Laporan Tugas Pemrograman
Learning

Dosen Pembimbing:
TJOKORDA AGUNG BUDI WIRAYUDA, S.T., M.T.
CII3C3-IF-44-11

Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi.

Disusun oleh:
1301204125 Ryan Chandra Hadi

S1 Informatika
Universitas Telkom
Bandung, Jawa Barat 2022

Kata Pengantar

Dengan mengucapkan puji dan rasa syukur kepada Allah SWT yang telah memberikan rahmat sehingga kita dapat menyelesaikan tugas dari mata kuliah Pembelajaran Mesin dengan tema “Implementasi K-Means/DBScan/Hierarchical)” dengan benar dan tepat waktu.

Untuk memenuhi nilai tugas pada mata kuliah Pembelajaran Mesin, maka dibuatkan tugas yang dapat saya selesaikan. Tidak hanya itu, tujuan dari pembuatan laporan dan pengerjaan tugas ini adalah untuk menambah wawasan tentang pembahasan Implementasi K-Means bagi kita semua.

Saya mengucapkan terima kasih kepada semua pihak mulai dari Pak Tjokorda Agung Budi Wirayuda, S.T., M.T. selaku dosen pembimbing yang telah memberikan saya tugas besar untuk membuat Algoritma K-Means.

Saya sangat menyadari laporan yang saya susun masih jauh dari kata sempurna. Tetapi saya akan terus berusaha untuk selalu menjadi lebih baik untuk kedepannya.

Pernyataan : Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi.

Bandung, 30 November 2022

Ryan Chandra Hadi

BAB I
PENDAHULUAN

Case-Based 2

Scenario

Mengikuti keberhasilan tugas sebelumnya, Anda diberi kesempatan lebih lanjut untuk mengesankan Atasan atau Dosen Anda mengenai kemampuan Anda untuk mengimplementasikan algoritma unsupervised dan menganalisis data. Anda diminta untuk melakukan beberapa analisis dan menghasilkan luaran yang berguna menggunakan dataset yang tersedia online sebagai berikut:

[untuk NIM akhir GENAP gunakan data ini]
https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data

[untuk NIM akhir GANJIL gunakan data ini]
https://archive.ics.uci.edu/ml/datasets/Water+Treatment+Plant

Dataset masih terdapat missing value atau outlier. Harap lakukan perbaikan terhadap hal ini, selanjutnya anda harus menganalisa data tersebut. Jika perlu konversi variable kategori menjadi integer. Jika perlu lakukan normalisasi data melalui fitur rescaling. Jika perlu lakukan analisa elbow. Jika perlu lakukan analisa dengan plot data secara visual. Jika perlu lakukan transformasi data secara logaritmik. Dan masih banyak kemungkinan Analisa data yang dapat anda lakukan.
Hint: Anda bebas memilih satu dari tiga alat analisis data yaitu Weka, R, atau Python untuk membantu Anda menganalisis data dan menunjukkan pra-pemrosesan data yang diperlukan.

Tugas Anda

Tujuan dari tugas ini yaitu Anda diharapkan mampu menjelaskan, mengimplementasikan, menganalisis, dan mendesain teknik pembelajaran mesin unsupervised learning yaitu kmeans/dbscan/hierarchical.

Pertama, selidiki masalah kualitas data yang telah diberikan di atas. Jelaskan keputusan Anda mengenai pendekatan pra-pemrosesan data. Jelajahi kumpulan data dengan meringkas data menggunakan statistik dan mengidentifikasi masalah kualitas data apa pun. Tidak ada batasan jumlah ringkasan yang akan dilaporkan tetapi Anda diharapkan hanya melaporkan yang paling relevan.

Kedua, pilih salah satu dari metode unsupervised learning yang telah dipelajari yaitu kmeans/dbscan/hierarchical. Anda hanya perlu memilih satu metode untuk diterapkan. Gunakan algoritma tersebut untuk memberikan beberapa output/outcome dengan menggunakan variasi parameter, kemudian membuat laporannya dan menganalisis hasilnya.

BAB II
PEMBAHASAN

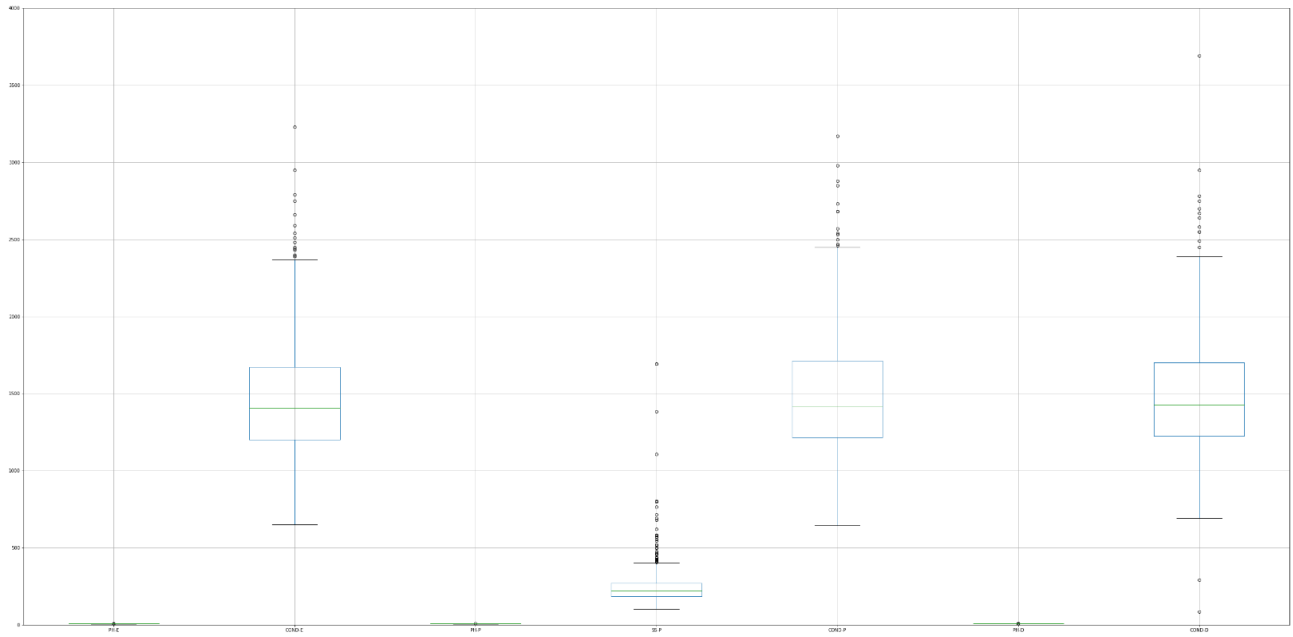
2.1 Ikhtisar Kumpulan Data yang Dipilih

Pada bagian ini saya mendapatkan data ganjil dengan nim yang diakhiri dengan "5" maka dari itu saya akan memakai data water treatment sebagai acuan pengerjaan soal.

	0	1	2	3	4	5	6	7	8	9	...	29	30	31	32	33	34	35	36	37	38
0	D-1/3/90	44101	1.50	7.8	?	407	166	66.3	4.5	2110	...	2000	?	58.8	95.5	?	70.0	?	79.4	87.3	99.6
1	D-2/3/90	39024	3.00	7.7	?	443	214	69.2	6.5	2660	...	2590	?	60.7	94.8	?	80.8	?	79.5	92.1	100
2	D-4/3/90	32229	5.00	7.6	?	528	186	69.9	3.4	1666	...	1888	?	58.2	95.6	?	52.9	?	75.8	88.7	98.5
3	D-5/3/90	35023	3.50	7.9	205	588	192	65.6	4.5	2430	...	1840	33.1	64.2	95.3	87.3	72.3	90.2	82.3	89.6	100
4	D-6/3/90	36924	1.50	8.0	242	496	176	64.8	4.0	2110	...	2120	?	62.7	95.6	?	71.0	92.1	78.2	87.5	99.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
522	D-26/8/91	32723	0.16	7.7	93	252	176	56.8	2.3	894	...	942	?	62.3	93.3	69.8	75.9	79.6	78.6	96.6	99.6
523	D-27/8/91	33535	0.32	7.8	192	346	172	68.6	4.0	988	...	950	?	58.3	97.8	83.0	59.1	91.1	74.6	90.7	100
524	D-28/8/91	32922	0.30	7.4	139	367	180	64.4	3.0	1060	...	1136	?	65.0	97.1	76.2	66.4	82.0	77.1	88.9	99
525	D-29/8/91	32190	0.30	7.3	200	545	258	65.1	4.0	1260	...	1326	39.8	65.9	97.1	81.7	70.9	89.5	87.0	89.5	99.8
526	D-30/8/91	30488	0.21	7.5	152	300	132	69.7	?	1073	...	1224	?	69.5	?	81.7	76.4	?	81.7	86.4	?

527 rows × 39 columns

Pada data di atas masih berbentuk file *raw* yang masih butuh banyak perbaikan, dari pengurangan dimensi kolom yang tidak dibutuhkan dan juga nilai yang valuenya masih bernilai '?' sehingga tidak bisa kita langsung proses untuk dilakukannya metode K-Means.



Pada boxplot di atas hanya terdeteksi 8 kolom dari 39 kolom yang tersedia yang menandakan data masih banyak yang harus diubah untuk bisa menampilkan data secara menyeluruh. Maka dari itu dibutuhkan *proses pre-processing* data agar bisa menjadi data yang lebih baik sebelum kita klasifikasi menggunakan metode K-Means.

2.2 Ringkasan pra-pemrosesan data yang diusulkan

Pada bagian pra-pemrosesan data ini saya memakai beberapa metode atau teknik diharapkan bisa membuat data water-treatment menjadi lebih baik lagi sebelum dilakukan klasifikasi terhadap dataset tersebut. Pada kesempatan ini saya menggunakan metode sebagai berikut:

1) Reduksi dimensi untuk memudahkan data untuk diproses, saya memilih kolom date karena bersifat string yang akan membuat data menjadi lebih susah untuk diproses.

```
df1 = df.drop(['Date'], axis=1)
```

2) Mengubah semua tipe data yang ada di dataset tersebut ke dalam bentuk float agar memudahkan saya untuk memproses data tersebut.

```
df1 = df1.astype(float)
```

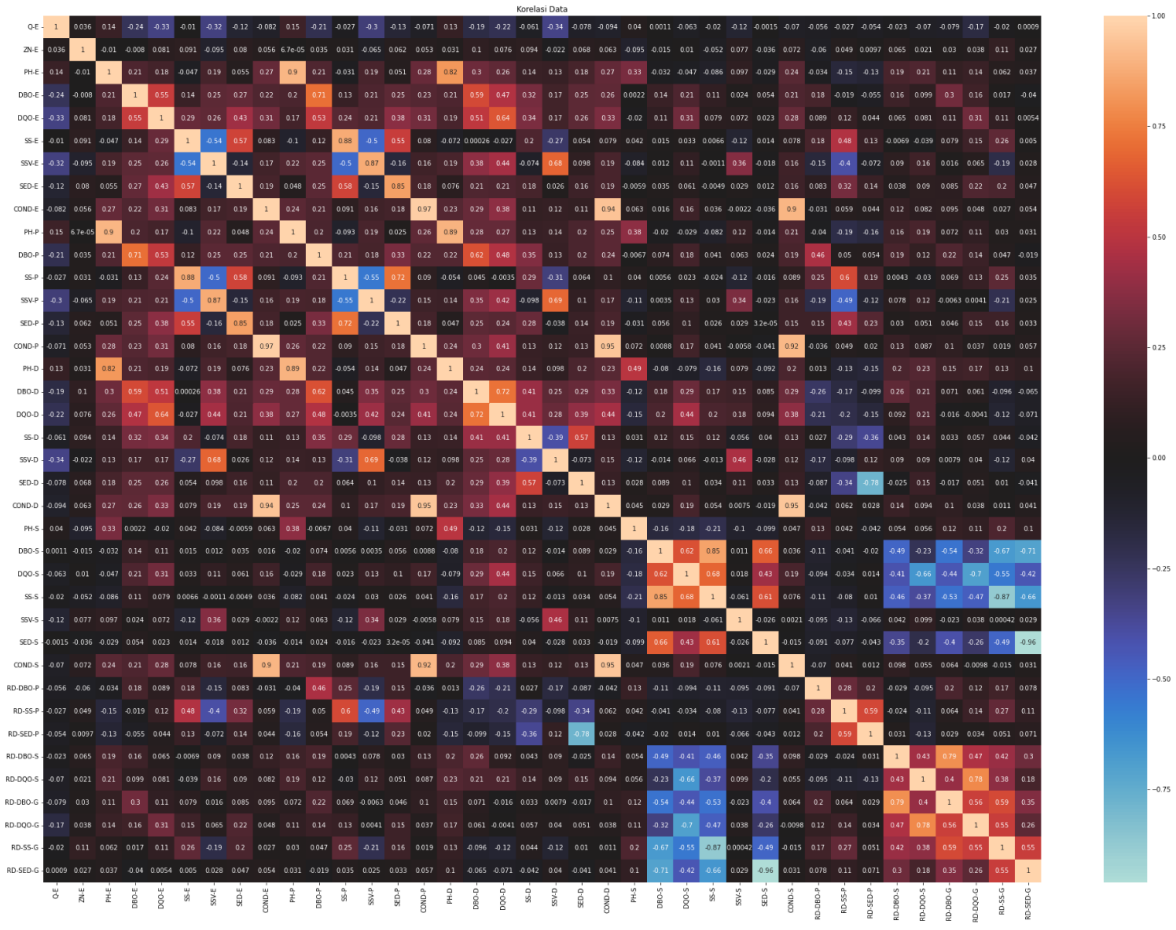
3) Me-*replace* data yang berisikan '?' menjadi nan (not a number) dan merubahnya lagi menjadi nilai mean sesuai dengan kolom yang terkait.

```
df1 = df1.replace('?', np.nan)
```

```
for column_name in df1:
  df1[column_name] = df1[column_name].fillna(df1[column_name].mean())
```

4) Menggunakan Heatmap untuk melihat korelasi yang ada pada dataset ini.

```
plt.figure(figsize=(31,25))
sns.heatmap(df1.corr(), center=0, annot=True)
plt.title("Korelasi Data")
plt.savefig("heatmap.png")
plt.show()
```



5) Menggunakan normalisasi untuk mengurangi pecilan atau outliers yang ada di dalam dataset tersebut.

```
df1 = ((df1-df1.min()) / (df1.max()-df1.min()))*99+1
```

	Q-E	ZN-E	PH-E	DBO-E	DQO-E	SS-E	SSV-E	SED-E	COND-E	PH-P	...	COND-S	RD-DBO-P	RD-SS-P	RD-SED-P	RD-DBO-S	RD-DQO-S	RD-DBO-G	RD-DQO-G	RD-SS-G	RD-SED-G
0	68.379205	5.149701	50.5	39.362934	38.527907	4.524607	74.215877	12.401685	57.006592	50.50	...	40.909091	49.536240	59.331498	95.173348	87.122045	72.188679	89.784896	76.536122	86.555556	99.377358
1	58.332974	9.595808	45.0	39.362934	42.672093	7.012565	78.214485	17.963483	78.119426	34.00	...	58.787879	49.536240	61.403084	94.422535	87.122045	83.396226	89.784896	76.661597	91.888889	100.000000
2	44.887210	15.523952	39.5	39.362934	52.456977	5.561257	79.179666	9.342697	39.962776	34.00	...	37.515152	49.536240	58.677313	95.280607	87.122045	54.443396	89.784896	72.019011	88.111111	97.665094
3	50.415902	11.077844	56.0	43.324324	59.363953	5.872251	73.250696	12.401685	69.290423	42.25	...	36.060606	41.987261	65.219163	94.958830	91.530636	74.575472	91.302326	80.174905	89.111111	100.000000
4	54.177550	5.149701	61.5	52.324324	48.773256	5.042932	72.147632	11.011236	57.006592	50.50	...	44.545455	49.536240	63.583700	95.280607	87.122045	73.226415	93.732558	75.030418	86.777778	99.221698
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
522	45.864724	1.177844	45.0	16.081081	20.684884	5.042932	61.116992	6.283708	10.328034	34.00	...	8.848485	49.536240	63.147577	92.813651	71.501734	78.311321	77.744186	75.532319	96.888889	99.377358
523	47.471488	1.652096	50.5	40.162162	31.505814	4.835602	77.387187	11.011236	13.936409	42.25	...	9.090909	49.536240	58.786344	97.640303	86.609249	60.877358	92.453488	70.513308	90.333333	100.000000
524	46.258500	1.592814	28.5	27.270270	33.923256	5.250262	71.596100	8.230337	16.700271	17.50	...	14.727273	49.536240	66.091410	96.889491	78.826590	68.452830	80.813953	73.650190	88.333333	98.443396
525	44.810038	1.592814	23.0	42.108108	54.413953	9.293194	72.561281	11.011236	24.377666	9.25	...	20.484848	50.436943	67.072687	96.889491	85.121387	73.122642	90.406977	86.072243	89.000000	99.688679
526	41.442166	1.326048	34.0	30.432432	26.210465	2.762304	78.903900	12.662602	17.199302	9.25	...	17.393939	49.536240	70.997797	89.868535	85.121387	78.830189	89.784896	79.422053	85.555556	98.577716

Pada data di atas sudah dinormalisasikan dengan range nilai antara 1 sampai 100.

6) Menggunakan library yang tidak langsung menyinggung K-Means dengan tujuan untuk memudahkan dalam proses data berbentuk tabular seperti dataset di atas.

```
# import yang dibutuhkan
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import random
from sklearn.decomposition import PCA
```

7) Meng-*import* data dari github untuk load dataset dan mengatur nilai header = none karena data tersebut tidak memiliki nama kolom sebelumnya.

```
url = 'https://raw.githubusercontent.com/bloodsking/Tupro2/main/water-treatment.csv
df = pd.read_csv(url, header=None)
```

2.3 Menerapkan algoritma K-Means

Pada kesempatan ini saya menggunakan metode K-Means untuk mengerjakan tugas ini. Metode K-Means sendiri merupakan algoritma unsupervised learning yang dipakai untuk mengelompokkan dataset yang belum di label ke dalam kluster yang berbeda. Simbol K pada K-means clustering menentukan jumlah cluster yang digunakan. Kemudian metode tersebut diaplikasikan ke dalam sebuah dataset untuk melabeli atau mengklasifikasikan data ke dalam cluster tertentu.

Dengan menggunakan ini diharapkan dapat menemukan nilai K yang paling optimal dan sesuai untuk bisa melabeli data yang tersedia. Pada K-Means ini saya menggunakan hyper parameter sebagai berikut:

1) Nilai K dengan variasi 2,3,5,7.
2) Maximal iterasi sebanyak 50 kali.

```
cluster = pd.DataFrame({
    "PH-E" : df1["PH-E"],
    "PH-P" : df1["PH-P"],
    "PH-D" : df1["PH-D"],
    "DBO-E" : df1["DBO-E"],
    "DBO-P" : df1["DBO-P"],
    "DQO-E" : df1["DQO-E"],
    "DQO-D" : df1["DQO-D"],
    "SS-E" : df1["SS-E"],
    "SS-P" : df1["SS-P"],
    "SSV-E" : df1["SSV-E"],
    "SSV-P" : df1["SSV-P"],
    "SED-E" : df1["SED-E"],
    "SED-P" : df1["SED-P"],
    "COND-E" : df1["COND-E"],
    "COND-P" : df1["COND-P"],
    })
```

Di sini saya mengambil data input pada dataset karena mereka memiliki nilai korelasi yang tinggi jika dilihat dari heatmap yang tadi telah dibuat.

```python
def random_cent(cluster, k):
    centroids = []
    random.seed(30)
    for i in range(k):
        centroid = cluster.apply(lambda x: random.uniform(1, 100))
        centroids.append(centroid)
    return pd.concat(centroids, axis=1)
```

Pada function di atas akan men-*generate* sebaran centroid secara acak dengan range 1 sampai 100 karena pada data ini kita telah menormalisasikan dataset ini dengan range 1 sampai dengan 100 dan saya juga menggunakan random.seed agar data random tidak berubah pada saat di run kembali.

```python
def label(cluster, centroids):
    canberra = centroids.apply(lambda x: (np.abs(cluster - x)/(np.abs(cluster)+np.abs(x))).sum(axis=1))
    return canberra.idxmin(axis=1)
```

$$distance(x-y) = \sum \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

Pada function di atas akan mengambil value yang nilainya masuk ke dalam rumus, di sini saya menggunakan canberra untuk menghitung jarak tersebut.

```python
def new_cent(cluster, labels, k):
    return cluster.groupby(labels).apply(lambda x: np.exp(np.log(x).mean())).T
```

Pada function di atas kita set centroid baru dengan memakai rumus geometric mean sebagai berikut.

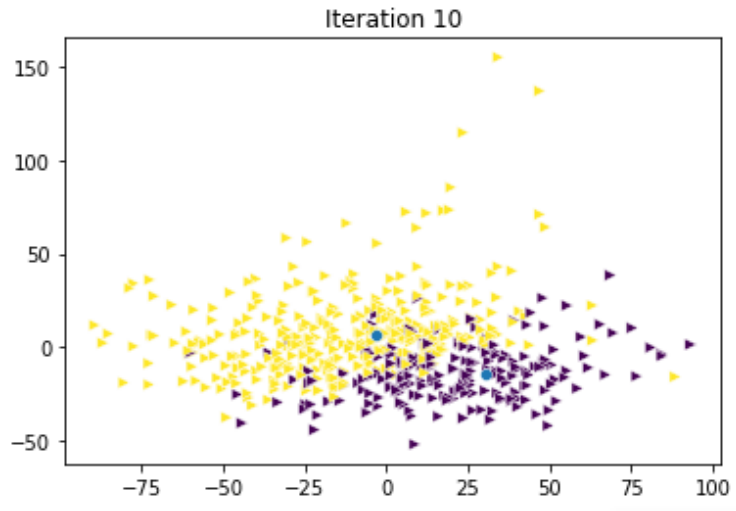
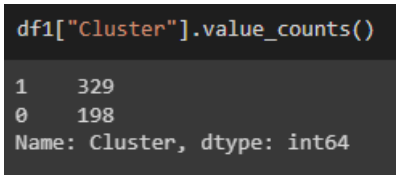
```
def plot_cluster(cluster, labels, centroids, itter):
    pca = PCA(n_components=2)
    cluster2D = pca.fit_transform(cluster)
    centroids2D = pca.transform(centroids.T)

    plt.title(f'Iteration {itter}')
    sns.scatterplot(x=cluster2D[:,0], y=cluster2D[:,1], c=labels, marker=">")
    sns.scatterplot(x=centroids2D[:,0], y=centroids2D[:,1], marker="o")
    plt.show()
```

Pada function di atas saya akan menampilkan visualisasi data berupa scatter plot berdasarkan centroid dan data yang telah ditentukan dengan mengatur marker pada value dataset berbentuk “>” sedangkan pada centroid dengan marker “o” untuk memudahkan dalam segi pembacaan data.

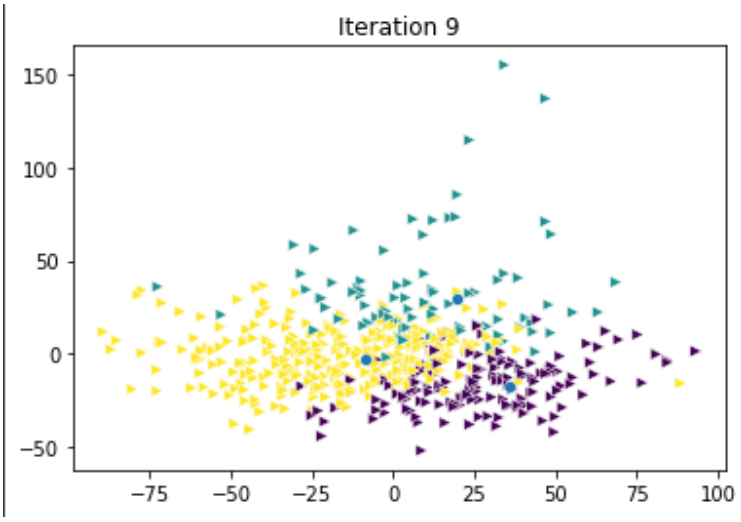
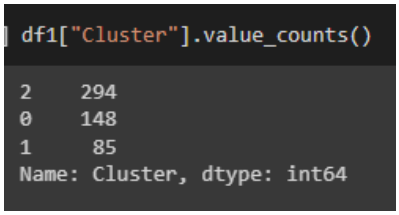
```
def final(k):
    max = 50
    centroids = random_cent(cluster, k)
    temp = pd.DataFrame()
    iteration = 1

    while iteration < max and not (centroids.equals(temp)):
        temp = centroids
        labels = label(cluster, centroids)
        centroids = new_cent(cluster, labels, k)
        plot_cluster(cluster, labels, centroids, iteration)
        df1["Cluster"] = labels
        iteration +=1
```

Pada function di atas saya akan melakukan *looping* untuk memperlihatkan visualisasi scatter plot dengan mengatur labels, centroids dan juga new centroid dengan menggunakan kondisi dimana iterasi akan berhenti jika data yang sebelumnya memiliki nilai yang sama dengan data yang sedang berjalan.

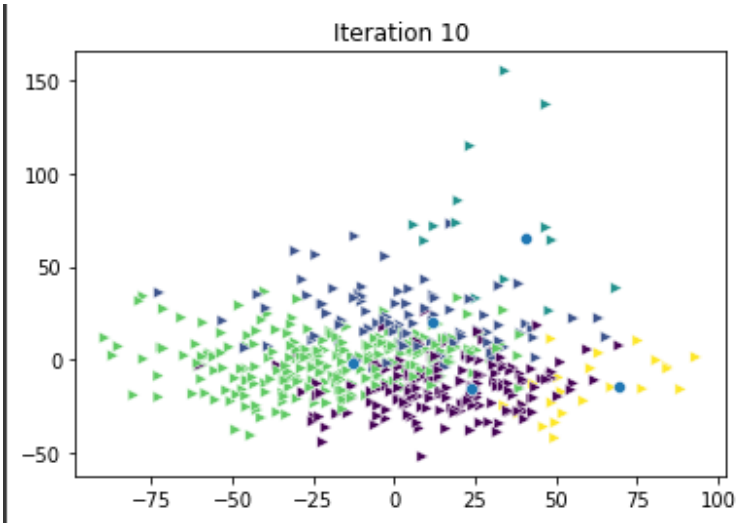
2.4 Evaluasi hasil

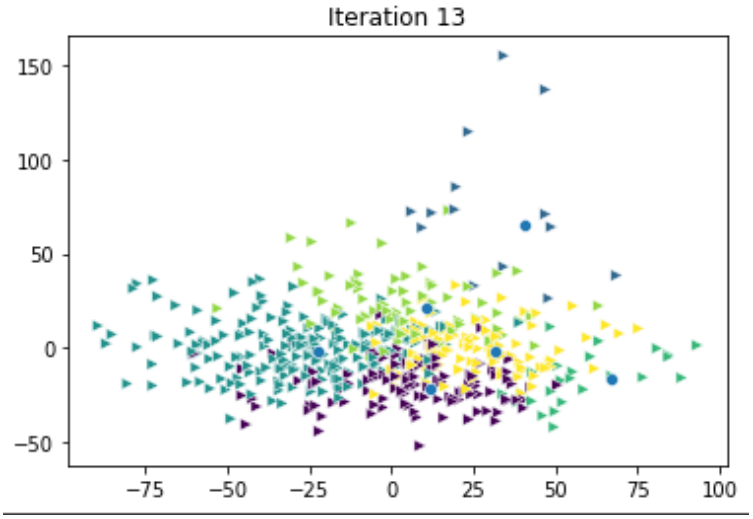
Visualisasi Iterasi ke-n	K=2
	


```
df1["Cluster"].value_counts()

1    329
0    198
Name: Cluster, dtype: int64
```

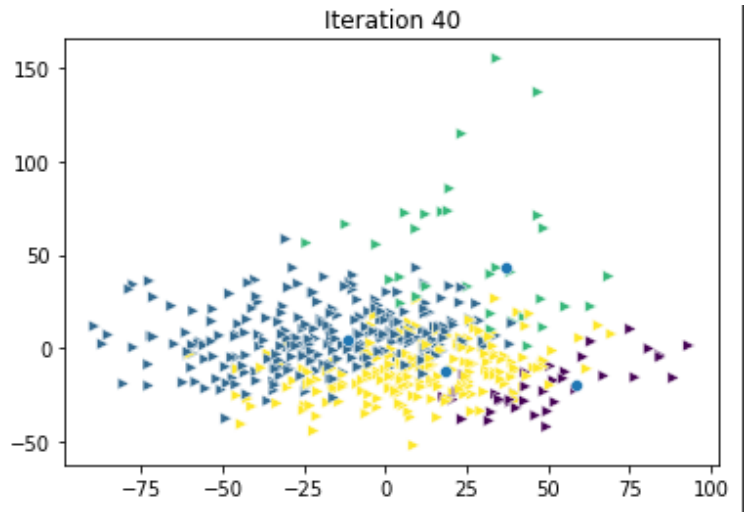
Visualisasi Iterasi ke-n	K=3
	


```
df1["Cluster"].value_counts()

2    294
0    148
1     85
Name: Cluster, dtype: int64
```

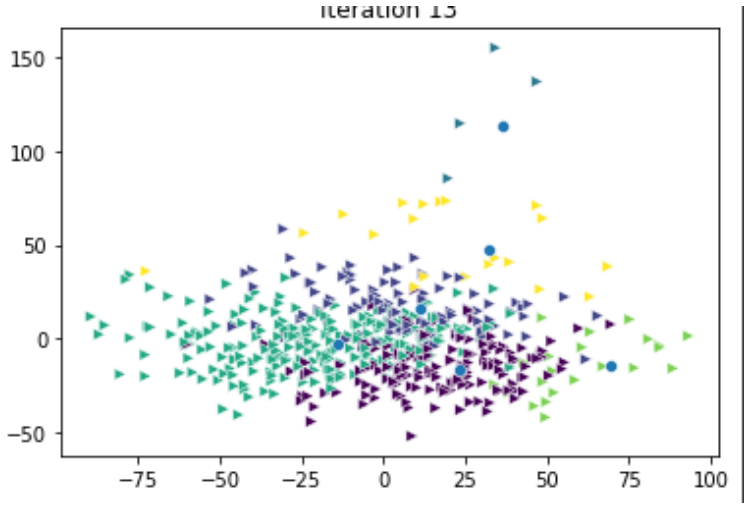
Visualisasi Iterasi ke-n	K=5
	[476] df1["Cluster"].value_counts() 3 238 0 161 1 91 4 23 2 14 Name: Cluster, dtype: int64

Visualisasi Iterasi ke-n	K=7
	df1["Cluster"].value_counts() 3 191 0 113 6 108 5 76 4 25 2 14 Name: Cluster, dtype: int64

Visualisasi Iterasi ke-n	K=4

	`df1["Cluster"].value_counts()`
	`1  257`
	`3  200`
	`0  37`
	`2  33`
	`Name: Cluster, dtype: int64`

Visualisasi Iterasi ke-n	K=6

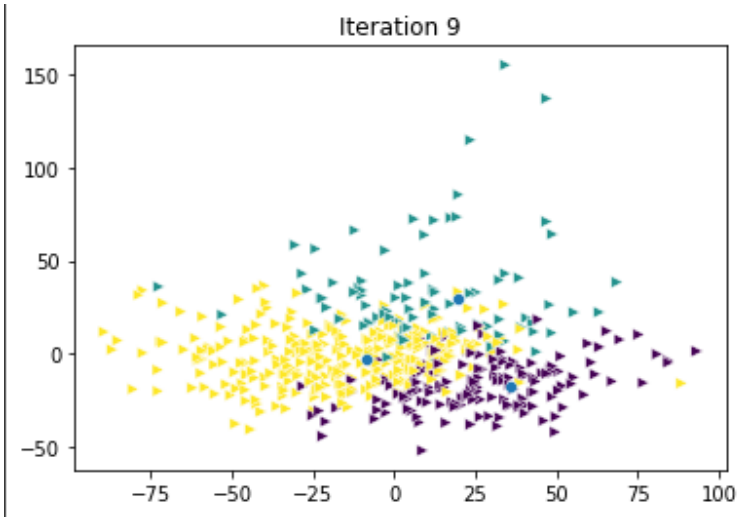
	`df1["Cluster"].value_counts()`
	`3  224`
	`0  156`
	`1  100`
	`4  23`
	`5  20`
	`2  4`
	`Name: Cluster, dtype: int64`

BAB III
KESIMPULAN

Setelah kita perhatikan pada scatter plot yang telah kita visualisasikan, kita bisa melihat nilai k yang paling optimal dengan melihat persebaran data di setiap cluster ada pada nilai k=3 dengan persebaran data sebagai berikut,

```
df1["Cluster"].value_counts()
```

```
2    294
0    148
1     85
Name: Cluster, dtype: int64
```



Saya tidak menggunakan elbow method dikarenakan masih belum mengimplementasikannya tanpa menggunakan library, tetapi dengan pengamatan yang memanfaatkan scatter plot dan nilai pada setiap cluster kita bisa lihat nilai k yang optimal dengan memperhatikan persebaran data yang terlihat pada data yang ditampilkan.

Referensi:

- https://sis.binus.ac.id/2022/01/31/clustering-algoritma-k-means/
- https://stackabuse.com/k-means-elbow-method-and-silhouette-analysis-with-yellowbrick-and-scikit-learn/
- https://www.statology.org/canberra-distance-python/
- https://www.w3schools.com/python/python_lambda.asp

Laporan =
https://docs.google.com/document/d/13bsGExXKPI4hytoWSI3A5K7weQjZ80R-y9WjThCw_X4/edit?usp=sharing

Slide =
https://www.canva.com/design/DAFRYgf5R2k/TKBGWmYH2RsBal8lEEqfGg/view?utm_content=DAFRYgf5R2k&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Colab =
https://colab.research.google.com/drive/1yZRqkUj30U76_Q6k_GMvE6sMBZw09V4F?usp=sharing

Video Laporan =
https://youtu.be/35DrUxCLVCQ