



Data Mining Klastering & K-Means

Presentation

Team



Dionisisus Avedo
A11.2021.13228



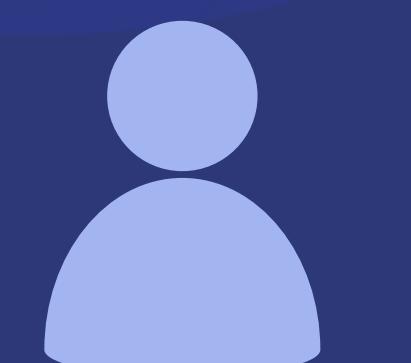
Ibrahim M
A11.2021.13603



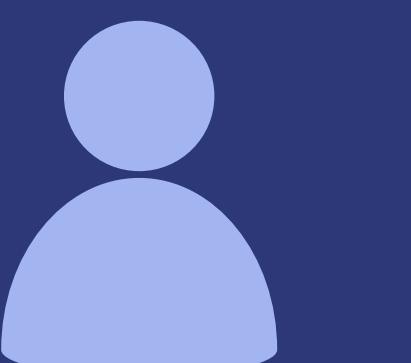
Fauzan Febryan
A11.2021.13255



Aloysius Aprillio S.N
A11.2021.13463



Qotrunanda Nabila
A11.2021.13613



Valentino Aldo
A11.2021.13838

Roadmap

Q1

Konsep
Dasar
Klustering

Q2

Konsep
Dasar
K-Means

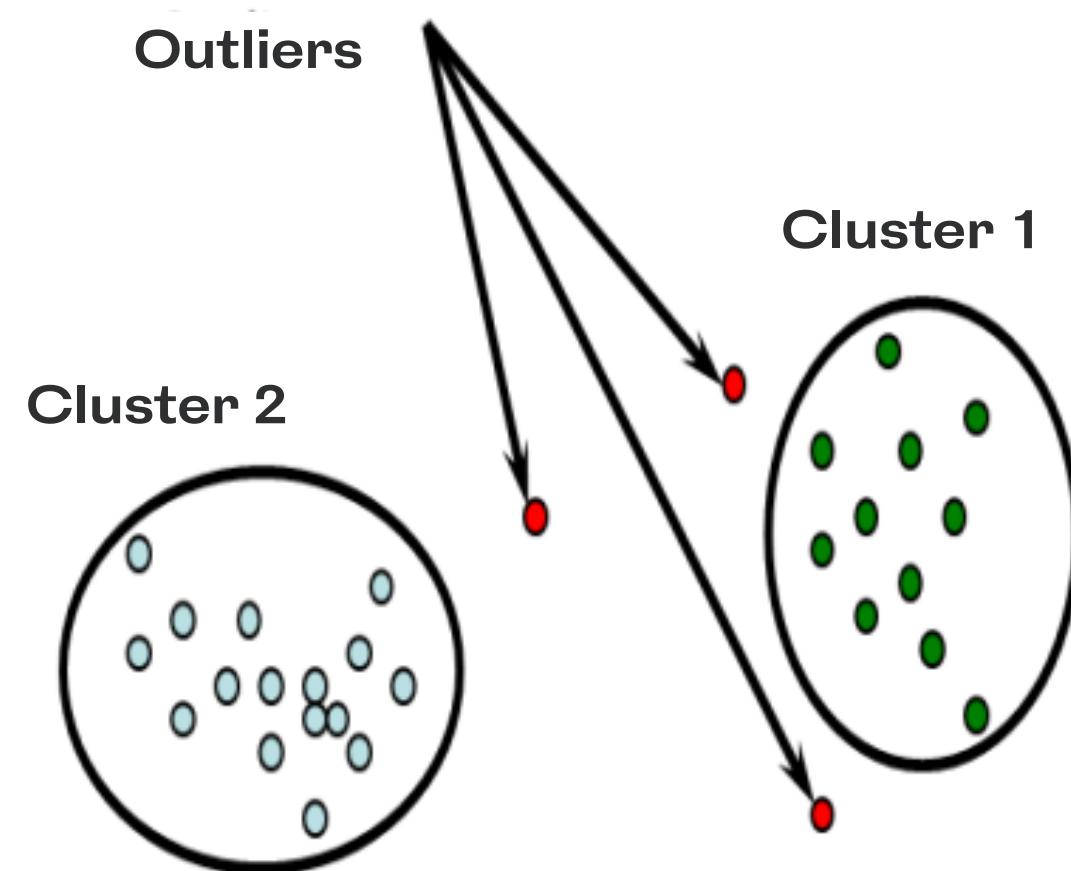
Q3

Perhitungan
Manual
K-Means

Q4

Implementasi
K-Means

Konsep Dasar Klastering



Teknik pengelompokan data yang mirip atau serupa ke dalam kelompok-kelompok yang signifikan. Tujuan utama klustering adalah mengidentifikasi pola atau struktur dalam data. Suatu klaster (cluster) adalah koleksi data yang mirip antara satu dengan yang lain, dan memiliki perbedaan bila dibandingkan dengan data dari klaster lain.

Pengelompokan mengacu pada pengelompokan rekaman, observasi, atau kasus ke dalam kelas objek serupa. Cluster adalah kumpulan record yang serupa satu sama lain, dan berbeda dengan record di cluster lain.

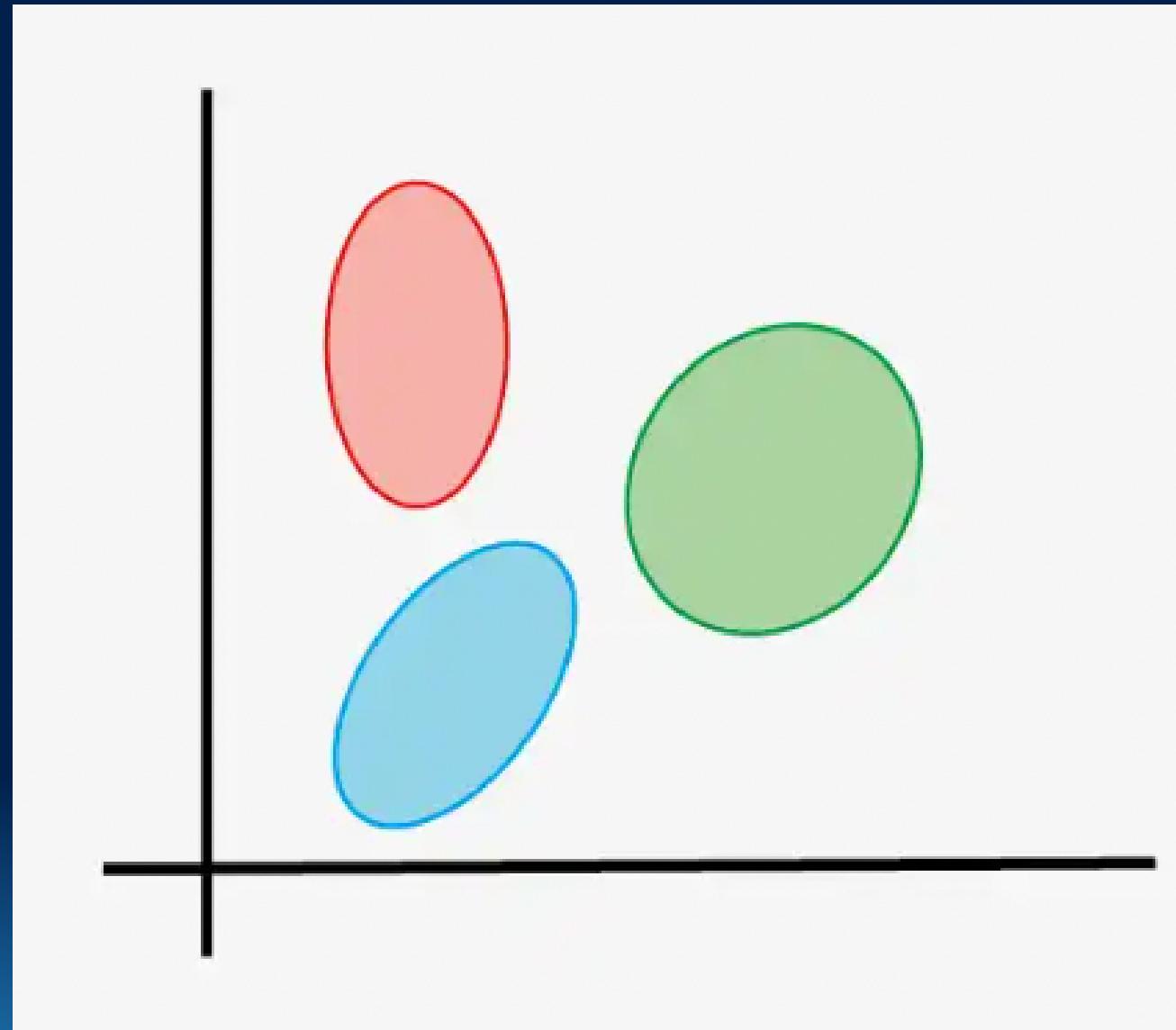
Konsep Dasar Klastering



Contohnya dalam penggunaan sosial media, klustering digunakan untuk mengelompokkan posting atau pengguna berdasarkan topik, sentimen, atau pola tertentu.

Penerapan yang sering kita jumpai yaitu dapat digunakan untuk mengidentifikasi topik yang sedang tren di Twitter.

Konsep Dasar K-Means



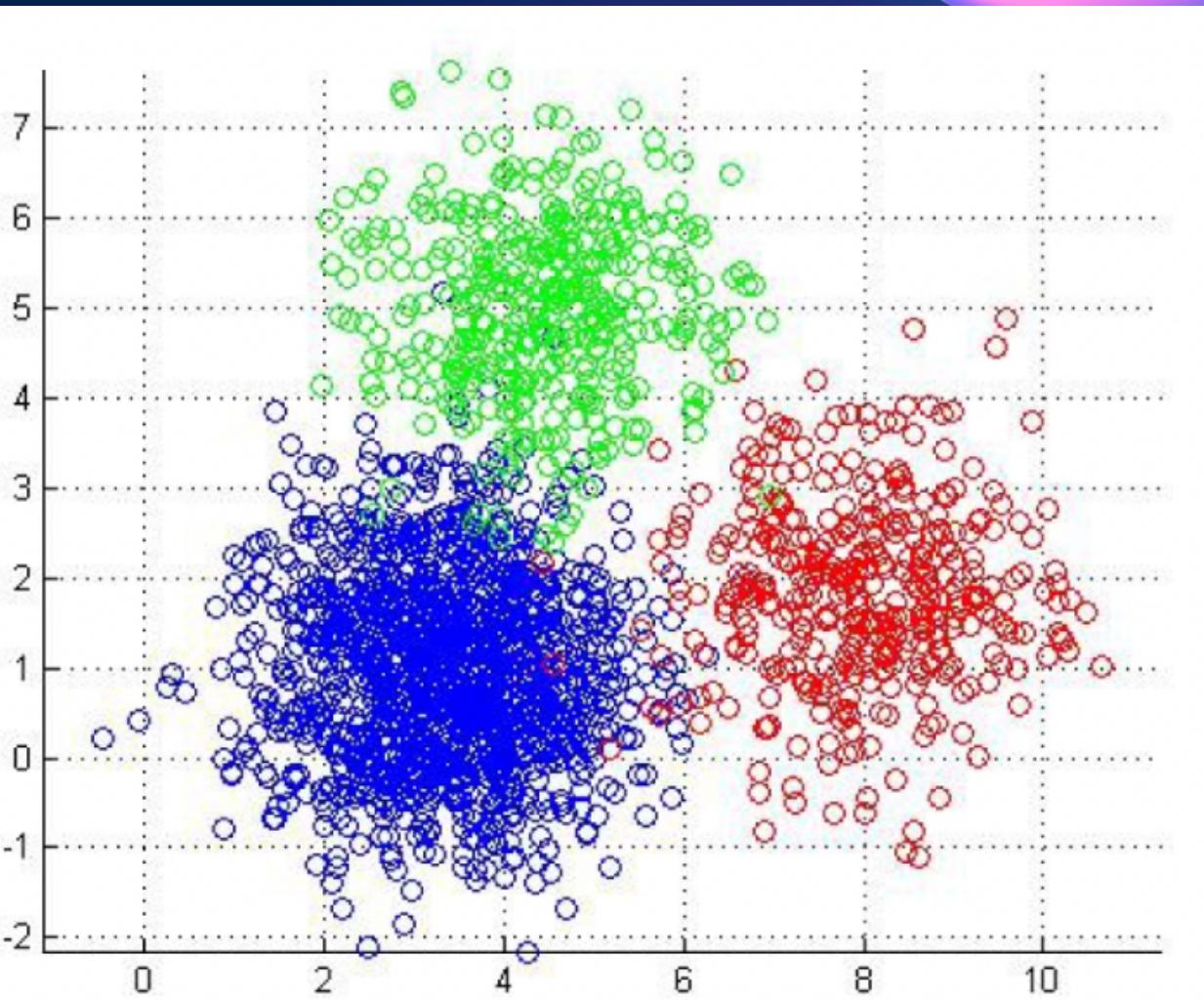
K-Means adalah suatu metode penganalisaan data atau metode Data Mining yang melakukan proses pemodelan unssupervised learning dan menggunakan metode mengelompokkan data yang belum dilabeli ke dalam kluster yang berbeda berdasarkan attribut/fitur.

Konsep Dasar K-Means

Algoritma k-means adalah algoritma untuk mengelompokkan n objek berdasarkan atribut ke dalam k partisi. objek yang digunakan nantinya akan digambarkan dalam bentuk titik didalam ruangan vector berdimensi d.

K-means bekerja dengan mensegmentasi objek yang ada kedalam kelompok sehingga objek yang berada dalam masing-masing kelompok lebih serupa satu sama lain.

Algoritma K-Means memisahkan data dengan optimal melalui perulangan yang memaksimalkan hasil dari partisi hingga tidak ada perubahan data dalam setiap segmentasi.



K-Means Clustering

Kelbihan K-Means



Simple dan mudah untuk diterapkan



Mampu mengelompokkan data dalam jumlah besar dengan cepat



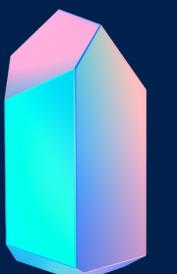
Mudah beradaptasi dengan data baru

Kekurangan K-Means



Perlu menentukan nilai k secara manual

Memilih nilai k yang salah dapat mengakibatkan hasil pengelompokan yang buruk, dan memilih nilai k yang optimal membutuhkan pengetahuan yang cukup mengenai kumpulan data.



Sangat bergantung pada inisialisasi awal

Jika nilai inisialisasi kurang baik, maka pengelompokan yang dihasilkan pun menjadi kurang optimal.

Kekurangan K-Means



sulit mengelompokkan data di mana cluster memiliki ukuran dan kepadatan yang bervariasi.

Kelompok dalam data yang memiliki ukuran dan kepadatan berbeda dapat mengakibatkan beberapa cluster jadi tidak seimbang. Hal ini dapat mempersulit interpretasi cluster yang dihasilkan.



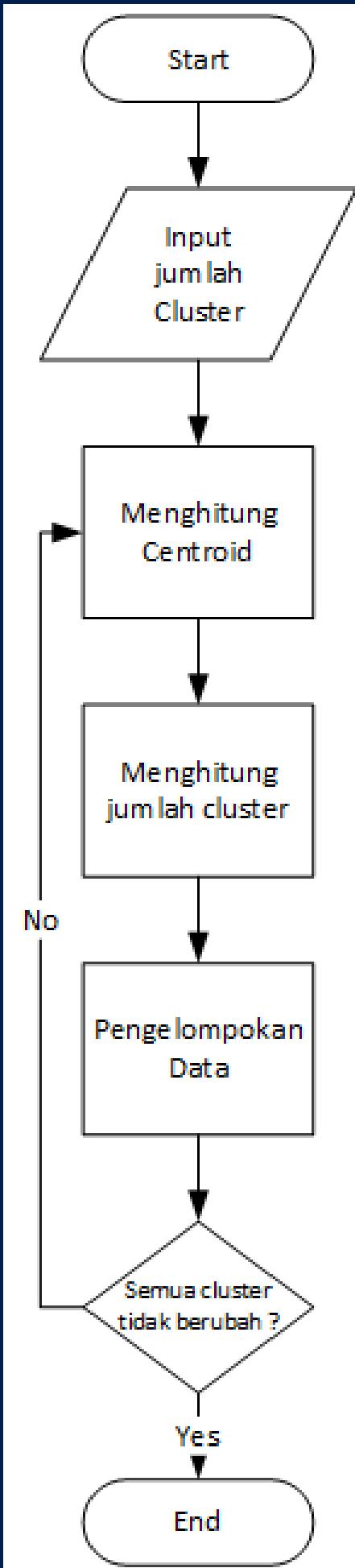
Mudah terganggu dengan data yang tidak valid

Karena cara kerja K-Means merata-rata nilai dalam setiap kelompok, maka data yang tidak valid dapat mengacaukan pusat segmen

Contoh Perhitungan Manual K-Means

Secara sederhana algoritma K-Means dimulai dari tahap berikut :

- 1. Pilih K buah titik centroid.**
- 2. Menghitung jarak data dengan centroid.**
- 3. Update nilai titik centroid.**
- 4. Ulangi langkah 2 dan 3 sampai nilai dari titik centroid tidak lagi berubah.**



Jadi dari flowchart disamping, kita memiliki input dan 3 buah proses. Yaitu pertama adalah proses menghitung centroid, kemudian proses kedua menghitung data yang akan dikelompokkan dengan centroid, kemudian proses ketiga adalah mengelompokkan data berdasarkan jarak terdekat (minimum distance). Dan kita membuat perulangan dengan kondisi "apakah posisi centroid tetap dan tidak ada perubahan terhadap datanya ?" apabila ya maka kita selesai melakukan pengelompokan. Tapi apabila masih ada perubahan centroid maka kita update kembali nilai centroid melalui proses pertama.

Contoh Soal:

Untuk meningkatkan pemahaman kita, mari kita bahas contoh soal berikut lengkap dengan perhitungannya. Dimisalkan kita memiliki sampel data dalam tabel berikut. Ada 6 buah data yang akan kita kelompokkan menjadi 2 cluster. Kita sebut saja K1 dan K2.

Sample Data	X	Y	Kelompok / Cluster
1	100	50	
2	40	60	
3	30	70	
4	90	10	
5	65	40	
6	25	35	

Pertama kita akan menghitung centroid. Kita ambil data ke-1 dan ke-2 sebagai perhitungan pertama. Kita menggunakan rumus Euclidean Distance untuk mendapatkan jarak minimum data terhadap centroid.

Cluster	X	Y
K1	100	50
K2	40	60

Berikut ini adalah rumus dari Euclidean Distance :

$$[(x, y), (a, b)] = \sqrt{(x - a)^2 + (y - b)^2}$$

Perhitungan Pertama

Kita mulai menghitung centroid pertama. Kita akan menentukan jarak dari data 1 ke data 1, data 1 ke data 2, data 2 ke data 1 dan data 2 ke data 2.

$$\text{Cluster 1 } \underline{(100, 50)} = \sqrt{(100 - 100)^2 + (50 - 50)^2} = 0$$

(jarak cluster 1 ke cluster 1)

Jarak dari Cluster 2 ke cluster 1 $(40, 60) \leftrightarrow (100, 50)$ =

$$\sqrt{(40 - 100)^2 + (60 - 50)^2} = \sqrt{(60)^2 + (10)^2} = \sqrt{3600 + 100} = \sqrt{3700} = 60.83$$

Jarak dari cluster 1 ke cluster 2 $(100, 50) \leftrightarrow (40, 60)$ =

$$\sqrt{(100 - 40)^2 + (50 - 60)^2} = \sqrt{(60)^2 + (-10)^2} = \sqrt{3600 + 100} = \sqrt{3700} = 60.83$$

Jarak cluster 2 ke cluster 2 $\underline{(40, 60)} = \sqrt{(40 - 40)^2 + (60 - 60)^2} = 0$

Pada bagian yang diberikan garis merah adalah data dan perhitungan, sedangkan yang diberikan garis biru adalah hasil yang akan kita masukkan ke dalam tabel untuk menentukan data tersebut akan masuk kedalam cluster K1 atau K2. Sehingga kita dapatkan hasil seperti berikut :

Cluster	Centroid		Kelompok Cluster
	X	Y	
K1 (100, 50)	0	60.83	1
K2 (40, 60)	60.83	0	2

Dari tabel diatas, kita lihat jarak minimum dari data 1 ke data 1 adalah 0 dan jarak minimum dari data 2 ke data 2 yaitu 0. Sehingga yang menjadi centroid K1 adalah data 1 dan data 2 menjadi centroid K2. Selanjutnya kita beralih ke perhitungan kedua untuk data ke-3.

Perhitungan Kedua

Setelah mendapatkan centroid, kita beralih ke data ke-3 yaitu (30, 70). Kita mulai hitung jarak data ke-3 terhadap centroid 1 dan centroid 2. Sehingga hasilnya nanti kita mengetahui data 3 akan masuk ke cluster K1 atau K2. Berikut adalah perhitungannya :

Langkah selanjutnya kita beralih ke data 3 yaitu (30, 70). Kita mulai menghitung jarak dataset terhadap cluster 1.

■ Centroid
■ Dataset

$$(100, 50) \leftrightarrow (30, 70) =$$

$$\sqrt{(30 - 100)^2 + (70 - 50)^2} = \sqrt{(-70)^2 + (20)^2} = \sqrt{4900 + 400} = \sqrt{5300} = 72.80$$

Kemudian kita hitung jarak dataset terhadap cluster 2.

$$(40, 60) \leftrightarrow (30, 70) =$$

$$\sqrt{(30 - 40)^2 + (70 - 60)^2} = \sqrt{(-10)^2 + (10)^2} = \sqrt{100 + 100} = \sqrt{200} = 14.14$$

Garis hijau sebagai centroid, garis merah sebagai dataset dan garis biru sebagai hasil. Sebenarnya perhitungan $(x-a)^2$ dan $(y-b)^2$ dapat dibalik menjadi $(a-x)^2$ dan $(b-y)^2$ karena hasilnya pasti positif sebab dikuadratkan.

Semuanya dihitung dengan Euclidean Distance dan hasilnya kita masukkan dalam tabel berikut :

Dataset	Euclidean Distance		Kelompok Cluster
	Cluster 1	Cluster 2	
(30 , 70)	72.80	14.14	2

Dilihat dari jarak minimum data terhadap centroid, data ke-3 lebih dekat ke K2 dengan nilai 14.14.

Selanjutnya kita meng-UPDATE nilai Centroid. Karena data masuk ke K2, maka centroid K2 diupdate dengan cara :

$$X_{\text{centroid_baru}} = (x_{\text{K2}} + x_{\text{data3}})/2$$

$$Y_{\text{centroid_baru}} = (y_{\text{K2}} + y_{\text{data3}})/2$$

Sehingga centroid yang baru kita dapatkan , pada garis merah adalah persamaan perhitungannya dan pada kotak merah adalah hasilnya.

Cluster	X	Y
K1	100	50
K2	$= \frac{40 + 30}{2} = 35$	$= \frac{60 + 70}{2} = 65$

Jadi Cluster Centroid yang baru adalah sebagai berikut :

Cluster	X	Y
K1	100	50
K2	35	65

Perhitungan Ketiga

Berlanjut ke data ke-4 yaitu (90, 10), kita mulai menghitung jarak antara dataset dan Centroid K1, sama seperti langkah di perhitungan kedua. dan berikut adalah perhitungannya :

$$(100, 50) \leftrightarrow (90, 10) =$$

$$\sqrt{(90 - 100)^2 + (10 - 50)^2} = \sqrt{(-10)^2 + (-40)^2} = \sqrt{100 + 1600} = \sqrt{1700} = 41.23$$

Berikut ini adalah perhitungannya :

$$(35, 65) \leftrightarrow (90, 10) =$$

  Dataset

$$\sqrt{(90 - 35)^2 + (10 - 65)^2} = \sqrt{(55)^2 + (-55)^2} = \sqrt{3025 + 3025} = \sqrt{6050} = 77.78$$

 centroid baru

Dari kedua perhitungan jarak dataset ke K1 dan K2, didapatkan hasil sebagai berikut :

Dataset	Euclidean Distance		Kelompok Cluster
	Cluster 1	Cluster 2	
(90 , 10)	41.23	77.78	1

Dari hasil diatas, kita mendapatkan jika dataset ke-4 masuk dalam cluster 1. Jadi seperti langkah sebelumnya, kita update kembali centroid K1 dengan dataset ke-3.

Kemudian kita update Centroid

Cluster	X	Y
K1	$= \frac{100 + 90}{2} = 95$	$= \frac{50 + 10}{2} = 30$
K2	35	65

Jadi Cluster Centroid yang baru adalah sebagai berikut :

Cluster	X	Y
K1	95	30
K2	35	65

Perhitungan selanjutnya untuk dataset ke-5 dan ke-6 mengikuti langkah-langkah yang telah kita lakukan pada dataset ke-3 dan ke-4. Yang perlu diperhatikan adalah menggunakan centroid yang telah di update dan juga melakukan update centroid yang baru.

Dari hasil perhitungan hingga dataset ke-6 didapatkan hasil seperti tabel berikut :

Sample Data	X	Y	Kelompok / Cluster
1	100	50	1
2	40	60	2
3	30	70	2
4	90	10	1
5	65	40	1
6	25	35	2

Contoh Implementasi

Penerapan Algoritma K-Means Clustering untuk menentukan gaji karyawan termasuk rendah, rata-rata, atau tinggi

Pencarian kelompok gaji karyawan saat ini masih dilakukan secara manual, sehingga masih banyak ditemukan kesimpangan dalam penerimaan gaji, hal tersebut mengakibatkan banyak karyawan yang mengeluh saat penerimaan gaji

Untuk itu diperlukan teknik algoritma K-Means klustering, Sehingga penggajian dapat dikelompokkan kedalam beberapa kluster yaitu gaji karyawan rendah, rata-rata, tinggi.

Contoh Implementasi

	A	B	C
1	Hari	Total	
2	22	3529000	
3	22	3300000	
4	22	2725000	
5	20	2500000	
6	21	2550000	
7	22	2813000	
8	22	2813000	
9	23	2850000	
10	23	2850000	

Data tersebut merupakan data karyawan yang memiliki atribut Hari dan Total(Gaji). Dataset tersebut disimpan dengan nama file databaru.csv, dataset yang digunakan berjumlah 50 data.

Tahap Preprocessing Data

Import library

```
[ ] #Import Library yang akan digunakan  
#%matplotlib inline  
import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
from sklearn.cluster import KMeans
```

import matplotlib.pyplot as plt

pernyataan impor yang digunakan untuk mengimpor modul pyplot dari pustaka Matplotlib. Modul pyplot digunakan untuk membuat visualisasi grafik, plot, dan diagram dalam Python.

Tahap Preprocessing Data

Import library

```
[ ] #Import Library yang akan digunakan  
#%matplotlib inline  
import matplotlib.pyplot as plt  
import numpy as np  
import pandas as pd  
from sklearn.cluster import KMeans
```

from sklearn.cluster import KMeans
import yang digunakan dalam Python untuk mengimpor
modul KMeans dari pustaka Scikit-learn (sklearn).

Import dataset



```
#Menyiapkan data dan memanggil dataset  
dataset = pd.read_csv('databaru.csv')  
dataset.keys()
```

Visualisasi persebaran data



```
plt.scatter(X[:,0], X[:,1], label='True Position')  
plt.xlabel("Hari")  
plt.ylabel("Total Gaji(juta)")  
plt.title("Grafik Persebaran Data Penggajian Karyawan")  
plt.show()
```

ylabel dan **xlabel** digunakan untuk menentukan sumbu x dan y dimana x adalah Hari kerja sedangkan Y adalah Total Gaji

Membuat objek kmeans dari modul KMeans

```
kmeans = KMeans(n_clusters=3)  
kmeans.fit(x)
```

n_clusters=3 digunakan untuk menentukan jumlah klaster menjadi 3

Menentukan hasil klasterisasi dengan centroid

```
print(kmeans.cluster_centers_)
```

[[2.3600000e+01 3.3378000e+06]	[2.0000000e+01 1.93330435e+06]
[2.15454545e+01 2.68495455e+06]]	

Menampilkan titik koordinat pusat pada kmeans kluster 3

Membuat objek kmeans dari modul KMeans

```
kmeans = KMeans(n_clusters=3)  
kmeans.fit(x)
```

n_clusters=3 digunakan untuk menentukan jumlah klaster menjadi 3

Menentukan hasil klasterisasi dengan centroid

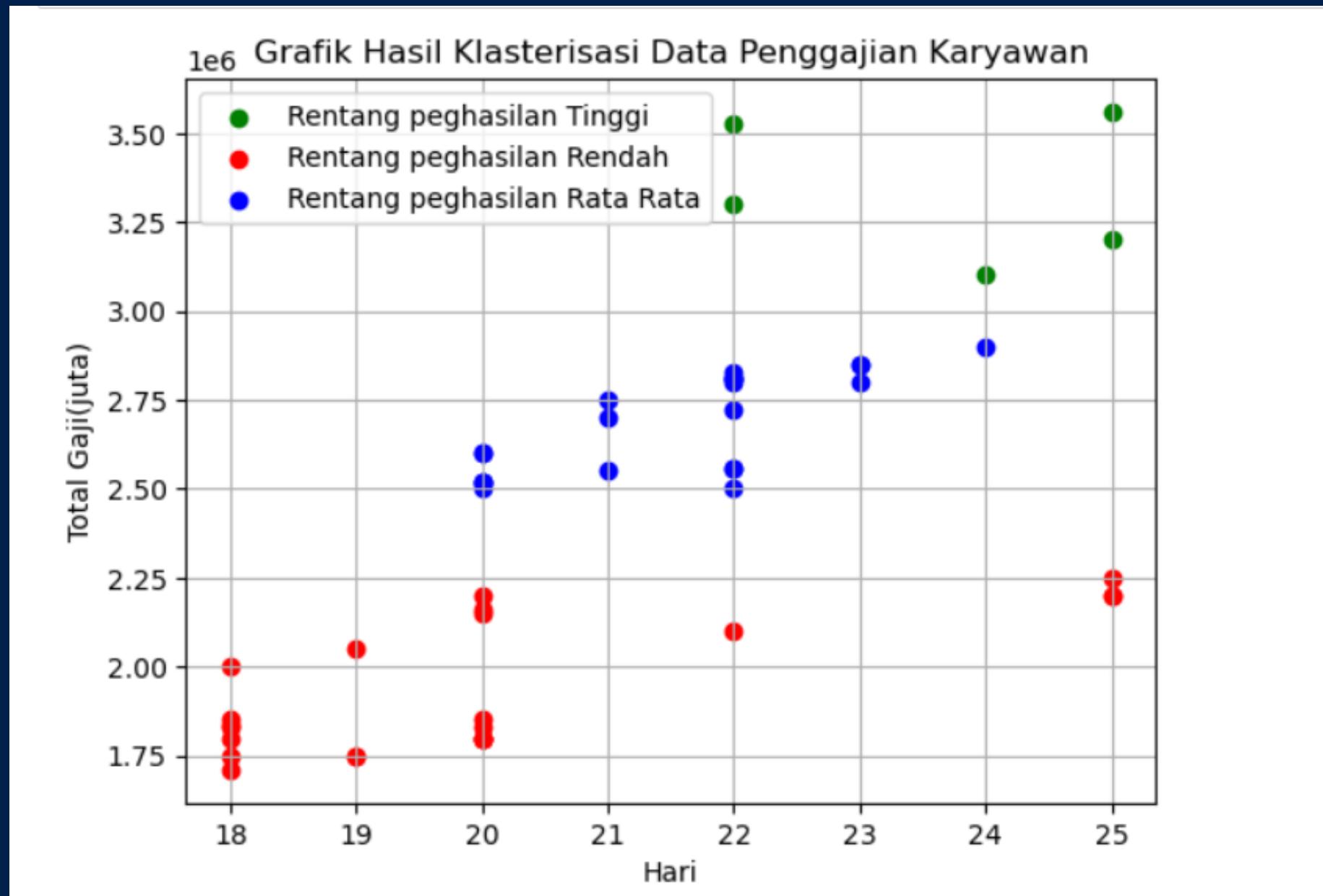
```
print(kmeans.cluster_centers_)
```



```
[[2.3600000e+01 3.3378000e+06]  
 [2.0000000e+01 1.93330435e+06]  
 [2.15454545e+01 2.68495455e+06]]
```

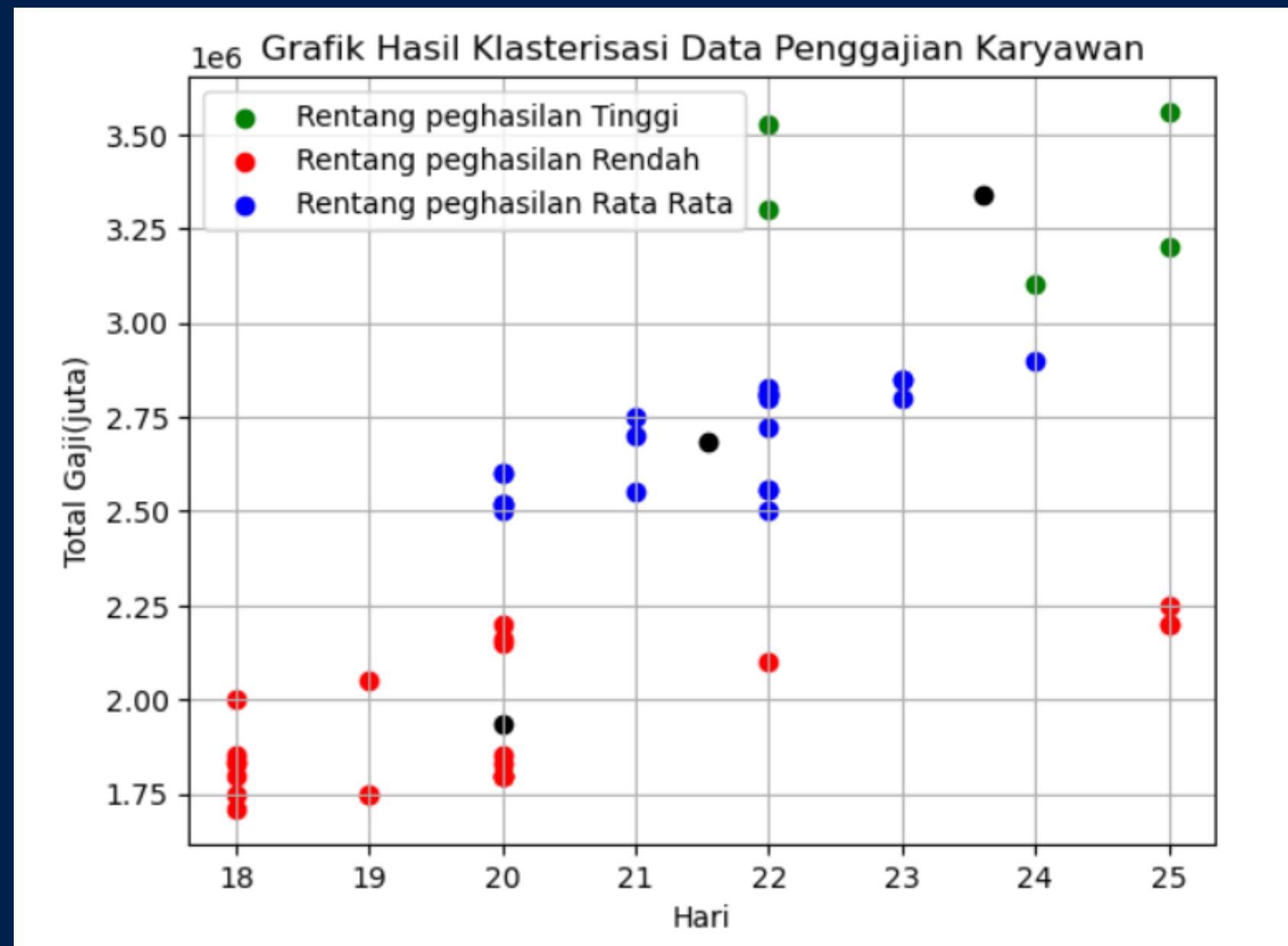
Menampilkan titik koordinat pusat pada kmeans kluster 3

Visualisasi pengelompokan data dalam satu scatter plot



**memvisualisasikan hasil
klasterisasi dengan "centroid"
dari masing masing klaster**

```
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color="black")
```



sehingga dapat dilihat bahwa

- total hari kerja <22 = kelompok warna biru melambangkan penghasilan yang lebih tinggi dari kelompok merah artinya kelompok warna biru juga lebih tinggi rentan hari kerjanya daripada kelompok merah
 - total hari kerja >22 = kelompok warna hijau melambangkan penghasilan yang lebih tinggi dari kelompok biru artinya kelompok warna hijau juga lebih tinggi rentan hari kerjanya daripada kelompok biru

Implementasi K-Means





Thank
you