BAB 8

8.1 Capaian Praktikum Pertemuan 8

Pada praktikum pertemuan ke-8 ini mahasiswa diharapkan mampu memahami konsep dasar penggunaan routing pada Flask sebagai salah satu framework web berbasis Python. Selain itu, mahasiswa juga dituntut untuk dapat menerapkan hasil scraping data dari berbagai sumber web ke dalam sistem routing Flask. Dengan demikian, mahasiswa tidak hanya memahami alur pengambilan data melalui scraping, tetapi juga mampu mengintegrasikan setiap elemen hasil scraping ke dalam tampilan web menggunakan routing yang tepat pada aplikasi Flask.

8.2 Indikator Capaian

- Mahasiswa membuat konsep routing pada flask dengan ketentuannya.
- Mahasiswa membuat website dengan template jinja untuk mendukung data yang ditampilkan dari scrape.
- Mahasiswa membuat routing dengan serta menerapkan disetiap routingnya hasil dari scrape.

8.3 Landasan Teori

Menurut Grinberg (2018) dalam bukunya *"Flask Web Development"*, routing adalah mekanisme yang digunakan untuk menghubungkan URL tertentu dengan fungsi (view function) dalam aplikasi Flask. Routing bertindak sebagai penghubung antara permintaan dari pengguna (request) dengan respons yang akan diberikan oleh server. Setiap route ditentukan menggunakan dekorator @app.route, yang mengarahkan URL ke fungsi Python tertentu.

Menurut Mitchell (2018) dalam bukunya *"Web Scraping with Python"*, web scraping adalah teknik untuk mengekstrak data dari situs web secara otomatis menggunakan program. Dalam Python, proses ini biasanya dilakukan dengan bantuan

library seperti requests untuk mengambil halaman web dan BeautifulSoup untuk mem-parsing struktur HTML. Web scraping banyak digunakan dalam pengumpulan data untuk analisis, riset, dan aplikasi web dinamis.

Sementara itu, menurut Sonmez & Dede (2020), integrasi data hasil scraping ke dalam aplikasi web dapat meningkatkan fungsionalitas dan dinamika konten situs. Dengan framework seperti Flask, hasil scraping dapat diolah dan ditampilkan secara real-time melalui template HTML menggunakan Jinja2.

8.4 Pelaksanaan Praktikum

8.4.1 Percobaan Pertama

Pada percobaan pertama mahasiswa menampilkan hasil scraping dari situs BolaSport melalui routing Flask. Data diambil menggunakan requests dan BeautifulSoup, lalu ditampilkan ke halaman web menggunakan template Jinja2.

a. Script / Setting Program

app.py

```
from flask import Flask, render_template, request
from main import (
    scrape_bolasport,
    scrape_detik_jatim,
    scrape_detik_jateng,
    scrape_detik_jabar,
    scrape_liputan6,
    scrape_cnnindonesia,
    scrape_detail_berita
)

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/bola-sport')
def bola_sport():
    data = scrape_bolasport()
    return render_template('bola-sport.html', data=data)

@app.route('/detik-jatim')
def detik_jatim():
    data = scrape_detik_jatim()
    return render_template('detik-jatim.html', data=data)
```

```
@app.route('/detik-jateng')
def detik_jateng():
    data = scrape_detik_jateng()
    return render_template('detik-jateng.html', data=data)

@app.route('/detik-jabar')
def detik_jabar():
    data = scrape_detik_jabar()
    return render_template('detik-jabar.html', data=data)

@app.route('/berita-gabungan')
def berita_gabungan():
    liputan6_data = scrape_liputan6()
    cnn_data = scrape_cnnindonesia()
        return   render_template('berita_gabungan.html',   liputan6=liputan6_data,
cnn=cnn_data)

@app.route('/berita')
def detail_berita():
    url = request.args.get('url')
    if not url:
        return 'URL berita tidak ditemukan', 400
    detail = scrape_detail_berita(url)
    return render_template('detail_berita.html', berita=detail)

if __name__ == '__main__':
    app.run(debug=True)
```

main.py

```
import requests

from bs4 import BeautifulSoup

def scrape_bolasport():

    url = 'https://www.bolasport.com/'

    res = requests.get(url)

    soup = BeautifulSoup(res.text, 'html.parser')

    return soup.find_all('div', class_='news-list__item clearfix')

def scrape_detik_jatim():

    url = 'https://www.detik.com/jatim'

    headers = {'User-Agent': 'Mozilla/5.0'}

    res = requests.get(url, headers=headers)

    soup = BeautifulSoup(res.text, 'html.parser')

    return soup.find_all('article')

def scrape_detik_jateng():
```

```
    url = 'https://www.detik.com/jateng'
    headers = {'User-Agent': 'Mozilla/5.0'}
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, 'html.parser')
    return soup.find_all('article')
def scrape_detik_jabar():
    url = 'https://www.detik.com/jabar'
    headers = {'User-Agent': 'Mozilla/5.0'}
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, 'html.parser')
    return soup.find_all('article')
def scrape_liputan6():
    url = 'https://www.liputan6.com/citizen6'
    headers = {'User-Agent': 'Mozilla/5.0'}
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, 'html.parser')
     articles = soup.find_all('article', class_='articles--iridescent-list--text-
item')
    data = []
    seen = set()
    for article in articles:
            link_tag = article.find('a', class_='articles--iridescent-list--text-
item__title-link')
         title_span = article.find('span', class_='articles--iridescent-list--text-
item__title-link-text')
        img_tag = article.find('img')
        if not link_tag or not title_span:
            continue
        title = title_span.get_text(strip=True)
        link = link_tag['href']
        image = ''
        if img_tag:
            image = (
```

```
                img_tag.get('data-src') or
                img_tag.get('src') or
                img_tag.get('data-original') or
                ''
            )
        if title not in seen:
            data.append({'title': title, 'link': link, 'image': image})
            seen.add(title)
    return data
def scrape_cnnindonesia():
    url = 'https://www.cnnindonesia.com/gaya-hidup'
    headers = {'User-Agent': 'Mozilla/5.0'}
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, 'html.parser')
    data = []
    anchors = soup.find_all('a', class_='flex group items-center gap-4')
    for anchor in anchors:
        link = anchor.get('href')
        img_tag = anchor.find('img')
        h2_tag = anchor.find('h2')
        if not link or not img_tag or not h2_tag:
            continue
        image = img_tag.get('src', '')
        title = h2_tag.get_text(strip=True)
        data.append({
            'title': title,
            'link': link,
            'image': image
        })
    return data
def scrape_detail_berita(url):
    headers = {'User-Agent': 'Mozilla/5.0'}
```

```
    res = requests.get(url, headers=headers)
    soup = BeautifulSoup(res.text, 'html.parser')

    # Liputan6
    title = soup.find('h1', class_='read-page--header--title')
    if title:
        title = title.get_text(strip=True)
        img_url = ''
             gallery_div = soup.find('div', class_='read-page--photo-gallery--item__content')
        if gallery_div:
            img_tag = gallery_div.find('img')
            if img_tag:
                img_url = img_tag.get('data-src') or img_tag.get('src') or ''
        content_div = soup.find('div', class_='article-content-body__item-page')
        content_html = str(content_div) if content_div else ''
        return {'title': title, 'image': img_url, 'content': content_html}

    # CNN Indonesia
      title = soup.find('h1', class_='mb-2 text-[32px] text-cnn_black font-merriweather')
    if title:
        title = title.get_text(strip=True)
        img_tag = soup.find('img', class_='w-full')
        img_url = img_tag.get('src', '') if img_tag else ''
         content_div = soup.find('div', class_='detail-text text-cnn_black text-sm grow min-w-0')
        content_html = str(content_div) if content_div else ''
        return {'title': title, 'image': img_url, 'content': content_html}
    return {'title': 'Berita tidak ditemukan', 'image': '', 'content': ''}
```

bola-sport.html

```
{% extends 'index.html' %}
```

```
{% block body %}
<h2>Bola Sport</h2>
<div class="row">
    {% for item in data %}
    <div class="col-3">
        {{ item.find('img')|safe }}
    </div>
    <div class="col-9">
        <a href="{{ item.find('a', {'class': 'news-list__link'})['href'] }}">
            <p class="fs-6">{{ item.find('img')['alt']|safe }}</p>
        </a>
    </div>
    {% endfor %}
</div>
{% endblock %}
```

detik-jabar.html

```
{% extends 'index.html' %}
{% block body %}
<h2>Detik Jabar</h2>
<div class="row">
    {% for item in data %}
    <div class="col-3">
        {% if item.find('img') and item.find('img').has_attr('src') %}
        <img src="{{ item.find('img')['src'] }}" class="img-fluid">
        {% endif %}
    </div>
    <div class="col-9">
        {% if item.find('a') and item.find('a').has_attr('href') %}
        <a href="{{ item.find('a')['href'] }}">
            <p>{{ item.find('h2').text.strip() if item.find('h2') else 'Tanpa Judul'
}}</p>
```

```
        </a>
        {% endif %}
    </div>
    {% endfor %}
</div>
{% endblock %}
```

detik-jateng.html

```
{% extends 'index.html' %}
{% block body %}
<h2>Detik Jateng</h2>
<div class="row">
    {% for item in data %}
    <div class="col-3">
        {% if item.find('img') and item.find('img').has_attr('src') %}
        <img src="{{ item.find('img')['src'] }}" class="img-fluid">
        {% endif %}
    </div>
    <div class="col-9">
        {% if item.find('a') and item.find('a').has_attr('href') %}
        <a href="{{ item.find('a')['href'] }}">
            <p>{{ item.find('h2').text.strip() if item.find('h2') else 'Tanpa Judul' }}</p>
        </a>
        {% endif %}
    </div>
    {% endfor %}
</div>
{% endblock %}
```

detik-jatim.html

```
{% extends 'index.html' %}
```

```
{% block body %}

<h2>Detik Jatim</h2>

<div class="row">

    {% for item in data %}

    <div class="col-3">

        {% if item.find('img') %}

        <img src="{{ item.find('img')['src'] }}" class="img-fluid">

        {% endif %}

    </div>

    <div class="col-9">

        <a href="{{ item.find('a')['href'] }}">

            <p>{{ item.find('h2').text.strip() if item.find('h2') else 'Tanpa Judul'
}}</p>

        </a>

    </div>

    {% endfor %}

</div>

{% endblock %}
```

berita_gabungan.html

```
<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

    <title>Berita Gabungan</title>

    <style>

        body {

            font-family: 'Segoe UI', sans-serif;

            background-color: #f9f9f9;

            padding: 20px;

        }

        h2 {
```

```
            color: #2c3e50;
        }
        .berita-container {
            display: flex;
            flex-wrap: wrap;
            gap: 20px;
        }
        .berita-card {
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 2px 4px rgba(0,0,0,0.1);
            width: 300px;
            overflow: hidden;
            transition: 0.3s;
        }
        .berita-card:hover {
            transform: scale(1.03);
        }
        .berita-card img {
            width: 100%;
            height: 180px;
            object-fit: cover;
        }
        .berita-card .content {
            padding: 15px;
        }
        .berita-card h4 {
            font-size: 18px;
            margin: 10px 0;
        }
        .berita-card a {
            text-decoration: none;
```

```
            color: #3498db;
        }
    </style>
</head>
<body>
    <h2>Berita dari Liputan6</h2>
    <div class="berita-container">
        {% for item in liputan6 %}
        <div class="berita-card">
            <img src="{{ item.image }}" alt="Gambar Berita">
            <div class="content">
                <h4>{{ item.title }}</h4>
                <a href="/berita?url={{ item.link }}">Baca Selengkapnya</a>
            </div>
        </div>
        {% endfor %}
    </div>

    <h2>Berita dari CNN Indonesia</h2>
    <div class="berita-container">
        {% for item in cnn %}
        <div class="berita-card">
            <img src="{{ item.image }}" alt="Gambar Berita">
            <div class="content">
                <h4>{{ item.title }}</h4>
                <a href="/berita?url={{ item.link }}">Baca Selengkapnya</a>
            </div>
        </div>
        {% endfor %}
    </div>
</body>
</html>
```

detail_berita.html

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>{{ berita.title }}</title>
    <style>
        body {
            font-family: 'Segoe UI', sans-serif;
            background-color: #f2f2f2;
            margin: 0;
            padding: 30px;
        }
        .container {
            max-width: 800px;
            margin: auto;
            background-color: #ffffff;
            border-radius: 10px;
            box-shadow: 0 4px 12px rgba(0,0,0,0.1);
            overflow: hidden;
        }
        .header-image {
            width: 100%;
            max-height: 400px;
            object-fit: cover;
        }
        .content {
            padding: 25px 30px;
        }
        .content h1 {
            font-size: 26px;
```

```
            color: #2c3e50;
            margin-bottom: 20px;
        }
        .content p {
            font-size: 16px;
            line-height: 1.8;
            color: #333;
            margin-bottom: 15px;
        }

        .back-button {
            display: inline-block;
            margin: 20px 30px;
            background-color: #3498db;
            color: white;
            padding: 10px 18px;
            text-decoration: none;
            border-radius: 6px;
            transition: background 0.3s;
        }
        .back-button:hover {
            background-color: #2a80b9;
        }
    </style>
</head>
<body>
    <div class="container">
        {% if berita.image %}
            <img src="{{ berita.image }}" alt="Gambar Berita" class="header-image">
        {% endif %}
        <div class="content">
            <h1>{{ berita.title }}</h1>
```

```
            {{ berita.content|safe }}
        </div>
    </div>
    <a class="back-button" href="/">← Kembali ke Halaman index</a>
</body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Berita Terkini</title>
                                                                                <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="/">Portal Berita</a>
      <div class="collapse navbar-collapse">
        <ul class="navbar-nav ms-auto">
                <li  class="nav-item"><a  class="nav-link"  href="/bola-sport">Bola
Sport</a></li>
              <li  class="nav-item"><a  class="nav-link"  href="/detik-jatim">Detik
Jatim</a></li>
             <li  class="nav-item"><a  class="nav-link"  href="/detik-jateng">Detik
Jateng</a></li>
              <li  class="nav-item"><a  class="nav-link"  href="/detik-jabar">Detik
Jabar</a></li>
           <li class="nav-item"><a class="nav-link" href="/berita-gabungan">Berita
Gabungan</a></li>
        </ul>
```

```
      </div>
    </div>
  </nav>

  <div class="container mt-4">
    {% block body %}{% endblock %}
  </div>
</body>
</html>
```

b. Langkah Uji Coba

- app.py

- **from flask import Flask,** mengimpor Flask (membuat instance aplikasi), **render_template** (untuk merender HTML dari folder templates), dan **request** (mengakses parameter request).
- **from main import (),** mengimpor semua fungsi scraping dari file main.py. Masing-masing fungsi bertugas mengambil data berita dari situs tertentu.
- **app = Flask(__name__),** membuat objek Flask utama bernama app. Ini adalah inti dari aplikasi web Flask.
- **@app.route('/') def home(),** ketika pengguna mengakses URL root (/), fungsi home() akan dipanggil dan merender index.html.
- **@app.route('/bola-sport') def bola_sport(),** mengakses URL /bola-sport akan memicu fungsi scrape_bolasport(), dan hasilnya dikirim ke template bola-sport.html.
- **@app.route('/detik-jatim')def detik_jatim(),** menyediakan data dari Detik Jatim untuk dirender di detik-jateng.html.
- **@app.route('/detik-jateng) def detik_jateng(),** menyediakan data dari Detik Jateng untuk dirender di detik-jateng.html.

- **@app.route('/detik-jabar)def detik_jabar(),** menyediakan data dari Detik Jabar untuk direnderdi detik-jateng.html.
- **@app.route('/berita-gabungan') def berita_gabungan(),** menampilkan berita gabungan dari dua sumber (Liputan6 & CNN). Masing-masing hasil scraping dikirim ke template berita_gabungan.html.
- **@app.route('/berita') def detail_berita(),** Endpoint /berita menerima parameter url melalui query string. Jika url ditemukan, maka fungsi scrape_detail_berita(url) dipanggil dan hasilnya ditampilkan di detail_berita.html.

- main.py

- **import requests,** untuk mengambil data HTML dari website. **from bs4 import BeautifulSoup,** untuk mem-parsing HTML dan mengekstrak informasi.
- **def scrape_bolasport(),** mengambil daftar berita dari BolaSport berdasarkan elemen div dengan class tertentu.
- **def scrape_detik_jatim(), def scrape_detik_jateng(), def scrape_detik_jabar(),** mengambil semua tag <article> dari halaman Detik wilayah (Jatim, Jateng, Jabar).
- **def scrape_liputan6(),** mengambil berita dari Liputan6. **for article in articles,** menghindari duplikat judul. **data.append({'title': ..., 'link': ..., 'image': ...}),** mengembalikan list dict: title, link, image.
- **def scrape_cnnindonesia(),** mengambil berita dari CNN Indonesia, bagian gaya hidup. **data.append({'title': ..., 'link': ..., 'image': ...}),** ambil data dari tag <a> dengan struktur gambar dan judul.
- **def scrape_detail_berita(url), if title: return {...},** menampilkan detail berita: Jika url dari Liputan6, ambil judul, gambar, dan isi. Jika url dari CNN, lakukan hal serupa. Jika tak cocok, kembalikan "berita tidak ditemukan".

- templates (bola-sport.html)

- **{% extends 'index.html' %},** template ini mewarisi struktur dari index.html (biasanya ada kerangka HTML dasar seperti <html>, <head>, <body>, dll).
- **{% block body %},** memulai blok konten bernama body, yang akan menggantikan blok body di index.html.
- **<h2>Bola Sport</h2>,** menampilkan judul “Bola Sport” di halaman.
- ** ,** membuat baris (row) Bootstrap sebagai kontainer untuk grid kolom.
- **{% for item in data %},** melakukan perulangan untuk setiap elemen dalam variabel data (berisi hasil scraping scrape_bolasport()).
- ** ,** membuat kolom selebar 3 (dari 12) untuk gambar berita.
- **{{ item.find('img')|safe }},** menampilkan elemen <img> dari objek item. Gunakan |safe agar HTML gambar dirender langsung, bukan sebagai teks.
- ** ,** kolom lebar 9 untuk judul berita.
- **<a href="{{ item.find('a', {'class': 'news-list__link'})['href'] }}">,** tautan ke berita lengkap, diambil dari elemen <a> dengan class news-list__link.
- ** {{ item.find('img')['alt']|safe }} ,** menampilkan teks alternatif dari gambar sebagai judul berita dalam paragraf ukuran kecil (fs-6 dari Bootstrap).
- **{% endfor %},** menutup blok body yang dimulai sebelumnya.

- templates (detik_jabar.html)

- **{% extends 'index.html' %},** template ini mewarisi dari index.html, artinya hanya mengisi bagian tertentu dari kerangka besar (<html>, <head>, dll).
- **{% block body %},** memulai blok bernama body yang akan menggantikan isi blok body di index.html.
- **<h2>Detik Jabar</h2>,** menampilkan judul halaman “Detik Jabar”.

- ** ,** memulai **baris (row)** layout Bootstrap untuk grid 12 kolom.
- **{% for item in data %},** perulangan untuk setiap elemen item dalam data (hasil dari scrape_detik_jabar()).
- ** ,** membuat kolom selebar 3 (untuk gambar berita).
- **{% if item.find('img') and item.find('img').has_attr('src') %},** cek apakah elemen <img> ada dan punya atribut src sebelum menampilkan gambar.
- **<img src="{{ item.find('img')['src'] }}" class="img-fluid">,** menampilkan gambar berita dengan Bootstrap img-fluid agar responsive.
- **{% endif %},** akhir kondisi if untuk gambar.
- ** ,** kolom sebesar 9 untuk menampilkan judul/link berita.
- **{% if item.find('a') and item.find('a').has_attr('href') %},** cek apakah ada tag <a> dan memiliki atribut href.
- **<a href="{{ item.find('a')['href'] }}">,** membuat tautan ke halaman berita.
- ** {{ item.find('h2').text.strip() if item.find('h2') else 'Tanpa Judul' }} ,** menampilkan judul berita dari elemen <h2>. Jika tidak ada tag <h2>, tampilkan teks “Tanpa Judul”.
- **{% endfor %},** akhir dari perulangan for.
- **{% endblock %},** menutup blok body.

- templates (detik_jateng.html)

- **{% extends 'index.html' %},** template ini mewarisi dari index.html, artinya hanya mengisi bagian tertentu dari kerangka besar (<html>, <head>, dll).
- **{% block body %},** memulai blok bernama body yang akan menggantikan isi blok body di index.html.
- **<h2>Detik Jateng</h2>,** menampilkan judul halaman “Detik Jateng”.

- ** ,** membuat baris grid Bootstrap untuk mengatur tata letak berita dalam 2 kolom (gambar + teks).
- **{% for item in data %},** melakukan perulangan terhadap setiap elemen item dalam variabel data (hasil scraping dari scrape_detik_jateng()).
- ** ,** membuat kolom berukuran 3 (dari total 12) untuk menampilkan gambar berita.
- **{% if item.find('img') and item.find('img').has_attr('src') %},** memastikan elemen <img> ada dan memiliki atribut src sebelum ditampilkan.
- **<img src="{{ item.find('img')['src'] }}" class="img-fluid">,** menampilkan gambar berita dari tag <img> dengan class Bootstrap img-fluid agar gambar responsif.
- **{% endif %},** akhir dari kondisi untuk memeriksa dan menampilkan gambar.
- ** ,** membuat kolom berukuran 9 untuk teks (judul berita).
- **{% if item.find('a') and item.find('a').has_attr('href') %},** memastikan ada tag <a> dan memiliki atribut href untuk membuat tautan yang valid.
- **<a href="{{ item.find('a')['href'] }}">,** membuat tautan ke halaman berita berdasarkan atribut href dari tag <a>.
- ** {{ item.find('h2').text.strip() if item.find('h2') else 'Tanpa Judul' }} ,** menampilkan judul berita dari tag <h2>, jika ada. Jika tidak ada, tampilkan "Tanpa Judul".
- **{% endfor %},** menutup perulangan berita (for).
- **{% endblock %},** menutup blok body.

- templates (detik_jatim.html)

- **{% extends 'index.html' %},** template ini mewarisi dari index.html, artinya hanya mengisi bagian tertentu dari kerangka besar (<html>, <head>, dll).
- **{% block body %},** memulai blok bernama body yang akan menggantikan isi blok body di index.html.

- **<h2>Detik Jatim</h2>,** menampilkan judul halaman “Detik Jatim”.
- ** ,** membuat baris (row) Bootstrap grid system agar berita bisa dibagi ke dalam kolom.
- **{% for item in data %},** melakukan perulangan untuk setiap elemen item dalam variabel data (yang dikirim dari fungsi scrape_detik_jatim() di app.py).
- ** ,** membuat kolom berukuran 3 (dari total 12) untuk menampilkan gambar berita.
- **{% if item.find('img') %},** mengecek apakah item memiliki tag <img>. Tidak memeriksa apakah ada **src**, jadi bisa error jika img tanpa atribut src.
- **<img src="{{ item.find('img')['src'] }}",** menampilkan gambar berita. **class="img-fluid">,** membuat gambar responsif (menyesuaikan ukuran layar).
- **{% endif %},** menutup kondisi pengecekan gambar.
- ** ,** kolom sebesar 9 digunakan untuk menampilkan judul dan link berita.
- **<a href="{{ item.find('a')['href'] }}">,** membuat tautan (link) ke halaman berita berdasarkan tag <a> di dalam item.
- ** {{ item.find('h2').text.strip() if item.find('h2') else 'Tanpa Judul' }} ,** menampilkan judul berita dari tag <h2>, jika ada. Jika tidak ada, tampilkan “Tanpa Judul”.
- **{% endfor %},** menutup perulangan berita (for).
- **{% endblock %},** menutup blok body.

- templates (berita_gabungan.html)

- **<!DOCTYPE html> <html lang="id">,** menandakan dokumen ini adalah HTML5 dan menggunakan bahasa Indonesia (lang="id").
- **<meta charset="UTF-8">,** menentukan encoding karakter menjadi UTF-8 (standar karakter internasional).

- **<title>Berita Gabungan</title>,** judul halaman yang akan muncul di tab browser: **"Berita Gabungan"**.
- **<style> body { font-family, background-color, padding },** mengatur font default, warna latar belakang (#f9f9f9), dan padding pada seluruh halaman.
- **h2 { color },** warna teks pada elemen <h2> (judul kategori berita).
- **.berita-container { display, flex-wrap, gap },** mengatur layout container berita agar berderet fleksibel dengan jarak antar elemen (gap).
- **.berita-card { background-color, border-radius, box-shadow, width, overflow, transition },** styling untuk tiap kartu berita, seperti latar putih, sudut membulat, bayangan lembut, dan efek transisi saat hover.
- **.berita-card:hover { transform },** saat hover, kartu berita akan membesar sedikit (efek interaktif).
- **.berita-card img { width, height, object-fit },** gambar dalam kartu berita akan menyesuaikan lebar penuh dan dipotong rapi sesuai rasio 180px.
- **.berita-card .content { padding },** konten teks di dalam kartu akan diberi jarak dari pinggir.
- **.berita-card h4 { font-size, margin },** margin berita dalam kartu akan berukuran sedang dan diberi margin atas-bawah.
- **.berita-card a { text-decoration, color },** tautan tidak bergaris bawah dan berwarna biru terang.
- **<body> <h2>Berita dari Liputan6</h2>,** judul bagian pertama: berita yang berasal dari **Liputan6**.
- ** {% for item in liputan6 %},** membuat container grid, lalu perulangan Jinja2 untuk setiap berita dari liputan6 (data hasil scraping).

- ** <img src="{{ item.image }}" alt="Gambar Berita">,** menampilkan gambar berita (item.image) dengan deskripsi alternatif "Gambar Berita".
- ** <h4>{{ item.title }}</h4>,** menampilkan judul berita (item.title).
- **Baca Selengkapnya,** menampilkan link ke halaman detail berita, dengan url sebagai parameter query string.
- **{% endfor %},** menutup blok kartu dan blok perulangan berita dari Liputan6.
- **<h2>Berita dari CNN Indonesia</h2> {% for item in cnn %},** judul bagian kedua: berita dari **CNN Indonesia**. Memulai loop untuk data dari cnn.
- ** <img src="{{ item.image }}" alt="Gambar Berita">,** menampilkan masing-masing berita CNN dalam format kartu yang sama seperti Liputan6.
- ** <h4>{{ item.title }}</h4>,** menampilkan judul berita (item.title).
- **Baca Selengkapnya,** menampilkan link ke halaman detail berita, dengan url sebagai parameter query string.
- **{% endfor %},** menutup blok kartu dan blok perulangan berita dari CNN Indonesia.

- templates detail_berita.html

- **<!DOCTYPE html> <html lang="id">,** mendeklarasikan dokumen HTML5 dan bahasa halaman sebagai Bahasa Indonesia.
- **<meta charset="UTF-8">,** mengatur encoding karakter ke UTF-8 agar mendukung karakter internasional.
- **<title>{{ berita.title }}</title>,** menampilkan judul berita sebagai judul halaman (dinamis dari variabel berita).

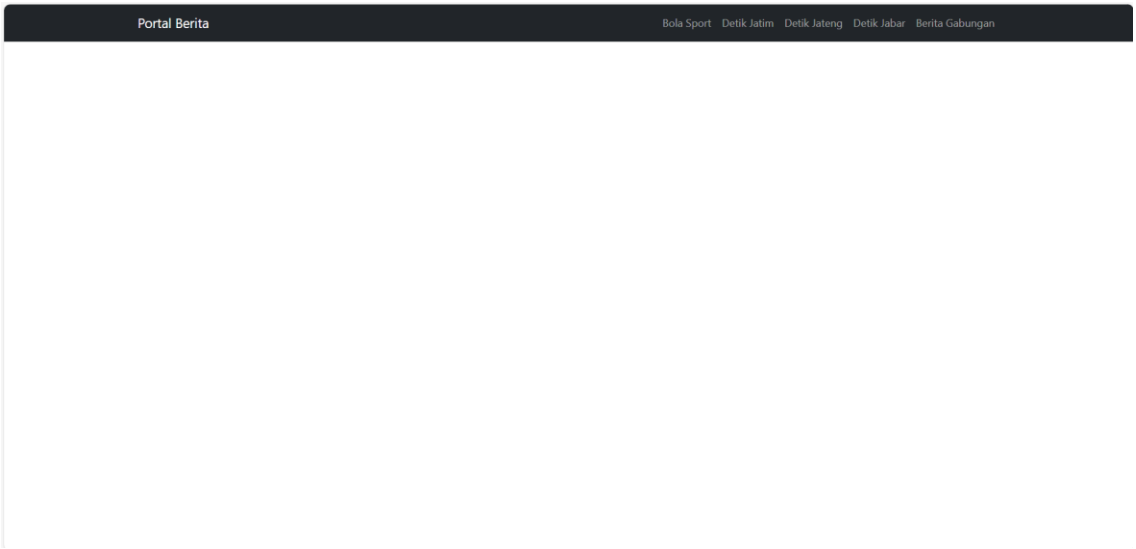
- **<style> body { font-family, background-color, margin, padding },** mengatur font, warna latar belakang, dan padding global halaman.
- **.container { max-width, margin, background-color, border-radius, box-shadow, overflow },** gaya utama kontainer berita: batas lebar, tengah, warna putih, border-radius, dan bayangan.
- **.header-image { width, max-height, object-fit },** mengatur gambar header agar memenuhi lebar penuh, tinggi maksimum 400px, dan tidak terdistorsi.
- **.content { padding },** menambahkan ruang di dalam konten utama.
- **.content h1 { font-size, color, margin-bottom } .content p { font-size, line-height, color, margin-bottom },** style untuk judul berita dan paragraf konten: ukuran, warna, dan spasi antar elemen.
- **.back-button { display, margin, background-color, color, padding, text-decoration, border-radius, transition } .back-button:hover { background-color },** gaya untuk tombol kembali: warna biru, teks putih, padding, dan efek hover.
- **<body> ,** elemen pembungkus utama untuk konten berita.
- **{% if berita.image %} <img src="{{ berita.image }}" alt="Gambar Berita" class="header-image"> {% endif %},** jika ada gambar (berita.image tidak kosong), tampilkan gambar di atas konten.
- ** <h1>{{ berita.title }}</h1> {{ berita.content|safe }},** menampilkan judul berita dan isi konten (berita.content) secara aman (boleh mengandung HTML).
- **← Kembali ke Halaman index,** menutup div kontainer, dan membuat tombol kembali ke halaman utama (/).

- templates (index.html)

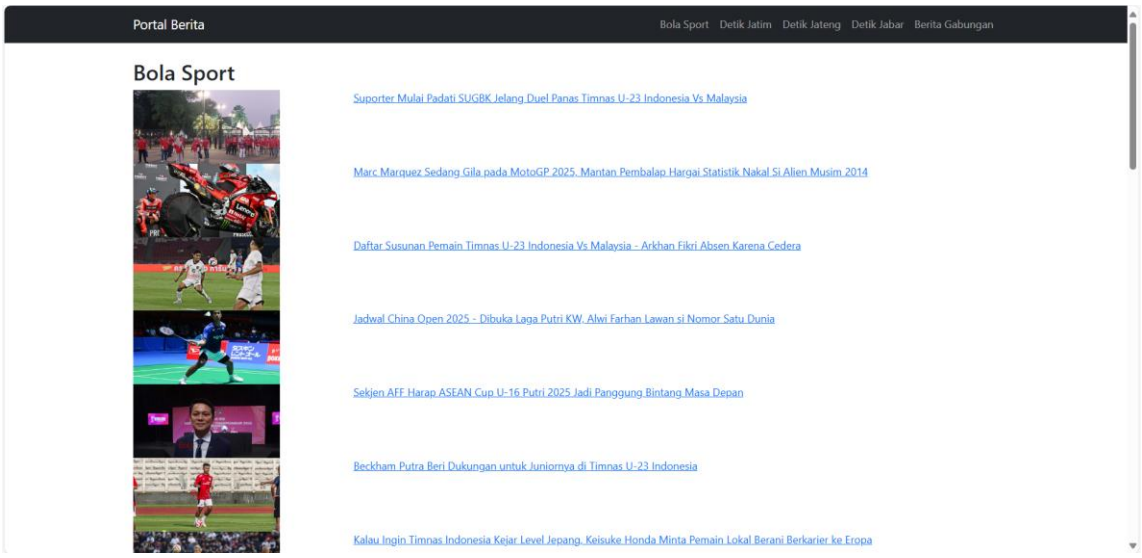
- **<!DOCTYPE html> <html lang="en">,** mendeklarasikan dokumen HTML5 dan menetapkan bahasa halaman ke **Bahasa Inggris** (en).
- **<meta charset="UTF-8">,** menetapkan encoding karakter menjadi UTF-8 agar mendukung karakter internasional dan simbol.
- **<title>Berita Terkini</title>,** menampilkan judul halaman browser sebagai "Berita Terkini".
- **<linkhref="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">,** menyisipkan CDN Bootstrap 5.3 untuk mendukung komponen dan gaya CSS Bootstrap.
- **<body> <nav class="navbar navbar-expand-lg navbar-dark bg-dark">,** membuat navbar Bootstrap yang gelap (bg-dark) dan akan **melebar** di layar besar (navbar-expand-lg).
- ** ,** membungkus isi navbar dengan kelas container agar rata tengah dan tidak full-width.
- **Portal Berita,** link judul navbar yang akan membawa pengguna ke halaman utama (/), dengan gaya navbar-brand.
- ** ,** container untuk elemen navbar yang bisa "collapse" (menyusut di layar kecil).
- **<ul class="navbar-nav ms-auto">,** daftar menu navigasi, ms-auto membuat menu berada di kanan (margin-start auto).
- **<li class="nav-item">Bola Sport,** item menu yang mengarah ke halaman /bola-sport.
- **<li class="nav-item">Detik Jatim,** link ke halaman kategori berita “Detik Jatim”.
- **<li class="nav-item">Detik Jateng,** link ke halaman kategori berita “Detik Jateng”.

- **<li class="nav-item">Detik Jabar,** link ke halaman kategori berita “Detik Jabar”.
- **<li class="nav-item">Berita Gabungan,** link ke halaman kategori berita “Gabungan Berita”.
- ** ,** container untuk isi utama halaman dengan margin atas (mt-4) agar tidak menempel ke navbar.
- **{% block body %}{% endblock %},** placeholder Jinja2 untuk konten dinamis. Template lain akan mengisi blok ini (misalnya: detik_jatim.html).

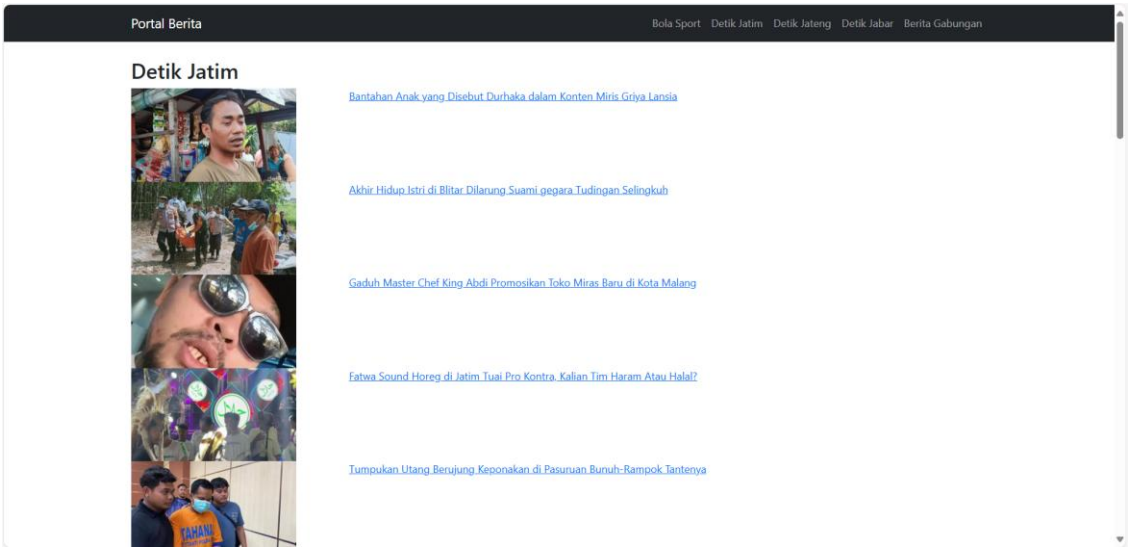
c. Hasil Uji Coba



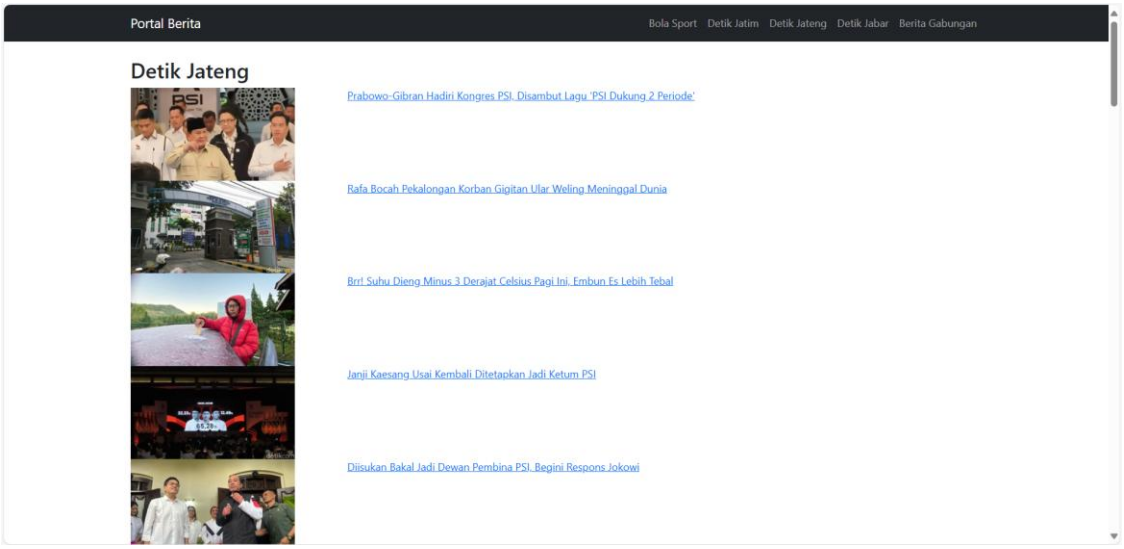
Gambar 8.1 Tampilan Halaman Index (index.html)



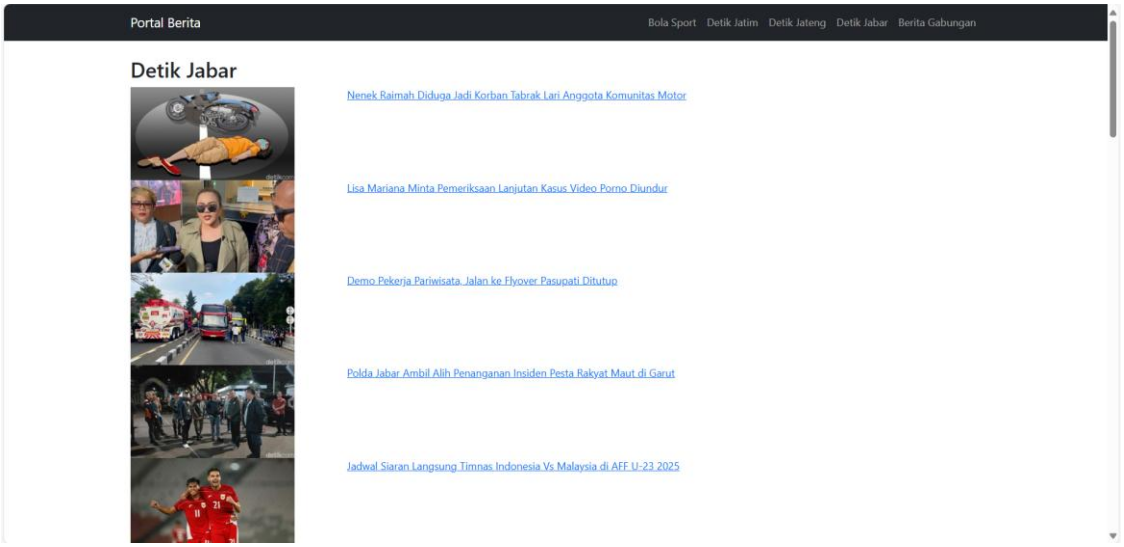
Gambar 8.2 Tampilan Halaman Bola Sport (bola-sport.html)



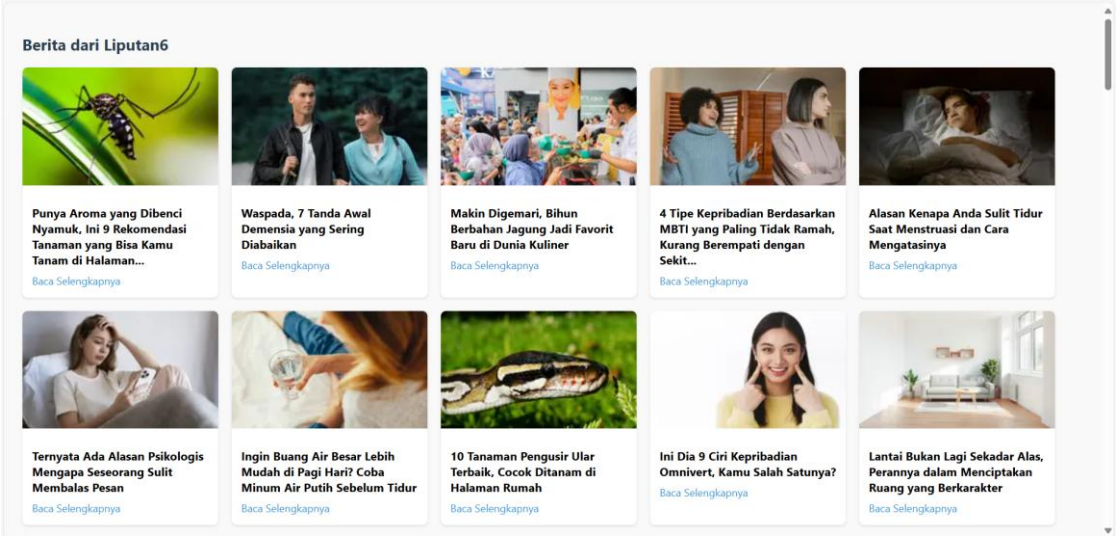
Gambar 8.3 Tampilan Halaman Detik Jatim (detik-jatim.html)



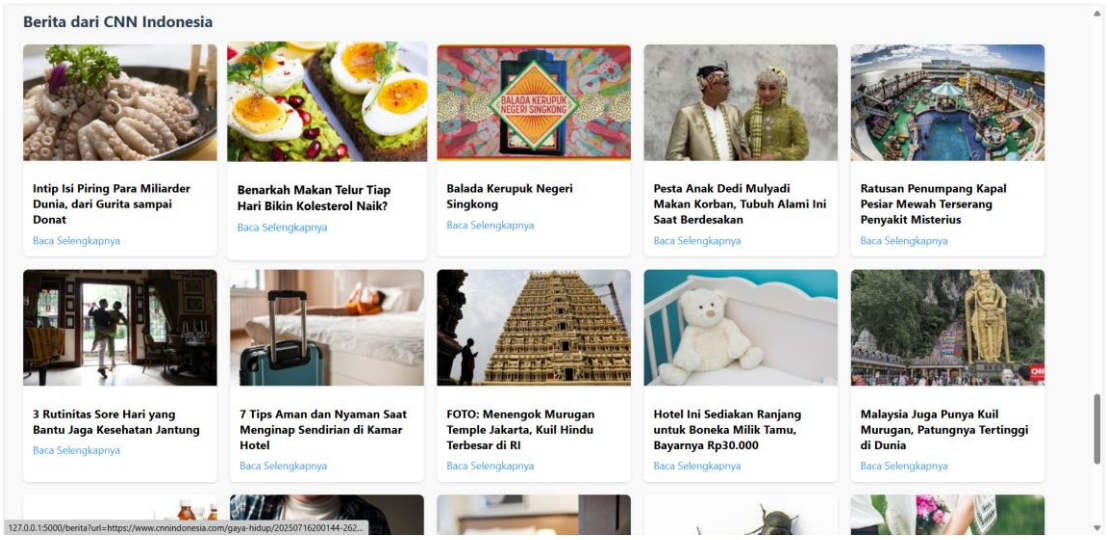
Gambar 8.4 Tampilan Halaman Detik Jateng (detik-jateng.html)



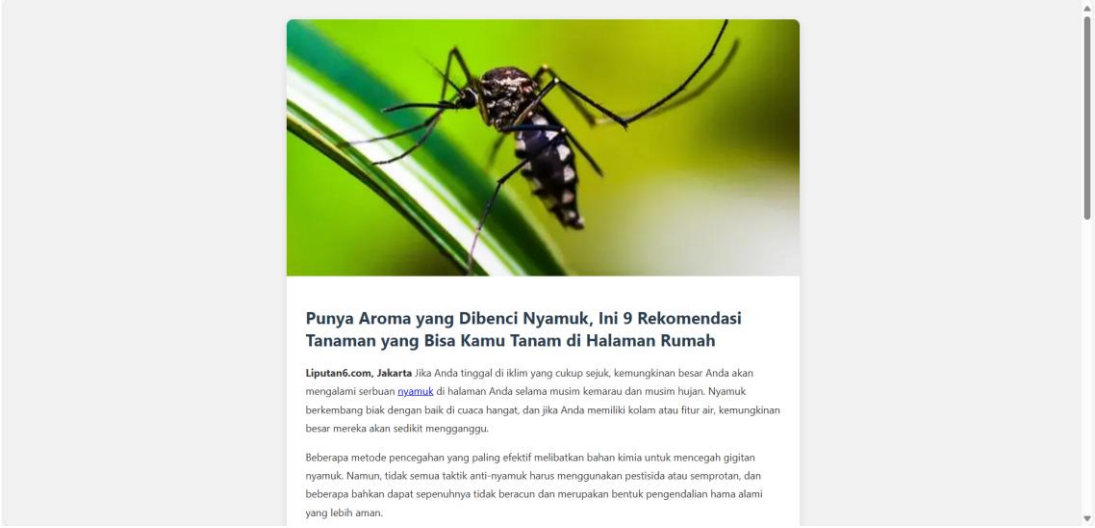
Gambar 8.5 Tampilan Halaman Detik Jabar (detik-jabar.html)



Gambar 8.6 Tampilan Halaman Berita Gabungan Liputan6 (berita_gabungan.html)



Gambar 8.7 Tampilan Halaman Berita Gabungan CNN Indonesia (berita_gabungan.html)

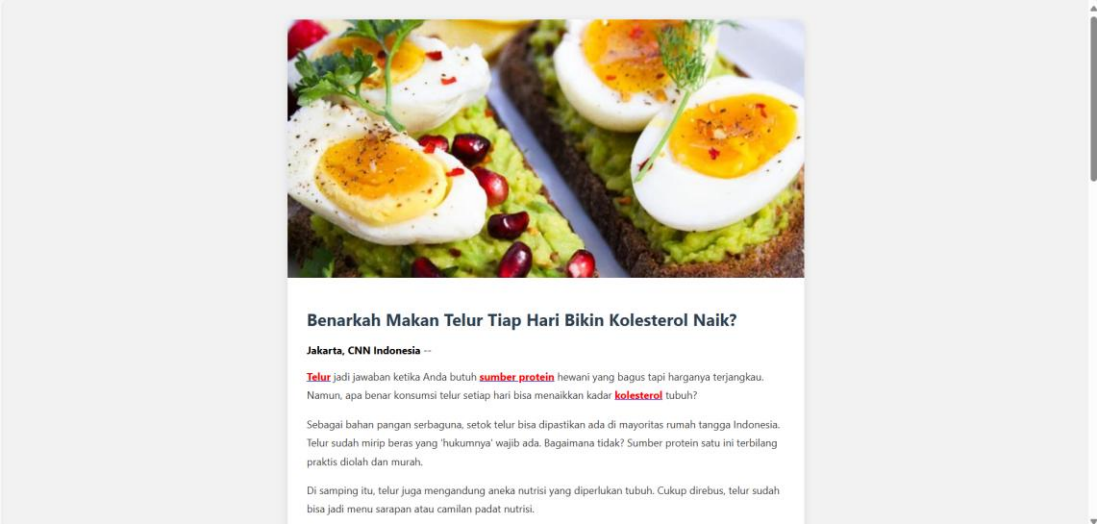


Punya Aroma yang Dibenci Nyamuk, Ini 9 Rekomendasi Tanaman yang Bisa Kamu Tanam di Halaman Rumah

Liputan6.com, Jakarta Jika Anda tinggal di iklim yang cukup sejuk, kemungkinan besar Anda akan mengalami serbuan nyamuk di halaman Anda selama musim kemarau dan musim hujan. Nyamuk berkembang biak dengan baik di cuaca hangat, dan jika Anda memiliki kolam atau fitur air, kemungkinan besar mereka akan sedikit mengganggu.

Beberapa metode pencegahan yang paling efektif melibatkan bahan kimia untuk mencegah gigitan nyamuk. Namun, tidak semua taktik anti-nyamuk harus menggunakan pestisida atau semprotan, dan beberapa bahkan dapat sepenuhnya tidak beracun dan merupakan bentuk pengendalian hama alami yang lebih aman.

Gambar 8.8 Tampilan Halaman Detail Berita Liputan6 (detail_berita.html)



Benarkah Makan Telur Tiap Hari Bikin Kolesterol Naik?

Jakarta, CNN Indonesia --

Telur jadi jawaban ketika Anda butuh **sumber protein** hewani yang bagus tapi harganya terjangkau. Namun, apa benar konsumsi telur setiap hari bisa menaikkan kadar **kolesterol** tubuh?

Sebagai bahan pangan serbaguna, setok telur bisa dipastikan ada di mayoritas rumah tangga Indonesia. Telur sudah mirip beras yang 'hukumnya' wajib ada. Bagaimana tidak? Sumber protein satu ini terbilang praktis diolah dan murah.

Di samping itu, telur juga mengandung aneka nutrisi yang diperlukan tubuh. Cukup direbus, telur sudah bisa jadi menu sarapan atau camilan padat nutrisi.

Gambar 8.9 Tampilan Halaman Detail Berita CNN Indonesia (detail_berita.html)

d. Analisa Hasil

Secara umum, sistem yang dibangun menunjukkan integrasi data scraping berita dari berbagai sumber. Template index.html berperan sebagai layout utama dengan navigasi navbar Bootstrap yang mengarahkan ke berbagai sumber berita seperti Bola Sport, Detik Jatim, Detik Jateng, Detik Jabar dan Berita Gabungan.

Tiap halaman turunan seperti detik_jatim.html mewarisi struktur dasar dari index.html dan menampilkan daftar berita dari sumber terkait dengan tata letak berbasis grid Bootstrap, memisahkan gambar dan judul secara proporsional.

berita_gabungan.html menggabungkan dua sumber berita (Liputan6 dan CNN) dalam dua blok berbeda, menggunakan card dengan bayangan dan transisi hover yang meningkatkan tampilan profesional dan estetika.

Sementara itu, detail_berita.html berfokus pada tampilan satu berita secara lengkap, menyajikan gambar header, isi konten, dan tombol kembali yang fungsional. Penyaringan konten dengan |safe juga menunjukkan bahwa konten HTML dari hasil scraping ditampilkan langsung tanpa di-escape, yang berguna untuk mempertahankan format artikel asli. Struktur layout dan hasil scraping tampak berhasil diterapkan dengan baik tanpa error tampilan atau pemformatan, mencerminkan bahwa template Jinja dan struktur data yang digunakan sudah sesuai dengan konteks scraping masing-masing situs berita.

8.5 Kesimpulan

8.5.1 Kesimpulan Percobaan 1

Percobaan ini berhasil menunjukkan bahwa integrasi antara scraping data berita dan penyajian antarmuka berbasis Flask serta template Jinja2 dapat berjalan dengan baik. Setiap halaman seperti bola-sport.html, detik_jatim.html, detik_jateng.html, detik_jabar.html, dan berita_gabungan.html mampu menampilkan hasil scraping dalam bentuk yang informatif dan terstruktur, baik gambar maupun judul berita. Template index.html berfungsi efektif sebagai kerangka dasar yang dapat diwarisi oleh halaman lain, sementara detail_berita.html memberikan tampilan konten berita secara lengkap. Antarmuka pengguna menggunakan Bootstrap membuat tampilan lebih rapi dan responsif. Dari sisi fungsionalitas, sistem mampu menavigasi antarhalaman dan menampilkan konten sesuai sumber masing-masing. Hal ini membuktikan bahwa alur pengambilan data, pemrosesan, dan rendering di sisi frontend telah berjalan dengan baik tanpa error atau bug tampilan yang berarti.

Mengetahui:

Dosen Pengampu Mata Kuliah

Arif Hadi Sumitro , M.Kom

NIKP. xxx

DAFTAR PUSTAKA

1. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
2. Mitchell, R. (2018). *Web Scraping with Python: Collecting Data from the Modern Web* (2nd ed.). O'Reilly Media.
3. *Sonmez, E., & Dede, E. (2020). Web Scraping and Data Integration in Python Using Flask Framework. International Journal of Engineering Research and Technology (IJERT), 9(5).*