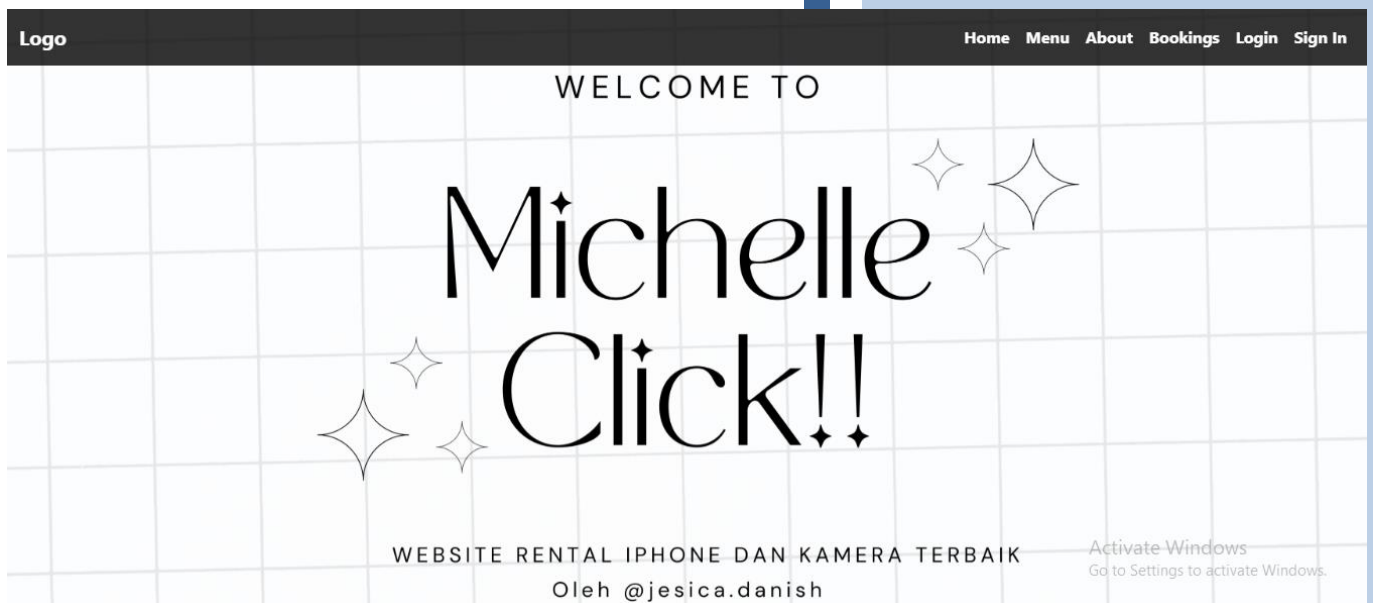


Website Rental Kamera



KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, modul Mini Project Website Rental Kamera ini dapat diselesaikan dengan baik. Modul ini dirancang sebagai panduan dalam memahami serta mengembangkan sistem berbasis web untuk kebutuhan rental elektronik dan gadget, yang mencakup kamera, lensa, audio, video, iPhone, dan drone.

Website rental ini dikembangkan menggunakan teknologi Flask (Python) sebagai backend, SQLite sebagai database, dan Bootstrap untuk tampilan antarmuka. Sistem ini menyediakan fitur booking online, daftar produk yang tersedia, serta perhitungan otomatis untuk total pembayaran dan deposit berdasarkan durasi sewa. Pengguna juga diwajibkan mengunggah bukti pembayaran saat melakukan booking, yang kemudian akan dikonfirmasi oleh admin sebelum proses penyewaan berjalan.

Selain fitur utama dalam sistem rental, website ini juga menyediakan halaman review dan kritik & saran (krisar), di mana pengguna dapat memberikan ulasan mengenai pengalaman mereka dalam menggunakan layanan rental ini. Hal ini bertujuan untuk meningkatkan kualitas layanan berdasarkan masukan langsung dari pelanggan.


Kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam pengembangan proyek ini, baik dari segi teknis maupun non-teknis. Semoga modul ini dapat menjadi referensi yang bermanfaat bagi pengembang, mahasiswa, serta siapa saja yang tertarik dalam membangun sistem rental berbasis web.

Akhir kata, kami berharap proyek ini dapat terus berkembang dan memberikan manfaat bagi masyarakat luas dalam memenuhi kebutuhan rental elektronik dan gadget secara mudah dan efisien.

Malang, 25 Februari 2025

Penulis

BAB I PENDAHULUAN

 WebRK	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: <i>Feb. 2025</i>	TI	SE	04	V	Page 1 - 27
	Program Keahlian: TI-SE-TE	<i>Website Rental</i> Kamera	Rev.: 06/11/2024	Dibuat oleh: Jesica / Eko				

1.1. Pengantar

Di era digital seperti saat ini apalagi dengan banyaknya tempat - tempat wisata, kebutuhan akan perangkat elektronik dan gadget semakin meningkat, semua orang ingin mengabadikan momen - momen berharga agar dapat menjadi kenangan yang berharga. Namun, tidak semua orang memiliki akses untuk membeli perangkat tersebut karena harganya yang relatif mahal. Oleh karena itu, layanan rental kamera menjadi solusi yang efektif dan ekonomis bagi masyarakat yang membutuhkan akses sementara terhadap perangkat berkualitas tinggi tanpa harus membelinya.


Dengan perkembangan teknologi, sistem rental berbasis web menjadi alternatif lebih praktis dibandingkan metode konvensional. Pengguna tidak perlu lagi untuk melakukan survei ketempat - tempat rental kamera, tetapi mereka bisa melakukan booking secara online, melihat daftar produk yang disewakan dan membandingkan harga secara online. Oleh karena itu, dibuatlah website rental kamera berbasis Flask dan SQLite yang bertujuan untuk memberikan pengalaman penyewaan yang lebih mudah, transparan, dan terorganisir.

Dari banyaknya aplikasi maupun bahasa pemrograman yang digunakan untuk membuat sebuah web, saya memilih untuk menggunakan bahasa pemrograman Python dengan menggunakan Microframework Flask. Dalam pengembangan web, istilah "microframework" merujuk pada kerangka kerja (framework) aplikasi web yang dirancang dengan ukuran yang lebih kecil dan fungsionalitas yang lebih minimalis dibandingkan dengan kerangka kerja "full-stack". Flask adalah kerangka kerja (framework) aplikasi web mikro (microframework) yang menyediakan fungsionalitas inti yang minimal dan ditulis dalam bahasa pemrograman Python.

Flask diciptakan oleh Armin Ronacher dari Pocoo, sebuah kelompok penggemar Python internasional yang dibentuk pada tahun 2004. Menurut Ronacher, ide ini awalnya adalah lelucon April Mop yang cukup populer untuk dijadikan aplikasi serius. Nama ini merupakan plesetan dari framework Bottle yang sudah ada sebelumnya. Sejak tahun 2018, data dan objek terkait Flask dapat dirender dengan **Bootstrap**. Flask telah menjadi populer di kalangan penggemar Python. Pada Oktober 2020, Flask memiliki jumlah bintang kedua terbanyak di GitHub di antara framework pengembangan web Python, hanya sedikit di belakang Django, dan terpilih sebagai framework web paling populer dalam Python Developers Survey selama bertahun-tahun antara 2018 dan 2022.

SQLite adalah sistem manajemen basis data relasional (RDBMS) yang tertanam. Ini berarti SQLite tidak memerlukan server terpisah dan menyimpan seluruh basis data dalam satu file. SQLite sangat ringan dan portabel, sehingga ideal untuk aplikasi yang membutuhkan basis data lokal. SQLite sering digunakan dalam aplikasi seluler, aplikasi desktop, dan perangkat tertanam. Dalam pengembangan web, SQLite berguna untuk aplikasi skala kecil atau pengujian. Keuntungan utama dari SQLite adalah kemudahan penggunaan, ketiadaan konfigurasi server, dan portabilitasnya.

1.2. Tujuan

 WebRk	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: <i>Feb. 2025</i>	<div>TI</div>	<div>SE</div>	<div>04</div>	<div>V</div>	Page 2 - 27
	Program Keahlian: TI-SE-TE	<i>Website Rental</i> Kamera	Rev.: 06/11/2024	Dibuat oleh: <i>Jesica / Eko</i>				

Tujuan pembuatan modul Website Rental Camera :

1. Meningkatkan efisiensi bisnis rental.
2. Mempermudah pelanggan dalam proses penyewaan.
3. Menghindari kehilangan barang rental dengan sistem otomatis.

1.3. Manfaat

Manfaat pembuatan modul Website Rental Camera :

1. Kemudahan Akses – Pelanggan dapat melakukan pemesanan kapan saja dan di mana saja tanpa harus datang langsung ke lokasi rental.
2. Efisiensi Manajemen – Admin dapat mengelola produk, transaksi, dan data pelanggan dengan lebih mudah dan terorganisir.
3. Transaksi Transparan – Sistem otomatis menghitung biaya sewa, denda keterlambatan, dan deposit sehingga pelanggan mendapatkan informasi yang jelas.
4. Keamanan Data – Dengan sistem login dan autentikasi pengguna, data pelanggan dan transaksi lebih terjamin keamanannya.
5. Peningkatan Jangkauan Pasar – Website rental dapat diakses oleh lebih banyak orang, meningkatkan jumlah pelanggan potensial.

1.4. Sistematika Modul

Sistematika penulisan modul teaching aids Website Rental Camera ini tersusun atas empat bab, penjelasan untuk setiap bab adalah sebagai berikut

1. **Bab I. Pendahuluan**, membahas mengenai latar belakang pembuatan teaching aids, tujuan, manfaat pembuatan teaching aids, dan sistematika penulisan modul .
2. **Bab II. Spesifikasi Teaching Aids**, berisi tinjauan struktur aplikasi.
3. **Bab III. Backend**, berisi proyek-projek lab untuk membuat coding pemrograman python.
4. **Bab IV. Frontend**, berisi proyek-projek lab untuk membuat coding pemrograman HTML, CSS, JavaScript, Bootstrap.

BAB II

SPESIFIKASI TEACHING AID

Bab ini menjelaskan secara rinci struktur proyek aplikasi rental elektronik & gadget, termasuk bagaimana sistem dikembangkan, bagaimana data diproses, serta bagaimana komponen-komponen utama bekerja bersama untuk membentuk sistem yang fungsional dan efisien.

2.1. Struktur Folder & File

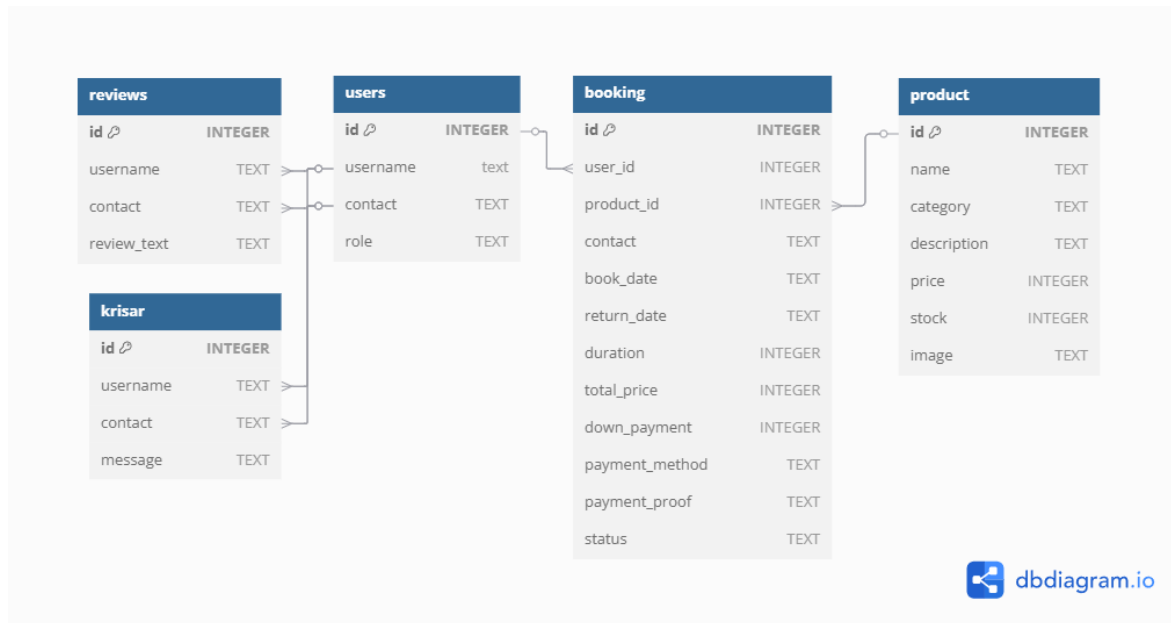
Struktur direktori aplikasi dibuat agar rapi, mudah dipahami, dan memudahkan pengelolaan kode dalam proyek ini. Berikut adalah struktur folder dan file utama yang digunakan dalam proyek ini:

Website Rental Camera/ --static/ --img/ --js/ --css/ --uploads/ --templates/ --user/ --base.html --index.html --menu.html --booking.html --login.html --sign_in.html --krisar.html --reviews.html --app.py --rental.db	<ul style="list-style-type: none"> ➤ Static = menyimpan file statis (CSS, JS, dan Gambar) ➤ Img, js,css = menyimpan gambar, file js, dan file css ➤ Uploads = menyimpan file yang di upload dari website seperti gambar produk dan bukti pembayaran ➤ Templates = untuk menyimpan file html ➤ User = untuk menyimpan file file yang detampilkan di user ➤ Base = template dasar yang memuat navbar dan footer untuk users ➤ Index = tampilan awal website ➤ Menu= menampilkan katalog produk ➤ Booking = form booking ➤ Login, sign in = untuk login dan register ➤ App = file utama untuk menjalankan aplikasi ➤ Rental.db = sebagai file database untuk menyimpan data web
--	--

2.2. Struktur Database

Berikut adalah struktur folder dan file utama yang digunakan dalam proyek ini:

users	id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT NOT NULL, password TEXT NOT NULL
bookings	id INTEGER PRIMARY KEY AUTOINCREMENT, user_id INTEGER NOT NULL, product_id INTEGER NOT NULL, contact TEXT NOT NULL, book_date TEXT NOT NULL, return_date TEXT NOT NULL, duration INTEGER NOT NULL, total_price INTEGER NOT NULL, down_payment INTEGER NOT NULL, payment_method TEXT NOT NULL, payment_proof TEXT NOT NULL, status TEXT DEFAULT 'Pending', FOREIGN KEY (user_id) REFERENCES users(id), FOREIGN KEY (product_id) REFERENCES products(id)
products	id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, category TEXT NOT NULL, description TEXT NOT NULL, price INTEGER NOT NULL, stock INTEGER DEFAULT 1, image TEXT
reviews	id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT NOT NULL, contact TEXT NOT NULL, review_text TEXT NOT NULL
krisar	id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT NOT NULL, contact TEXT NOT NULL, message TEXT NOT NULL



Gambar 2. 1 'rental.db' database schema

2.3 ALUR KERJA SISTEM

Sistem rental ini bekerja dengan mengikuti alur proses yang terstruktur agar penyewaan dapat berjalan dengan lancar. Berikut adalah alur kerja sistem rental:

1. Pelanggan mengunjungi website dan melihat daftar produk rental yang tersedia.
2. Pelanggan memilih produk yang ingin disewa dan mengisi formulir pemesanan dengan tanggal sewa dan durasi penggunaan.
3. Sistem menghitung total biaya secara otomatis, termasuk harga sewa dan deposit jika diperlukan.
4. Pelanggan mengunggah bukti pembayaran setelah menyelesaikan transaksi.
5. Admin menerima dan memverifikasi pembayaran, kemudian menyetujui pemesanan.
6. Barang dikirim atau diambil oleh pelanggan sesuai dengan ketentuan yang berlaku.
7. Pelanggan menggunakan barang selama durasi yang telah ditentukan.
8. Pelanggan mengembalikan barang tepat waktu.
9. Admin menyelesaikan transaksi dan memperbarui status peminjaman di sistem.

Proses ini dirancang untuk memastikan setiap penyewaan berjalan dengan transparan dan efisien, serta meminimalisir kesalahan dalam pengelolaan data.

2.4 PENJELASAN SETIAP KOMPONEN DALAM APLIKASI

2.4.1 Backend (Flask & SQLite)

Backend bertanggung jawab untuk mengatur semua data dan logika bisnis dalam sistem rental. Flask digunakan sebagai framework utama karena fleksibilitas dan kemudahan penggunaannya. Beberapa tugas utama backend dalam sistem ini meliputi:

1. Mengatur rute API yang menangani request dari frontend.
2. Mengelola data pengguna, transaksi, dan produk rental dengan SQLite.
3. Menjalankan logika bisnis, seperti perhitungan harga dan validasi pembayaran

2.4.2 Frontend (HTML, CSS, Bootstrap)

Frontend bertanggung jawab untuk menampilkan informasi kepada pengguna dan memungkinkan mereka berinteraksi dengan sistem. Beberapa fitur utama frontend meliputi:

1. Desain responsif menggunakan Bootstrap agar tampilan nyaman di semua perangkat.
2. Formulir interaktif untuk proses booking dan pembayaran.
3. Tampilan dashboard admin untuk mengelola transaksi dengan mudah.

2.4.3 Sistem Autentikasi & Role Management

Aplikasi ini menggunakan sistem autentikasi untuk mengelola akses pengguna dengan peran berbeda:

1. Admin memiliki akses penuh untuk mengelola produk, pengguna, dan transaksi.
2. Pelanggan hanya dapat melihat daftar produk, melakukan pemesanan, dan mengunggah bukti pembayaran.
3. Sistem login dan logout diterapkan untuk memastikan hanya pengguna terautentikasi yang dapat mengakses halaman tertentu.

2.4.4 Sistem Booking & Pembayaran


Sistem pemesanan dirancang agar mudah digunakan oleh pelanggan:

1. Booking hanya dapat dilakukan jika produk tersedia.
2. Sistem otomatis menghitung harga sewa berdasarkan durasi penggunaan.
3. Pelanggan harus mengunggah bukti pembayaran sebelum booking dikonfirmasi oleh admin.

BAB III

IMPLEMENTASI SISTEM (BACKEND)

Bab ini menjelaskan bagaimana sistem rental elektronik & gadget dikembangkan secara teknis, mencakup langkah-langkah dalam membangun backend, frontend, serta pengelolaan database. Setiap bagian akan dijelaskan secara detail agar pembaca dapat memahami bagaimana sistem ini diimplementasikan dari awal hingga dapat dijalankan.

 WebRk	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: <i>Feb. 2025</i>	TI	SE	04	V	Page 8 - 27
	Program Keahlian: TI-SE-TE	<i>Website</i> Kamera	<i>Rental</i> Rev.: 06/11/2024	Dibuat oleh: <i>Jesica / Eko</i>				

3.1 Pendahuluan

Implementasi sistem merupakan tahap di mana seluruh konsep perancangan yang telah dijelaskan sebelumnya diterapkan dalam bentuk kode dan sistem yang fungsional. Pada tahap ini, kita akan membangun:

1. Backend menggunakan Flask untuk menangani request dan mengelola data dalam SQLite.
2. Frontend menggunakan HTML, CSS, dan Bootstrap untuk antarmuka pengguna yang responsif.

Fitur utama seperti autentikasi pengguna, proses booking, dan verifikasi pembayaran

3.2 Persiapan Proyek

Sebelum mulai mengembangkan sistem, kita perlu menginstal beberapa perangkat lunak dan menyiapkan lingkungan pengembangan agar proyek dapat berjalan dengan baik.

3.2.1 Instalasi Python

Python adalah bahasa pemrograman yang digunakan dalam proyek ini. Untuk memastikan Python telah terinstal, gunakan perintah berikut di terminal atau command prompt:

```
python --version
```

Jika Python belum terinstal, silakan unduh dan instal dari situs resminya:
<https://www.python.org/downloads/>

3.2.2 Instalasi Flask

Flask adalah framework berbasis Python yang akan digunakan untuk membangun backend aplikasi ini. Untuk menginstal Flask, jalankan perintah berikut dalam virtual environment yang telah diaktifkan:

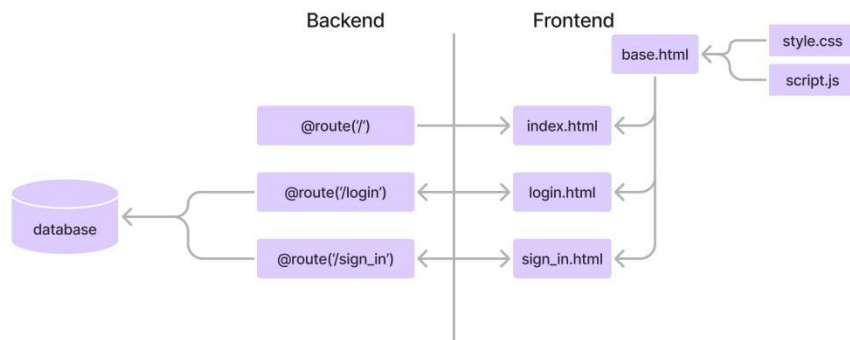
```
pip install flask
```

Setelah proses instalasi selesai, kita dapat mengecek apakah Flask telah terinstal dengan perintah berikut:

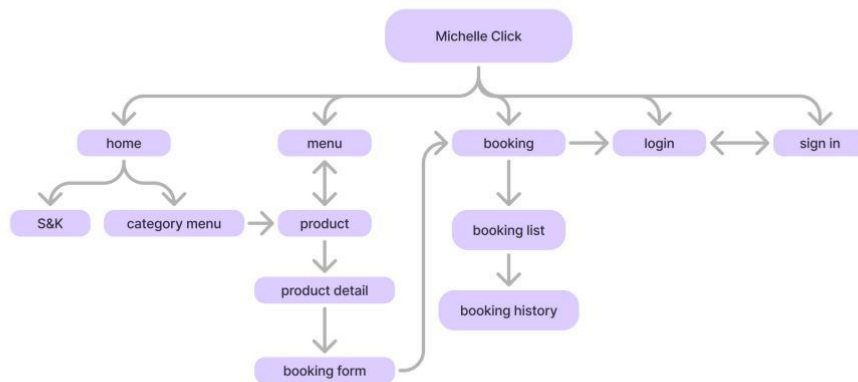
```
python -m flask --version
```

3.2.3 Site Map / Desain Sistem

Pembuatan desain sistem ini sangat berguna dalam pembangunan aplikasi karena dapat membantu pemahaman tentang hubungan antar halaman dan mempermudah proses coding karena terdapat panduan mengenai struktur sistem. Berikut adalah site map dan desain sistem.



Gambar 3. 1 Desain Sistem



Gambar 3. 2 Site Map

3.3 Implementasi sistem (backend)

Pada bagian ini, kita akan membahas bagaimana cara mengimplementasikan router dalam aplikasi rental kamera berbasis Flask. Router berfungsi sebagai penghubung antara permintaan pengguna dan tampilan halaman yang sesuai. Dalam sistem ini, kita akan membuat tiga router utama:

1. Router Home – Menampilkan halaman utama dengan daftar produk yang tersedia.
2. Router Login – Mengelola autentikasi pengguna agar dapat masuk ke sistem.
3. Router Sign In – Memungkinkan pengguna baru untuk mendaftar.

Setiap router akan menggunakan Flask dan SQLite sebagai basis data untuk mengelola informasi pengguna dan produk. Langsung ke tutorialnya :

1. Di dalam file main.py ketik kode dibawah ini

```

main.py x
1  from flask import Flask, render_template, url_for, request, redirect, flash, session
2  import sqlite3 as sql
3
4  app = Flask(__name__)
5
6  def init_db(): 1 usage
7      with sql.connect('rental.db') as conn:
8          cursor = conn.cursor()
9          cursor.execute('''
10             CREATE TABLE IF NOT EXISTS users (
11                 id INTEGER PRIMARY KEY AUTOINCREMENT,
12                 username TEXT NOT NULL UNIQUE,
13                 contact TEXT NOT NULL,
14                 role TEXT NOT NULL CHECK(role IN ('admin', 'user'))
15             )
16         ''')
17
18  def get_db_connection(): 14 usages
19      conn = sql.connect('rental.db')
20      conn.row_factory = sql.Row
21      return conn
22

```

Gambar 3. 3 main.py

- Baris 1 : untuk mengimport modul dari library flask
- Baris 2 : untuk mengimport modul sqlite3 dan 'as' untuk mengubah nama ke sql agar mempermudah proses coding
- Baris 4 : untuk inisialisasi aplikasi Flask
- Baris 6 - 16 : fungsi 'init_db' dibuat untuk membuka koneksi database sql dan membuat tabel di dalam 'rental.db'
- Baris 18 - 21 : menghubungkan database 'rental.db' dan mengatur hasil query agar bisa diakses sebagai dictionary

2. Route home

```

25
26 @app.route('/')
27 def home():
28     conn = get_db_connection()
29     is_admin = False
30
31     if 'user_id' in session:
32         user_role = conn.execute(sql: 'SELECT role FROM users WHERE id = ?', parameters: (session['user_id'],)).fetchone()
33         if user_role and user_role['role'] == 'admin':
34             is_admin = True
35
36     conn.close()
37     return render_template(template_name_or_list: 'index.html', is_admin=is_admin)

```

Gambar 3. 4 route('/')

Kode ini menangani halaman utama (/) dalam aplikasi Flask dan memeriksa apakah pengguna yang login adalah admin atau bukan. Jika pengguna adalah admin, maka halaman yang dirender bisa menampilkan fitur khusus admin.

- Baris 26 : membuat route untuk halaman utama
- Baris 27 : membuat fungsi home
- Baris 28 : menghubungkan dengan database
- Baris 29 : sebagai default bahwa pengguna bukan admin
- Baris 30 - 36 : mengecek adanya user_id dalam session, jika pengguna sudah login maka variabel user_role akan berjalan untuk memilih role dari tabel users sesuai user_id di session, dan jika role pengguna tersebut adalah admin maka is_admin akan berubah menjadi true.
- Baris 37 : untuk mengembalikan ke file index.html dan menyertakan variabel is_admin agar dapat digunakan di html.

3. Route register

```

39 @app.route(rule: '/register', methods=['GET', 'POST'])
40 def sign_in():
41     if request.method == 'POST':
42         username = request.form['username']
43         contact = request.form['contact']
44
45         with get_db_connection() as conn:
46             existing_user = conn.execute(sql: 'SELECT * FROM users WHERE username = ?', parameters: (username,)).fetchone()
47             if existing_user:
48                 flash(message: 'Username sudah digunakan! Silakan pilih username lain.', category: 'danger')
49                 return redirect(url_for('sign_in'))
50
51             conn.execute(sql: 'INSERT INTO users (username, contact, role) VALUES (?, ?, "user")',
52                         parameters: (username, contact))
53             conn.commit()
54             flash(message: 'Registrasi berhasil! Silakan login.', category: 'success')
55             return redirect(url_for('login'))
56
57     return render_template('sign_in.html')

```

Gambar 3. 5 route('/register')

- Baris 39 : membuat route /register, menangani dua metode http yakni GET (mengambil data dari server tanpa mengubahnya) dan POST (mengirim, menyimpan, dan mengubah data)
- Baris 40 : membuat fungsi sign_in yang akan dieksekusi ketika pengguna mengakses /register
- Baris 41 - 43 : untuk memproses form registrasi, jika metode yang digunakan adalah POST maka : mengambil username yang dikirim dari formulir (request.form['username']) ; mengambil contact dari formulir (request.form['contact'])

- Baris 45 - 53 : untuk menutup koneksi database secara otomatis, dan menjalankan kode sql 'SELECT' untuk mencari pengguna dengan yang sama serta jika username sudah ada maka akan menampilkan pesan error menggunakan flash() dan mengalihkan pengguna ke halaman registrasi. Lalu akan menyimpan data pengguna baru seperti username, contact, dan role default (user) ke database menggunakan code sql 'INSERT'
- Baris 54 - 55 : ketika registrasi berhasil akan menampilkan pesan sukses dengan flash() dan akan mengalihkan pengguna ke halaman login.
- Baris 57 : Jika pengguna mengakses /register dengan metode GET, halaman registrasi (sign_in.html) akan ditampilkan.

4. Route Login

```

60 @app.route(rule='/login', methods=['GET', 'POST'])
61 def login():
62     if request.method == 'POST':
63         username = request.form['username']
64         contact = request.form['contact']
65         conn = get_db_connection()
66         user = conn.execute(sql: 'SELECT * FROM users WHERE username = ?', parameters: (username,)).fetchone()
67         conn.close()
68
69         if user and user['contact'] == contact:
70             session['user_id'] = user['id']
71             session['username'] = user['username']
72             flash(message: 'Login Berhasil!', category: 'success')
73             return redirect(url_for('home'))
74         else:
75             flash(message: 'Username atau nomor salah!', category: 'danger')
76     return render_template('login.html')

```

Gambar 3. 6 route('/login')

- Baris 60 : Menentukan rute /login yang bisa diakses dengan metode GET dan POST. GET untuk menampilkan halaman login. POST untuk memproses login setelah pengguna mengirimkan form.
- Baris 62 - 67 : Jika pengguna mengirim form login, maka akan otomatis mengambil username dan contact yang dikirim dari form dan nantinya akan dihubungkan ke database untuk mencari data user dari username.
- Baris 69 - 75 : Untuk mengecek apakah user ditemukan dan kontakannya sesuai, jika sesuai maka akan lanjut ke session untuk menyimpan informasi user yang login, memunculkan flash() pesan sukses dan redirect ke halaman home. Tetapi jika salah akan muncul pesan flash() danger.
- Baris 76 : menampilkan halaman login

5. Route Logout dan inisialisasi Flask

```

78 @app.route('/logout')
79 def logout():
80     session.clear()
81     flash(message='Logout berhasil!', category='info')
82     return redirect(url_for('home'))
83
84 if __name__ == '__main__':
85     init_db()
86     app.run(debug=True, host='0.0.0.0')

```

Gambar 3. 7 route('/logout')

- `@app.route('/logout')` → Menentukan rute untuk logout.
- `session.clear()` → Menghapus semua data sesi pengguna (logout user).
- `flash('Logout berhasil!', category='info')` → Menampilkan pesan notifikasi.
- `return redirect(url_for('home'))` → Mengarahkan user kembali ke halaman utama (home).

BAB IV

IMPLEMENTASI SISTEM (FRONTEND)

Setelah pembuatan backend, kita akan beralih dalam pembuatan frontend yakni tampilan halaman yang akan dibuat dengan menggunakan CSS, JavaScript, dan Bootstrap. Tujuannya adalah untuk menciptakan pengalaman pengguna yang intuitif, responsif, dan mudah digunakan.

4.1 Pendahuluan

Frontend merupakan bagian dari sistem yang berinteraksi langsung dengan pengguna. Dalam proyek ini, frontend dikembangkan menggunakan teknologi berikut:

1. HTML untuk struktur halaman.
2. CSS & Bootstrap untuk desain dan tata letak yang responsif.
3. JavaScript untuk meningkatkan interaktivitas.

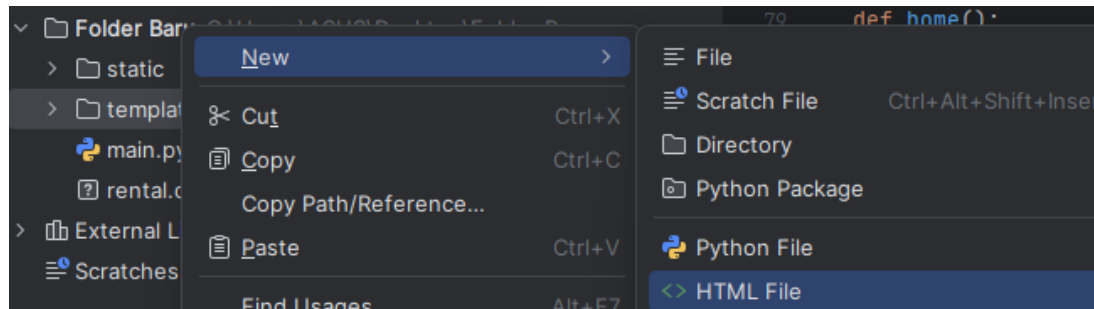
Dengan kombinasi teknologi ini, sistem dapat memberikan pengalaman pengguna yang optimal di berbagai perangkat.

4.2 Implementasi Frontend

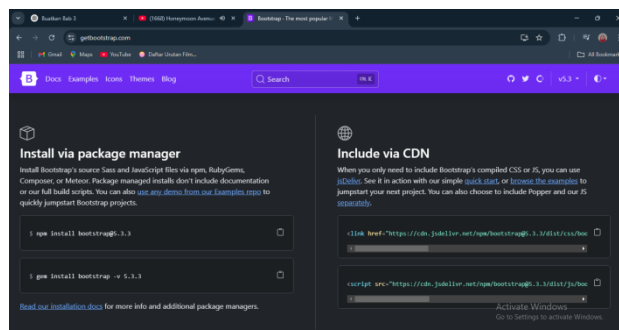
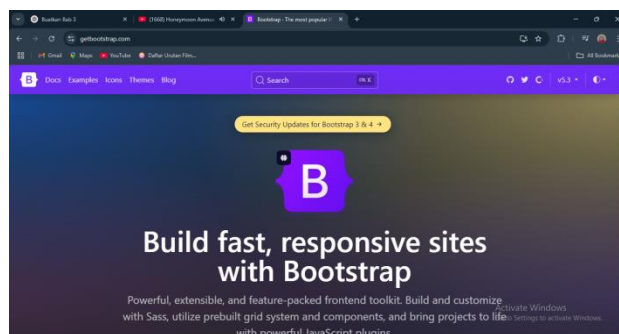
Sebelum membuat file html yang lain, kita akan membuat base.html terlebih dahulu. Base.html ini berfungsi sebagai file utama yang akan digunakan oleh file lainnya tanpa harus mengetik lagi di file berikutnya.

4.2.1 Pembuatan base.html

1. Buat file di dalam folder templates, dengan klik kanan dan pilih HTML File



2. Dalam base.html kita akan membuat navbar, di Bootstrap terdapat beberapa pilihan navbar yang bisa kita gunakan. Pertama kita masuk ke website bootstrap untuk menghubungkannya dengan file html



3. Copy link bootstrap tersebut dan Paste di html seperti di bawah ini.

Setelah itu kita kembali ke halaman bootstrap > docs > navbar. Di halaman tersebut terdapat banyak variasi navbar, setelahnya pilih model navbar dan copy paste di file html

```

2 <html lang="en">
3 <head>
4 <title>
5 </title>
6 </head>
7 <body>
8 <div class="container">
9 <div class="nav navbar-expand-lg navbar-light bg-transparent fixed-top">
10 <div class="container-fluid">
11 <a class="navbar-brand" href="{url_for('home')}">Logo</a>
12 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" data-bs-keyboard="true" data-bs-toggle="collapse">
13 <span class="navbar-toggler-icon"></span>
14 </button>
15 <div class="collapse navbar-collapse" id="navbarNav">
16 <ul class="navbar-nav ms-auto">
17 <li class="nav-item"><a class="nav-link" href="{url_for('home')}">Home</a></li>
18 <li class="nav-item"><a class="nav-link" href="{url_for('menu')}">Menu</a></li>
19 <li class="nav-item"><a class="nav-link" href="{url_for('booking_list')}">Bookings</a></li>
20 {%- if 'User_id' not in session %}
21 <li class="nav-item"><a class="nav-link" href="{url_for('login')}">Login</a></li>
22 <li class="nav-item"><a class="nav-link" href="{url_for('sign_in')}">Sign Up</a></li>
23 {%- else %}
24 <li class="nav-item"><a class="nav-link" href="{url_for('logout')}">Logout</a></li>
25 {%- endif %}
26 </ul>
27 </div>
28 </div>
29 </div>
30 </body>
31 </html>

```

Lalu tambahkan styling css dan javascript agar navbar menjadi transparan dan ketika di scroll maka akan berubah backgroundnya. Untuk style css dan javascript sendiri bisa menggunakan file css/js atau menggunakan tag html `<style>` / `<script>`

CSS>>

```

2 <html lang="en">
3 <head>
4 <title>
5 </title>
6 </head>
7 <body>
8 <div class="container">
9 <div class="nav navbar-expand-lg navbar-light bg-transparent fixed-top">
10 <div class="container-fluid">
11 <a class="navbar-brand" href="{url_for('home')}">Logo</a>
12 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" data-bs-keyboard="true" data-bs-toggle="collapse">
13 <span class="navbar-toggler-icon"></span>
14 </button>
15 <div class="collapse navbar-collapse" id="navbarNav">
16 <ul class="navbar-nav ms-auto">
17 <li class="nav-item"><a class="nav-link" href="{url_for('home')}">Home</a></li>
18 <li class="nav-item"><a class="nav-link" href="{url_for('menu')}">Menu</a></li>
19 <li class="nav-item"><a class="nav-link" href="{url_for('booking_list')}">Bookings</a></li>
20 {%- if 'User_id' not in session %}
21 <li class="nav-item"><a class="nav-link" href="{url_for('login')}">Login</a></li>
22 <li class="nav-item"><a class="nav-link" href="{url_for('sign_in')}">Sign Up</a></li>
23 {%- else %}
24 <li class="nav-item"><a class="nav-link" href="{url_for('logout')}">Logout</a></li>
25 {%- endif %}
26 </ul>
27 </div>
28 </div>
29 </div>
30 </body>
31 </html>

```

```

12 <style>
13 .navbar {
14     background-color: transparent !important;
15     transition: background-color 0.3s ease-in-out;
16 }
17
18 .navbar a {
19     color: black !important; /* Warna teks navbar awal */
20     font-weight: bold;
21 }
22
23 /* Navbar saat di-scroll */
24 .navbar-scrolled {
25     background-color: rgba(0, 0, 0, 0.8) !important; /* Navbar jadi hitam transparan */
26 }
27
28 .navbar-scrolled a {
29     color: white !important; /* Teks berubah jadi putih saat di-scroll */
30 }
31
32 .navbar a:hover {
33     color: #FFD700 !important; /* Warna emas saat hover */
34     text-decoration: underline;
35 }

```

JavaScript>>

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0X
</script>
</body>
</html>

```

Setelah itu kita akan membuat footer dengan mengetikkan kode berikut

Html>>

```

<html lang="en">
<body>
<main>
    {% block content %}{% endblock %}
</main>


<footer class="footer mt-5 py-4 bg-dark text-light">
    <div class="container">
        <div class="row">
            <div class="col-md-3">
                <h5>Michelle Click!</h5>
                <ul class="list-unstyled">
                    <li><a href="/" class="text-light text-decoration-none">Home</a></li>
                </ul>
            </div>
            <div class="col-md-3">
                <h5>Layanan Pelanggan</h5>
                <ul class="list-unstyled">
                    <li><a href="{{ url_for('krisar') }}" class="text-light text-decoration-none">Kritik & Saran</a></li>
                    <li><a href="{{ url_for('reviews') }}" class="text-light text-decoration-none">Reviews</a></li>
                </ul>
            </div>
            <div class="col-md-3">
                <h5>Kategori Rental</h5>
                <ul class="list-unstyled">
                    <li><a href="{{ url_for('menu', category='Camera') }}" class="text-light text-decoration-none">Kamera</a></li>
                    <li><a href="{{ url_for('menu', category='Audio') }}" class="text-light text-decoration-none">Audio</a></li>
                    <li><a href="{{ url_for('menu', category='Drone') }}" class="text-light text-decoration-none">Drone</a></li>
                </ul>
            </div>
        </div>
    </div>
</footer>

```

```

<html lang="en">
<body>
<footer class="footer mt-5 py-4 bg-dark text-light">
    <div class="container">
        <div class="row">
            <div class="col-md-3">
                <h5>Hubungi Kami</h5>
                <div class="social-links">
                    <a href="#" class="text-light me-2"><i class="fab fa-facebook-f"></i></a>
                    <a href="#" class="text-light me-2"><i class="fab fa-twitter"></i></a>
                    <a href="#" class="text-light me-2"><i class="fab fa-instagram"></i></a>
                    <a href="https://wa.me/6281234567890" class="text-light me-2"><i class="fab fa-whatsapp"></i></a>
                </div>
            </div>
            <div class="col-md-3">
                <h5>Hubungi Kami</h5>
                <div class="social-links">
                    <a href="#" class="text-light me-2"><i class="fab fa-facebook-f"></i></a>
                    <a href="#" class="text-light me-2"><i class="fab fa-twitter"></i></a>
                    <a href="#" class="text-light me-2"><i class="fab fa-instagram"></i></a>
                    <a href="https://wa.me/6281234567890" class="text-light me-2"><i class="fab fa-whatsapp"></i></a>
                </div>
            </div>
            <div class="col-md-3">
                <h5>Hubungi Kami</h5>
                <div class="social-links">
                    <a href="#" class="text-light me-2"><i class="fab fa-facebook-f"></i></a>
                    <a href="#" class="text-light me-2"><i class="fab fa-twitter"></i></a>
                    <a href="#" class="text-light me-2"><i class="fab fa-instagram"></i></a>
                    <a href="https://wa.me/6281234567890" class="text-light me-2"><i class="fab fa-whatsapp"></i></a>
                </div>
            </div>
        </div>
    </div>
</footer>

```

 WebRk	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: <i>Feb.</i> 2025	TI	SE	04	V	Page 18 - 27
	Program Keahlian: TI-SE-TE	<i>Website</i> Kamera	<i>Rental</i> Rev.: 06/11/2024	Dibuat oleh: Jesica / Eko				

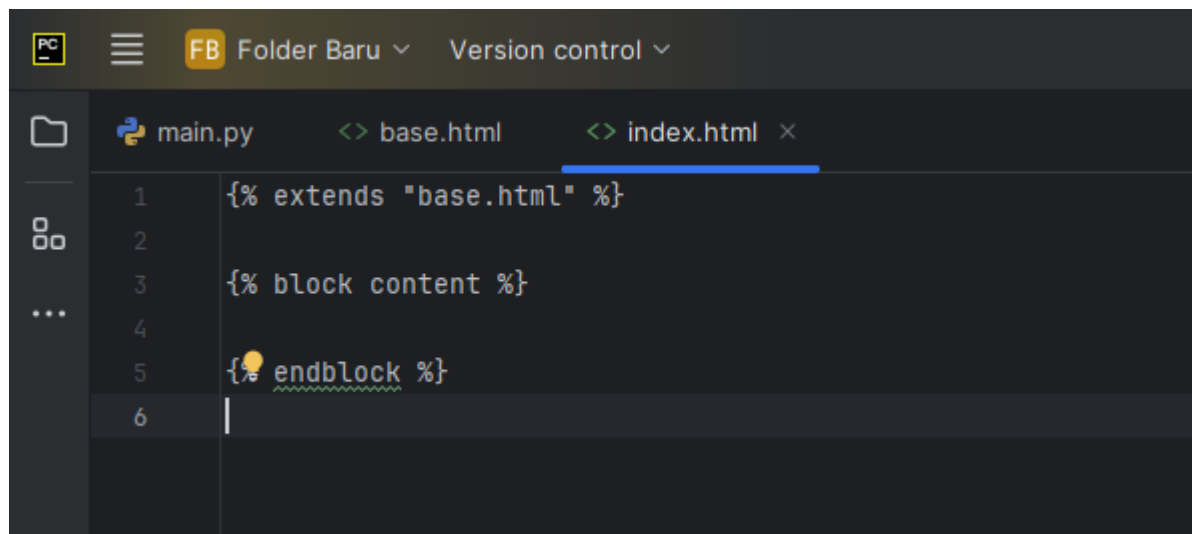
4.2.2 Pembuatan index.html

Pada file base.html terdapat kode seperti ini :

```
<main>
    {% block content %}{% endblock %}
</main>
```

Kode ini adalah kode Jinja, yakni kode yang digunakan untuk menghubungkan kode python dan kode html. Ketika base.html di teruskan ke file index.html, isi coding akan berada di kode jinja tersebut dan akan ditampilkan di halaman routing.

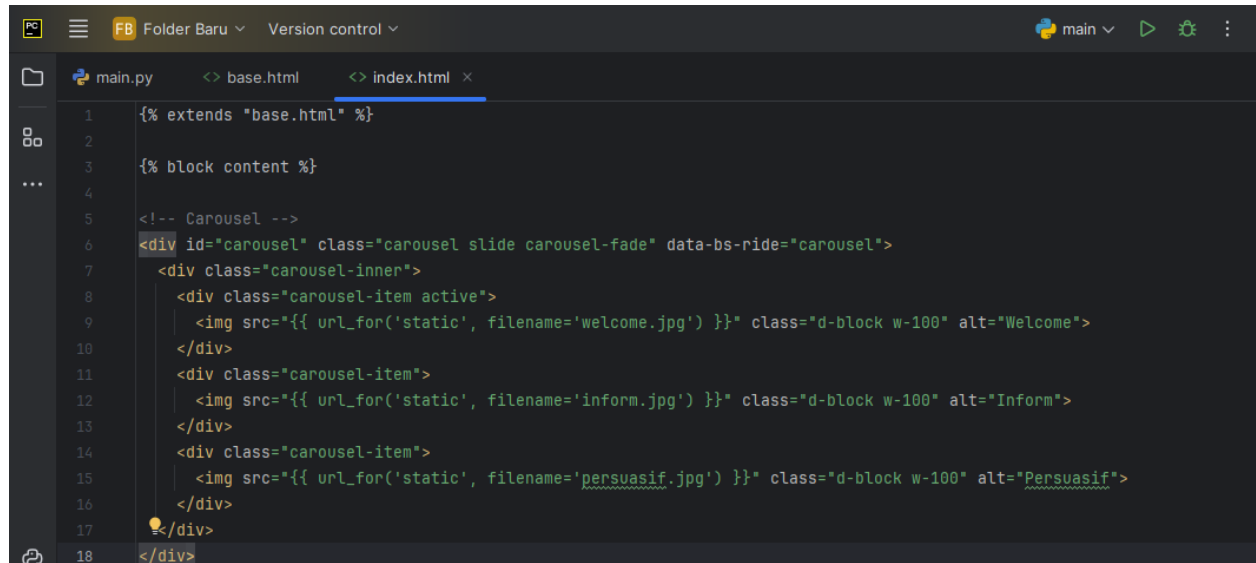
1. Pada bagian awal index.html ketik kode ini



```
1 {% extends "base.html" %}
2
3 {% block content %}
4
5 {% endblock %}
6
```

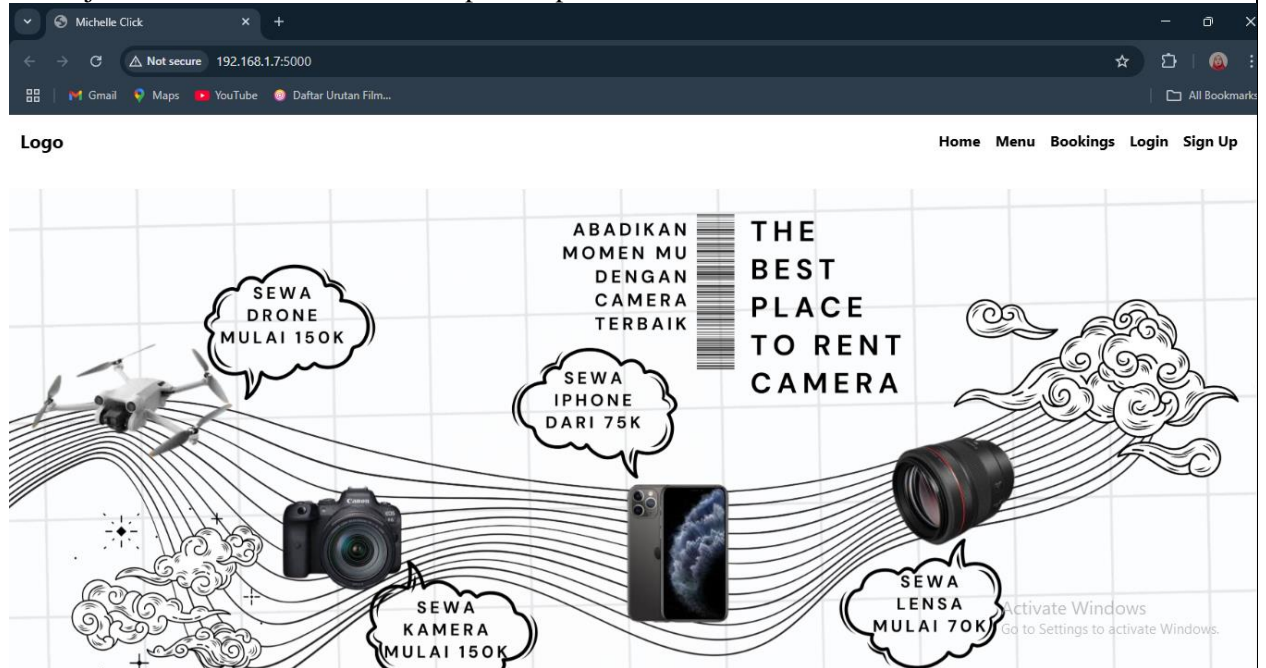
Kode di baris satu berfungsi untuk meng-extends base.html, dan baris 4 adalah tempat codingan kita nantinya.

2. Di bagian halaman utama kita akan menampilkan banner website yang secara otomatis dengan menggunakan kode bootstrap carousel



```
1 {% extends "base.html" %}
2
3 {% block content %}
4
5 <!-- Carousel -->
6 <div id="carousel" class="carousel slide carousel-fade" data-bs-ride="carousel">
7   <div class="carousel-inner">
8     <div class="carousel-item active">
9       
10     </div>
11     <div class="carousel-item">
12       
13     </div>
14     <div class="carousel-item">
15       
16     </div>
17   </div>
18 </div>
```

Jika dijalankan maka akan muncul tampilan seperti ini



3. Pada bagian bawah banner kita akan memunculkan menu kategori produk seperti iphone, kamera dan lain-lain.

Html>>

```

PC  FB Folder Baru  Version control
main.py  base.html  index.html x

22  <!-- Kategori Rental -->
23  <div class="container text-center">
24    <div class="row row-cols-2 row-cols-md-3 row-cols-lg-5 align-items-center">
25      <div class="col">
26        <a href="{{ url_for('menu', category='Audio') }}" class="text-decoration-none">
27          
28          <h6 class="mt-2">Audio</h6>
29        </a>
30      </div>
31      <div class="col">
32        <a href="{{ url_for('menu', category='Camera') }}" class="text-decoration-none">
33          
34          <h6 class="mt-2">Camera</h6>
35        </a>
36      </div>
37      <div class="col">
38        <a href="{{ url_for('menu', category='Drone') }}" class="text-decoration-none">
39          
40          <h6 class="mt-2">Drone</h6>
41        </a>
42      </div>
43      <div class="col">
44        <a href="{{ url_for('menu', category='iPhone') }}" class="text-decoration-none">
45          
46          <h6 class="mt-2">iPhone</h6>
47        </a>
48      </div>
49      <div class="col">
50        <a href="{{ url_for('menu', category='Lensa') }}" class="text-decoration-none">
51          
52          <h6 class="mt-2">Lensa</h6>
53        </a>
54      </div>
55    </div>
56  </div>
57

```

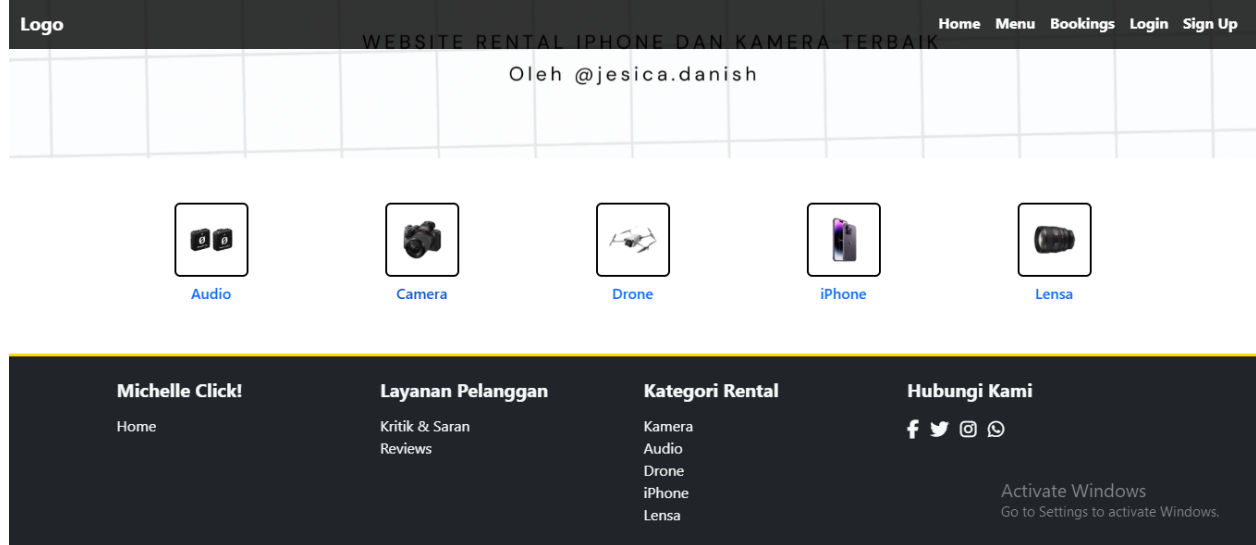
Css>>

```

36
37  .category-img {
38    width: 80px; /* Ukuran gambar */
39    height: 80px;
40    border: 2px solid #000;
41    border-radius: 10px;
42    padding: 5px;
43    transition: transform 0.3s ease, box-shadow 0.3s ease; /* Animasi lembut */
44  }
45
46  .category-img:hover {
47    transform: scale(1.1); /* Memperbesar gambar sedikit saat hover */
48    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3); /* Efek bayangan untuk tampilan timbul */
49  }

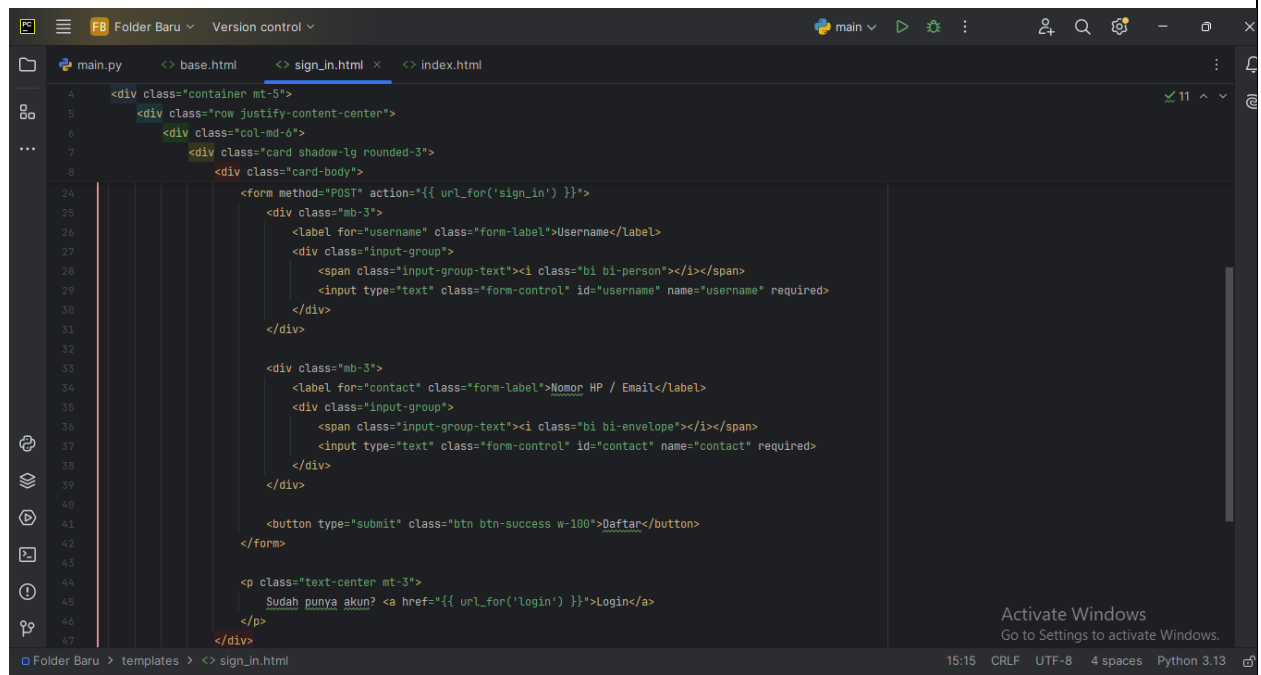
```

Output>>



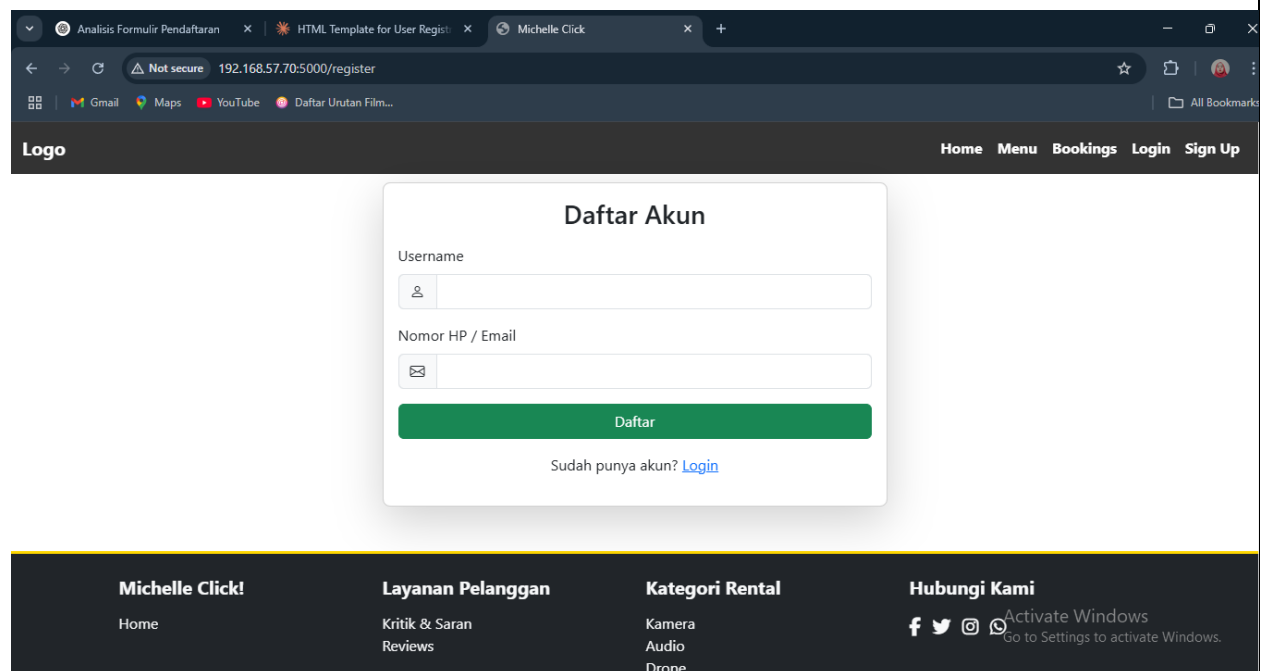
4.2.3 Pembuatan sign_in.html

1. Lakukan hal yang sama seperti index.html di awal
- 2.



```
4 <div class="container mt-5">
5   <div class="row justify-content-center">
6     <div class="col-md-6">
7       <div class="card shadow-lg rounded-3">
8         <div class="card-body">
9           <form method="POST" action="{{ url_for('sign_in') }}">
10            <div class="mb-3">
11              <label for="username" class="form-label">Username</label>
12              <div class="input-group">
13                <span class="input-group-text"><i class="bi bi-person"></i></span>
14                <input type="text" class="form-control" id="username" name="username" required>
15              </div>
16            </div>
17            <div class="mb-3">
18              <label for="contact" class="form-label">Nomor HP / Email</label>
19              <div class="input-group">
20                <span class="input-group-text"><i class="bi bi-envelope"></i></span>
21                <input type="text" class="form-control" id="contact" name="contact" required>
22              </div>
23            </div>
24            <button type="submit" class="btn btn-success w-100">Daftar</button>
25          </form>
26          <p class="text-center mt-3">
27            Sudah punya akun? <a href="{{ url_for('login') }}">Login</a>
28          </p>
29        </div>
30      </div>
31    </div>
32  </div>
33</div>
```

Kode diatas merupakan tampilan form sign_in dengan bootstrap yang akan memunculkan output seperti berikut



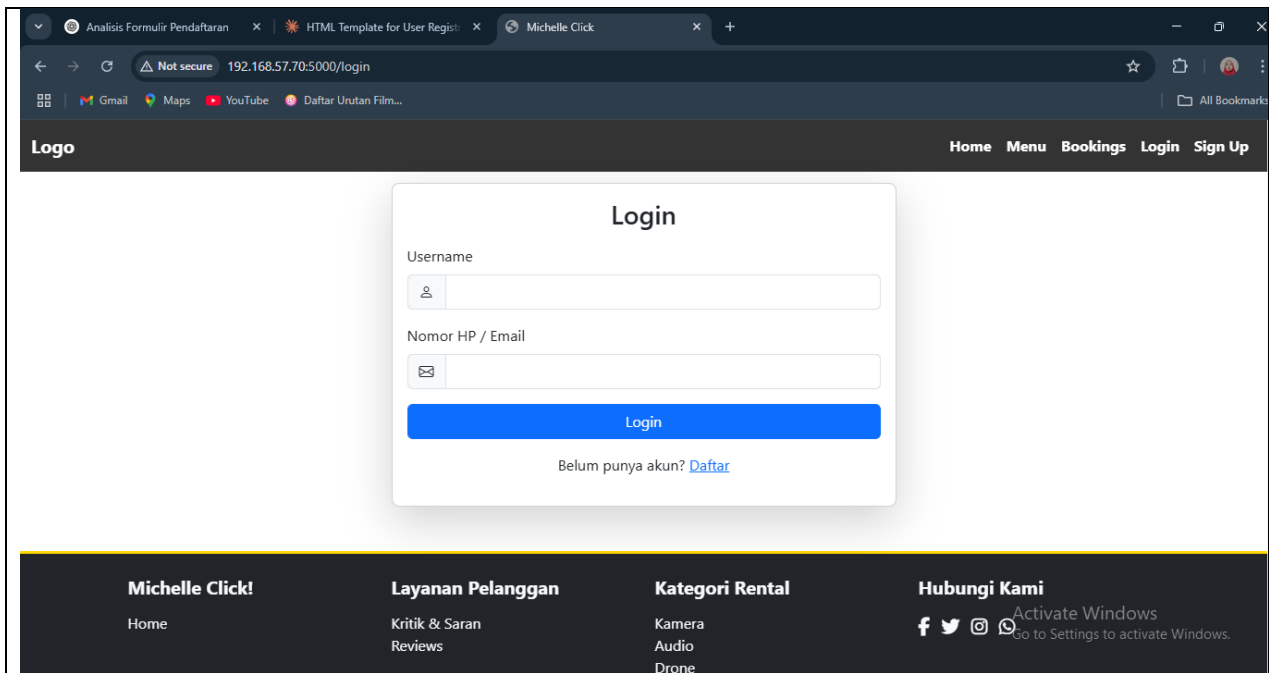
4.2.4 Pembuatan login.html

1.


```
1 {% extends "base.html" %}
2
3 {% block content %}
4 <div class="container mt-5">
5   <div class="row justify-content-center">
6     <div class="col-md-6">
7       <div class="card shadow-lg rounded-3">
8         <div class="card-body">
9           <h3 class="text-center mb-3">Login</h3>
10
11           <!-- Flash Messages -->
12           {% with messages = get_flashed_messages(with_categories=true) %}
13             {% if messages %}
14               {% for category, message in messages %}
15                 <div class="alert alert-{{ category }} alert-dismissible fade show" role="alert">
16                   {{ message }}
17                   <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="close"></button>
18                 </div>
19               {% endfor %}
20             {% endif %}
21           {% endif %}
22           <!-- Form Login -->
23           <form method="POST" action="{{ url_for('login') }}">
24             <div class="mb-3">
25               <label for="username" class="form-label">Username</label>
26               <div class="input-group">
27                 <span class="input-group-text"><i class="bi bi-person"></i></span>
28                 <input type="text" class="form-control" id="username" name="username" required>
29               </div>
30             </div>
31           </form>
32         </div>
33       </div>
34     </div>
35   </div>
36 </div>
```


```
4 <div class="container mt-5">
5   <div class="row justify-content-center">
6     <div class="col-md-6">
7       <div class="card shadow-lg rounded-3">
8         <div class="card-body">
9           <div class="input-group">
10             <span class="input-group-text"><i class="bi bi-person"></i></span>
11             <input type="text" class="form-control" id="username" name="username" required>
12           </div>
13           <small id="usernameHelp" class="text-danger"></small>
14         </div>
15         <div class="mb-3">
16           <label for="contact" class="form-label">Nomor HP / Email</label>
17           <div class="input-group">
18             <span class="input-group-text"><i class="bi bi-envelope"></i></span>
19             <input type="text" class="form-control" id="contact" name="contact" required>
20           </div>
21           <small id="contactHelp" class="text-danger"></small>
22         </div>
23         <button type="submit" class="btn btn-primary w-100">Login</button>
24       </div>
25       <div class="text-center mt-3">
26         <p>Belum punya akun? <a href="{{ url_for('sign_in') }}">Daftar</a></p>
27       </div>
28     </div>
29   </div>
30 </div>
```

Kode diatas akan memunculkan output



Lebih lengkapnya, silahkan cek di website di bawah sini

<https://github.com/jesicadanishara/Michelle-Click>

 WebRk	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: <i>Feb.</i> 2025	<div>TI</div>	<div>SE</div>	<div>04</div>	<div>V</div>	Page 25 - 27
	Program Keahlian: TI-SE-TE	<i>Website</i> Kamera	<i>Rental</i> Rev.: 06/11/2024	Dibuat oleh: <i>Jesica / Eko</i>				

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