**Model K-mean**

**CLUSTERING SURVIVAL TITANIG PASSANGER MENGGUNAKAN K-MEAN ALGORITM**

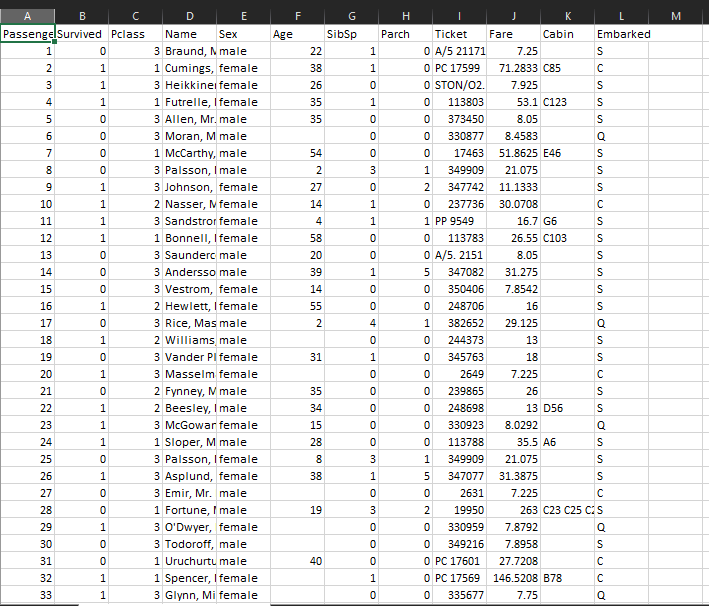
# PENDAHULUAN

Titanic data passanger adalah data penumpang yang menggunakan kapal titanic dari mulai identitas penumpang (nama, jenis kelamin,usia), pasanger id, bagasi,dan kelas penumpang.

# METODOLOGI

* 1. Dataset

Metode yang pertama dilakukan adalah pemilihan dataset dari sumber kaagle. Berikut gambaran dataset:



Gambar 1.1 Passanger dataset titanic

Kemudian akan dilakukan percobaan pengolahan data dengan menggunakan algoritma k-mean sesuai dengan teori pembelajaran pertemuan ke-8.

* 1. Proses data

Tools yang digunakan adalah python dengan beberapa library nya. Berikut scrip pengolahan datanya:

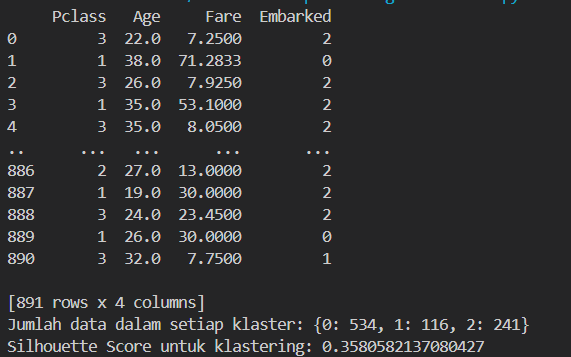
|  |
| --- |
| from sklearn.preprocessing import LabelEncoder  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from sklearn.decomposition import PCA  from sklearn.preprocessing import LabelEncoder, StandardScaler  import seaborn as sns  from sklearn.cluster import KMeans  from sklearn.metrics import silhouette\_score  # 1. Load dataset sumber kaagle  df = pd.read\_csv('passanger titanic train data.csv') # Ubah 'filename.csv' sesuai dengan nama file Anda  # 2. hilangkan data yang tidak dipakai  df = df.drop(['Name','Ticket','PassengerId','Cabin','Survived','SibSp','Parch','Sex'],axis=1)  # 3. Mengganti nilai NaN dalam kolom yg terdapat Nan dengan nilai tertentu untuk setiap kolom  replacement\_values = { 'Fare': df['Fare'].median(),}  df.fillna(replacement\_values, inplace=True)  # 4. fungsi replace null pada kolom age  def impute\_train\_age(cols):  Age = cols.iloc[0]  Pclass = cols.iloc[1]  if pd.isnull(Age):  if Pclass == 1:  return 37  elif Pclass == 2:  return 29  else:  return 24  else:  return Age  df['Age'] = df[['Age','Pclass']].apply(impute\_train\_age,axis=1)  label = LabelEncoder()  data\_column = ['Embarked']  for column in data\_column:  df[column] = label.fit\_transform(df[column])  print(df)  features = df  # 5 Normalisasi fitur-fitur  scaler = StandardScaler()  X\_scaled = scaler.fit\_transform(features)  # Metode Elbow untuk menentukan jumlah klaster optimal  inertia = []  K = range(1, 11)  for k in K:  kmeans = KMeans(n\_clusters=k, random\_state=0)  kmeans.fit(X\_scaled)  inertia.append(kmeans.inertia\_)  # # Plot inertia dg jumlah klaster  plt.plot(K, inertia, 'bo-')  plt.xlabel('Jumlah Klaster')  plt.ylabel('Inertia')  plt.title('Metode Elbow')  plt.show()  # PCA untuk reduksi  pca = PCA(2)  X\_pca = pca.fit\_transform(X\_scaled)  # # Menerapkan K-Means dengan jumlah klaster yang dipilih (misalnya 4)  kmeans = KMeans(n\_clusters=3, random\_state=0)  y\_kmeans = kmeans.fit\_predict(X\_scaled)  # Plot cluster  plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=y\_kmeans, s=50, cmap='viridis')  centers = pca.transform(kmeans.cluster\_centers\_)  plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75, marker='X')  plt.title('Hasil Klastering dengan K-Means')  # plt.xlabel('Komponen Utama 1')  # plt.ylabel('Komponen Utama 2')  plt.show()  # # Menampilkan jumlah data dalam setiap klaster  unique, counts = np.unique(y\_kmeans, return\_counts=True)  print(f'Jumlah data dalam setiap klaster: {dict(zip(unique, counts))}')  # # Menambahkan hasil klastering ke dataframe asli  df['Cluster'] = y\_kmeans  # Membuat plot distribusi  for feature in features:  plt.figure(figsize=(10, 6))  sns.boxplot(x='Cluster', y=feature, data=df)  plt.title(f'Distribusi {feature}')  plt.show()  # Silhouette Score  silhouette\_avg = silhouette\_score(X\_scaled, y\_kmeans)  print(f'Silhouette Score untuk klastering: {silhouette\_avg}') |

# Intepretasi output

Dari prosesing data diatas, beberapa kolom yang diambil diantaranya:

1. Pclass : tiket class (kelas 1,kelas 2 dan kelas 3)
2. Age : usia penumpang
3. Fare : tarif penumpang
4. Embarked : Pelabuhan naiknya penumpang (0 = Cherbourg, 1 = Queenstown, 2 = Southampton)

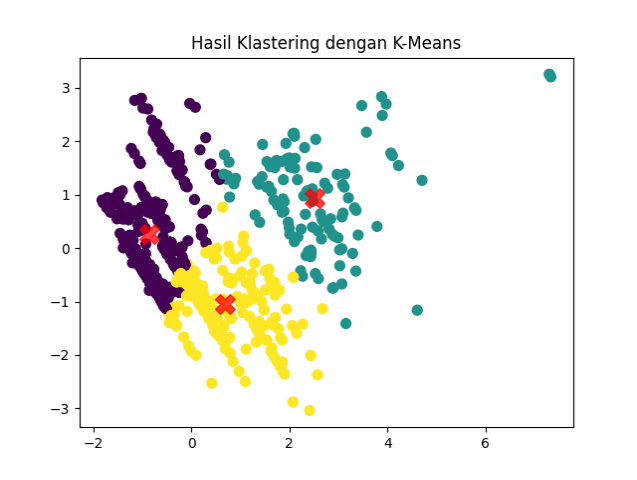
Pemilihan nilai k bada pengolahan data kali ini yaitu k = 3, artinya akan dilakukan tiga cluster dari data diatas.



Gambar 1.2 output data

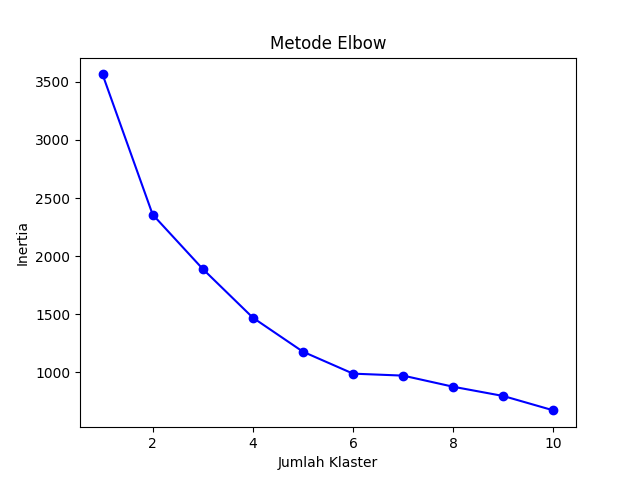
Dari output diatas bisa di dapat, Dari data cluster 1-3 itu ada penyeberan berbeda-beda, diantaranya cluster 1 sebanyak 534 data, cluster 2 sebanyak 116 data dan cluster 3 sebanyak 241 dengan jumlah data 891.

Nilai silhouette Score 0.35



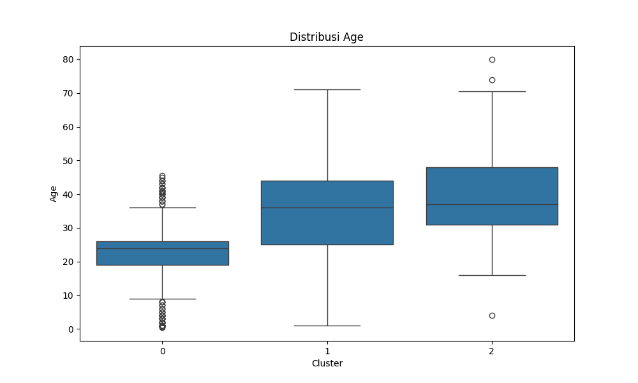
Gambar 1.3 klastering k-mean

Dari gambar diatas menggambarkan pengclusteran data dengan nilai x berwarana merah sebagai centroid(pusat data)



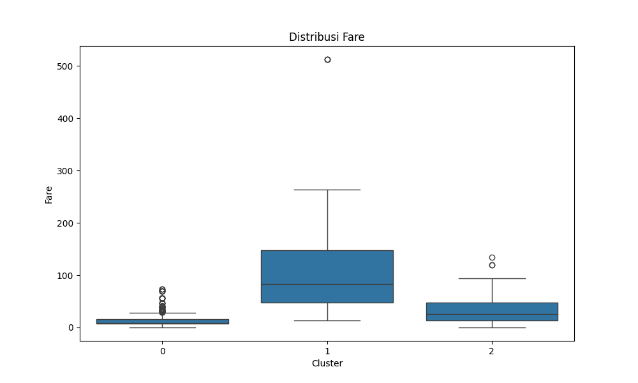
Gambar 1.4 Grafik elbow

Untuk menentukan jumlah klaster yang optimal, kita harus memilih nilai k pada "siku" yaitu titik setelah distorsi/inersia mulai menurun secara linier. Jadi untuk data yang diberikan, kita simpulkan bahwa jumlah klaster yang optimal untuk data tersebut adalah 6.



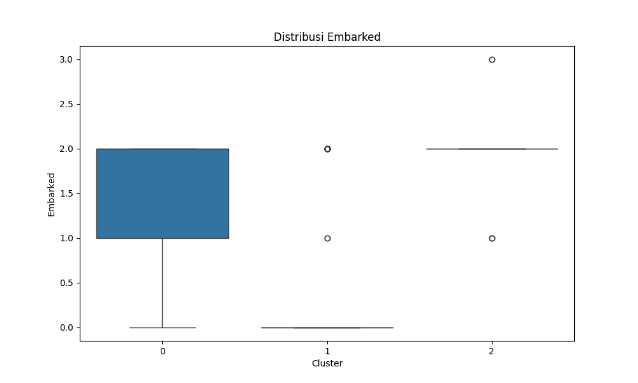
Gambar 1.4 Grafik Distribusi Age (usia)

Dari grafik distribusi diatas bisa dilihat usia penumpang untuk cluster 1 berada dirange usia 20-30, cluster 2 dirange 25-40 dan cluster 3 dirange di 35-40.



Gambar 1.4 Grafik Distribusi Fare (tarif penumpang)

Dari grafik distribusi diatas bisa dilihat tarif penumpang pengclusteran di cluster 2 lebih banyak, artinya tarif range 50-100



Gambar 1.4 Grafik Distribusi Embarked (Pelabuhan tempat naiknya penumpang)

Dari grafik distribusi diatas bisa dilihat tempat penumpang naik kapal titanic tersebut untuk cluster 2 lebih banyak dibanding dengan cluster lainnya.

# KESIMPULAN DAN SARAN

Didapat Nilai silhouette Score 0.35 artinya pengclusteran data kurang baik, dan hal ini dari referensi yang saya dapat bisa dilakukan pengclusteran dengan iterasi beberap kali agar penclusteran lebih optimah, juga bisa dengan metode pendekatan berbeda seperti selain Euclid distance ada manhattan distance dan minkawski distance.

Dari pengolahan data k-mean untuk data titanic pessanger ini lebih akan optimal jika nilai k itu sama dengan 6 sesuai dengan grafik elbow yg didapat.

Jadi bisa disimpulkan secara sederhana untuk hasil pengclusteran data pada data titanic passanger masih kurang baik.

# DAFTAR PUSTAKA

[1] https://www.kaggle.com/code/mrisdal/exploring-survival-on-the-titanic#introduction