K-Means Clustering

ANGGA SULISTIANGGA 2007411004

Import Library

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
```

Dataset

```
df = pd.read_csv("go_track_tracks.csv")
df.head()
```

	id	id_android	speed	time	distance	rating	rating_bus	rating_weat
0	1	0	19.210586	0.138049	2.652	3	0	
1	2	0	30.848229	0.171485	5.290	3	0	
2	3	1	13.560101	0.067699	0.918	3	0	
3	4	1	19.766679	0.389544	7.700	3	0	
4	8	0	25.807401	0.154801	3.995	2	0	

Dataset memiliki 10 Feature. tetapi hanya Feature Speed dan Distance saja yang akan digunakan. maka 8 Feature lain akan di drop.

```
df = df.drop(["linha", "car_or_bus", "rating_weather", "rating_bus", "rating", "time"], axis = 1)
df.head()
```

	id	id_android	speed	distance
0	1	0	19.210586	2.652
1	2	0	30.848229	5.290
2	3	1	13.560101	0.918
3	4	1	19.766679	7.700
4	8	0	25.807401	3.995

Setelah data di drop tentukan Feature yang akan di kluster. Feature Speed = X Feature Distance = Y

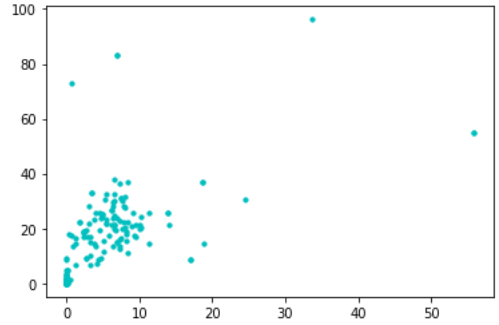
```
klustered_data = df.iloc[:, 2:4]
klustered_data.head()
```

	speed	distance
0	19.210586	2.652
1	30.848229	5.290
2	13.560101	0.918
3	19.766679	7.700
4	25.807401	3.995

Setelah dikluster maka buat Plot untuk memvisualisasikan persebaran data.

```
plt.scatter(df.distance, df.speed, s = 10, c = 'c', marker = 'o', alpha =1)
plt.show
```

```
<function matplotlib.pyplot.show(*args, **kw)>
```



Menentukan Nilai K. Tetapi sebelum menentukan nilai K harus mengubah dari data frame menjadi array.

```
df_array = np.array(klustered_data)
print(df_array)
```

```
[1.44400981e+01 3.87200000e+00]
[1.63567325e+01 1.26000000e+00]
[1.75427999e+01 5.85700000e+00]
[9.45181557e+00 2.61600000e+00]
[9.45181557e+00 2.61600000e+00]
[1.62635039e+01 7.33400000e+00]
[2.12235944e+01 6.14900000e+00]
[1.94236545e+01 4.59500000e+00]
[2.07996291e+01 8.84900000e+00]
[8.72437242e+00 1.69510000e+01]
[8.72437242e+00 1.69510000e+01]
```

```
[2.15138018e+01 1.41400000e+01]
[2.75257703e+01 8.04800000e+00]
[2.81045207e+01 7.77300000e+00]
[2.23224625e+01 1.84200000e+00]
[2.23224625e+01 1.84200000e+00]
[3.23773040e+01 5.38100000e+00]
[2.23779245e+01 7.76900000e+00]
[2.49082559e+01 6.59700000e+00]
[3.71409017e+01 1.86020000e+01]
[3.71409017e+01 1.86020000e+01]
[7.29267545e+01 6.61000000e-01]
[6.63753526e-02 1.00000000e-03]
[8.33281351e+01 6.97600000e+00]
[8.33281351e+01 6.97600000e+00]
[9.62060288e+01 3.37390000e+01]
[3.23767476e+00 3.30000000e-02]
[5.82100412e-01 2.00000000e-03]
[1.38683788e+00 6.00000000e-03]
[1.80865138e-01 6.00000000e-03]
[1.87306515e-01 6.00000000e-03]
[2.74035168e+00 1.00000000e-02]
[1.55177191e+00 1.40000000e-02]
[8.91614215e-01 2.90000000e-02]
[1.26197446e+00 1.68000000e-01]
[1.98334721e+01 1.01010000e+01]
[1.68934215e+01 9.52300000e+00]
[2.05827023e+01 1.01510000e+01]
[3.07454686e-01 1.41000000e-01]
[3.07454686e-01 1.41000000e-01]
[2.16717437e+01 5.39500000e+00]
[2.16717437e+01 5.39500000e+00]
[3.32011780e+01 3.47300000e+00]
[3.32011780e+01 3.47300000e+00]
[2.68955892e+01 6.19900000e+00]
[2.68955892e+01 6.19900000e+00]
[3.10853313e+01 7.47300000e+00]
[3.10853313e+01 7.47300000e+00]
```

Menstandarkan kembali ukuran variabel . Agar data dapat kembali seperti jenis data diawal sebelum di array kan.

```
scaler = MinMaxScaler()
scaled_x = scaler.fit_transform(df_array)
scaled_x
```

```
       [9.59005652e-04, 0.00000000e+00],
       [1.74023986e-03, 6.63451021e-04],
       [2.69738408e-01, 2.50138966e-01],
       [2.69738408e-01, 2.50138966e-01],
       [1.45984320e-02, 7.17244347e-05],
       [1.73349676e-01, 1.33299862e-01],
       [1.13637587e-01, 1.50280622e-01],
       [3.15888254e-01, 4.38594918e-01],
       [7.01686029e-02, 5.91188653e-02],
       [2.66592088e-01, 1.64033782e-01],
       [1.83424124e-01, 1.66364826e-01],
       [2.54635277e-01, 8.70017393e-02],
       [1.20137670e-01, 9.15203787e-02],
       [1.51992589e-01, 2.01133246e-01],
       [4.04065940e-03, 1.07586652e-04],
       [9.42560575e-02, 8.33796554e-02],
       [2.64475521e-01, 8.85079524e-02],
       [2.64475521e-01, 8.85079524e-02],
       [2.01328167e-01, 1.22074988e-01],
       [7.41015219e-02, 7.43244455e-02],
       [1.84353102e-02, 1.07586652e-04],
       [1.68647757e-03, 1.39862648e-03],
       [1.35330904e-02, 3.58622174e-04],
       [2.41065661e-01, 9.74197135e-02],
       [2.44221046e-01, 6.71699331e-02],
       [1.49211922e-02, 2.38483745e-03],
       [5.47284209e-03, 1.97242195e-04],
       [3.80859883e-01, 1.31452958e-01],
       [3.25974844e-01, 1.43861285e-01],
       [2.65630080e-01, 8.13534401e-02],
       [3.95216323e-01, 1.16821173e-01],
       [2.92450674e-01, 1.14382542e-01],
       [2.26643869e-01, 5.62140257e-02],
       [2.47605654e-01, 8.41148308e-02],
       [2.94469276e-01, 5.61064391e-02],
       [3.12298584e-01, 1.17861177e-01],
       [3.13567412e-01, 1.38159192e-01],
       [1.18922807e-02, 2.51035522e-04],
       [8.66399259e-03, 8.96555434e-05],
       [1.41712211e-02, 3.94484391e-04]])
```

Tentukan nilai K. Contohnya misalkan K = 3

```
kmeans = KMeans(n_clusters = 3, random_state = 123)
kmeans.fit(scaled_x)
```

```
KMeans(n_clusters=3, random_state=123)
```

Visualisasi Kluster

Centroid

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

```
[[0.23246003 0.12863014]
 [0.02863662 0.00851813]
 [0.77224472 0.47782221]]
```

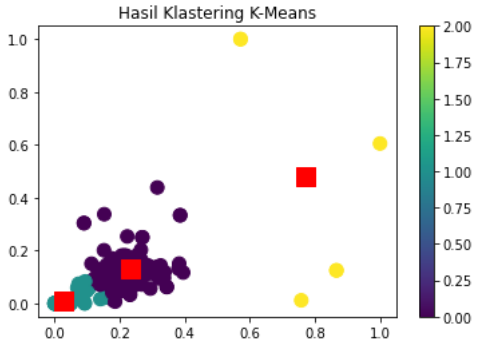
Menampilkan kluster dan Menambahkan kolom data frame driver .

```
print(kmeans.labels_)
df['klaster'] = kmeans.labels_
```

```
[0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 2 2 0 1 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 2 1 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1
 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
 0 1 0 1 1 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0
 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1]
```

Visualisasi Hasil Kluster

```
output = plt.scatter(scaled_x[:, 0], scaled_x[:, 1], s = 100, c = df.klaster, marker = 'o', alpha = 1)
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c = 'red', s = 200, alpha = 1, marker = 's');
plt.title('Hasil Klastering·K-Means')
plt.colorbar(output)
plt.show()
```



✓ 0 d selesai pada 23.11