Nama = Haura Athaya Salka, Tri Ayu Syifa'ur Rohmah

NIM = 1301183454, 1301180254

Kelas = IF-42-04

Link =

Tugas Besar MK Pembelajaran Mesin

Classification

2021

1. READ DATA

   - Read data salju_train.csv

```
[2] df_train = pd.read_csv('salju_train.csv')
    df_train.head()
```

	id	Tanggal	KodeLokasi	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	ArahAnginTerkencang	KecepatanAnginTerkencang	ArahAngin9am	ArahAngin3pm	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kel
0	1	01/08/2014	C4	10.4	15.5	4.8	NaN	NaN	WSW	24.0	NaN	WSW	0.0	13.0	78.0	
1	2	15/07/2014	C10	9.0	17.0	8.0	2.6	7.4	NaN	NaN	SW	WNW	13.0	20.0	80.0	
2	3	16/02/2011	C46	18.2	32.0	0.0	NaN	NaN	ESE	44.0	SE	SE	15.0	26.0	62.0	
3	4	08/08/2012	C38	7.3	24.5	0.0	8.4	10.4	SSW	54.0	N	SW	13.0	19.0	25.0	
4	5	29/10/2016	C7	5.9	20.3	0.0	3.6	12.6	N	37.0	NNW	ESE	22.0	19.0	55.0	

   - Read data salju_test.csv

```
[6] df_test = pd.read_csv('salju_test.csv')
    df_test.head()
```

	Tanggal	KodeLokasi	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	ArahAnginTerkencang	KecepatanAnginTerkencang	ArahAngin9am	ArahAngin3pm	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelemba
0	04/11/2010	C39	11.0	27.5	0.0	NaN	6.4	WSW	46.0	W	W	20.0	28.0	39.0	
1	26/03/2015	C35	10.0	19.9	0.2	NaN	NaN	WNW	56.0	W	NW	24.0	33.0	76.0	
2	22/03/2016	C18	9.2	27.2	0.0	5.2	10.4	SSW	33.0	NE	N	13.0	19.0	89.0	
3	09/12/2011	C31	17.7	27.0	0.0	4.6	6.7	SW	35.0	E	SSE	20.0	15.0	55.0	
4	20/05/2017	C14	2.3	7.9	88.0	NaN	NaN	NW	46.0	W	WNW	13.0	9.0	98.0	

2. EXPLORATION DATA

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

```
[10] # HAPUS KOLOM YANG TIDAK DIPERLUKAN (id, kodelokasi, tanggal)
     cols = [0,1,2]
     df_train.drop(df_train.columns[cols], axis=1, inplace=True)
     df_train
```

	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	ArahAnginTerkencang	KecepatanAnginTerkencang	ArahAngin9am	ArahAngin3pm	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelembaban3pm	Tekanan9am
0	10.4	15.5	4.8	NaN	NaN	WSW	24.0	NaN	WSW	0.0	13.0	78.0	70.0	1020.1
1	9.0	17.0	8.0	2.6	7.4	NaN	NaN	SW	WNW	13.0	20.0	80.0	61.0	1015.2
2	18.2	32.0	0.0	NaN	NaN	ESE	44.0	SE	SE	15.0	26.0	62.0	42.0	NaN
3	7.3	24.5	0.0	8.4	10.4	SSW	54.0	N	SW	13.0	19.0	25.0	17.0	1019.2
4	5.9	20.3	0.0	3.6	12.6	N	37.0	NNW	ESE	22.0	19.0	55.0	48.0	1019.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
109090	20.1	23.7	0.0	7.2	8.9	ESE	43.0	SE	ESE	24.0	26.0	74.0	70.0	1019.3
109091	15.7	25.2	0.0	NaN	NaN	SSE	37.0	SSE	E	28.0	19.0	52.0	52.0	1018.4
109092	7.5	20.4	1.6	NaN	NaN	NW	33.0	N	NW	4.0	13.0	92.0	51.0	1014.5
109093	10.8	29.8	0.0	7.8	11.2	E	48.0	ESE	SE	13.0	26.0	35.0	18.0	1020.0
109094	12.3	27.4	9.0	NaN	NaN	WNW	35.0	NNE	NE	11.0	15.0	71.0	78.0	1010.4

109095 rows × 21 columns

- 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.

```
cols = [0,1]
df_test.drop(df_test.columns[cols], axis=1, inplace=True)
df_test
```

	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	ArahAnginTerkencang	KecepatanAnginTerkencang	ArahAngin9am	ArahAngin3pm	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelembaban3pm	Tekanan9am
0	11.0	27.5	0.0	NaN	6.4	WSW	46.0	W	W	20.0	28.0	39.0	56.0	1013.0
1	10.0	19.9	0.2	NaN	NaN	WNW	56.0	W	NW	24.0	33.0	76.0	32.0	1017.0
2	9.2	27.2	0.0	5.2	10.4	SSW	33.0	NE	N	13.0	19.0	89.0	27.0	1018.6
3	17.7	27.0	0.0	4.6	6.7	SW	35.0	E	SSE	20.0	15.0	55.0	41.0	1010.2
4	2.3	7.9	88.0	NaN	NaN	NW	46.0	W	WNW	13.0	9.0	98.0	95.0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18177	7.8	25.8	0.0	8.0	13.2	NE	31.0	ENE	NW	22.0	13.0	59.0	21.0	1017.1
18178	12.4	26.6	0.0	8.0	11.6	NW	37.0	NE	WNW	17.0	20.0	59.0	25.0	1022.2
18179	6.7	22.9	0.0	NaN	NaN	NaN	NaN	NaN	NW	0.0	4.0	84.0	35.0	NaN
18180	12.5	26.7	0.0	9.0	10.8	NE	35.0	SW	ENE	2.0	26.0	56.0	45.0	1021.9
18181	12.3	15.9	8.2	NaN	NaN	SE	31.0	SW	SSE	15.0	15.0	87.0	74.0	NaN

18182 rows × 21 columns

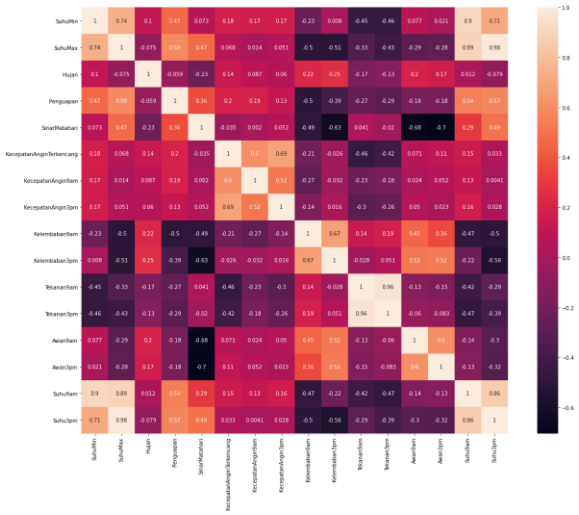
- Menyimpan data train dan data test yang sudah diolah ke variabel baru yaitu train dan test

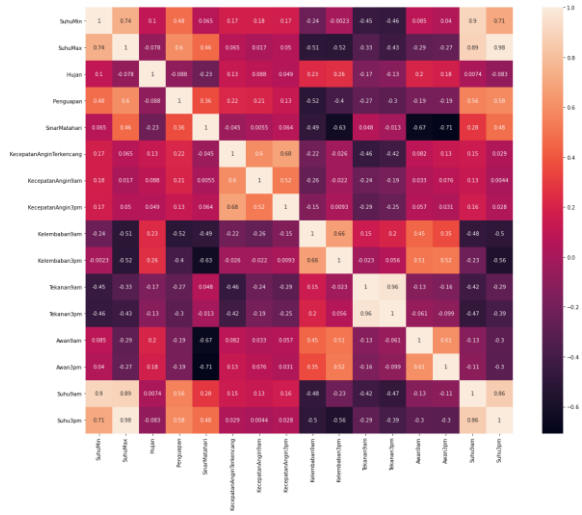
```
[14] train = df_train.copy()
     test = df_test.copy()
```

3. PREPARATION DATA TRAIN & TEST

Dalam menyiapkan data train dan test, dilakukan beberapa hal:

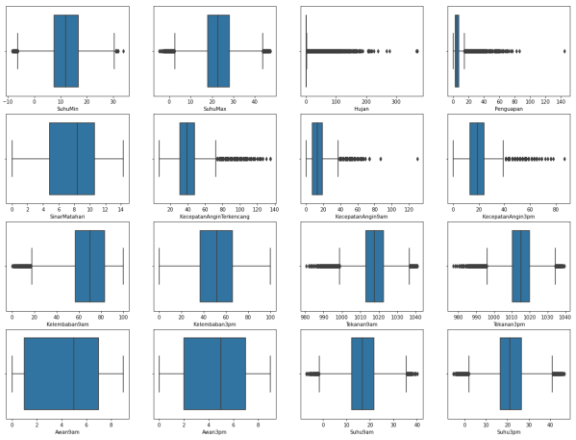
I. Korelasi Matriks
Matriks ini berfungsi untuk menampilkan korelasi setiap fitur.

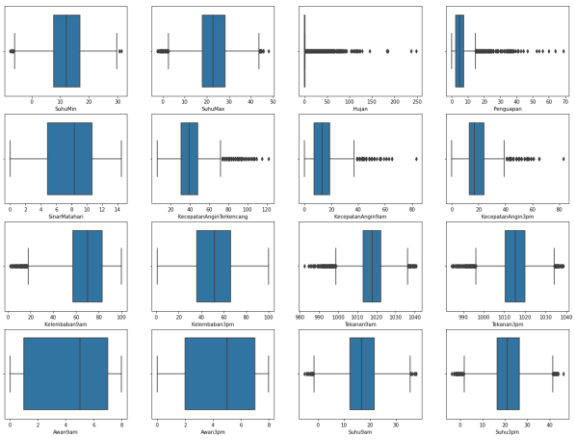




II. Finding Outliers

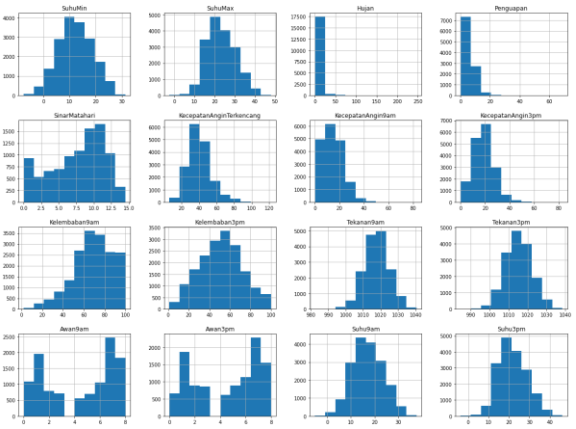
Menemukan outliers pada kedua data untuk dibersihkan nanti.

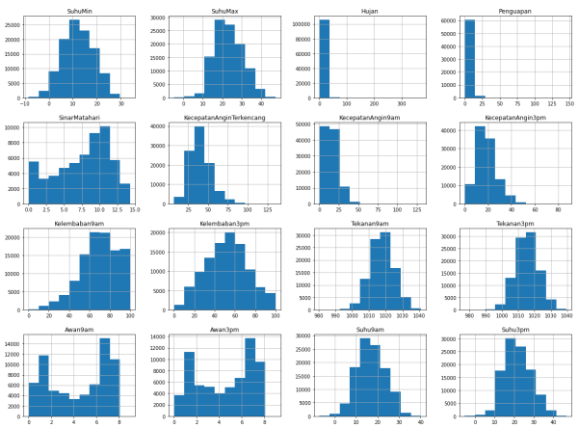




III. Distribution Data

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.**