

BAB 4

4.1 Capaian Praktikum Pertemuan 4

Pada praktikum pertemuan ke-4 ini mahasiswa diharapkan mampu memahami ketentuan dan etika dalam melakukan scraping, khususnya terkait pengambilan gambar dari internet. Mahasiswa juga diharapkan dapat mengimplementasikan penggunaan library BeautifulSoup dan requests untuk mengambil URL gambar dari sebuah website, serta mampu mengambil file gambar tersebut dan menyimpannya ke dalam direktori yang telah ditentukan secara tepat.

4.2 Indikator Capaian

- Mahasiswa tidak melanggar aturan norma dan etika dalam pengambilan gambar.
- Mahasiswa melakukan scraping URL gambar dengan menggunakan BeautifulSoup dan Request.
- Mahasiswa melakukan penyimpanan gambar dengan menggunakan BeautifulSoup dan Request.
- Mahasiswa melakukan penyimpanan file gambar pada direktori yang ditentukan.
- Mahasiswa melakukan analisa website yang akan di scraping.

4.3 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, K. (2021), Requests merupakan library HTTP untuk Python yang digunakan untuk mengirim permintaan HTTP ke server. Menurut Reitz (2021), requests adalah library paling sederhana dan elegan untuk melakukan komunikasi client-server dalam Python.

Menurut Auerbach (2012), scraping harus memperhatikan aspek legalitas dan etika, terutama ketika mengambil data atau konten dari situs web yang dilindungi hak cipta. Scraper harus membaca robots.txt dan memperhatikan izin penggunaan data.

4.4 Pelaksanaan Praktikum

4.4.1 Percobaan Pertama

Pada percobaan pertama ini, dilakukan untuk menguji kemampuan program dalam mengambil gambar dari situs dongeng ceritarakyat.com dengan memanfaatkan atribut data-lazy-src, data-src, dan src pada tag . Program berhasil mendeteksi dan mengumpulkan URL gambar yang valid, lalu menyimpannya ke dalam folder bernama gambar_dongeng yang dibuat secara otomatis di direktori kerja. Hasil percobaan menunjukkan bahwa program dapat berjalan dengan baik dalam mengenali elemen gambar dari berbagai atribut dan menyimpan file secara terstruktur tanpa error.

a. Script / Setting Program

Main.py

```
import os
import requests
from bs4 import BeautifulSoup

# URL target
url = 'https://dongengceritarakyat.com/'
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')

# Ambil semua gambar dari elemen <img>
images = []
for img in soup.find_all('img'):
    img_url = img.get('data-lazy-src') or img.get('data-src') or img.get('src')
```

```

if img_url and img_url.endswith(('.jpg', '.png', '.gif')):
    images.append(img_url)

print(f"Jumlah gambar ditemukan: {len(images)}")
print("Contoh URL gambar:", images[:3])

# ☑ Ganti folder penyimpanan di sini
save_path = os.path.join(os.getcwd(), 'gambar_dongeng')
if not os.path.exists(save_path):
    os.makedirs(save_path)

# Download dan simpan gambar
for img in images:
    try:
        response = requests.get(img)
        filename = os.path.basename(img)
        filepath = os.path.join(save_path, filename)
        with open(filepath, 'wb') as f:
            f.write(response.content)
        print(f"{filename} berhasil disimpan di {save_path}")
    except Exception as e:
        print(f"Gagal menyimpan {img}: {e}")

```

b. Langkah Uji Coba

- **BeautifulSoup**: Untuk mem-parsing dan mengekstrak elemen HTML.
- **requests**: Untuk mengirim permintaan HTTP (GET) ke situs web.
- **url = 'https://dongengceritarakyat.com/'**, menyimpan alamat situs target.
- **requests.get(url)**: mengambil konten halaman web.
- **soup = BeautifulSoup(page.content, 'html.parser')**, membaca isi HTML dari halaman yang diambil agar bisa diolah.
- **for img in soup.find_all('img')**: menemukan semua tag dalam halaman.
- **img.get(...)**: mengambil nilai atribut gambar (**data-lazy-src**, **data-src**, atau **src**).
- **endswith(...)**: memastikan hanya gambar dengan format .jpg, .png, atau .gif yang diambil.
- **images.append(full_url)**: Menyimpan URL gambar ke dalam list images.
- **print(f"Jumlah gambar ditemukan: {len(images)}")**, menampilkan jumlah gambar yang ditemukan.

- `print("Contoh URL gambar:", images[:3])`, menampilkan 3 URL gambar pertama sebagai contoh hasil scraping.
- `os.getcwd()`: mengambil direktori saat ini.
- `os.path.join(...)`: menggabungkan path direktori dengan nama folder `gambar_dongeng`.
- `requests.get(img)`: mengambil konten gambar dari URL.
- `os.path.basename(img)`: mengambil nama file dari URL gambar.
- `open(..., 'wb')`: membuka file dalam mode tulis biner (wb) untuk menyimpan data gambar.
- `f.write(...)`: menulis isi gambar ke file.
- `print(...)`: menampilkan notifikasi bahwa gambar telah berhasil disimpan.
- `print(f"Gagal menyimpan {img}: {e}")`, menampilkan pesan error agar pengguna tahu gambar mana yang gagal diproses.

c. Hasil Uji Coba

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4 & "C:/...
Jumlah gambar ditemukan: 0
Contoh URL gambar: []
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4

```

Gambar 4.1 Hasil Scraping dari halaman dongeng cerita rakyat

- Program berhasil dijalankan, namun tidak menemukan gambar karena struktur HTML halaman dongengceritarakyat.com tidak sesuai dengan metode scraping yang digunakan. Diperlukan analisa lebih lanjut terhadap struktur `` atau beralih ke situs yang lebih terbuka untuk scraping.

d. Analisa Hasil

Hasil uji coba menunjukkan bahwa meskipun program berhasil mengakses situs dongengceritarakyat.com, tidak ada gambar yang berhasil ditemukan atau disimpan. Hal ini disebabkan karena elemen pada halaman tersebut kemungkinan tidak menggunakan atribut standar seperti src, data-src, atau data-lazy-src, atau gambar dimuat secara dinamis menggunakan JavaScript yang tidak dapat ditangani oleh requests dan BeautifulSoup.

4.4.2 Percobaan Kedua

Pada percobaan kedua ini, mahasiswa diperintahkan untuk melakukan percobaan scraping gambar dari website menggunakan library requests dan BeautifulSoup, di mana program mengambil URL gambar dari elemen dan menyimpannya secara otomatis ke dalam folder yang telah ditentukan.

a. Script / Setting Program

Tugas.py

```
import os
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin

# Ganti dengan URL baru
url = 'https://books.toscrape.com/'
page = requests.get(url)
soup = BeautifulSoup(page.content, 'html.parser')

images = []
for img in soup.find_all('img'):
    img_url = img.get('src')
    if img_url and img_url.endswith(('.jpg', '.png', '.gif')):
        # Gabungkan URL relatif jadi absolut
        full_url = urljoin(url, img_url)
        images.append(full_url)

print(f"Jumlah gambar ditemukan: {len(images)}")
print("Contoh URL gambar:", images[:3])
# Simpan ke folder hasil_gambar
save_path = os.path.join(os.getcwd(), 'hasil_gambar')
if not os.path.exists(save_path):
    os.makedirs(save_path)

for img in images:
```

```

try:
    response = requests.get(img)
    filename = os.path.basename(img)
    filepath = os.path.join(save_path, filename)

    with open(filepath, 'wb') as f:
        f.write(response.content)

    print(f"{filename} berhasil disimpan di {save_path}")
except Exception as e:
    print(f"Gagal menyimpan {img}: {e}")

```

b. Langkah Uji Coba

- **BeautifulSoup**: Untuk mem-parsing dan mengekstrak elemen HTML.
- **requests**: Untuk mengirim permintaan HTTP (GET) ke situs web.
- **from urllib.parse import urljoin**, untuk menggabungkan URL relatif menjadi URL absolut.
- **url = 'https://books.toscrape.com/'**, situs web yang dituju.
- **BeautifulSoup(page.content, 'html.parser')**: Mem-parsing HTML dari halaman untuk diolah.
- **soup.find_all('img')**: Menemukan semua elemen di halaman.
- **img.get('src')**: Mengambil atribut src dari tag , yaitu URL gambar.
- **endswith(...)**: Memastikan hanya file gambar yang diambil.
- **urljoin(url, img_url)**: Menggabungkan URL relatif (img_url) dengan URL utama (url) agar menjadi URL lengkap.
- **images.append(full_url)**: Menyimpan URL gambar ke dalam list images.
- **print(f"Jumlah gambar ditemukan: {len(images)}")**, menampilkan jumlah gambar yang ditemukan.
- **print("Contoh URL gambar:", images[:3])**, menampilkan 3 URL gambar pertama sebagai contoh hasil scraping.
- **os.getcwd()**: mengambil direktori saat ini.
- **os.path.join(...)**: menggabungkan path folder hasil gambar.
- **requests.get(img)**: mengambil konten gambar dari URL.

c. Hasil Uji Coba

```
PS C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4 & "C:/Program Files/Python312/python.exe" "c:/stikom/smt 4\Praktikum Data Scraping\pertemuan4\scraping.py"
Jumlah gambar ditemukan: 20
Contoh URL gambar: ['https://books.toscrape.com/media/cache/2c/da/2cadad67c44b002e7ead0cc35693c0e8b.jpg', 'https://books.toscrape.com/media/cache/3e/ef/3eeff99c9d9aef34639f510662022830.jpg']
2cadad67c44b002e7ead0cc35693c0e8b.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
260c6ae16bce31c8f8c95dadd9f4a1c.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
3eeff99c9d9aef34639f510662022830.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
3251cf3a3412f5f3f339e42cac2134093.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
bea5697f2534a2f86a3ef27b5a8c12a6.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
68339b4c9bc034267e1da61ab3b34f8.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
92274a95b7c251fea59a2b8a78275ab4.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
3d54940e57e662cddd1f3ff0e0c78cc64.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
66883b91f6804b2323c8369331cb7dd1.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
5846057e2802268153befffd352b06c.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
bef44da28c98f905a3ebec0b87be8530.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
94b1b8b244bce9677c2f29ccc890d4d2.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
81c4a973364e17d01f217e1188253d5e.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
54607fe8945897cdcced044103b10b6.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
553310a7162dfbc2c6d19a84da0df9e1.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
0ea3aef48557576e1a85ba7eфеа8cb7.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
94b1b8b244bce9677c2f29ccc890d4d2.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
81c4a973364e17d01f217e1188253d5e.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
54607fe8945897cdcced044103b10b6.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
553310a7162dfbc2c6d19a84da0df9e1.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
94b1b8b244bce9677c2f29ccc890d4d2.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
81c4a973364e17d01f217e1188253d5e.jpg berhasil disimpan di C:\stikom\smt 4\Praktikum Data Scraping\pertemuan4\hasil_gambar
```

Gambar 4.2 Hasil Scraping dari halaman books.toscrape

- Hasil scraping berjalan dengan sukses, di mana website target memberikan respon yang sesuai sehingga seluruh gambar berhasil dikumpulkan dan disimpan ke dalam folder yang telah ditentukan. Program ini menggunakan metode urljoin() untuk menggabungkan URL relatif menjadi URL absolut, sehingga gambar dapat diakses dan diunduh dengan benar. Selain itu, program juga menyaring file yang diambil berdasarkan ekstensi gambar seperti .jpg, .png, dan .gif, sehingga hanya file yang valid yang diproses dan disimpan.

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.

4.5 Kesimpulan

4.5.1 Kesimpulan Percobaan 1

Mahasiswa telah melakukan percobaan, bahwa proses scraping gambar pada situs dongengceritarakyat.com belum berhasil mendeteksi dan menyimpan gambar karena tidak ditemukan URL gambar yang sesuai dalam struktur HTML-nya. Hal ini menunjukkan bahwa tidak semua situs dapat di-scrape dengan metode HTML statis menggunakan requests dan BeautifulSoup, terutama jika elemen gambar dimuat secara dinamis melalui JavaScript atau menggunakan atribut khusus yang tidak umum. Oleh karena itu, diperlukan analisa lebih dalam terhadap struktur website atau penggunaan tools scraping yang mendukung pemrosesan dinamis seperti Selenium.

4.5.2 Kesimpulan Percobaan 2

Mahasiswa telah berhasil melakukan percobaan scraping menggunakan situs books.toscrape.com yang menunjukkan bahwa program berhasil dijalankan dengan baik dan sesuai harapan. Program mampu mengakses situs, mengambil elemen , mengekstrak URL gambar menggunakan atribut src, dan menggabungkannya menjadi URL absolut dengan urljoin. Seluruh gambar yang ditemukan kemudian berhasil disimpan ke dalam folder hasil_gambar tanpa error. Percobaan ini membuktikan bahwa teknik scraping berbasis HTML statis dengan requests dan BeautifulSoup sangat efektif jika digunakan pada situs yang struktur HTML-nya sederhana dan tidak memiliki pembatasan scraping.

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. Reitz, K. (2021). *Requests: HTTP for Humans*. <https://docs.python-requests.org>
4. Auerbach, D. (2012). *Scraping By: How Web Scrapers Help and Hurt*. Slate Technology.
<https://slate.com/technology/2012/03/web-scraping-legal-issues.html>