TUGAS BESAR
PEMBELAJARAN MESIN TAHAP 1

Laporan Observasi

Ditujukan untuk memenuhi salah satu tugas mata kuliah Pembelajaran Mesin.

Oleh:
Muhammad Daffa Regenta S (1301184291)
Email: daffaregenta@student.telkomuniversity.ac.id

IF-42-07

PROGRAM STUDI INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2021

Muhammad Daffa Regenta S - 1301184291
IF-42-07

A. Formulasi Masalah

Untuk tugas besar yang akan dilakukan ini adalah melakukan metode task clustering dengan mem-prediksi apakah besok akan turun salju apa tidak menggunakan dataset salju yang sudah diberikan.

B. Eksplorasi dan Persiapan Data

Untuk persiapan data, menggunakan dataset salju yang sudah diberikan. Dataset tersebut di analisis terlebih dahulu apakah ada kecacatan terhadap dataset apa tidak, dengan memperhatikan isi dataset tersebut sesuai apa tidak, contoh nya dengan memperhatikan isi dataset tersebut lengkap, cacat, tidak sesuai dengan kategori, tidak konsisten, atau cacat.

Cara yang dilakukan adalah:

1. Import dataset salju

Import Data

```
1 data = pd.read_csv('salju_train.csv')
2 data
```

	id	Tanggal	KodeLokasi	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	ArahAnginTerkencang
0	1	01/06/2014	C4	10.4	15.5	4.8	NaN	NaN	WSW
1	2	15/07/2014	C10	9.0	17.0	8.0	2.6	7.4	NaN
2	3	16/02/2011	C46	18.2	32.0	0.0	NaN	NaN	ESE
3	4	08/08/2012	C36	7.3	24.5	0.0	8.4	10.4	SSW
4	5	29/10/2016	C7	5.9	20.3	0.0	3.6	12.6	N
...	...	...	...	...	...	...	...	...	...
109090	109091	31/01/2009	C38	20.1	23.7	0.0	7.2	8.9	ESE
109091	109092	03/11/2010	C16	15.7	25.2	0.0	NaN	NaN	SSE
109092	109093	11/11/2010	C17	7.5	20.4	1.6	NaN	NaN	NW
109093	109094	16/04/2012	C11	10.8	29.8	0.0	7.8	11.2	E
109094	109095	09/10/2011	C16	12.3	27.4	9.0	NaN	NaN	WNW

109095 rows × 24 columns

2. Cek *missing value*, apakah ada kategori yang null

Cek Missing Value

```
1 data_kosong = data.isnull()
2 data_kosong
3 #TRUE menandakan bahwa terdapat missing value.
```

	id	Tanggal	KodeLokasi	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	Ara
0	False	False	False	False	False	False	True	True	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	True	True	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
109090	False	False	False	False	False	False	False	False	
109091	False	False	False	False	False	False	True	True	
109092	False	False	False	False	False	False	True	True	
109093	False	False	False	False	False	False	False	False	
109094	False	False	False	False	False	False	True	True	

109095 rows × 24 columns

3. Melihat jumlah *missing value* di setiap kategori

Jumlah Missing Value Tiap Kategori

```
[121]  1 for column in data_kosong.columns.values.tolist():
       2   print(column)
       3   print(data_kosong[column].value_counts())
       4   print("")
```

```
id
False    109095
Name: id, dtype: int64

Tanggal
False    109095
Name: Tanggal, dtype: int64

KodeLokasi
False    109095
Name: KodeLokasi, dtype: int64
```

4. Cek data info

Data Info

```
[122]  1 data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 109095 entries, 0 to 109094
Data columns (total 24 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   id                      109095 non-null  int64
 1   Tanggal                 109095 non-null  object
 2   KodeLokasi              109095 non-null  object
 3   SuhuMin                 107973 non-null  float64
 4   SuhuMax                 108166 non-null  float64
 5   Hujan                   106664 non-null  float64
 6   Penguapan               62071 non-null   float64
 7   SinarMatahari           56716 non-null   float64
 8   ArahAnginTerkencang     101351 non-null  object
 9   KecepatanAnginTerkencang 101399 non-null  float64
 10  ArahAngin9am            101172 non-null  object
 11  ArahAngin3pm            105898 non-null  object
 12  KecepatanAngin9am       107742 non-null  float64
 13  KecepatanAngin3pm       106792 non-null  float64
 14  Kelembaban9am           107093 non-null  float64
 15  Kelembaban3pm           105721 non-null  float64
 16  Tekanan9am              97768 non-null   float64
 17  Tekanan3pm              97787 non-null   float64
 18  Awan9am                 67251 non-null   float64
```

Muhammad Daffa Regenta S - 1301184291
IF-42-07

5. Drop kolom yang tidak terlalu berfungsi

Drop Kolom yang tidak terpakai

```
[123] 1 #salju_train
      2 data.drop('id', axis = 1, inplace = True)
      3 data
```

	Tanggal	KodeLokasi	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	ArahAngin
0	01/06/2014	C4	10.4	15.5	4.8	NaN	NaN	
1	15/07/2014	C10	9.0	17.0	8.0	2.6	7.4	
2	16/02/2011	C46	18.2	32.0	0.0	NaN	NaN	
3	08/08/2012	C36	7.3	24.5	0.0	8.4	10.4	
4	29/10/2016	C7	5.9	20.3	0.0	3.6	12.6	
...	...	...	...	...	...	...	...	
109090	31/01/2009	C38	20.1	23.7	0.0	7.2	8.9	
109091	03/11/2010	C16	15.7	25.2	0.0	NaN	NaN	
109092	11/11/2010	C17	7.5	20.4	1.6	NaN	NaN	
109093	16/04/2012	C11	10.8	29.8	0.0	7.8	11.2	
109094	09/10/2011	C16	12.3	27.4	9.0	NaN	NaN	

109095 rows × 23 columns

6. Isi missing value yang kosong tadi untuk melengkapi data set, dengan mmengisi median dan modus.

Isi missing value

```
1 data.fillna(data.median(), inplace = True)
2 data.fillna(data.mode().iloc[0])
3 data
```

	Tanggal	KodeLokasi	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	A
0	01/06/2014	C4	10.4	15.5	4.8	4.8	8.4	
1	15/07/2014	C10	9.0	17.0	8.0	2.6	7.4	
2	16/02/2011	C46	18.2	32.0	0.0	4.8	8.4	
3	08/08/2012	C36	7.3	24.5	0.0	8.4	10.4	
4	29/10/2016	C7	5.9	20.3	0.0	3.6	12.6	
...	...	...	...	...	...	...	...	
109090	31/01/2009	C38	20.1	23.7	0.0	7.2	8.9	
109091	03/11/2010	C16	15.7	25.2	0.0	4.8	8.4	
109092	11/11/2010	C17	7.5	20.4	1.6	4.8	8.4	
109093	16/04/2012	C11	10.8	29.8	0.0	7.8	11.2	
109094	09/10/2011	C16	12.3	27.4	9.0	4.8	8.4	

109095 rows × 23 columns

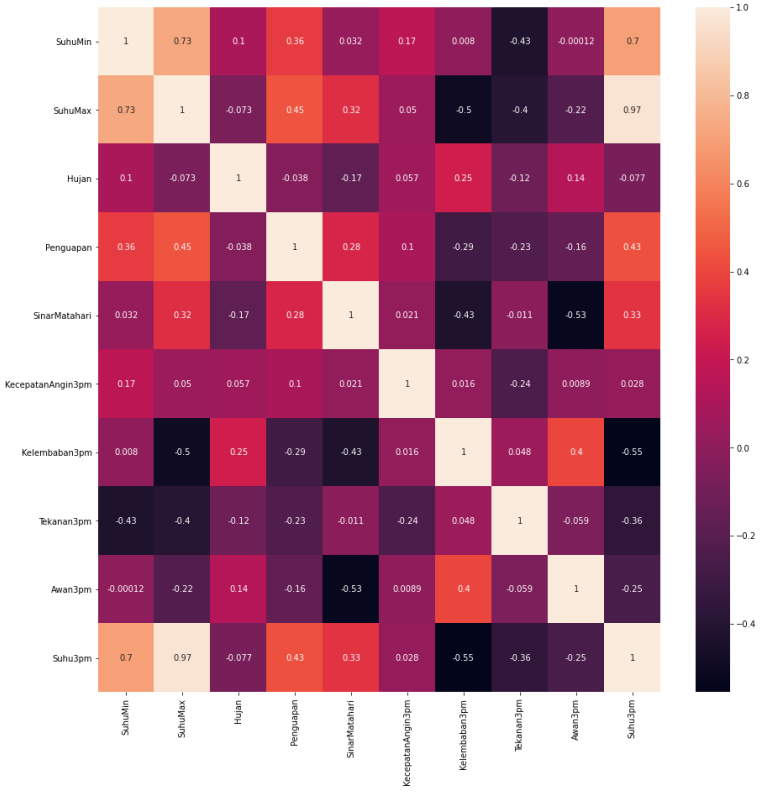
7. L:alu memlih data dengan kategori numerical

Data dengan numerical

```
[126] 1 numdata = pd.DataFrame(columns = ['SuhuMin', 'SuhuMax', 'Hujan', 'Penguapan', 'Sinar
      2 numdata['SuhuMin'] = data['SuhuMin']
      3 numdata['SuhuMax'] = data['SuhuMax']
      4 numdata['Hujan'] = data['Hujan']
      5 numdata['Penguapan'] = data['Penguapan']
      6 numdata['SinarMatahari'] = data['SinarMatahari']
      7 numdata['KecepatanAngin3pm'] = data['KecepatanAngin3pm']
      8 numdata['Kelembaban3pm'] = data['Kelembaban3pm']
      9 numdata['Tekanan3pm'] = data['Tekanan3pm']
     10 numdata['Awan3pm'] = data['Awan3pm']
     11 numdata['Suhu3pm'] = data['Suhu3pm']
     12 numdata
```

Untuk mempermudah proses pemodelan.

C. Pemodelan



Berdasarkan pemodelan korelasi diatas, yang paling tinggi adalah variable suhuMin dan Suhu3pm. Dan akan dilakukan percobaan menggunakan fitur yang kurang korelasi salah satunya suhuMin dan Awan3pm.

Muhammad Daffa Regenta S - 1301184291
IF-42-07

Normalisasi

```
[129]  1 columns_new = numdata.columns
```

```
1 scaler = MinMaxScaler()
2 norm_data = scaler.fit_transform(numdata)
3 norm_data_scalled = pd.DataFrame(norm_data, columns = columns_new)
4
5 norm_data_scalled
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_data.py:375: RuntimeWarning: All-NaN slice encc
  data_min = np.nanmin(X, axis=0)
/usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_data.py:376: RuntimeWarning: All-NaN slice encc
  data_max = np.nanmax(X, axis=0)
```

	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMarahari	KecepatanAngin3pm	Kelembaban3pm	Tekanan3pm
0	0.445755	0.389635	0.012938	0.033103	NaN	0.149425	0.76	0.6624
1	0.412736	0.418426	0.021563	0.017931	NaN	0.229885	0.61	0.6000
2	0.629717	0.706334	0.000000	0.033103	NaN	0.298851	0.42	0.6096
3	0.372642	0.562380	0.000000	0.057931	NaN	0.218391	0.17	0.6368
4	0.339623	0.481766	0.000000	0.024828	NaN	0.218391	0.48	0.6016
...	...	...	...	...	...	...	...	...
109090	0.674528	0.547025	0.000000	0.049655	NaN	0.298851	0.70	0.6480
109091	0.570755	0.575816	0.000000	0.033103	NaN	0.218391	0.52	0.6256
109092	0.377358	0.483685	0.004313	0.033103	NaN	0.149425	0.51	0.5360
109093	0.455189	0.664107	0.000000	0.053793	NaN	0.298851	0.18	0.6192
109094	0.490566	0.618042	0.024259	0.033103	NaN	0.172414	0.76	0.4528

109095 rows × 11 columns

Normalisasi berguna untuk lebih mudah dalam visualisasi dan klusterisasi data.

D. Evaluasi

K-Means

+ Code

```
[131]  1 def euclidean_distance (x1,x2) :
       2     return np.sqrt(np.sum((x1-x2)**2))
```

Menggunakan algoritma Euclidean distance untuk jarak nya.

```
[132]  1 class KMeans_scratch :
       2     def __init__(self, K=2, maks_iteration=150):
       3         self.K = K
       4         self.maks_iteration = maks_iteration
       5         self.centroids = []
       6         self.clusters = [[] for i in range(self.K)]
       7
```

Menggunakan class K-Means dan constructor nya K Maks iteration itu berguna untuk sebera[a banyak iterasi yang dilakukan, dengan default 150.

```
def predict(self, X) :
    self.X = X
    self.baris , self.column = X.shape
```

Untuk menghasilkan label dari hasil klusterisasi.

```
# RANDOM CENTROID
centroid_indexes = np.random.choice(self.baris,self.K,replace=False)
for index in centroid_indexes :
    self.centroids.append(self.X[index])

# ITERASI SEBANYAK MAKS ITERASI ATAU CENTROID TIDAK BERUBAH POSISI
for i in range(self.maks_iteration) :

    # MENENTUKAN CLUSTER ATAU CENTROID TERDEKAT SETIAP TITIK
    clusters = [[] for i in range(self.K)]
    for idx, row in enumerate(self.X) :
        # MENCARI CENTROID TERDEKAT DENGAN MEMBANDINGKAN JARAK CENTROID
        closest_centroids_index = self.select_nearest_centroid(row)
        clusters[closest_centroids_index].append(idx)

    self.clusters = clusters

    # UPDATE CENTROID
    old_centroids = self.centroids
    self.centroids = self.generate_new_centroids()

    # CEK APAKAH CENTROID BERUBAH ATAU TIDAK
    isChange = False
    for i,old_centroid in enumerate(old_centroids) :
        # MENGHITUNG JARAK OLD CENTROID DAN NEW CENTROID
        distance = euclidean_distance(old_centroid, self.centroids[i])
        if (distance != 0) :
            isChange = True
    if (isChange == False) :
        break

# MENENTUKAN LABEL KLASTERISASI
labels = self.generate_labels()
return labels
```

Selanjutnya yaitu generate random centroidnya, setelah generate, lakukan looping sebanyak max iterasi, atau selama centroid tidak berubah posisi nya.
Lalu tentukan cluster terdekat, centroid itu pusat terdekat titiknya. Lalu kita update untuk melhat rata2 dari cluster set.

Kita gunakan label klasterisasi untuk menentukan tititk termasuk klaster dimana.

E. Eksperimen

Eksperimen #1

Muhammad Daffa Regenta S - 1301184291
IF-42-07

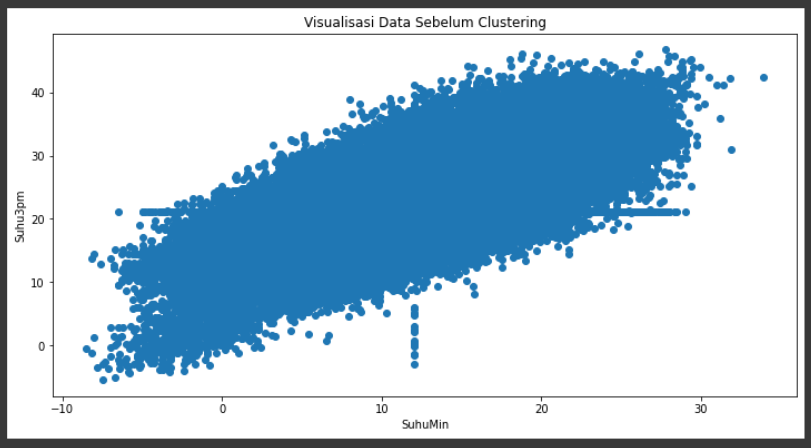
Eksperimen #1

```
1 data1 = data.loc[:, ("SuhuMin", "Suhu3pm")]
2 data1.head()
```

	SuhuMin	Suhu3pm
0	10.4	15.0
1	9.0	15.5
2	18.2	29.6
3	7.3	23.2
4	5.9	18.1

kita gunakan adalah SuhuMin dan Suhu3pm yang korelasi nya baik.

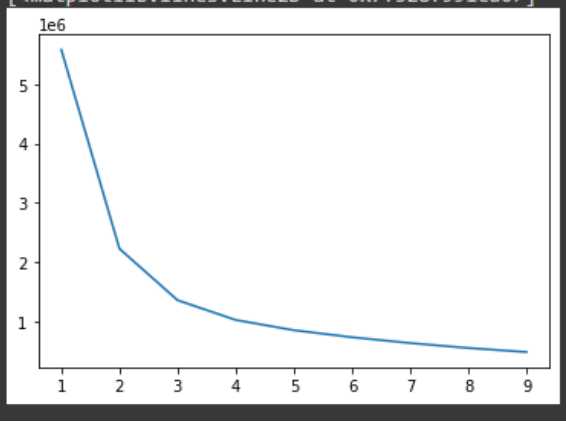
```
1 fig, ax = plt.subplots(figsize=(12,6))
2 ax.scatter(data1.iloc[:,0] , data1.iloc[:,1] )
3 ax.set_xlabel(data1.columns[0])
4 ax.set_ylabel(data1.columns[1])
5 plt.title("Visualisasi Data Sebelum Clustering")
6 plt.show()
```



Muhammad Daffa Regenta S - 1301184291
IF-42-07

```
[158]  1 plt.figure()
       2 plt.plot(k_range,sse)
```

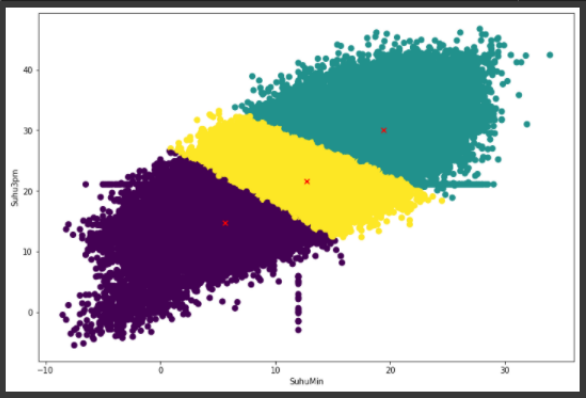
```
[<matplotlib.lines.Line2D at 0x7f328f991cd0>]
```



Clustering

```
[181]  1 model = KMeans_scratch(K=3,maks_iteration=10)
       2 y1_pred = model.predict(X1)
```

```
1 plt.figure(figsize=(12,8))
2 plt.scatter(X1[:, 0], X1[:, 1], c=y1_pred, s=50)
3
4 centroid = model.centroids
5 plt.scatter(centroid[:, 0], centroid[:, 1], c='red',marker="x", linewidths=2)
6 plt.xlabel(data1.columns[0])
7 plt.ylabel(data1.columns[1])
8 plt.show()
```



Hasil klusteriasi..

Muhammad Daffa Regenta S - 1301184291
IF-42-07

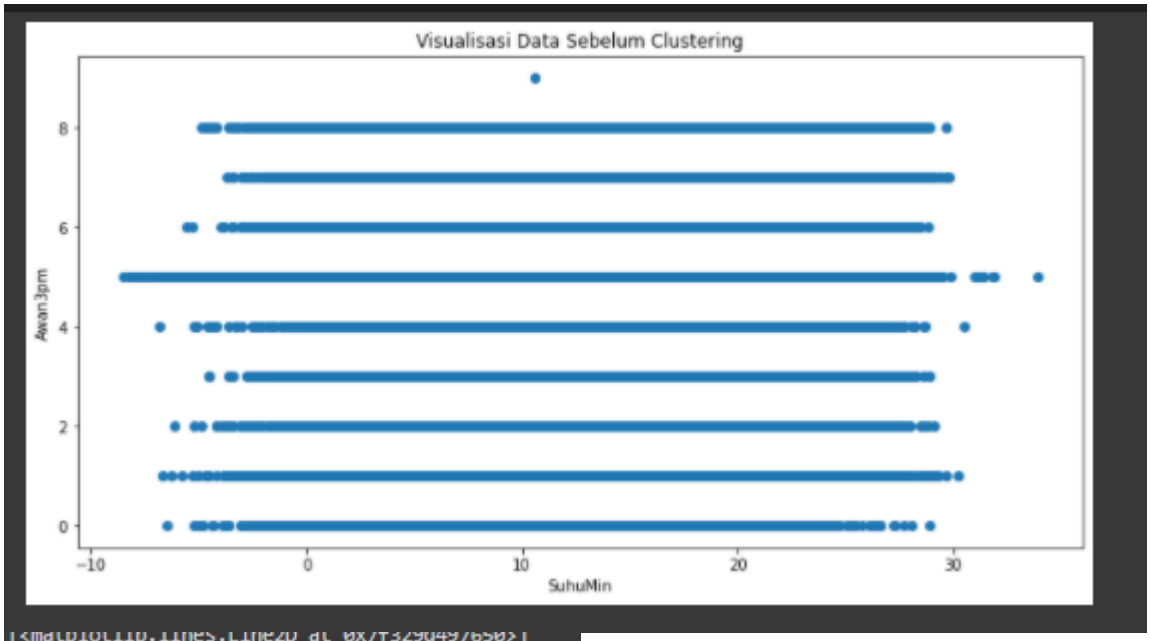
Eksperimen #2

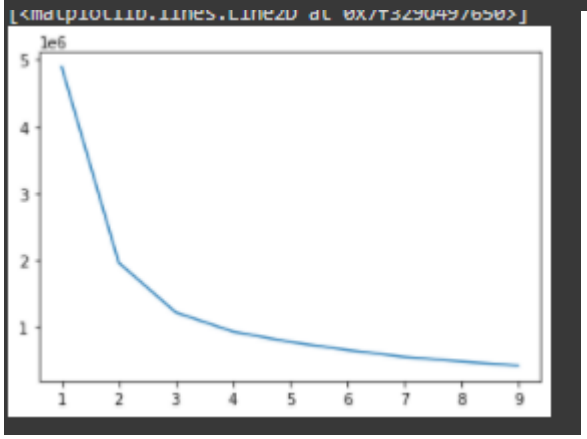
Eksperimen 2

```
[183]  1 data2 = data.loc[:, ("SuhuMin", "Awan3pm")]
       2 data2.head()
```

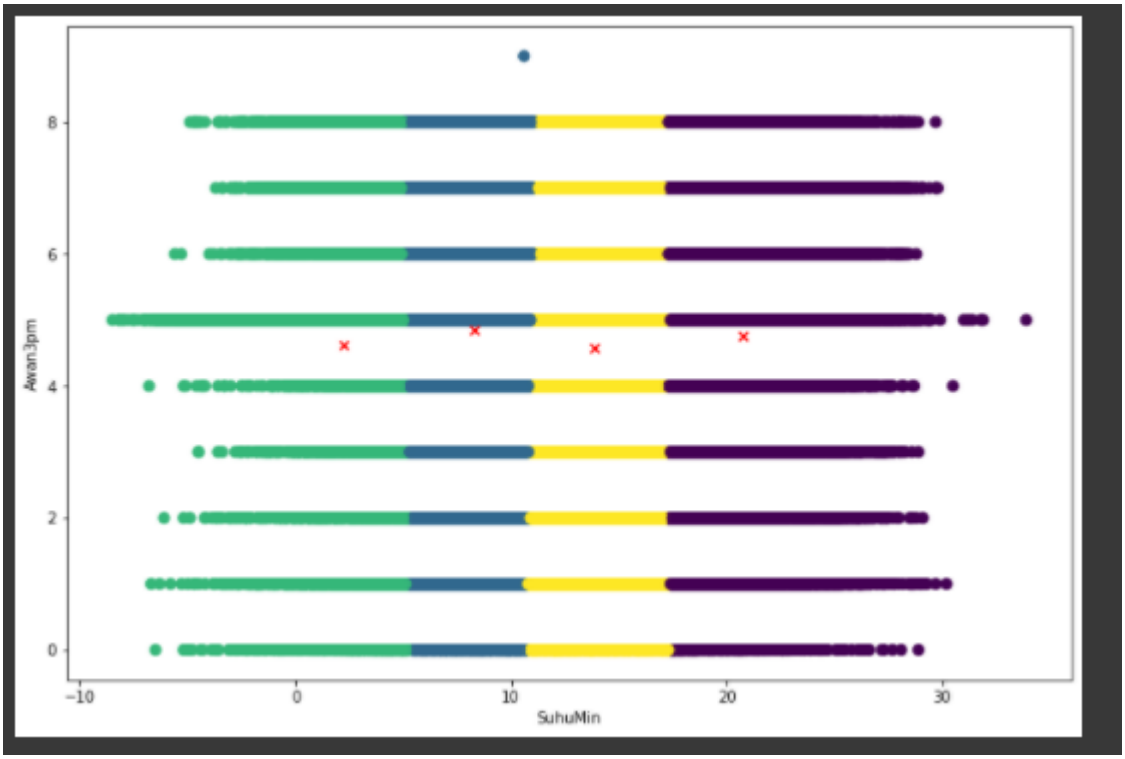
	SuhuMin	Awan3pm
0	10.4	5.0
1	9.0	5.0
2	18.2	5.0
3	7.3	7.0
4	5.9	6.0

kita gunakan adalah SuhuMin dan Awan3pm yang korelasi nya kurang baik.





Muhammad Daffa Regenta S - 1301184291
IF-42-07



Berikut adalah hasil klasterisasi nya.

D. Evaluasi

Evaluasi

```
[171]  1 score_data1 = silhouette_score(X1,y1_pred)
       2 score_data2 = silhouette_score(X2,y2_pred)

[173]  1 print("score eksperimen 1 : " + str(score_data1))
       2 print("score eksperimen 2 : " + str(score_data2))

     score eksperimen 1 : 0.4343302012832219
     score eksperimen 2 : 0.350962768993057
```

Menggunakan Shillouette score.

E. Kesimpulan

Dalam tugas ini, pentingnya dalam preprocessing adalah bahwa ternyata banyak data yang tidak cukup sempurna untuk di testing dengan dataset yang cukup banyak, dan pengolahan clustering, kita dapat menggunakan metode k-means untuk clustering dapat mempermudah karena cukup umum dan mudah untuk di implementasikan.