MODUL 7. PYTHON WEB SCRAPING
SCRAPING MULTIPLE PAGES

7. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa :

1. Mahasiswa memahami konsep scraping multiple pages atau scraping lebih dari satu halaman web untuk mengumpulkan data yang lebih luas dan representatif.
2. Mahasiswa memahami struktur URL dinamis pada situs web yang memungkinkan untuk mengubah parameter seperti nomor halaman atau filter pencarian untuk mengakses halaman yang berbeda..

7.1 INDIKATOR

Setelah mempelajari Bab ini, mahasiswa mampu :

1. Mahasiswa mampu menggunakan teknik navigasi, seperti mengikuti tautan dan menangani halaman selanjutnya atau nomor halaman untuk mengakses konten dari berbagai halaman web.
2. Mahasiswa mampu menangani paginasi, yaitu tata letak halaman yang memerlukan kursor atau tombol "Next" untuk mengakses halaman selanjutnya.
3. Mahasiswa mampu menggabungkan hasil scraping dari beberapa halaman menjadi satu set data yang lengkap dan terstruktur.

7.2 URAIAN MATERI

Terdapat dua cara untuk mengekstraksi data pada website yang memiliki lebih dari 1 halaman.

1. Cara pertama adalah menulis dengan membuat kode yang ditujukan untuk ekstraksi Alamat web yang sama (looping)
2. Cara kedua adalah dengan kode yang ditujukan untuk ekstraksi pada URL situs di halaman lain

- **Scraping pada lebih dari 1 halaman website**

  Kode dibawah ini merupakan contoh penggunaan BeautifulSoup untuk scraping pada lebih dari 1 halaman website. Kode tersebut me-retrieve semua URL yang ada pada web tersebut.

```
from bs4 import BeautifulSoup
import requests

start_url = "https://proxyway.com/guides"

def scrape_page(url):
    print ("URL: " + url)
    r = requests.get(url)
    soup = BeautifulSoup(r.content, "html.parser")
    get_data(soup)
    next_page_link = soup.find("a", class_="next")
    if next_page_link is not None:
        href = next_page_link.get("href")
        scrape_page(href)
    else:
        print ("Done")


def get_data(content):
    #we could do some scraping of web content here
    pass

def main():
    scrape_page(start_url)


if __name__ == "__main__":
    main()
```

Latihan 1

```
import requests
from bs4 import BeautifulSoup

def scrape_multiple_pages(url_template, num_pages):
    # Membuat list untuk menyimpan hasil scraping dari setiap halaman
    all_data = []

    # Melakukan iterasi sebanyak jumlah halaman yang ingin di-scrape
    for page in range(1, num_pages+1):
        # Membuat URL untuk setiap halaman menggunakan template URL
        url = url_template.format(page)

        # Mengambil konten HTML dari halaman web
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')

        # Melakukan scraping data dari halaman
```

```
        # Misalnya, disini kita hanya mencari judul artikel
        article_titles = soup.find_all('h2', class_='article-title')

        # Menyimpan hasil scraping dari halaman ini ke dalam list
        for title in article_titles:
            all_data.append(title.text.strip())

    return all_data

# URL template dengan {} sebagai tempat untuk nomor halaman
url_template = 'https://example.com/page/{}'
# Jumlah halaman yang ingin di-scrape
num_pages = 3

# Memanggil fungsi untuk melakukan scraping
scraped_data = scrape_multiple_pages(url_template, num_pages)

# Menampilkan hasil scraping
for idx, data in enumerate(scraped_data, start=1):
    print(f"{idx}. {data}")
```

Pastikan untuk mengganti url_template dengan URL yang sesuai dengan struktur halaman web yang ingin Anda scrape

- **Scraping pada suatu page tertentu dalam suatu website**

  Kode program dibawah merupakan bentuk implementasi BeautifulSoup untuk menambang data pada suatu page tertentu dalam suatu website.

```
import requests
from bs4 import BeautifulSoup as bs

URL = 'https://proxyway.com/guides/page/2'

req = requests.get(URL)
soup = bs(req.text, 'html.parser')

titles = soup.find_all('h3',attrs = {'class','archive-list__title'})

print(titles)

print(titles[1].text)
```

Latihan 2

```
import requests
from bs4 import BeautifulSoup

def scrape_single_page(url):
    # Mengambil konten HTML dari halaman web
    response = requests.get(url)
    soup = BeautifulSoup(response.content, 'html.parser')

    # Melakukan scraping data dari halaman
    # Misalnya, disini kita mencari judul artikel
    article_titles = soup.find_all('h2', class_='article-title')

    # Menyimpan hasil scraping ke dalam list
    scraped_data = []
    for title in article_titles:
        scraped_data.append(title.text.strip())

    return scraped_data

# URL halaman yang ingin di-scrape
url = 'https://example.com/page-to-scrape'

# Memanggil fungsi untuk melakukan scraping
scraped_data = scrape_single_page(url)

# Menampilkan hasil scraping
for idx, data in enumerate(scraped_data, start=1):
    print(f"{idx}. {data}")
```

Pastikan untuk mengganti url_template dengan URL yang sesuai dengan struktur halaman web yang ingin Anda scrape

- **Scraping pada banyak halaman dalam suatu website**

  Kode program ini merupakan bentuk implementasi BeautifulSoup untuk menambang data pada banyak halaman pada sebuah website.

```python
import requests
from bs4 import BeautifulSoup as bs

URL = 'https://proxyway.com/guides/page/'

for page in range(1, 9):
    print("\n")
    print("Sub Titles Page : ", page, "\n")

    req = requests.get(URL + str(page) + '/')
    soup = bs(req.text, 'html.parser')

    titles = soup.find_all('h3', attrs={'class', 'archive-list__title'})

    if page == 9 :
        for i in range(0, 10):
            print(f"{i+1} " + titles[i].text)
    else :
        for i in range(0, 12):
            print(f"{i+1} " + titles[i].text)
```

Latihan 3

```
import requests
from bs4 import BeautifulSoup
def scrape_multiple_pages(base_url, num_pages):
    scraped_data = []

    # Loop melalui setiap halaman untuk di-scrape
    for page_num in range(1, num_pages+1):
        # Membuat URL untuk setiap halaman
        url = f"{base_url}/page/{page_num}"

        # Mengambil konten HTML dari halaman web
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')
```

```
        # Melakukan scraping data dari halaman
        # Misalnya, disini kita mencari judul artikel
        article_titles = soup.find_all('h2', class_='article-title')

        # Menambahkan hasil scraping ke dalam list
        for title in article_titles:
            scraped_data.append(title.text.strip())

    return scraped_data

# URL base dari website yang ingin di-scrape
base_url = 'https://example.com/articles'

# Jumlah halaman yang ingin di-scrape
num_pages = 5

# Memanggil fungsi untuk melakukan scraping
scraped_data = scrape_multiple_pages(base_url, num_pages)

# Menampilkan hasil scraping
for idx, data in enumerate(scraped_data, start=1):
    print(f"{idx}. {data}")
```

Pastikan untuk mengganti base_url dengan URL base dari website yang ingin Anda scrape dan num_pages dengan jumlah halaman yang ingin Anda scrape.

- **Menyimpan data dalam forat CSV.**

```
import requests
from bs4 import BeautifulSoup as bs
import csv

URL = 'https://proxyway.com/guides/page/'

titles_list = []
title_page9 = []

for page in range(1,10):
    req = requests.get(URL + str(page) + '/')
    soup = bs(req.text, 'html.parser')

    titles = soup.find_all('h3', attrs={'class', 'archive-list__title'})

    count = 1
    for title in titles:
        d = {}
        d['Page Number'] = f'Page {page}'
        d['Title Number'] = f'Title {count}'
        d['Title Name'] = title.text
        count += 1
        titles_list.append(d)
```

```
    if page == 9 :
        count = 1
        for title in titles:
            e = {}
            e['Page Number'] = f'Page {page}'
            e['Title Number'] = f'Title {count}'
            e['Title Name'] = title.text
            count+=1
            title_page9.append(e)


titles_all = titles_list + title_page9

#print(title_page9)

filename = 'titles.csv'
with open(filename, 'w', newline='') as f:
    w = csv.DictWriter(f,['Page Number','Title Number','Title Name'])
    w.writeheader()
    
    w.writerows(titles_all)
```

- **TUGAS LATIHAN SCRAPING WEBSITE**
  1. Silahkan scraping data di website https://proxyway.com/reviews atau https://proxyway.com/news
  2. Ambil data judul/title di setiap halaman
  3. Simpan data ke dalam format CSV