dan pusat massa cluster (rata-rata aritmatika dari semua titik data yang termasuk dalam cluster itu) minimal.

Semakin sedikit variasi yang kita miliki dalam cluster, semakin homogen (serupa) titik data dalam cluster yang sama.

Langkah-langkah metode K-Means

1. Memilih jumlah *cluster* awal (K) yang ingin dibuat

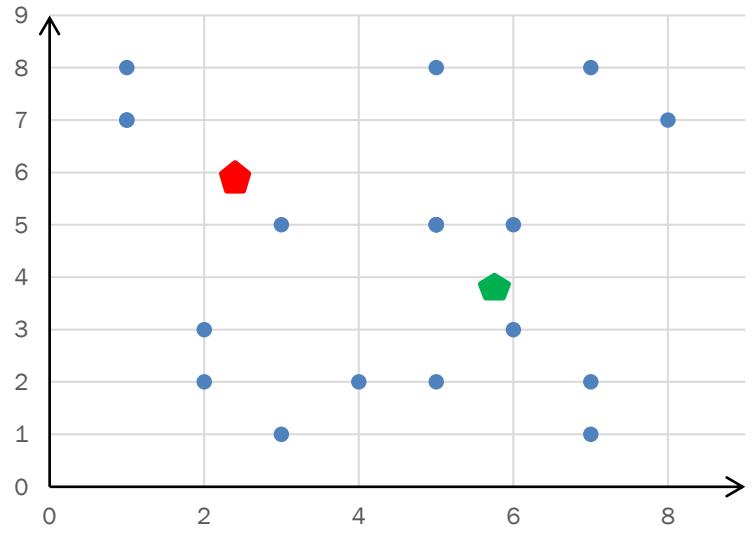
Sebagai contoh terdapat data 2 dimensi seperti yang ditampilkan dalam grafik, langkah pertama adalah memilih jumlah kluster. Misal kita pilih untuk membaginya ke dalam 2 kluster.



Langkah-langkah metode K-Means

2. Memilih titik secara random sebanyak K buah, di mana titik ini akan menjadi pusat (*centroid*) dari masing-masing kelompok (*clusters*).

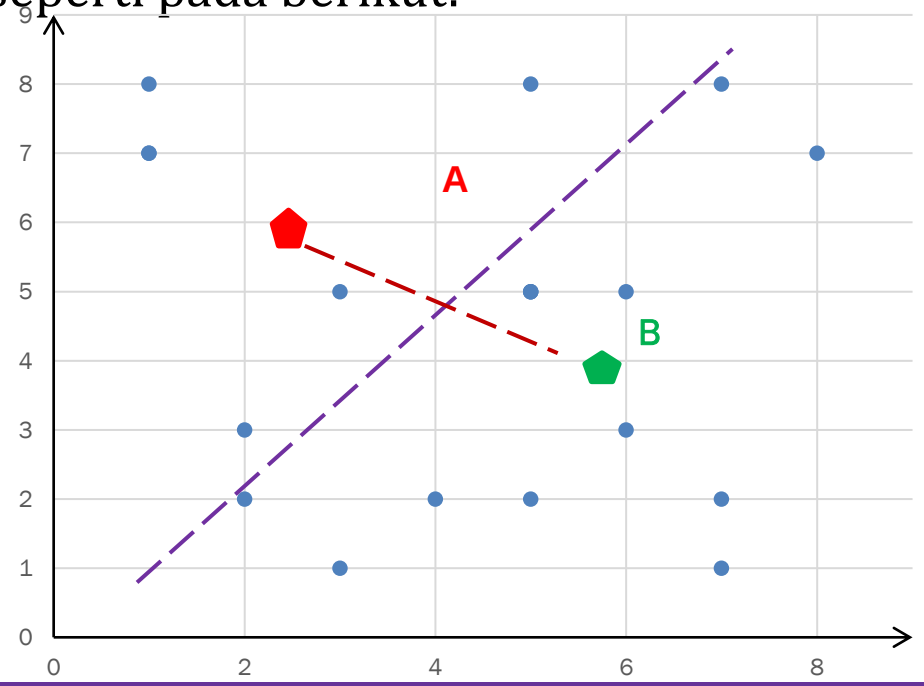
Langkah kedua adalah menentukan titik pusatnya. Dalam satu klaster terdapat satu titik pusat atau yang disebut dengan *centroid*. Penentuan awal posisi titik pusat ini bebas, karena nantinya algoritma K-Means akan merubah posisi tiap titik hingga dicapai solusi paling optimal. Pada Gambar 2, titik merah mewakili pusat dari kluster 1, dan biru untuk kluster 2. Dengan demikian, maka masing-masing data point akan memilih titik pusat (*centroid*) yang paling dekat. Jika sudah dipilih maka data point tersebut akan menjadi bagian dari klusternya



Langkah-langkah metode K-Means

3. Dari dataset yang kita miliki, buat dataset yang terdekat dengan titik *centroid* sebagai bagian dari *cluster* tersebut. Sehingga secara total akan terbentuk *clusters* sebanyak K buah.

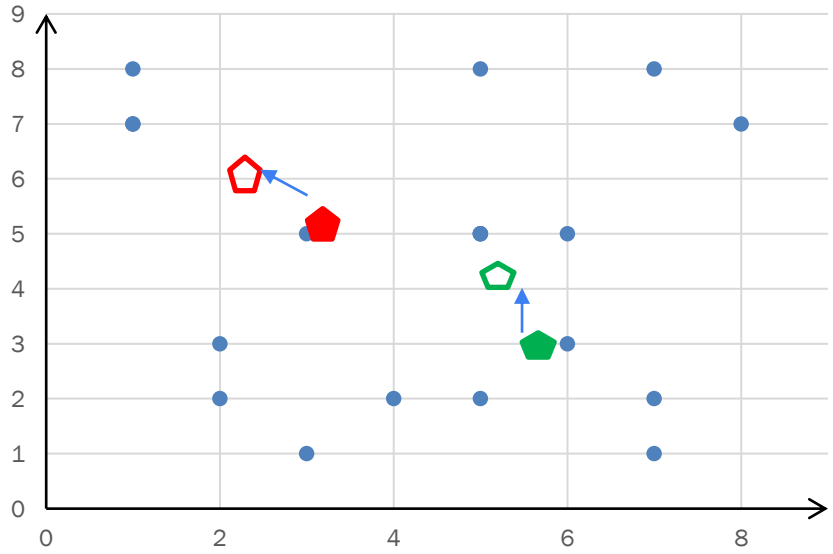
Berdasarkan pengelompokan pada langkah ke dua, maka setiap titik data saat telah tergabung dalam salah satu kluster. Titik data yang diwakili dengan symbol ‘x’ berwarna merah masuk ke kluster 1, dan symbol ‘x’ berwarna biru masuk ke kluster 2, seperti pada berikut:



Langkah-langkah metode K-Means

4. Lakukan kalkulasi, dan tempatkan pusat *centroid* yang baru untuk setiap *cluster*-nya. Langkah ini dilakukan untuk menemukan centroid yang paling tepat untuk maisng-masing klaster.

Langkah ke-empat adalah melakukan penghitungan sesuai algoritma K-Means, yaitu mencari posisi titik pusat yang paling sesuai untuk sestiap klasternya berdasarkan penghitungan jarak terdekat. Penghitungan jarak masing-masing titik data ke pusat klaster dapat menggunakan metode Euclidean distance, ilustrasi penghitungan jarak dapat dilihat pada Gambar 4



Langkah-langkah metode K-Means

5. Dari dataset yang kita miliki ambil titik *centroid* terdekat, sehingga dataset tadi menjadi bagian dari *cluster* tersebut. Jika masih ada data yang berubah kelompok (pindah *cluster*), kembali ke langkah 4. Jika tidak, maka *cluster* yang terbentuk sudah baik.

Langkah terakhir dari algoritma K-Means adalah melakukan pengecekan pada titik pusat yang telah ditentukan sebelumnya. Pilih titik pusat terdekat, dan masuk ke dalam kluster tersebut. Jika masih ada perpindahan kluster, kembali ke langkah 4. Algoritma K-Means akan terus mencari titik pusatnya, sampai pembagian datasetnya optimum dan posisi titik pusat tidak berubah lagi

Optimasi K-Means

Dari pembahasan di atas, dapat dianalisa bahwa salah satu faktor krusial baik tidaknya metode ini adalah saat menentukan jumlah klusternya (nilai K). Karena hasil pengemlompokan akan menghasilkan analisa yang berbeda untuk jumlah klaster yang berbeda juga.

Jika terlalu sedikit K (misal 2), maka pembagian kluster menjadi cepat, namun mungkin ada informasi tersembunyi yang tidak terungkap.

Jika K=8, maka terlalu banyak kluster. Mungkin akan terlalu sulit untuk membuat analisa atau memilih dukungan keputusan dari hasil cluster.

Optimasi K-Means

Untuk mengatasi ini, maka dapat ditambahkan fungsi optimasi yang akan memilih jumlah awal kluster secara tepat. kita gunakan di sesi latihan dan sebuah metode *elbow* yang akan membantu kita untuk memilih nilai K yang tepat dengan menggunakan *metric*WCSS (*Within Cluster Sum of Squares*), contoh penghitungan untuk tiga klaster:

$$wcss = \sum_{P_i \text{ in cluster } 1} jarak\ (P_i,\ C_1)^2 + \sum_{P_i \text{ in cluster } 2} jarak\ (P_i,\ C_2)^2$$

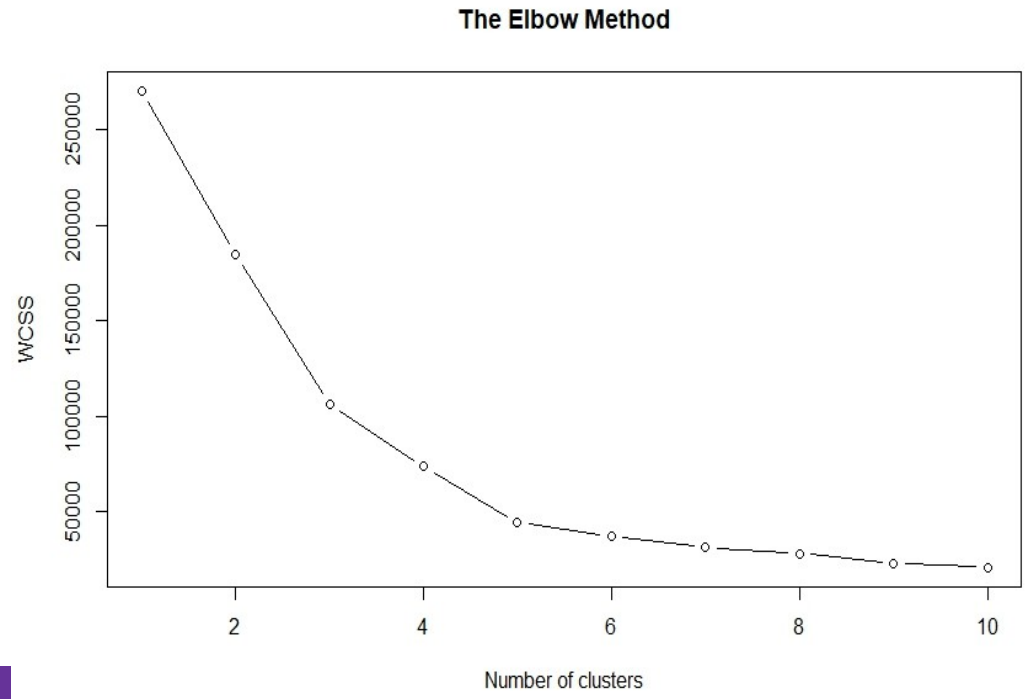
Optimasi K-Means

Pada metric di atas dapat diamati bahwa WCSS sebagai variabel dependennya. Kemudian ada simbol Sigma (seperti E), yang menyatakan jumlah kuadrat dari jarak tiap titik Pi yang ada pada kluster 1. *Sum of Squares* (jumlah kuadrat) adalah menjumlahkan hasil kuadrat dari masing-masing jarak. Selanjutnya hasil penjumlahan kluster 1 ditambah dengan hasil kudrat jarak untuk tiap data poin terhadap titik pusat kluster dua, dan seterusnya sesuai jumlah kluster yang kita inginkan.

Untuk mengetahui jumlah klaster yang paling baik untuk studi kasus yang diuji coba adalah dengan cara melihat perbandingan WCSS untuk 2 kluster, 3 kluster, 4 dan seterusnya. Yang kita pilih adalah ketika perubahan nilai WCSS nya sangat signifikan, seperti sebuah siku (elbow). Oleh karena itu cara pemilihan ini disebut dengan *elbow method.*

Optimasi K-Means

Grafik perhitungan WCSS untuk sebuah contoh dataset. Semakin kecil skor WCSS, semakin baik. Sumbu x adalah jumlah kluster, sumbu y adalah skor WCSS. Bisa dilihat bahwa saat K=1, nilai WCSS sangat tinggi. Kemudian menurun terus sampai K=5 terlihat membentuk seperti sebuah siku. Mulai K=6 sampai K=10 penurunan skor WCSS sudah tidak signifikan. Dengan demikian, dapat diketahui bahwa jumlah kluster yang tepat untuk grafik di atas adalah 5. Contoh hasil perhitungan WCSS dapat dilihat pada grafik berikut:



Studi Kasus K-Means

Pada studi kasus ini, kita akan mengaplikasikan metode K-Means ini untuk sebuah permasalahan nyata. Misalnya, seorang *data scientist* diminta untuk menganalisis data pelanggan toko. Data tersebut adalah data *member* atau pelanggan dimana hasil yang diinginkan sebuah analisa kecenderungan pembelian dari suatu kelompok pelanggan sehingga dapat memperkuat hubungan mereka terhadap konsumen. Misal untuk penguatan marketing, strategi penawaran yang tepat, dan sebagainya.

Studi Kasus K-Means

Contoh coding menggunakan Bahasa python:

```
# Mengimpor library
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Mengimpor dataset
dataset = pd.read_csv('customer.csv')
X = dataset.iloc[:, [4, 5]].values

# Optimasi K-Means dengan metode elbow untuk menentukan jumlah klaster yang
tepat
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Cluster Number')
plt.ylabel('WCSS')
plt.show()
```

Studi Kasus K-Means

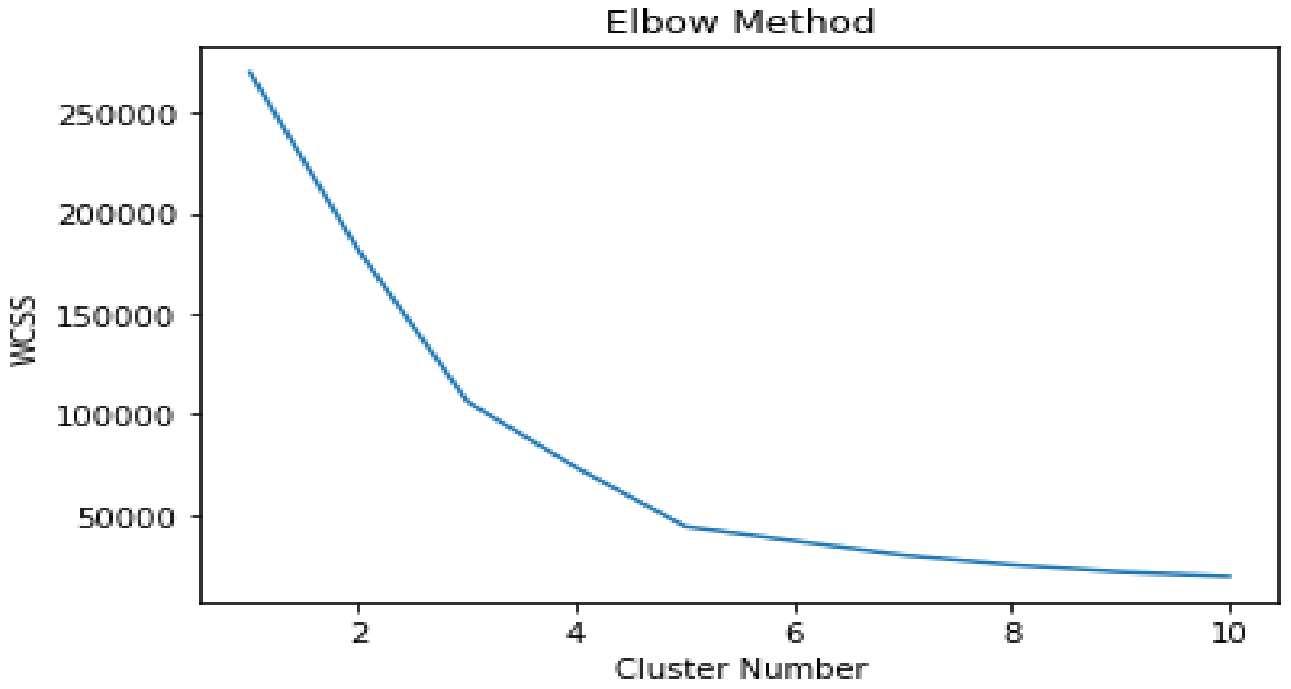
Contoh coding menggunakan Bahasa python 2:

```
# Proses K-Means Clustering
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)

# Visualisasi hasil clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'blue', label
= 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'red', label
= 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'magenta',
label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label
= 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'green',
label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s =
300, c = 'yellow', label = 'Centroids')
plt.title('Consumers Cluster')
plt.xlabel('Yearly Salary')
plt.ylabel('Yearly expense rating (1-100)')
plt.legend()
```

Studi Kasus K-Means

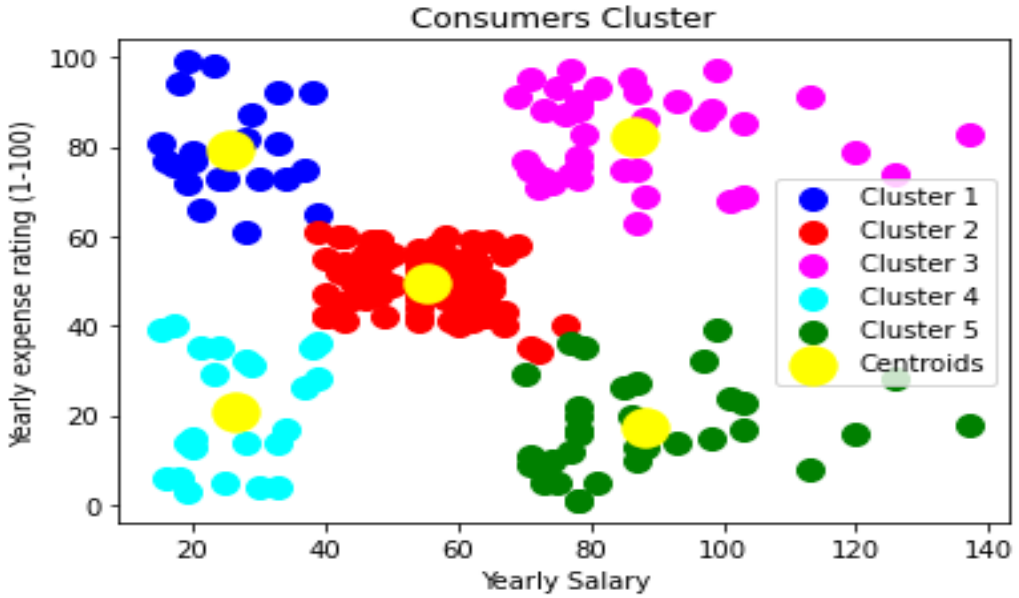
Hasil pengelompokan data menggunakan metode K-Means yang dioptimasi dengan metrics WCSS:



Pada Gambar di atas, dapat diamati bahwa garis grafik yang berbentuk siku terdapat pada klaster 3 dan 5, maka dapat disimpulkan bahwa pembagian klaster sebanyak 3 atau 5 adalah jumlah yang paling optimum

Studi Kasus K-Means

Hasil pengelompokan data menggunakan metode K-Means yang dioptimasi dengan metrics WCSS:



Pada dua gambar di atas dapat diamati hasil pengelompokan data set yang terbagi dalam 3 dan 5 klaster. Hasil dari pengelompokan tersebut dapat digunakan sebagai dukungan keputusan untuk menentukan startegi yang dituju oleh pemilik toko terhadap pelangganya

TUGAS PEKAN DEPAN PRESENTASI MODEL MACHINE LEARNING
(BISA AMBIL DARI PAPER)

KLASIFIKASI :
KELOMPOK 1 : Support Vector Machine (SVM)
KELOMPOK 2 : Decision Tree
KELOMPOK 3 : Naïve Bayes Classifier

REGRESI :
KELOMPOK 4 : Support Vector Regression
KELOMPOK 5 : Random Forest Regression

CLUSTERING :
KELOMPOK 6 : Hierachical
KELOMPOK 7 : DBSCAN

YANG HARUS DIPRESENTASIKAN
1. DEFINISI
2. LANGKAH-LANGKAH PENYELESAIAN ALGORITMA BESERTA CONTOH/STUDI KASUS DAN PENJELASANNYA
3. PARAMETER EVALUASI DAN PENJELASANNYA (PERFORMA MODEL/ALGORITMA SESUAI TASK-NYA)

Terima Kasih