

# **LAPORAN TUGAS BESAR**

## **MACHINE LEARNING**



**Disusun Oleh :**

**Rachma Indira**

**1301170006**

**IF - 41 -03**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**2020**

## A. Formulasi Masalah

### a. Penjelasan mengenai masalah

Untuk task **klasifikasi**, saya melakukan klasifikasi dengan *class title\_status*. Tujuannya adalah untuk mengetahui bagaimana kondisi mobil tersebut. kondisi mobil ini dibagi menjadi beberapa jenis seperti *clean*, *automatic*, *gas*, *excellent*, *cylinders*, *new*, *diesel*, dan *cab*. Sedangkan untuk **klustering**, saya melakukan pengelompokan di daerah mana saja terdapat mobil bekas yang diperkirakan akan dijual kembali atau dapat dikatakan sebagai cabang. Dataset yang digunakan untuk masing-masing task adalah :

- Klasifikasi = id, price, year, manufacturer, model, condition, odometer, title\_status(class), vin, type, lat long.
- Clustering = lat, long.

### b. Mengidentifikasi Data Sources

Dataset yang saya gunakan adalah dataset *used\_cars*, dimana dataset ini berisi 26 atribut/*feature* dan 20.000 baris data. Dataset ini saya dapatkan dari link yang sudah diberikan, sehingga tidak perlu melakukan “*collecting data*”. Feature-feature yang ada pada dataset sangat berpengaruh untuk melakukan task klasifikasi dan *clustering* yang sudah dijelaskan di poin (1).

### c. Mengidentifikasi “*Learning problems*”

Dataset ini mempunyai features/label yang sangat berhubungan dengan permasalahan yang akan diselesaikan di poin (1). Dan menurut pendapat saya, dataset ini bisa menyelesaikan masalah-masalah lainnya yang berhubungan dengan dataset(*used\_cars*). Dengan memanfaatkan penggunaan dataset dan *Machine Learning*.

### d. Potensi Bias dan Ethics

Permasalahan yang diselesaikan dengan *Machine Learning* diharapkan dapat membantu para pelaku usaha bisnis di bidang mobil bekas agar dapat meningkatkan penjualannya dengan melihat kondisi mobil seperti apakah yang dapat dijual.

## B. Eksplorasi dan persiapan data

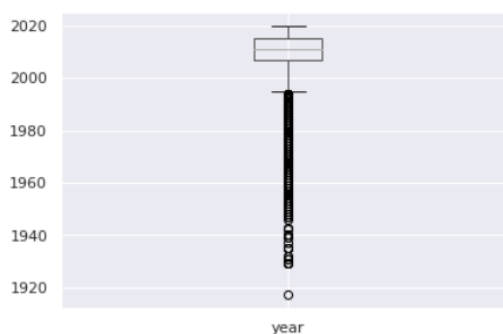
Dalam tahap ini saya melakukan data *cleansing*, eksplorasi data, *encode* data, dan feature selection. Dataset yang saya dapatkan masih harus “dibersihkan” karena masih terdapat *missing value* yang cukup banyak, seperti gambar dibawah ini :

```
print(dataset.isnull().sum())
```

Unnamed: 0	0
id	0
url	0
region	0
region_url	0
price	0
year	12
manufacturer	705
model	265
condition	9152
cylinders	7085
fuel	73
odometer	2389
title_status	110
transmission	190
vin	6645
drive	4642
size	13115
type	3659
paint_color	5514
image_url	0
description	0
county	20001
state	0
lat	1031
long	1031
dtype: int64	

Gambar jumlah *missing value* pada dataset

Sehingga perlu dilakukan “*treatment*” untuk *missing value* tersebut. Hal yang saya lakukan adalah mengganti *missing value* dengan nilai rata-rata (mean) untuk feature dengan tipe data *float* dan mengganti *missing value* dengan simbol (-) untuk feature dengan tipe data selain *object*. Karena untuk tipe data *float* masih bisa diganti nilai rata-rata dari data sebelumnya. Sedangkan untuk tipe data *object* tidak bisa diganti. Tetapi untuk feature *title\_status* karena digunakan sebagai class maka saya menghapus *missing value* pada feature ini. Dengan alasan karena feature ini digunakan sebagai class, maka akan kurang efektif jika diganti hanya dengan simbol (-). Data *cleansing* selanjutnya adalah *outlier*. Dalam dataset ini masih banyak *outlier* di tiap-tiap feature. Tetapi tidak semua *outlier* dihapus. Karena *outlier* tersebut nilainya masih masuk akal. Contohnya seperti *outlier* yang ada pada feature *year* seperti gambar dibawah ini :



**Gambar outlier pada feature year**

Jika dilihat pada gambar untuk nilai dibawah 2000 dianggap sebagai *outlier*. Tetapi pada kenyataannya banyak mobil-mobil bekas diproduksi pada tahun dibawah 2000 seperti tahun 1940. Sehingga saya tidak menghapus *outlier* yang ada pada feature *year*.

Sedangkan contoh *outlier* yang dihapus adalah pada feature *price*. Pada feature ini terdapat  $price \leq 0$ . Hal ini tidak masuk akal karena tidak mungkin menjual mobil seharga kurang dari sama dengan 0 rupiah. sehingga saya menghapus nilai  $price = 0$ .



**Gambar outlier pada feature price**

## ▼ MENGHAPUS OUTLIER DI ATRIBUT PRICE


```
df = df[df.price > 0]
```

```
[11] data.describe()
```

	id	price	year	odometer	lat	long
count	1.740800e+04	1.740800e+04	17408.000000	1.740800e+04	17408.000000	17408.000000
mean	7.043212e+09	8.798878e+04	2009.345242	1.024656e+05	40.572577	-86.904214
std	4.704549e+06	8.935022e+06	8.133930	7.763948e+04	4.509480	18.567203
min	7.032597e+09	1.000000e+00	1917.000000	0.000000e+00	-51.812200	-155.901000
25%	7.040080e+09	5.695000e+03	2006.000000	6.045600e+04	37.289500	-84.411800
50%	7.043899e+09	1.049500e+04	2011.000000	9.916435e+04	38.273500	-77.609400
75%	7.047130e+09	1.899000e+04	2015.000000	1.315405e+05	44.457800	-76.243700
max	7.050101e+09	1.172420e+09	2020.000000	2.500005e+06	59.746600	9.095700

Selanjutnya teknik eksplorasi data. Teknik eksplorasi data yang saya gunakan adalah sebagai berikut :

No	Gambar	Tujuan
1		Untuk mengetahui dimensi dari dataset

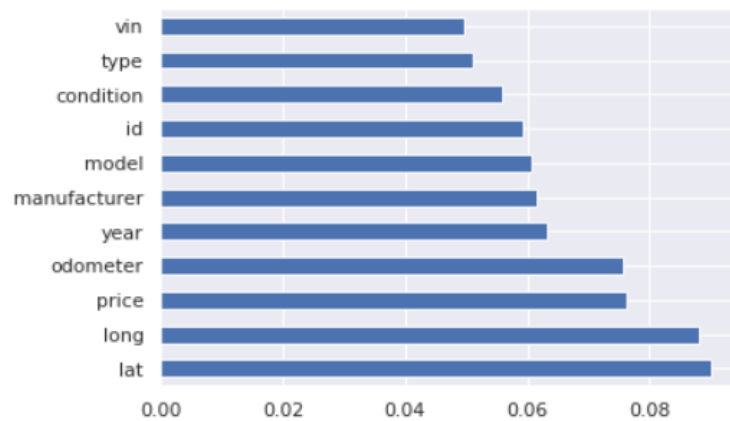
2	<div><div><div><div></div><div>df.dtypes</div></div><div><div></div><div>priceint64 odometerfloat64 dtype: object</div></div></div></div>	Untuk mengetahui tipe data dari tiap feature yang ada pada dataset																																																												
3	<div><div><div><div></div><div>df.head()</div></div><div><div></div><div><table><thead><tr><th></th><th>id</th><th>region</th><th>price</th><th>year</th><th>manufacturer</th><th>model</th><th>condition</th><th>cylinders</th><th>fuel</th></tr></thead><tbody><tr><td>0</td><td>7034441763</td><td>salt lake city</td><td>17899</td><td>2012.0</td><td>volkswagen</td><td>golf r</td><td>excellent</td><td>4 cylinders</td><td>gas</td></tr><tr><td>1</td><td>7034440610</td><td>salt lake city</td><td>0</td><td>2016.0</td><td>ford</td><td>f-150</td><td>excellent</td><td>-</td><td>gas</td></tr><tr><td>2</td><td>7034440588</td><td>salt lake city</td><td>46463</td><td>2015.0</td><td>gmc</td><td>sierra 1500</td><td>excellent</td><td>-</td><td>gas</td></tr><tr><td>3</td><td>7034440546</td><td>salt lake city</td><td>0</td><td>2016.0</td><td>ford</td><td>f-150</td><td>excellent</td><td>-</td><td>gas</td></tr><tr><td>4</td><td>7034406932</td><td>salt lake city</td><td>49999</td><td>2018.0</td><td>ford</td><td>f-450</td><td>-</td><td>-</td><td>diesel</td></tr></tbody></table></div></div></div></div>		id	region	price	year	manufacturer	model	condition	cylinders	fuel	0	7034441763	salt lake city	17899	2012.0	volkswagen	golf r	excellent	4 cylinders	gas	1	7034440610	salt lake city	0	2016.0	ford	f-150	excellent	-	gas	2	7034440588	salt lake city	46463	2015.0	gmc	sierra 1500	excellent	-	gas	3	7034440546	salt lake city	0	2016.0	ford	f-150	excellent	-	gas	4	7034406932	salt lake city	49999	2018.0	ford	f-450	-	-	diesel	Untuk melihat keseluruhan dataset
	id	region	price	year	manufacturer	model	condition	cylinders	fuel																																																					
0	7034441763	salt lake city	17899	2012.0	volkswagen	golf r	excellent	4 cylinders	gas																																																					
1	7034440610	salt lake city	0	2016.0	ford	f-150	excellent	-	gas																																																					
2	7034440588	salt lake city	46463	2015.0	gmc	sierra 1500	excellent	-	gas																																																					
3	7034440546	salt lake city	0	2016.0	ford	f-150	excellent	-	gas																																																					
4	7034406932	salt lake city	49999	2018.0	ford	f-450	-	-	diesel																																																					
4	<div><div><div><div></div><div>df.describe()</div></div><div><div></div><div><table><thead><tr><th></th><th>id</th><th>price</th><th>year</th><th>odometer</th><th>lat</th></tr></thead><tbody><tr><td>count</td><td>1.989100e+04</td><td>1.989100e+04</td><td>19891.000000</td><td>1.989100e+04</td><td>19891.000000</td></tr><tr><td>mean</td><td>7.043196e+09</td><td>7.700512e+04</td><td>2009.830625</td><td>9.911200e+04</td><td>40.401774</td></tr><tr><td>std</td><td>4.669101e+06</td><td>8.358778e+06</td><td>7.927286</td><td>7.485017e+04</td><td>4.325235</td></tr><tr><td>min</td><td>7.032597e+09</td><td>0.000000e+00</td><td>1917.000000</td><td>0.000000e+00</td><td>-51.812200</td></tr><tr><td>25%</td><td>7.040110e+09</td><td>3.912500e+03</td><td>2007.000000</td><td>5.616950e+04</td><td>37.293500</td></tr><tr><td>50%</td><td>7.043859e+09</td><td>8.795000e+03</td><td>2011.000000</td><td>9.916435e+04</td><td>38.294600</td></tr><tr><td>75%</td><td>7.047065e+09</td><td>1.750000e+04</td><td>2015.000000</td><td>1.273835e+05</td><td>44.191700</td></tr><tr><td>max</td><td>7.050101e+09</td><td>1.172420e+09</td><td>2020.000000</td><td>2.500005e+06</td><td>59.746600</td></tr></tbody></table></div></div></div></div>		id	price	year	odometer	lat	count	1.989100e+04	1.989100e+04	19891.000000	1.989100e+04	19891.000000	mean	7.043196e+09	7.700512e+04	2009.830625	9.911200e+04	40.401774	std	4.669101e+06	8.358778e+06	7.927286	7.485017e+04	4.325235	min	7.032597e+09	0.000000e+00	1917.000000	0.000000e+00	-51.812200	25%	7.040110e+09	3.912500e+03	2007.000000	5.616950e+04	37.293500	50%	7.043859e+09	8.795000e+03	2011.000000	9.916435e+04	38.294600	75%	7.047065e+09	1.750000e+04	2015.000000	1.273835e+05	44.191700	max	7.050101e+09	1.172420e+09	2020.000000	2.500005e+06	59.746600	Untuk mendapatkan informasi dari dataset. Seperti mengetahui nilai <i>mean,max,min</i> , dll						
	id	price	year	odometer	lat																																																									
count	1.989100e+04	1.989100e+04	19891.000000	1.989100e+04	19891.000000																																																									
mean	7.043196e+09	7.700512e+04	2009.830625	9.911200e+04	40.401774																																																									
std	4.669101e+06	8.358778e+06	7.927286	7.485017e+04	4.325235																																																									
min	7.032597e+09	0.000000e+00	1917.000000	0.000000e+00	-51.812200																																																									
25%	7.040110e+09	3.912500e+03	2007.000000	5.616950e+04	37.293500																																																									
50%	7.043859e+09	8.795000e+03	2011.000000	9.916435e+04	38.294600																																																									
75%	7.047065e+09	1.750000e+04	2015.000000	1.273835e+05	44.191700																																																									
max	7.050101e+09	1.172420e+09	2020.000000	2.500005e+06	59.746600																																																									
5	<div><div><div><div></div><div></div></div></div></div>	<b>Heat map</b> , untuk melihat korelasi antar atribut																																																												

Selanjutnya adalah *encode* data. *Encode* data ini hanya saya lakukan pada task klasifikasi. Hal ini untuk melanjutkan pada tahap selanjutnya, yaitu Feature Selection. *Encode* data ini adalah dengan merubah data dari string menjadi numeric seperti gambar dibawah ini :

	id	region	price	year	manufacturer	model	condition	cylinders	fuel	odometer	title_status	transmission	vin	drive	size	type	paint_color	state	lat	long
0	1197	12	1290	77	39	2191	1	4	3	3305	0	2	8644	1	1	5	1	0	1599	311
1	1195	12	0	81	13	1759	1	0	3	5	0	1	1808	1	0	0	0	0	1594	309
2	1194	12	2127	80	14	3277	1	0	3	237	0	1	5065	1	0	0	11	0	1594	309
3	1193	12	0	81	13	1759	1	0	3	5	0	1	1611	1	0	0	0	0	1594	309
4	1184	12	2158	83	13	1838	0	0	1	3699	0	1	1502	1	0	9	11	0	1590	324
5	1182	12	1043	73	33	0	0	0	3	6687	0	1	697	1	0	9	10	0	1590	324
6	1181	12	1941	82	13	1814	0	0	1	5218	0	1	1466	1	0	9	11	0	1590	324
7	1180	12	1072	71	14	3276	0	0	3	4813	0	1	2911	1	0	9	0	0	1590	324
8	1178	12	919	78	13	1793	0	0	3	9308	0	1	1432	1	0	9	11	0	1590	324
9	1177	12	1941	77	33	0	0	0	1	7130	0	1	4669	1	0	9	10	0	1590	324

Gambar hasil Encode pada dataset task klasifikasi

Tahap selanjutnya adalah feature selection. Tahap ini dilakukan pada task klasifikasi. Dengan menggunakan teknik *feature importance*. Yaitu dengan memilih 10 feature yang memiliki korelasi yang besar dengan class yang sudah ditentukan. Maka akan didapat grafik berisi 10 feature yang akan digunakan untuk tahap pemodelan klasifikasi, seperti gambar dibawah ini :



Gambar Feature selection. Diambil 11 feature karena 'id' akan dihapus

## C. Pemodelan

### 1. Klasifikasi

Pemodelan klasifikasi dengan menggunakan algoritma KNN. Dengan alasan algoritma ini cukup mudah digunakan dengan karakteristik dataset yang memiliki feature yang bisa dibilang tidak banyak. Alasan kedua juga jumlah data yang cukup sedikit dan pengklasifikasian yang cukup sederhana. Sehingga algoritma KNN bisa digunakan dan menghemat waktu pengerjaan.

### 2. Clustering

Pemodelan clustering dengan menggunakan algoritma K-Means. Dengan alasan yang hampir sama dengan task klasifikasi, yaitu jumlah data dan feature yang bisa dikatakan tidak banyak sehingga tidak memerlukan algoritma khusus.

Alasan lainnya juga karena pengelompokan/clustering hanya membutuhkan 2 features dengan hasil cluster berupa features **cabang**, sehingga tidak membutuhkan kriteria algoritma yang sulit. Dan dengan pemilihan algoritma K-Means dapat menghemat waktu pengerjaan

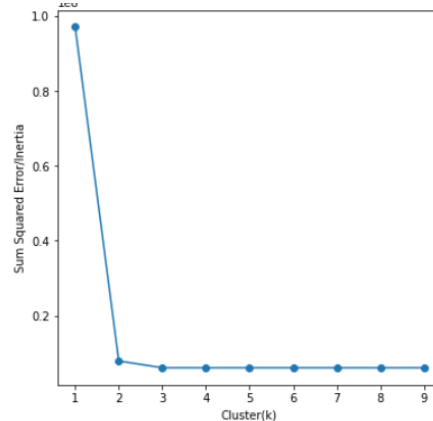
## D. Eksperimen

### 1. Klasifikasi

Eksperimen yang dilakukan pada task klasifikasi ada di tahap feature selection. Jika pada model pertama menggunakan tahap feature selection maka pada tahap kedua tidak menggunakan feature selection. Hal ini untuk melihat seberapa penting tahap feature selection pada pemodelan klasifikasi. Sehingga pada model eksperimen ini menggunakan **semua feature**.

### 2. Clustering

Eksperimen yang dilakukan pada task clustering ada di tahap pemilihan “k”. Apabila pada model pertama menggunakan  $k = 2$ , maka pada model kedua menggunakan  $k = 3$ . Untuk pemilihan ‘k’ pada model pertama didasari dari perkiraan grafik *Elbow Method*.



*gambar elbow method*

Sedangkan pemilihan ‘k’ untuk model kedua untuk melihat bagaimana cluster yang terbentuk jika menggunakan ‘k’ yang tidak optimal(dilihat dari grafik Elbow Method). Nilai ‘k’ yang diambil adalah  $k = 3$ .

## E. Hasil Evaluasi

### 1. Klasifikasi

Pada model klasifikasi saya memilih untuk menggunakan evaluasi dengan melihat nilai akurasi dan confusion matrix. Alasan saya menggunakan akurasi adalah karena saya ingin mengetahui berapa persentase benar dan salah dari hasil prediksi(model) dengan data testing(jadi membutuhkan nilai TN dan FN). Karena saya hanya membutuhkan informasi ketepatan prediksi dan data testing.

$$\text{Akurasi} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

Sehingga saya tidak membutuhkan nilai **precision**, **Recall**, dan **F1 Score** karena nilai TN(True Negative) atau FN(False negative) masih saya butuhkan untuk melihat seberapa akurat model yang telah dibuat

$$\text{Precision} = (\text{TP}) / (\text{TP} + \text{FP}) \text{ *Tidak ada TN atau FN}$$

$$\text{Recall} = (\text{TP}) / (\text{TP} + \text{FN}) \text{ * Tidak ada TN}$$

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

Sedangkan apabila saya menggunakan precision dengan bertujuan untuk mempermudah atau menggambarkan nilai TP,TF,FN,FT yang didapat dari hasil evaluasi. Berikut adalah hasil evaluasi pemodelan klasifikasi :

- **Model 1**

**Confusion Matrix :**

Predicted		Actual			
		0	1	2	5
0	3363	4			
1	12	0			
2	2	0	0		
4	93	0			
5	8	0			

```
[[3363  0  0  0  4]
 [ 12  0  0  0  0]
 [  2  0  0  0  0]
 [ 93  0  0  0  0]
 [  8  0  0  0  0]]
```

$$\text{Akurasi} : 3363 + 0 + 0 + 0 + 0 / 3482 = 0,965 \sim 0.97$$



	precision	recall	f1-score	support
0	0.97	1.00	0.98	3367
1	0.00	0.00	0.00	12
2	0.00	0.00	0.00	2
4	0.00	0.00	0.00	93
5	0.00	0.00	0.00	8
accuracy			0.97	3482
macro avg	0.19	0.20	0.20	3482
weighted avg	0.94	0.97	0.95	3482

Gambar evaluasi dari model eksperimen

- Model Eksperimen

### Confusion Matrix

Predicted \ Actual	0	1	2	4	5
0	3367	0	0	0	0
1	0	12	0	0	0
2	0	0	2	0	0
4	0	0	0	93	0
5	0	0	0	0	8

$$\text{Akurasi} = \frac{3367 + 0 + 0 + 0 + 0}{3482} = 0.96 \sim 0.97$$

### Akurasi :

	precision	recall	f1-score	support
0	0.97	1.00	0.98	3367
1	0.00	0.00	0.00	12
2	0.00	0.00	0.00	2
4	0.00	0.00	0.00	93
5	0.00	0.00	0.00	8
accuracy			0.97	3482
macro avg	0.19	0.20	0.20	3482
weighted avg	0.94	0.97	0.95	3482

Gambar evaluasi dari model eksperimen

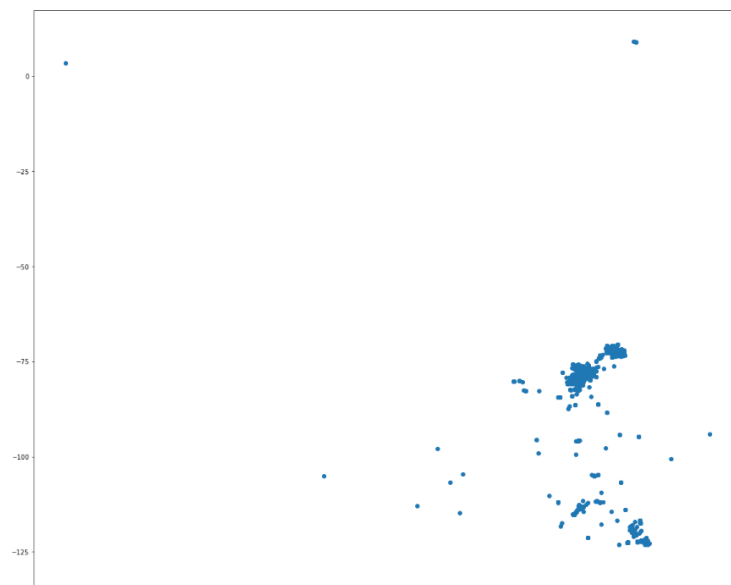
## 2. Clustering

Untuk menentukan hasil evaluasi dari pemodelan clustering saya menggunakan SSE(Sum Squared Error) dari masing-masing model. Adapun rumus dari SSE ini adalah :

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} dist^2(m_i, x)$$

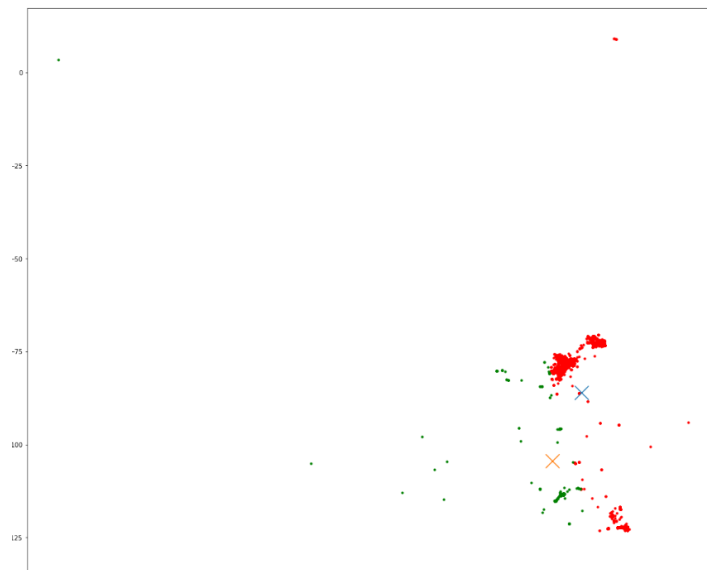
Evaluasi ini pun disertai dengan Elbow Method, untuk menemukan 'k' yang optimal dan menampilkan grafik antara jumlah cluster dan nilai SSE/inertia.

Alasan saya menggunakan SSE dalam mengevaluasi model clustering ini adalah karena saya ingin mengetahui mana model yang lebih baik performansinya jika dilihat dari nilai SSE. Karena saya juga mencoba metode *Elbow Method* yang berkaitan dengan SSE. Sehingga saya memilih menggunakan SSE agar evaluasi tersebut dapat divisualisasikan dengan *Elbow Method* yang sudah dibuat sebelumnya. Sedangkan untuk evaluasi lainnya seperti *Silhouette Coefficient* saya masih kurang paham bagaimana cara memvisualisasikannya, karena saya sendiri butuh bentuk "visual" dari evaluasi yang saya buat. Adapun hasil dari kedua model adalah :

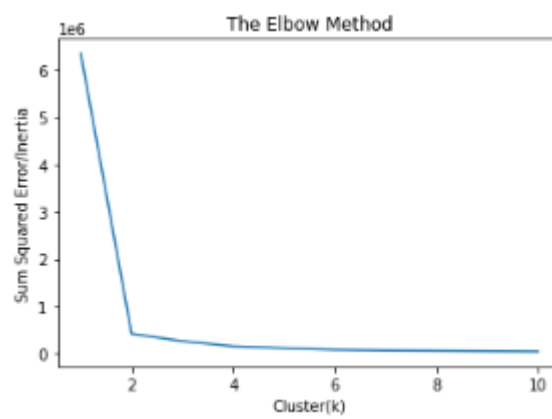


Gambar visualisasi sebelum data di cluster

- **Model 1**



**Gambar clustering dengan nilai k = 2**



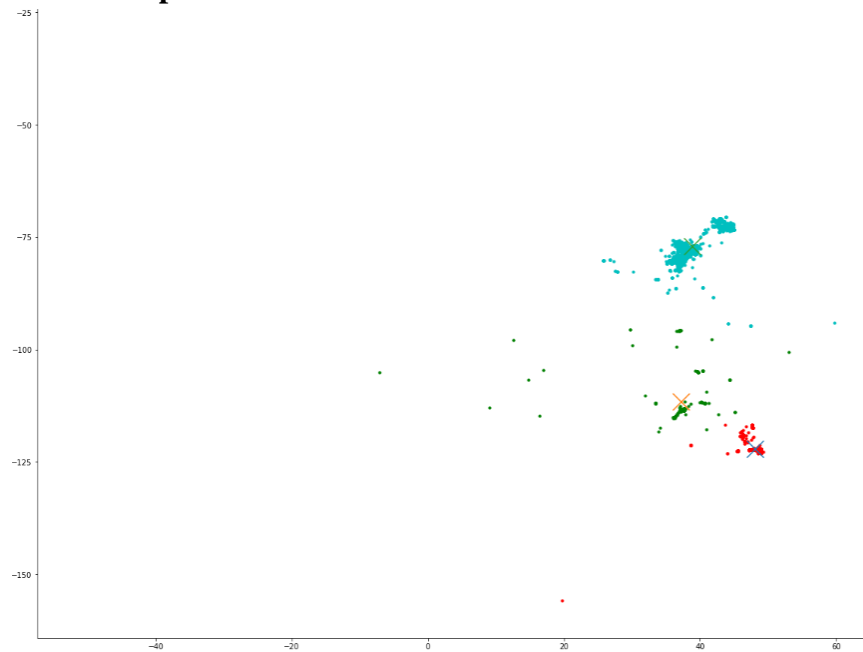
**Gambar hasil Elbow Method dengan nilai k = 2 adalah yg paling optimal**

```
#didaoat dari self.jarak pada model KM dikuadratkan
sse1=km.jarak**2
sse1
```

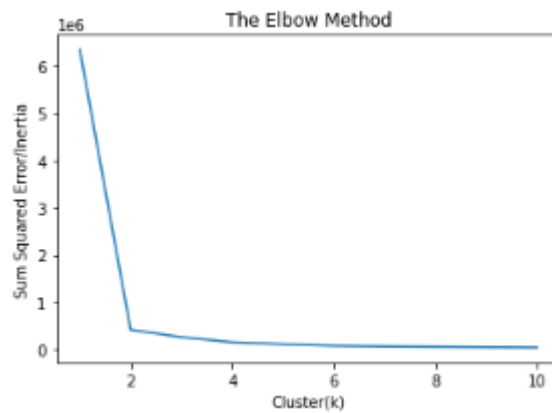
265795588455.95013

**Gambar hasil evaluasi menggunakan SSE**

- **Model Eksperimen**



**Gambar clustering dengan nilai k = 3**



**Gambar hasil Elbow Method dengan nilai k = 3**

```
sse2 = km2.jarak**2
sse2
```

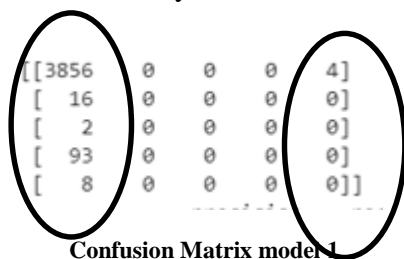
2426932734.7782755

**Gambar hasil evaluasi menggunakan SSE**

## F. Kesimpulan

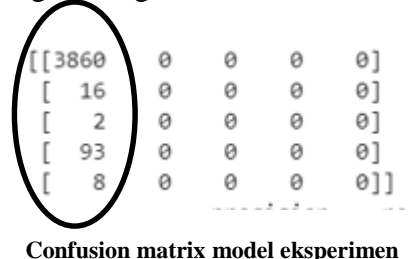
### 1. Klasifikasi

Melihat hasil evaluasi, nilai akurasi model 1 sama dengan nilai akurasi model eksperimen. Mengapa hal ini bisa terjadi walaupun model eksperimen tidak melalui tahapan feature selection, salah satu jawabannya adalah *imbalanced* yang berasal dari dataset. Apa itu *imbalanced* dataset? *Imbalanced* ini diakibatkan oleh feature yang dipilih menjadi class mempunyai data yang sangat monoton. Kita bisa melihatnya dari confusion matrix masing-masing model.



[ 3856	0	0	0	4]
[ 16	0	0	0	0]
[ 2	0	0	0	0]
[ 93	0	0	0	0]
[ 8	0	0	0	0]]

Confusion Matrix model 1



[[ 3860	0	0	0	0]
[ 16	0	0	0	0]
[ 2	0	0	0	0]
[ 93	0	0	0	0]
[ 8	0	0	0	0]]

Confusion matrix model eksperimen

Apa yang bisa disimpulkan dari gambar diatas adalah data pada class terlalu banyak memiliki data yang sama. Pada kasus ini adalah data yang pertama terlalu mendominasi, sehingga model hanya mempelajari inputan data dengan data yang pertama(yang dilingkari). Karena model terlalu banyak mempelajari jenis data di class yang sama maka akan mengakibatkan nilai akurasi tinggi tetapi jika di test dengan data yang berbeda akan mengakibatkan akurasi berkurang atau overfitting. Hal ini pula disebabkan karena pada class title\_status mempunyai 6 macam data. Dan menurut saya untuk mengklasifikasi menjadi 6 jenis dengan dataset yang berjumlah 20.000 baris masih sangat kurang. Hal lain yang dapat dilihat, pada model 1 dapat memprediksi 2 class (class 0 dan 5) sedangkan model 2 hanya 1 class(class 0 saja)

### 2. Clustering

Kesimpulan yang saya dapatkan dari model 1 dan model eksperimen adalah hasil dari SSE model 1 lebih besar dari model eksperimen. Dan Lalu dataset ini kurang bisa dieksplorasi atau clustering. Mengapa? Karena masing-masing atribut yang bertipe data float ini terpaut range yang cukup tinggi. Sehingga perserabaran yang didapatkan kurang merata. Menurut saya hal ini mengakibatkan hasil cluster menjadi kurang optimal.