Laporan Tugas Besar
Pembelajaran Mesin
Tahap 1 : Clustering

Kaenova Mahendra **Auditama**
1301190324
CII3C3-IF-43-02

Telkom University

Pendahuluan

Tugas Besar pada Mata Kuliah Pembelajaran Mesin (CII3C3-IF-43-02) merupakan tugas pertama dari dua projek tugas yang ada. Pada tugas ini, saya diminta untuk membuat suatu sistem atau **model yang dapat mengklusterisasi** dari dataset yang disediakan.

Permasalahan

Pada kasus ini kami diberikan suatu dataset terkait ketertarikan pelanggan dengan beberapa atribut-atribut. Data yang diberikan sebesar **285.831 *records***. Adapun beberapa atribut seperti *id, Jenis_Kelamin, Umur, SIM, Kode_Daerah, Sudah_Asuransi, Umur_Kendaraan, Kendaraan_Rusak, Premi, Kanal_Penjualan, Lama_Berlangganan*, dan *Tertarik*.

Dengan data tersebut, akhirnya dipilihlah ***K-means*** sebagai model yang digunakan untuk melakukan pengklusterisasian data

Eksplorasi dan Pra-Pemrosesan Data

```
[ ]  1 df_raw = pd.read_csv("https://raw.githubusercontent.com/kaenova/Malin_Tubes1/main/data/raw/kendaraan_train.csv")
     2 df_raw.head()
```

	id	Jenis_Kelamin	Umur	SIM	Kode_Daerah	Sudah_Asuransi	Umur_Kendaraan	Kendaraan_Rusak	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
0	1	Wanita	30.0	1.0	33.0	1.0	< 1 Tahun	Tidak	28029.0	152.0	97.0	0
1	2	Pria	48.0	1.0	39.0	0.0	> 2 Tahun	Pernah	25800.0	29.0	158.0	0
2	3	NaN	21.0	1.0	46.0	1.0	< 1 Tahun	Tidak	32733.0	160.0	119.0	0
3	4	Wanita	58.0	1.0	48.0	0.0	1-2 Tahun	Tidak	2630.0	124.0	63.0	0
4	5	Pria	50.0	1.0	35.0	0.0	> 2 Tahun	NaN	34857.0	88.0	194.0	0

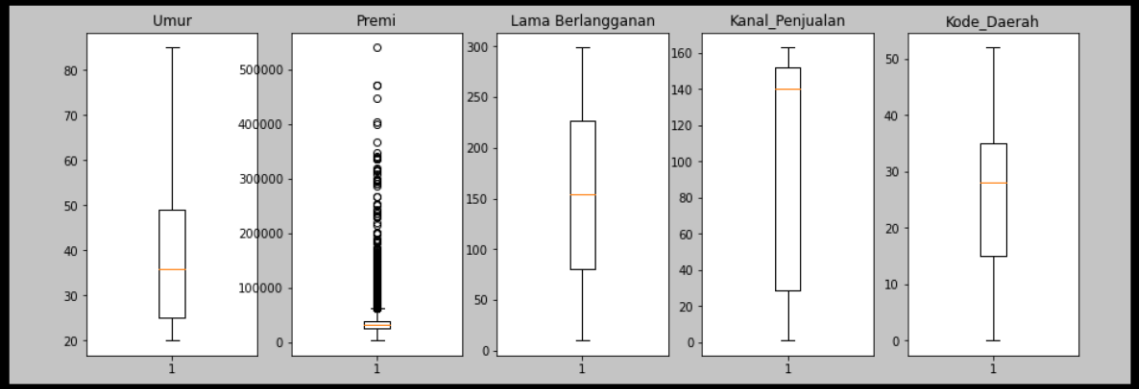
```
[ ]  1 len(df_raw)
```

```
285831
```

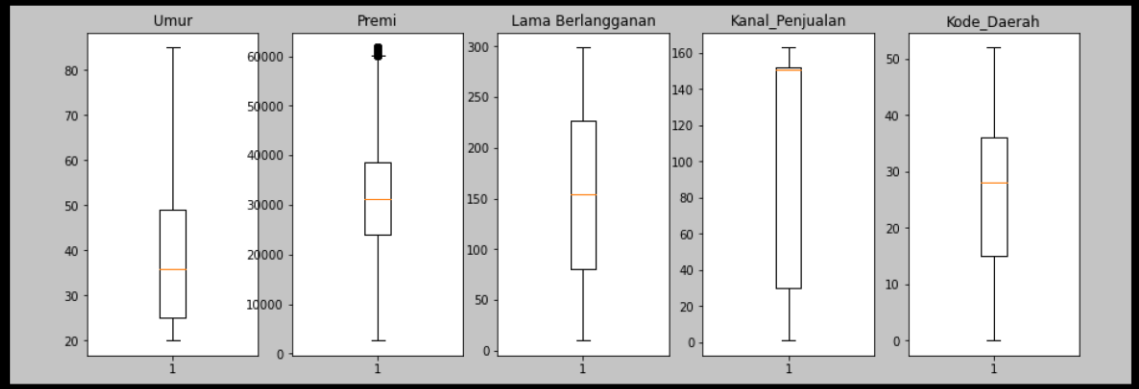
Pertama, lakukan penghilangan data yang terlihat jelas sebagai data categorical seperti *Jenis_Kelamin, SIM, Sudah_Asuransi, Umur_Kendaraan*, dan *Kendaraan_Rusak*

Kedua, melakukan penghilangan *records* jika pada salah satu data ada yang kosong atau *NaN*

Eksplorasi dan Pra-Pemrosesan Data (cont.)



Sebelum Dilakukan Penghilangan Outlier



Setelah Dilakukan Penghilangan Outlier

Ketiga, melakukan penghilangan outlier pada tipe data tertentu. Digunakan metode *interquartile range* sebagai penghilang data outlier dengan ketentuan seperti di bawah:

$$\text{ValidData} \in (x \geq (Q_1 - 1.5 \cdot IQR)) \; and \; (x \leq (Q_3 + 1.5 \cdot IQR))$$

Eksplorasi dan Pra-Pemrosesan Data (cont.)

Melakukan pemeriksaan nilai korelasi terhadap atribut-atribut yang ada.

	id	Umur	Kode_Daerah	Premi	Kanal_Penjualan	Lama_Berlangganan	Tertarik
id	*1.000000*	*0.002691*	*0.000597*	*0.002643*	*-0.001621*	*0.001875*	*0.000203*
Umur	*0.002691*	*1.000000*	*0.044503*	*0.046519*	*-0.574807*	*-0.001055*	*0.108781*
Kode_Daerah	*0.000597*	*0.044503*	*1.000000*	*-0.004068*	*-0.044871*	*-0.003771*	*0.010484*
Premi	*0.002643*	*0.046519*	*-0.004068*	*1.000000*	*-0.105819*	*0.001831*	*0.019686*
Kanal_Penjualan	*-0.001621*	*-0.574807*	*-0.044871*	*-0.105819*	*1.000000*	*0.000017*	*-0.139186*
Lama_Berlangganan	*0.001875*	*-0.001055*	*-0.003771*	*0.001831*	*0.000017*	*1.000000*	*0.001819*
Tertarik	*0.000203*	*0.108781*	*0.010484*	*0.019686*	*-0.139186*	*0.001819*	*1.000000*

Setelah dilakukan eksplorasi dan pra-pemrosesan data, tersisa 166.395 *records* yang siap dimasukkan ke dalam model. Hal ini tidak menjadi masalah karena secara keseluruhan data yang tersisa tetap berjumlah besar.

```
[ ]  1 describe = df_dropna_dropcategorical.describe()
     2 describe
     3 iqr_premi = float(describe["Premi"].loc["75%"] - describe["Premi"].loc["25%"])
     4 q1_bound = float(describe["Premi"].loc["25%"]) - (iqr_premi * 1.5)
     5 q2_bound = float(describe["Premi"].loc["75%"]) + (iqr_premi * 1.5)
     6 final_df = df_dropna_dropcategorical.copy()
     7 final_df.reset_index(drop=True, inplace=True)
     8 final_df = final_df[(final_df["Premi"] > q1_bound) & (final_df["Premi"] < q2_bound)]
     9 len(final_df)

166396
```

Pemodelan

Secara sederhana model yang dibuat memiliki algoritma seperti berikut

Algorithm 1 Algoritma *K-Means*

Require: $k\_value, max\_step, convergence\_threshold, data$

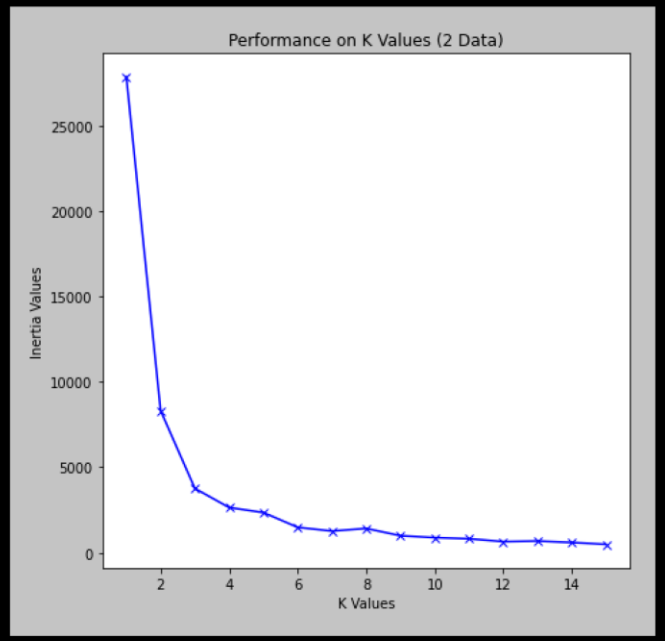
$convergence \leftarrow False$
$step \leftarrow 0$
$normalize\_data \leftarrow min\_max\_normalization(data)$
$centroid \leftarrow initialize\_centroids(data)$
while (not convergence) **and** (step < max_step) **do**
    $initial\_point \leftarrow centroid$
    $distance \leftarrow calculate\_euclidean(normalize\_data, initial\_point)$
    $cluster \leftarrow clustering(distance)$
    $new\_point \leftarrow centroid\_normalization(data, point, cluster)$
    $convergence \leftarrow convergence\_check(initial\_point, new\_point, convergence\_threshold)$
    if convergence **then**
        $point \leftarrow new\_point$
        break
    else
        $point \leftarrow new\_point$
        $step \leftarrow step + 1$
    end if
end while
$inertia \leftarrow calculate\_inertia(data, cluster, point)$
$point \leftarrow min\_max\_denormalization(point, data)$
return $cluster, point, inertia$

Pemodelan(cont.)

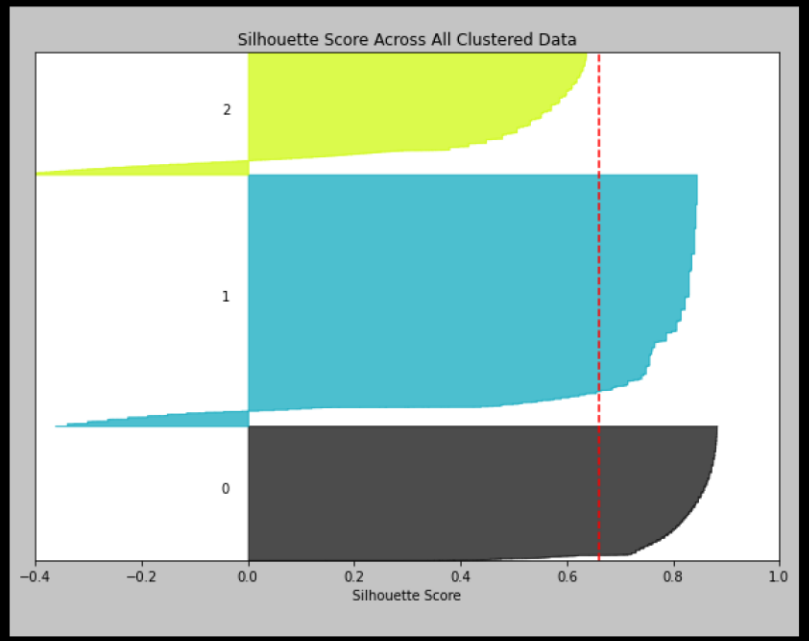
Pengimplementasian
Python dapat dilihat melalui link referensi:
https://kaenova-link.pages.dev/school/malin_tubes1
atau
https://bit.ly/KaenovaMalinTubes1

Hasil(utama)

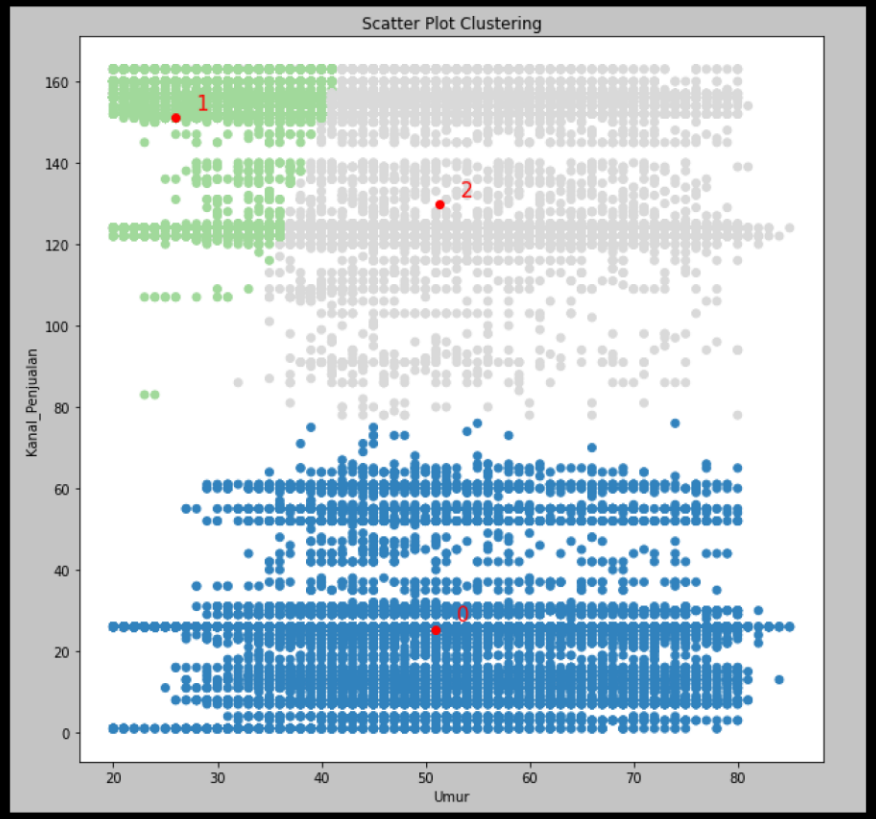
Dengan melihat korelasi pada eksplorasi dan pra-pemrosesan data, saya melakukan klusterisasi terhadap *Kanal_Penjualan* dan *Umur*



Inertia Scores on *K* runs



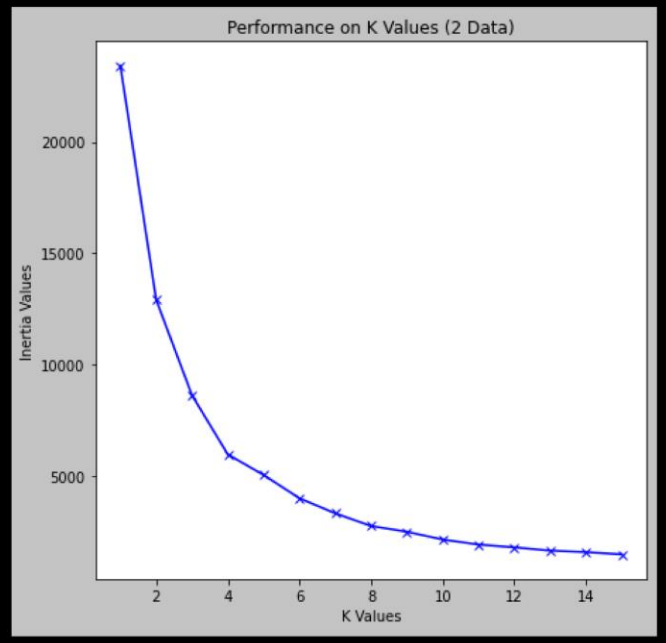
Silhouette Score on *K = 3* runs



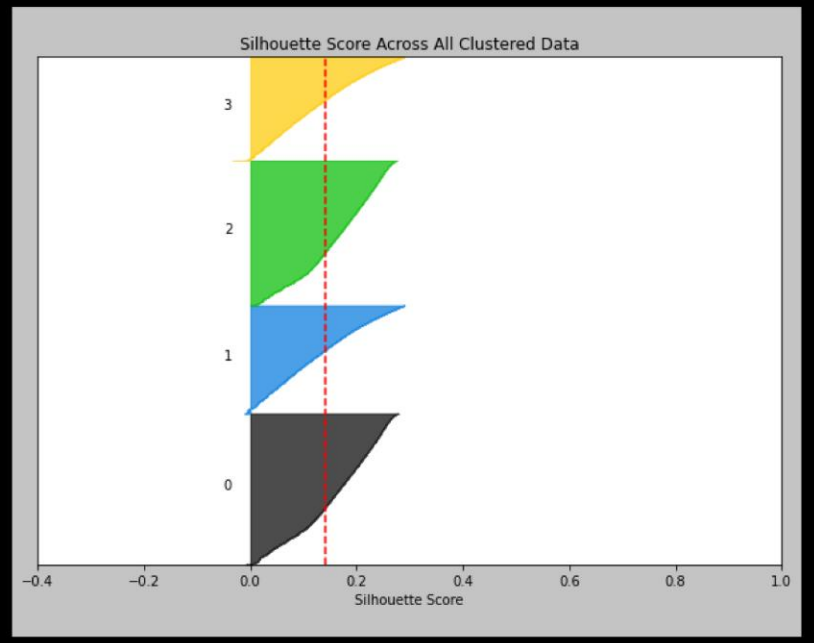
Scatter Plot Clustering on K = 3 runs

Hasil(eksperimen)

Saya mencoba melihat performa atribut lain dengan korelasi yang rendah.
Pada gambar di bawah merupakan atribut *Lama_Berlangganan* dan *Umur*



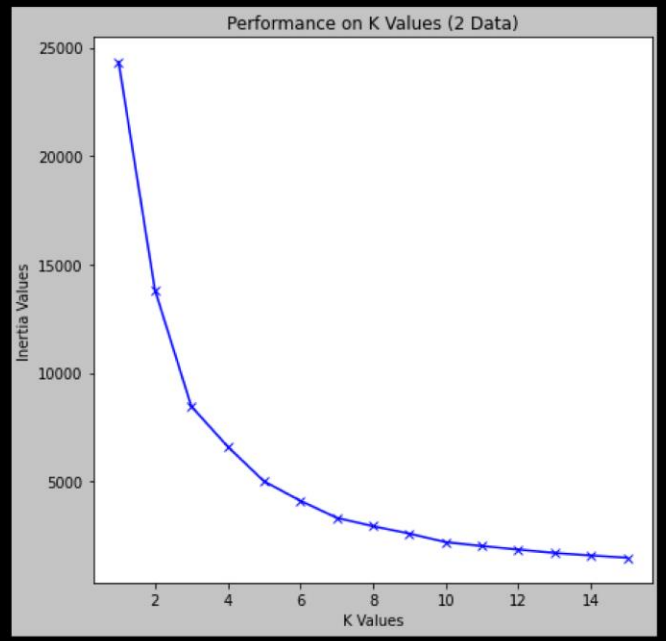
Inertia Scores on *K* runs



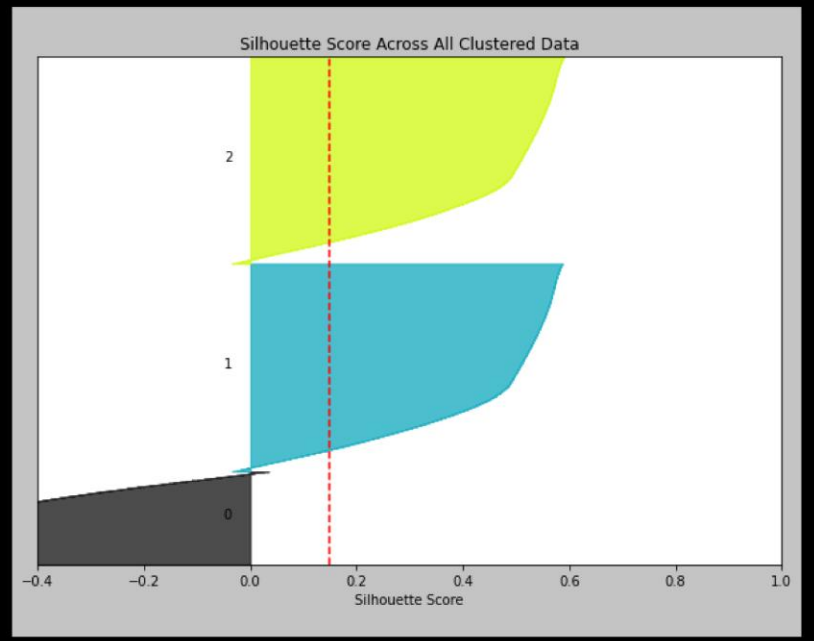
Silhouette Score on *K = 4* runs

Hasil(eksperimen)

Saya mencoba melihat performa atribut lain dengan korelasi yang rendah.
Pada gambar di bawah merupakan atribut *Lama_Berlangganan* dan Premi



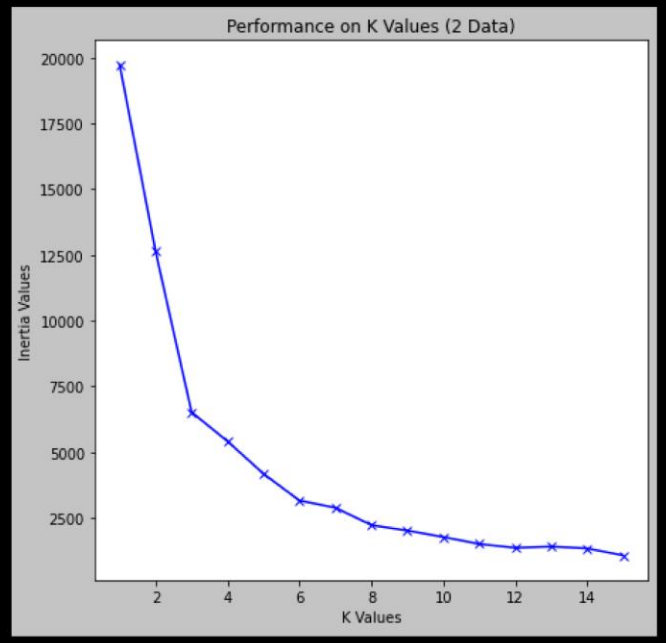
Inertia Scores on *K* runs



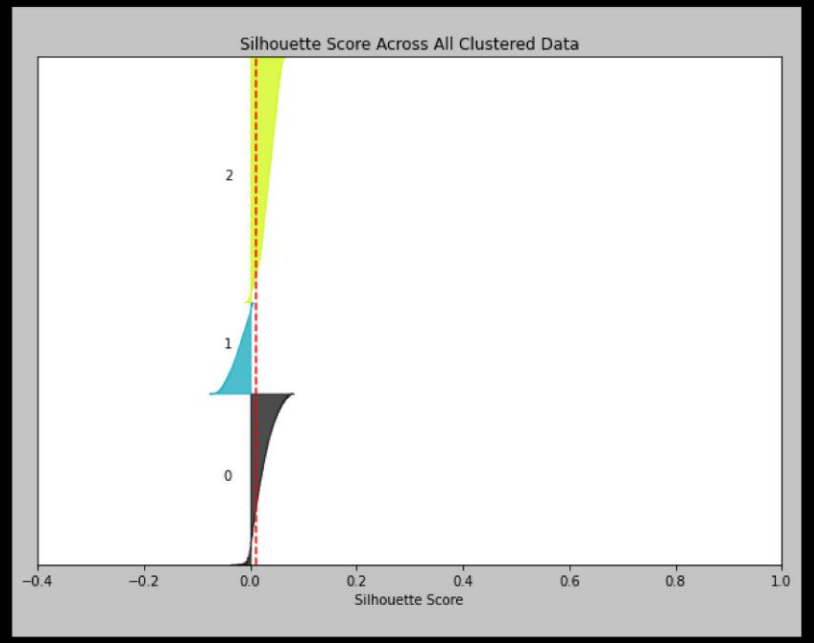
Silhouette Score on *K = 3* runs

Hasil(eksperimen)

Saya mencoba melihat performa atribut lain dengan korelasi yang rendah.
Pada gambar di bawah merupakan atribut Umur dan Premi



Inertia Scores on *K* runs

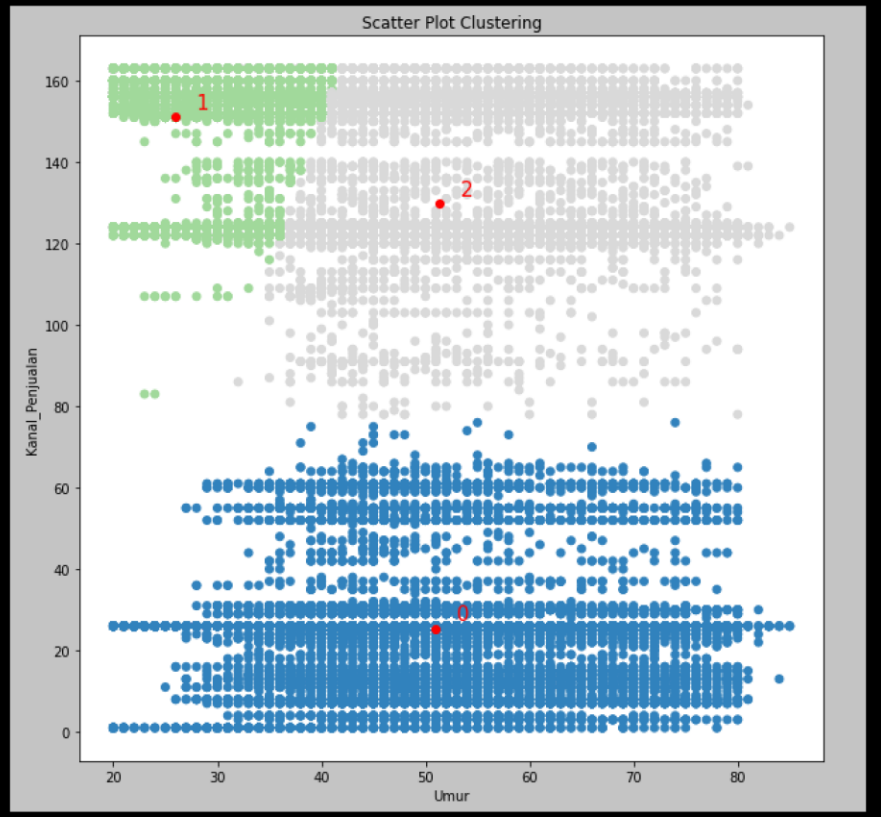


Silhouette Score on *K = 3* runs

Kesimpulan

Model ***K-Means*** dapat melakukan klusterisasi dengan baik pada dataset yang sudah disediakan. Dengan beberapa eksplorasi dan pra-pemrosesan data, model ini dapat mengklusterisasi dengan baik menggunakan atribut *Umur* dan *Kanal_Penjualan*.

Selain itu, ***K-means*** tidak baik dalam melakukan klusterisasi dengan nilai korelasi antar atribut yang rendah.



Scatter Plot Clustering on K = 3 runs with Kanal_Penjualan and Umur Attributes

Referensi

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The ad vantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

Brownlee, J. (2020). 10 Clustering Algorithms With Python.

Brus, P. (2021). Clustering: How to find hyperparameters using inertia.

Hu´n, H. X. and Huong, N. T. X. (2012). An extension of the k-means algorithm for mixed data. *Journal of Computer Science and Cybernetics*, 22(3):267–274.

Patel, V. R. and Mehta, R. G. (2011). Impact of outlier removal and normalization approach in modified k-means clustering algorithm. *International Journal of Computer Science Issues (IJCSI)*, 8(5):331.

Shahapure, K. R. and Nicholas, C. (2020). Cluster quality analysis using silhouette score. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 747–748. IEEE.

Tabak, J. (2004). Geometry : the language of space and form. Facts On File, New York.