LAPORAN TUGAS BESAR CLUSTERING MACHINE LEARNING

Alam Rizki Fitriansyah

1301180452

IF – 42 – 10

1. Formulasi Masalah

1.1 Identifikasi Masalah

Melakukan clustering dan memodelkan struktur data pada dataset Salju agar data tersebut dapat dipelajari dan dikelompokan untuk menentukan nilai yang sejenis dan berpola dalam suatu area untuk mendapatkan hasil yang baik

1.2 Identifikasi Sumber Data

Data yang digunakan disini yaitu data Salju dan data yang digunakan adalah data numerik dan data nominal. Agar dapat dilakukan clustering maka data harus di-eksplorasi terlebih dahulu

2. Eksplorasi dan Teknik Persiapan data

Teknik persiapan data yang digunakan :

2.1 Check Empty

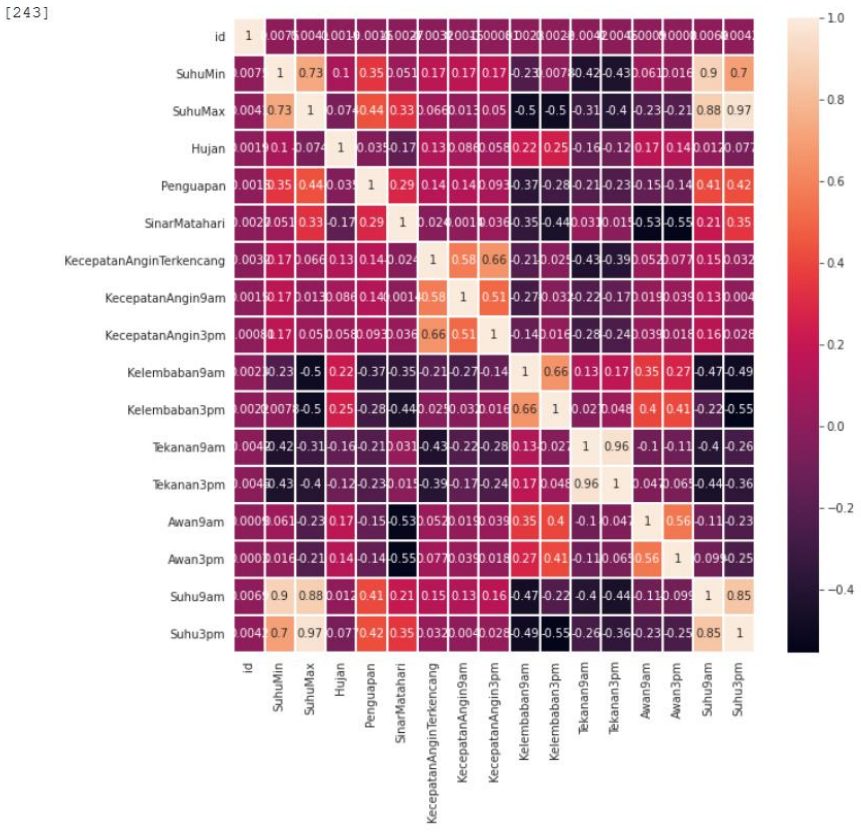
Persiapan yang pertama dilakukan adalah memastikan bahwa tidak ada missing value atau data yang kosong, karena jika banyak data yang kosong maka akan berpengaruh terhadap tingkat akurasi dari suatu data, maka yang harus pertama diakukan adalah Check empty atau Check Missing Value.
Dan ternyata pada data yang digunakan masih banyak data yang kosong sehingga cara yang dilakukan untuk menghilangkannya adalah dengan cara mengisi data yang kosong dengan rata – rata dari setiap kolomnya. Gambar dibawah menunjukan bahwa masih banyak data yang kosong atau tidak memiliki nilai

```
[238] df.isna().sum()
```

```
id                            0
Tanggal                       0
KodeLokasi                    0
SuhuMin                    1122
SuhuMax                     929
Hujan                      2431
Penguapan                 47024
SinarMatahari             52379
ArahAnginTerkencang        7744
KecepatanAnginTerkencang   7696
ArahAngin9am               7923
ArahAngin3pm               3197
KecepatanAngin9am          1353
KecepatanAngin3pm          2303
Kelembaban9am              2002
Kelembaban3pm              3374
Tekanan9am                11327
Tekanan3pm                11308
Awan9am                   41844
Awan3pm                   44471
Suhu9am                    1340
Suhu3pm                    2698
dtype: int64
```

2.2 Correlation

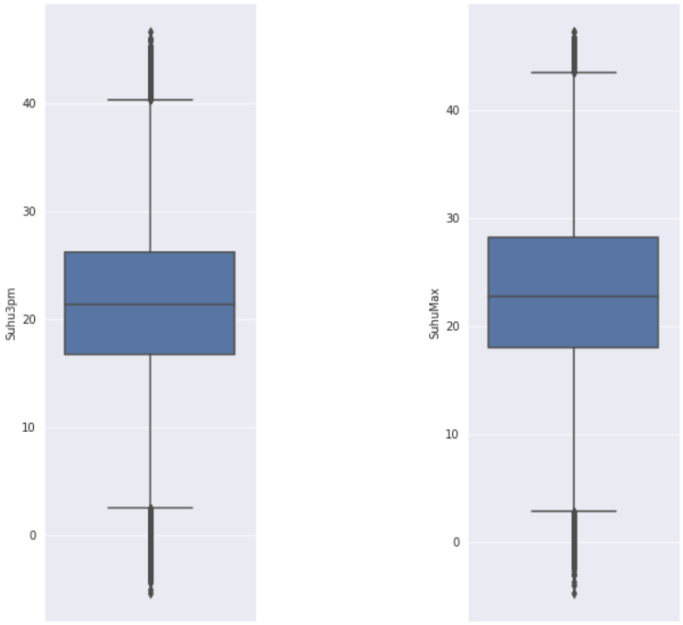
Korelasi disini bertujuan untuk mencari data yang memiliki korelasi yang tinggi dari suatu atribut data untuk melakukan clustering. Saat melakukan check terhadap korelasi kita mencari angka yang paling mendekati angka 1 akan semakin baik.



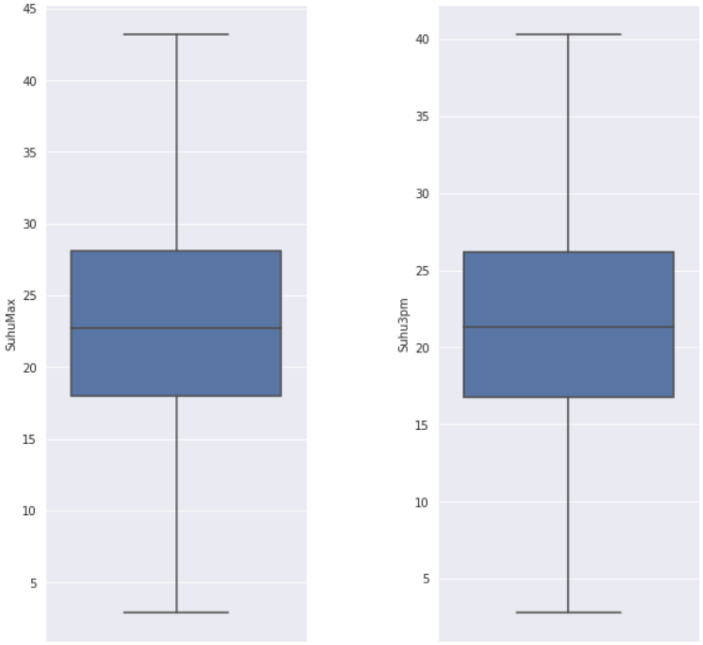
Pada gambar diatas menunjukan bahwa data yang memiliki korelasi yang terbaik yaitu data *Suhu3pm* dan data *SuhuMax* yang menunjukan angka 0.97. Maka data yang dapat dipakai adalah *Suhu3pm* dan *SuhuMax*.

2.3 Outlier Check

Kemudian kita harus melakukan Check terhadapat data tersebut agar data tersebut tidak memiliki Outlier yang dapat mengganggu pengambilan keputusan dan kesimpulan. Pengecekan outlier dapat dilakukan dengan cara membagi nilai dalam quartil 1, quartil 2 dan quartil 3. Hasil pengecekan outlier pertama



Dari gambar diatas menunjukan kedua data tersebut masih memiliki outlier. Dan setelah dilakukan handling data yang diluar outlier sudah di drop



2.4 Scalling

Dilakukan scaling agar nilai dari data yang digunakan rentangnya tidak terlalu jauh atau tidak bervariasi dan random. Scaling yang digunakan disini adalah MinMaxScaler dan membuat data rentang tersebut menjadi 0 -1.

```
[256] n = MinMaxScaler()
      data = n.fit_transform(dtrain.astype(float))
      data
```

```
array([[0.32533333, 0.31265509],
       [0.33866667, 0.34987593],
       [0.71466667, 0.72208437],
       ...,
       [0.45333333, 0.43424318],
       [0.704     , 0.6674938 ],
       [0.54666667, 0.60794045]])
```

3. Pemodelan Clustering

Pemodelan clustering yang yang digunakan adalah k-Means clustering. Data yang ada dimasukan kedalam kelompok yang dilakukan secara acak pada masing – masing fitur data.

3.1 Inisialisasi titik centroid

3.2 Perhitungan Jarak

Pada perhitungan jarak yang digunakan disini adalah menggunakan rumus *Eucledian Distance*

$$\text{dist}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Dimana x adalah data dari tiap fitur dan y adalah centroid dan n adalah jumlah cluster yang ditentukan

```
def jarak(data2,centro): #euclidian
  for i in centro.keys():
    data2['Centroid_{}'.format(i)] = (
        np.sqrt(
            (data2['SuhuMax'] - centro[i][0]) ** 2 + (data2['Suhu3pm'] - centroids[i][1]) ** 2)
    )
  return data2
```