LAPORAN

MACHINE LEARNING : CASE-BASED 2

Disusun untuk memenuhi tugas case-based 2

Mata Kuliah CII3C3 - Machine Learning

Oleh :

1301204351 - Nico Valentino

S1 INFORMATIKA
TELKOM UNIVERSITY
2022

KATA PENGANTAR

Dengan memanjatkan Puji Syukur kehadirat Tuhan Yang Maha Esa atas karunia dan rahmat-Nya, saya dapat menyusun makalah penugasan yang berjudul "LAPORAN MACHINE LEARNING : CASE-BASED 2'' dengan lancar.

Adapun maksud penyusunan laporan ini untuk memenuhi tugas Mata Kuliah CII3C3 - Machine Learning. Rasa terima kasih saya tidak terkirakan kepada yang terhormat Bapak Dosen selaku pembimbing mata kuliah Machine Learning, serta semua pihak yang telah membagikan pengetahuannya dalam penyusunan makalah ini yang tidak bisa saya sebutkan satu persatu.

Saya menyadari bahwa laporan ini masih jauh dari kata sempurna dengan keterbatasan yang saya miliki. Oleh karena itu, dengan tangan terbuka saya menerima segala saran dan kritik dari pembaca yang bersifat membangun agar saya dapat memperbaiki dan menyempurnakan dalam pembuatan laporan kedepannya.

Bandung, 2 Desember 2022

Nico Valentino

BAB I
PENDAHULUAN

1.1 Skenario Tugas

Anda diminta untuk melakukan beberapa analisis dan menghasilkan seperangkat aturan yang berguna menggunakan dataset berikut.
https://archive.ics.uci.edu/ml/datasets/Water+Treatment+Plant

Anda perlu mempelajari data dengan hati-hati. Kemudian pilih teknik pra-pemrosesan data apa yang akan dilakukan untuk meningkatkan kualitas data tersebut. Akan ada banyak hal yang harus diuraikan dan kemudian Anda harus mengumpulkan case-based ini sebagai karya individu.

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. 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.

1.2 Tujuan Tugas

Adapun tujuan dari tugas ini yaitu mahasiswa diharapkan mampu menjelaskan, mengimplementasikan, menganalisis, dan mendesain algoritma unsupervised learning yang telah dipelajari yaitu kmeans/dbscan/hierarchical serta menghasilkan beberapa output/outcome dengan menggunakan variasi parameter.

BAB II

IMPLEMENTASI

2.1 Kumpulan Data yang Dipilih

Pada analisis data kali ini, data yang digunakan berasal dari pengukuran harian sensor di instalasi pengolahan air limbah perkotaan. Tujuannya adalah untuk mengklasifikasikan keadaan operasional pembangkit untuk memprediksi kesalahan melalui variabel keadaan pembangkit pada setiap tahapan proses perawatan. Domain ini telah dinyatakan sebagai domain yang tidak terstruktur. Dataset dapat diakses pada link https://archive.ics.uci.edu/ml/datasets/Water+Treatment+Plant.
(Data tidak ditampilkan secara keseluruhan untuk menghemat ruang)

	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 dataset water treatment terdapat jumlah baris data sebanyak 527, dengan 39 kolom, dengan deskripsi sebagai berikut :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 527 entries, 0 to 526
Data columns (total 39 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       527 non-null    object
 1   1       527 non-null    object
 2   2       527 non-null    object
 3   3       527 non-null    float64
 4   4       527 non-null    object
 5   5       527 non-null    object
 6   6       527 non-null    object
 7   7       527 non-null    object
 8   8       527 non-null    object
 9   9       527 non-null    int64
 10  10      527 non-null    float64
 11  11      527 non-null    object
 12  12      527 non-null    int64
 13  13      527 non-null    object
 14  14      527 non-null    object
 15  15      527 non-null    int64
 16  16      527 non-null    float64
 17  17      527 non-null    object
 18  18      527 non-null    object
 19  19      527 non-null    object
 20  20      527 non-null    object
 21  21      527 non-null    object
 22  22      527 non-null    int64
 23  23      527 non-null    object
 24  24      527 non-null    object
 25  25      527 non-null    object
 26  26      527 non-null    object
 27  27      527 non-null    object
 28  28      527 non-null    object
 29  29      527 non-null    object
 30  30      527 non-null    object
 31  31      527 non-null    object
 32  32      527 non-null    object
 33  33      527 non-null    object
 34  34      527 non-null    object
 35  35      527 non-null    object
 36  36      527 non-null    object
 37  37      527 non-null    object
 38  38      527 non-null    object
dtypes: float64(3), int64(4), object(32)
memory usage: 160.7+ KB
```

2.2 Pre-processing Data

Pre-processing data adalah proses penurunan data sebelum digunakan untuk memastikan atau meningkatkan kinerja, dan merupakan langkah penting dalam proses penambangan data.

2.2.1 Dropping Data

Tahap pertama adalah membuang data yang dianggap tidak dibutuhkan dalam proses clustering. Didapati bahwa terdapat kolom yang berisikan tanggal yaitu pada kolom pertama. Karena kolom tanggal tidak dapat digunakan dalam proses clustering, oleh karena itu kolom tersebut akan dihapus. Lalu membuang data yang duplikat, namun pada dataset ini tidak terdapat data yang duplikat.

```
[222] #Karena pada kolom dengan index 0 merupakan kolom date yang bertipe data string
      #dan tidak bisa diolah dalam clustering, oleh karena itu kolom tersebut akan di drop.
      df.drop(0, axis='columns' ,inplace = True)
```

```
[223] # Drop data yang duplikat
      df = df.drop_duplicates(keep = 'first')
```

2.2.2 Missing Value

Tahap kedua yaitu mengolah missing value. Pada dataset water treatment, terdapat banyak data kosong yang berisi string ‘?’. Karena data yang berisi string ‘?’ ini akan mengganggu proses analisis data, oleh karena itu akan diganti oleh mean dan median tergantung oleh nilai skewness nya. Nilai skewness yang besar menandakan sebaran data nya variatif, yaitu skewnessnya lebih kecil dari -1 atau lebih besar 1 akan diisi oleh median, sedangkan sisanya akan diisi oleh mean.

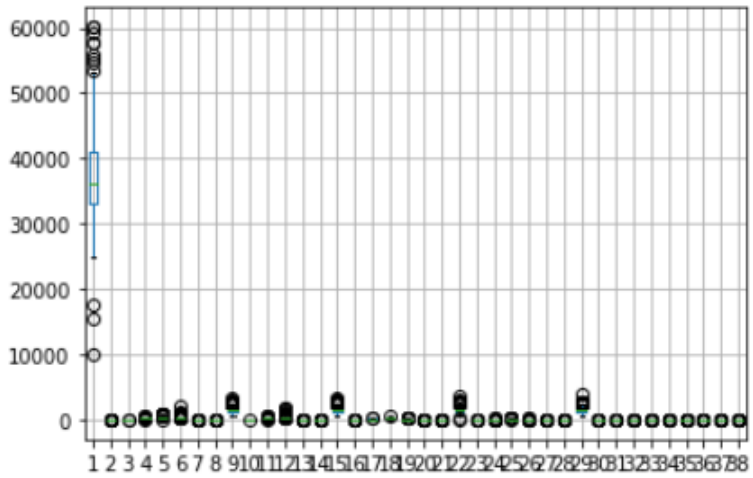
```
[231] #Setelah itu, nilai '?' yang sudah diubah menjadi NULL tadi, diisi dengan median
      for i in range(1, 39):
        skewness = df[i].skew(axis = 0 , skipna = True)
        if skewness <= 1 and skewness >= -1 :
          df[i] = df[i].fillna(df[i].mean())
        else :
          df[i] = df[i].fillna(df[i].median())
```

2.2.3 Handling Outlier

Tahap ketiga adalah melakukan koreksi terhadap outlier. Outlier adalah data yang memiliki karakteristik unik yang terlihat sangat berbeda dibanding data-data yang lainnya.

```
pd.DataFrame.boxplot(df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f57b18682e0>
```



Pada box plot di atas dapat dilihat terdapat banyak outlier yang ada di banyak kolom. Oleh karena itu akan outlier akan dihandling dengan cara memasukkan nilai maksimum dan minimum yang didapatkan dari perhitungan interkuartil ke dalam data

yang merupakan outlier pada dataset.

```
[236] def interquartile(df,x):
        q1 = (df[x].quantile(0.25))
        q3 = (df[x].quantile(0.75))
        interquartile = q3 - q1
        maximum = q3 + (1.5 * interquartile)
        minimum = q1 - (1.5 * interquartile)
        return maximum, minimum
```

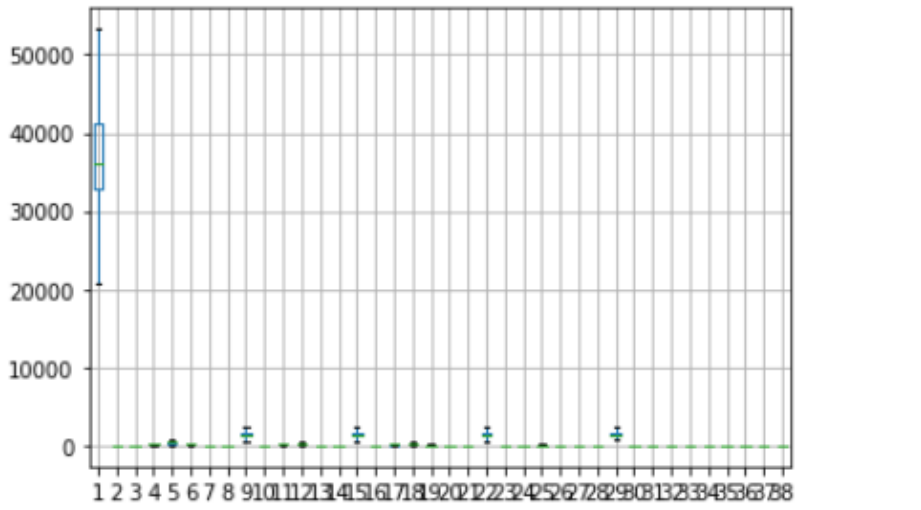
```
[237] def sub_outliners(df, x, maximum, minimum):
        more_than = (df[x] > maximum)
        less_than = (df[x] < minimum)
        print('more_than: ', more_than,'| less_than: ', less_than)
        df[x] = df[x].mask(more_than, maximum, axis = 0)
        df[x] = df[x].mask(less_than, minimum, axis = 0)
        return df
```

```
[238] # menghilangkan outlier
      for i in range(1,39):
        maximum, minimum = interquartile(df, i)
        df = sub_outliners(df, i, maximum, minimum)
```

Berikut adalah tampilan box plot setelah outliernya berhasil diatasi.

```
pd.DataFrame.boxplot(df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f57b0481a30>
```



2.2.4 Scaling

Scaling dilakukan agar skala persebarannya memperkecil perbedaan antar data. Pada kasus ini, scaling dilakukan dengan bantuan library StandardScaler, lalu data hasil scaling akan disimpan di data frame baru yang bernama df_baru.

```
[280] scale = StandardScaler()

      scale.fit(df)
      temp = scale.transform(df)
```

2.2.5 Selected Feature

Tahap Selected Feature atau biasa disebut Feature Selection adalah tahap pemilihan kolom atau fitur yang akan digunakan pada proses modelling data. Kolom yang dipilih untuk digunakan pada tahap modelling data hanya yang dianggap relevan, tahap ini dilakukan untuk menghindari penggunaan data yang tidak relevan pada proses modelling data yang dapat mengganggu hasil dari proses modelling.

Pertama-tama, setiap kolom pada dataset diberikan nama untuk memudahkan proses feature selection.

```
[ ]  #Memberikan nama kepada kolom untuk membantu proses selected feature
     df.columns =['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' ,'SSV-D' ,'SED-D' ,'COND-D' ,'PH-S' ,'DBO-S' ,'DQO-S' ,
                  'SS-S' ,'SSV-S' ,'SED-S' ,'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']

     df_baru = pd.DataFrame(temp, index = df.index , columns = df.columns)
     df_baru
```

Setelah itu, simpan semua data yang akan digunakan pada proses modelling ke dalam sebuah data frame yang baru. Pada kasus kali ini, kolom yang digunakan untuk proses

modelling adalah data input yang terdapat pada dataset.

```
[ ]  # Memilih data yang merupakan data input pada dataset untuk digunakan pada proses modelling
     df_baru=df_baru[['Q-E', 'ZN-E', 'PH-E', 'DBO-E' ,'DQO-E' ,'SS-E' ,'SSV-E','SED-E','COND-E']]
     df_baru.head()
```

	Q-E	ZN-E	PH-E	DBO-E	DQO-E	SS-E	SSV-E	SED-E	COND-E
0	1.112007	-0.362106	-0.041111	0.018015	0.016779	-0.750428	0.397841	0.052287	1.696286
1	0.294148	0.523991	-0.449899	0.018015	0.333653	0.035415	0.656079	1.264844	2.410516
2	-0.800465	1.705455	-0.858686	0.018015	1.081826	-0.422993	0.718413	-0.614619	0.515214
3	-0.350377	0.819357	0.367676	0.301720	1.609949	-0.324763	0.335508	0.052287	2.410516
4	-0.044143	-0.362106	0.776463	0.946275	0.800161	-0.586710	0.264270	-0.250852	1.696286

Lalu, semua data input yang sudah didapatkan tadi, direduksi dimensinya menjadi 3 dimensi dengan bantuan library PCA.

```
[ ]  # Menggunakan bantuan library PCA untuk mereduksi dimensi
     # yang ada pada dataframe hingga menjadi 3 dimensi
     test = PCA(n_components = 3)

     df_cluster = test.fit_transform(df_baru)
     df_cluster = pd.DataFrame(data = df_cluster, columns = ['x', 'y', 'z'])
     df_cluster
```

	x	y	z
0	0.035587	-0.727052	1.213437
1	1.518799	-0.207117	0.809469
2	0.613644	-0.474792	-0.419002
3	1.878416	-0.767187	1.024496
4	1.146193	-1.223291	0.976359
...	...	...	...
522	-2.768199	-0.174390	-1.322011
523	-0.861114	-1.050322	-1.050088
524	-1.662375	-0.431242	-2.060587
525	0.443668	0.115293	-2.492323
526	-1.459186	-1.213855	-1.982486

527 rows × 3 columns

2.3 Implementasi K-Means

2.3.1 Membangun Algoritma K-Means

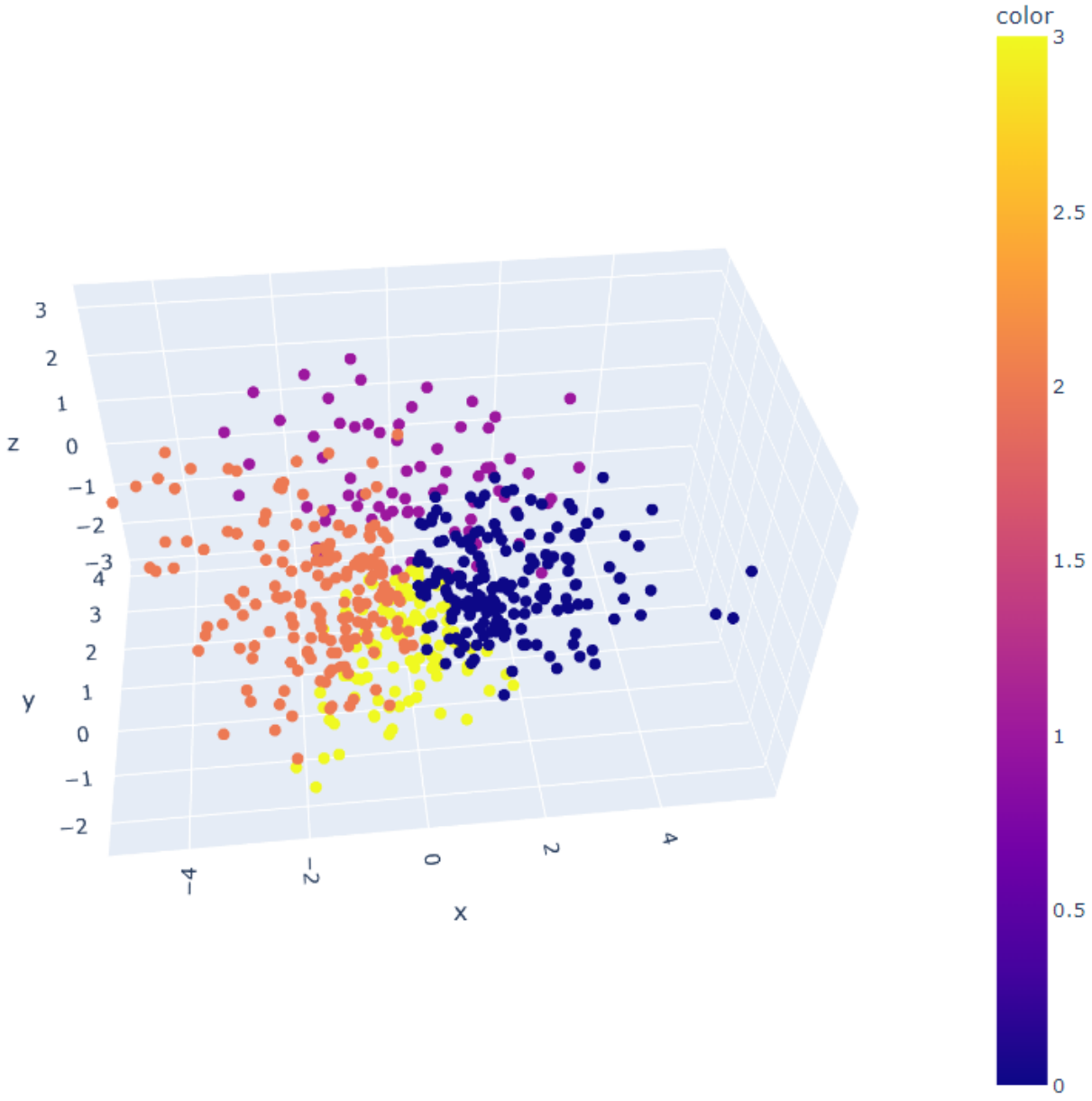
K-Means merupakan salah satu algoritma unsupervised learning. K-Means merupakan metode non-hierarchy. Tujuan algoritma K-Means adalah mengelompokkan data menjadi sebanyak K-cluster, dimana K ditentukan oleh user. Untuk membangun algoritma K-Means, dibutuhkan bantuan function penghitungan euclidean distance, serta library numpy untuk generate angka random yang akan digunakan sebagai centroid awal. Centroid pertama di generate dengan range antara nilai terkecil hingga nilai terbesar pada data yang akan dilakukan clustering.

```
def euclidean(a, b, ax = 1):
  return np.linalg.norm(a-b, axis = ax)

def centroid(array_xyz, k):
  min = np.min(array_xyz)
  max = np.max(array_xyz)

  centroid1 = np.random.randint(min, max, size = k)
  centroid2 = np.random.randint(min, max, size = k)
  centroid3 = np.random.randint(min, max, size = k)
  centroid = np.array(list(zip(centroid1, centroid2, centroid3)))
  return centroid
```

Lalu dilakukan algoritma K-Means, dengan loop yang akan berhenti jika nilai centroid tidak berubah, atau nilai centroid yang lama dengan nilai centroid yang baru memiliki value euclidean distance 0. Pada percobaan kali ini, digunakan nilai K sebesar 4 yang berarti nanti akan terbentuk 4 cluster.

```
[ ]  array_cluster = np.zeros(len(array_xyz))

     #menentukan centroid secara random untuk 3 cluster
     k = 4
     centroid = centroid(array_xyz, k)
     array_centroid = np.zeros(centroid.shape)

     titik = []
     temp = []

     mark = euclidean(centroid, array_centroid, None)

     while mark != 0:
       for i in range(len(array_xyz)):
         jarak = euclidean(array_xyz[i], centroid)
         cluster = np.argmin(jarak)
         array_cluster[i] = cluster
       array_centroid = deepcopy(centroid)

       for i in range(k):
         titik = [array_xyz[j] for j in range(len(array_xyz)) if array_cluster[j] == i]
         centroid[i] = np.mean(titik, axis = 0)
         temp.append(array_cluster)

       mark = euclidean(centroid, array_centroid, None)
```

Lalu didapatkan hasil clustering dengan persebaran data sebagai berikut



Nilai akhir dari centroid adalah

```
[ ]  centroid

     array([[ 1,  0,  0],
            [ 0,  2,  0],
            [-1,  0,  0],
            [ 0,  0, -1]])
```

Setelah itu, masukkan hasil clustering ke dalam data frame

```
[ ] df["Cluster"] = array_cluster
    df
```

	Q-E	ZN-E	PH-E	DBO-E	DQO-E	SS-E	SSV-E	SED-E	COND-E	PH-P	...	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	Cluster
0	44101.0	1.50	7.8	188.714286	407.0	166.0	66.3	4.5	2110.0	7.9	...	39.085806	58.8	95.5	85.40	70.0	90.20	79.4	87.3	99.6	0.0
1	39024.0	3.00	7.7	188.714286	443.0	214.0	69.2	6.5	2378.5	7.7	...	39.085806	60.7	94.8	85.40	80.8	90.20	79.5	92.1	100.0	0.0
2	32229.0	5.00	7.6	188.714286	528.0	186.0	69.9	3.4	1666.0	7.7	...	39.085806	58.2	95.6	85.40	52.9	90.20	75.8	88.7	98.5	0.0
3	35023.0	3.50	7.9	205.000000	588.0	192.0	65.6	4.5	2378.5	7.8	...	33.100000	64.2	95.3	87.30	72.3	90.20	82.3	89.6	100.0	0.0
4	36924.0	1.50	8.0	242.000000	496.0	176.0	64.8	4.0	2110.0	7.9	...	39.085806	62.7	95.6	85.40	71.0	92.10	78.2	87.5	99.5	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
522	32723.0	0.16	7.7	93.000000	252.0	176.0	56.8	2.3	894.0	7.7	...	39.085806	62.3	93.3	72.95	75.9	81.05	78.6	96.6	99.6	2.0
523	33535.0	0.32	7.8	192.000000	346.0	172.0	68.6	4.0	988.0	7.8	...	39.085806	58.3	97.8	83.00	59.1	91.10	74.6	90.7	100.0	3.0
524	32922.0	0.30	7.4	139.000000	367.0	180.0	64.4	3.0	1060.0	7.5	...	39.085806	65.0	97.1	76.20	66.4	82.00	77.1	88.9	99.0	3.0
525	32190.0	0.30	7.3	200.000000	545.0	258.0	65.1	4.0	1260.0	7.4	...	39.800000	65.9	97.1	81.70	70.9	89.50	87.0	89.5	99.8	3.0
526	30488.0	0.21	7.5	152.000000	300.0	132.0	69.7	4.5	1073.0	7.4	...	39.085806	69.5	93.3	81.70	76.4	90.20	81.7	86.4	99.7	3.0

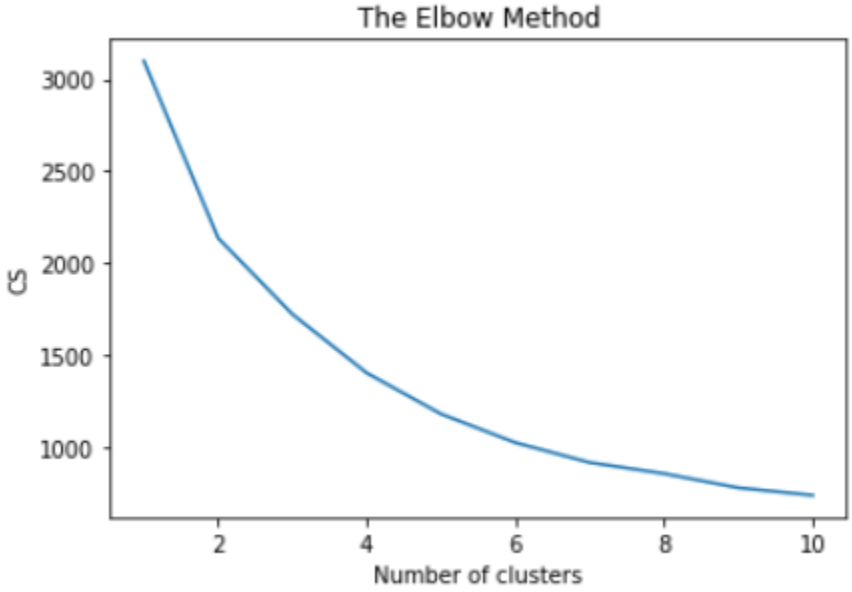
527 rows × 39 columns

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

```
0.0    180
2.0    169
3.0    107
1.0     71
Name: Cluster, dtype: int64
```

2.3.1 Evaluasi K-Means

Pada tahap evaluasi, digunakan metode Elbow Plot untuk mengecek apakah nilai K yang digunakan pada proses modelling sudah tergolong baik.



Setelah ditampilkan Elbow Plot seperti gambar di atas, dapat disimpulkan nilai K terbaik untuk dataset yang diberikan adalah 2, karena pada saat nilai K = 2, terbentuk Sudut paling kecil atau mendekati siku-siku dibanding nilai K yang lain.

BAB III
PENUTUP

3.1 Kesimpulan

Dari kajian analisis ini dapat disimpulkan bahwa algoritma K-Means dapat mengelompokkan data keadaan operasional dengan memanfaatkan data input dari catatan pengukuran harian sensor di instalasi pengolahan air limbah perkotaan. Terbukti bahwa algoritma K-Means dapat mengelompokkan data dari suatu dataset ke dalam sebanyak K cluster setidaknya untuk kasus ini.

3.2 Evaluasi

Pada kasus ini, digunakan K = 4 pada proses modelling data, namun pada saat ditampilkan nilai K terbaik menurut elbow plot, didapati nilai 2 sebagai nilai terbaik. Namun perlu ditinjau kembali bahwa elbow plot hanyalah salah satu metode untuk menentukan nilai K terbaik dan nilai K yang dihasilkan bukan sesuatu yang mutlak, atau belum tentu nilai K yang dihasilkan merupakan yang terbaik.

Lampiran

Link google colab :
https://colab.research.google.com/drive/1dysbu_KqiRPXJIM0mk4TgvX8YEiH2BuB?usp=sharing

Link video youtube :
https://youtu.be/AdBgpg9gUDY

Link Slide :
https://www.canva.com/design/DAFTx1aUbHI/LDiyQ9VSnFR_Yn5I9X-12A/view?utm_content=DAFTx1aUbHI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Referensi

https://sis.binus.ac.id/2022/01/31/clustering-algoritma-k-means/

https://www.kaggle.com/code/prashant111/k-means-clustering-with-python

https://www.youtube.com/watch?v=mX_rZc46FNo&ab_channel=MangSaswj