

# **Laporan Praktikum**

“Membuat Website Biografi/CV dengan HTML dan Scraping Menggunakan Python”

Mata Kuliah: Proyek 1: Pengembangan Perangkat Lunak Desktop



Disusun oleh:

Nama : Arkan Ramadhan Nugraha

NIM : 241524033

Kelas : 1B-TI4

**POLITEKNIK NEGERI BANDUNG**

**2025**

# DAFTAR ISI

Laporan Praktikum.....	1
DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
BAB I PENDAHULUAN.....	1
I.1. Latar Belakang.....	1
I.2. Rumusan Masalah.....	1
I.3. Tujuan Praktikum.....	1
BAB II PEMBAHASAN.....	2
II.1. Sejarah dan Perkembangan Web.....	2
II.1.1. Web 1.0.....	2
II.1.2. Web 2.0.....	2
II.1.3. Web 3.0.....	2
II.2. Penemu Web dan Kontribusinya.....	3
II.3. Dasar-Dasar Struktur HTML.....	3
II.3.1. Document Metadata.....	3
II.3.2. Content Sections.....	4
II.3.3. Block Text Content.....	4
II.3.4. Lists.....	4
II.3.5. Preformatted Text.....	4
II.3.6. Inline Elements.....	5
II.3.7. Tag Gambar.....	5
II.3.8. Tabel.....	5
II.3.9. Formulir.....	5
II.4. Pembuatan Web Biografi Menggunakan HTML dan CSS dan Deploy ke Github Pages.....	6
II.4.1. Struktur Halaman HTML.....	6
II.4.2. Penataan Tampilan dengan CSS.....	6
II.4.3. Pengujian dan Perbaikan.....	7
II.4.4. Deploy ke Github Pages.....	7
II.5. Data Scraping Menggunakan Python.....	7
II.5.1. Pengerjaan.....	7
II.5.2. Penjelasan Kode Scraping.....	8
BAB III HASIL DAN ANALISIS.....	10
III.1. Hasil.....	10
III.2. Analisis dan Tantangan.....	10
BAB IV KESIMPULAN DAN SARAN.....	11
IV.1. Kesimpulan.....	11
IV.2. Saran.....	11
LAMPIRAN.....	12

## DAFTAR GAMBAR

Gambar II.1: Document Metadata HTML.....	4
Gambar II.2: Block Text Content HTML.....	4
Gambar II.3: Teks Preformatted HTML.....	5
Gambar II.4: Contoh Inline Elements HTML.....	5
Gambar II.5: Tag Gambar HTML.....	5
Gambar II.6: Tag Tabel HTML.....	5
Gambar II.7: Formulir HTML.....	5
Gambar II.8: Python Scraping Mengambil Halaman HTML.....	8
Gambar II.9: Python Scraping Mengecek Status Request.....	8
Gambar II.10: Python Scraping Parsing Halaman Menggunakan BeautifulSoup.....	8
Gambar II.11: Python Scraping Mengambil Heading Pertama.....	8
Gambar II.12: Python Scraping Mengambil Heading h1 - h6.....	8
Gambar II.13: Python Scraping Mengambil Semua Paragraf dari Article dengan Id "tentang".....	9
Gambar II.14: Python Scraping Mengambil Semua Link dari elemen Nav.....	9
Gambar II.15: Python Scraping Mengambil Semua Input dari Footer.....	9
Gambar II.16: Python Scraping Gagal Request.....	9

# BAB I PENDAHULUAN

## I.1. Latar Belakang

Teknologi web telah berkembang dari media statis di Web 1.0 menjadi lingkungan interaktif dengan AI dan desentralisasi di Web 3.0. Pemahaman evolusi web, HTML, dan teknik pengambilan data (*data scraping*) sangat penting untuk pengembangan web modern.

## I.2. Rumusan Masalah

1. Apa perbedaan utama antara Web 1.0, Web 2.0, dan Web 3.0?
2. Bagaimana membangun dan mendesain website dengan HTML dan CSS?
3. Bagaimana cara melakukan web scraping menggunakan Python?
4. Bagaimana cara deploy website menggunakan GitHub Pages?

## I.3. Tujuan Praktikum

1. Memahami evolusi web dari Web 1.0 hingga Web 3.0.
2. Menerapkan cara membangun dan mendesain website dengan HTML dan CSS.
3. Menerapkan teknik web scraping menggunakan Python.
4. Memahami cara deploy website ke GitHub Pages.

## **BAB II PEMBAHASAN**

### **II.1. Sejarah dan Perkembangan Web**

#### **II.1.1. Web 1.0**

Web 1.0 adalah tahap awal World Wide Web (1989–2004) yang berfokus pada penyajian informasi secara statis. Pada era ini, sebagian besar pengguna hanya berperan sebagai konsumen konten, sementara pembuat konten masih terbatas. Situs pribadi banyak ditemukan di layanan seperti GeoCities atau Tripod, dan umumnya berisi halaman statis yang jarang diperbarui. Ciri-ciri utama Web 1.0 meliputi:

1. Halaman web statis tanpa banyak interaktivitas.
2. Tidak ada fitur media sosial atau partisipasi pengguna.
3. Dibangun dengan HTML sederhana tanpa pemrograman dinamis.
4. Konten hanya bisa dibaca oleh pengguna (read-only).

Karena keterbatasannya, Web 1.0 lebih menyerupai kumpulan dokumen yang diakses secara online daripada platform yang memungkinkan interaksi pengguna.

#### **II.1.2. Web 2.0**

Web 2.0 adalah evolusi dari Web 1.0 yang menekankan partisipasi pengguna, interaktivitas, dan kolaborasi. Dikenalkan pertama kali oleh Darcy DiNucci pada 1999 dan dipopulerkan oleh Tim O'Reilly pada 2004, istilah ini menggambarkan perubahan dalam cara pengguna berinteraksi dengan situs web. Web 2.0 memungkinkan pengguna untuk berkolaborasi dan berinteraksi, tidak hanya sebagai konsumen konten, tetapi juga sebagai pembuat konten. Ciri-ciri utama Web 2.0 adalah:

1. Konten yang dapat dibuat dan dibagikan oleh pengguna (read-write).
2. Adanya media sosial dan komunitas online.
3. Penggunaan teknologi AJAX untuk meningkatkan pengalaman pengguna.
4. Aplikasi berbasis web seperti Google Docs dan Facebook.

Dengan fitur seperti situs media sosial, blog, dan video sharing, Web 2.0 mengubah web menjadi platform yang lebih dinamis dan interaktif, memungkinkan kolaborasi global antar pengguna.

#### **II.1.3. Web 3.0**

Web 3.0, atau Web Semantik, adalah evolusi dari World Wide Web yang mengintegrasikan teknologi kecerdasan buatan dan blockchain. Web ini bertujuan

untuk menciptakan interaksi yang lebih cerdas antara pengguna dan sistem dengan memanfaatkan data yang lebih terstruktur dan dapat dimengerti oleh mesin. Dibandingkan dengan Web 2.0, Web 3.0 menekankan desentralisasi menggunakan teknologi blockchain, memungkinkan pengguna untuk memiliki kontrol lebih besar atas data mereka. Ciri-ciri utama Web 3.0 adalah:

1. Data yang lebih terstruktur dan dimengerti oleh mesin.
2. Desentralisasi menggunakan teknologi blockchain.
3. Penggunaan AI untuk menyaring dan menyesuaikan informasi.
4. Interaksi yang lebih cerdas antara pengguna dan sistem.

Namun, kritik muncul terkait penguasaan teknologi oleh beberapa pihak tertentu dan potensi hilangnya privasi akibat pengumpulan data yang lebih luas.

## **II.2. Penemu Web dan Kontribusinya**

Sir Timothy John Berners-Lee (lahir 8 Juni 1955), atau dikenal sebagai TimBL, adalah seorang ilmuwan komputer asal Inggris yang dikenal sebagai penemu World Wide Web. Pada tahun 1989, saat bekerja di CERN, ia mengembangkan tiga teknologi utama yang membentuk dasar web:

1. HTML (HyperText Markup Language) – bahasa markup untuk membuat halaman web.
2. HTTP (HyperText Transfer Protocol) – protokol komunikasi antara server dan klien.
3. URL (Uniform Resource Locator) – sistem pengalamatan unik untuk halaman web.

Motivasi Berners-Lee berasal dari kebutuhan untuk menyimpan, memperbarui, dan menemukan dokumen serta data di CERN yang besar dan terus berubah, serta mendistribusikannya ke kolaborator di luar CERN. Inspirasi untuk desain webnya juga datang dari model hiper teks Ted Nelson yang memungkinkan dokumen terhubung secara bebas melalui hyperlink.

Saat ini, Berners-Lee adalah seorang profesor riset di Universitas Oxford dan profesor emeritus di Massachusetts Institute of Technology (MIT).

## **II.3. Dasar-Dasar Struktur HTML**

### **II.3.1. Document Metadata**

Metadata ditempatkan dalam tag `<head>`, mencakup informasi seperti judul halaman dan link ke CSS eksternal.

```

11  <!doctype html>
10  <html lang="id">
9    <head>
8      <title>Intro to HTML</title>
7      <link rel="stylesheet" href="./style.css" />
6      <style>
5        hr {
4          margin: 40px;
3        }
2      </style>
1    </head>
12  </html>

```

Gambar II.1: Document Metadata HTML

### II.3.2. Content Sections

Konten utama berada dalam `<body>`, menggunakan elemen semantik seperti `<header>`, `<main>`, `<article>`, `<aside>`, dan `<footer>`.

### II.3.3. Block Text Content

Elemen seperti `<div>` dan `<p>` digunakan untuk mengelompokkan teks.

```

<body>
|  <div><p>Text here</p></div>
</body>

```

Gambar II.2: Block Text Content HTML

### II.3.4. Lists

1. Ordered List: `<ol>`.
2. Unordered List: `<ul>`.

### II.3.5. Preformatted Text

Tag `<pre>` menampilkan teks dengan format asli.

```
<pre>Code block</pre>
```

Gambar II.3: Teks Preformatted HTML

### II.3.6. Inline Elements

Elemen seperti `<a>`, `<span>`, `<b>`, `<i>`, `<code>` digunakan untuk memformat teks dalam satu baris.

```
<a href="#">Link</a>
```

Gambar II.4: Contoh Inline Elements HTML

### II.3.7. Tag Gambar

Tag `<img>` menyisipkan gambar, dengan atribut `src` dan `alt`.

```

```

Gambar II.5: Tag Gambar HTML

### II.3.8. Tabel

Tabel dibuat dengan `<table>`, `<tr>`, `<th>`, `<td>`.

```
<table><tr><th>Header</th></tr></table>
```

Gambar II.6: Tag Tabel HTML

### II.3.9. Formulir

Formulir menggunakan `<form>`, dengan berbagai elemen input seperti `<input>`, `<select>`, dan `<button>`.

```
<form><input type="text" /><button>Submit</button></form>
```

Gambar II.7: Formulir HTML



## **II.4. Pembuatan Web Biografi Menggunakan HTML dan CSS dan Deploy ke Github Pages**

Pada pembuatan web biografi ini, digunakan dua teknologi utama, yaitu HTML (HyperText Markup Language) dan CSS (Cascading Style Sheets). HTML digunakan untuk struktur dan konten halaman web, sementara CSS digunakan untuk mempercantik tampilan dan tata letak elemen-elemen pada halaman. Lalu website dideploy ke GitHub Pages, supaya dapat diakses di perangkat apa saja. Link panduan praktikum ada di Lampiran 3. Link inspirasi website ada di lampiran 4.

### **II.4.1. Struktur Halaman HTML**

Pada tahap pertama, pembuatan struktur halaman dilakukan dengan HTML. Struktur dasar dari halaman web biografi ini terdiri dari beberapa bagian penting, yaitu:

1. Navigation Bar (Navbar), yaitu menu navigasi yang memudahkan pengguna berpindah antar bagian halaman. Navbar saya bagi jadi dua bagian, bagian kanan untuk menu-menu nya, dan bagian kiri untuk logo website, logo website hanya berupa teks.
2. Main Content, yaitu bagian utama yang menyajikan informasi biografi/CV, terdiri dari bagian tentang, bagian keterampilan, bagian pengalaman, dan bagian fakta menarik.
3. Footer, berisi informasi hak cipta dan form kontak.

File tersedia di Lampiran 1

### **II.4.2. Penataan Tampilan dengan CSS**

Setelah struktur halaman selesai, CSS digunakan untuk memberikan gaya pada tampilan dan tata letak agar lebih menarik dan responsif. Berikut beberapa pengaturan yang diterapkan:

1. Pemilihan warna. Saya memilih warna merah sebagai warna utama dan hitam sebagai warna dasar.
2. Tata letak Navbar dibuat horizontal, agar lebih rapih. Menggunakan flexbox. Navbar diberi warna utama.
3. Hover effect ditambahkan pada link navigasi.
4. Main content diberi background gambar. Setiap bagian artikel nya diberi warna dasar dengan tingkat ketembusan/opacity 0.5, diberi border tipis dengan warna utama, dan dirapihkan menggunakan flexbox.
5. Footer diberi warna utama, dan dirapihkan menggunakan flexbox.

File tersedia di Lampiran 1

### II.4.3. Pengujian dan Perbaikan

Setelah implementasi HTML dan CSS, dilakukan pengujian di berbagai browser untuk memastikan tata halaman terlihat rapih. Beberapa hal yang diperbaiki dan ditambahkan:

1. Navbar diperbaiki agar tetap terlihat saat di-scroll. Solusinya menggunakan “position: fixed;” pada style navbar
2. Halaman dapat di-scroll ke-kanan, dikarenakan navbar yang melebihi cakupan layar. Solusinya, padding di navbar dihilangkan, dan ditambahkan margin di bagian kiri dan bagian kanan navbar.

### II.4.4. Deploy ke Github Pages

Setelah website selesai dibuat, langkah terakhir adalah mendeploy web ke GitHub Pages agar bisa diakses secara online. Berikut langkah-langkahnya:

1. Buat repository baru
2. Upload file HTML, CSS, dan aset-aset yang digunakan ke repository tersebut
3. Commit changes
4. Buka repository di GitHub
5. Klik Settings > Pages
6. Pada bagian Branch, pilih “main” dan folder “/ (root)”
7. Klik save
8. Tunggu beberapa menit, lalu akses web melalui <https://username.github.io/repository-name>
9. Agar link berubah menjadi <https://username.github.io>, ubah nama repository menjadi username.github.io

Link hasil deploy ada di Lampiran 2

## II.5. Data Scraping Menggunakan Python

Untuk mengambil data biografi secara otomatis, dapat digunakan Python dengan library BeautifulSoup. Berikut adalah contoh kode scraping biografi dari Wikipedia dan menyimpannya ke dalam file `scraping.txt`

### II.5.1. Pengerjaan

Salin kode Data Scraping Python dari Lampiran 3 ke file baru `scrape.py`, dan ganti nilai variabel url menjadi link Github Pages yang sudah dideploy. Lalu jalankan kode. Hasil ada di Lapiran 1.

## II.5.2. Penjelasan Kode Scraping

### 1. Mengambil Halaman HTML

```
3 # URL website yang telah kamu buat (jika dijalankan secara lokal, gunakan localhost, misalnya http://localhost:8000/index.html)
2 url = "http://127.0.0.1:5500/index.html"
1
7 # Mengambil konten halaman menggunakan requests
1 response = requests.get(url)
```

Gambar II.8: Python Scraping Mengambil Halaman HTML

`requests.get(url)` mengirim HTTP GET Request ke url

### 2. Mengecek Apakah Halaman Berhasil Diambil

```
if response.status_code == 200:
```

Gambar II.9: Python Scraping Mengecek Status Request

HTTP Status Code 200 artinya request/permintaan berhasil. Jika berhasil, kode lanjut ke parsing HTML

### 3. Parsing Halaman Menggunakan BeautifulSoup

```
2 # Parsing HTML menggunakan BeautifulSoup
1 soup = BeautifulSoup(response.text, "html.parser")
```

Gambar II.10: Python Scraping Parsing Halaman Menggunakan BeautifulSoup

Dalam linguistik komputasi, parsing merujuk pada analisis formal oleh komputer terhadap sebuah kalimat atau rangkaian kata lain menjadi struktur yang lebih terorganisir, lebih mudah dipahami, dan lebih mudah diolah oleh komputer. `response.text` berisi HTML dari halaman yang diambil. BeautifulSoup digunakan untuk memarsing HTML menjadi lebih mudah untuk mencari sesuatu.

### 4. Mengambil Data dari HTML

#### 1. Mengambil Heading Utama

```
1 # Contoh 1: Mengambil heading utama (h1)
2 h1 = soup.find("h1")
3 print("Heading Utama:", h1.get_text() if h1 else "Tidak ditemukan")
```

Gambar II.11: Python Scraping Mengambil Heading Pertama

`soup.find("h1")` mencari elemen h1 pertama dalam halaman

#### 2. Mengambil Semua Heading (h1 – h6)

```
5 # Contoh 2: Mengambil semua heading (h1 hingga h6)
6 for level in range(1, 7):
7     for heading in soup.find_all(f"h{level}"):
8         print(f"H{level}:", heading.get_text())
```

Gambar II.12: Python Scraping Mengambil Heading h1 - h6

Loop dari h1 – h6 untuk mencari semua heading. `soup.find_all(f"h{level}")` mengambil semua elemen dengan tag heading

### 3. Mengambil Semua Paragraf dalam <article id="tentang">

```
10 # Contoh 3: Mengambil semua paragraf dalam <article id="tentang">
11 tentang = soup.find("article", id="tentang")
12 if tentang:
13     paragraphs = tentang.find_all("p")
14     for idx, p in enumerate(paragraphs, 1):
15         print(f"Paragraf {idx} dalam Tentang:", p.get_text())
```

Gambar II.13: Python Scraping Mengambil Semua Paragraf dari Article dengan Id "tentang"

soup.find("article", id="tentang") mencari elemen article dengan id tentang. Jika ditemukan, semua elemen paragraf dicetak.

### 4. Mengambil Semua Link dalam <nav>

```
17 # Contoh 4: Mengambil semua link dalam navigasi
18 nav = soup.find("nav")
19 if nav:
20     links = nav.find_all("a")
21     for link in links:
22         print("Link Navigasi:", link.get("href"), "-", link.get_text())
```

Gambar II.14: Python Scraping Mengambil Semua Link dari elemen Nav

Mencari elemen nav, lalu mengambil semua link a dalam nav dan dicetak href (URL tujuan) dan teks dalam tag a

### 5. Mengambil Semua Input Form dalam <footer>

```
8 footer = soup.find("footer")
7 if footer:
6     inputs = footer.find_all("input")
5     for inp in inputs:
4         tipe = inp.get("type")
3         nama = inp.get("name")
2         print("Input Form:", nama, "Tipe:", tipe)
```

Gambar II.15: Python Scraping Mengambil Semua Input dari Footer

Mencari elemen footer, lalu mengambil semua input di dalam footer, lalu mencetak tipe dan nama input.

### 6. Jika Gagal Mengambil Halaman

```
else:
    print("Gagal mengambil halaman. Status Code:", response.status_code)
```

Gambar II.16: Python Scraping Gagal Request

Mencetak pesan error dan status code request nya.

## **BAB III HASIL DAN ANALISIS**

### **III.1. Hasil**

Dari praktikum, ditemukan bahwa:

1. Web 1.0 bersifat statis, Web 2.0 bersifat interaktif, dan Web 3.0 menggunakan AI serta blockchain.
2. Penemu World Wide Web adalah Sir Timothy John Berners-Lee
3. HTML digunakan untuk struktur dan konten halaman web, sementara CSS digunakan untuk mempercantik tampilan dan tata letak elemen-elemen pada halaman.
4. Deploy website artinya mengunggah website agar dapat diakses oleh pengguna melalui internet
5. Web Scraping dengan Python memungkinkan pengambilan data secara otomatis dan efisien.

### **III.2. Analisis dan Tantangan**

1. Terkadang tampilan website tidak seperti yang diinginkan. Untuk memudahkan perbaikan, fitur Inspect di browser sangat membantu.

## **BAB IV KESIMPULAN DAN SARAN**

### **IV.1. Kesimpulan**

Dari hasil penelitian dan praktik yang telah dilakukan, dapat disimpulkan bahwa perkembangan web dari Web 1.0 hingga Web 3.0 memberikan dampak besar dalam dunia digital. Struktur HTML menjadi dasar utama dalam pembuatan web, dan teknik web scraping sangat bermanfaat dalam ekstraksi data otomatis.

### **IV.2. Saran**

1. Pengembangan website biografi/CV dapat ditingkatkan dengan penggunaan JavaScript

# LAMPIRAN

1. Lampiran 1 – Repository Hasil Praktikum  
<https://github.com/Arney1/arney1.github.io>
2. Lampiran 2 – Link Hasil Deploy Website Biografi/CV  
<https://arney1.github.io>
3. Lampiran 3 – Link Panduan dan Lab Praktikum  
<https://github.com/kyeiki/proyek1-2025-lab-1/blob/main/lab1.MD>
4. Lampiran 4 – Link Website Inspirasi  
<https://elizabeth-tif.github.io/>