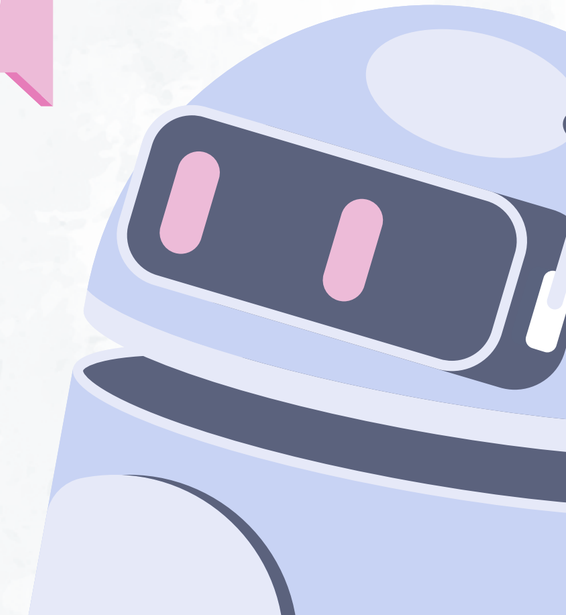
Tugas Akhir
Algoritma & Struktur Data

Dataset : Hujan
Kelompok 5



Anggota Kelompok:

Putu Agus Arya Suta
2208541012

I Dewa Ayu Pradnya Pratiwi T.
2208541013

Ni Putu Dina Agustina
2208541043

1. Eksplorasi Data Hujan

Dataset Hujan merupakan suatu data set yang berisi kejadian seperti curah hujan, kelembapan, kecepatan angin, temperature, dan sebagainya pada suatu wilayah dalam waktu tertentu. Data ini digunakan untuk memprediksi apakah hari ini akan terjadi hujan atau tidak.

(a) Mengimport Library

Sebelum masuk ke eksplorasi, data akan diimpor terlebih dahulu. Library yang akan digunakan, seperti ; pandas, warnings, matplotlib, seaborn, dsb.

(b) Menampilkan Data, Tipe Data, dan Data Kosong

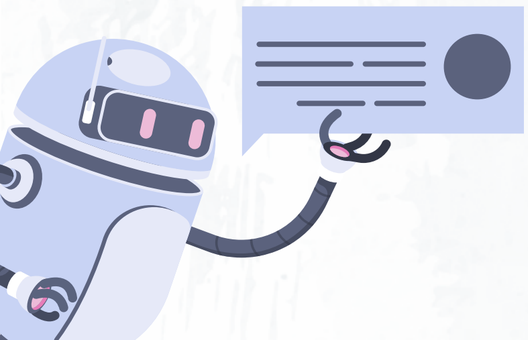
Akan dimunculkan data set dengan menggunakan link pada github dalam bentuk csv.
Selanjutnya ditampilkan tipe data dari setiap kolomnya, serta data kosong dari setiap kolomnya.

Berikut adalah output dari dataset hujan

Ukuran data : 145.460 x 23

MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm
13.4	22.9	0.6	NaN	NaN	W	44.0	W	WNW
7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	WSW
12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	WSW
9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	E
17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	NW
...	...	...	...	...	...	...	...	...
2.8	23.4	0.0	NaN	NaN	E	31.0	SE	ENE
3.6	25.3	0.0	NaN	NaN	NNW	22.0	SE	N
5.4	26.9	0.0	NaN	NaN	N	37.0	SE	WNW
7.8	27.0	0.0	NaN	NaN	SE	28.0	SSE	N
14.9	NaN	0.0	NaN	NaN	NaN	NaN	ESE	ESE

;



```
Date              object
Location          object
MinTemp          float64
MaxTemp          float64
Rainfall         float64
Evaporation      float64
Sunshine         float64
WindGustDir       object
WindGustSpeed    float64
WindDir9am        object
WindDir3pm        object
WindSpeed9am     float64
WindSpeed3pm     float64
Humidity9am      float64
Humidity3pm      float64
Pressure9am      float64
Pressure3pm      float64
Cloud9am         float64
Cloud3pm         float64
Temp9am          float64
Temp3pm          float64
RainToday         object
RainTomorrow      object
dtype: object
```

Seluruh kolom pada data dengan tipe datanya masing-masing

```
Date                0
Location            0
MinTemp          1485
MaxTemp          1261
Rainfall         3261
Evaporation     62790
Sunshine        69835
WindGustDir     10326
WindGustSpeed   10263
WindDir9am      10566
WindDir3pm       4228
WindSpeed9am     1767
WindSpeed3pm     3062
Humidity9am      2654
Humidity3pm      4507
Pressure9am     15065
Pressure3pm     15028
Cloud9am        55888
Cloud3pm        59358
Temp9am          1767
Temp3pm          3609
RainToday        3261
RainTomorrow     3267
dtype: int64
```

Jumlah data kosong pada masing-masing kolom

2. Mengatasi Data Kosong

Karena terdapat data kosong yang cukup banyak, maka akan dilakukan beberapa metode untuk mengatasi data yang kosong tersebut.

(a) Drop data yang tidak terlalu berpengaruh

Pada eksplorasi data sebelumnya, terdapat 5 kolom dengan tipe data object dan tidak kami gunakan yaitu : Date, Location, WindGustDir, WindDir9am, WindDir3pm.

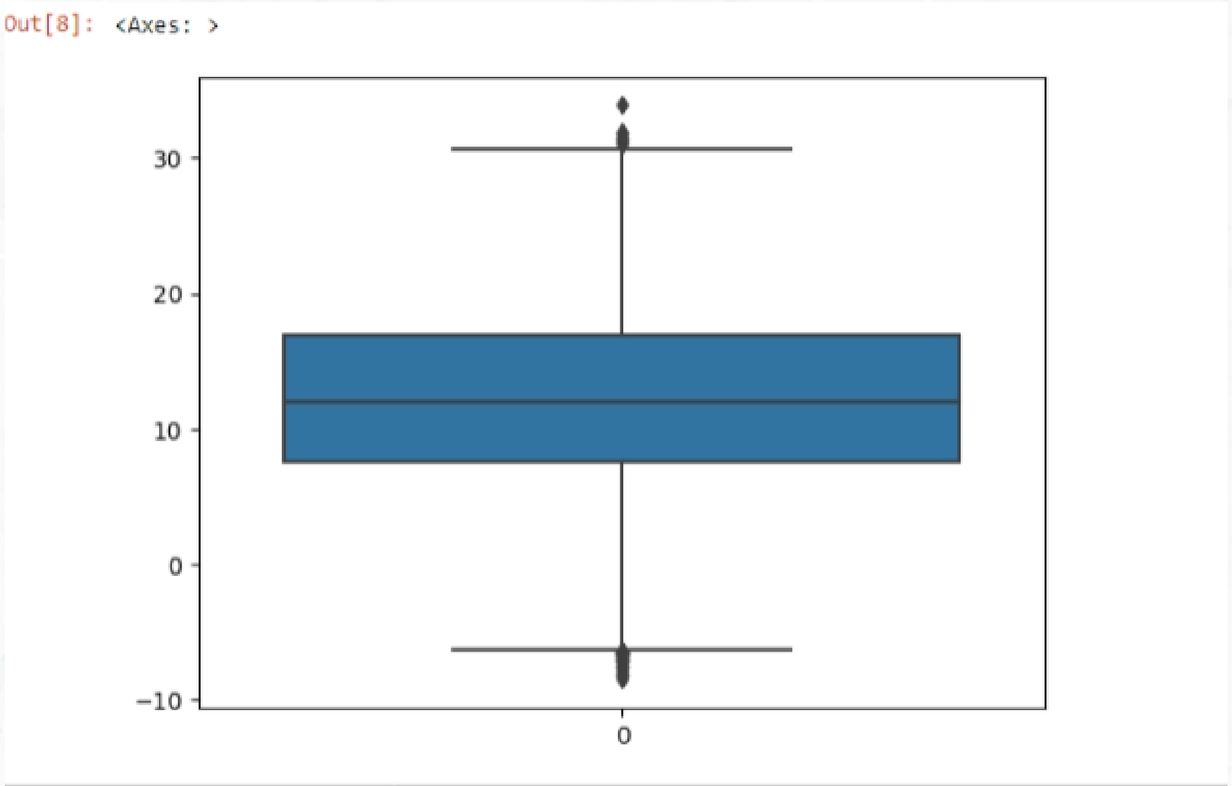
```
#menggunakan metode drop pada kolom dengan tipe data object
data_hujan = data_hujan.drop(['Date', 'Location', 'WindGustDir', 'WindDir9am','WindDir3pm'], axis=1)
data_hujan
```

(b) Mengisi data menggunakan Mean dan Median dengan memeriksa outlier

Tujuan dari pemeriksaan outlier atau pencilan adalah untuk mengetahui langkah selanjutnya dalam mengisi data yang kosong, baik dengan menggunakan mean ataupun median. Berikut kode yang digunakan untuk mengecek outlier pada salah satu kolom.

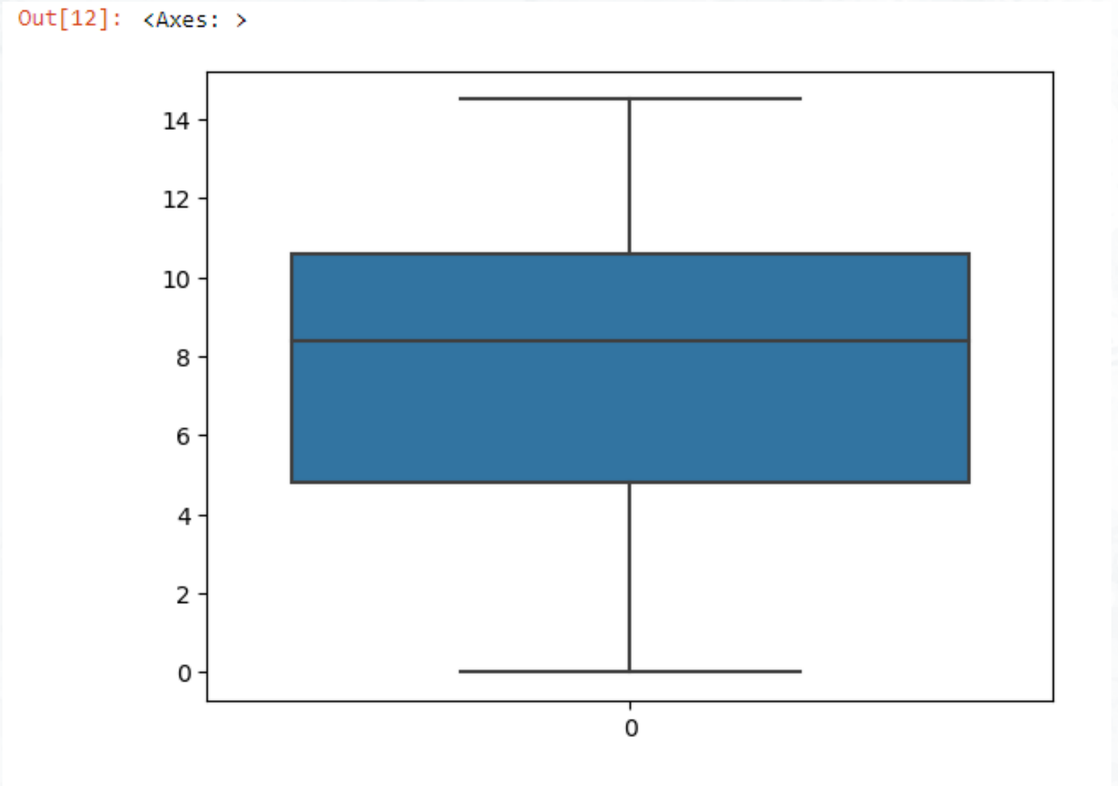
```
#Menggunakan Boxplot untuk mengetahui posisi Outlier pada 'MinTemp'
import seaborn as sns
sns.boxplot(data_hujan['MinTemp'])
```

Out[8]: <Axes: >



Gambar di atas adalah boxplot dari kolom MinTemp dan terlihat bahwa terdapat banyak outlier atau pencilan. Sehingga untuk mengisi data kosong pada MinTemp digunakan nilai median dari MinTemp karena median tidak terpengaruh oleh adanya outlier atau pencilan. Begitu pun untuk kolom yang lain berlaku sama apabila terdapat outlier

Out[12]: <Axes: >



Gambar di atas adalah boxplot dari kolom Sunshine dan terlihat bahwa tidak terdapat outlier atau pencilan. Sehingga untuk mengisi data kosong pada Sunshine digunakan nilai mean dari Sunshine itu sendiri. Begitu pun untuk kolom lain berlaku sama apabila tidak terdapat outlier

Dari hasil pemeriksaan outlier, didapat dari 23 kolom, hanya 4 kolom saja yang tidak memiliki outlier yakni ; Sunshine, Humidity3pm, Cloud9am, Cloud3pm, Sehingga keempat kolom tersebut akan diisi dengan mean dan kolom yang lain akan diisi dengan median.

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3p
count	143975.000000	144199.000000	142199.000000	82670.000000	75625.000000	135197.000000	143693.000000	142398.000000	142806.000000	140953.0000
mean	12.194034	23.221348	2.360918	5.468232	7.611178	40.035230	14.043426	18.662657	68.880831	51.5391
std	6.398495	7.119049	8.478060	4.193704	3.785483	13.607062	8.915375	8.809800	19.029164	20.7959
min	-8.500000	-4.800000	0.000000	0.000000	0.000000	6.000000	0.000000	0.000000	0.000000	0.0000
25%	7.600000	17.900000	0.000000	2.600000	4.800000	31.000000	7.000000	13.000000	57.000000	37.0000
50%	12.000000	22.600000	0.000000	4.800000	8.400000	39.000000	13.000000	19.000000	70.000000	52.0000
75%	16.900000	28.200000	0.800000	7.400000	10.600000	48.000000	19.000000	24.000000	83.000000	66.0000
max	33.900000	48.100000	371.000000	145.000000	14.500000	135.000000	130.000000	87.000000	100.000000	100.0000

Gambar disamping merupakan statistika deskriptif data hujan untuk mengetahui nilai mean dan median tiap-tiap kolom

Gambar disamping merupakan data hujan terbaru setelah diisi dengan mean dan median

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	P
0	13.4	22.9	0.6	4.8	7.611178	44.0	20.0	24.0	71.0	22.0	1007.7	
1	7.4	25.1	0.0	4.8	7.611178	44.0	4.0	22.0	44.0	25.0	1010.6	
2	12.9	25.7	0.0	4.8	7.611178	46.0	19.0	26.0	38.0	30.0	1007.6	
3	9.2	28.0	0.0	4.8	7.611178	24.0	11.0	9.0	45.0	16.0	1017.6	
4	17.5	32.3	1.0	4.8	7.611178	41.0	7.0	20.0	82.0	33.0	1010.8	
...	...	...	...	...	...	...	...	...	...	...	...	...
145455	2.8	23.4	0.0	4.8	7.611178	31.0	13.0	11.0	51.0	24.0	1024.6	
145456	3.6	25.3	0.0	4.8	7.611178	22.0	13.0	9.0	56.0	21.0	1023.5	
145457	5.4	26.9	0.0	4.8	7.611178	37.0	9.0	9.0	53.0	24.0	1021.0	
145458	7.8	27.0	0.0	4.8	7.611178	28.0	13.0	7.0	51.0	24.0	1019.4	
145459	14.9	22.6	0.0	4.8	7.611178	39.0	17.0	17.0	62.0	36.0	1020.2	

145460 rows × 18 columns

Setelah dilakukan pengisian data kosong dengan nilai mean atau median, kembali dilakukan pemeriksaan terkait data kosong pada data.

```
data_hujan.isnull().sum()
```

```
MinTemp            0
MaxTemp            0
Rainfall           0
Evaporation        0
Sunshine           0
WindGustSpeed      0
WindSpeed9am       0
WindSpeed3pm       0
Humidity9am        0
Humidity3pm        0
Pressure9am        0
Pressure3pm        0
Cloud9am           0
Cloud3pm           0
Temp9am            0
Temp3pm            0
RainToday       3261
RainTomorrow    3267
dtype: int64
```

```
data_hujan = data_hujan.dropna()
```

Terlihat bahwa terdapat 2 kolom dengan tipe object yang masih berisi data kosong, sehingga dilakukan metode dropna dan tidak ada lagi data yang kosong.

```
data_hujan.isnull().sum()
```

```
MinTemp         0
MaxTemp         0
Rainfall        0
Evaporation     0
Sunshine        0
WindGustSpeed   0
WindSpeed9am    0
WindSpeed3pm    0
Humidity9am     0
Humidity3pm     0
Pressure9am     0
Pressure3pm     0
Cloud9am        0
Cloud3pm        0
Temp9am         0
Temp3pm         0
RainToday       0
RainTomorrow    0
dtype: int64
```

3. One Hot Encode Data

Proses one hot encode data merupakan metode untuk mengubah variabel kategorikal menjadi variabel numerik. Hal ini bertujuan untuk memudahkan proses analisis data ketika seluruh data berupa numerik seperti data float ataupun integer.

```
In [32]: data_hujan['RainTomorrow'].unique()
Out[32]: array(['No', 'Yes'], dtype=object)

In [33]: data_hujan['RainTomorrow'] = data_hujan['RainTomorrow'].map({'No':0, 'Yes':1})

In [34]: data_hujan = pd.get_dummies(data_hujan)

In [35]: data_hujan
```

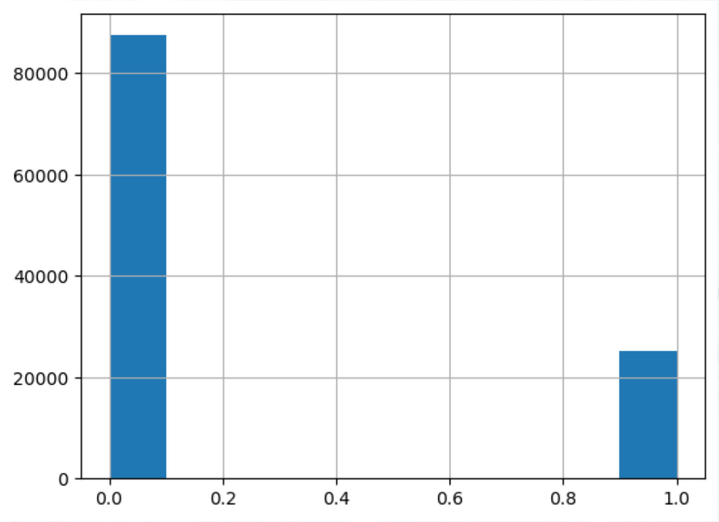
Out[35]:

ıdSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainTomorrow	RainToday_No	RainToday_Yes
24.0	71.0	22.0	1007.7	1007.1	8.000000	4.50993	16.9	21.8	0	1	0
22.0	44.0	25.0	1010.6	1007.8	4.447461	4.50993	17.2	24.3	0	1	0
26.0	38.0	30.0	1007.6	1008.7	4.447461	2.00000	21.0	23.2	0	1	0
9.0	45.0	16.0	1017.6	1012.8	4.447461	4.50993	18.1	26.5	0	1	0
20.0	82.0	33.0	1010.8	1006.0	7.000000	8.00000	17.8	29.7	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
13.0	59.0	27.0	1024.7	1021.2	4.447461	4.50993	9.4	20.9	0	1	0
11.0	51.0	24.0	1024.6	1020.3	4.447461	4.50993	10.1	22.4	0	1	0
9.0	56.0	21.0	1023.5	1019.1	4.447461	4.50993	10.9	24.5	0	1	0
9.0	53.0	24.0	1021.0	1016.8	4.447461	4.50993	12.5	26.1	0	1	0
7.0	51.0	24.0	1019.4	1016.5	3.000000	2.00000	15.1	26.0	0	1	0

4. Menyeimbangkan Data

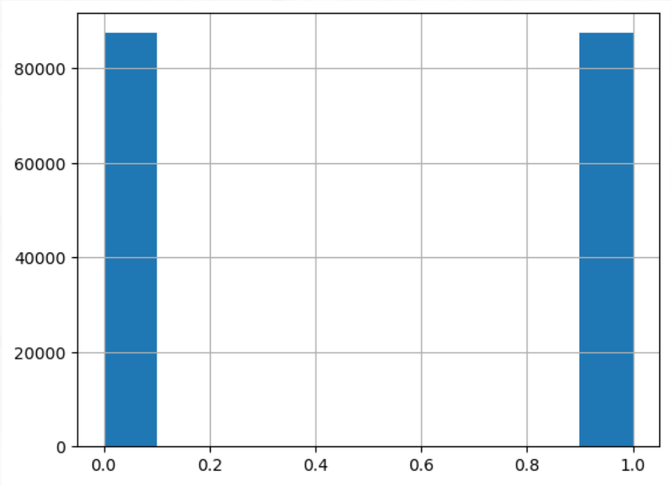
- Mendefinisikan variabel x dan y (train dan test)

```
In [40]: X = data_hujan.drop(['RainToday_Yes'],axis=1)
         y = data_hujan['RainToday_Yes']

In [41]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
```



- Menggunakan OverSampler

```
In [44]: from imblearn.over_sampling import RandomOverSampler

In [45]: X_os,y_os = RandomOverSampler().fit_resample(X_train,y_train)
```



- Menampilkan Model

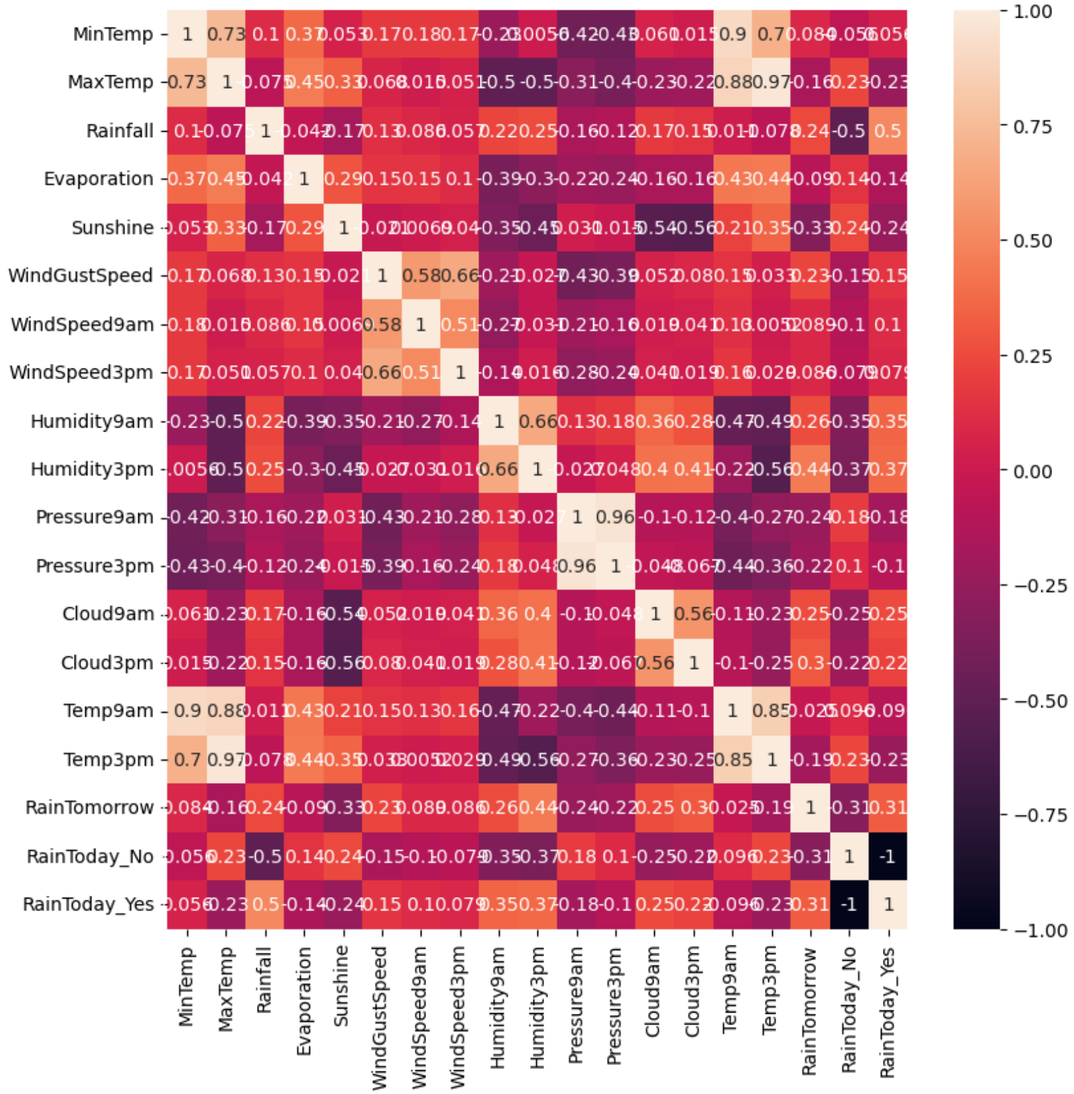
```
In [47]: from sklearn.linear_model import LogisticRegression

In [48]: model_lr = LogisticRegression()

In [49]: model_lr.fit(X_os,y_os)

Out[49]: ▾ LogisticRegression
         LogisticRegression()
```

- Menampilkan Akurasi Model

```
In [50]: #ini merupakan akurasi skor x dan y test menggunakan metode oversampling
         model_lr.score(X_test,y_test)

Out[50]: 0.9995738333688472
```

5. Korelasi

- Menampilkan Korelasi Data



Berdasarkan korelasi pada gambar di samping maka kami membagi data hujan menjadi beberapa bagian agar dapat terlihat dengan jelas korelasi antar kolomnya. Setelah itu diperoleh korelasi terbesar dibentuk oleh MaxTemp dan Temp3Pm dengan nilai 0.969864

```
In [64]: data_hujan6 = data_hujan[['MaxTemp', 'Temp3pm', 'RainToday_No', 'RainToday_Yes']]
         data_hujan6.corr()
```

Out[64]:

	MaxTemp	Temp3pm	RainToday_No	RainToday_Yes
MaxTemp	1.000000	0.969864	0.228636	-0.228636
Temp3pm	0.969864	1.000000	0.232598	-0.232598
RainToday_No	0.228636	0.232598	1.000000	-1.000000
RainToday_Yes	-0.228636	-0.232598	-1.000000	1.000000

Setelah itu, kami akan menggunakan nilai rata-rata dari MaxTemp dan Temp3Pm untuk menentukan kemungkinan Hujan atau tidak

6.Hasil Analisis

6.1. Menentukan Nilai Ambang Batas

1. Pertama kita menampilkan statistik deskriptif dari data hujan dan diperoleh nilai rata-rata dari MaxTemp adalah 23.233735 dan nilai rata-rata dari Temp3Pm adalah 21.682127
2. Kita dapat menentukan ambang batas dengan menggunakan nilai rata-ratanya masing-masing dan diperoleh :
    - threshold_temp3pm = 21.68 # Nilai ambang batas temperatur maksimum untuk menentukan ada hujan atau tidak
    - threshold_maxtemp = 23.24 # Nilai ambang batas kecepatan angin untuk menentukan ada hujan atau tidak
3. Berdasarkan data yang diberikan dapat dilihat bahwa nilai di bawah MaxTemp dan Temp3Pm dapat menghasilkan Hujan sehingga dapat ditetapkan bahwa nilai di bawah ambang batas tersebut dapat berpotensi terjadinya Hujan.
4. Kemudian dapat disusun program untuk memasukkan besaran MaxTemp dan Temp3Pm untuk mengetahui adanya potensi Hujan hari ini atau tidak

6.2. Menampilkan Kode dan Hasil dari Analisis Data Hujan

[73]:
```python
def predict_rain(suhu, temp3pm) :
    # Menentukan apakah ada hujan berdasarkan Max Temperature dan Kecepatan Angin
    threshold_maxtemp = 23.24 # Nilai ambang batas temperatur maksimum untuk menentukan ada hujan atau tidak
    threshold_temp3pm = 21.68  # Nilai ambang batas temperatur pukul 3 sore untuk menentukan ada hujan atau tidak

    if suhu < threshold_maxtemp and temp3pm < threshold_temp3pm:
        return "Hujan"
    else:
        return "Tidak Hujan"

# Input suhu dan temperatur pukul 3 sore dari pengguna
input_maxtemp = float(input("Masukkan suhu maksimal : "))
input_temp3pm = float(input("Masukkan temperatur pukul 15.00 : "))

# Memprediksi apakah hujan atau tidak berdasarkan suhu dan temperatur pukul 3 sore yang diinput
hasil_prediksi = predict_rain(input_maxtemp, input_temp3pm)
print("Prediksi: ", hasil_prediksi)
```

```
Masukkan suhu maksimal : 15.4
Masukkan temperatur pukul 15.00 : 20.1
Prediksi:  Hujan
```

THANK YOU