

1. WEB SCRAPING

Scraping dilakukan dari situs web world weather online (<https://www.worldweatheronline.com/>). World weather online bertujuan untuk memberikan prakiraan cuaca harian, per jam, menit global, cuaca historis, cuaca laut, zona waktu, astronomi, dan banyak lagi.

a. Request

- Request Data Prediksi

```
import requests

url_pred = "http://api.worldweatheronline.com/premium/v1/weather.ashx?key=7fa2c1a51f8d490d89b84438230904&q=Jakarta&format=json&num_of_days=5&includelocation=yes&showlocaltime=yes&aqi=yes"

payload_pred = {}

headers_pred = {}

response_pred = requests.request("GET", url_pred, headers=headers_pred, data=payload_pred)

print(response_pred.text)
```

- Request Data Aktual

```
import requests

url_hist = "http://api.worldweatheronline.com/premium/v1/past-weather.ashx?key=e5c73e6801fb445293003332230904&q=Jakarta&format=json&date=2023-04-09&enddate=2023-04-13&includelocation=yes"

payload_hist = {}

headers_hist = {}

response_hist = requests.request("GET", url_hist, headers=headers_hist, data=payload_hist)

print(response_hist.text)
```

b. Ubah Tipe Data ke JSON

- Untuk Data Prediksi

- Mengubah tipe data setelah request

```
import json

json_data_pred = json.loads(response_pred.text)

json_data_pred['data']
```

- Mengubah tipe data jika sudah disimpan file json

```

# jika json_data_pred sudah disave dalam file .json terpisah untuk
mengatasi error saat run kembali...
# ...gunakan cara berikut untuk load kembali

# Ganti tipe ke JSON
import json

# Open the JSON file
with open('pred.json') as f_pred:
    # Load the JSON data into Python object
    json_data_pred = json.load(f_pred)

# Print the data
print(json_data_pred)

```

- **Untuk Data Aktual**

- Mengubah tipe data setelah request

```

import json
json_data_hist = json.loads(response_hist.text)
json_data_hist['data']

```

- Mengubah tipe data jika sudah disimpan file json

```

# jika json_data_hist sudah disave dalam file .json terpisah untuk
mengatasi error saat run kembali...
# ...gunakan cara berikut untuk load kembali

```

```

# Ganti tipe ke JSON
import json

# Open the JSON file
with open('history.json') as f_hist:
    # Load the JSON data into Python object
    json_data_hist = json.load(f_hist)

# Print the data
print(json_data_hist)

```

c. Membangun Data Frame

- **Untuk Data Prediksi**

```

#proses dataframe untuk data prediksi cuaca 5 hari kedepan sejak tanggal
9 April 2023 - 13 April 2023

```

```

import pandas as pd

data_pred=[]

rows = json_data_pred['data']['weather']

for i in range(0, len(rows)):

```

```

        date = json_data_pred['data']['weather'][i]['date']

        sunrise = json_data_pred['data']['weather'][i]['astronomy'][0]['sunrise']

        sunset = json_data_pred['data']['weather'][i]['astronomy'][0]['sunset']

        moonrise = json_data_pred['data']['weather'][i]['astronomy'][0]['moonrise']

        moonset = json_data_pred['data']['weather'][i]['astronomy'][0]['moonset']

        moon_phase = json_data_pred['data']['weather'][i]['astronomy'][0]['moon_phase']

        maxtempC = json_data_pred['data']['weather'][i]['maxtempC']
        maxtempF = json_data_pred['data']['weather'][i]['maxtempF']
        mintempC = json_data_pred['data']['weather'][i]['mintempC']
        mintempF = json_data_pred['data']['weather'][i]['mintempF']
        avgtempC = json_data_pred['data']['weather'][i]['avgtempC']
        avgtempF = json_data_pred['data']['weather'][i]['avgtempF']
        uv_index = json_data_pred['data']['weather'][i]['uvIndex']

    data_pred.append(

        (date, sunrise, sunset, moonrise, moonset, moon_phase, maxtempC,
        maxtempF, mintempC, mintempF, avgtempC, avgtempF, uv_index))

df_pred = pd.DataFrame(data_pred, columns=['Date', 'Prediksi Sunrise',
'Prediksi Sunset', 'Prediksi Moonrise', 'Prediksi Moonset', 'Prediksi
Fase Bulan', 'Prediksi Max Temp (C)', 'Prediksi Max Temp (F)', 'Prediksi
Min Temp (C)', 'Prediksi Min Temp (F)', 'Prediksi Avg Temp (C)',
'Prediksi Avg Temp (F)', 'Prediksi UV Index'])

df_pred

```

- Untuk Data Aktual

```

#proses dataframe untuk history data cuaca dari 9-13 April 2023

import pandas as pd

data_hist=[]

rows = json_data_hist['data']['weather']

for i in range(0, len(rows)):

    date = json_data_hist['data']['weather'][i]['date']

```

```

sunrise = json_data_hist['data']['weather'][i]['astronomy'][0]['sunrise']

sunset = json_data_hist['data']['weather'][i]['astronomy'][0]['sunset']

moonrise = json_data_hist['data']['weather'][i]['astronomy'][0]['moonrise']

moonset = json_data_hist['data']['weather'][i]['astronomy'][0]['moonset']

moon_phase = json_data_hist['data']['weather'][i]['astronomy'][0]['moon_phase']

maxtempC = json_data_hist['data']['weather'][i]['maxtempC']

maxtempF = json_data_hist['data']['weather'][i]['maxtempF']

mintempC = json_data_hist['data']['weather'][i]['mintempC']

mintempF = json_data_hist['data']['weather'][i]['maxtempF']

avgtempC = json_data_hist['data']['weather'][i]['avgtempC']

avgtempF = json_data_hist['data']['weather'][i]['avgtempF']

uv_index = json_data_hist['data']['weather'][i]['uvIndex']


data_hist.append(

    (date, sunrise, sunset, moonrise, moonset, moon_phase, maxtempC, maxtempF, mintempC,
    mintempF, avgtempC, avgtempF, uv_index)

)

df_hist = pd.DataFrame(data_hist, columns=['Date', 'Sunrise', 'Sunset', 'Moonrise',
'Moonset', 'Fase Bulan', 'Max Temp (C)', 'Max Temp (F)', 'Min Temp (C)', 'Min Temp (F)', 'Avg
Temp (C)', 'Avg Temp (F)', 'UV Index'])

df_hist

```

2. PRE-PROCESSING

Dilakukan penggantian tipe data untuk beberapa aspek yang diambil yang sebelumnya berupa object diubah menjadi bentuk integer.

```

# Mengganti tipe data pada df_hist menjadi integer
df_raw = ['Max Temp (C)', 'Max Temp (F)', 'Min Temp (C)', 'Min Temp (F)', 'Avg
Temp (C)', 'Avg Temp (F)', 'UV Index']
for i in range (len(df_raw)):
    df_hist[df_raw[i]] = df_hist[df_raw[i]].astype("int")

```

Dilakukan pengecekan tipe data untuk mengetahui apakah kolom yang diganti sudah berubah menjadi bentuk integer.

```

# pengecekan tipe data kolom pada df_hist
df_hist.info()

```

Dilakukan penggantian tipe data untuk beberapa aspek yang diambil yang sebelumnya berupa object diubah menjadi bentuk integer.

```

# Mengganti tipe data kolom pada 'df_raw' menjadi integer

```

```
df_raw1 = ['Prediksi Max Temp (C)', 'Prediksi Max Temp (F)', 'Prediksi Min
Temp (C)', 'Prediksi Min Temp (F)', 'Prediksi Avg Temp (C)', 'Prediksi Avg
Temp (F)', 'Prediksi UV Index']
for i in range (len(df_raw1)):
    df_pred[df_raw1[i]] = df_pred[df_raw1[i]].astype("int")
```

Dilakukan pengecekan tipe data untuk mengetahui apakah kolom yang diganti sudah berubah menjadi bentuk integer.

```
# pengecekan tipe data kolom pada df_pred
df_pred.info()
```

3. ANALISIS DATA

a. UV Index di Jakarta dari 9-13 April 2023

Dilakukan visualisasi dengan menggunakan line plot dengan bantuan package matplotlib. Visualisasi dilakukan untuk mengetahui perubahan yang terjadi pada UV Index dalam rentang waktu 5 hari

```
# visualisasi uv index pada 'df_hist'

import matplotlib.pyplot as plt

plt.plot(df_hist['Date'], df_hist['UV Index'])

# Menambahkan judul dan label sumbu

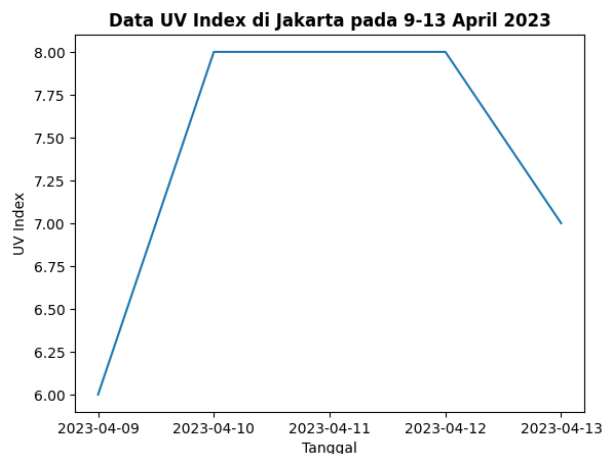
plt.title('Data UV Index di Jakarta pada 9-13 April 2023',
fontweight='bold')

plt.xlabel('Tanggal')

plt.ylabel('UV Index')

# Menampilkan plot

plt.show()
```



b. Perbandingan Data Average Temperature (F) di Jakarta 9-13 April 2023

Kembali menggunakan line plot, dilakukan visualisasi untuk membandingkan data prediksi dan aktual dari rata-rata temperatur dalam fahrenheit dengan rentang waktu 5 hari. Visualisasi memanfaatkan package matplotlib dimana pada output akan terlihat dua warna berbeda untuk masing-masing data yang digunakan.

```
# visualisasi average temperature (F) pada 'df_hist'

import matplotlib.pyplot as plt

plt.plot(df_hist['Date'], df_hist['Avg Temp (F)'], label = 'Data
Aktual')

plt.plot(df_pred['Date'], df_pred['Prediksi Avg Temp (F)'], label =
'Data Prediksi')

# Menambahkan judul dan label sumbu

plt.title('Perbandingan Data Average Temperature (F) di Jakarta pada 9-
13 April 2023', fontweight='bold')

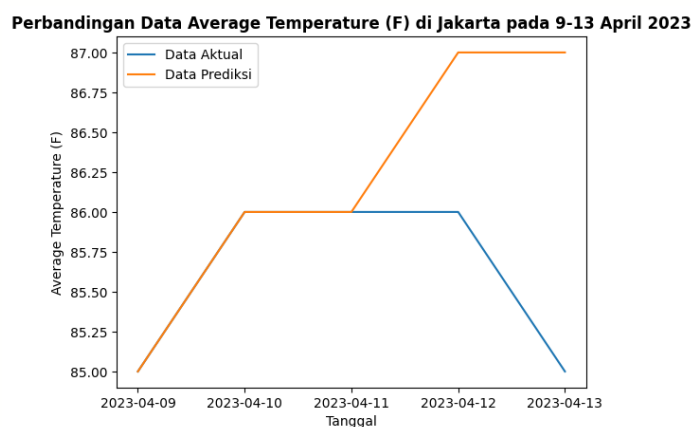
plt.xlabel('Tanggal')

plt.ylabel('Average Temperature (F)')

plt.legend()

# Menampilkan plot

plt.show()
```



c. Frekuensi Fase Bulan di Jakarta pada 9 -13 April 2023

Pada bagian ini, akan dihitung jumlah frekuensi dari fase bulan pada rentang 5 hari. Menggunakan value counts, akan dilihat frekuensi dari fase bulan yang terjadi.

```
# tabel frekuensi fase bulan pada 'df_hist'

num_loc = df_hist['Fase Bulan'].value_counts()

df_fasebulan= pd.DataFrame(num_loc).reset_index()

df_fasebulan.columns = ['Nama Fase','Jumlah']

df_fasebulan
```

Frekuensi yang sudah diperoleh selanjutnya dibentuk menjadi bar plot dengan memanfaatkan package matplotlib.

```
# visualisasi frekuensi fase bulan pada 'df_hist'

import matplotlib.pyplot as plt

plt.bar(df_fasebulan['Nama Fase'], df_fasebulan['Jumlah'])

# Menambahkan judul dan label sumbu

plt.title('Frekuensi Fase Bulan di Jakarta pada 9-13 April 2023',
fontweight='bold')

plt.xlabel('Fase Bulan')

plt.ylabel('Frekuensi')

# Menampilkan plot

plt.show()
```

