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

```
# import data
df = pd.read_excel('/content/drive/MyDrive/Project Mobile Dan Data Science/SYARAT/Data Pelaporan Infrastruktur.xlsx')
df.head()
```

	Timestamp	Nama	Jenis Infrastruktur	Lokasi	Faktor Pennyebab	Tanggal	Tingkat Kerusakan
0	2023-12-19 21:56:19.253	Tegar Muhammad Jiwa	Jalan	Kecamatan Tampan	Faktor Cuaca	2021-10-12	6
1	2023-12-19 22:28:42.075	M. Hibatillah Hasanin	Jalan	Kecamatan Rumbai	Faktor Intensitas Penggunaan	2023-08-07	6
2	2023-12-20 11:32:29.595	David	Jalan	Kecamatan Payung Sekaki	Faktor Usia Infrastruktur	2023-08-08	5
3	2023-12-20	Vita	Jembatan Penyebrangan	Kecamatan	Faktor	2023-	5

```
# amati
df.shape
```

```
(80, 8)
```

```
df.describe()
```

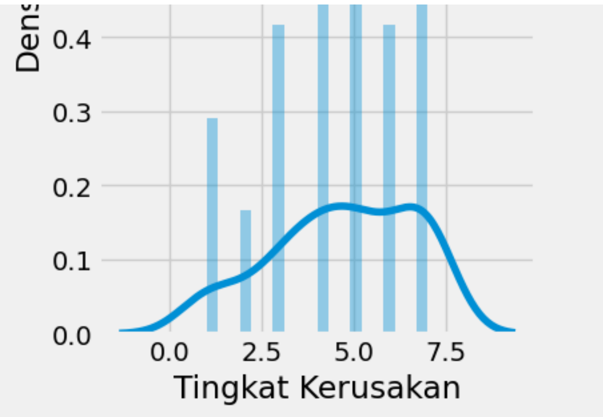
	Tingkat Kerusakan	Tingkat Dampak
count	80.0000	80.000000
mean	4.7000	4.912500
std	1.8987	1.843179
min	1.0000	1.000000
25%	3.0000	4.000000
50%	5.0000	5.000000
75%	6.2500	7.000000
max	7.0000	7.000000

```
# cek null data
df.isnull().sum()
```

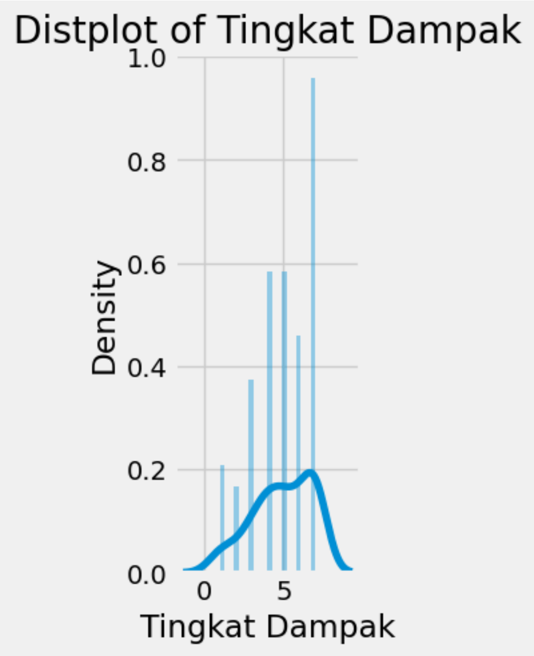
```
Timestamp               0
Nama                    0
 Jenis Infrastruktur    0
Lokasi                  0
Faktor Pennyebab        0
Tanggal                 0
Tingkat Kerusakan       0
Tingkat Dampak          0
dtype: int64
```

```
# tingkatkan visualisasi data
plt.style.use('fivethirtyeight')
```

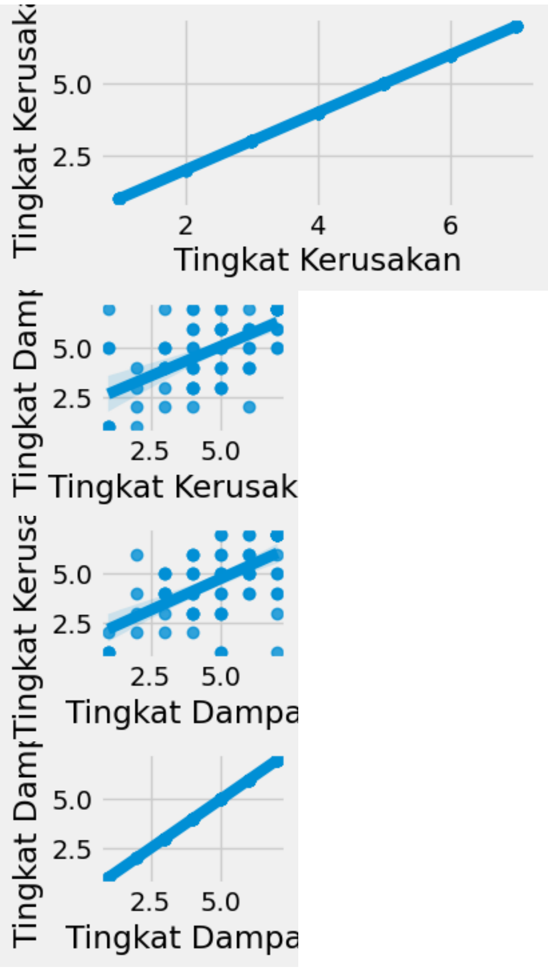
```
# amati masing-masing fitur
plt.figure(1 , figsize = (15 , 6))
n = 0
for x in ['Tingkat Kerusakan' , 'Tingkat Dampak']:
  n += 1
  plt.subplot(1 , 3 , n)
  plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
  sns.distplot(df[x] , bins = 20)
  plt.title('Distplot of {}'.format(x))
  plt.show()
```



```
<ipython-input-13-a9c81b3c06f1>:8: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df[x] , bins = 20)
```



```
# Ploting untuk mencari relasi antara Age , Annual Income and Spending Score
plt.figure(1 , figsize = (15 , 7))
n = 0
for x in ['Tingkat Kerusakan' , 'Tingkat Dampak']:
  for y in ['Tingkat Kerusakan' , 'Tingkat Dampak']:
    n += 1
    plt.subplot(3 , 3 , n)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.regplot(x = x , y = y , data = df)
    plt.ylabel(y.split()[0]+' '+y.split()[1] if len(y.split()) > 1 else y )
    plt.show()
```

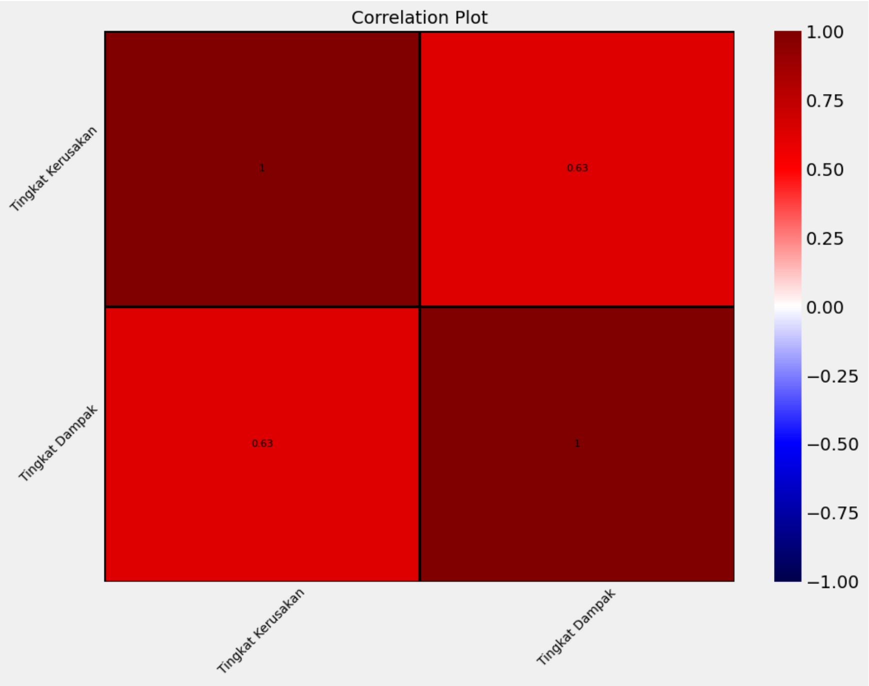


```
#Generate a heatmap for the feature correlations
corr_matrix = df.corr()
corr_matrix
```

```
<ipython-input-40-4d16b3f01176>:2: FutureWarning: The default value of numeric_only i
  corr_matrix = df.corr()
```

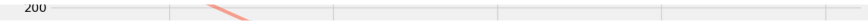
	Tingkat Kerusakan	Tingkat Dampak
Tingkat Kerusakan	1.00000	0.62538
Tingkat Dampak	0.62538	1.00000

```
#Format the heatmap to make it easier to read
fig = plt.figure(figsize=(10,7.5))
sns.heatmap(corr_matrix,
            cmap='seismic',
            linewidths=0.75,
            linecolor='black',
            cbar=True,
            vmin=-1,
            vmax=1,
            annot=True,
            annot_kws={'size':8, 'color':'black'})
plt.tick_params(labelsize=10, rotation = 45)
plt.title('Correlation Plot', size = 14);
```

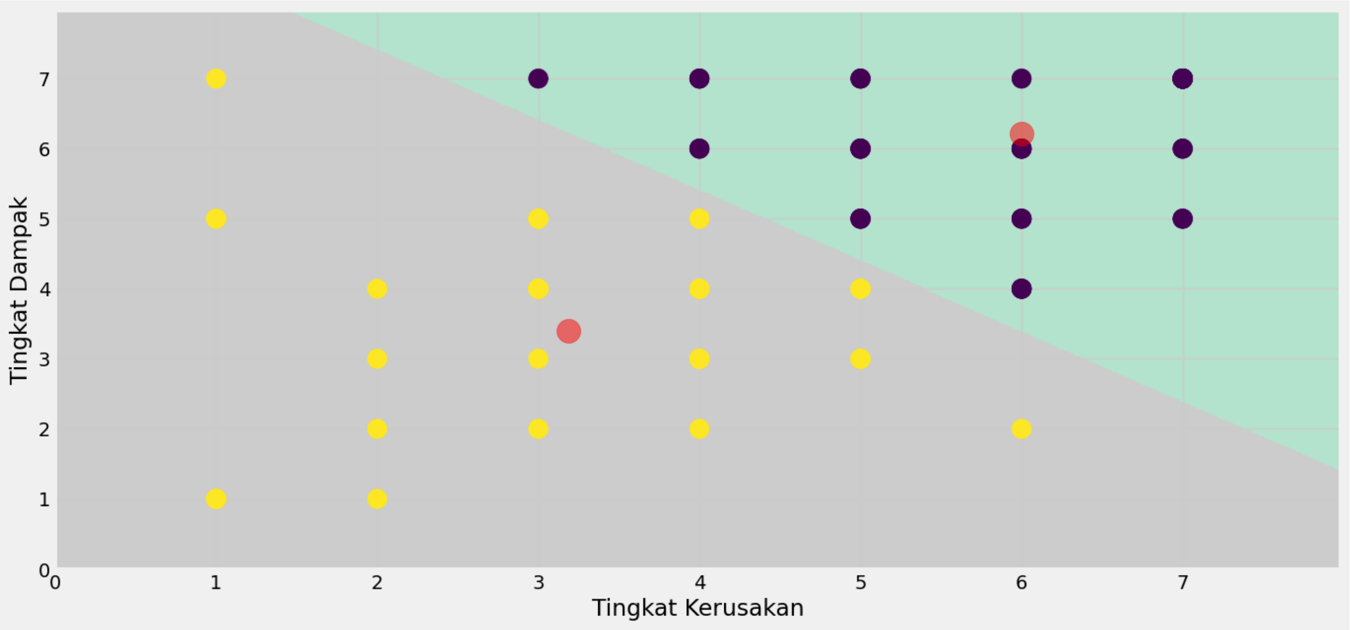


```
# rancang K-Means untuk spending score vs annual income
# Kmeans, menentukan jumlah kluster dengan elbow
X1 = df[['Tingkat Kerusakan' , 'Tingkat Dampak']].iloc[: , :].values
inertia = []
for n in range(1 , 11):
  algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10 ,max_iter=300,
                      random_state= 111) )
  algorithm.fit(X1)
  inertia.append(algorithm.inertia_)


# plot elbow
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```

```
# bangun K-Means
algorithm = (KMeans(n_clusters = 2 ,init='k-means++', n_init = 10 ,max_iter=300,
                    tol=0.0001, random_state= 111 , algorithm='elkan') )
algorithm.fit(X1)
labels2 = algorithm.labels_
centroids2 = algorithm.cluster_centers_
```



```
# siapkan data untuk plot dan imshow
labels2 = algorithm.labels_
centroids2 = algorithm.cluster_centers_
step = 0.02
x_min, x_max = X1[:, 0].min() - 1, X1[:, 0].max() + 1
y_min, y_max = X1[:, 1].min() - 1, X1[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, step), np.arange(y_min, y_max, step))
Z1 = algorithm.predict(np.c_[xx.ravel(), yy.ravel()]) # array diratakan 1D


plt.figure(1 , figsize = (15 , 7) )
plt.clf()
Z1 = Z1.reshape(xx.shape)
plt.imshow(Z1 , interpolation='nearest',
           extent=(xx.min(), xx.max(), yy.min(), yy.max()),
           cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')
plt.scatter( x ='Tingkat Kerusakan'  ,y = 'Tingkat Dampak' , data = df , c = labels2 ,s = 200 )
plt.scatter(x = centroids2[: , 0] , y = centroids2[: , 1] , s = 300 , c = 'red' ,
            alpha = 0.5)
plt.ylabel('Tingkat Dampak') , plt.xlabel('Tingkat Kerusakan')
plt.show()
```



```
# coba prediksi
data = [[6, 7],[6, 4], [3, 5]]
print(data)
print(algorithm.predict(data))
```

```
[[6, 7], [6, 4], [3, 5]]
[0 0 1]
```