

BAB 5

5.1 Capaian Praktikum Pertemuan 5

Pada praktikum pertemuan ke-5 ini mahasiswa diharapkan mampu memahami penggunaan function dalam Python yang digunakan untuk melakukan proses scraping data dari web. Selain itu, mahasiswa juga diharapkan dapat memahami cara kerja modul requests dan BeautifulSoup dalam menentukan serta mengekstrak sub link dari sebuah halaman utama. Dengan pemahaman tersebut, mahasiswa mampu melakukan proses scraping secara menyeluruh, termasuk mengambil data dari halaman utama maupun dari sub link yang menjadi bagian dari struktur data web tersebut.

5.2 Indikator Capaian

- Mahasiswa melakukan pemanggilan function yang sudah dibuat.
- Mahasiswa membuat function baru untuk melakukan sub scraping.
- Mahasiswa melakukan scraping secara menyeluruh sari sub URL yang ditemukan.

5.3 Landasan Teori

Menurut Mitchell (2018) dalam bukunya "*Web Scraping with Python*", web scraping adalah teknik otomatisasi untuk mengekstrak informasi dari situs web menggunakan program. Proses ini mencakup pengambilan data melalui permintaan HTTP dan parsing HTML untuk mengambil elemen spesifik seperti teks, gambar, atau link.

Menurut Richardson, L. (2023), BeautifulSoup adalah library Python yang digunakan untuk memarsing dokumen HTML dan XML. Menurut dokumentasi resmi, BeautifulSoup memungkinkan navigasi struktur HTML secara mudah dan fleksibel, serta sering digunakan untuk mencari elemen dengan tag tertentu.

Menurut Reitz (2015), Library requests pada Python digunakan untuk melakukan permintaan HTTP ke suatu halaman web, requests menyederhanakan proses pengambilan data dari web karena API-nya bersifat human-friendly dan fleksibel.

Dalam proses web scraping, terkadang informasi penting tersebar di halaman-halaman sub-link. Menurut Prasad et al. (2018), pengambilan data dari sub-link merupakan bagian penting dari teknik scraping yang mendalam (deep scraping), di mana program menavigasi dari halaman utama ke halaman-halaman detail untuk mendapatkan informasi yang lebih kaya.

5.4 Pelaksanaan Praktikum

5.4.1 Percobaan Pertama

Pada percobaan pertama mahasiswa melakukan scraping dari situs *quotes.toscrape.com*, di mana program mengekstrak kutipan beserta nama penulis dari halaman utama, lalu secara otomatis mengikuti tautan detail ke halaman profil masing-masing penulis. Dari halaman detail tersebut, scraper mengambil informasi tambahan seperti nama lengkap, tanggal lahir, tempat lahir, dan deskripsi singkat penulis, yang kemudian disimpan dalam bentuk file teks per penulis di dalam folder Hasil.

a. Script / Setting Program

scraper.py

```
from bs4 import BeautifulSoup
import requests
import fungsi

BASE_URL = "https://quotes.toscrape.com"

def get_details(relative_url, directory):
    url = BASE_URL + relative_url
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")

    nama = soup.find("h3", class_="author-title").text.strip()
    tanggal_lahir = soup.find("span", class_="author-born-date").text.strip()
    tempat_lahir = soup.find("span", class_="author-born-location").text.strip()
    deskripsi = soup.find("div", class_="author-description").text.strip()
```

```

isi = f>Nama: {nama}\nTanggal Lahir: {tanggal_lahir}\nTempat Lahir:
{tempat_lahir}\n\nDeskripsi:\n{deskripsi}\n\n"
file_path = f"{directory}/{nama.replace(' ', '_')}.txt"

with open(file_path, 'w', encoding='utf-8') as f:
    f.write(isi)

def main_scraper(url, directory):
    fungsi.create_directory(directory)
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")
    quotes = soup.find_all("div", class_="quote")

    for quote in quotes:
        teks = quote.find("span", class_="text").text.strip()
        author = quote.find("small", class_="author").text.strip()
        detail_link = quote.find("a")["href"]

        print(f"Quote: {teks}")
        print(f"Author: {author}")
        print(f"Detail URL: {BASE_URL + detail_link}\n")

        fungsi.write_to_file(f"{directory}/quotes.txt", f"{teks} - {author}\n")

    get_details(detail_link, directory)

# Jalankan scraper
main_scraper("https://quotes.toscrape.com", "Hasil")

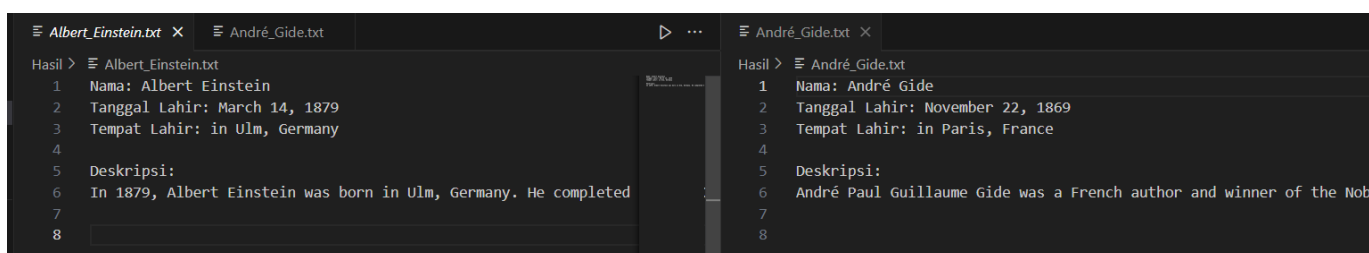
```

b. Langkah Uji Coba

- **BeautifulSoup**: Untuk mem-parsing dan mengekstrak elemen HTML.
- **requests**: Untuk mengirim permintaan HTTP (GET) ke situs web.
- **BASE_URL** = <https://quotes.toscrape.com>, menyimpan alamat situs target.
- **def get_details(relative_url, directory)**, mengambil detail penulis dari halaman profil mereka.
- **url = BASE_URL + relative_url**, pastikan hasil gabungan benar.
- **response = requests.get(url)**, pastikan respon HTTP = 200.
- **soup = BeautifulSoup(response.text, "html.parser")**, pastikan HTML berhasil diparsing.
- **isi = f>Nama: {nama}\nTanggal Lahir: {tanggal_lahir}\nTempat Lahir: {tempat_lahir}\n\nDeskripsi:\n{deskripsi}\n\n"**, pastikan format teks benar sebelum ditulis ke file.

- **file_path = f"{directory}/{nama.replace(' ', '_')}.txt"**, pastikan penamaan file benar.
- **with open(file_path, 'w', encoding='utf-8') as f: f.write(isi)**, cek folder Hasil/ apakah file berhasil ditulis dan isinya sesuai.
- **def main_scraper(url, directory)**, pastikan fungsi dipanggil di bagian bawah program.
- **fungsi.create_directory(directory)**, pastikan folder Hasil/ dibuat jika belum ada.
- **quotes = soup.find_all("div", class_="quote")**, pastikan elemen quote ditemukan.
- **for quote in quotes**, looping akan dilakukan pada setiap kutipan di halaman.
- **fungsi.write_to_file(f"{directory}/quotes.txt", f"{teks} - {author}\n")**, buka file quotes.txt di folder Hasil/ dan pastikan kutipan tersimpan dengan benar.
- **get_details(detail_link, directory)**, pastikan detail author diambil dan disimpan ke file masing-masing nama author.
- **main_scraper("https://quotes.toscrape.com", "Hasil")**, jalankan program dengan python nama_file.py dan pastikan folder Hasil/ berisi: quotes.txt → berisi daftar kutipan, file per author → Albert_Einstein.txt, Jane_Austen.txt, dll.

c. Hasil Uji Coba



Gambar 5.1 Hasil Scraping dari halaman Quotes to Scrape

- Program berhasil dijalankan, fungsi `get_details()` dan `main_scraper()` berjalan sesuai harapan. Data dari halaman detail (sub-link author) berhasil diambil dan

disimpan. Program berhasil membuat file .txt per penulis dengan konten terstruktur. Tidak ditemukan error parsing, encoding, maupun file write.

d. Analisa Hasil

Hasil uji coba menunjukkan bahwa Struktur Data tersimpan rapi. Setiap informasi penulis seperti nama, tanggal lahir, tempat lahir, dan deskripsi berhasil diambil dan disimpan ke dalam file .txt secara terstruktur dan sesuai format yang diinginkan.

5.4.2 Percobaan Kedua

Pada percobaan kedua dilakukan dengan mengubah URL target menjadi halaman khusus tag tertentu, yaitu "https://quotes.toscrape.com/tag/inspirational/page/1/", serta mengganti folder penyimpanan hasil menjadi "DataScraping". Pada percobaan ini, program tidak hanya mengambil kutipan dan informasi penulis seperti sebelumnya, tetapi juga melakukan penelusuran berdasarkan tag dari halaman detail penulis untuk mendapatkan kutipan tambahan yang relevan. Dengan demikian, percobaan ini telah berhasil menampilkan scraping hingga dua tingkat kedalaman sub-URL dan menyimpan data secara otomatis dalam folder baru yang telah ditentukan.

a. Script / Setting Program

Main.py

```
from bs4 import BeautifulSoup
import requests
import fungsi

BASE_URL = "https://quotes.toscrape.com"

def scrape_by_tag(tag_url, directory, author_name):
    """Level 3: scrape kutipan dari halaman tag penulis"""
    url = BASE_URL + tag_url
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")
    quotes = soup.find_all("div", class_="quote")

    fungsi.write_to_file(f"{directory}/{author_name.replace(' ', '_')}.txt",
        "\nKutipan lain berdasarkan tag:\n")
```

```

        for q in quotes:
            text = q.find("span", class_="text").text.strip()
            fungsi.write_to_file(f"{directory}/{author_name.replace(' ', '_')}.txt", f"-{text}")

def get_details(relative_url, directory):
    """Level 2: scrape data author dan tag"""
    url = BASE_URL + relative_url
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")

    nama = soup.find("h3", class_="author-title").text.strip()
    tanggal_lahir = soup.find("span", class_="author-born-date").text.strip()
    tempat_lahir = soup.find("span", class_="author-born-location").text.strip()
    deskripsi = soup.find("div", class_="author-description").text.strip()

    isi = f>Nama: {nama}\nTanggal Lahir: {tanggal_lahir}\nTempat Lahir: {tempat_lahir}\n\nDeskripsi:\n{deskripsi}\n"
    file_path = f"{directory}/{nama.replace(' ', '_')}.txt"

    with open(file_path, 'w', encoding='utf-8') as f:
        f.write(isi)

    tag_section = soup.find_all("a", class_="tag")
    if tag_section:
        tag_href = tag_section[0]["href"]
        scrape_by_tag(tag_href, directory, nama)

def main_scraper(url, directory):
    """Level 1: scrape kutipan dan link penulis"""
    fungsi.create_directory(directory)
    response = requests.get(url)
    soup = BeautifulSoup(response.text, "html.parser")
    quotes = soup.find_all("div", class_="quote")

    for quote in quotes:
        teks = quote.find("span", class_="text").text.strip()
        author = quote.find("small", class_="author").text.strip()
        detail_link = quote.find("a")["href"]

        print(f"Quote: {teks}")
        print(f"Author: {author}")
        print(f"Detail URL: {BASE_URL + detail_link}\n")

        fungsi.write_to_file(f"{directory}/quotes.txt", f"{teks} - {author}\n")

        get_details(detail_link, directory)

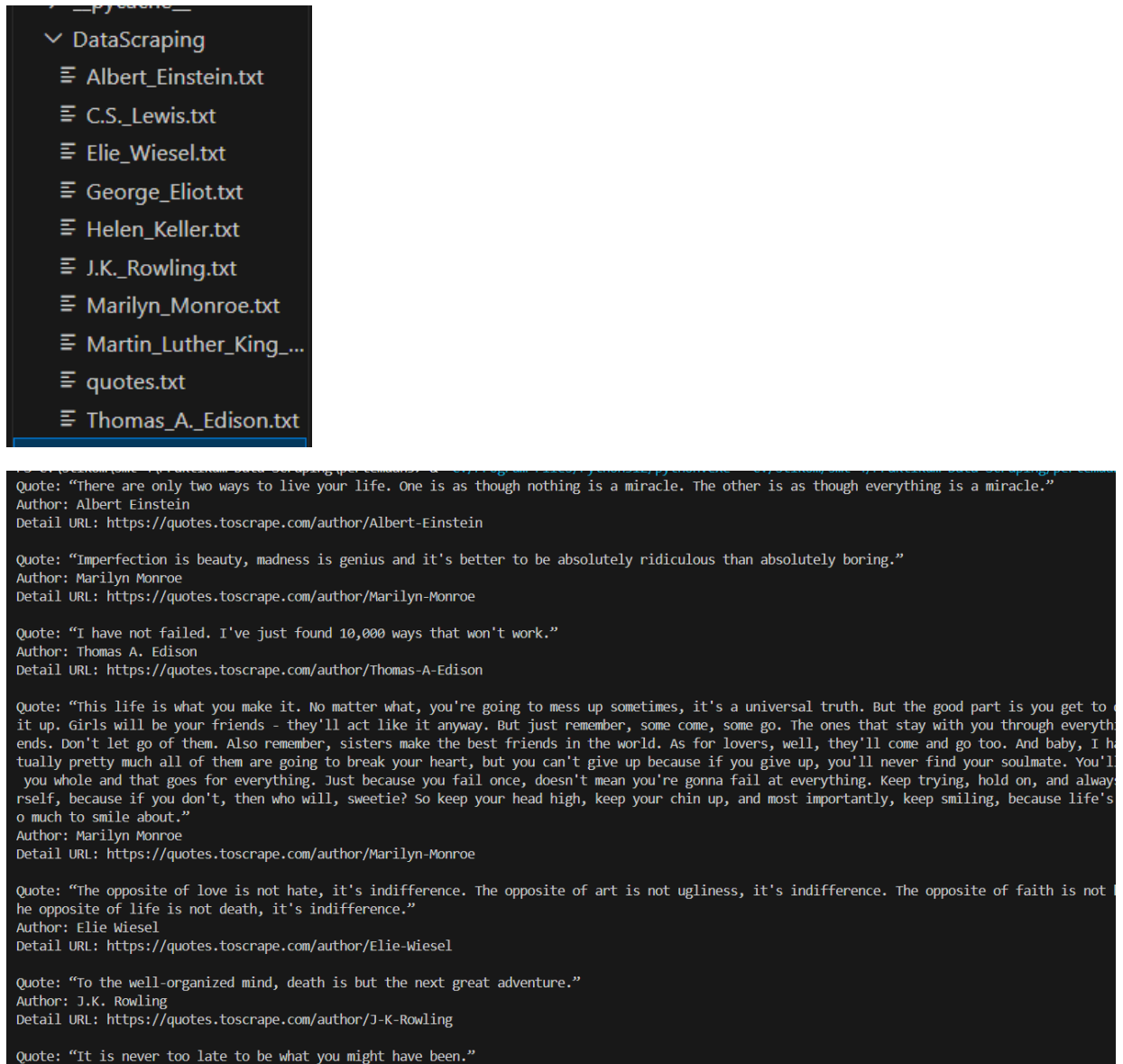
# Jalankan scraper dengan URL baru dan folder penyimpanan baru
main_scraper("https://quotes.toscrape.com/tag/inspirational/page/1/",
            "DataScraping")

```

b. Langkah Uji Coba

- **BeautifulSoup**: Untuk mem-parsing dan mengekstrak elemen HTML.
- **requests**: Untuk mengirim permintaan HTTP (GET) ke situs web.
- **url = 'https://books.toscrape.com/'**, situs web yang dituju.
- **def scrape_by_tag(tag_url, directory, author_name)**, fungsi dipanggil dari **get_details()**, pastikan parameter valid.
- **url = BASE_URL + tag_url**, pastikan hasil gabungan URL benar.
- **quotes = soup.find_all("div", class_="quote")**, cek **len(quotes)** untuk memastikan kutipan ditemukan di halaman tag.
- **fungsi.write_to_file(f"{directory}/{author_name.replace(' ', '_')}.txt", "\nKutipan lain berdasarkan tag:\n")**, lihat file output apakah bagian heading kutipan dari tag ditambahkan.
- **def get_details(relative_url, directory)**, fungsi dipanggil dari **main_scraper()**, parameter **relative_url** valid.
- **isi = f"Nama: {nama}\nTanggal Lahir: {tanggal_lahir}\nTempat Lahir: {tempat_lahir}\n\nDeskripsi:\n{deskripsi}\n"**, cetak isi untuk memastikan format teks sesuai.
- **file_path = f"{directory}/{nama.replace(' ', '_')}.txt"**, pastikan nama file dihasilkan dengan benar tanpa spasi.
- **with open(file_path, 'w', encoding='utf-8') as f: f.write(isi)**, cek apakah file penulis berhasil dibuat dan isinya lengkap.
- **def main_scraper(url, directory)**, fungsi dipanggil di akhir kode, url dan directory sudah ditentukan.
- **fungsi.create_directory(directory)**, periksa apakah folder DataScraping/ otomatis dibuat jika belum ada.

c. Hasil Uji Coba



```

DataScraping
├── Albert_Einstein.txt
├── C.S._Lewis.txt
├── Elie_Wiesel.txt
├── George_Eliot.txt
├── Helen_Keller.txt
├── J.K._Rowling.txt
├── Marilyn_Monroe.txt
├── Martin_Luther_King_...
├── quotes.txt
└── Thomas_A_Edison.txt

Quote: "There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."
Author: Albert Einstein
Detail URL: https://quotes.toscrape.com/author/Albert-Einstein

Quote: "Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than absolutely boring."
Author: Marilyn Monroe
Detail URL: https://quotes.toscrape.com/author/Marilyn-Monroe

Quote: "I have not failed. I've just found 10,000 ways that won't work."
Author: Thomas A. Edison
Detail URL: https://quotes.toscrape.com/author/Thomas-A-Edison

Quote: "This life is what you make it. No matter what, you're going to mess up sometimes, it's a universal truth. But the good part is you get to
it up. Girls will be your friends - they'll act like it anyway. But just remember, some come, some go. The ones that stay with you through everyth
ends. Don't let go of them. Also remember, sisters make the best friends in the world. As for lovers, well, they'll come and go too. And baby, I h
tually pretty much all of them are going to break your heart, but you can't give up because if you give up, you'll never find your soulmate. You'll
you whole and that goes for everything. Just because you fail once, doesn't mean you're gonna fail at everything. Keep trying, hold on, and alway
rself, because if you don't, then who will, sweetie? So keep your head high, keep your chin up, and most importantly, keep smiling, because life's
o much to smile about."
Author: Marilyn Monroe
Detail URL: https://quotes.toscrape.com/author/Marilyn-Monroe

Quote: "The opposite of love is not hate, it's indifference. The opposite of art is not ugliness, it's indifference. The opposite of faith is not
he opposite of life is not death, it's indifference."
Author: Elie Wiesel
Detail URL: https://quotes.toscrape.com/author/Elie-Wiesel

Quote: "To the well-organized mind, death is but the next great adventure."
Author: J.K. Rowling
Detail URL: https://quotes.toscrape.com/author/J-K-Rowling

Quote: "It is never too late to be what you might have been."
```

Gambar 5.2 Hasil Scraping dari halaman books.toscrape

- Hasil scraping menunjukkan bahwa program berhasil menampilkan kutipan, nama penulis, dan link detail dari halaman tag *inspirational*. Data ditampilkan rapi di terminal dan menunjukkan bahwa proses scraping level pertama berjalan dengan baik tanpa error.

d. Analisa Hasil

Program uji coba scraping gambar berjalan dengan sangat baik. Semua komponen mulai dari akses web, parsing HTML, pengambilan URL, hingga penyimpanan file telah diimplementasikan dengan benar. Folder hasil_gambar terisi dengan file gambar yang sesuai, menandakan bahwa program dapat digunakan sebagai dasar scraping gambar dari situs lain yang serupa.

5.5 Kesimpulan

5.5.1 Kesimpulan Percobaan 1

Mahasiswa telah melakukan percobaan 1, program berhasil melakukan proses web scraping pada halaman utama situs *Quotes to Scrape*. Program mampu mengekstrak kutipan, nama penulis, dan tautan ke halaman detail penulis dengan tepat. Seluruh data yang diperoleh ditampilkan secara rapi di terminal dan disimpan dalam file teks tanpa error. Hal ini menunjukkan bahwa fungsi `main_scraper()` telah berjalan sesuai harapan, serta proses parsing dan penyimpanan data telah berhasil dilakukan dengan benar.

5.5.2 Kesimpulan Percobaan 2

Mahasiswa telah berhasil melakukan percobaan 3, program berhasil melakukan web scraping pada halaman tag *inspirational* dari situs *quotes.toscrape* dengan dua tingkat kedalaman URL. Program tidak hanya menampilkan kutipan dan penulis dari halaman utama, tetapi juga mengambil informasi detail penulis serta kutipan tambahan berdasarkan tag yang dimiliki penulis tersebut. Semua data berhasil disimpan dalam file teks dengan struktur yang rapi, tanpa mengalami error saat pengambilan atau penulisan data. Hal ini menunjukkan bahwa program mampu menelusuri dan mengambil data dari sub-link secara menyeluruh sesuai dengan tujuan scraping tingkat lanjut.

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. Richardson, L. (2023). *Beautiful Soup Documentation*. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
3. Kenneth Reitz, *Requests Documentation*, 2015. <https://docs.python-requests.org>
4. Prasad, K. et al. (2018). "Web Scraping Techniques and Applications", *International Journal of Computer Applications*, Vol. 182, No. 10.