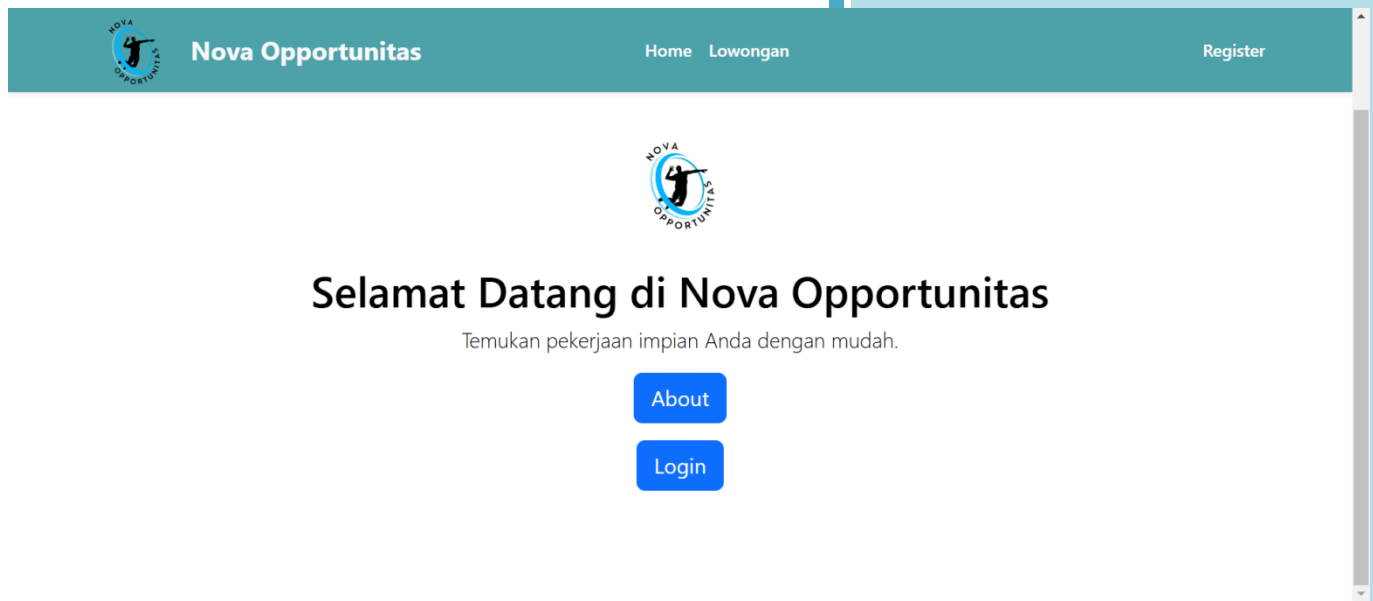


Web Lowongan Pekerjaan



KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, modul pembelajaran ini dapat siselesaikan dengan baik. Modul ini dirancang untuk membantu siswa dan praktisi dalam memahami dan mengembangkan aplikasi berbasis web menggunakan Flask dan SQLite. Dengan adanya modul ini, diharapkan pembaca dapat memperoleh pemahaman mendalam tentang cara membangun website sederhana hingga kompleks dengan menggunakan teknologi yang tersedia.

Pembuatan website menggunakan Flask dapat dilakukan dengan beberapa Langkah utama. Pertama, pastikan lingkungan pengembangan telah dipersiapkan dengan menginstal Python dan Flask di PyCharm menggunakan fitur Python Packages di PyCharm dengan mencari dan menginstal Flask melalui menu Python Packages. Setelah itu, kita Membuat struktur proyek yang terdiri dari file utama app.py , direktori templates/ untuk menyimpan file HTML, serta direktori static/ untuk menyimpan file CSS dan JavaScript.


Setelah sturktur proyek siap, kita mulai dengan Membuat rute (routing) di dalam app.py menggunakan decorator @app.route(' / ') yang akan menangani permintaan HTTP dari klien. Kita dapat menggunakan render_template untuk menampilkan halaman web denga HTML yang telah disusun di dalam folder templates/ . Selanjutnya, kita dapat menghubungkan aplikasi Flask dengan database SQLite untuk Menyimpan dan mengelola data. Hal ini dilakukan dengan menggunakan Pustaka sqlite3 dalam Python.

Setelah semua komponen selesai di buat, webite dapat dijalankan dengan perintah python app.py, dan dapat diakses melalui browser di alamat <http://127.0.0.1:5000/> . Dengan mengikuti Langkah-langkah tersebut, pembaca diharapkan mampu mengembangkan website berbasis Flask secara mandiri.

Kami mengucapkan teria kasih kepada semua pihak yang telah memberikan dukungan dalam penyusunan modul ini. Semoga modul ini dapat menjadi referensi yayng bermanfaat dan mendukung pembelajaran dalam dunia teknologi.

Malang, 5 Maret 2025

PENULIS

	Bidang Keahlian:	Lab:	Tanggal:	TI	SE	04	VT	Page 1 - 16
	Teknologi Informasi	SE - 209	MAR. 2025					
	Program Keahlian:	Universal	Rev.:	Dibuat oleh:				
	TI-SE-TE	Internet of Things	05/03/2025	EKO-Erlanaga				

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam era digital saat ini, pembuatan website menjadi keterampilan yang sangat penting. Website tidak hanya digunakan untuk keperluan bisnis, tetapi juga sebagai sarana komunikasi, edukasi, dan hiburan. Penggunaan framework web mempermudah proses pengembangan website, baik untuk pemula maupun profesional. Flask merupakan salah satu framework berbasis Python yang ringan dan fleksibel, sehingga cocok digunakan untuk mengembangkan website dengan cepat dan efisien. Dengan Flask, pengembang dapat membuat aplikasi web sederhana maupun kompleks dengan struktur kode yang mudah dipahami.

Selain itu, database juga memegang peranan penting dalam pengolahan data dalam sebuah website. Salah satu database yang sering digunakan bersama Flask adalah SQLite. SQLite merupakan database yang ringan dan mudah digunakan, sehingga sangat cocok untuk proyek kecil hingga menengah. Dengan kombinasi Flask dan SQLite, pengembang dapat membuat website yang memiliki performa baik dengan pengolahan data yang efektif.

Modul ini disusun untuk memberikan panduan Langkah demi Langkah dalam membangun website menggunakan Flask DAN SQLite. Modul ini mencakup berbagai aspek, mulai dari instalasi Flask, pembuatan struktur proyek, pembuatan tampilan dengan HTML dan CSS, hingga pengelolaan data dengan SQLite .

1.2. Tujuan

Tujuan dari penyusunan modul ini adalah:

1. Memberikan pemahaman dasar tentang Flask dan cara penggunaannya dalam pembuatan website.
2. Menjelaskan struktur proyek Flask dan fungsinya masing-masing.
3. Menajarkan cara menghubungkan Flask dengan database SQLite.
4. Membantu siswa dan praktisi dalam membuat website yang dapat berjalan secara local maupun online.



1.3. Manfaat

Dengan mempelajari modul ini, pembaca diharapkan dapat:

1. Memahami dasar-dasar pengembangan web menggunakan Flask.
2. Mampu membuat proyek web sederhana dengan struktur yang baik.
3. Mengelola database menggunakan SQLite dalam website Flask.
4. Mengembangkan website yang interaktif dan dinamis.

1.4. Sistematika Modul

Modul ini disusun menjadi beberapa bab dengan sistematika sebagai berikut:

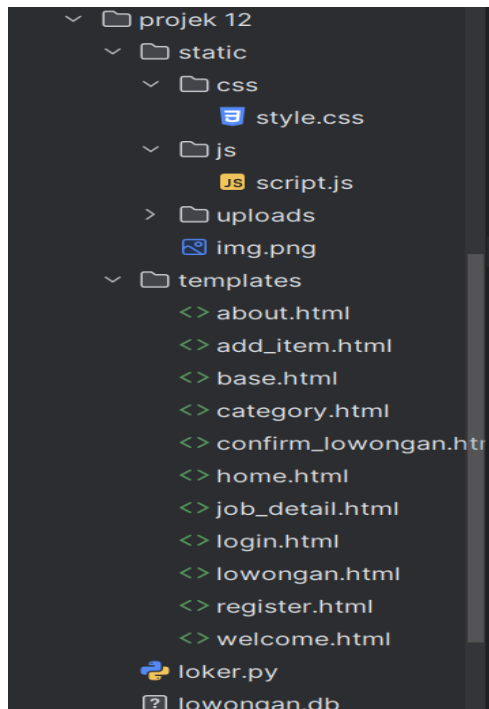
1. **Bab I: Pendahuluan** - Membahas latar belakang, tujuan, manfaat, dan sistematika penulisan modul.
2. **Bab II: Struktur Proyek Flask** - Menguraikan struktur direktori proyek Flask dan Desain sistem.
3. **Bab III: Backend dengan MICROFRAMEWORK FLASK PYTHON** - Membahas cara mengelola database dengan SQLite dalam aplikasi Flask.
4. **Bab IV: Frontend dengan HTML CSS dan JS** - Membahas cara mengelola Frontend .

BAB II

STRUKTUR PROYEK FLASK

2.1. Struktur Direktori Proyek

Saat membangun website dengan Flask, kita perlu memiliki struktur proyek yang rapi agar mempermudah pengelolaan file dan pemanggilan fungsi. Struktur direktori proyek Flask umumnya terlihat seperti berikut:



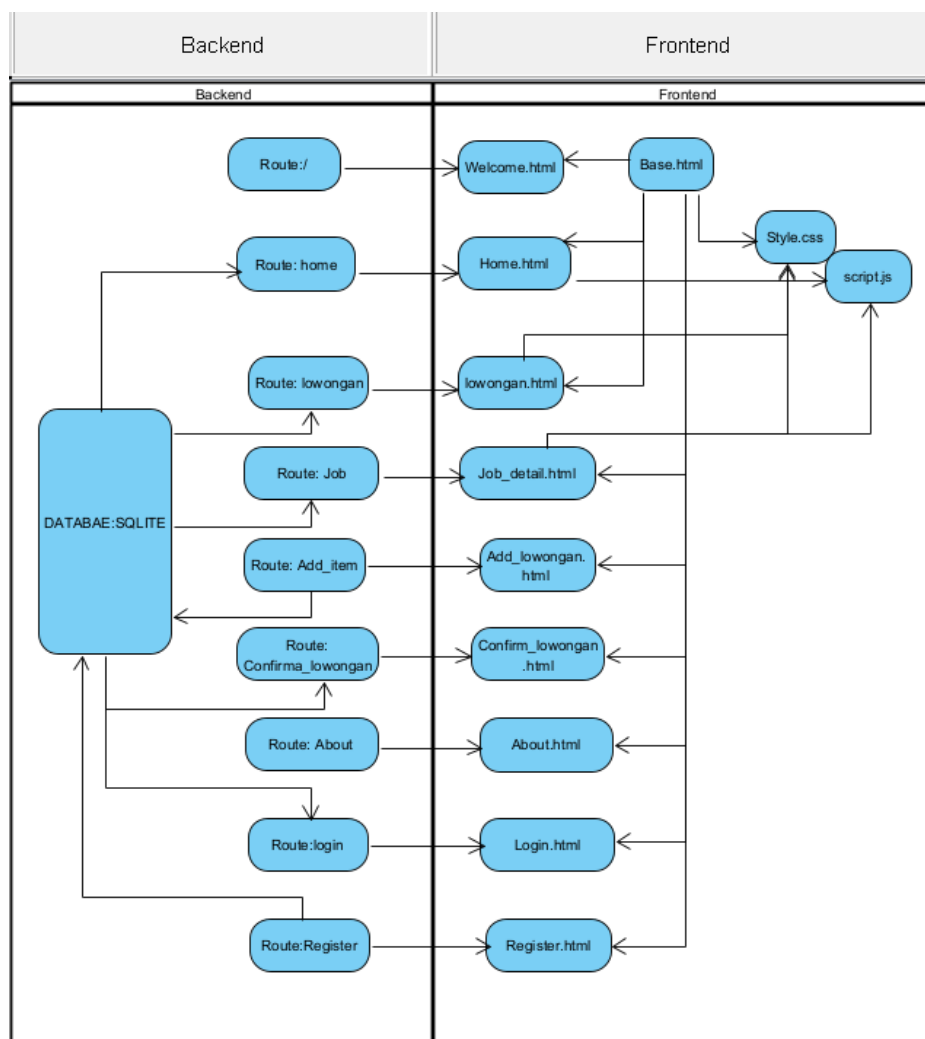
Setiap folder memiliki fungsinya masing-masing:

- static/ Folder ini Menyimpan file static seperti CSS, JavaScript, dan gambar.
- templates/ Folder yang menyimpan file HTML yang akan dirender oleh Flask.
- loker.py File utama yang menjalankan website Flask, menangani request, dan menghubungkan ke database.

- lowongan.db Database SQLite untuk Menyimpan data pesanan.

2.2. Desain Sistem

Cara kerja website dapat dilihat pada flowchart berikut.



*Keterangan:

Flask akan bekerja sebagai Backend yang menangani input data dan disimpan ke database; HTML, CSS, dan JavaScript akan menjadi tampilan utama website; SQLite berperan sebagai database, yang menyimpan data-data yang telah dimasukkan oleh pengguna.

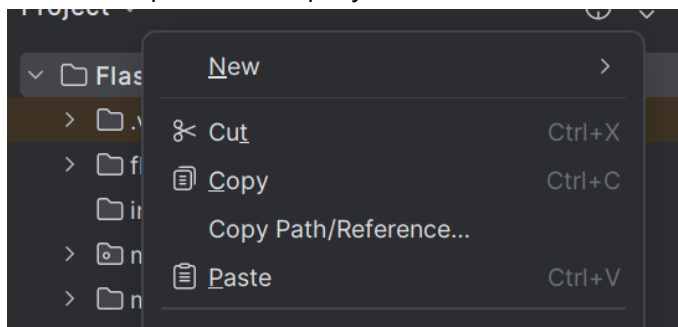
BAB III

BAKEND DENGAN MICROFRAMEWORK FLASK PYTHON

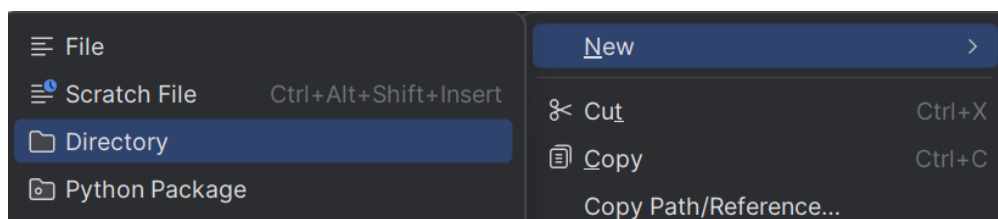
3.1.Membuat File Di Pycharm

Untuk Membuat setiap file di **Pycharm**, lakukan Langkah – Langkah berikut:

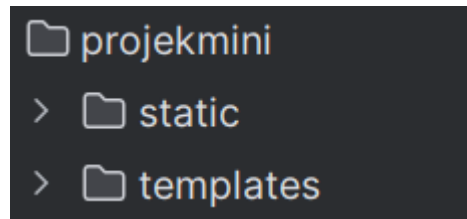
1. Klik kanan pada folder proyek (folder anda).



2. Terus klik **new** => **Directory** buat nama terserah kalian (projekmini).

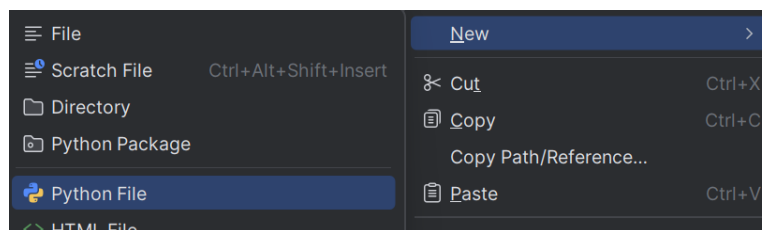


3. Terus klik kanan yang directory yang anda buat.
 - Buat nama folder " templates " untuk file HTML.
 - Buat nama folder " static " untuk file CSS, JAVASCRIPT, dan Gambar.

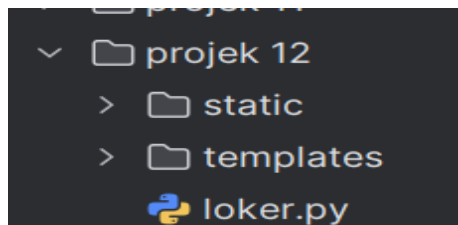


4. Untuk Membuat file app.py (file utama)

- Klik kanan ke directory yang anda buat
- Klik **new** => **Python File**



5. Ketik nama file : " loker.py ", lalu klik **enter** .



3.2.Membangun Backend dengan Flask Python

Berikut adalah cara memulai membangun Backend Website dengan Flask :

1. Ketik kode di bawah ini


```

from flask import Flask, render_template, request, redirect, url_for, session, jsonify
from werkzeug.utils import secure_filename
from werkzeug.security import generate_password_hash, check_password_hash
import os
import sqlite3

app = Flask(__name__)

```

Kode tersebut mengimpor beberapa modul dan fungsi dari Flask, modul sqlite3 di Python, serta inisialisasi aplikasi Flask.

2. Koneksi ke database

```

DATABASE = "lowongan.db"

# Fungsi koneksi database
def get_db():
    conn = sqlite3.connect(DATABASE)
    conn.row_factory = sqlite3.Row
    return conn

```

Dengan fungsi get_db(), kalian dapat dengan mudah mengelola koneksi ke database SQLite dan menjalankan operasi database dalam aplikasi Flask.

3. Buat tabel jika belum tersedia

```

# Inisialisasi database
def create_table():
    conn = get_db()
    cur = conn.cursor()
    # Tabel lowongan kerja
    cur.execute('''
        CREATE TABLE IF NOT EXISTS lowongan (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            title TEXT NOT NULL,
            description TEXT NOT NULL,
            gaji TEXT,
            company_address TEXT,
            syarat TEXT,
            logo TEXT,
            name TEXT,
            kategori TEXT,
            status TEXT DEFAULT 'pending'
        )
    ''')

```

Tabel ini akan terkoneksi dengan DB Browser, yang mana data-data seperti title, description, dan lainnya akan tersimpan di dalamnya.

4. Buat route untuk halaman Pertama

```

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

```

Kode di atas adalah contoh penggunaan route dalam Flask. Pada fungsi index, route akan mengembalikan file HTML yaitu welcome.html.

5. Halaman Login

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        conn = get_db()
        user = conn.execute(sqlalchemy.sql("SELECT * FROM users WHERE email=?", parameters=(email,))).fetchone()
        conn.close()

        if user and check_password_hash(user["password"], password):
            session['user_id'] = user["id"]
            session['role'] = user["role"]
            return redirect(url_for('home_page'))
        else:
            return render_template(template_name_or_id='login.html', error="Email atau Password salah!")

    return render_template('login.html')
```

Pada route ini akan mengambil data user yang sudah ada atau terdaftar di database yang berfungsi untuk user melakukan login dalam halaman web. Disini juga menggunakan session yang berguna untuk menyimpan data pengguna atau informasi seperti user ID, sehingga pengguna tidak perlu login ulang setiap kali berpindah halaman

1. Menjalankan program

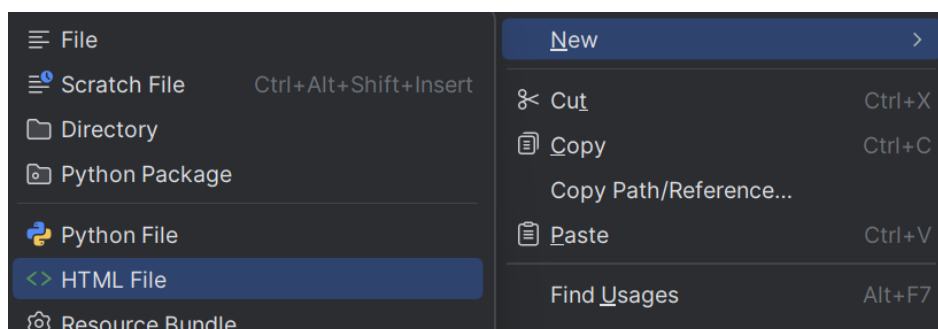
```
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Kode di atas berfungsi untuk menjalankan program yang telah dibuat sebagai auto-reload, debugger jika terjadi error, serta akses dari jaringan eksternal.

BAB IV FRONTEND DENGAN HTML CSS DAN JS

4.1. Membuat Struktur Halaman HTML

Buat file "base.html" di dalam templates/ sebagai base untuk semua file html yang ada, biasanya digunakan untuk membuat navbar yang nantinya akan tampil atau terhubung ke semua file html



4.2. Code Base.html

Ikuti yang ada di dalam foto tersebut

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> <!-- Membuat tampilan responsif di perangkat mobi

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"> <!-- Memuat Bootstrap dari CDN --

  <!-- Custom CSS -->
  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}"> <!-- Memuat file CSS kustom dari folder static -->

  <!-- Bootstrap Icons -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css"> <!-- Memuat ikon Bootstrap dari CDN

  <!-- Custom JavaScript -->
  <script src="{{ url_for('static', filename='js/script.js') }}"></script> <!-- Memuat file JavaScript kustom dari folder static -->

  <title>{% block title %}{% endblock %}</title> <!-- Placeholder untuk judul halaman yang akan diisi oleh template lain -->
</head>
<body>

  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg fixed-top shadow-sm"> <!-- Navbar dengan efek shadow dan tetap di posisi atas saat di-scroll -->
    <div class="container"> <!-- Membuat navbar responsif dengan container Bootstrap -->
      <a class="navbar-brand text-light fw-bold d-flex align-items-center" href="{{ url_for('index') }}"> <!-- Logo dan nama aplikasi
```

Didalam file base.html kita bisa menghubungkan CSS,JS maupun Bootstrap yang nantinya kita tidak perlu menghubungkan satu persatu kepada halaman html yang ada. Selain itu kita juga bisa menuliskan code untuk membuat navbar pada base yang nantinya navbar akan muncul di setiap file yang terhubung dengan base. Untuk code yang lebih lengkap bisa dilihat melalui link Github yang berada pada bagian akhir modul.

4.3. Code Login.html

Berikut Potongan code dari login.html

Code:

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

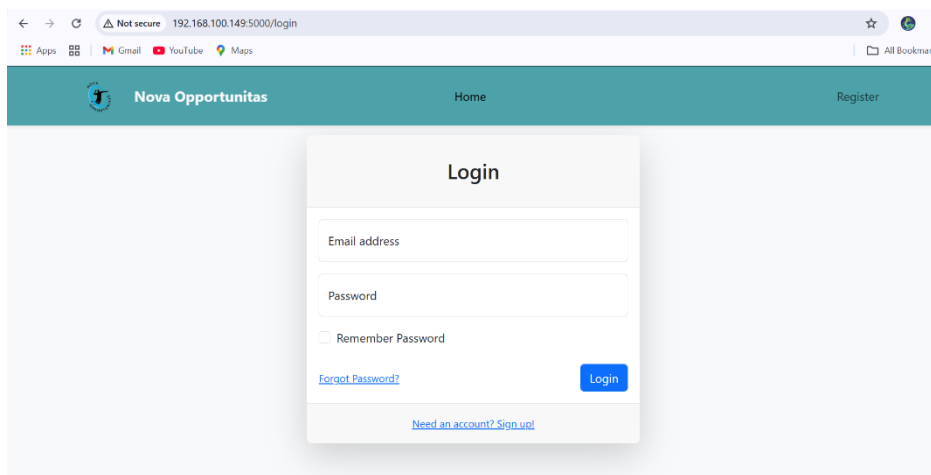
{% block title %}login{% endblock %}

{% block content %}

<div id="login_authentication">
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-12">
        <div class="card shadow-lg border-0 rounded-lg mt-3">
          <div class="card-header">
            <h3 class="text-center font-weight-light my-4">login</h3>
          </div>
          <div class="card-body">
            <div class="text-center">
              <div class="text-danger">{{ error }}</div>
            </div>
            <form method="POST">
              <div class="form-floating mb-3">
                <input class="form-control" id="inputEmail" type="email" name="email" placeholder="name@example.com" required />
                <label for="inputEmail">Email address</label>
              </div>
              <div class="form-floating mb-3">
                <input class="form-control" id="inputPassword" type="password" name="password" placeholder="Password" required />
                <label for="inputPassword">Password</label>
              </div>
              <div class="form-check mb-3">
                <input class="form-check-input" id="inputRememberPassword" type="checkbox" value="" />
                <label class="form-check-label" for="inputRememberPassword">Remember Password</label>
              </div>
              <div class="d-flex align-items-center justify-content-between mt-4 mb-0">
                <a href="#">Forgot Password?</a>
                <button type="submit" class="btn btn-primary">login</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

</div>
```

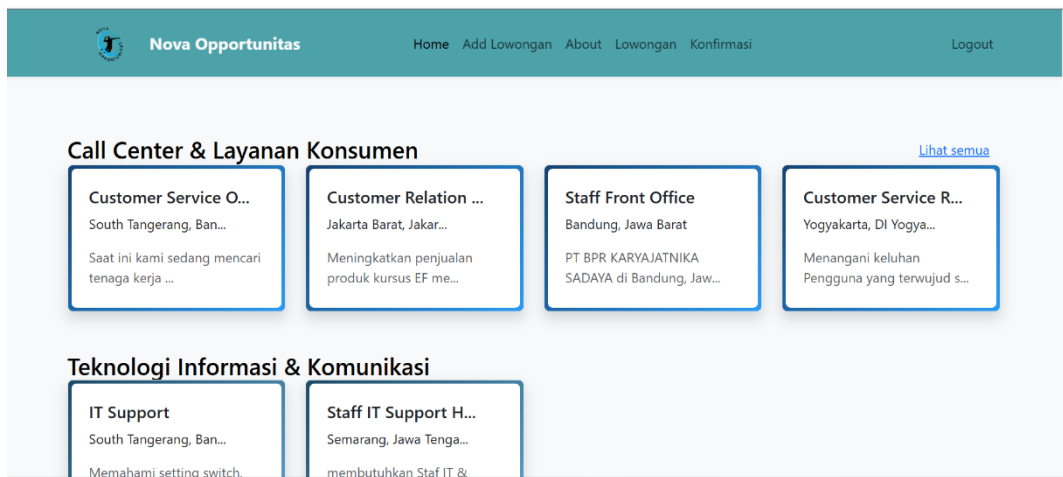
Output:



Pada awal code terdapat {% extends 'base.html' %} yang berfungsi untuk menghubungkan dengan file base.html. Untuk hasil Outputnya semua konten yang ada dalam base akan berada di atas setelah itu dilanjutkan dengan konten yang ditulis file yang melakukan extends dengan base, disini berarti form login.

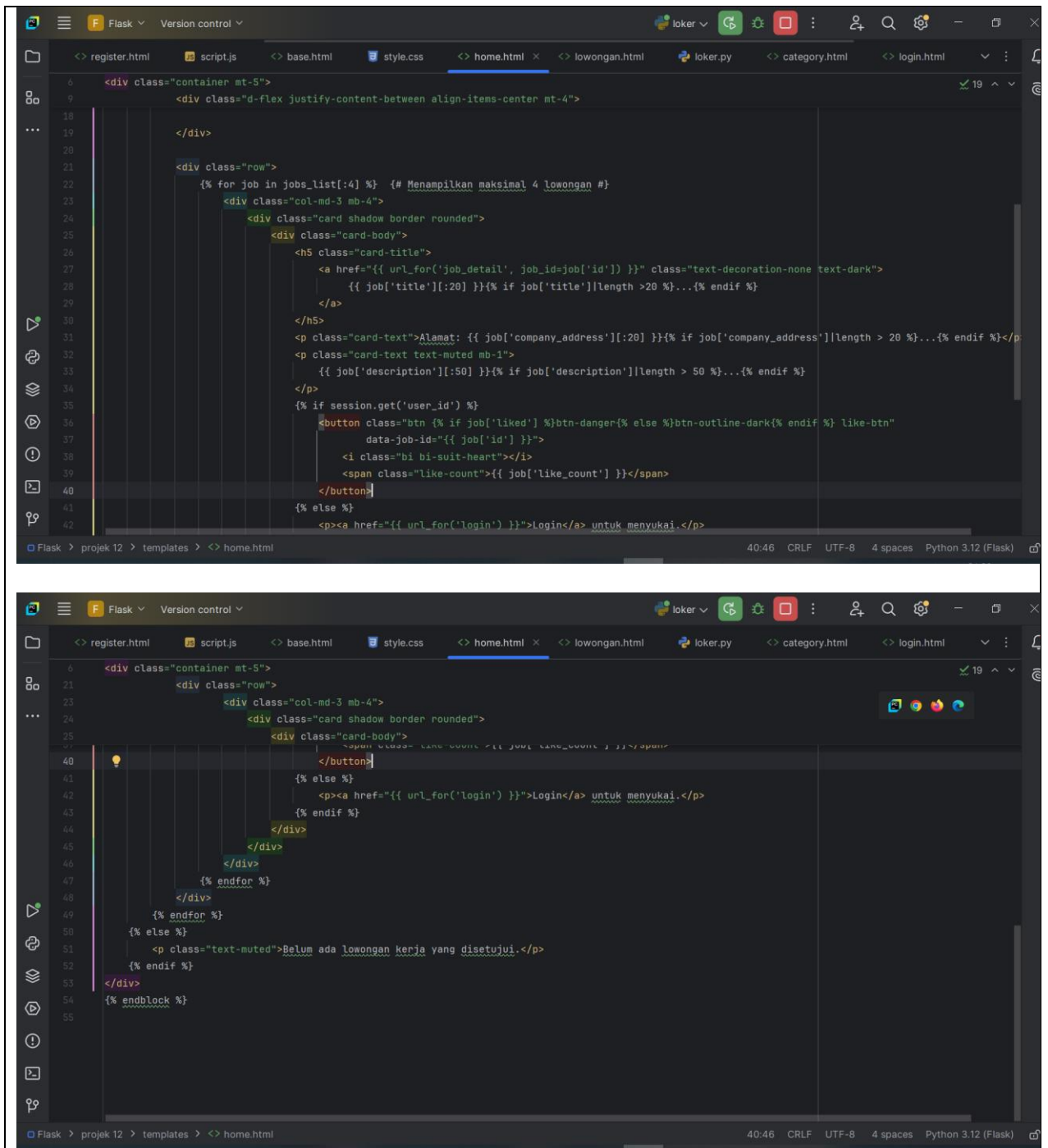
4.4. Home.html

Ouput:



Code HTML

```
1 {% extends 'base.html' %}
2
3 {% block title %}Home{% endblock %}
4
5 {% block content %}
6 <div class="container mt-5">
7     {% if jobs %}
8         {% for category, jobs_list in jobs.items() %}
9             <div class="d-flex justify-content-between align-items-center mt-4">
10                 <h3 class="m-0">{{ category }}</h3>
11                 {% if jobs_list|length > 4 %}
12                     {% if session.get('user_id') %}
13                         <a href="{{ url_for('category_jobs', category=category) }}" class="btn btn-link">Lihat semua</a>
14                     {% else %}
15                         <a href="{{ url_for('login') }}" class="btn btn-link">Login untuk melihat semua</a>
16                     {% endif %}
17                 {% endif %}
18             </div>
19
20             <div class="row">
21                 {% for job in jobs_list[:4] %} {# Menampilkan maksimal 4 lowongan #}
22                     <div class="col-md-3 mb-4">
23                         <div class="card shadow border rounded">
24                             <div class="card-body">
25                                 <h5 class="card-title">
26                                     <a href="{{ url_for('job_detail', job_id=job['id']) }}" class="text-decoration-none text-dark">
```



Link Github Untuk Code yang lengkap:

<https://github.com/erlanggawigadarma/webFlaskLoker.git>

DAFTAR PUSTAKA

1. <https://www.youtube.com/watch?v=p068JokuThU&list=PLOkVupluCljuPtTkhO6jmA76uQR994Wvi&index=1>
2. <https://getbootstrap.com/>
3. <https://www.w3schools.com/>
4. <https://colorhunt.co/>
5. <https://startbootstrap.com/template/sb-admin>

PENYUSUN



Eko Subiyantoro adalah widyaiswara di BBPPMPV BOE-VEDC Malang di Departemen Teknologi Informasi. Lahir di Banyuwangi pada tahun 1975. Pendidikan SD, SMP, dan STM (sekarang SMK) diselesaikan di kota kelahirannya. Selanjutnya ia menyelesaikan S1 di Program Studi Pendidikan Teknik Elektronika FPTK IKIP Yogyakarta (sekarang UNY) pada tahun 1998. Sarjana Sains Terapan (SST) diperolehnya melalui program beasiswa Dikmenjur (sekarang Direktorat Pembinaan Sekolah Menengah Kejuruan) di PENS ITS Surabaya pada tahun 2002 pada Prodi Teknologi Informasi dengan predikat cum laude. Magister Teknik diperolehnya melalui program beasiswa unggulan BPKLN di ITS Surabaya Program Studi Teknik Elektro konsentrasi Jaringan Cerdas Multimedia pada tahun

2009 juga dengan predikat cum laude. Doktor diperolehnya juga melalui beasiswa unggulan bagi pegawai Kemdikbud di Prodi S3 Ilmu Komputer FMIPA Universitas Gadjah Mada.

Pengalaman luar negeri diperolehnya melalui program Inwent dari pemerintah Jerman pada tahun 2003 dalam program Advance Automation Training. Selain itu, pada tahun 2006 mengikuti Expert Program Maintenance and Repair di Abha College Kingdom of Saudi Arabia (KSA). Selanjutnya pada tahun 2013 melalui IGI kerja sama Indonesia–Jerman diikutkan dalam program Modern Management School di Kassel Jerman.

Penulis memiliki pengalaman kerja di PPPPTK BOE/VEDC Malang sebagai Kepala Departemen Teknologi Informasi selama 2 periode 2010–2013 dan 2013–2015. Selain itu juga sebagai tim pengembang Java Education Network Indonesia (JENI), sebuah sistem yang diinisiasi BPKLN Kemdikbud dalam kurun waktu tahun 2007–2009. Sertifikasi yang diperoleh dalam bidang pemrograman adalah Sun Certified Java Programmer (SCJP) pada tahun 2008.

Eko Subiyantoro dapat dihubungi melalui alamat e-mail ekovedc@gmail.com