BAB 6

6.1 Capaian Praktikum Pertemuan 6

Pada praktikum pertemuan ke-6 ini mahasiswa diharapkan mampu memahami perbedaan antara penggunaan BeautifulSoup dan Pandas dalam proses web scraping, di mana BeautifulSoup digunakan untuk mengekstrak data secara detail dari struktur HTML atau XML, sedangkan Pandas lebih difokuskan pada pembacaan dan manipulasi data dalam bentuk tabel, seperti yang terdapat pada elemen HTML <table>. Selain itu, mahasiswa juga ditargetkan mampu mengimplementasikan teknik scraping menggunakan Pandas, termasuk membaca tabel dari situs web, mengolah data yang diambil, serta menyimpannya dalam format yang sesuai untuk analisis lebih lanjut.

6.2 Indikator Capaian

- Mahasiswa tidak melanggar aturan norma dan etika dalam pengambilan gambar.
- Mahasiswa melakukan scraping URL Gambar dengan menggunakan Beautifulsoap dan Request.

6.3 Landasan Teori

Menurut Mitchell (2018) dalam bukunya "*Web Scraping with Python*", web scraping adalah teknik otomatisasi untuk mengambil data dari situs web, dengan cara mengekstrak konten HTML dan mengolahnya sesuai kebutuhan pengguna. Ia menjelaskan bahwa BeautifulSoup merupakan salah satu library Python yang populer dan dirancang untuk navigasi serta pencarian elemen dalam dokumen HTML secara efisien dan fleksibel, sehingga cocok digunakan untuk scraping data yang tidak berbentuk tabel secara langsung.

Sementara itu, menurut McKinney (2017) dalam buku *"Python for Data Analysis"*, Pandas adalah pustaka analisis data yang sangat kuat dan banyak digunakan untuk membaca, memproses, dan menyimpan data dalam berbagai format, termasuk tabel HTML. Fungsi read_html() pada Pandas memungkinkan pengguna mengambil langsung tabel dari halaman web tanpa perlu parsing manual, sehingga lebih efisien untuk data yang sudah terstruktur dalam format tabel.

6.4 Pelaksanaan Praktikum

6.4.1 Percobaan Pertama

Pada percobaan pertama mahasiswa melakukan scraping dari situs *Wikipedia*, di mana program menunjukkan proses web scraping menggunakan library Pandas untuk mengambil data tabel dari situs Wikipedia yang berisi daftar populasi negara-negara di dunia.

a. Script / Setting Program

pandas.py

```
import pandas as pd

# URL berisi tabel populasi negara-negara
url                                                                                =
'https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population'

# Ambil semua tabel dengan class 'wikitable'
df_list = pd.read_html(url, header=0, attrs={'class': 'wikitable'})

# Cek jumlah tabel dan pilih yang sesuai
print(f"Jumlah tabel ditemukan: {len(df_list)}")
print("Tampilkan kolom-kolom dari tabel pertama:")
print(df_list[0].columns)

# Gunakan tabel pertama (data populasi negara-negara)
df = df_list[0]

# Tampilkan beberapa baris awal
print("Data Populasi Negara-Negara:")
print(df.head())

# Simpan ke Excel
df.to_excel("population_by_country.xlsx", index=False)

print("\nData berhasil disimpan ke population_by_country.xlsx")
```

b. Langkah Uji Coba

- **url='https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population',** menyimpan URL halaman Wikipedia yang berisi data populasi negara-negara dalam sebuah variabel url.
- **df_list = pd.read_html(url, header=0, attrs={'class': 'wikitable'}),** mengambil semua tabel dengan class "wikitable" dari halaman web.
- **print(f"Jumlah tabel ditemukan: {len(df_list)}"),** menampilkan jumlah tabel yang berhasil di-scrape dari halaman tersebut.
- **print("Tampilkan kolom-kolom dari tabel pertama:") print(df_list[0].columns),** menampilkan nama kolom dari tabel pertama dalam df_list.
- **df = df_list[0],** menyimpan tabel pertama (index ke-0) ke dalam variabel df.
- **print("Data Populasi Negara-Negara:") print(df.head()),** menampilkan 5 baris pertama dari tabel populasi.
- **df.to_excel("population_by_country.xlsx", index=False),** enyimpan DataFrame df ke file Excel bernama population_by_country.xlsx tanpa menyertakan indeks.
- **print("\nData berhasil disimpan ke population_by_country.xlsx"),** menampilkan pesan konfirmasi ke terminal.

c. Hasil Uji Coba

```
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan6> & "C:/Program Files/Python312/python.exe" "c:/sti
Jumlah tabel ditemukan: 1
Tampilkan kolom-kolom dari tabel pertama:
Index(['Location', 'Population', '% of world', 'Date',
       'Source (official or from the United Nations)', 'Notes'],
      dtype='object')
Data Populasi Negara-Negara:
       Location  Population % of world        Date Source (official or from the United Nations) Notes
0         World  8232000000       100%  13 Jun 2025                      UN projection[1][3]   NaN
1         India  1417492000      17.3%   1 Jul 2025                  Official projection[4]   [b]
2         China  1408280000      17.2%  31 Dec 2024                    Official estimate[5]   [c]
3  United States   340110988       4.2%   1 Jul 2024                    Official estimate[6]   [d]
4     Indonesia   284438782       3.5%  30 Jun 2025             National annual projection[7]   NaN

Data berhasil disimpan ke population_by_country.xlsx
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan6>
```

	Location	Populatio	% of worl	Date	Source (official or from the United Nations	Note
2	World	8232000000	100%	13 Jun 2025	UN projection[1][3]	
3	India	1417492000	17.3%	1 Jul 2025	Official projection[4]	[b]
4	China	1408280000	17.2%	31 Dec 2024	Official estimate[5]	[c]
5	United States	340110988	4.2%	1 Jul 2024	Official estimate[6]	[d]
6	Indonesia	284438782	3.5%	30 Jun 2025	National annual projection[7]	
7	Pakistan	241499431	2.9%	1 Mar 2023	2023 census result[8]	[e]
8	Nigeria	223800000	2.7%	1 Jul 2023	Official projection[9]	
9	Brazil	212583750	2.6%	1 Jul 2024	Official projection[10]	
10	Bangladesh	169828911	2.1%	14 Jun 2022	2022 census result[11]	[f]

Gambar 6.1 Hasil Scraping dari halaman Wikipedia

- Program berhasil dijalankan, proses scraping berhasil mengambil tabel dari halaman Wikipedia secara utuh. Struktur dan format tabel tetap terjaga dengan baik setelah dikonversi ke Excel. Data dapat digunakan langsung untuk analisis lebih lanjut, misalnya sorting, filtering, atau visualisasi.

d. Analisa Hasil

Hasil scraping menunjukkan bahwa data populasi berhasil diambil dengan baik dari Wikipedia, menampilkan 10 negara dengan populasi terbesar di dunia secara rapi. India dan China mendominasi dengan lebih dari sepertiga populasi global, sementara Indonesia berada di posisi kelima dengan 3,5%. Data disertai tanggal dan sumber resmi, sehingga dapat dipercaya dan layak untuk analisis lebih lanjut.

6.4.2 Percobaan Kedua

Pada percobaan kedua dilakukan dengan proses scraping data COVID-19 dari Wikipedia menggunakan Pandas dengan fokus pada kolom "Cases" dan "Deaths". Data dibersihkan dari karakter non-numerik seperti koma dan tanda tambah, lalu difilter agar hanya menyisakan nilai numerik yang valid. Setelah itu, dilakukan konversi tipe data ke float dan ditambahkan kolom baru bernama "Jumlah" yang merupakan hasil perkalian antara jumlah kasus dan jumlah kematian. Hasil akhir disimpan dalam file Excel covid_tambah_kolom.xlsx untuk analisis lanjutan.

a. Script / Setting Program

Tugas.py

```
import pandas as pd

# URL sumber data
url    =    'https://en.wikipedia.org/wiki/Template:COVID-19_pandemic_data#covid-19-
pandemic-data'

# Atribut HTML tabel
attrs = {'class': 'wikitable'}

# Membaca tabel
df_list = pd.read_html(url, header=0, index_col=0, attrs=attrs)
df = df_list[0]

# Tampilkan kolom-kolom awal untuk analisa
print("Kolom tersedia:", df.columns)

# Bersihkan nama kolom jika perlu
df.columns = df.columns.str.replace(r"\[.*\]", "", regex=True).str.strip()

# Ganti koma & tanda + di angka agar bisa dikonversi ke float
df["Cases"]    =    df["Cases"].astype(str).str.replace(",",    "").str.replace("+",
"").str.strip()
df["Deaths"]   =   df["Deaths"].astype(str).str.replace(",",   "").str.replace("+",
"").str.strip()

# Hapus baris dengan nilai kosong atau bukan angka
df = df[df["Cases"].str.isnumeric() & df["Deaths"].str.isnumeric()]

# Konversi ke float
df["Cases"] = df["Cases"].astype(float)
df["Deaths"] = df["Deaths"].astype(float)

# Tambahkan kolom baru "Jumlah"
df["Jumlah"] = df["Cases"] * df["Deaths"]

# Simpan ke Excel
df.to_excel("covid_tambah_kolom.xlsx", index=True)

# Tampilkan hasil akhir
print(df[["Cases", "Deaths", "Jumlah"]].head())
```

b. Langkah Uji Coba

- **url='https://en.wikipedia.org/wiki/Template:COVID19_pandemic_data#covid-19-pandemic-data',** menyimpan URL halaman Wikipedia berisi data COVID-19 ke variabel url.

- **attrs = {'class': 'wikitable'},** menentukan atribut HTML (class="wikitable") sebagai filter untuk mengambil tabel yang diinginkan.
- **df_list = pd.read_html(url, header=0, index_col=0, attrs=attrs),** membaca semua tabel dari halaman yang memiliki class wikitable.
- **df = df_list[0],** mengambil tabel pertama dari df_list untuk diproses**.**
- **print("Kolom tersedia:", df.columns),** menampilkan semua nama kolom dari tabel.
- **df.columns = df.columns.str.replace(r"\[.*\]", "", regex=True).str.strip(),** membersihkan nama kolom dari catatan kaki (misalnya [1], [2]) dan spasi.
- **df["Cases"] = df["Cases"].astype(str).str.replace(",", "").str.replace("+", "").str.strip(),** menghapus koma, tanda plus, dan spasi dari kolom "Cases" lalu ubah ke tipe string.
- **df["Deaths"] = df["Deaths"].astype(str).str.replace(",", "").str.replace("+", "").str.strip(),** menghapus koma, tanda plus, dan spasi dari kolom "Deaths" lalu ubah ke tipe string.
- **df = df[df["Cases"].str.isnumeric() & df["Deaths"].str.isnumeric()],** memfilter hanya baris yang berisi angka valid pada kolom "Cases" dan "Deaths".
- **df["Cases"] = df["Cases"].astype(float)**
  df["Deaths"] = df["Deaths"].astype(float), mengubah tipe data "Cases" dan "Deaths" dari string ke float.
- **df["Jumlah"] = df["Cases"] * df["Deaths"],** membuat kolom baru "Jumlah" hasil dari "Cases" dikali "Deaths".
- **df.to_excel("covid_tambah_kolom.xlsx", index=True),** menyimpan hasil akhir ke file Excel.
- **print(df[["Cases", "Deaths", "Jumlah"]].head()),** menampilkan 5 baris pertama dari kolom hasil akhir.

c. Hasil Uji Coba

```
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan6> & "C:/Program Files/Python312/python.exe
Kolom tersedia: Index(['Location', 'Cases', 'Deaths'], dtype='object')
c:\stikom\smt 4\Praktikum Data Scraping\pertemuan6\tugas.py:27: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/i
  df["Cases"] = df["Cases"].astype(float)
c:\stikom\smt 4\Praktikum Data Scraping\pertemuan6\tugas.py:28: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/i
  df["Deaths"] = df["Deaths"].astype(float)
c:\stikom\smt 4\Praktikum Data Scraping\pertemuan6\tugas.py:31: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/i
  df["Jumlah"] = df["Cases"] * df["Deaths"]
          Cases     Deaths        Jumlah
NaN  775866783.0  7057132.0  5.475394e+15
NaN  185822587.0  1262988.0  2.346917e+14
NaN  103436829.0  1193165.0  1.234172e+14
NaN   99373219.0   122304.0  1.215374e+13
NaN   45041748.0   533623.0  2.403531e+13
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan6>
```

Location	Case	Deat	Jumlah
World[a]	7,76E+08	7057132	5,47539E+15
European Union[b]	1,86E+08	1262988	2,34692E+14
United States	1,03E+08	1193165	1,23417E+14
China[c]	99373219	122304	1,21537E+13
India	45041748	533623	2,40353E+13
France	38997490	168091	6,55513E+12
Germany	38437756	174979	6,7258E+12
Brazil	37511921	702116	2,63377E+13
South Korea	34571873	35934	1,24231E+12

Gambar 6.2 Hasil Scraping COVID-19 pada halaman Wikipedia

- Hasil scraping dari Wikipedia yang berisi data jumlah kasus (Case) dan kematian (Deat) akibat COVID-19 di berbagai wilayah, seperti *World*, *European Union*, *United States*, *India*, dan lainnya. Selain itu, kolom baru bernama **Jumlah** berhasil ditambahkan sebagai hasil perkalian antara jumlah kasus dan jumlah kematian pada masing-masing wilayah.

Nilai-nilai dalam kolom Case, Deat, dan Jumlah ditampilkan dalam **notasi ilmiah (exponential)** karena besarnya angka, contoh:

- World[a] memiliki 776 juta kasus dan 7 juta kematian, menghasilkan nilai Jumlah sebesar **5.47×10^{15}**.
- Brazil memiliki lebih dari 37 juta kasus dan 702 ribu kematian, dengan hasil perkalian sekitar **2.63×10^{13}**.

d. Analisa Hasil

Program uji coba scraping menunjukkan bahwa data kasus dan kematian COVID-19 berhasil di-scrape dan diolah dengan baik, termasuk penambahan kolom baru "Jumlah" yang merupakan hasil perkalian antara jumlah kasus dan kematian. Nilai-nilai ditampilkan dalam notasi ilmiah karena ukuran datanya sangat besar, mencerminkan dampak global pandemi. Data ini siap digunakan untuk analisis lebih lanjut seperti perbandingan antar negara atau visualisasi.

6.5 Kesimpulan

6.5.1 Kesimpulan Percobaan 1

Mahasiswa telah melakukan percobaan 1, program berhasil melakukan proses scraping tabel populasi negara dari Wikipedia menggunakan Pandas berhasil dilakukan dengan baik, mulai dari pembacaan tabel HTML, pemilihan data, hingga penyimpanan ke file Excel. Data yang diperoleh memiliki struktur rapi dan mencakup informasi penting seperti jumlah populasi, persentase terhadap populasi dunia, dan sumber data. Hasil ini membuktikan bahwa Pandas sangat efektif untuk mengambil dan mengelola data tabular dari situs web secara otomatis dan efisien.

6.5.2 Kesimpulan Percobaan 2

Mahasiswa telah berhasil melakukan percobaan 2, program berhasil melakukan proses scraping data COVID-19 dari Wikipedia menggunakan Pandas, termasuk pembersihan data, konversi tipe numerik, dan penambahan kolom baru "Jumlah"

sebagai hasil perkalian antara jumlah kasus dan kematian. Data yang awalnya tidak terstruktur berhasil diolah menjadi tabel rapi dan informatif, lalu disimpan dalam file Excel untuk kebutuhan analisis lebih lanjut. Percobaan ini menunjukkan bahwa Pandas sangat andal dalam mengelola data web berskala besar secara efisien.

Mengetahui:

Dosen Pengampu Mata Kuliah

Arif Hadi Sumitro , M.Kom

NIKP. xxx

DAFTAR PUSTAKA

1. Mitchell, R. (2018). *Web Scraping with Python: Collecting Data from the Modern Web* (2nd ed.). O'Reilly Media.
2. *McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.*