

**LAPORAN**  
**Case-Based 02 Dataset Water Treatment Plant Dengan Algoritma Kmeans Clustering**  
**2021/2022**



**Disusun oleh :**  
Muhammad Fadil Maulana Akbar (1301204297)  
**(IF 44 04)**

**S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**

## Daftar isi

<b>BAB I</b>	<b>3</b>
1.1 Dataset Water Treatment Plant	3
1.2 Strategi Pra-pemrosesan Data	3
1.3 Diagram/Plot	3
<b>BAB II</b>	<b>4</b>
2.1 Tools Pra-Pemrosesan Data	4
2.2 Pra-Pemrosesan Data	4
<b>BAB III</b>	<b>12</b>
3.1 Tools Training Data	12
3.2 K-Means Clustering	12
3.2 Implementasi	12
<b>BAB IV</b>	<b>15</b>
4.1 Evaluasi dengan Elbow Method	15
4.2 Kesimpulan	16
<b>BAB V</b>	<b>17</b>
5.1 Daftar Pustaka	17
5.2 Link	17

## BAB I

## 1.1 Dataset Water Treatment Plant

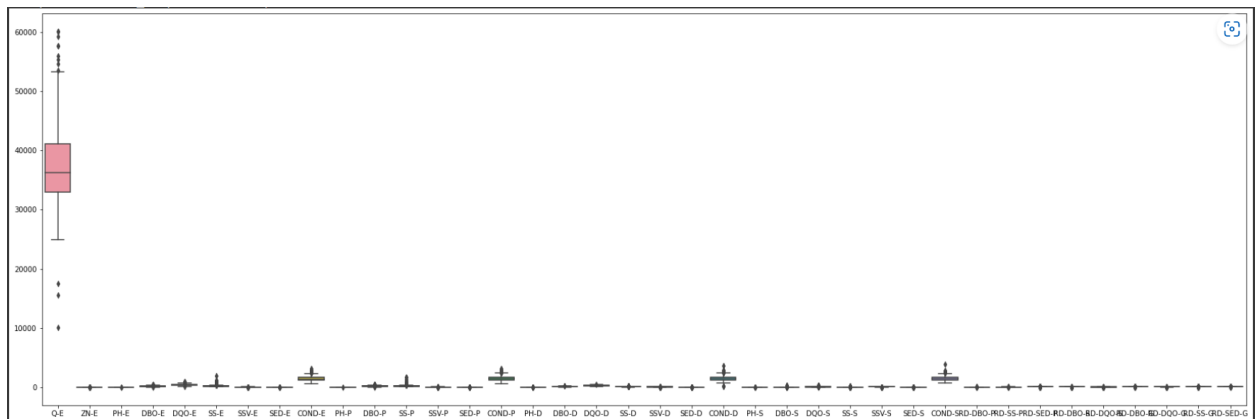
Dataset yang digunakan adalah dataset water treatment plant yang memiliki beberapa fitur untuk dianalisis. Dataset ini akan dianalisis dengan algoritma unsupervised learning k-means clustering untuk mengelompokkan fitur yang memiliki nilai yang mirip.

## 1.2 Strategi Pra-pemrosesan Data

- Mengubah data categorical menjadi numeric value
- Mencari value dari atribut yang hilang dan mengisi nilai tersebut dengan rata-rata dari atribut tersebut.
- Mencari outliers dan menghapus outliers dengan menghitung z score.
- Mencari korelasi dari tiap atribut, atribut yang memiliki korelasi tinggi akan dipilih untuk model training.
- Mengimplementasikan PCA (Principal Component Analysis) untuk mereduksi dimensi.

### 1.3 Diagram/Plot

- BoxPlot



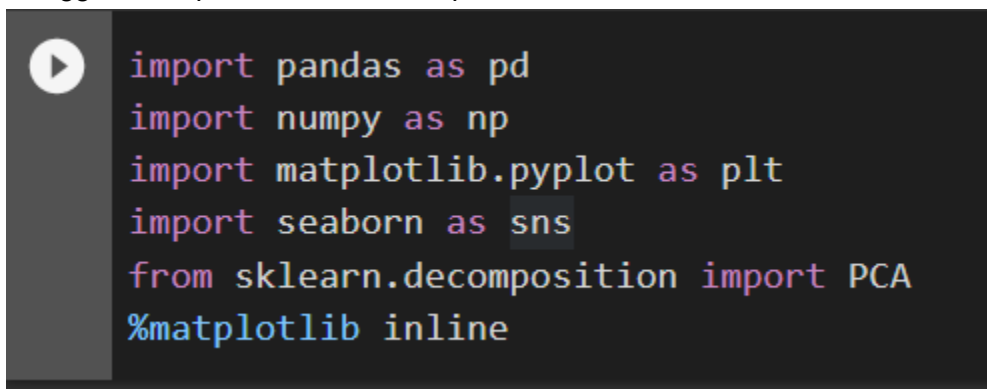
## BAB II

### 2.1 Tools Pra-Pemrosesan Data

- Google Colab
- pandas
- numpy
- matplotlib.pyplot
- Seaborn

### 2.2 Pra-Pemrosesan Data

1. Langkah pertama adalah import semua library dan read water-treatment-plant.csv menggunakan pandas lalu menampilkan informasi dari dataset.



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
%matplotlib inline
```



```
!wget https://raw.githubusercontent.com/fadilmr/CaseBased2/main/water-treatment.csv
--2022-12-02 00:01:03-- https://raw.githubusercontent.com/fadilmr/CaseBased2/main/water-treatment.csv
```

Index	Days	Q-E	ZN-E	PH-E	DBO-E	DQO-E	SS-E	SSV-E	SED-E	COND-E	PH-P	DBO-P	SS-P	SSV-P	SED-P	COND-P	PH-D	DBO-D	DQO-D	SS-D
0	D-1/3/90	44101	1.5	7.8	?	407	166	66.3	4.5	2110	7.9	?	228	70.2	5.5	2120	7.9	?	280	94
1	D-2/3/90	39024	3	7.7	?	443	214	69.2	6.5	2660	7.7	?	244	75.4	7.7	2570	7.6	?	474	96
2	D-4/3/90	32229	5	7.6	?	528	186	69.9	3.4	1666	7.7	?	220	72.7	4.5	1594	7.7	?	272	92
3	D-5/3/90	35023	3.5	7.9	205	588	192	65.6	4.5	2430	7.8	236	268	73.1	8.5	2280	7.8	158	376	96
4	D-6/3/90	36924	1.5	8.0	242	496	176	64.8	4	2110	7.9	?	236	57.6	4.5	2020	7.8	?	372	88

1	Q-E	527	non-null	object
2	ZN-E	527	non-null	object
3	PH-E	527	non-null	float64
4	DBO-E	527	non-null	object
5	DQO-E	527	non-null	object
6	SS-E	527	non-null	object
7	SSV-E	527	non-null	object
8	SED-E	527	non-null	object
9	COND-E	527	non-null	int64
10	PH-P	527	non-null	float64
11	DBO-P	527	non-null	object
12	SS-P	527	non-null	int64
13	SSV-P	527	non-null	object
14	SED-P	527	non-null	object
15	COND-P	527	non-null	int64
16	PH-D	527	non-null	float64
17	DBO-D	527	non-null	object
18	DQO-D	527	non-null	object
19	SS-D	527	non-null	object
20	SSV-D	527	non-null	object
21	SED-D	527	non-null	object
22	COND-D	527	non-null	int64
23	PH-S	527	non-null	object
24	DBO-S	527	non-null	object
25	DQO-S	527	non-null	object
26	SS-S	527	non-null	object
27	SSV-S	527	non-null	object
28	SED-S	527	non-null	object
29	COND-S	527	non-null	object
30	RD-DBO-P	527	non-null	object
31	RD-SS-P	527	non-null	object



- Langkah selanjutnya adalah mencari value “?” dan mengganti value tersebut dengan NaN

```
df.columns[df.isin(["?"]).any()]
df = df.replace('?', np.nan)
```

```
df.head()
```

	Days	Q-E	ZN-E	PH-E	DBO-E	DQO-E	SS-E	SSV-E	SED-E	COND-E	...
0	D-1/3/90	44101	1.5	7.8	NaN	407	166	66.3	4.5	2110	...
1	D-2/3/90	39024	3	7.7	NaN	443	214	69.2	6.5	2660	...
2	D-4/3/90	32229	5	7.6	NaN	528	186	69.9	3.4	1666	...
3	D-5/3/90	35023	3.5	7.9	205	588	192	65.6	4.5	2430	...
4	D-6/3/90	36924	1.5	8.0	242	496	176	64.8	4	2110	...

3. mencari nilai atribut yang kosong dan mengisi nilai kosong tersebut dengan rata-rata dari masing-masing atribut

	<code>df.isna().sum()</code>	
	Days	0
	Q-E	18
	ZN-E	3
	PH-E	0
	DBO-E	23
	DQO-E	6
	SS-E	1
	SSV-E	11
	SED-E	25
	COND-E	0
	PH-P	0
	DBO-P	40
	SS-P	0
	SSV-P	11
	SED-P	24
	COND-P	0
	PH-D	0
	DBO-D	28
	DQO-D	9
	SS-D	2
	SSV-D	13
	SED-D	25
	COND-D	0
	PH-S	1
	DBO-S	23
	DQO-S	18
	SS-S	5
	SSV-S	17
	SED-S	28

Karena atribut days adalah objek, maka akan diubah menjadi index dari dataframe

```
df_temp = df.set_index('Days', inplace=False)
df_temp = df_temp.astype(float)
```

Setelah days diubah menjadi index, maka missing value dapat diisi dengan rata-ratanya.

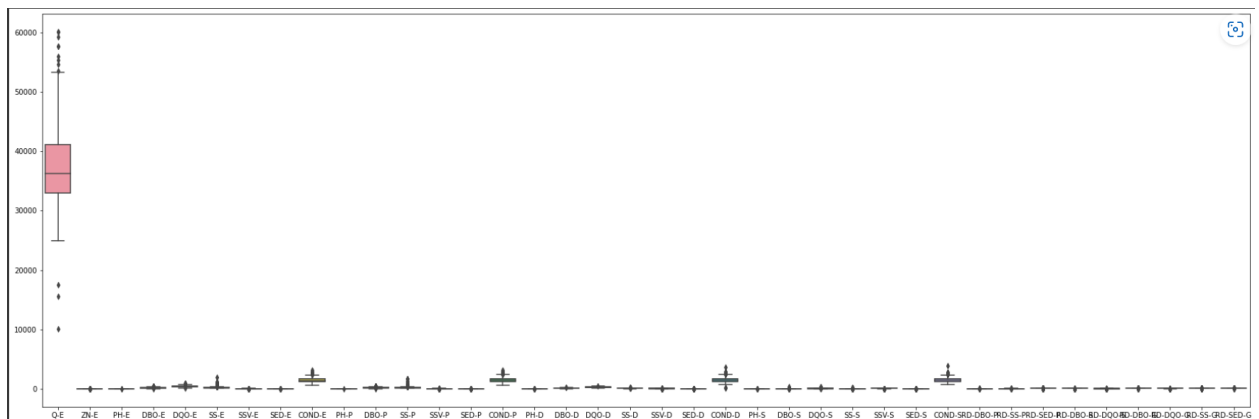
```
values = {"O-E": df_temp["O-E"].mean(), "Z-E": df_temp["ZN-E"].mean(), "DBO-E": df_temp["DBO-E"].mean(), "DOO-E": df_temp["DOO-E"].mean(), "SS-E": df_temp["SS-E"].mean(), "SVS-E": df_temp["SSV-E"].mean(), "SED-E": df_temp["SED-E"].mean(), "DBO-D": df_temp["DBO-D"].mean(), "SSV-P": df_temp["SSV-P"].mean(), "SED-P": df_temp["SED-P"].mean(), "DBO-D": df_temp["DBO-D"].mean(), "DOO-D": df_temp["DOO-D"].mean(), "SS-D": df_temp["SS-D"].mean(), "SSV-D": df_temp["SSV-D"].mean(), "SED-D": df_temp["SED-D"].mean(), "PH-S": df_temp["PH-S"].mean(), "DBO-S": df_temp["DBO-S"].mean(), "DOO-S": df_temp["DOO-S"].mean(), "SS-S": df_temp["SS-S"].mean(), "SSV-S": df_temp["SSV-S"].mean(), "COND-S": df_temp["COND-S"].mean(), "SSVS-S": df_temp["RD-DBO-P"].mean(), "RD-DBO-P": df_temp["RD-DBO-P"].mean(), "RD-RD-SEP-S": df_temp["RD-SEP-P"].mean(), "RD-SEP-P": df_temp["RD-SEP-P"].mean(), "RD-DBO-S": df_temp["RD-DBO-S"].mean(), "RD-DOO-S": df_temp["RD-DOO-S"].mean(), "RD-DOO-G": df_temp["RD-DOO-G"].mean(), "RD-SS-G": df_temp["RD-SS-G"].mean(), "RD-SEG-G": df_temp["RD-SEG-G"].mean()}
```

```
df_temp = df_temp.fillna(value=values)
```

```
[1] df_temp.isna().sum()
```

Q-E	0
ZN-E	0
PH-E	0
DBO-E	0
DQO-E	0
SS-E	0
SSV-E	0
SED-E	0

#### 4. Membuat boxplot untuk melihat outliers

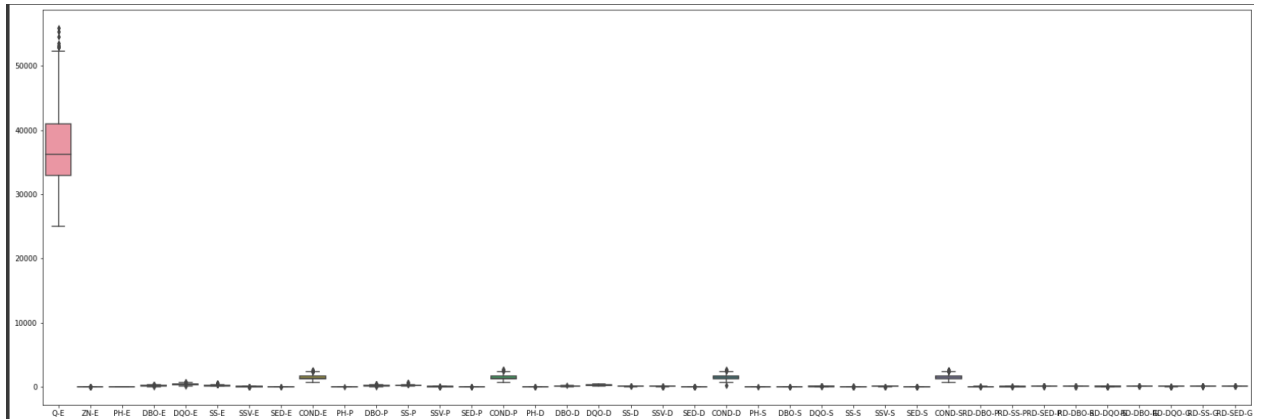


Dapat dilihat bahwa terdapat beberapa atribut yang memiliki outliers, outliers ini akan dihapus dengan menghitung z scores dan jika z scores tersebut lebih dari  $3 * \text{standar deviasi}$  maka value tersebut akan dihapus.



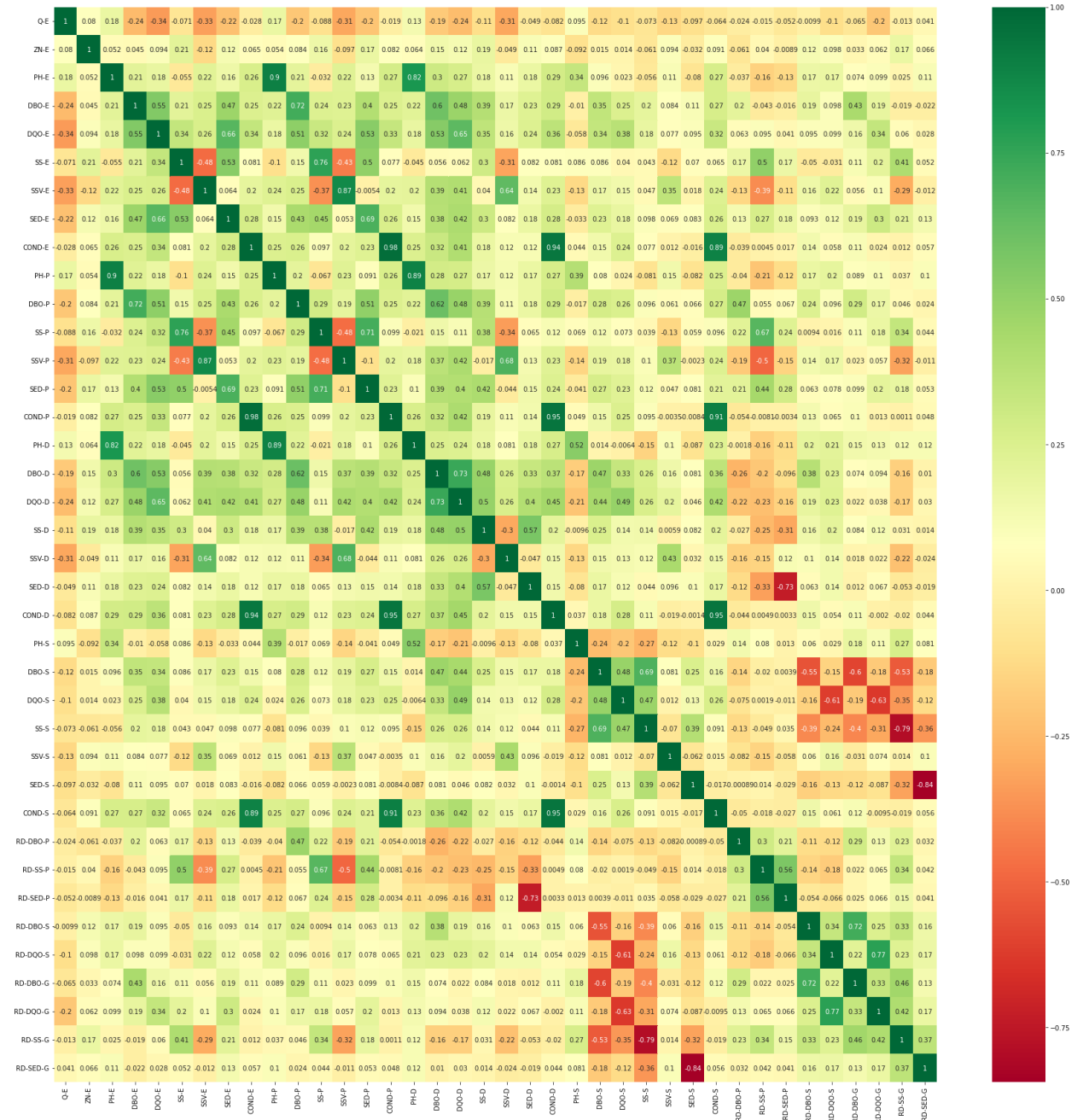
```
def remove_outliers(df):
    df = df[(np.abs(df-df.mean()) <= (3*df.std()))]
    return df
```

```
df_temp = remove_outliers(df_temp)
df_temp.describe()
```



- Memilih fitur/atribut yang akan digunakan dengan menggunakan metode pearson correlation.

```
corr = df.corr()
top_corr_features = corr.index
plt.figure(figsize=(20,20))
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



```
def select_feature(df, threshold):
    corr = set()
    cor_matrix = df.corr()
    for i in range(len(cor_matrix.columns)):
        for j in range(i):
            if abs(cor_matrix.iloc[i, j]) > threshold:
                colname = cor_matrix.columns[i]
                corr.add(colname)
    return corr
```

```
corr_features = select_feature(df_temp, 0.92)
corr_features, len(corr_features)
```

```
({'COND-D', 'COND-P', 'COND-S'}, 3)
```

```
df_selected = df_temp[corr_features]
```

Setelah dipilih maka terdapat 3 fitur yang dipilih untuk model training.

6. Melakukan Min-Max Scaling dengan nilai minimum = 1 dan maksimal = 10

```
df_scaled = ((df_selected - df_selected.min()) / (df_selected.max() - df_selected.min())) * 9 + 1
df_scaled.describe()
```

## BAB III

### 3.1 Tools Training Data

- Google Colab
- Sklearn.PCA

### 3.2 K-Means Clustering

K-means merupakan salah satu algoritma yang bersifat unsupervised learning. K-Means memiliki fungsi untuk mengelompokkan data kedalam data cluster. Algoritma ini dapat menerima data tanpa ada label kategori. K-Means Clustering Algoritma juga merupakan metode non-hierarchy. Metode Clustering Algoritma adalah mengelompokkan beberapa data ke dalam kelompok yang menjelaskan data dalam satu kelompok memiliki karakteristik yang sama dan memiliki karakteristik yang berbeda dengan data yang ada di kelompok lain.

#### 3.2 Implementasi

1. Membuat centroid dengan menggunakan sample dari dataset yang sudah di normalisasi.

```
def create_random_centroids(df, k):  
    centroids = []  
    for i in range(k):  
        centroid = df.apply(lambda x : float(x.sample()))  
        centroids.append(centroid)  
    return pd.concat(centroids, axis = 1)
```

2. Memberikan label kepada masing-masing value dengan mengitung euclidean distance.

```
def labelling(df, centroid):  
    distance = centroid.apply(lambda x : np.sqrt(((df - x) ** 2).sum(axis = 1)))  
    return distance.idxmin(axis = 1)
```

3. Menghitung centroid baru dengan cara menjumlahkan seluruh nilai logaritma dari atribut pada kelas tertentu lalu dirata-ratakan dan dihitung nilai exponensial dari rata-rata tersebut.

```
def new_centroid(df, label, k):  
    return df.groupby(label).apply(lambda x : np.exp(np.log(x).mean())).T
```

4. Membuat plot untuk visualisasi dan menggunakan pca untuk mereduksi dimensi dari data train menjadi dimensi dua.

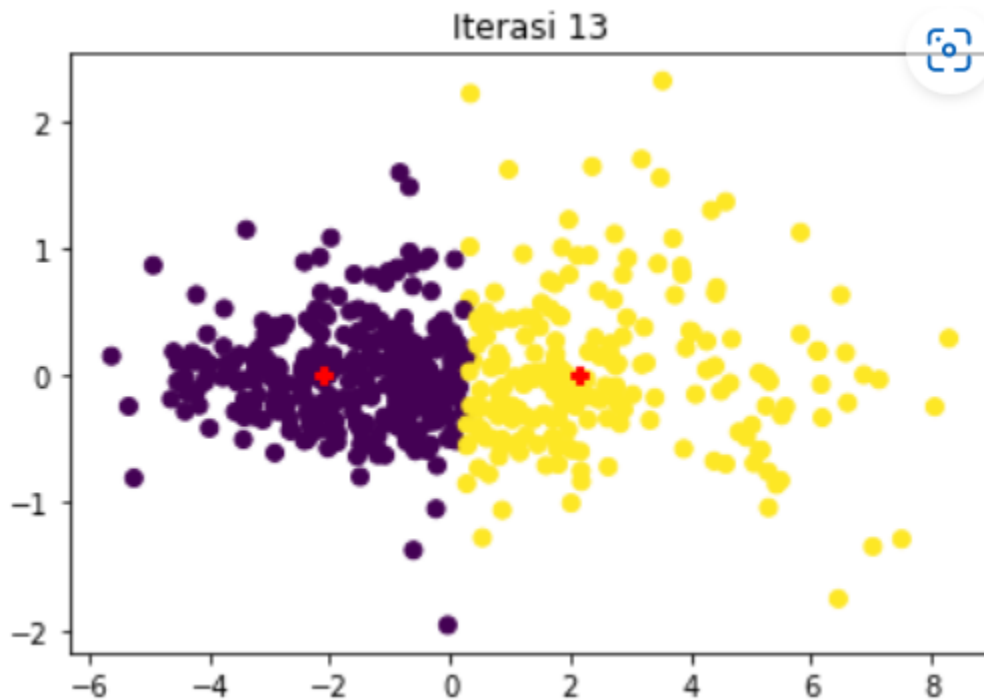
```
def plot(df, label, centroid, n):  
    pca = PCA(n_components=2)  
    df_2d = pca.fit_transform(df)  
    centroid_2d = pca.fit_transform(centroid.T)  
    plt.title(f'Iterasi {n}')  
    plt.scatter(x = df_2d[:,0], y = df_2d[:,1], c = label)  
    plt.scatter(x = centroid_2d[:,0], y = centroid_2d[:,1], marker = 'P', color = "red")  
    plt.show()
```

5. Membuat fungsi kmeans dengan looping selama centroid yang baru tidak sama dengan centroid yang lama.

```
def kmeans(df, k, nmax, centroid, old_centroid):  
    n = 1  
    while n < nmax and not centroid.equals(old_centroid):  
        old_centroid = centroid  
        label = labelling(df, centroid)  
        centroid = new_centroid(df, label, k)  
        plot(df, label, centroid, n)  
        n += 1  
    return centroid
```

6. Melakukan proses training dengan maksimum iterasi 100 dan 2 centroid.

```
centroid = create_random_centroids(df_scaled, 2)
old_centroid = pd.DataFrame()
result = kmeans(df_scaled, 2, 100, centroid, old_centroid)
```



Proses training akan berhenti pada iterasi ke 13 karena centroid tidak bergerak saat perhitungan iterasi 12 ke 13.

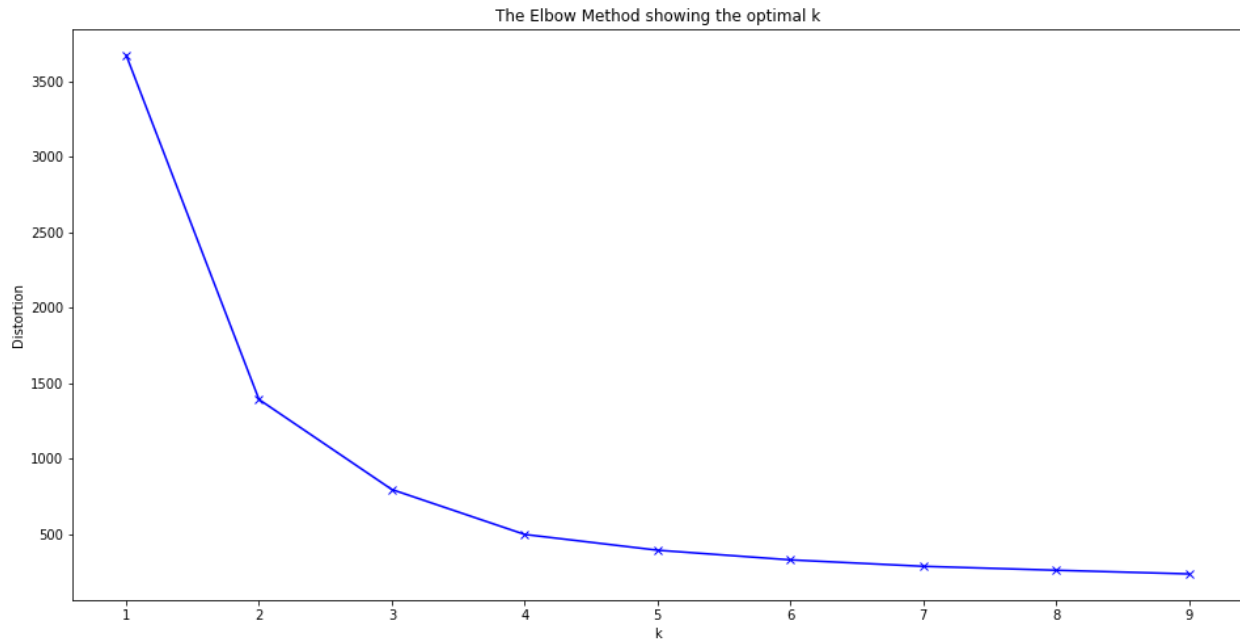
result		
	0	1
COND-S	3.559350	6.136319
COND-P	3.519081	6.139679
COND-D	4.535052	6.703823

## BAB IV

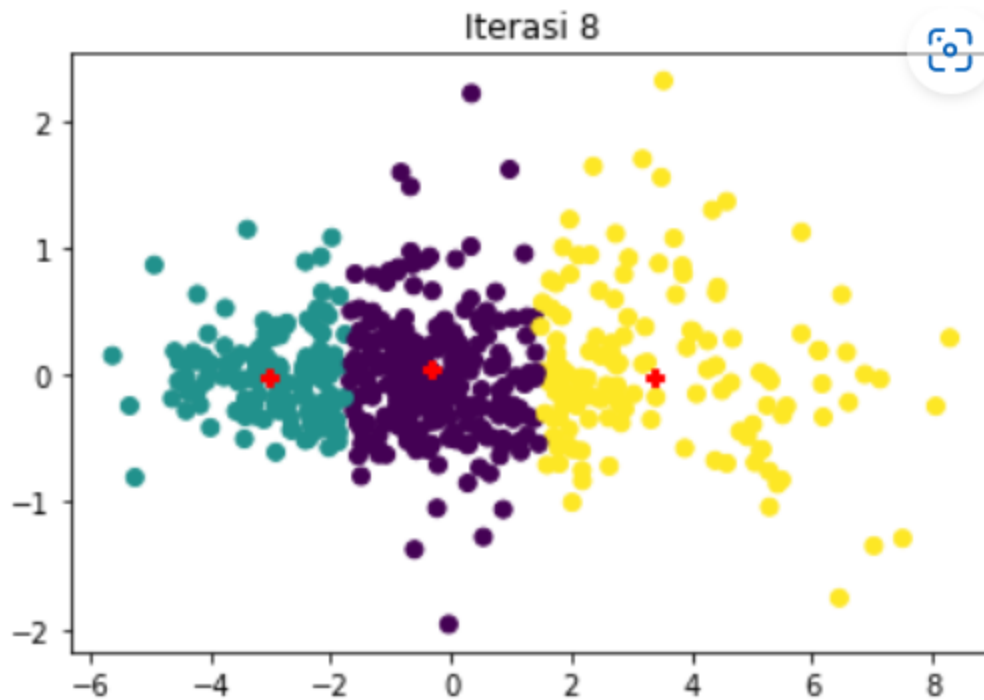
### 4.1 Evaluasi dengan Elbow Method

Elbow method digunakan untuk mencari nilai k terbaik untuk model train. Pada elbow method ini akan digunakan library sklearn.clustering yaitu kmeans untuk mencari nilai k terbaik dan akan di implementasikan kepada model yang sudah dibuat.

```
from sklearn.cluster import KMeans
distortions = []
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k)
    kmeanModel.fit(df_scaled)
    distortions.append(kmeanModel.inertia_)
```



Nilai k yang optimal adalah 3 karena nilai k mulai stabil sejak k = 3. Maka selanjutnya akan diimplementasi nilai k = 3 pada model yang sudah saya buat.



	0	1	2
<b>COND-S</b>	4.480517	2.840873	6.653450
<b>COND-P</b>	4.419416	2.802852	6.738750
<b>COND-D</b>	5.304916	3.871654	7.185851

## 4.2 Kesimpulan



Dilihat dari hasil clustering bahwa masih terdapat outliers dan beberapa titik yang salah cluster, maka untuk model ini masih belum cukup bagus. Dari hasil tersebut dapat saya simpulkan bahwa:

- Algoritma K-means sensitif terhadap outliers.
- Teknik pre-processing sangat berdampak terhadap hasil clustering.
- Pemilihan fitur untuk clustering juga berdampak terhadap clustering, semakin banyak fitur maka akan semakin banyak outliers.




## BAB V

### 5.1 Daftar Pustaka

- [Clustering Algoritma \(K-Means\) – School of Information Systems \(binus.ac.id\)](https://www.binus.ac.id/school-of-information-systems/)
- <https://stackoverflow.com/questions/23199796/detect-and-exclude-outliers-in-a-pandas-dataframe>
- <https://predictivehacks.com/k-means-elbow-method-code-for-python/>
- <https://www.geeksforgeeks.org/python-random-sample-function/>
-  K-means Clustering From Scratch In Python [Machine Learning Tutorial]
-  Tutorial 2- Feature Selection-How To Drop Features Using Pearson Correlation

### 5.2 Link

- <https://github.com/fadilmr/CaseBased2>
- [https://colab.research.google.com/drive/1Zf6DtNGwOAmPuck\\_XFWHly\\_9vUblJOle?usp=sharing](https://colab.research.google.com/drive/1Zf6DtNGwOAmPuck_XFWHly_9vUblJOle?usp=sharing)
-  ML-CaseBased02
- <https://www.youtube.com/watch?v=oXManSxLpfM>