Session 35

Unsupervised Learning

DigitalSkola
Uncover The World Of Digital Skills With Us

Table of Content

What will We Learn Today?

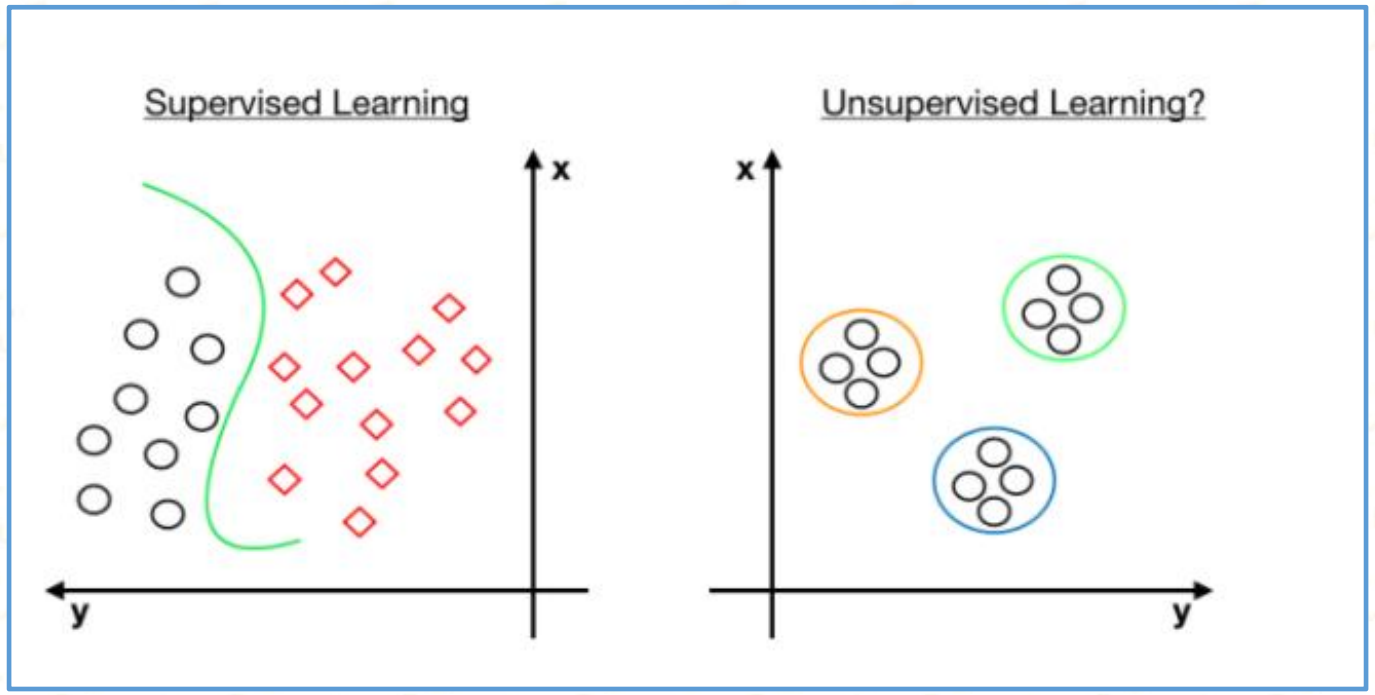
1. Unsupervised Learning
2. K-Means
3. K-Medoids
4. DBSCAN

DigitalSkola
Uncover The World Of Digital Skills With Us

Unsupervised Learning

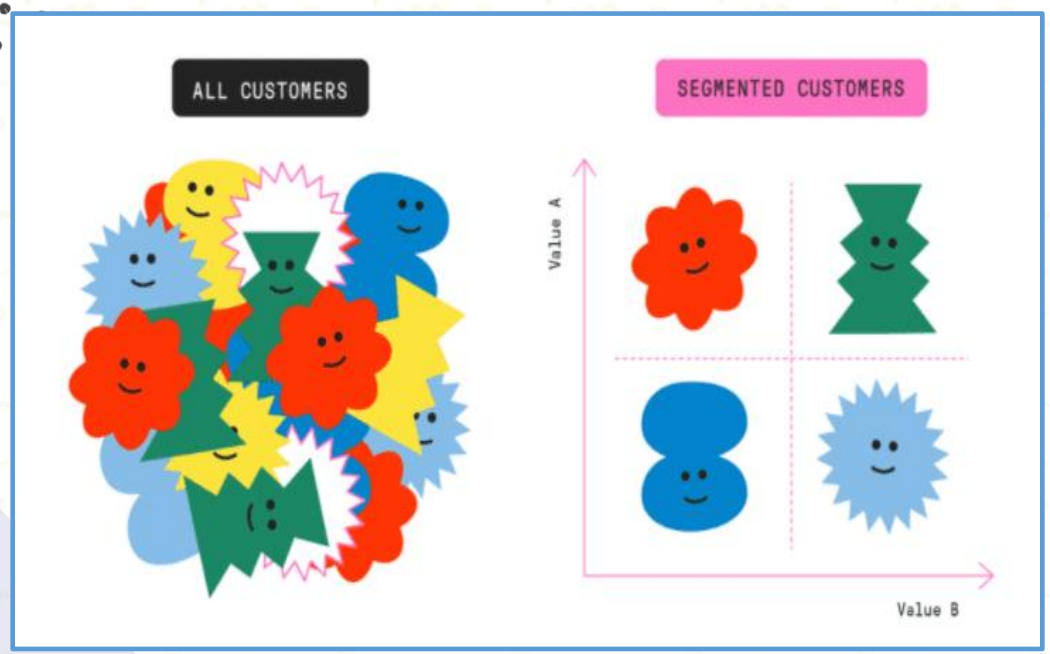
Supervised vs Unsupervised

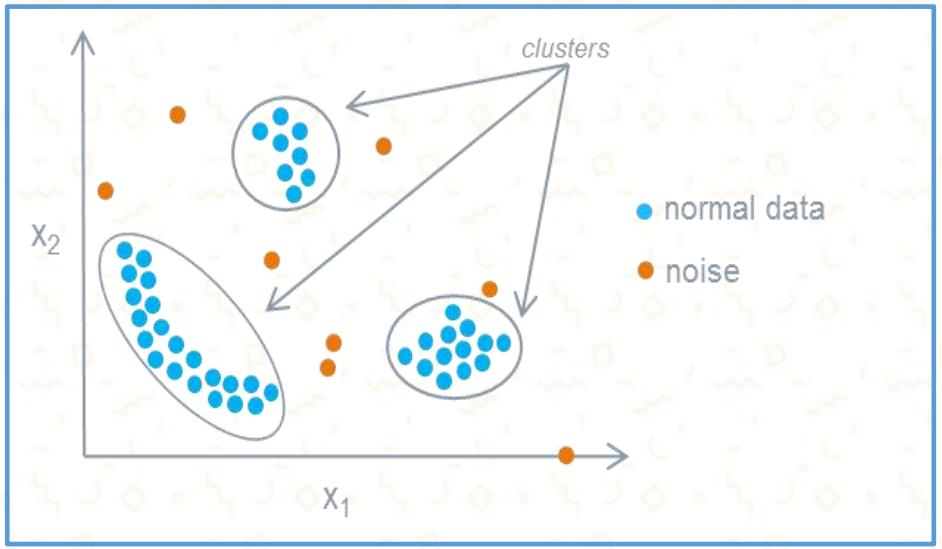
- **Supervised** = Mempelajari untuk memprediksi hasil.
  - Kita tahu target label-nya, sehingga kita membuat model yang bisa memprediksi label.
- **Unsupervised** = Mencari pola atau characteristic dari data.
  - Kita tidak tahu target label-nya, sehingga kita membuat model yang bisa mengelompokkan data.



Application of Unsupervised Learning

- Customer segmentation.
  - Memahami kelompok pelanggan yang berbeda untuk membangun pemasaran atau strategi bisnis lainnya.
- Anomaly detection.
- Recommender systems, yang mencoba mengelompokkan user dengan pola yang sama untuk memberi recomendasi content yang mirip.



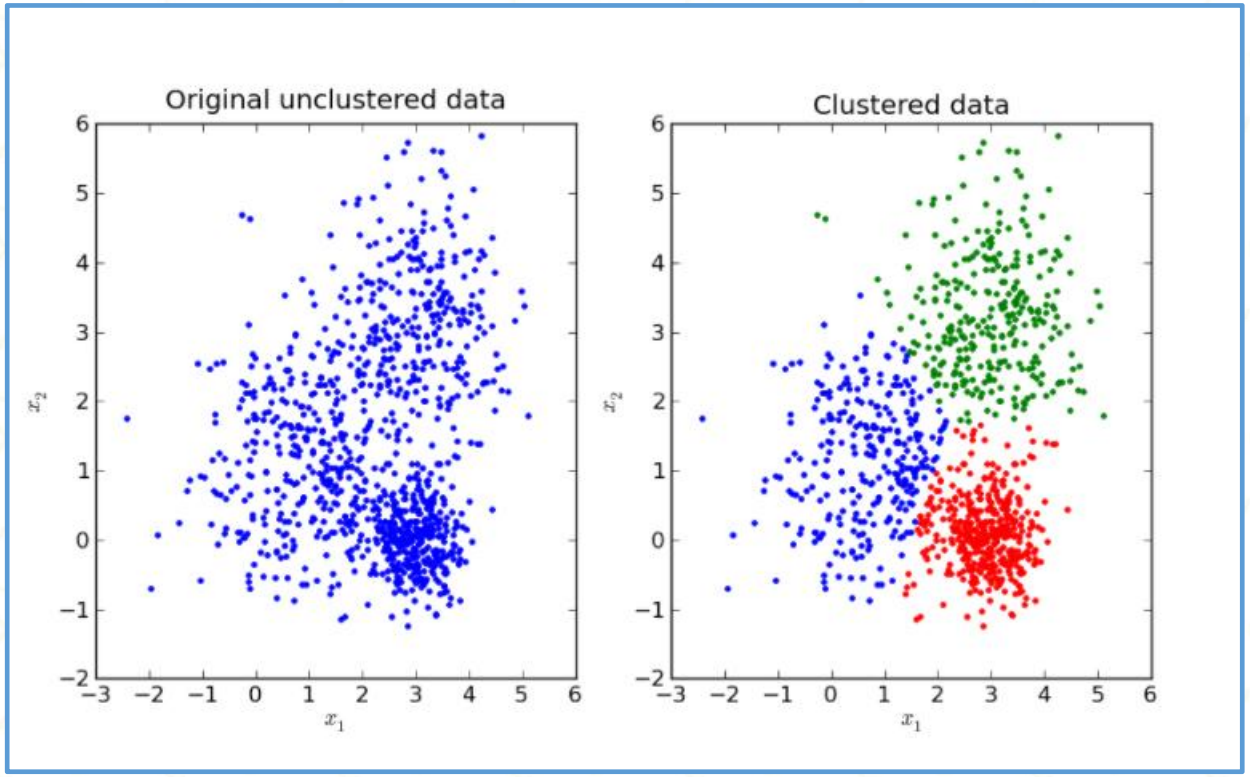


DigitalSkola
Uncover The World Of Digital Skills With Us

K-Means

K-Means

- K-means clustering algorithm mencoba mengelompokkan item ke dalam bentuk cluster.
- Jumlah kelompok direpresentasikan sebagai *K*.

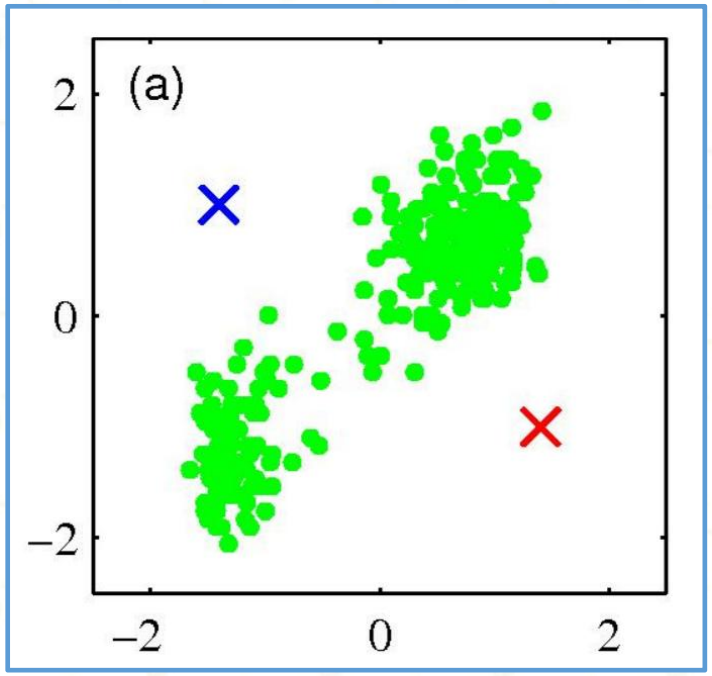


How K-Means works?

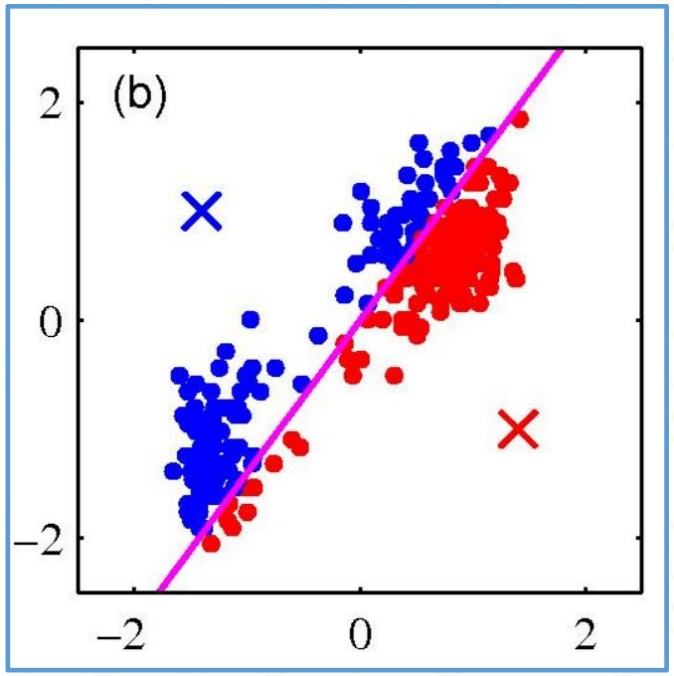
1. **Memilih object sejumlah k** sebagai awalan titik tengah cluster
2. **Menetapkan setiap object** ke dalam cluster dengan titik tengah terdekat
3. **Mengupdate titik tengah cluster** sebagai nilai rata rata dari point yang ada dalam cluster
4. Kembali lagi ke Step 2, berhenti ketika tidak ada perubahan

How K-Means works?

- Mimilih titik acak sejumlah K sebagai titik tengah (means)
  - Disini jumlah K=2

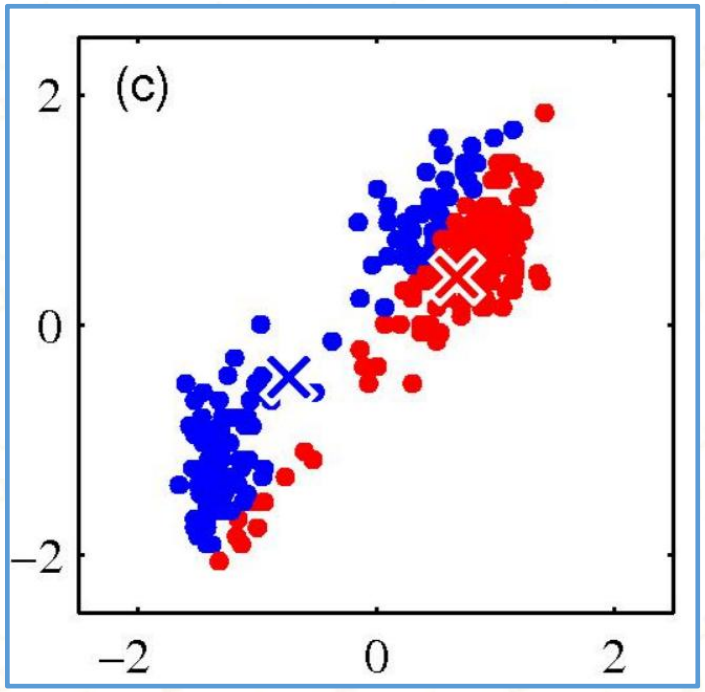


- Iterative Step 1
  - Menetapkan data point ke titik tengah cluster terdekat

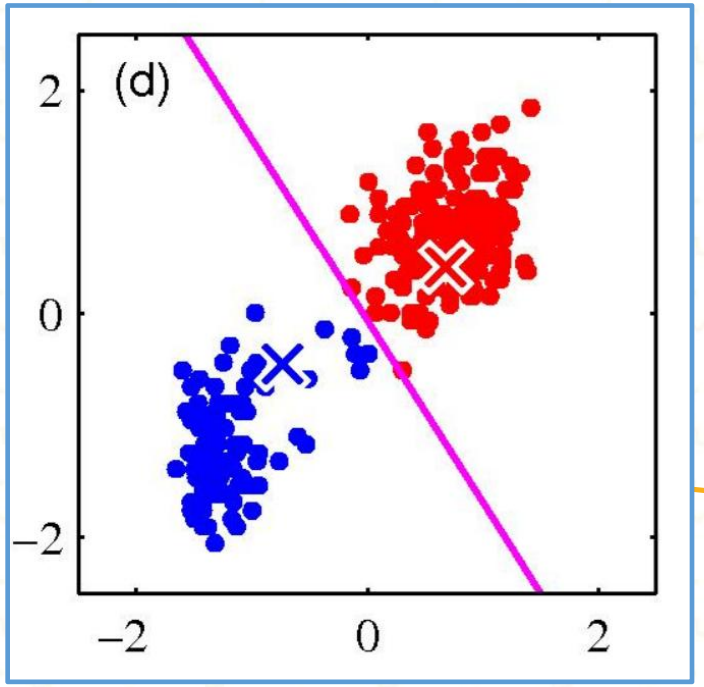


How K-Means works?

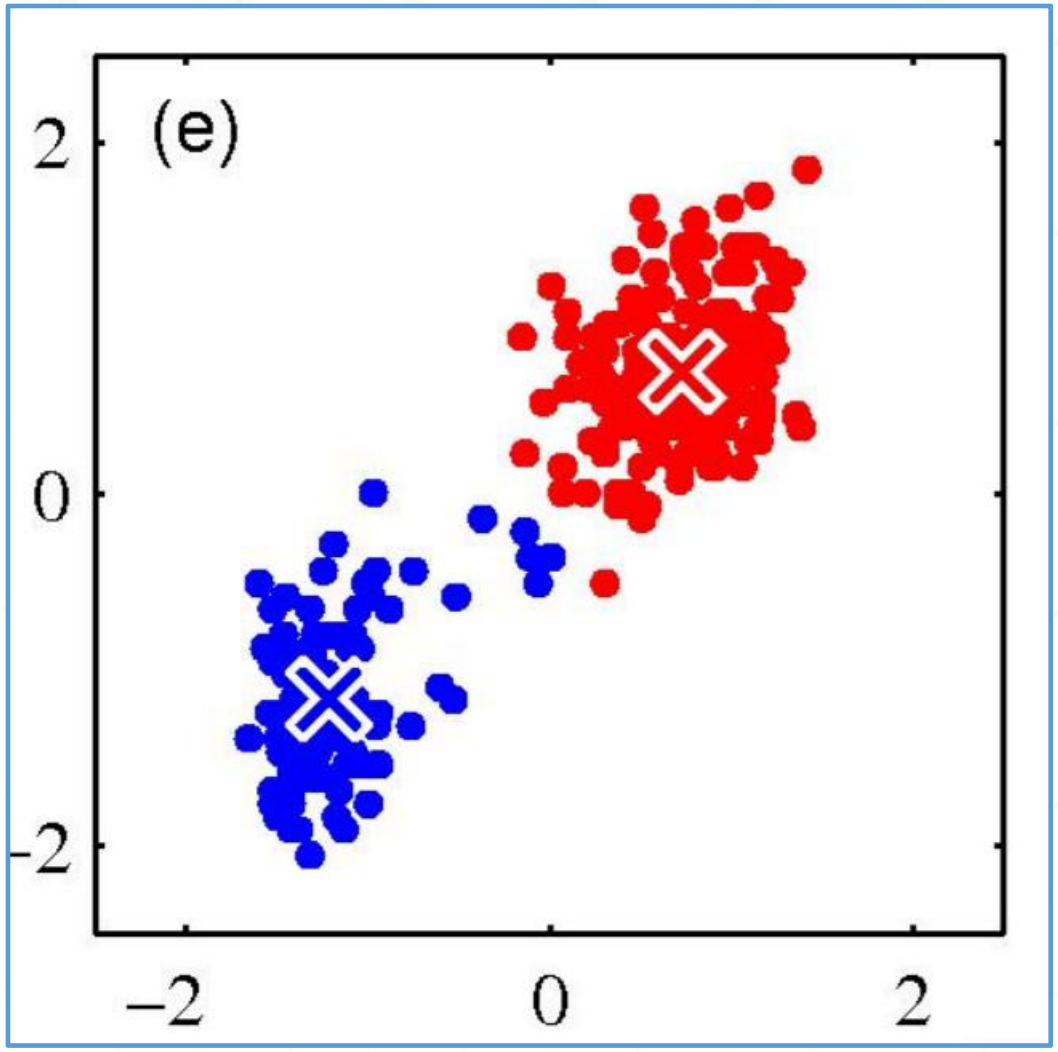
- Iterative Step 2
  - Mengupdate titik tengah
  - Ganti titik tengah menjadi rata rata dari poin yang dalam cluster



- Ulangi proses ini sampai “convergence”



How K-Means works?



Evaluating clustering performance

1. **Inertia**
   - Jumlah kuadrat (Sum of squared distance) jarak dari setiap titik (xi) dengan cluster-nya (Ck).
   - Jika inertia kecil, artinya poin dekat satu sama lain.

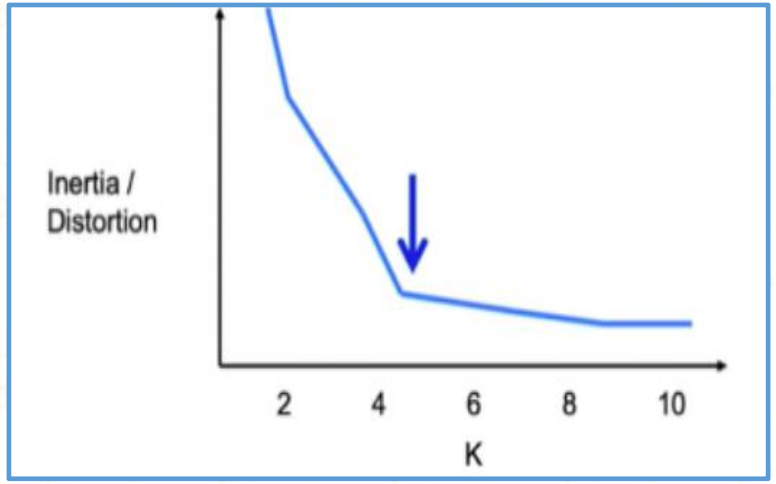
$$\sum_{i=1}^{n}(x_i - C_k)^2$$

2. **Silhoutte score**
   - a : jarak rata-rata ke semua titik lain dalam clusternya.
   - b : jarak rata-rata ke semua titik lain di cluster terdekat berikutnya.
   - Score diatara -1 to 1. Bagus jika score mendekati 1.

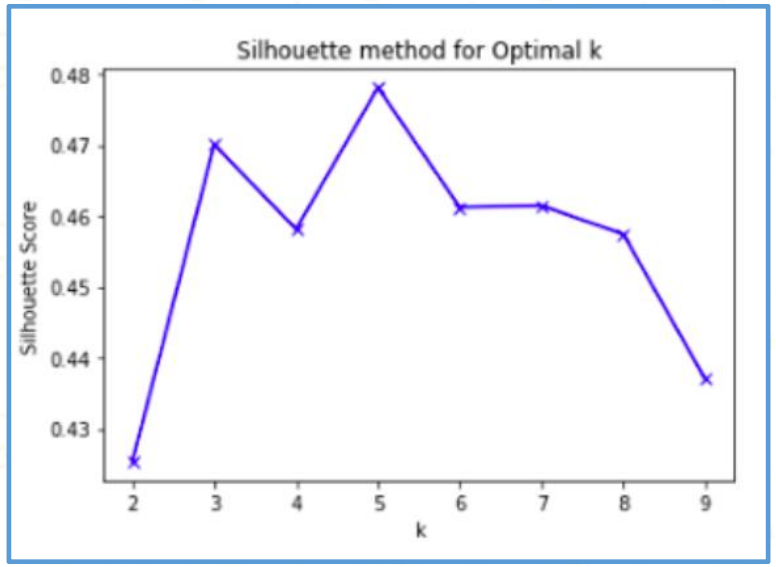
$$SC = \frac{b-a}{\max(a,b)}$$

How to choose the K?

1. **Elbow method**



2. **Silhoutte score**
   - Semakin tinggi nilai, semakin baik



Discussion on the K-means

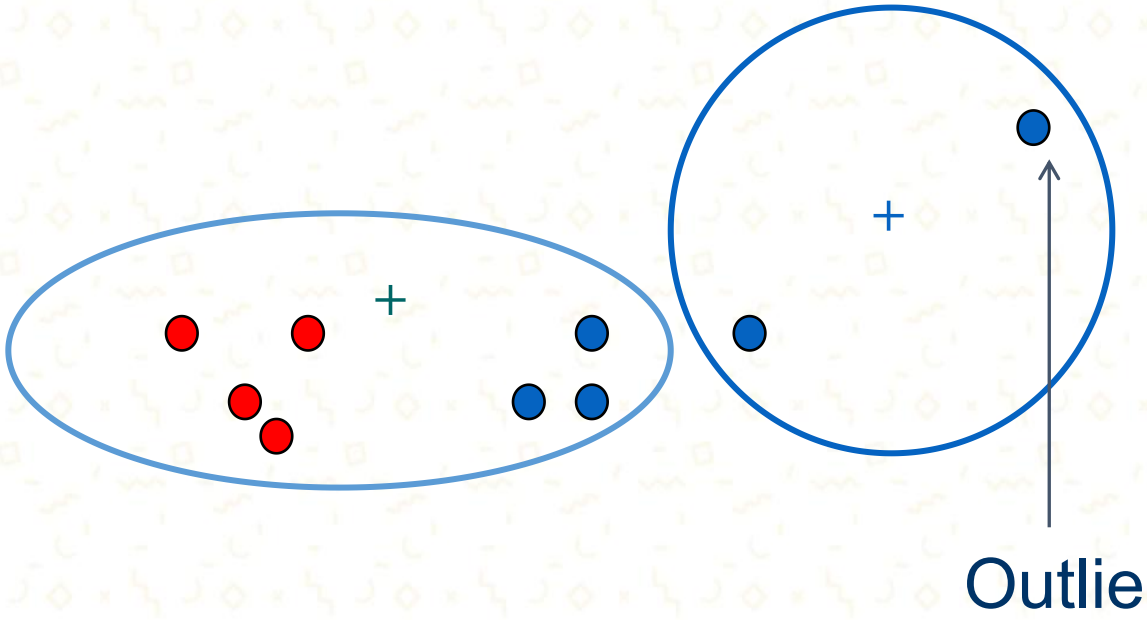
- **Keuntungan K-means**
  - Mudah untuk di-implementasikan
  - Mampu menangani dataset yang besar
  - Generalisasi cluster untuk berbagai bentuk dan ukuran.

- **Kerugian K-means**
  - Sensitive terhadap outliers.
  - Memilih nilai k butuh usaha yang lebih.
  - Ketika jumlah dimensi meningkat, skalabilitasnya menurun.

https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/

A Problem of K-Means

- Sensitive terhadap outliers
- Outlier: objek dengan nilai yang sangat besar (atau kecil)

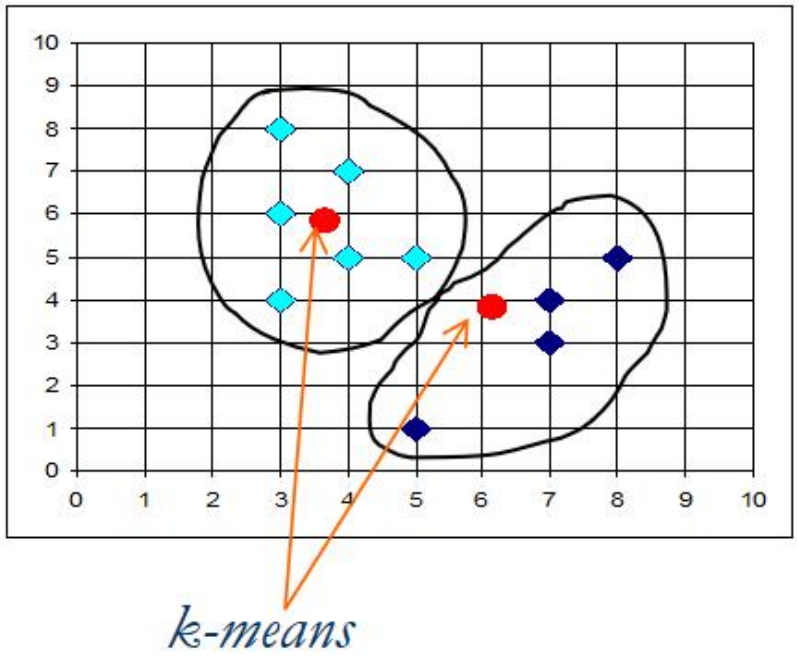


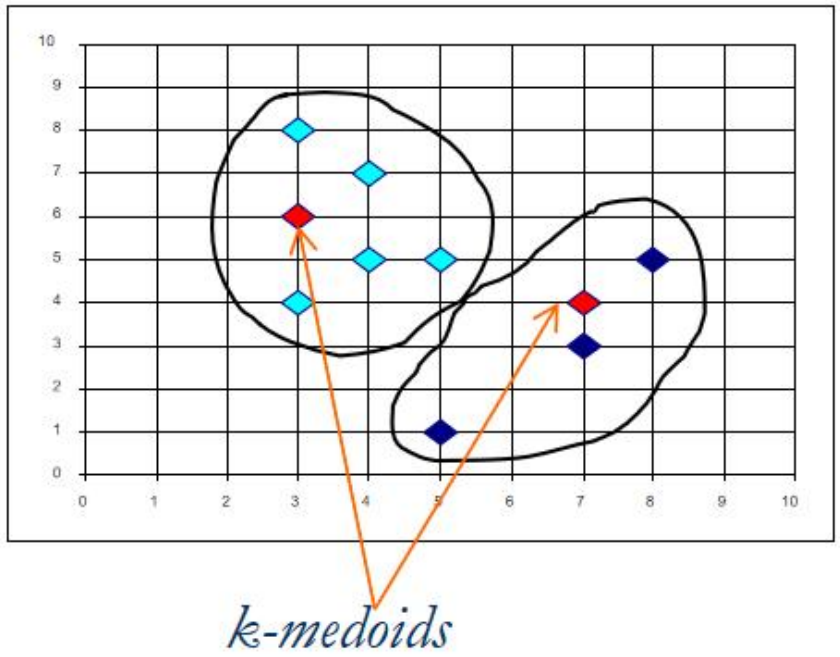
DigitalSkola
Uncover The World Of Digital Skills With Us

K-Medoids

K-Medoids

- K-medoids: Memilih representatif object sebanyak k, yang disebut sebagai medoids.
  - K-Means meminimalkan jumlah kuadrat (sum-of-squares) dalam cluster
  - K-Medoids meminimalkan jumlah jarak (sum of distances) antara setiap titik dengan medoid dari cluster nya.





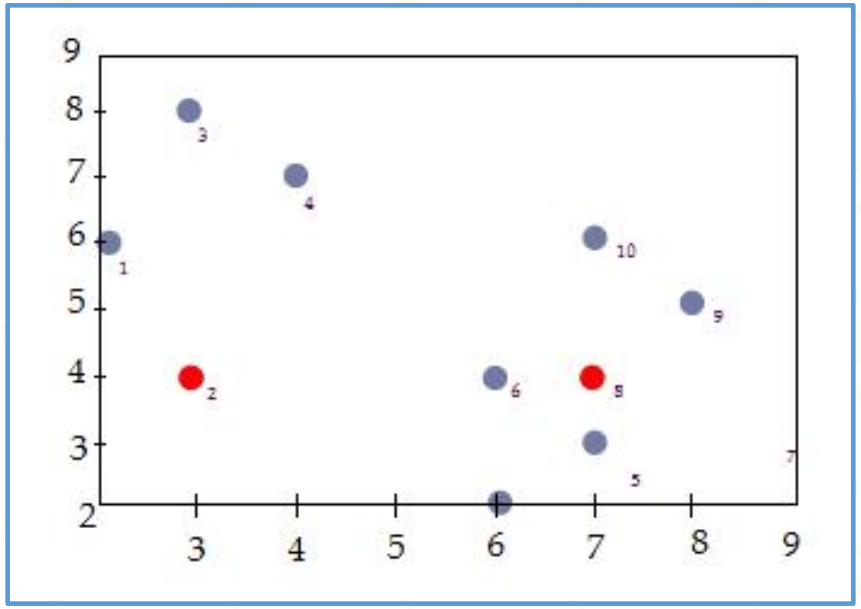
How K-Medoids (PAM) works?

- Partitioning Around Medoids (PAM)

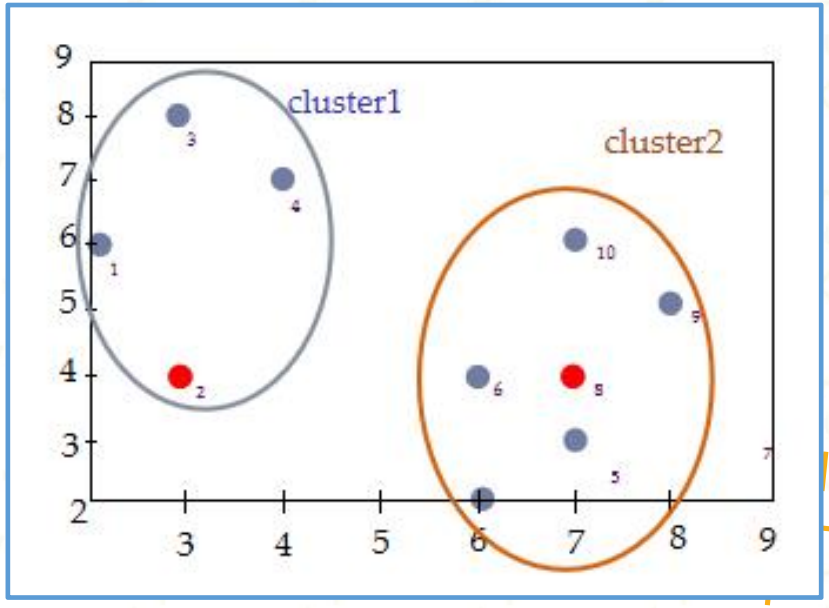
- 1. Initialize: pilih titik acak sebanyak k, yang kita sebut sebagai medoids.
- 2. Repeat:
  - Tetapkan setiap titik ke cluster dengan jarak terdekat dengan medoid m.
  - Memilih objek baru oi dalam cluster
  - Hitung “total cost of swapping S”, antara medoid m dengan oi
  - If S < 0:
    - Ganti m dengan oi untuk membentuk kumpulan medoids.
- Ulangi proses ini dan berhenti ketika “convergence”.

How K-Medoids works?

- Pilih medoid secara acak sebanyak K
- Contoh K=2

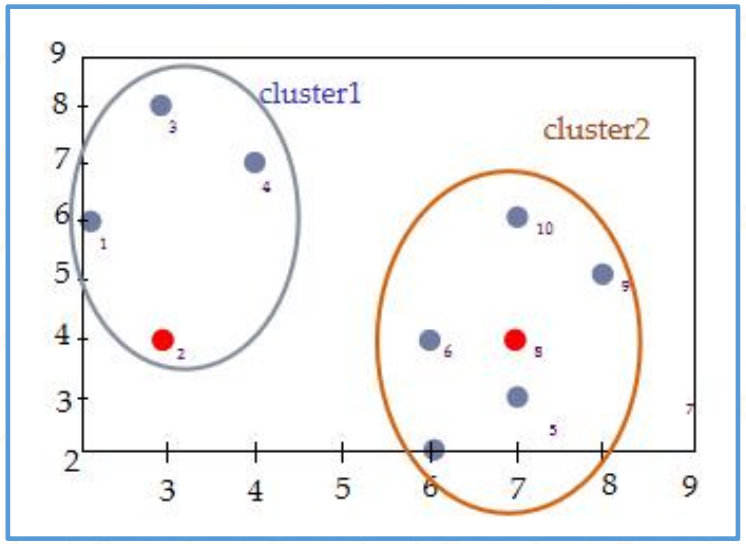


- Tetapkan data points ke titik tengah cluster terdekat



How K-Medoids works?

Data Objects	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



Hitung “absolute error criterion” [jika Medoids nya adalah (O2,O8)]

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - O_i| = (|O_1 - O_2| + |O_3 - O_2| + |O_4 - O_2|) +$$

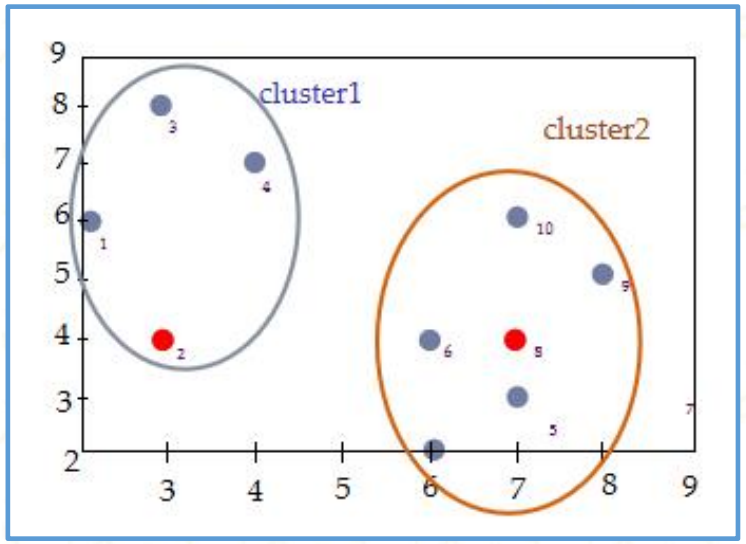
$$(|O_5 - O_8| + |O_6 - O_8| + |O_7 - O_8| + |O_9 - O_8| + |O_{10} - O_8|)$$

Nilai “absolute error criterion” [Jika Medoids nya adalah (O_2,O_8)]

E = (3+4+4)+(3+1+1+2+2) = 20

How K-Medoids works?

Data Objects	A_1	A_2
0_1	2	6
0_2	3	4
0_3	3	8
0_4	4	7
0_5	6	2
0_6	6	4
0_7	7	3
0_8	7	4
0_9	8	5
0_{10}	7	6

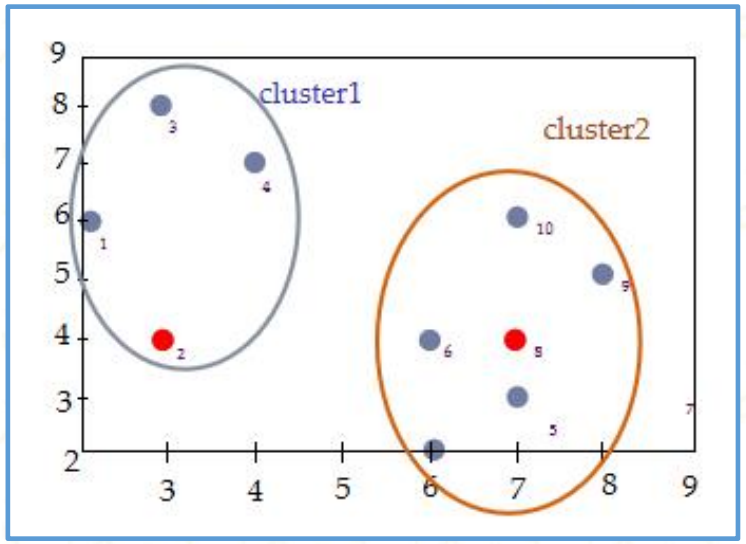


- Pilih object baru secara acak 0_7
- Ganti 0_8 dengan 0_7
- Hitung “absolute error criterion” [Jika Medoids nya adalah (0_2,0_7)

$E = (3+4+4)+(2+2+1+3+3) = 22$

How K-Medoids works?

Data Objects	A_1	A_2
0_1	2	6
0_2	3	4
0_3	3	8
0_4	4	7
0_5	6	2
0_6	6	4
0_7	7	3
0_8	7	4
0_9	8	5
0_{10}	7	6



→Hitung “cost function”

Absolute error [0_2 ,0_7] - Absolute error [for 0_2 ,0_8]

S=22-20

S> 0 => Ide yang buruk jika mengganti 0_8 dengan 0_7

Discussion on the K-medoids

- Keuntungan:
  - Mudah dipahami dan diimplementasikan
  - K-Medoid Algorithm bekerja dengan cepat dan bisa “converges” dalam jumlah steps tertentu.
  - PAM tidak terlalu sensitive terhadap outliers dibanding algorithms lain.

- Kerugian:
  - Hasilnya bisa berbeda ketika dijalankan, karena nilai medoids k adalah dipilih secara acak saat pertama kali
  - PAM algorithm dalam K-medoid clustering bekerja baik pada dataset tapi tidak bisa “scale well” untuk dataset yang besar.

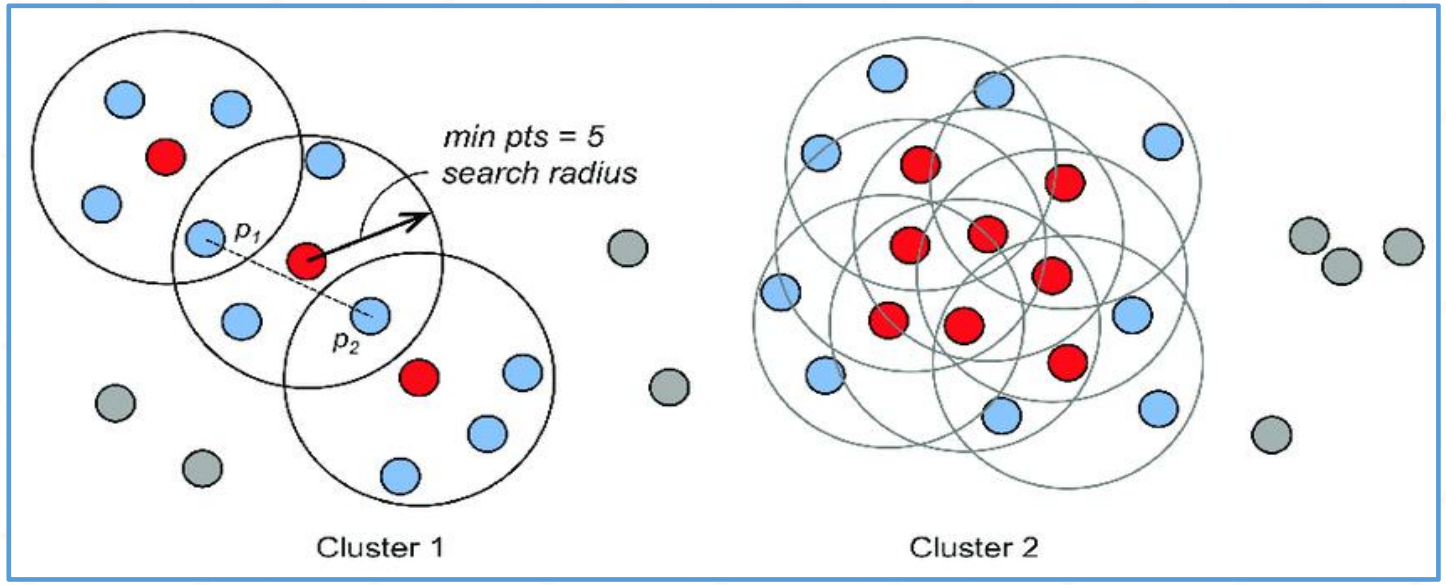
https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/

DigitalSkola
Uncover The World Of Digital Skills With Us

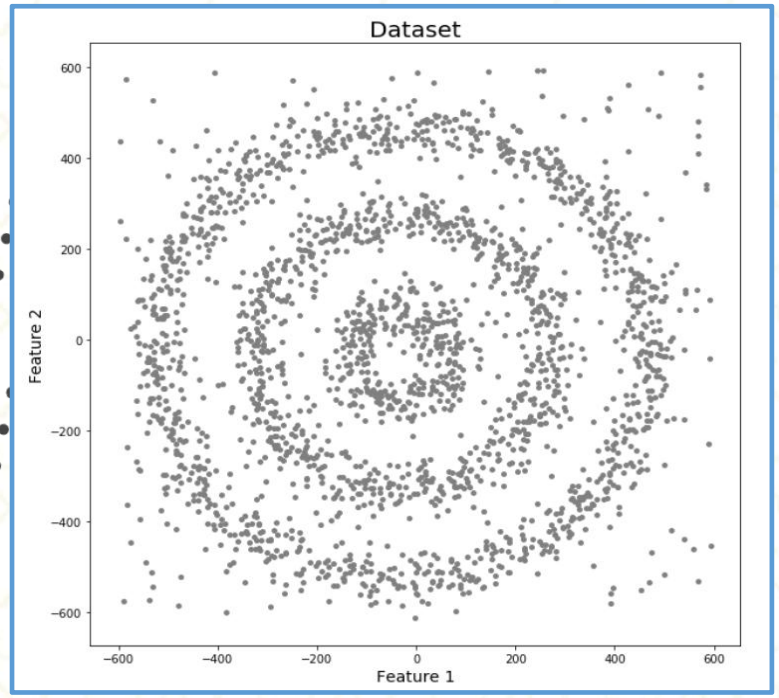
DBSCAN

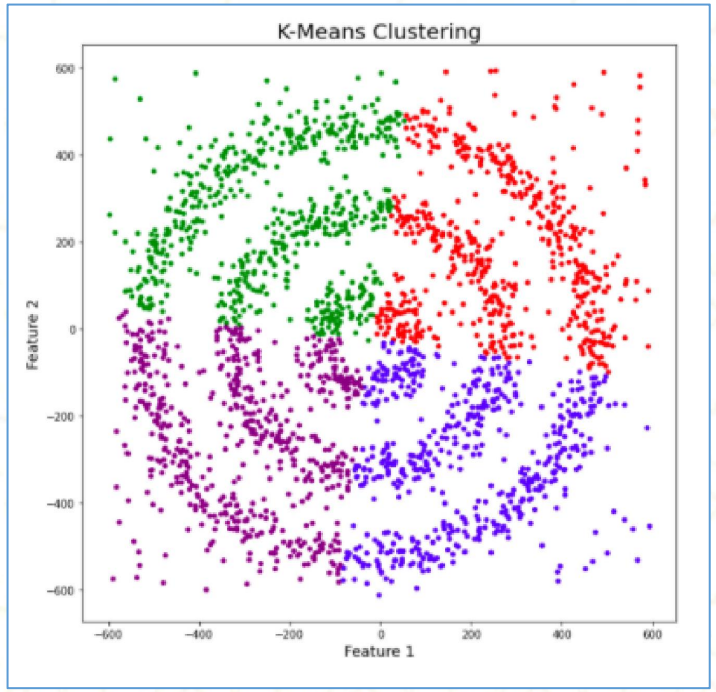
DBSCAN

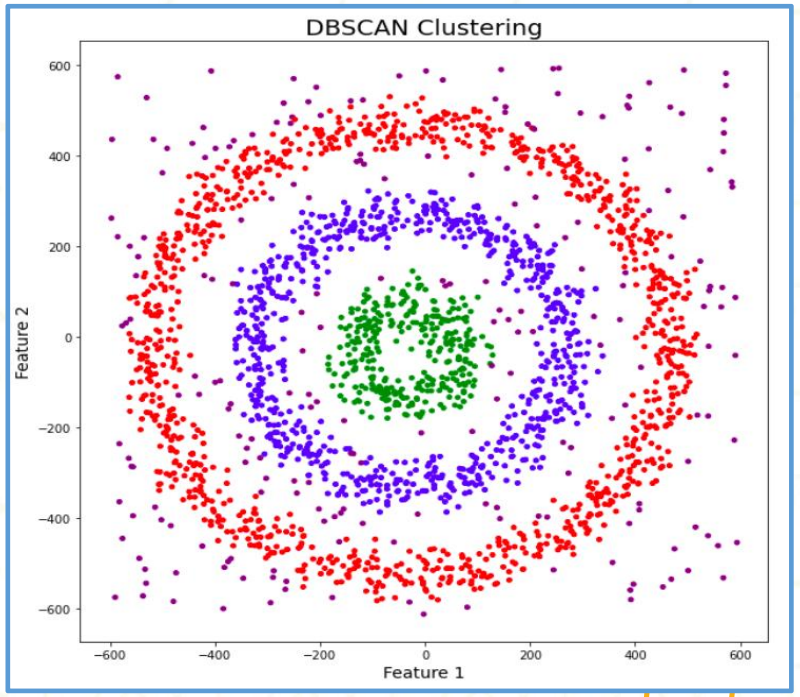
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN) adalah
  - algorithma berbasis density-based
  - Diusulkan oleh Martin Ester, Hans-Peter Kriegel, Jörg Sander dan Xiaowei Xu pada tahun 1996



Why DBSCAN?







DBSCAN dapat meng-cluster data points dengan benar, dan bisa mendeteksi noise

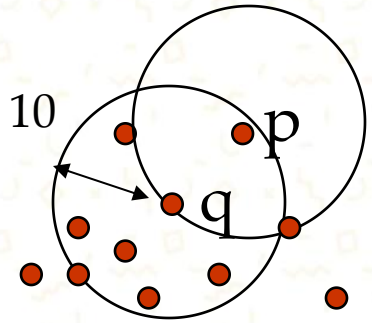
https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/

How DBSCAN works?

- Mengelompokkan objects dalam area yang padat (dense region)
- Fitur utama
  - Bisa menemukan cluster yang bentuknya “arbitrary”
  - Bisa menghandle noise
  - Satu kali scanning
  - Membutuhkan “density parameters”
- Density parameters
  - ***Radius*** ε : jarak untuk menentukan tetangga
  - ***MinPts*** : Jumlah minimum poin dalam lingkungan (neighborhood)

Definitions

- Core object
  - ε-neighborhood yang mempunyai object sejumlah ***MinPts***
- Directly density-reachable
  - ***p*** bisa disebut “directly density-reachable” dari ***q*** jika ***q*** adalah “core object”, dan ***p*** adalah ε-neighborhood dari ***q***

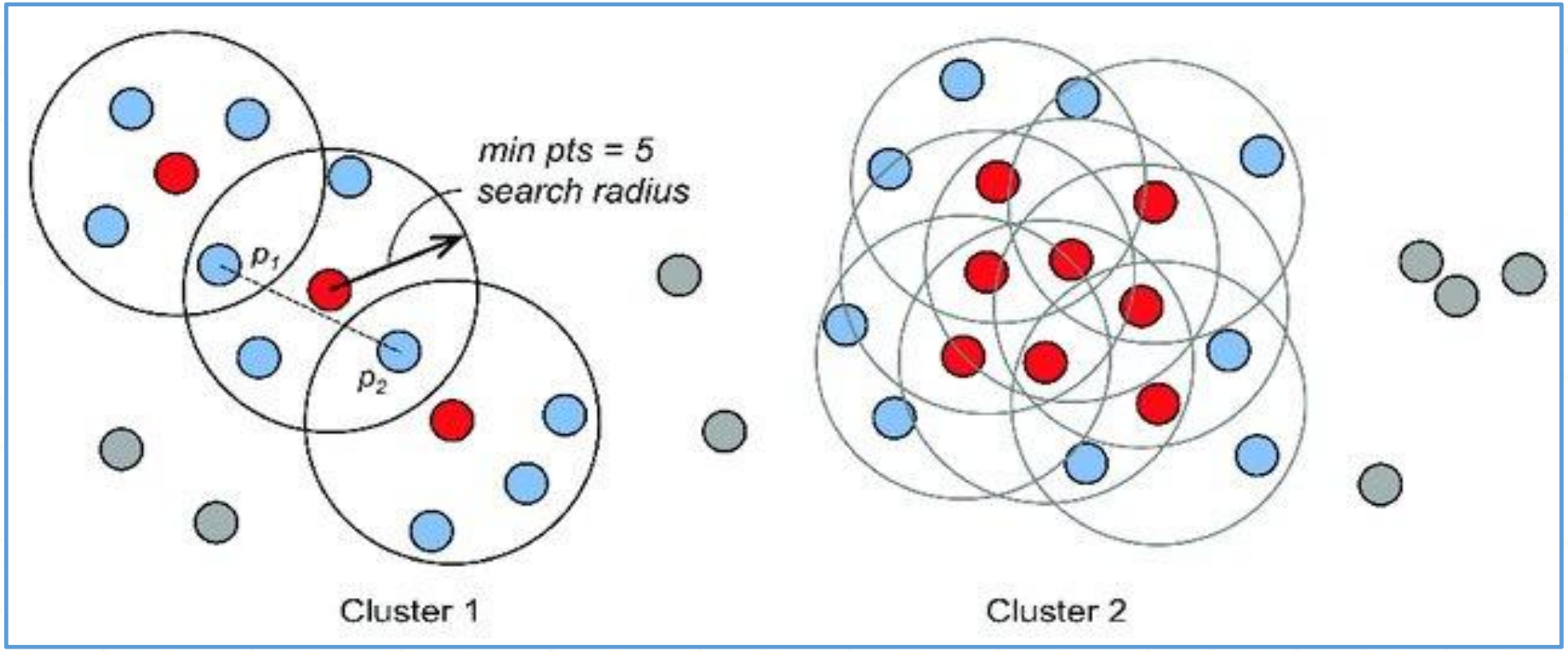


$\varepsilon = 10$
MinPts = 5

DBSCAN

- Sebuah *cluster* - koleksi dari point yang “density-connected”
- Menemukan clusters dengan bentuk “arbitrary”
  1. Memilih secara acak sebuah point ***p***
  2. Menemukan semua ε-neighborhood dari ***p***
  3. Jika ***p*** adalah sebuah core object, maka sebuah cluster bisa dibentuk
  4. Dari core object ***p***, mencari secara terus menerus object yang “density-reachable” (sehingga bisa tergabung menjadi cluster)
  5. Proses ini dikerjakan terus menerus sampai tidak ada point baru yang bisa ditambahkan ke dalam cluster
- Permasalahan dalam DBSCAN
  - Memilih parameter ε dan ***MinPts***

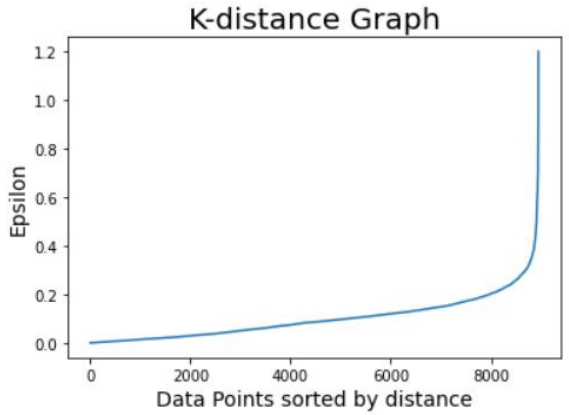
DBSCAN



- Ada 3 tipe points
  - core point (merah)
  - border points (biru)
  - noise points (abu abu)

How to determine MinPts and eps

- MinPts
  - Menggunakan “domain knowledge”
  - Semakin besar data set, semakin besar nilai dari MinPts
  - Jika dataset lebih banyak noise-nya, isikan MinPts dengan nilai yang lebih besar
  - Umumnya, MinPts >= atau = jumlah dimensi dalam data set
  - Untuk 2-dimensional data, gunakan MinPts = 4 (Ester et al., 1996).
  - Jika dataset lebih dari 2 dimensions, pilih MinPts = 2*dim, yang mana dim= jumlah dimensi dalam dataset (Sander et al., 1998).
- Epsilon
  - sorted k-dist graph



https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd

DigitalSkola
Uncover The World Of Digital Skills With Us

Lets Practice!

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU