Nama : Abraham Arghya Silaen

IMPORT LIBRARY

```
import pandas as pd
import numpy as np
```

[1] ✓ 1.1s

IMPORT DATASET

```
dataset = pd.read_csv('weather_classification_data.csv')
dataset
```

[6] ✓ 0.0s

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	14.0	73	9.5	82.0	partly cloudy	1010.82	2	Winter	3.5	inland	Rainy
1	39.0	96	8.5	71.0	partly cloudy	1011.43	7	Spring	10.0	inland	Cloudy
2	30.0	64	7.0	16.0	clear	1018.72	5	Spring	5.5	mountain	Sunny
3	38.0	83	1.5	82.0	clear	1026.25	7	Spring	1.0	coastal	Sunny
4	27.0	74	17.0	66.0	overcast	990.67	1	Winter	2.5	mountain	Rainy
...	...	...	...	...	...	...	...	...	...	...	...
13195	10.0	74	14.5	71.0	overcast	1003.15	1	Summer	1.0	mountain	Rainy
13196	-1.0	76	3.5	23.0	cloudy	1067.23	1	Winter	6.0	coastal	Snowy
13197	30.0	77	5.5	28.0	overcast	1012.69	3	Autumn	9.0	coastal	Cloudy
13198	3.0	76	10.0	94.0	overcast	984.27	0	Winter	2.0	inland	Snowy
13199	-5.0	38	0.0	92.0	overcast	1015.37	5	Autumn	10.0	mountain	Rainy

13200 rows × 11 columns

MENGECEK DATASET

```
dataset.info()
dataset.isnull().sum()
```

[7] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13200 entries, 0 to 13199
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Temperature          13200 non-null  float64
 1   Humidity             13200 non-null  int64
 2   Wind Speed           13200 non-null  float64
 3   Precipitation (%)    13200 non-null  float64
 4   Cloud Cover          13200 non-null  object
 5   Atmospheric Pressure 13200 non-null  float64
 6   UV Index             13200 non-null  int64
 7   Season               13200 non-null  object
 8   Visibility (km)      13200 non-null  float64
 9   Location             13200 non-null  object
 10  Weather Type         13200 non-null  object
dtypes: float64(5), int64(2), object(4)
memory usage: 1.1+ MB
```

```
Temperature            0
Humidity               0
Wind Speed             0
Precipitation (%)      0
Cloud Cover            0
Atmospheric Pressure   0
UV Index               0
Season                 0
```

MENGUBAH DATA KATEGORIKAL MENJADI NUMERIK

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

dataset['Season'] = le.fit_transform(dataset['Season'])
dataset['Weather Type'] = le.fit_transform(dataset['Weather Type'])

dataset
```

[38] ✓ 0.0s

	Temperature	Humidity	Wind Speed	Precipitation (%)	Cloud Cover	Atmospheric Pressure	UV Index	Season	Visibility (km)	Location	Weather Type
0	14.0	73	9.5	82.0	3	3253	2	3	3.5	1	1
1	39.0	96	8.5	71.0	3	3313	7	1	10.0	1	0
2	30.0	64	7.0	16.0	0	4027	5	1	5.5	2	3
3	38.0	83	1.5	82.0	0	4659	7	1	1.0	0	3
4	27.0	74	17.0	66.0	2	1384	1	3	2.5	2	1
...	...	...	...	...	...	...	...	...	...	...	...
13195	10.0	74	14.5	71.0	2	2532	1	2	1.0	2	1
13196	-1.0	76	3.5	23.0	1	5053	1	3	6.0	0	2
13197	30.0	77	5.5	28.0	2	3437	3	0	9.0	0	0
13198	3.0	76	10.0	94.0	2	857	0	3	2.0	1	2
13199	-5.0	38	0.0	92.0	2	3702	5	0	10.0	2	1

13200 rows × 11 columns

MEMISAHKAN FITUR DAN LABEL

```
x = dataset.drop(columns=['Weather Type', 'Cloud Cover', 'Location'])
y = dataset['Weather Type']
```

[39] ✓ 0.0s

MEMISAHKAN DATA TRAINING DAN TESTING

```
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, random_state=1)
```

[40] ✓ 0.0s

MEMBANGUN MODEL

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier()
model.fit(x_train, y_train)

y_pred = model.predict(x_test)
```

[41] ✓ 0.1s

MENGHITUNG AKURASI DAN ERROR PADA MODEL

```
from sklearn.metrics import accuracy_score, classification_report

accuracy = accuracy_score(y_test, y_pred)

print('accuracy: ', accuracy)
print('classification report: ', classification_report(y_test, y_pred))
```

[43] ✓ 0.0s

```
accuracy:  0.8666666666666667
classification report:               precision    recall  f1-score   support

           0       0.83      0.80      0.82       680
           1       0.81      0.84      0.83       678
           2       0.92      0.94      0.93       660
           3       0.91      0.88      0.90       622

    accuracy                           0.87      2640
   macro avg       0.87      0.87      0.87      2640
weighted avg       0.87      0.87      0.87      2640
```