

**LAPORAN PROJECT AKHIR SEMESTER
MATA KULIAH SISTEM OPERASI**



**PROGRAM MENENTUKAN CUACA, TEMPERATURE, DAN KELEMBAPAN UDARA
DI INDONESIA**

DISUSUN OLEH : YOSUA SATRIA BARA HARMONI (21083010029)

DOSEN PENGAMPU : MOHAMMAD IDHOM, SP., S.KOM., MT.

**PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR
Jl. Rungkut Madya No.1, Gn.Anyar, Kec. Gn. Anyar, Kota SBY, Jawa Timur 60294
2022**

1. Menjalankan program

```
yosua@yosua-VirtualBox:~/UAS$ pip install -r req.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r req.txt (line 1)) (2.25.1)
Requirement already satisfied: bs4 in /home/yosua/.local/lib/python3.10/site-packages (from -r req.txt (line 2)) (0.0.1)
Requirement already satisfied: termcolor in /home/yosua/.local/lib/python3.10/site-packages (from -r req.txt (line 3)) (2.1.1)
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages (from bs4->-r req.txt (line 2)) (4.10.0)
```

Gambar 1. Tampilan gambar Halaman instalasi module dan library

a. Memanggil module yaitu “pip install -r req.txt

Langkah pertama yang kita lakukan adalah memanggil berbagai module dan library yang ada, yaitu :

a. Request :

Requests adalah modul Python yang bisa digunakan untuk mengirim berbagai request HTTP. Requests adalah library yang mudah digunakan dengan banyak fitur mulai dari melempar parameter dalam URL sampai mengirim header khusus dan verifikasi SSL

b. bs4 dan beautifulshop4 :

Modul bs4 merupakan singkatan dari modul beautifulshop4. BeautifulSoup merupakan library Python yang digunakan untuk menarik data dari file HTML dan XML. Serta dokumen HTML atau XML yang ditulis dalam pengkodean ASCII atau UTF-8, saat dokumen tersebut di load BeautifulSoup dapat mengubahnya menjadi Unicode.

c. termcolor :

Termcolor merupakan sebuah library untuk melakukan pemanggilan warna.

b. Melakukan “python3 main.py

```
yosua@yosua-VirtualBox:~/UAS$ python3 main.py
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...x
yosua@yosua-VirtualBox:~/UAS$
```

Gambar 2. Tampilan gambar Halaman pemanggilan kode script python

Untuk memanggil script, diperlukan pemanggilan dengan mengetik “python3 main.py.

2. Tampilan Halaman Utama

```
yosua@yosua-VirtualBox:~/UAS$ python3 main.py

Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...x
yosua@yosua-VirtualBox:~/UAS$
```

Gambar 3. Tampilan Utama Halaman Script Penentuan Cuaca

Pada halaman utama, terdapat tampilan utama halaman penentuan cuaca. Halaman tampilan cuaca berisi :

- [1] Update data
- [2] Set Provinsi
- [3] Set Kota / Kabupaten
- [4] Set Tanggal
- [5] Open Weather Forecast
- [6] Look Options
- [X] Exit

3. Tampilan pilihan menu [2]

```
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...2
[1] . Aceh
[2] . Bali
[3] . Bangka Belitung
[4] . Banten
[5] . Bengkulu
[6] . DI Yogyakarta
[7] . DKI Jakarta
[8] . Gorontalo
[9] . Jambi
[10] . Jawa Barat
[11] . Jawa Tengah
[12] . Jawa Timur
[13] . Kalimantan Barat
[14] . Kalimantan Selatan
[15] . Kalimantan Tengah
[16] . Kalimantan Timur
[17] . Kalimantan Utara
[18] . Kepulauan Riau
[19] . Lampung
[20] . Maluku
[21] . Maluku Utara
[22] . Nusa Tenggara Barat
[23] . Nusa Tenggara Timur
[24] . Papua
[25] . Papua Barat
[26] . Riau
[27] . Sulawesi Barat
[28] . Sulawesi Selatan
[29] . Sulawesi Tengah
[30] . Sulawesi Tenggara
[31] . Sulawesi Utara
[32] . Sumatera Barat
[33] . Sumatera Selatan
[34] . Sumatera Utara
Set Province (its number) ...
```

Gambar 4. Tampilan Halaman Pilihan 2 Set Provinsi

Pada halaman pilihan 2, berisi nama “Set Provinsi” untuk menentukan pilihan provinsi. Ada 34 provinsi di Indonesia. Untuk itu, kita harus mengetikkan nomer provinsi yang ingin kita pilih. Saya mencontohkan memilih pilihan 12 yaitu Jawa Timur.

4. Tampilan pilihan menu [3]

```
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...3
[1] . Bangkalan
[2] . Banyuwangi
[3] . Batu
[4] . Bojonegoro
[5] . Bondowoso
[6] . Gresik
[7] . Jember
[8] . Jombang
[9] . Kabupaten Blitar
[10] . Kabupaten Kediri
[11] . Kabupaten Madiun
[12] . Kabupaten Malang
[13] . Kabupaten Mojokerto
[14] . Kabupaten Pasuruan
[15] . Kabupaten Probolinggo
[16] . Kota Blitar
[17] . Kota Kediri
[18] . Kota Madiun
[19] . Kota Malang
[20] . Kota Mojokerto
[21] . Kota Pasuruan
[22] . Kota Probolinggo
[23] . Lamongan
[24] . Lumajang
[25] . Magetan
[26] . Nganjuk
[27] . Ngawi
[28] . Pacitan
[29] . Pamekasan
[30] . Ponorogo
[31] . Sampang
[32] . Sidoarjo
[33] . Situbondo
[34] . Sumenep
[35] . Surabaya
[36] . Trenggalek
[37] . Tuban
[38] . Tulungagung
```

Gambar 5. Tampilan Halaman Pilihan 3 Set Kota / Kabupaten

Setelah kita mengetikkan pilihan provinsi, kita akan keluar ke menu utama. Setelah itu, kita akan melakukan pemilihan kota / kabupaten dengan mengetik pilihan angka 3. Setelah itu, kita pilih dengan mengetikkan angka kota / kabupaten yang akan dituju. Saya mencontohkan memilih

angka 6 yaitu Gresik.

5. Tampilan pilihan menu [4]

```
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...4
[1] . Sab, 18 Des
[2] . Min, 19 Des
[3] . Sen, 20 Des
[4] . Sel, 21 Des
[5] . Rab, 22 Des
[6] . Kam, 23 Des
[7] . Jum, 24 Des
Set Date (its number) ....
```

Gambar 6. Tampilan Halaman Pilihan 4 Set Tanggal

Setelah kita mengetikkan pilihan kabupaten / kota, kita akan mencoba mengetik pilihan 4 yaitu set tanggal. Setelah meng klik pilihan 4, kita akan disediakan mulai dari 18 Desember-24 Desember. Saya akan mencoba mengetik tanggal 24 Desember.

6. Tampilan pilihan menu [1]

```
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...1
```

Gambar 7. Tampilan Halaman Pilihan 1 Update Data

Setelah mengklik tanggal, kita akan diarahkan ke halaman utama. Disini, kita harus melakukan konfigurasi update berkala dengan mengklik pilihan 1 yaitu Update data.

7. Tampilan pilihan menu [6]

```
connection error...
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...6
current options
Province          = Jawa Timur
District / city   = Gresik
Date              = Jum, 24 Des
```

Gambar 8. Tampilan Halaman Pilihan 6 Look Options

Setelah update data, kita harus melihat ulang apakah pilihan yang kita update sudah sesuai dengan permintaan. Kita harus mengklik pilihan 6 yaitu Look Options untuk melihat ulang. Setelah pilihannya benar, maka kita bisa keluar menuju halaman utama.

8. Tampilan pilihan menu [5]

```
^
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...5
[1]
Hour : 01:00 WIB
Condition :Cerah Berawan
Temperature :26°C
Humidity :90 %
Wind speed & direction =10 km/jamBarat Daya
[2]
Hour : 07:00 WIB
Condition :Berawan
Temperature :26°C
Humidity :95 %
Wind speed & direction =20 km/jamBarat
[3]
Hour : 13:00 WIB
Condition :Hujan Ringan
Temperature :30°C
Humidity :70 %
Wind speed & direction =10 km/jamBarat
[4]
Hour : 19:00 WIB
Condition :Berawan Tebal
Temperature :26°C
Humidity :90 %
Wind speed & direction =10 km/jamBarat Daya
```

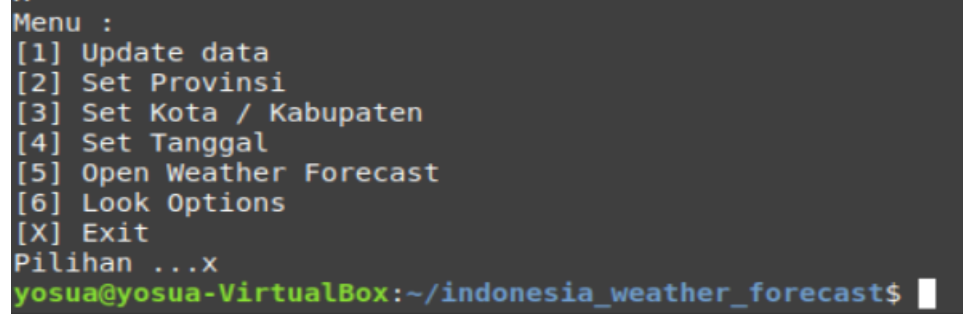
Gambar 9. Tampilan Halaman Pilihan 5 Open Weather Forecast

Untuk bisa melihat penentuan cuaca, kita dapat mengklik nomer 5 yaitu Open Weather Forecast.

Di pilihan nomer 5, terdapat berbagai opsi, yaitu :

- a. Pilihan waktu :
 - 1. Pukul 01.00
 - 2. Pukul 07.00
 - 3. Pukul 13.00
 - 4. Pukul 19.00
- b. Pilihan cuaca :
 - 1. Condition / kondisi awan
 - 2. Temperatur
 - 3. Humidity / kelembaban
 - 4. Arah angin dan kecepatan arah angin

9. Tampilan pilihan menu [X]



```
Menu :
[1] Update data
[2] Set Provinsi
[3] Set Kota / Kabupaten
[4] Set Tanggal
[5] Open Weather Forecast
[6] Look Options
[X] Exit
Pilihan ...x
yosua@yosua-VirtualBox:~/indonesia_weather_forecast$
```

Gambar 10. Tampilan Halaman Pilihan X Exit

Tampilan menu X digunakan untuk keluar dari opsi cuaca dan dari program python tersebut.

LAMPIRAN

1. Link Repository GitHub

<https://github.com/jidiyosua/21083010029/tree/PROJECT-AKHIR>

2. Script kode python main.py

```
import json
import retrieve_data
import datetime

l=open("last_update.txt","r")
isi=l.read()
l.close()
isi=isi.split(",")

#mengecek tanggal update terakhir
last_update=datetime.date(int(isi[0]),int(isi[1]),int(isi[2]))

#mengecek tanggal sekarang
date_now=datetime.datetime.now()
date_n=datetime.date(int(date_now.strftime("%Y")),int(date_now.strftime("%m")),int(date_now.strftime("%d")))

#untuk mengecek selisih tanggal sekarang dengan
tanggal terakhir update
update_int=date_n-last_update
update_int=update_int.days
if(update_int>5):
    print("")
class prakiraan_cuaca:
    def __init__(self):
        f=open("a.json","r")
        self.isi=json.loads(f.read())
        self.provinsi=0
        self.kota=0
        self.tanggal=0
    def lihat_daftar_provinsi(self):
        for isi in self.isi:
            print("[{}] .
{}".format(self.isi.index(isi)+1,isi["provinsi"]))
            self.provinsi=int(input("Set Province (its
number) ...."))
            self.provinsi=self.provinsi-1
    def lihat_daftar_kabupaten(self):

daftar_kab=self.isi[self.provinsi]["data_kota"]
    for dk in daftar_kab:
        print("[{}] .
{}".format(daftar_kab.index(dk)+1, dk["kota"]))
```



```

        self.kota = int(input("Set City / district
(its number) ...."))
        self.kota = self.kota - 1
        def lihat_tanggal(self):

daftar_tgl=self.isi[self.provinsi]["data_kota"][self.kota]["prakiraan_cuaca_kota"]
        for dt in daftar_tgl:
            print("{}" .
format(daftar_tgl.index(dt)+1, dt["tanggal"]))
            self.tanggal = int(input("Set Date (its
number) ...."))
            self.tanggal = self.tanggal - 1
            def lihat_cuaca(self):
                daftar_cuaca =
self.isi[self.provinsi]["data_kota"][self.kota]["pr
akiraan_cuaca_kota"][self.tanggal]["prakiraan_cuaca
"]
                for dc in daftar_cuaca:
                    print("{}\n Hour : {} \nCondition :{}
\nTemperature :{} \nHumidity :{} \nWind speed &
direction ={}".format(daftar_cuaca.index(dc)+1,
dc["jam"], dc["kondisi"],dc["suhu"],
dc["kelembaban"], dc["arah_mata_angin"]))
                    input()
                def lihat_opsi(self):
                    prov=self.isi[self.provinsi]["provinsi"]
                    kab =
self.isi[self.provinsi]["data_kota"][self.kota]["ko
ta"]
                    tgl =
self.isi[self.provinsi]["data_kota"][self.kota]["pr
akiraan_cuaca_kota"][self.tanggal]["tanggal"]
                    print("current options")
                    print("Province          = {}".format(prov))
                    print("District / city = {}".format(kab))
                    print("Date              = {}".format(tgl))
                    input()

pc=prakiraan_cuaca()

def tampil_menu():
    print("Menu :")
    print("[1] Update data ")
    print("[2] Set Provinsi")
    print("[3] Set Kota / Kabupaten")
    print("[4] Set Tanggal ")
    print("[5] Open Weather Forecast")
    print("[6] Look Options")
    print("[X] Exit")

```

```

pil="0"
while (pil!="X") :
    tampil_menu()
    pil=input("Pilihan ...")
    if(pil=="2") :
        pc.lihat_daftar_provinsi()
    elif (pil=="1") :
        try:
            retrieve_data.data_retrieval()
        except:
            print("Connection error...")
    elif(pil=="3") :
        pc.lihat_daftar_kabupaten()
    elif(pil=="4") :
        pc.lihat_tanggal()
    elif(pil=="5") :
        pc.lihat_cuaca()
    elif(pil=="6") :
        pc.lihat_opsi()
    elif(pil=="x") :
        break
    elif(pil=="X") :
        break
    else:
        print("Wrong command...")
        input()

```

3. Script python retrive_data.py

```

import requests as req
from bs4 import BeautifulSoup as bs
import json
from termcolor import colored
import os
import datetime

class data_retrieval:
    head={"User-Agent": "Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:94.0) Gecko/20100101
Firefox/94.0"}
    def __init__(self):
        h =
self.data_cuaca_nasional("https://www.bmkg.go.id/cu
aca/prakiraan-cuaca-indonesia.bmkg")
        h = json.dumps(h)
        f = open("a.json", "w")
        f.write(h)
        f.close()
        tgl_now = datetime.datetime.now()
        l = open("last_update.txt", "w")
        l.write(tgl_now.strftime("%Y") + "," +

```

```

tgl_now.strftime("%m") + "," +
tgl_now.strftime("%d"))
        l.close()
        def data_cuaca_kota(self,link):

hasil_akses=req.get(link,headers=self.head,timeout=
5).text
        data=[]
        parsing_hasil=bs(hasil_akses,"html.parser")

lihat_tanggal=parsing_hasil.find_all("div",{"class"
:"prakicu-kabkota tab-v1 margin-bottom-
30"})[0].find_all("ul",{"class":"nav"})[0].find_all
("li")
        for lt in lihat_tanggal:

data.append({"tanggal":lt.find_all("a")[0].get_text
()})
        cuaca_per_hari
=parsing_hasil.find_all("div",{"class":"tab-content
no-padding"})[0].select('div[id*="TabPaneCuaca"]')
        if(len(cuaca_per_hari)!=len(data)):
            cuaca_per_hari.pop()
        for cph in cuaca_per_hari:
            hasil_hari=[]

cuaca_per_jam=cph.find_all("div",{"class":"cuaca-
flex-child"})
        for cpj in cuaca_per_jam:
            hasil_jam={}

hasil_jam["jam"]=cpj.find_all("h2",{"class":"kota"}
)[0].get_text()
            hasil_jam["kondisi"] =
cpj.find_all("div", {"class":
"kiri"})[0].find_all("p")[0].get_text()
            hasil_jam["suhu"] =
cpj.find_all("div", {"class":
"kanan"})[0].find_all("h2",{"class":"heading-
md"})[0].get_text()
            hasil_jam["kelembaban"] =
cpj.find_all("div", {"class":
"kanan"})[0].find_all("p")[0].get_text()
            hasil_jam["arah_mata_angin"] =
cpj.find_all("div", {"class":
"kanan"})[0].find_all("p")[1].get_text()
            hasil_hari.append(hasil_jam)

data[cuaca_per_hari.index(cph)]["prakiraan_cuaca"]=
hasil_hari
        notif = "{:, .2f}"

```

```

%.format(((cuaca_per_hari.index(cph) + 1) /
len(cuaca_per_hari)) * 100)
        print(colored(notif, "green"))
        return data

    def data_cuaca_provinsi(self, link):

hasil_akses=req.get(link,headers=self.head,timeout=
5).text
        data=[]
        parsing_hasil=bs(hasil_akses,"html.parser")

lihat_kota=parsing_hasil.find_all("table",{ "class":
"table table-hover table-striped table-prakicu-
provinsi"})[0].find_all("a")
        i=0
        while(i<len(lihat_kota)):
            try:
                os.system('cls')
                notif="Retrieving data at {} [in
progress]".format(lihat_kota[i].get_text())
                print(colored(notif,"blue"))
                link_kota =
"https://www.bmkg.go.id/cuaca/" +
lihat_kota[i]["href"]
                dc =
self.data_cuaca_kota(link_kota)

data.append({"kota":lihat_kota[i].get_text(),"praki
raan_cuaca_kota":dc})
                notif = "retrieving complete
[progress == {:.2f} %
complete]".format(((i+1)/len(lihat_kota))*100)
                print(colored(notif, "green"))
                i=i+1
            except:
                notif="Error... reconnecting"
                print(colored(notif,"red"))
        return data

    def data_cuaca_nasional(self, link):

hasil_akses=req.get(link,headers=self.head,timeout=
5).text
        data=[]
        parsing_hasil=bs(hasil_akses,"html.parser")

lihat_provinsi=parsing_hasil.find_all("div",{ "class
":"row list-cuaca-provinsi md-margin-bottom-
10"})[0].find_all("div",{ "class":"col-sm-4 col-xs-
6"})

```

```

        lihat_provinsi.pop()
        i=0
        while(i<len(lihat_provinsi)):
            try:
                notif = "Retrieving data on
province {} [in
progress]".format(lihat_provinsi[i].get_text())
                print(colored(notif, "yellow"))

            hasil_provinsi=lihat_provinsi[i].find_all("a")[0].g
            et_text()

            link_prov="https://www.bmkg.go.id/cuaca/prakiraan-
            cuaca-
            indonesia.bmkg"+lihat_provinsi[i].find_all("a")[0][
            "href"]

            dp=self.data_cuaca_provinsi(link_prov)

            data.append({"provinsi":hasil_provinsi,"data_kota":
            dp})

                notif = "retrieving province data
            is complete [progress == {:.2f} %
            complete]".format(((i + 1) / len(lihat_provinsi)) *
            100)

                print(colored(notif, "yellow"))
                i = i + 1
            except:
                notif="Error... reconnecting"
                print(colored(notif,"red"))
        return data

```