Dataset : used_cars.csv

- ***Clustering***
  - **Data *Preparation* / *Preprocessing* Skenario 1 ('Year' dan 'Price')**
    1. **Menyiapkan *Dataframe***

       Pertama, *dataset* diimpor dengan menggunakan *library* Pandas (gambar 1).

19996	7049052853	https://kpr.craigslist.org/ctd/d/pasco-diesels...	kennewick-pasco-richland	https://kpr.craigslist.org	49999	2016.0	ram	cummin 2500 laramie	NaN	NaN	...	4v
19997	7049052797	https://kpr.craigslist.org/ctd/d/yakima-2018-n...	kennewick-pasco-richland	https://kpr.craigslist.org	12127	2018.0	nissan	sentra	NaN	4 cylinders	...	fv
19998	7049049369	https://kpr.craigslist.org/ctd/d/walla-walla-2...	kennewick-pasco-richland	https://kpr.craigslist.org	0	2019.0	jeep	grand cherokee limited	NaN	6 cylinders	...	4v
19999	7049042761	https://kpr.craigslist.org/ctd/d/yakima-2017-t...	kennewick-pasco-richland	https://kpr.craigslist.org	14661	2017.0	toyota	corolla	NaN	4 cylinders	...	fv
20000	7049039610	https://kpr.craigslist.org/ctd/d/pasco-2019-to...	kennewick-pasco-richland	https://kpr.craigslist.org	35999	2019.0	toyota	4runner sr5 4x4 gas suv	NaN	NaN	...	4v

20001 rows × 25 columns

Gambar 1 Dataset used_cars.csv

Lalu, dilakukan penghapusan fitur/kolom 'url', 'region_url', 'vin', 'image_url', dan 'description' menggunakan fungsi drop() dari *library* Pandas. Penghapusan dilakukan karena fitur/kolom tersebut dirasa tidak terlalu penting dalam *clustering*, sehingga mengakibatkan pengurangan fitur/kolom dalam *dataset* menjadi 20 (gambar 2).

```
# menghapus fitur/kolom 'url', 'region_url', 'vin', 'image_url', 'description' karena tidak terlalu penting da
data.drop(['url', 'region_url', 'vin', 'image_url', 'description'], axis=1, inplace=True)
```

```
# melihat dataframe setelah kolom 'url', 'region_url', 'vin', 'image_url', dan 'description' dihapus
data
```

	id	region	price	year	manufacturer	model	condition	cylinders	fuel	odometer	title_status	transmission	drive	siz
0	7034441763	salt lake city	17899	2012.0	volkswagen	golf r	excellent	4 cylinders	gas	63500.0	clean	manual	4wd	compac
1	7034440610	salt lake city	0	2016.0	ford	f-150	excellent	NaN	gas	10.0	clean	automatic	4wd	Nal
2	7034440588	salt lake city	46463	2015.0	gmc	sierra 1500	excellent	NaN	gas	7554.0	clean	automatic	4wd	Nal
3	7034440546	salt lake city	0	2016.0	ford	f-150	excellent	NaN	gas	10.0	clean	automatic	4wd	Nal
4	7034406932	salt lake city	49999	2018.0	ford	f-450	NaN	NaN	diesel	70150.0	clean	automatic	4wd	Nal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
19996	7049052853	kennewick-pasco-richland	49999	2016.0	ram	cummin 2500 laramie	NaN	NaN	diesel	43406.0	clean	automatic	4wd	Nal
19997	7049052797	kennewick-pasco-richland	12127	2018.0	nissan	sentra	NaN	4 cylinders	gas	32867.0	clean	automatic	fwd	Nal
19998	7049049369	kennewick-pasco-richland	0	2019.0	jeep	grand cherokee limited	NaN	6 cylinders	gas	17514.0	clean	automatic	4wd	Nal
19999	7049042761	kennewick-pasco-richland	14661	2017.0	toyota	corolla	NaN	4 cylinders	gas	30486.0	clean	automatic	fwd	Nal
20000	7049039610	kennewick-pasco-richland	35999	2019.0	toyota	4runner sr5 4x4 gas suv	NaN	NaN	gas	22078.0	clean	automatic	4wd	Nal

20001 rows × 20 columns

Gambar 2 Dataset setelah fitur/kolom 'url', 'region_url', 'vin', 'image_url', dan 'description' dihapus

2. Null/NaN *Handling*

Dilakukan pengecekan apakah terdapat fitur/kolom yang memiliki data Null/NaN (gambar 3).

	id	price	year	odometer	county	lat	long
count	2.000100e+04	2.000100e+04	19989.000000	1.761200e+04	0.0	18970.000000	18970.000000
mean	7.043199e+09	7.664058e+04	2009.830657	9.916435e+04	NaN	40.394737	-86.300395
std	4.668820e+06	8.335762e+06	7.913613	7.963487e+04	NaN	4.440290	18.219242
min	7.032597e+09	0.000000e+00	1917.000000	0.000000e+00	NaN	-51.812200	-155.901000
25%	7.040114e+09	3.970000e+03	2007.000000	5.013300e+04	NaN	37.273700	-80.166800
50%	7.043866e+09	8.795000e+03	2011.000000	9.389900e+04	NaN	38.258600	-77.514200
75%	7.047065e+09	1.749500e+04	2015.000000	1.339090e+05	NaN	44.439500	-76.238400
max	7.050101e+09	1.172420e+09	2020.000000	2.500005e+06	NaN	59.746600	9.095700

Gambar 3 Pengecekan Null/NaN Handling

Karena fitur/kolom ‘County’ memiliki data NaN, maka yang dilakukan selanjutnya adalah penghapusan fitur/kolom ‘County’ menggunakan fungsi drop() dari *library* Pandas (gambar 4).

```
# menghapus fitur/kolom 'county' karena memiliki jumlah baris 0
data.drop('county', axis=1, inplace=True)
```

```
# memeriksa jumlah data setiap fitur setelah fitur/kolom 'county' dihapus
data.describe()
```

	id	price	year	odometer	lat	long
count	2.000100e+04	2.000100e+04	19989.000000	1.761200e+04	18970.000000	18970.000000
mean	7.043199e+09	7.664058e+04	2009.830657	9.916435e+04	40.394737	-86.300395
std	4.668820e+06	8.335762e+06	7.913613	7.963487e+04	4.440290	18.219242
min	7.032597e+09	0.000000e+00	1917.000000	0.000000e+00	-51.812200	-155.901000
25%	7.040114e+09	3.970000e+03	2007.000000	5.013300e+04	37.273700	-80.166800
50%	7.043866e+09	8.795000e+03	2011.000000	9.389900e+04	38.258600	-77.514200
75%	7.047065e+09	1.749500e+04	2015.000000	1.339090e+05	44.439500	-76.238400
max	7.050101e+09	1.172420e+09	2020.000000	2.500005e+06	59.746600	9.095700

Gambar 4 Penghapusan fitur/kolom 'county'

Selanjutnya, dilakukan penghapusan terhadap setiap baris dalam *dataset* yang memiliki setidaknya satu *record* bernilai Null/NaN menggunakan fungsi dropna() dari *library* Pandas, sehingga baris dan fitur/kolom mengalami pengurangan menjadi 4245 baris dan 19 kolom (gambar 5).

```
# menghapus baris yang memiliki nilai null/NaN
data = data.dropna()
```

```
# melihat dataframe setelah baris yang memiliki nilai null/NaN dihapus
data
```

	id	region	price	year	manufacturer	model	condition	cylinders	fuel	odometer	title_status	transmission	drive	size
0	7034441763	salt lake city	17899	2012.0	volkswagen	golf r	excellent	4 cylinders	gas	63500.0	clean	manual	4wd	compac
24	7034278551	salt lake city	4600	2008.0	honda	civic	good	4 cylinders	gas	110982.0	clean	automatic	fwd	mid-size
48	7033720842	salt lake city	28000	2004.0	ford	f550 mechanics service	good	10 cylinders	gas	67348.0	clean	automatic	4wd	full-size
57	7033598937	salt lake city	2500	2004.0	ford	mustang	good	6 cylinders	gas	129000.0	clean	manual	rwd	full-size
109	7050078672	st george	12000	2015.0	volkswagen	jetta	like new	4 cylinders	gas	65000.0	clean	automatic	fwd	full-size
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
19919	7049199145	kennewick-pasco-richland	10995	2014.0	toyota	corolla	excellent	4 cylinders	gas	70822.0	clean	automatic	fwd	compac
19920	7049195103	kennewick-pasco-richland	3900	2006.0	chevrolet	trailblazer lt	excellent	6 cylinders	gas	184000.0	clean	automatic	4wd	full-size
19927	7049187416	kennewick-pasco-richland	11995	2017.0	subaru	impreza 2.0i sport	like new	4 cylinders	gas	35050.0	rebuilt	automatic	4wd	mid-size
19944	7049187695	kennewick-pasco-richland	18995	2019.0	jeep	cherokee latitude fwd	excellent	4 cylinders	gas	4100.0	clean	automatic	fwd	full-size
19961	7049141967	kennewick-pasco-richland	11500	2017.0	honda	civic ex	like new	4 cylinders	gas	27415.0	rebuilt	automatic	fwd	mid-size

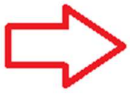
4245 rows × 19 columns

Gambar 5 Null Handling

3. *Categorical Encoding*

Selanjutnya, dilakukan pengkonversian tipe data 'object' (*categorical*) menjadi 'int32' (*numerical*) menggunakan fungsi LabelEncoder() dari *library* Sklearn (gambar 6).

```
Out[37]: id                int64
         region           object
         price             int64
         year            float64
         manufacturer     object
         model            object
         condition        object
         cylinders        object
         fuel             object
         odometer        float64
         title_status     object
         transmission     object
         drive            object
         size             object
         type             object
         paint_color      object
         state            object
         lat             float64
         long            float64
         dtype: object
```



```
Out[39]: id                int64
         region            int32
         price             int64
         year            float64
         manufacturer      int32
         model             int32
         condition         int32
         cylinders         int32
         fuel              int32
         odometer        float64
         title_status      int32
         transmission      int32
         drive             int32
         size              int32
         type              int32
         paint_color       int32
         state             int32
         lat             float64
         long            float64
         dtype: object
```

Gambar 6 Categorical Encoding

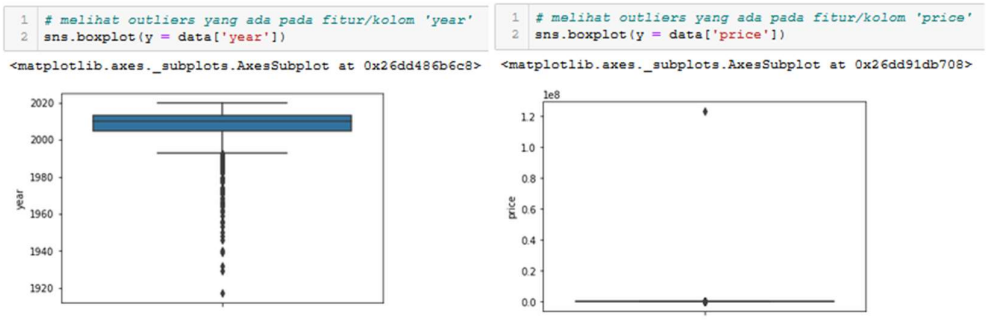
4. **Cek Korelasi**

Untuk menentukan fitur/kolom yang akan digunakan dalam *clustering* dilakukan pengecekan korelasi antar fitur/kolom dengan menggunakan fungsi corr() dari *library* Pandas, fungsi subplots() dari *library* Matplotlib, dan fungsi heatmap() dari *library* Seaborn (gambar 7).

Gambar 7 Cek korelasi dengan heatmap

5. ***Outliers Handling***

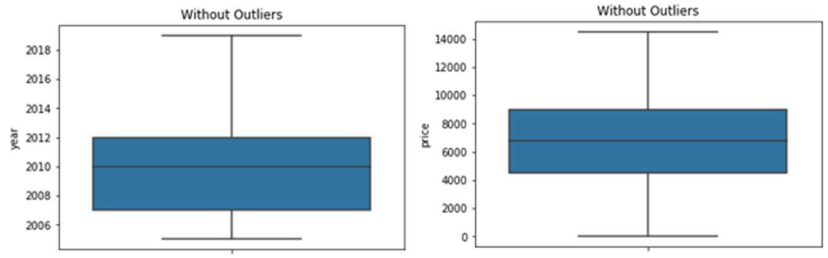
Dari hasil pengecekan korelasi, fitur/kolom 'year' dan 'price' memiliki nilai korelasi yang cukup tinggi yaitu 0,45, sehingga kedua fitur/kolom inilah yang dipilih untuk skenario 1. Selanjutnya, dilakukan pengecekan apakah terdapat *outliers* pada kedua fitur/kolom ini dengan menggunakan fungsi boxplot() dari *library* Seaborn (gambar 8).



Gambar 8 Pengecekan outliers

Karena pada fitur/kolom ‘year’ dan ‘price’ terdapat *outliers*, maka yang dilakukan selanjutnya adalah menghilangkan *outliers* tersebut dengan cara membuat variabel baru yang menampung kedua fitur/kolom dengan batasan nilai tertentu (gambar 9). Penghilangan *outliers* ini juga mengakibatkan berkurangnya baris *dataset* dari 4245 menjadi 2389 baris (gambar 10).

```
# menghapus outliers yang ada pada fitur/kolom 'year' dan 'price'
data_new = data[(data.year>=2005) # >=2005, karena outliers berada di <2005
        &
(data.price<=1.450000e+04)] # <=1.450000e+04, karena outliers berada di >1.450000e+04
```



Gambar 9 Outliers Handling

6. *Scaling*

Selanjutnya, dilakukan *scaling* kepada fitur/kolom ‘year’ dan ‘price’ agar nilai-nilai yang sudah dikonversi sebelumnya menjadi nilai-nilai dengan rentang dari 0 – 1. Fungsi yang digunakan adalah fungsi minmax_scaling() dari *library* Mlxtend (gambar 10).

	year	price
24	0.214286	0.317241
109	0.714286	0.827586
112	0.428571	0.275862
139	0.428571	0.758276
164	0.071429	0.551655
...	...	...
19908	0.500000	0.862069
19919	0.642857	0.758276
19920	0.071429	0.268966
19927	0.857143	0.827241
19961	0.857143	0.793103

2389 rows × 2 columns

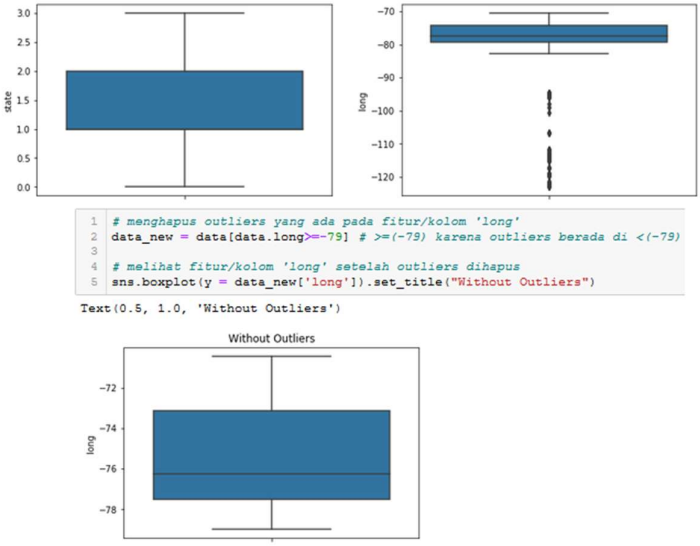
Gambar 10 Scaling

- **Data *Preparation* / *Preprocessing* Skenario 2 ('State' dan 'Long')**

  Data *preparation* yang dilakukan pada skenario 2 adalah sama dengan data *preparation* yang dilakukan pada skenario 1, yang berbeda hanyalah pada pemilihan fitur/kolom (dalam skenario 2 yang dipilih adalah fitur/kolom 'state' dan 'long'), sehingga akan menghasilkan perbedaan pada proses *outliers handling* dan *scaling*. Berikut adalah perbedaan dari skenario 2.

  1. ***Outliers Handling***

     Dalam skenario 2, yang memiliki *outliers* hanya fitur/kolom 'long', sehingga fitur/kolom 'state' tidak dilakukan *outliers handling* (gambar 11). Penghilangan *outliers* ini juga mengakibatkan berkurangnya baris *dataset* dari 4245 menjadi 2990 baris (gambar 12).



Gambar 11 Outliers Handling skenario 2

  2. ***Scaling***

     Untuk *scaling* kepada fitur/kolom pada skenario 2 menghasilkan rentang nilai sebagai berikut (gambar 12).

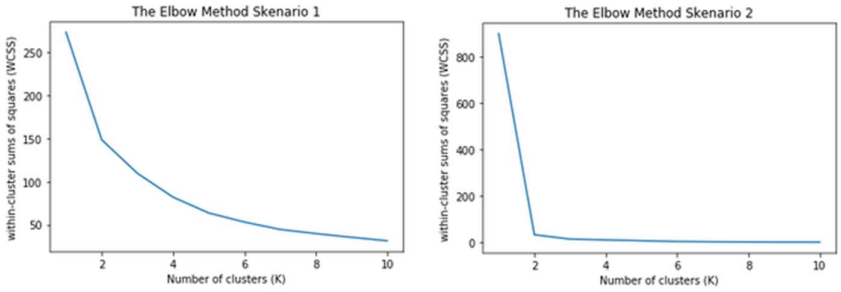
	state	long
665	1.0	0.672674
667	1.0	0.667908
681	1.0	0.792407
685	1.0	0.761154
686	1.0	0.761154
...	...	...
16557	0.0	0.095330
16560	0.0	0.095330
16566	0.0	0.099733
16569	0.0	0.095330
16570	0.0	0.095330

2990 rows × 2 columns

Gambar 12 Scaling skenario 2

- ***Clustering* Skenario 1 dan Skenario 2 dengan K-Means**

  Clustering dilakukan menggunakan metode K-Means, karena metode ini dirasa cukup mudah untuk digunakan dan dapat menangani *dataset* yang cukup besar. Dalam metode ini, nilai *K* sangatlah berpengaruh, sehingga sebelum masuk ke *clustering*, terlebih dahulu dilakukan pengecekan untuk nilai *K* yang mendekati optimal dengan metode Elbow atau e*lbow method* (gambar 13).



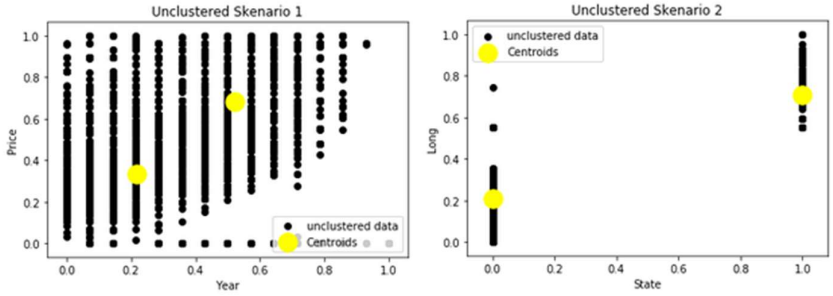
Gambar 13 Elbow Method

1. ***Elbow Method***

   Dari hasil pengecekan, didapatkan bahwa pengurangan variansi yang signifikan terjadi pada saat *K*=2, sehingga nilai *K* adalah 2. Nilai *K* ini menunjukkan seberapa banyak *cluster* yang ingin dibuat. Nilai *K* ini juga digunakan dalam perulangan untuk menentukan posisi *centroid*. *Centroid* itu sendiri diinisiasi secara acak dari rentang angka sebanyak *data points* (baris) dari masing-masing fitur/kolom.

2. ***Unclustered* Data**

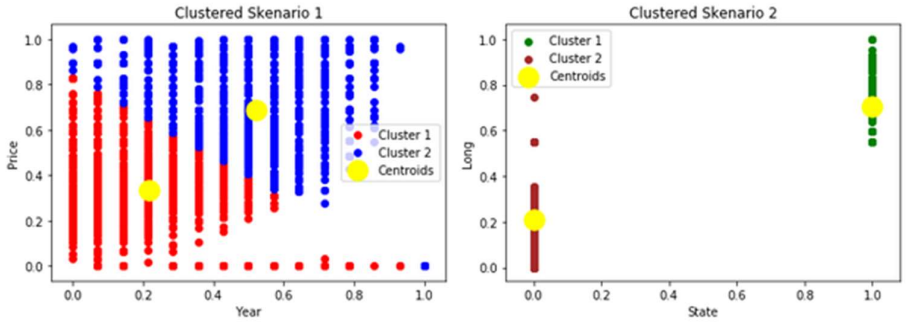
   Setelah menghitung *euclidian distance* ke *centroid*, didapatkan hasil data yang belum *ter-cluster* (belum diberi warna) tetapi sudah terdapat *centroid* di *data points* yang didapat secara acak (gambar 14).



Gambar 14 Unclustered data

3. ***Clustered* Data**

Langkah terakhir dalam *clustering* ini adalah pemberian warna pada visualisasi data sebelumnya, yaitu *unclustered data*. Dari hasil pewarnaan, terlihat bahwa data sudah berhasil *ter-cluster* menjadi *K* (dua) *cluster* (gambar 15).



Gambar 15 Clustered data

- ***Classification***
  - **Data *Preparation* / *Preprocessing* ('Year' dan 'Price')**

    Data *preparation* yang dilakukan pada *classification* adalah sama dengan data *preparation* yang dilakukan pada *clustering* skenario 1 maupun skenario 2, yang berbeda hanyalah pada *classification* tidak dilakukan *outliers handling*, sehingga tidak terjadi pengurangan baris pada fitur/kolom 'year' dan 'price', yaitu tetap 4245 baris (gambar 16). Dari hasil pengecekan korelasi, fitur/kolom 'year' dan 'price' ditentukan sebagai *x*, dan fitur/kolom 'condition' ditentukan sebagai kelas klasifikasi *y*. Selain itu, dalam *classification* terdapat satu proses tambahan, yaitu data *splitting*.

0	0.922330	0.000145
24	0.883495	0.000037
48	0.844660	0.000227
57	0.844660	0.000020
109	0.951456	0.000097
...	...	...
19919	0.941748	0.000089
19920	0.864078	0.000032
19927	0.970874	0.000097
19944	0.990291	0.000154
19961	0.970874	0.000093

4245 rows × 2 columns

Gambar 16 Classification Scaling

1. **Data *Splitting***

   Data *splitting* merupakan proses pemotongan data dari fitur/kolom yang dipilih menjadi data *train* dan data *test*. Pada *classification* kali ini, pemotongan data dibagi dengan persentase 80% untuk data *train*, dan 20% untuk data *test*. Data *splitting* dilakukan dengan menggunakan fungsi train_test_split() dari *library* Sklearn. Gambar 17 menunjukkan hasil pemotongan yaitu *x_train* dan *y_train* menjadi 3396 baris, *x_test* dan *y_test* menjadi 849 baris.

```
# melihat data train x
x_train
```

	year	price
13752	0.912621	0.000073
13537	0.912621	0.000152
8165	0.922330	0.000064
13499	0.864078	0.000058
10951	0.864078	0.000026
...	...	...
14915	0.883495	0.000040
1850	0.844660	0.000065
13249	0.689320	0.000012
16023	0.932039	0.000239
2788	0.864078	0.000065

3396 rows × 2 columns

```
# melihat data test x
x_test
```

	year	price
14771	0.912621	0.000036
3475	0.854369	0.000045
7458	0.912621	0.000146
10040	0.776699	0.000028
794	0.834951	0.000023
...	...	...
12889	0.951456	0.000138
9690	0.834951	0.000045
16039	0.912621	0.000047
14929	0.922330	0.000073
9655	0.941748	0.000211

849 rows × 2 columns

```
# melihat data train y
y_train
```

```
13752    3
13537    2
8165     0
13499    3
10951    0
        ..
14915    0
1850     0
13249    2
16023    2
2788     2
Name: condition, Length: 3396, dtype: int32
```

```
# melihat data test y
y_test
```

```
14771    0
3475     0
7458     0
10040    2
794      1
        ..
12889    2
9690     0
16039    0
14929    0
9655     0
Name: condition, Length: 849, dtype: int32
```

Gambar 17 Data Splitting

- ***Classification* dengan K-Nearest Neighbors**

  Classification pertama dilakukan dengan menggunakan metode K-Nearest Neighbors (KNN). *Classification* dilakukan dengan menggunakan fungsi KNeighborsClassifier() dari *library* Sklearn. Dalam metode ini, nilai *K* juga diperlukan untuk menentukan *neighbors*, sehingga digunakan nilai *K* yang sama seperti saat *clustering*, yaitu *K=2*. Berikut hasil laporan *classification* menggunakan fungsi classification_report() dari *library* Sklearn (gambar 18).

```
# melihat laporan hasil dari classification
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.60      0.81      0.69       440
           1       0.36      0.45      0.40        29
           2       0.58      0.36      0.45       321
           3       0.28      0.09      0.14        53
           4       0.00      0.00      0.00         3
           5       0.00      0.00      0.00         3

    accuracy                           0.58       849
   macro avg       0.30      0.29      0.28       849
weighted avg       0.56      0.58      0.55       849
```

Gambar 18 K-Nearest Neighbors classification

- ***Classification* dengan Gaussian Naïve Bayes**

  Classification kedua dilakukan dengan menggunakan metode Gaussian Naïve Bayes (GNB). *Classification* dilakukan dengan menggunakan fungsi GaussianNB() dari *library* Sklearn. Berikut hasil laporan *classification* menggunakan fungsi classification_report() dari *library* Sklearn (gambar 19).

```
# melihat laporan hasil dari classification
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.33      0.00      0.00       440
           1       0.40      0.21      0.27        29
           2       0.40      0.95      0.56       321
           3       0.16      0.21      0.18        53
           4       0.00      0.00      0.00         3
           5       0.00      0.00      0.00         3

    accuracy                           0.38       849
   macro avg       0.22      0.23      0.17       849
weighted avg       0.35      0.38      0.24       849
```

Gambar 19 Gaussian Naive Bayes classification

- Kesimpulan
  1. Null/NaN *Handling* dan *Outliers Handling* dapat menyebabkan penurunan drastis terhadap jumlah baris yang terdapat dalam *dataset*, sehingga memungkinkan penurunan permormansi *clustering* dan *classification*.
  2. *Outliers Handling* pada *clustering* skenario 2 menyebabkan penurunan jumlah baris lebih sedikit dari skenario 1, karena dalam skenario 2 hanya terdapat satu fitur/kolom yang mempunyai *outliers*, yaitu 'long'.
  3. Data *preparation*/*preprocessing* dan pemilihan fitur sangat berpengaruh terhadap model yang dihasilkan oleh proses *clustering*.
  4. Untuk *classification*, metode K-Nearest Neighbors mendapatkan akurasi yang lebih tinggi (0.58) dibandingkan dengan menggunakan metode Gaussian Naive Bayes (0.38).

Referensi

https://scikit-learn.org/
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6
https://www.digitalocean.com/community/tutorials/how-to-build-a-machine-learning-classifier-in-python-with-scikit-learn
https://www.dataquest.io/blog/sci-kit-learn-tutorial/
https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15
https://medium.com/@george.drakos62/handling-missing-values-in-machine-learning-part-1-dda69d4f88ca
https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html
https://scikit-learn.org/stable/modules/feature_selection.html
https://github.com/puppeteer/puppeteer
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html#pandas.DataFrame.fillna
https://mubaris.com/posts/kmeans-clustering/
https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
https://medium.com/datadriveninvestor/k-fold-cross-validation-6b8518070833
https://youtu.be/tyhJa4OnLuc
https://www.pluralsight.com/guides/deep-learning-model-perform-binary-classification
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html
https://medium.com/hacktive-devs/feature-engineering-in-machine-learning-part-1-a3904769cd93
https://drive.google.com/drive/folders/17_M8X4gvrIoMwQHucANu8GsWq-dv1a_g
https://keras.io/getting-started/sequential-model-guide/
https://medium.com/@dmitriy.kavyazin/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensional-dataset-577b2a7a5fe2
https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
https://docs.google.com/document/d/1BiR6N6YzZ1I3ZVhUnPt0QNWnQan_mWuPClWhLY-OJb8/edit
https://medium.com/@kurniasp/naive-bayes-classifier-using-scikit-learn-in-python-3067144af115
https://scikit-learn.org/stable/modules/naive_bayes.html
https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn
https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/
https://stackoverflow.com/questions/31421413/how-to-compute-precision-recall-accuracy-and-f1-score-for-the-multiclass-case
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html
https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

https://www.researchgate.net/post/How_can_fix_the_Error_Value_in_python_Unknown_label_type_continuous
https://github.com/jpmml/sklearn2pmml/issues/103
https://www.kaggle.com/pratsiuk/valueerror-unknown-label-type-continuous
https://stackoverflow.com/questions/41925157/logisticregression-unknown-label-type-continuous-using-sklearn-in-python
https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://stats.stackexchange.com/questions/319514/why-feature-scaling-only-to-training-set
https://datascience.stackexchange.com/questions/39932/feature-scaling-both-training-and-test-data
https://datascience.stackexchange.com/questions/54908/data-normalization-before-or-after-train-test-split
https://community.rapidminer.com/discussion/32592/normalising-data-before-data-split-or-after
https://stackoverflow.com/questions/49444262/normalize-data-before-or-after-split-of-training-and-testing-data
https://www.quora.com/Should-scaling-be-done-on-both-training-data-and-test-data-for-machine-learning-Can-one-do-scaling-on-only-the-training-data
https://hackernoon.com/what-steps-should-one-take-while-doing-data-preprocessing-502c993e1caa
https://stackoverflow.com/questions/30667525/importerror-no-module-named-sklearn-cross-validation
https://medium.com/@denzilsequeira/data-pre-processing-for-deep-learning-for-classification-or-regression-2bddb0b9183b
https://stackoverflow.com/questions/43784903/scikit-k-means-clustering-performance-measure
https://www.tutorialspoint.com/scikit_learn/scikit_learn_clustering_performance_evaluation.htm
https://towardsdatascience.com/clustering-evaluation-strategies-98a4006fcfc
https://scikit-learn.org/stable/modules/clustering.html
http://rasbt.github.io/mlxtend/user_guide/preprocessing/minmax_scaling/
https://machinelearningmastery.com/scale-machine-learning-data-scratch-python/
https://python-data-science.readthedocs.io/en/latest/normalisation.html
https://www.kaggle.com/rtatman/data-cleaning-challenge-scale-and-normalize-data
https://medium.com/machine-learning-id/melakukan-feature-scaling-pada-dataset-229531bb08de
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02
https://www.geeksforgeeks.org/python-how-and-where-to-apply-feature-scaling/amp/
https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
https://medium.com/datadriveninvestor/finding-outliers-in-dataset-using-python-efc3fce6ce32
https://www.mikulskibartosz.name/how-to-remove-outliers-from-seaborn-boxplot-charts/
https://seaborn.pydata.org/generated/seaborn.boxplot.html
https://stackoverflow.com/questions/53735603/extract-outliers-from-seaborn-boxplot
https://statinfer.com/104-3-5-box-plots-and-outlier-dectection-using-python/
https://medium.com/analytics-vidhya/outlier-treatment-9bbe87384d02

https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623
https://seaborn.pydata.org/generated/seaborn.heatmap.html
https://medium.com/labtek-indie/exploratory-data-analysis-7b9b0234ba05
https://www.dataquest.io/blog/settingwithcopywarning/
https://stackoverflow.com/questions/49712002/pandas-dropna-function-not-working
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.dropna.html
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html
https://www.w3resource.com/pandas/dataframe/dataframe-dropna.php
https://stackoverflow.com/questions/58868256/scikit-learn-label-encoder-resulting-in-error-argument-must-be-a-string-or-numb
https://scikit-learn.org/stable/modules/preprocessing_targets.html#label-encoding
https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd
https://stackoverflow.com/questions/24458645/label-encoding-across-multiple-columns-in-scikit-learn/30267328#30267328
https://www.kaggle.com/learn-forum/61148
https://thispointer.com/how-to-get-check-data-types-of-dataframe-columns-in-python-pandas/
https://www.duniailkom.com/tutorial-belajar-python-tipe-data-set-dalam-bahasa-python/
https://www.malasngoding.com/operasi-tipe-data-set/
https://www.malasngoding.com/tipe-data-bahasa-python/
https://medium.com/@16611092/mengenal-pandas-dalam-python-cc66d0c5ea40
https://stackoverflow.com/questions/24458645/label-encoding-across-multiple-columns-in-scikit-learn
https://www.datacamp.com/community/tutorials/categorical-data
https://realpython.com/convert-python-string-to-int/
https://stackoverflow.com/questions/55407713/how-to-encode-a-text-string-into-a-number-in-python
https://stackoverflow.com/questions/53420705/python-reversibly-encode-alphanumeric-string-to-integer
https://www.it-swarm.dev/id/python/menghapus-banyak-kolom-berdasarkan-nama-kolom-di-panda/1051210692/amp/
https://www.it-swarm.dev/id/python/hapus-kolom-dari-panda-dataframe/1070622164/amp/