LAPORAN PEMBELAJARAN MESIN TAHAP 1

Disusun Untuk Memenuhi Tugas Besar Mata Kuliah Pembelajaran Mesin

Dosen Pengampu : Agus Hartoyo, Ph.D

Disusun Oleh :

Naufal Haritsah Luthfi (1301194073)

PRODI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2021

Daftar Isi

1. **Task Clustering**
   a. Formulasi Masalah
      - Melakukan pengecekan data terlebih dahulu untuk melihat jenis serta isi variabel pada data
      - Merubah nilai yang kosong pada variabel tersebut.
      - Melakukan normalisasi pada dataset.
      - Menghilangkan outlier yang terdapat pada dataset.
      - Melihat korelasi antar variabel pada dataset menggunakan heatmap correlation.
      - Melihat penyebaran variabel data yang ditentukan menggunakan scatter plot sebelum dilakukan proses clustering
      - Melakukan proses clustering pada variabel tertentu pada dataset menggunakan metode K-Means.
      - Melakukan proses elbow method sebagai evaluasi dalam mencari nilai k terbaik terhadap proses clustering.
      - Melakukan eksperimen clustering dataset dengan nilai k terbaik setelah dilakukan evaluasi.

   b. Eksplorasi Dan Persiapan Data
      - Import Library

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

      Langkah pertama yang dilakukan adalah mengimport library yang akan digunakan yaitu library pandas, seaborn, numpy, matplotlib.pyplot, dan random.
        - Library Pandas, digunakan sebagai alat bantu statistik dalam pengolahan data
        - Library Seaborn, digunakan sebagai library tambahan dalam visualisasi data
        - Library Numpy, digunakan sebagai alat bantu operasi komputasi tipe data numerik
        - Library Random, digunakan sebagai alat untuk mencari nilai random
        - Library sklearn.cluster, yang digunakan hanya untuk evaluasi model.

- Import Dataset

```
[ ]  !gdown --id 1hsffcmhRqtm6HuXJBl_JrIqNT8R7smq7
```

Langkah selanjutnya adalah mengimport dataset yang terdapat pada google drive dengan menggunakan perintah !gdown serta letak alamat dataset tersebut.

- Data Understanding

```
data.describe()
```

	id	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
count	285831.000000	271617.000000	271427.000000	271525.000000	271602.000000	271262.000000	271532.000000	271839.000000	285831.000000
mean	142916.000000	38.844336	0.997848	26.405410	0.458778	30536.683472	112.021567	154.286302	0.122471
std	82512.446734	15.522487	0.046335	13.252714	0.498299	17155.000770	54.202457	83.694910	0.327830
min	1.000000	20.000000	0.000000	0.000000	0.000000	2630.000000	1.000000	10.000000	0.000000
25%	71458.500000	25.000000	1.000000	15.000000	0.000000	24398.000000	29.000000	82.000000	0.000000
50%	142916.000000	36.000000	1.000000	28.000000	0.000000	31646.000000	132.000000	154.000000	0.000000
75%	214373.500000	49.000000	1.000000	35.000000	1.000000	39377.750000	152.000000	227.000000	0.000000
max	285831.000000	85.000000	1.000000	52.000000	1.000000	540165.000000	163.000000	299.000000	1.000000

Pada bagian ini, pengecekan data dilakukan untuk melihat nilai dengan perintah data.describe(). Perintah data.describe akan menampilkan count, mean, std, min, kuartil, dan max.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 285831 entries, 0 to 285830
Data columns (total 12 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   id                 285831 non-null  int64
 1   Jenis_Kelamin      271391 non-null  object
 2   Umur               271617 non-null  float64
 3   SIM                271427 non-null  float64
 4   Kode_Daerah        271525 non-null  float64
 5   Sudah_Asuransi     271602 non-null  float64
 6   Umur_Kendaraan     271556 non-null  object
 7   Kendaraan_Rusak    271643 non-null  object
 8   Premi              271262 non-null  float64
 9   Kanal_Penjualan    271532 non-null  float64
 10  Lama_Berlangganan  271839 non-null  float64
 11  Tertarik           285831 non-null  int64
dtypes: float64(7), int64(2), object(3)
memory usage: 26.2+ MB
```

Selanjutnya data.info() akan menampilkan Non-Null, Count, dan Dtype pada data tersebut.

- Eksplorasi dan Persiapan Data

```
data = data.drop(['id', 'Tertarik'], axis=1)
data.sample(5)
```

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan
284545	Wanita	25.0	1.0	46.0	1.0	< 1 Tahun	Tidak	24042.0	152.0	139.0
201426	Wanita	61.0	1.0	3.0	1.0	1-2 Tahun	Tidak	14214.0	124.0	142.0
205564	Pria	50.0	1.0	28.0	0.0	1-2 Tahun	NaN	40059.0	26.0	181.0
20186	Pria	46.0	1.0	35.0	0.0	> 2 Tahun	Pernah	31637.0	26.0	220.0
117224	NaN	46.0	1.0	8.0	0.0	1-2 Tahun	Tidak	2630.0	157.0	235.0

Pada langkah eksplorasi dan persiapan data, terdapat proses drop data karena tidak digunakan yaitu data id dan data tertarik. Kemudian menampilkan sample data sebanyak 5 data

- Pengecekan Data Kosong

```
data['SIM'].fillna(method = 'ffill', inplace = True)
data['Kode_Daerah'] = data['Kode_Daerah'].replace(np.NaN, 28.0)
data['Jenis_Kelamin'].fillna(method = 'ffill', inplace = True)
data['Umur'] = data['Umur'].replace(np.NaN, data['Umur'].mean())
data['Sudah_Asuransi'].fillna(method = 'ffill', inplace = True)
data['Umur_Kendaraan'] = data['Umur_Kendaraan'].replace(np.NaN, "1-2 Tahun")
data['Kendaraan_Rusak'].fillna(method = 'ffill', inplace = True)
data['Premi'] = data['Premi'].replace(np.NaN, data['Premi'].mean())
data['Kanal_Penjualan'] = data['Kanal_Penjualan'].replace(np.NaN, 152.0)
data['Lama_Berlangganan'] = data['Lama_Berlangganan'].replace(np.NaN, data['Lama_Berlangganan'].mean())
```

Source code diatas adalah merubah nilai kosong pada kolom data.

- Untuk kolom data SIM menggunakan nilai tetangganya.
- Untuk kolom data Kode_Daerah menggunakan nilai modus pada data tersebut.
- Untuk kolom Jenis_Kelamin menggunakan nilai tetangganya.
- Untuk kolom Umur menggunakan nilai rata-rata pada data tersebut.
- Untuk kolom Sudah_Asuransi menggunakan nilai tetangganya.
- Untuk kolom Umur_Kendaraan menggunakan nilai modus pada data tersebut.
- Untuk kolom Kendaraan_Rusak menggunakan nilai tetangganya.
- Untuk kolom Premi menggunakan nilai rata-rata pada data tersebut.
- Untuk kolom Kanal_Penjualan menggunakan nilai modus pada data tersebut.
- Untuk kolom Lama_Berlangganan menggunakan nilai rata-rata pada data tersebut.

```
data.isna().sum()

Jenis_Kelamin         0
Umur                  0
SIM                   0
Kode_Daerah           0
Sudah_Asuransi        0
Umur_Kendaraan        0
Kendaraan_Rusak       0
Premi                 0
Kanal_Penjualan       0
Lama_Berlangganan     0
dtype: int64
```

Kemudian data dicek lagi apakah ada kolom yang kosong pada data tersebut.

- Normalisasi Data

```
data['Jenis_Kelamin'] = data['Jenis_Kelamin'].replace("Wanita", 1)
data['Jenis_Kelamin'] = data['Jenis_Kelamin'].replace("Pria", 0)

data['Kendaraan_Rusak'] = data['Kendaraan_Rusak'].replace("Pernah", 1)
data['Kendaraan_Rusak'] = data['Kendaraan_Rusak'].replace("Tidak", 0)

data['Umur_Kendaraan'] = data['Umur_Kendaraan'].replace("< 1 Tahun", 1)
data['Umur_Kendaraan'] = data['Umur_Kendaraan'].replace("1-2 Tahun", 2)
data['Umur_Kendaraan'] = data['Umur_Kendaraan'].replace("> 2 Tahun", 3)
```

Proses diatas adalah merubah kategori data kategorikal menjadi numerikal.

- Untuk kolom Jenis_Kelamin, jika Wanita di set dengan angka 1, Pria di set dengan angka 0.
- Untuk kolom Kendaraan_Rusak, jika Pernah di set dengan angka 1, Tidak di set dengan angka 0.
- Untuk kolom Umur_Kendaraan, jika < 1 tahun di set dengan angka 1, 1-2 tahun di set dengan angka 2, > 2 tahun di set dengan angka 3.

```
data.head()
```

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan
0	1.0	30.0	1.0	33.0	1.0	1	0.0	28029.0	152.0	97.0
1	0.0	48.0	1.0	39.0	0.0	3	1.0	25800.0	29.0	158.0
2	0.0	21.0	1.0	46.0	1.0	1	0.0	32733.0	160.0	119.0
3	1.0	58.0	1.0	48.0	0.0	2	0.0	2630.0	124.0	63.0
4	0.0	50.0	1.0	35.0	0.0	3	0.0	34857.0	88.0	194.0

Berikut merupakan hasil luaran kolom yang telah dirubah kategorikal menjadi numerikal.

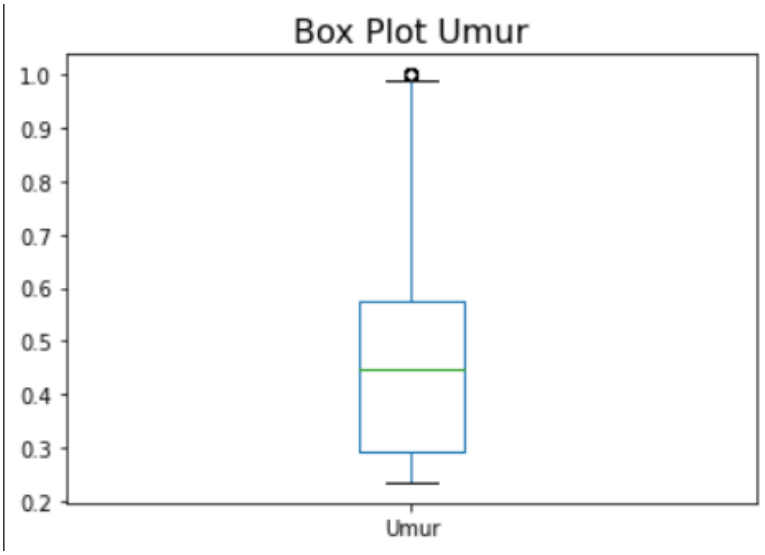
```
#MAX Scaling
for column in data.columns:
    data[column] = data[column]  / data[column].abs().max()

data.sample()
```

	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan
55326	1.0	0.552941	1.0	0.153846	0.0	0.666667	1.0	0.073555	0.760736	0.725753

Max Scaling merupakan teknik normalisasi data yang digunakan pada dataset ini. Cara kerja teknik ini adalah dengan mengiterasi setiap kolom yang ada pada dataset, kemudian mengganti nilainya dengan hasil pembagian kolom terhadap nilai maksimum serta absolut pada kolom tersebut.

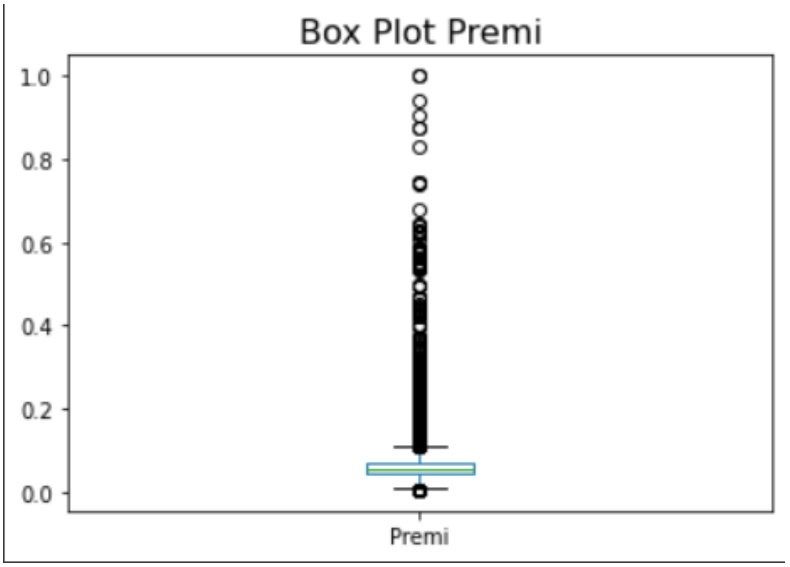
- Pengecekan Outlier
  - Sebelum Proses Drop Outlier Menggunakan Metode IQR

```
data['Umur'].plot(kind='box', figsize=(6, 4))

plt.title('Box Plot Umur', size=16)
plt.show()
```



Berikut merupakan source code untuk menampilkan Box Plot Umur. Terdapat outlier pada Box Plot Umur.

```
data['Premi'].plot(kind='box', figsize=(6, 4))

plt.title('Box Plot Premi', size=16)
plt.show()
```

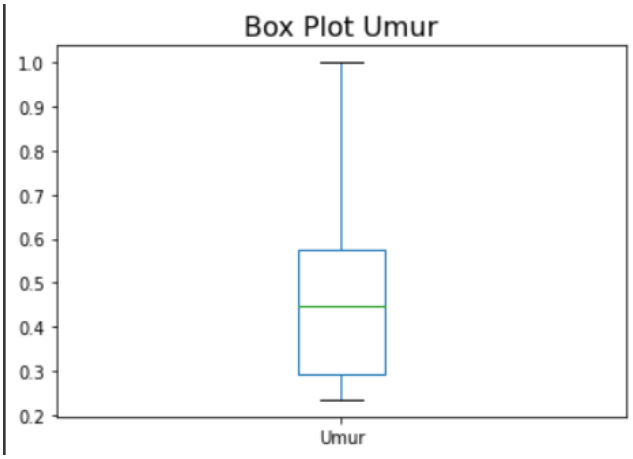


Berikut merupakan source code untuk menampilkan Box Plot Premi. Terdapat outlier pada Box Plot Premi.

- Sesudah Proses Drop Outlier Menggunakan Metode IQR

```
data_umur = data['Umur']
q1 = np.percentile(data_umur, 25)
q3 = np.percentile(data_umur, 75)

IQR = q3-q1
bawah = q1-(1.5*IQR)
atas = q3+(1.5*IQR)

for i in range(len(data['Umur'])):
   if data['Umur'][i] < bawah:
     data['Umur'][i] = bawah
   if data['Umur'][i] > atas:
     data['Umur'][i] = atas
```

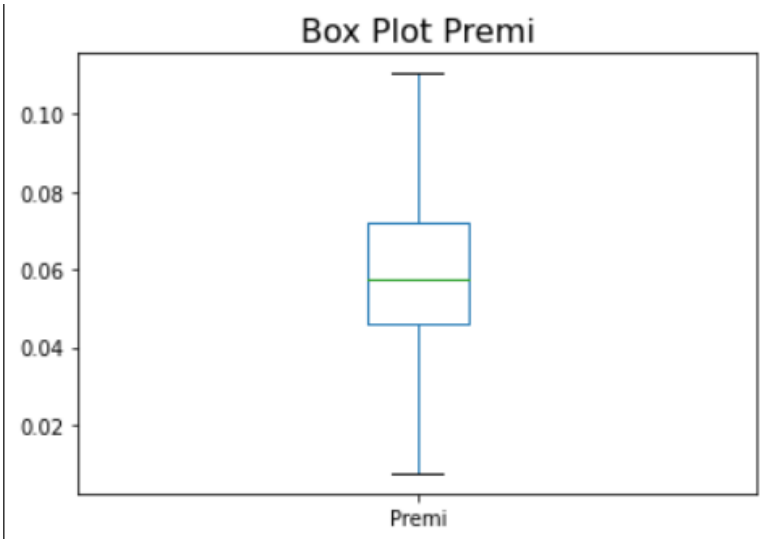
```
data['Umur'].plot(kind='box', figsize=(6, 4))

plt.title('Box Plot umur', size=16)
plt.show()
```



Kemudian dilakukan proses drop outlier menggunakan metode IQR. Cara kerja metode IQR adalah dengan cara mengurangi Q3 dengan Q1. Dengan menggunakan metode IQR, outlier dapat ditentukan melalui suatu nilai batas yang ditentukan. Sehingga data yang kurang dari batas bawah ataupun data yang melebihi batas atas akan disebut dengan outlier.

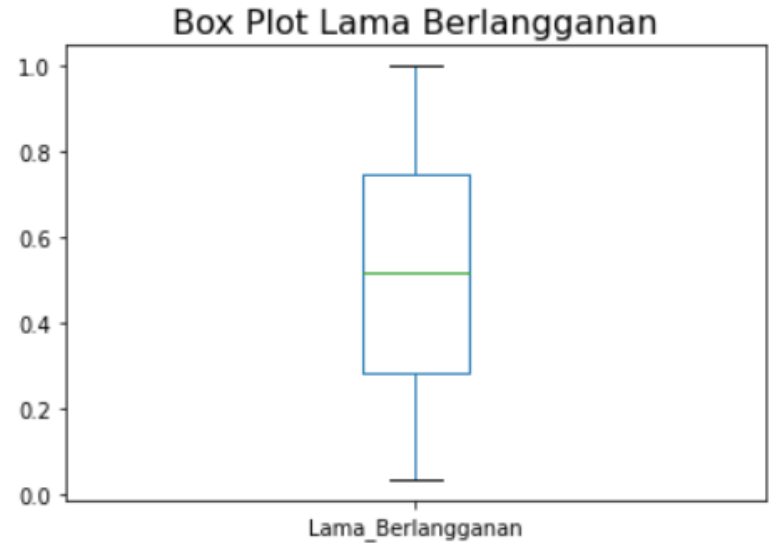
```
data_premi = data['Premi']
q1 = np.percentile(data_premi, 25)
q3 = np.percentile(data_premi, 75)

IQR = q3-q1
bawah = q1-(1.5*IQR)
atas = q3+(1.5*IQR)

for i in range(len(data['Premi'])):
   if data['Premi'][i] < bawah:
     data['Premi'][i] = bawah
   if data['Premi'][i] > atas:
     data['Premi'][i] = atas
```

```
data['Premi'].plot(kind='box', figsize=(6, 4))

plt.title('Box Plot Premi', size=16)
plt.show()
```



Berikut merupakan Box Plot Premi setelah diterapkan metode IQR.

```
data['Lama_Berlangganan'].plot(kind='box', figsize=(6, 4))

plt.title('Box Plot Lama Berlangganan', size=16)
plt.show()
```

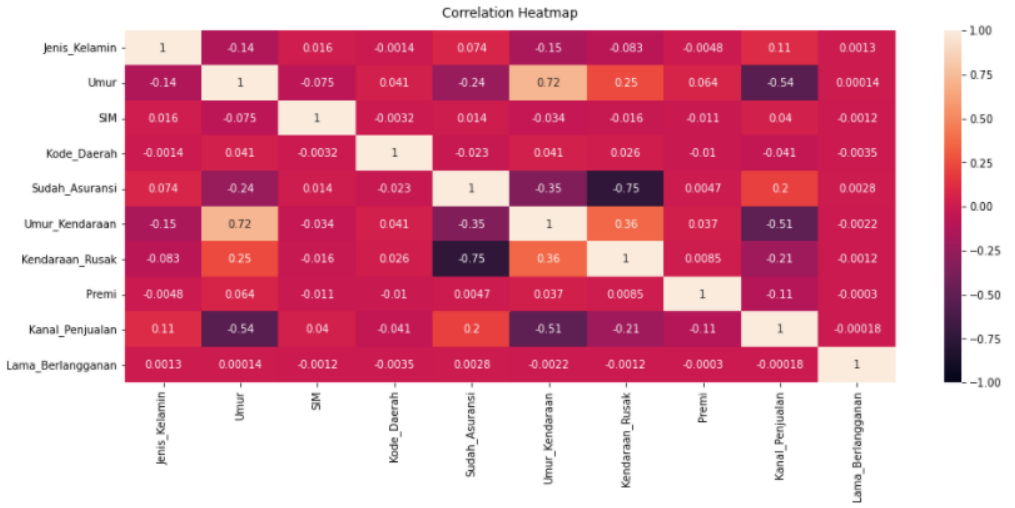


Berikut merupakan Box Plot Lama Berlangganan setelah diterapkan metode IQR.

c. Pemodelan

- Pengecekan Korelasi

```
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(data.corr(), vmin=-1, vmax=1, annot=True)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=12);
```

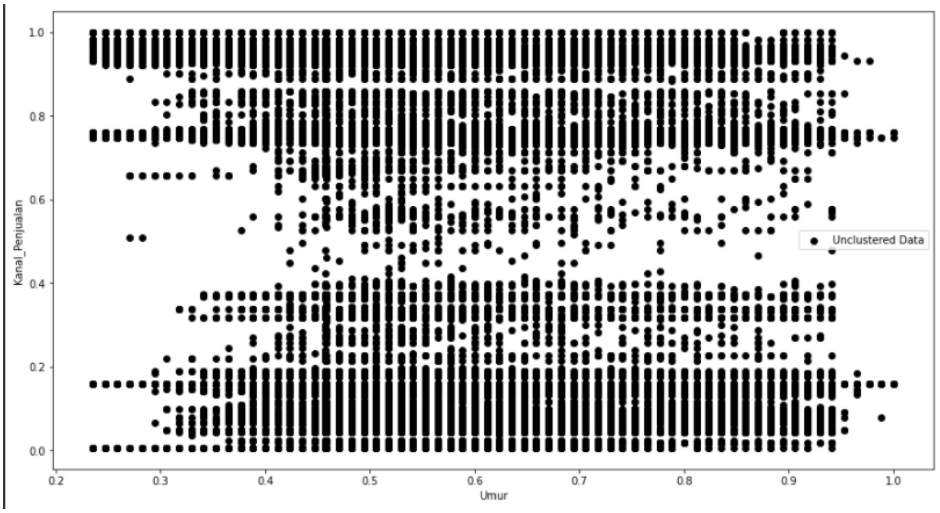


Pengecekan korelasi menggunakan heatmap diperlukan untuk mengecek variabel-variabel yang memiliki korelasi kuat terhadap variabel-variabel tersebut. Nilai korelasi yang paling tinggi positif yaitu 0.72 antara variabel Umur dengan variabel Umur_Kendaraan. Serta nilai korelasi yang paling tinggi negatif yaitu -0.75 antara variabel Kendaraan_Rusak dengan Sudah_Asuransi.

- Pemodelan Data
  - Sebelum Clustering

```
plt.figure(figsize=(15,8))
plt.scatter(data['Umur'], data['Kanal_Penjualan'], color='black', label="Unclustered Data")
plt.legend()
plt.xlabel("Umur")
plt.ylabel("Kanal_Penjualan")
plt.show()
```

Berikut merupakan source code untuk menampilkan scatter plot variabel umur terhadap kanal penjualan sebelum dilakukan proses clustering.



Scatter plot diatas merupakan data umur terhadap kanal_penjualan sebelum dilakukan proses clustering.

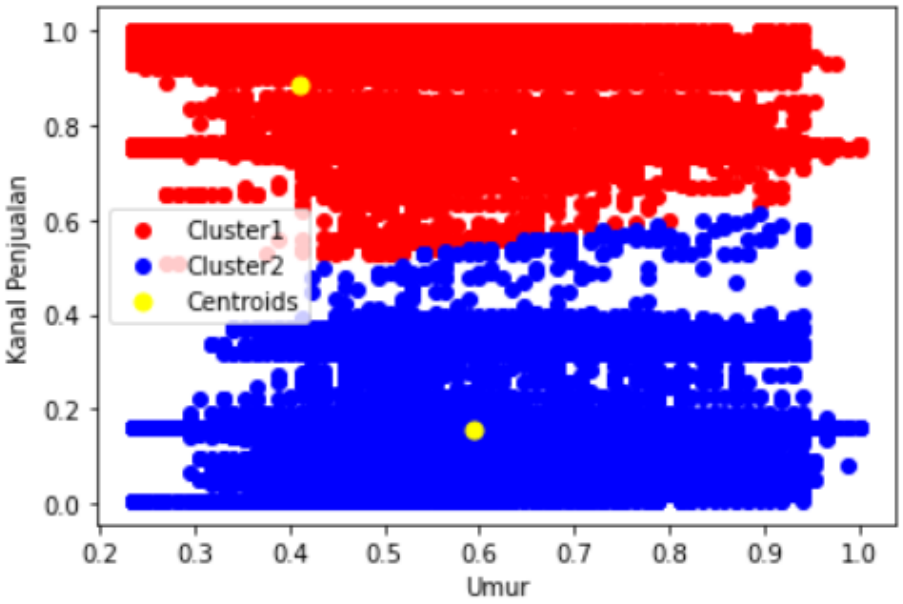
- Setelah Clustering Menggunakan Metode K-Means dengan K = 2

```
def kmeans(K, n_iter):
  Centroids=np.array([]).reshape(n,0)

  for i in range(K):
      rand=random.randint(0,m-1)
      Centroids=np.c_[Centroids,X[rand]]

  Output={}

  EuclidianDistance=np.array([]).reshape(m,0)
  for k in range(K): # melakukan perulangan untuk menghitung jarak setiap data pada dataset
    tempDist=np.sum((X-Centroids[:,k])**2,axis=1)
    EuclidianDistance=np.c_[EuclidianDistance,tempDist]

  C=np.argmin(EuclidianDistance,axis=1)+1

  Y={}
  for k in range(K):
      Y[k+1]=np.array([]).reshape(2,0)
  for i in range(m):
      Y[C[i]]=np.c_[Y[C[i]],X[i]]

  for k in range(K):
      Y[k+1]=Y[k+1].T

  for k in range(K):
      Centroids[:,k]=np.mean(Y[k+1],axis=0)

  for i in range(n_iter):
        EuclidianDistance=np.array([]).reshape(m,0)
        for k in range(K):
```

Metode yang digunakan dalam clustering atau pengelompokkan data pada dataset ini menggunakan metode K-Means. Cara kerja source code tersebut adalah melakukan perulangan hingga centroid yang terambil bernilai sama dengan centroid sebelumnya. Metode euclidean distance akan melakukan perulangan tiap-tiap data pada dataset untuk dicek jaraknya terhadap centroid. Jarak-jarak sebelumnya akan disimpan dalam array cluster i. Sedangkan nilai centroidnya akan menjadi perhitungan euclidean distance berikutnya pada dataset.

```
K = 2
Centroids, Output = kmeans(K, 1)

color=['red','blue','green','cyan']
labels=['Cluster1','Cluster2','Cluster3','Cluster4']
for k in range(K):
    plt.scatter(Output[k+1][:,0],Output[k+1][:,1],c=color[k],label=labels[k])
plt.scatter(Centroids[0,:],Centroids[1,:],s=50,c='yellow',label='Centroids')
plt.xlabel('Umur')
plt.ylabel('Kanal Penjualan')
plt.legend()
plt.show()
```

Untuk pemodelan data, nilai k diset menjadi 2. Berikut merupakan source code untuk menampilkan scatter plot variabel umur terhadap kanal penjualan setelah dilakukan proses clustering.



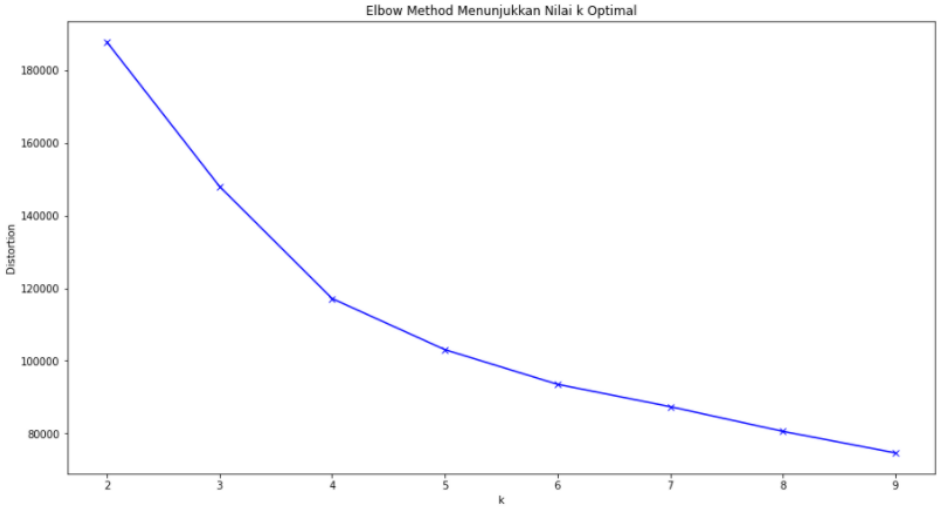
Dapat dilihat bahwa terdapat 2 kelompok yaitu cluster 1, cluster 2 dan kedua centroidsnya.

d. Evaluasi

- Elbow Method

```
distortions = []
k = range(2, 10)
for i in k:
    kmeanModel = KMeans(n_clusters=i)
    kmeanModel = kmeanModel.fit(data)
    distortions.append(kmeanModel.inertia_)

plt.figure(figsize=(15,8))
plt.plot(k, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('Elbow Method Menunjukkan Nilai k Optimal')
plt.show()
```

Proses Elbow Method ini untuk mencari nilai k optimal pada pengelompokkan data. Pengelompokkan data ditentukan oleh nilai k yang dimasukkan. Pada source code tersebut nilai k didefinisikan dari 2 hingga 10. Pada proses elbow method ini digunakan library sklearn.cluster untuk menentukan k yang optimal evaluasi model.



Penurunan nilai setelah k=4 sudah tidak signifikan. Maka nilai k optimal yang akan dipilih adalah 4.

e. Eksperimen

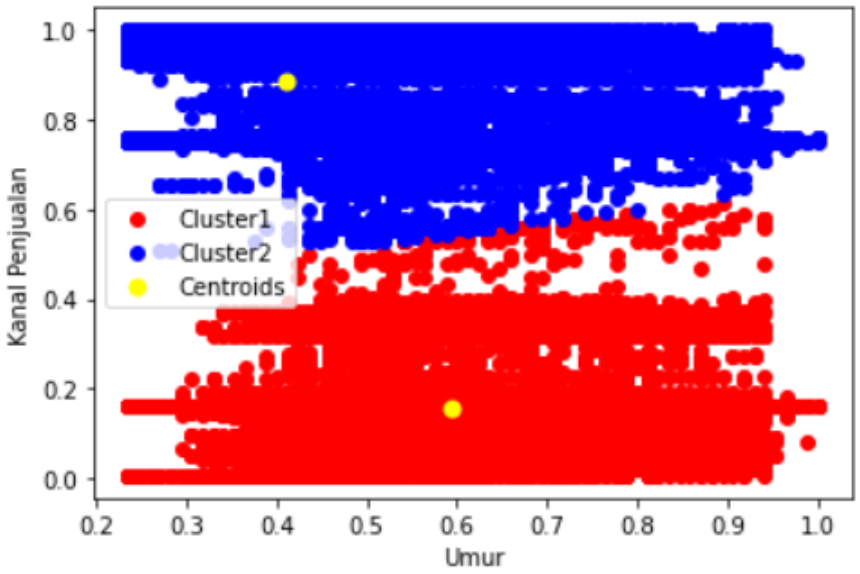
Eksperimen yang dilakukan adalah melakukan clustering dengan k random dan merubah nilai k dengan k optimal sesuai yang didapat pada Elbow Method.

- Clustering Dengan k Random

```
K = 2
Centroids, Output = kmeans(K, 1)
```

```
color=['red','blue','green','cyan']
labels=['Cluster1','Cluster2','Cluster3','Cluster4']
for k in range(K):
    plt.scatter(Output[k+1][:,0],Output[k+1][:,1],c=color[k],label=labels[k])
plt.scatter(Centroids[0,:],Centroids[1,:],s=50,c='yellow',label='Centroids')
plt.xlabel('Umur')
plt.ylabel('Kanal Penjualan')
plt.legend()
plt.show()
```

Untuk pemodelan data, nilai k diset menjadi 2. Berikut merupakan source code untuk menampilkan scatter plot variabel umur terhadap kanal penjualan setelah dilakukan proses clustering.



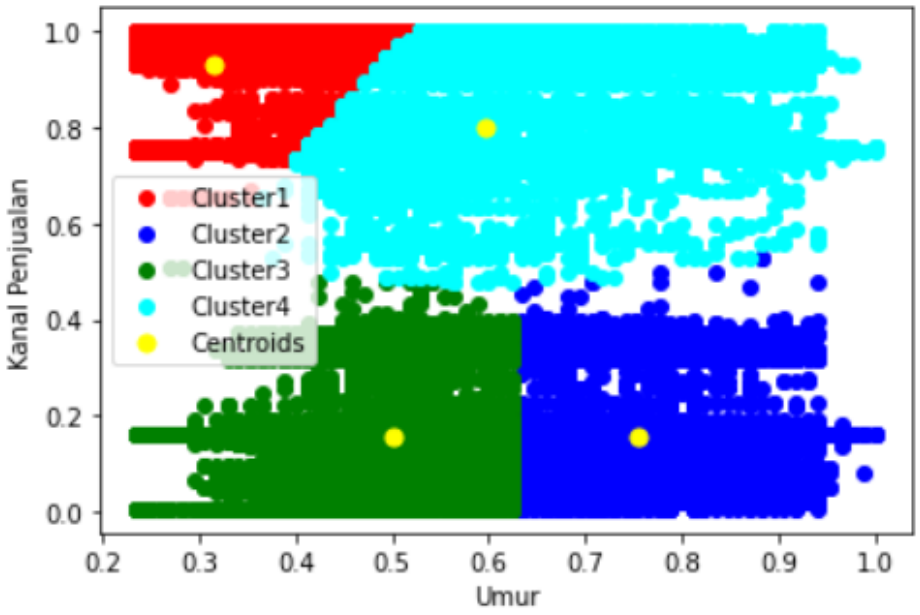
Dapat dilihat bahwa terdapat 2 kelompok yaitu cluster 1, cluster 2 dan kedua centroidsnya.

- Clustering Dengan k Optimal

```
n_iter=1
K= 4

Centroids, Output = kmeans(K, n_iter)
```

Berikut merupakan code untuk melakukan clustering ulang menggunakan k optimal yang sudah ditentukan pada evaluasi model. Pada proses eksperimen ini memanggil fungsi k-means kembali.

```
color=['red','blue','green', 'cyan']
labels=['Cluster1','Cluster2','Cluster3','Cluster4']
for k in range(K):
    plt.scatter(Output[k+1][:,0],Output[k+1][:,1],c=color[k],label=labels[k])
plt.scatter(Centroids[0,:],Centroids[1,:],s=50,c='yellow',label='Centroids')
plt.xlabel('Umur')
plt.ylabel('Kanal Penjualan')
plt.legend()
plt.show()
```



Dapat dilihat bahwa terdapat 4 cluster yaitu cluster 1, cluster 2 , cluster 3 dan cluster 4 serta keempat centroidsnya.

f. Kesimpulan

- Normalisasi data perlu dilakukan agar pada proses pengelompokkan data serta saat melakukan correlation heatmap terdapat perbedaan yang signifikan.
- Metode IQR untuk menghilangkan outlier perlu dilakukan untuk mengatasi terjadinya ketimpangan nilai variabel pada dataset.
- Scatter Plot sangat diperlukan untuk melihat persebaran data pada pemodelan data sebelum nantinya akan dilakukan clustering.
- Elbow Method diperlukan untuk menentukan k optimal. Dimana k optimal merupakan k yang paling tepat untuk clustering pada dataset tersebut.

2. Lampiran

- Google Colab
  https://colab.research.google.com/drive/150woOJnd4tE3zgdlcZVdMPA_Znq-9bLp?usp=sharing

- Link Video Presentasi
  https://youtu.be/0NOHlDb-Kf8

- Link Dataset Setelah Preprocessing
  https://drive.google.com/drive/folders/1r01VK6_QRNvYLu0IASzLlaloDKrgYCOU?usp=sharing