Nama = Akhmad Dian Muzaki

NIM = 1941720205

Kelas = TI-2G

Link = https://github.com/muzakidian/ML-Deteksi-Salju

UAS Machine Learning Deteksi Salju

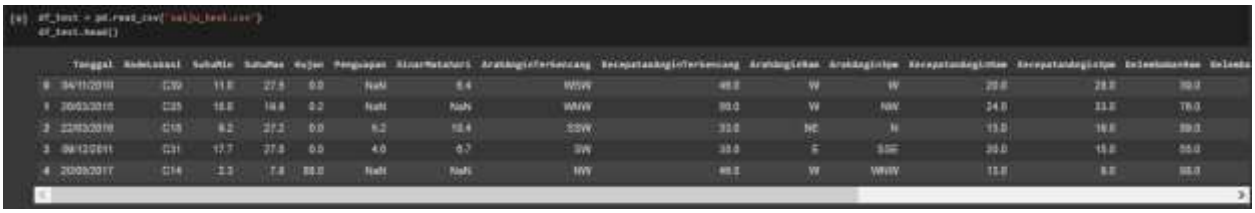
Classification

2021

1. READ DATA

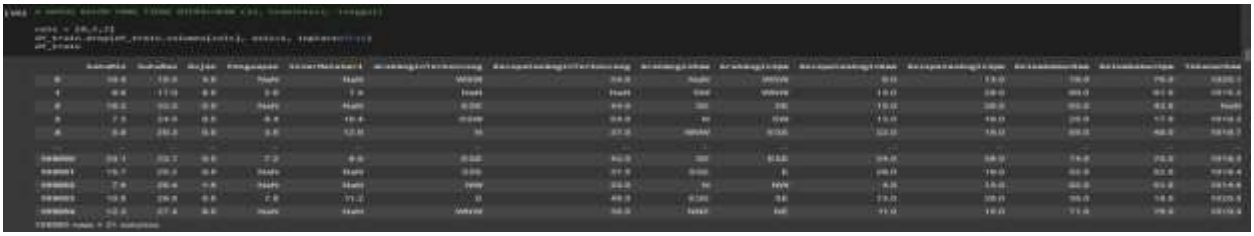
   - Read data salju_train.csv

   - Read data salju_test.csv



2. EXPLORATION DATA

   - Menghapus kolom yang tidak diperlukan pada data train dan data tes



- Mengelompokkan data numerik pada data train dan data test

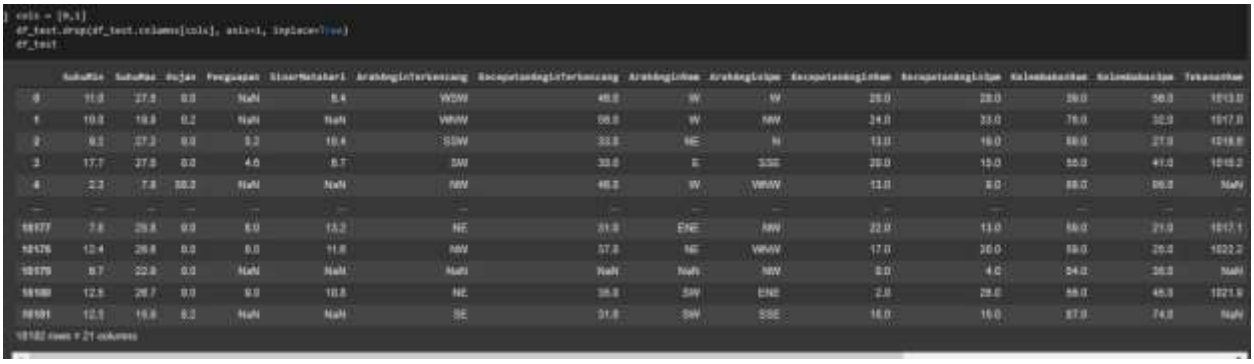
```
[12] df_train._get_numeric_data().columns

     Index(['SuhuMin', 'SuhuMax', 'Hujan', 'Penguapan', 'SinarMatahari',
            'KecepatanAnginTerkencang', 'KecepatanAngin9am', 'KecepatanAngin3pm',
            'Kelembaban9am', 'Kelembaban3pm', 'Tekanan9am', 'Tekanan3pm', 'Awan9am',
            'Awan3pm', 'Suhu9am', 'Suhu3pm'],
           dtype='object')

[13] df_test._get_numeric_data().columns

     Index(['SuhuMin', 'SuhuMax', 'Hujan', 'Penguapan', 'SinarMatahari',
            'KecepatanAnginTerkencang', 'KecepatanAngin9am', 'KecepatanAngin3pm',
            'Kelembaban9am', 'Kelembaban3pm', 'Tekanan9am', 'Tekanan3pm', 'Awan9am',
            'Awan3pm', 'Suhu9am', 'Suhu3pm'],
           dtype='object')
```

- Meng-drop kolom yang tidak diperlukan.



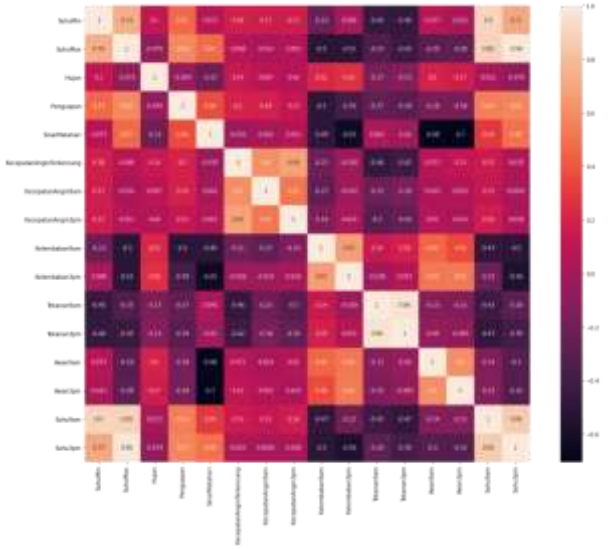
- Menyimpan data train dan data test yang sudah diolah ke variabel baru yaitu train dan test

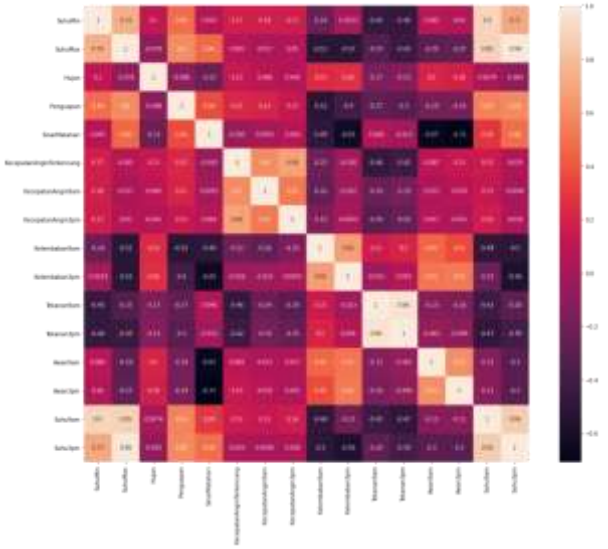
```
[14] train = df_train.copy()
     test = df_test.copy()
```

3. PREPARATION DATA TRAIN & TEST

Dalam menyiapkan data train dan test, dilakukan beberapa hal:

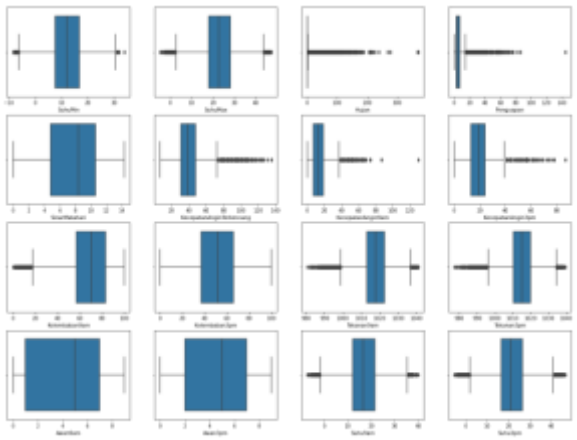
I. Korelasi Matriks
Matriks ini berfungsi untuk menampilkan korelasi setiap fitur.

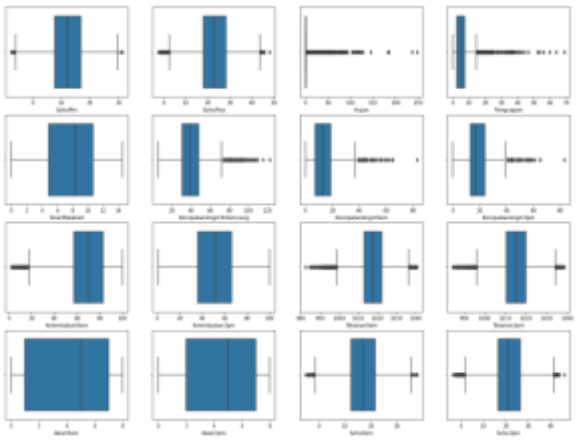




II. Finding Outliers

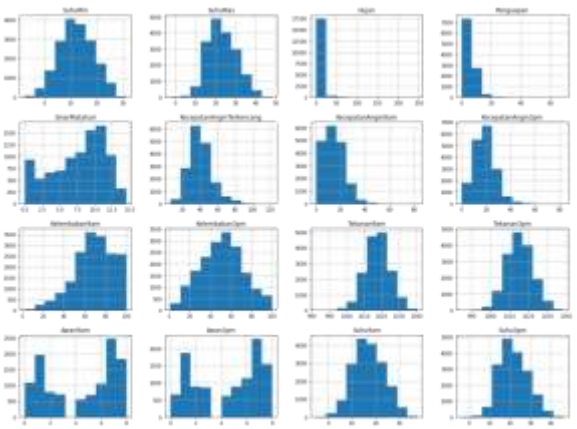
Menemukan outliers pada kedua data untuk dibersihkan nanti.

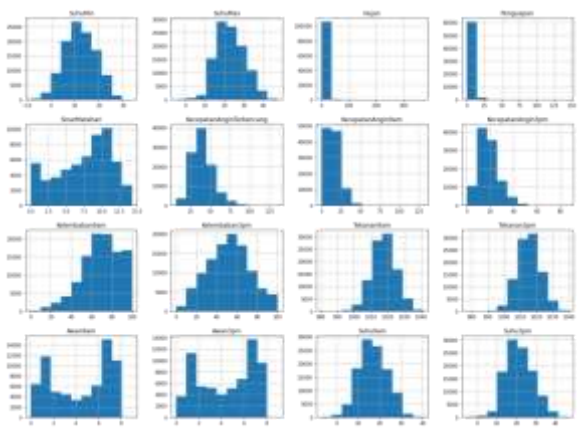




III. Distribution Data

Menampilkan persebaran data setiap fiturnya.





IV. Count Plot
Menampilkan count plot untuk fitur yang tipenya kategorikal.

V. Finding Missing Values
Mencari value yang bernilai NaN / Null untuk di-*impute* nanti.

VI. Dealing Missing Values
Untuk data numerik, value kosong diisi dengan mean dari kolom tersebut.
Untuk data kategorik, value kosong diisi dengan modus dari kolom tersebut.

VII. Dealing Outliers
Dengan library scipy, dilakukan *handle* outliers / pencilan untuk kolom yang memiliki outliers.

VIII. Feature Engineering
A. Binning Atribut "Arah Angin"
Melakukan binning untuk atribut ArahAnginTerkencang, ArahAngin9am, dan ArahAngin3pm menjadi ['W', 'E', 'S', 'N']

B. Encode Categorical Data
Mengubah data kategorikal menjadi data numerik.

C. Scaling
Karena value semua data berada pada range 0~1000, maka dilakukan scalling data untuk mempersempit range menjadi 0~1 saja.

4. CLASSIFICATION

- Hasil akurasi dan confusion matrix menggunakan metode K-Neighbors Classifier

```
[105] from sklearn import metrics

      print("Accuracy Dengan KNeighbors Classifier: ", metrics.accuracy_score(y_test, y_pred))

      Accuracy Dengan KNeighbors Classifier:  0.8410289464432205

[106] from sklearn.metrics import classification_report, confusion_matrix

      print(confusion_matrix(y_test, y_pred))
      print(classification_report(y_test, y_pred))

      [[12980   631]
       [ 2082  1373]]
                    precision    recall  f1-score   support

               0.0       0.86      0.95      0.91     13611
               1.0       0.69      0.40      0.50      3455

          accuracy                           0.84     17066
         macro avg       0.77      0.68      0.70     17066
      weighted avg       0.83      0.84      0.82     17066
```

- Hasil akurasi dan confusion matrix menggunakan metode Decision Tree Classifier

```
[108] print("Accuracy Dengan Decision Tree Classifier: ", metrics.accuracy_score(y_test, y_pred))

      Accuracy Dengan Decision Tree Classifier:  0.7664947849525372

[109] print(confusion_matrix(y_test, y_pred))
      print(classification_report(y_test, y_pred))

      [[11322  2289]
       [ 1696  1759]]
                    precision    recall  f1-score   support

               0.0       0.87      0.83      0.85     13611
               1.0       0.43      0.51      0.47      3455

          accuracy                           0.77     17066
         macro avg       0.65      0.67      0.66     17066
      weighted avg       0.78      0.77      0.77     17066
```

- Hasil akurasi dan confusion matrix menggunakan metode Naive-Bayes Classifier

```
[111] print("Accuracy Dengan Naive-Bayes Classifier: ", metrics.accuracy_score(y_test, y_pred))

      Accuracy Dengan Naive-Bayes Classifier:  0.8023555607640923

[112] print(confusion_matrix(y_test, y_pred))
      print(classification_report(y_test, y_pred))

      [[11558  2053]
       [ 1320  2135]]
                    precision    recall  f1-score   support

               0.0       0.90      0.85      0.87     13611
               1.0       0.51      0.62      0.56      3455

          accuracy                           0.80     17066
         macro avg       0.70      0.73      0.72     17066
      weighted avg       0.82      0.80      0.81     17066
```

- Hasil akurasi dan confusion matrix menggunakan metode Random Forest Classifier

```
[114] print("Accuracy Dengan Random Forest Classifier: ", metrics.accuracy_score(y_test, y_pred))

      Accuracy Dengan Random Forest Classifier:  0.8534513066916677

[115] print(confusion_matrix(y_test, y_pred))
      print(classification_report(y_test, y_pred))

      [[12925   686]
       [ 1815  1640]]
                    precision    recall  f1-score   support

               0.0       0.88      0.95      0.91     13611
               1.0       0.71      0.47      0.57      3455

          accuracy                           0.85     17066
         macro avg       0.79      0.71      0.74     17066
      weighted avg       0.84      0.85      0.84     17066
```

kNeighbors Classifier	**Decision Tree Classifier**	**Naive-Bayes Classifier**	**Random Forest Classifier**
0.8410289464432205	0.7664947849525372	0.8023555607640923	**0.8534513066916677**

Dari percobaan yang telah dilakukan dengan menggunakan data set dan data train diketahui bahwa menggunakan metode Random Forest Classifier menghasilkan hasil akurasi tertinggi dibandingkan dengan metode kNeighbors Classifier, Decision Tree Classifier, Naive-Bayes Classifier yaitu **0.8534513066916677.**