**MINI PROJEC**

**WEB FASHION STORE**



**2025**

# KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga modul ini dapat disusun dengan baik. Modul ini disusun sebagai panduan dalam pembuatan dan pengembangan Web Fashion Store, yang diharapkan dapat menjadi referensi bagi para pengembang web, pelajar, mahasiswa, maupun siapa saja yang tertarik dalam dunia e-commerce berbasis fashion.

Web Fashion Store merupakan platform digital yang dirancang untuk memudahkan penjual dan pembeli dalam berinteraksi serta melakukan transaksi secara online. Dalam modul ini, akan dibahas berbagai aspek pengembangan web, mulai dari desain tampilan, implementasi fitur, hingga integrasi sistem yang mendukung pengalaman pengguna yang optimal.

Kami berharap modul ini dapat memberikan wawasan serta panduan praktis dalam membangun Web Fashion Store yang modern, responsif, dan fungsional. Kami juga menyadari bahwa modul ini masih jauh dari kesempurnaan, oleh karena itu, kami sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa mendatang.

Akhir kata, semoga modul ini dapat memberikan manfaat dan inspirasi bagi para pembaca. Terima kasih atas segala dukungan dan partisipasi dalam proses penyusunan modul ini.

Malang, 26 Febuari 2025

# Penulis

# BAB I

# PENDAHULUAN

**1.1. Latar Belakang**

Dalam era digital saat ini, pembuatan website menjadi keterampilan yang sangat penting, terutama dalam industri fashion. Website tidak hanya digunakan untuk keperluan bisnis, tetapi juga sebagai sarana promosi, komunikasi dengan pelanggan, dan transaksi online. 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 pengelolaan 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 Fashion Store yang memiliki performa baik dengan pengelolaan data yang efektif.

Modul ini disusun untuk memberikan panduan langkah demi langkah dalam membangun website Fashion Store 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 produk dan transaksi dengan SQLite.

**1.2. Tujuan**

Tujuan dari penyusunan modul ini adalah:

1. Memberikan pemahaman dasar tentang Flask dan cara penggunaannya dalam pembuatan website Fashion Store.
2. Menjelaskan struktur proyek Flask dan fungsinya masing-masing.
3. Mengajarkan cara menghubungkan Flask dengan database SQLite untuk mengelola produk fashion dan transaksi pelanggan.
4. Membantu siswa dan praktisi dalam membuat website Fashion Store yang dapat berjalan secara online.

**1.3. Manfaat**

Dengan memperlajari modul ini, pembaca diharapkan dapat:

1. Memahami dasar-dasar pengembangan web menggunakan Flask dalam konteks Fashion Store.
2. Mampu membuat proyek web Fashion Store dengan struktur yang baik.
3. Mengelola database produk dan transaksi menggunakan SQLite dalam website Flask.
4. Mengembangkan website Fashion Store yang interaktif dan dinamis untuk pengalaman belanja online yang lebih baik.

**1.4. Sistematika Modul**

Modul ini disusun menjadi beberapa bab dengan sistematika sebagai berikut:

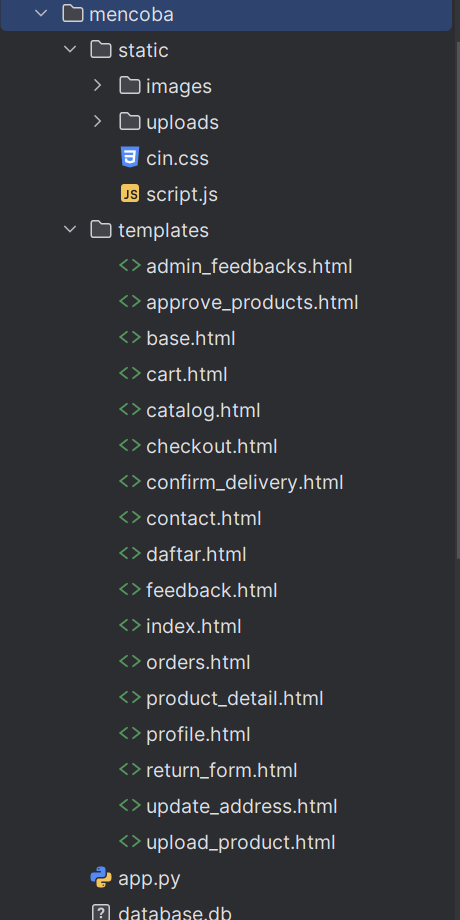
1. **Bab I: Pendahuluan** - Membahas latar belakang, tujuan, manfaat, dan sistematika penulisan modul.
2. **Bab II: Struktur Proyek Flask** - Menguraikan struktur direktori proyek Flask dan desain sistem Fashion Store.
3. **Bab III: Backend dengan Microframework Flask Python** - Membahas cara mengelola database produk dan transaksi dengan SQLite dalam aplikasi Flask.
4. **Bab IV: Frontend dengan HTML, CSS, dan JavaScript** - Membahas cara mengelola tampilan website Fashion Store menggunakan teknologi frontend.

# ****BAB II****

# ****TEKNOLOGI DAN FITUR WEB FASHION STORE****

**2.1.Struktur Direktori Proyek**

Dalam membangun website Fashion Store menggunakan Flask, penting untuk memiliki struktur proyek yang rapi agar pengelolaan kode lebih mudah dan terorganisir. Berikut adalah struktur direktori proyek Flask yang akan digunakan:

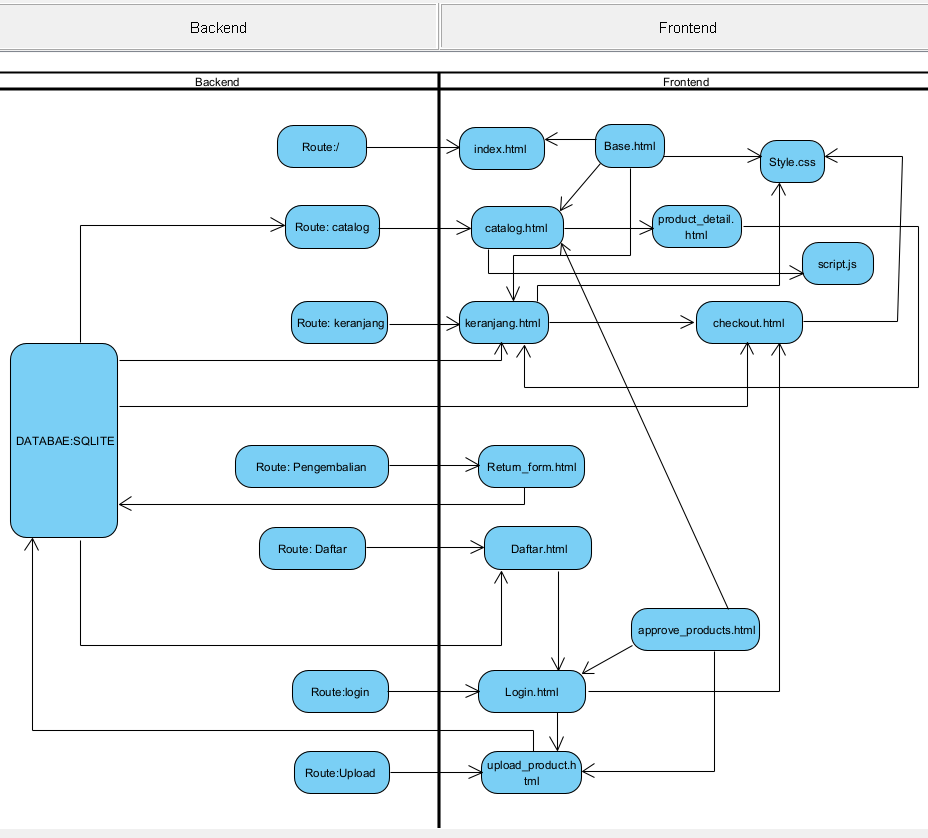
****

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.
* App.py File utama yang menjalankan website Flask,menangani request, dan menghubungkan ke database.
* Database.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;

SQLite3 berperan sebagai database, yang menyimpan data-data yang telah dimasukkan oleh pengguna.

# ****BAB III****

# ****PENGEMBANGAN DAN IMPLEMENTASI****

## 3.1 Pendahuluan

Dalam dunia pemrograman web, backend adalah bagian dari aplikasi yang bekerja di sisi server. Backend bertugas mengelola database, menangani permintaan pengguna, serta mengolah data sebelum dikirimkan ke frontend. Salah satu framework ringan yang digunakan untuk membangun backend adalah **Flask**.

Flask adalah **microframework** berbasis Python yang sederhana dan mudah dipelajari. Dengan Flask, serta aplikasi berbasis server dengan cepat dan efisien.

## 3.2 Instalasi Flask

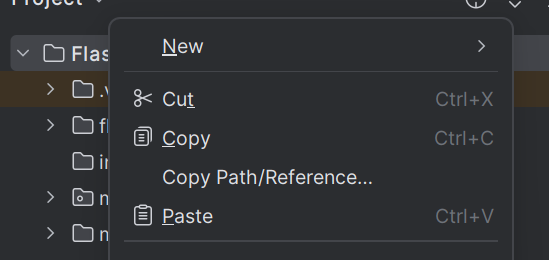
Sebelum menggunakan Flask, kita perlu menginstalnya terlebih dahulu. Pastikan Python sudah terpasang di komputer. Untuk menginstal Flask, gunakan perintah berikut di terminal atau command prompt:

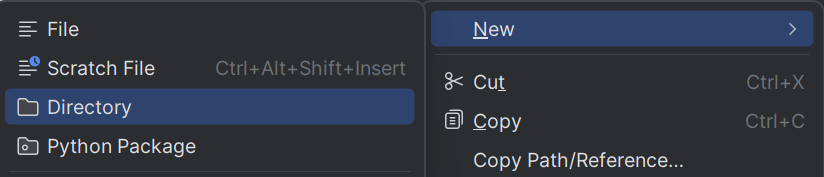
Pip install flask

Setelah instalasi selesai, kita bisa mulai membuat aplikasi sederhana menggunakan Flask.

**3.3.Membuat File Di Pycharm**

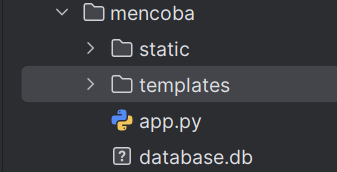
Untuk Membuat setiap file di **Pychram**, lakukan Langkah – Langkah berikut:

1. Klik kanan pada folder proyek (folder anda).
2. Terus klik **new** => **Directory** buat nama terserah kalian (mencoba).



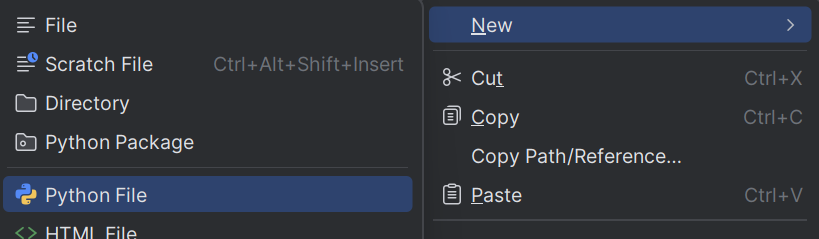
1. Terus klik kanan yang directory yang anda buat.

* Buat nama folder “ templates ” untuk file HTML.
* Buat nama folder “ static ” untuk file CSS, JAVASCRIPT, dan Gambar.

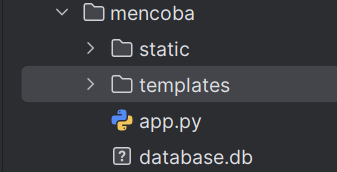


1. Untuk Membuat file app.py (file utama)

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



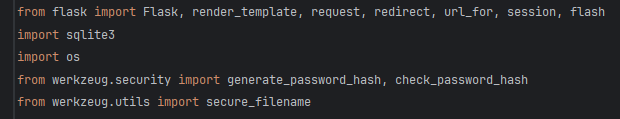
1. Ketik nama file : “ app.py ”, lalu klik **enter** .



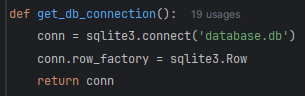
**3.4.Membangun Backend dengan Flask Python**

Berikut adalah cara memulai membangun Backend Website dengan Flask :

1. Ketik kode di bawah ini

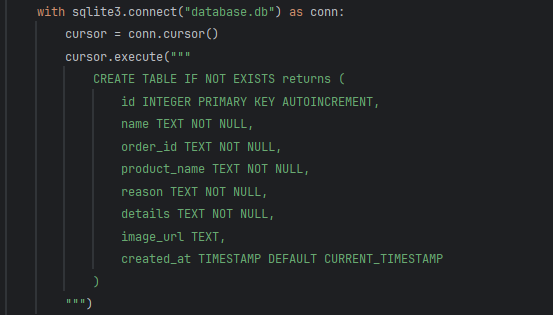


Kode tersebut mengimpor beberapa modul dan fungsi dari Flask, modul sqlite3 di Python, serta inisialisasi aplikasi Flask.

1. Koneksi ke database

Dengan fungsi connect\_db()connect\_db(), kaliandapat dengan mudah mengelola koneksi ke database SQLite dan menjalankan operasi database dalam aplikasi Flask.

1. Buat tabel dalam database jika belum tersedia

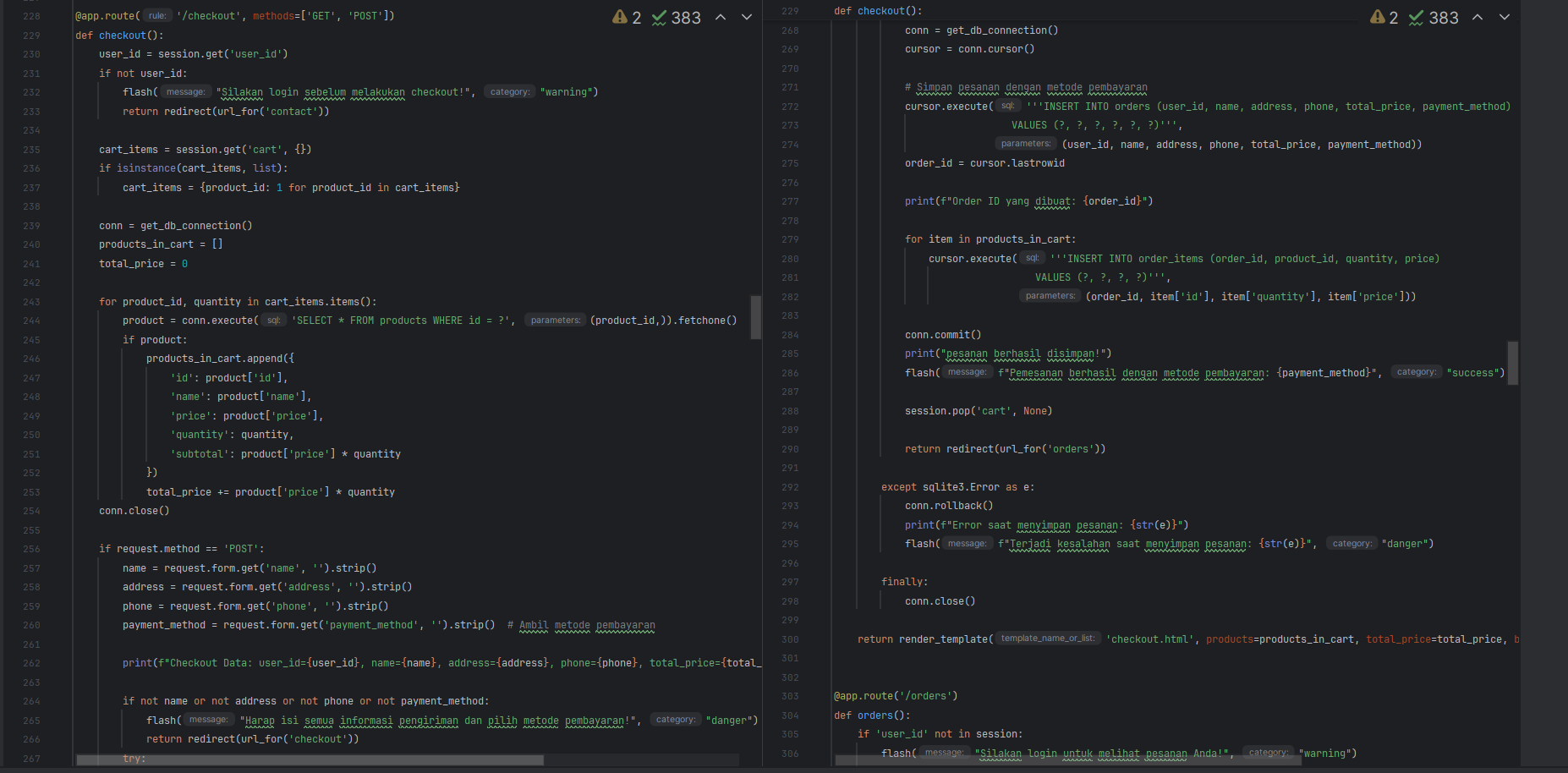


Tabel ini akan terkoneksi dengan DB Browser, yang mana data-data seperti nama, oreder\_id, product\_name,dan lainnya akan tersimpan di dalamnya.

1. Buat route untuk halaman utama

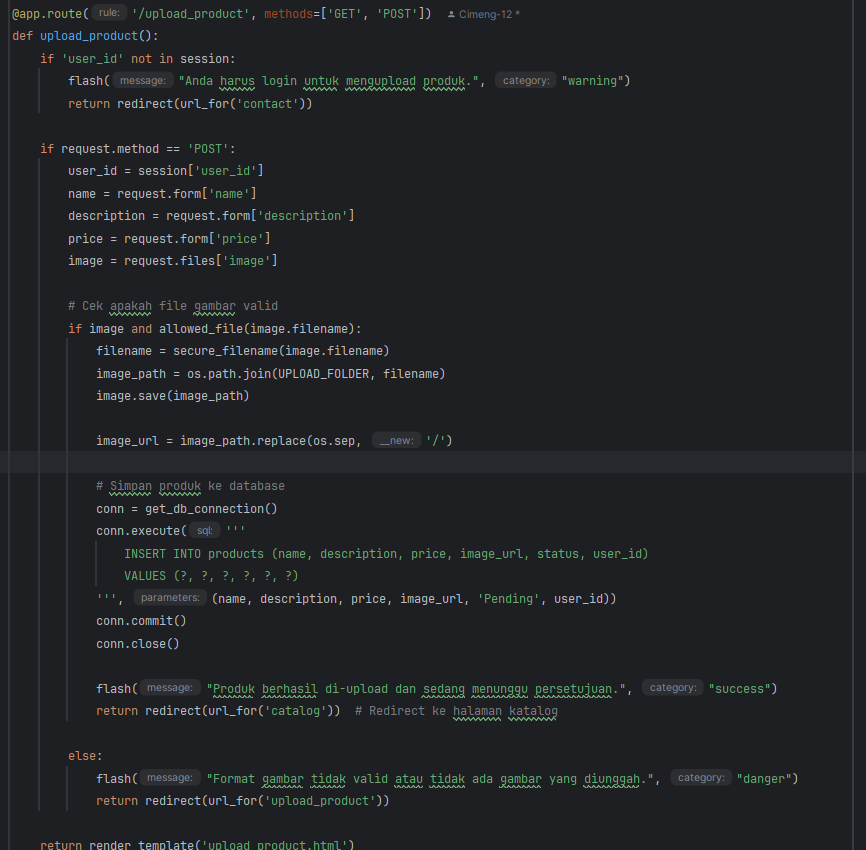


Kode di atas adalah contoh penggunaan route dalam Flask. Pada fungsi home\_page, route akan mengembalikan file HTML yaitu index.html.

1. Halaman checkout pelanggan

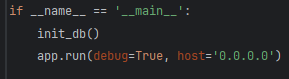
Pada route ini akan berisi data-data dalam database yang memuat nama, alamat, nomer telpon, dan metode pembayaran, serta menampilkan file checkout.html. Pelanggan juga dapat melakukan pemesanan pada halaman ini.

1. Halaman upload untuk menambhkan barang



Pada route ini berisi data-data nama produk,deskripsi produk,harga barang, dan gambar produk,serta manampilkaan file upload\_produk.html.pelanggan bisa manambahkan barang.

* Menjalankan program



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

## 3.7 Kesimpulan

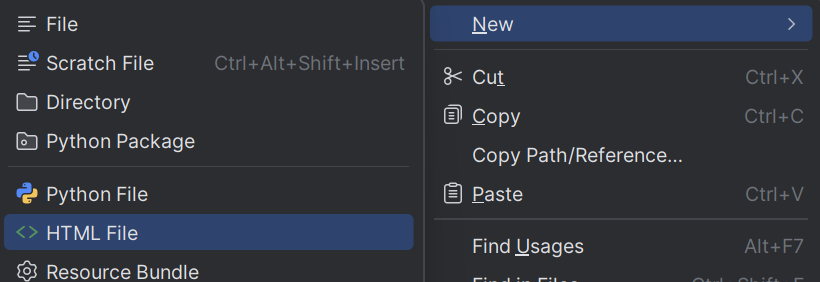
Pada bab ini, kita telah mempelajari dasar-dasar penggunaan Flask untuk membangun backend sebuah website. Mulai dari instalasi, routing, penggunaan template, hingga koneksi dengan database SQLite. Dengan memahami dasar-dasar ini, kita bisa mulai membuat aplikasi web yang lebih kompleks menggunakan Flask.

# ****BAB IV****

**FRONTEND DENGAN HTML, CSS, DAN JAVASCRIPT**

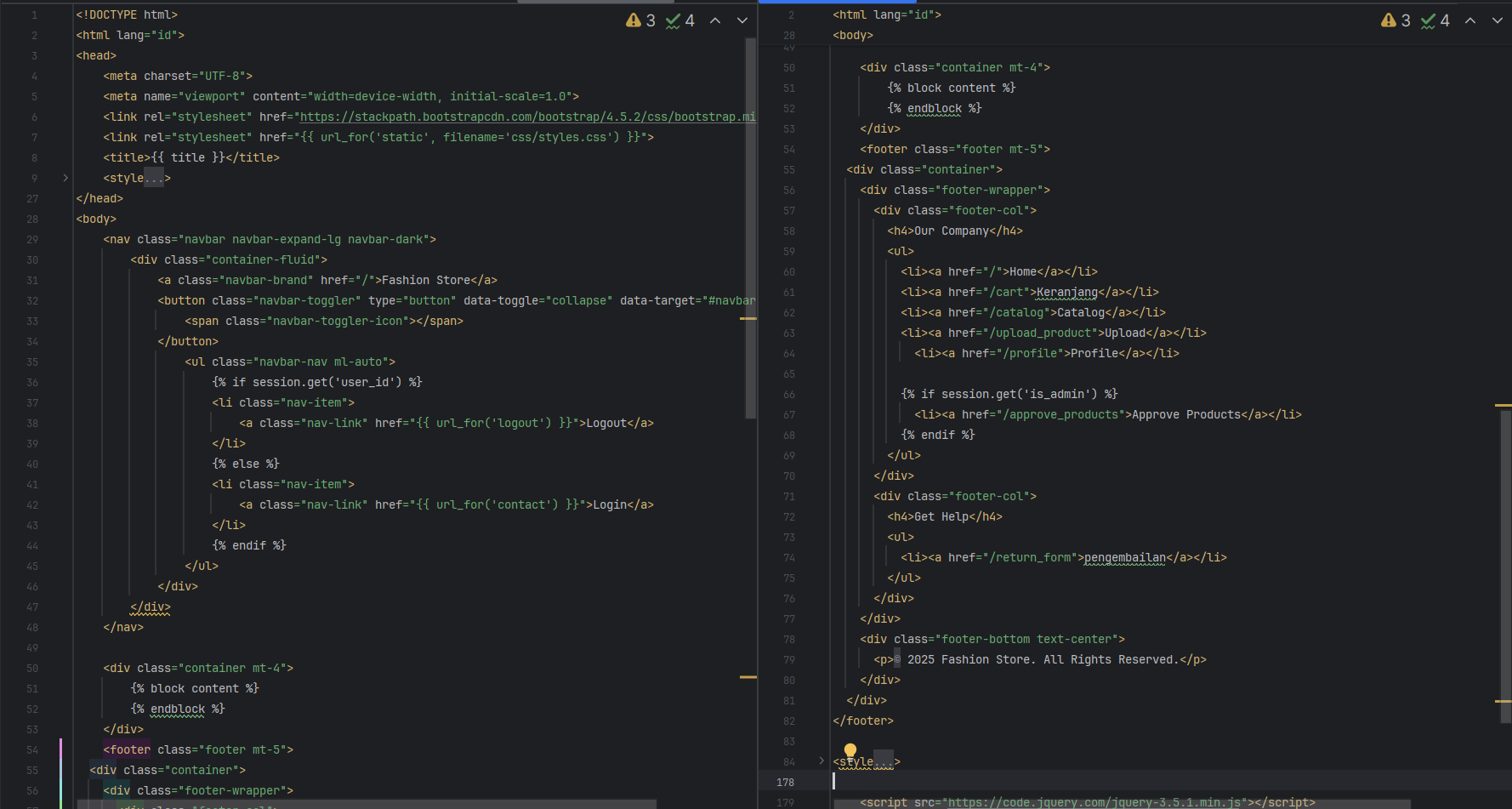
**4.1. Membuat Stuktur Halaman HTML**

Buat file “ base.html ” di dalam templates/ untuk menampilkan halaman utama website

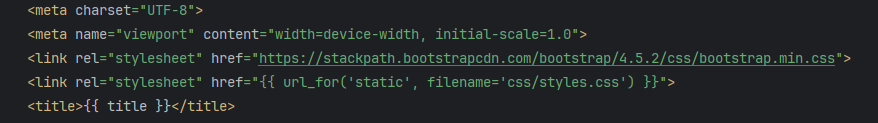


**4.2. Code base.html**

Ikuti yang ada di dalam foto tersebut



Di dalam tag <head> terdapat kode berikut,tambahkan link bootstrap dan link css

Sekarang saya tunjukkan hasil navbar saya, setelah itu kita membuat navbar yang sesuai dengan saya tunjukkan dibawah hasil navbar ada code HTML,CSS dan JS. kalian ikuti code dari foto dibawah ini :

* Output navbar



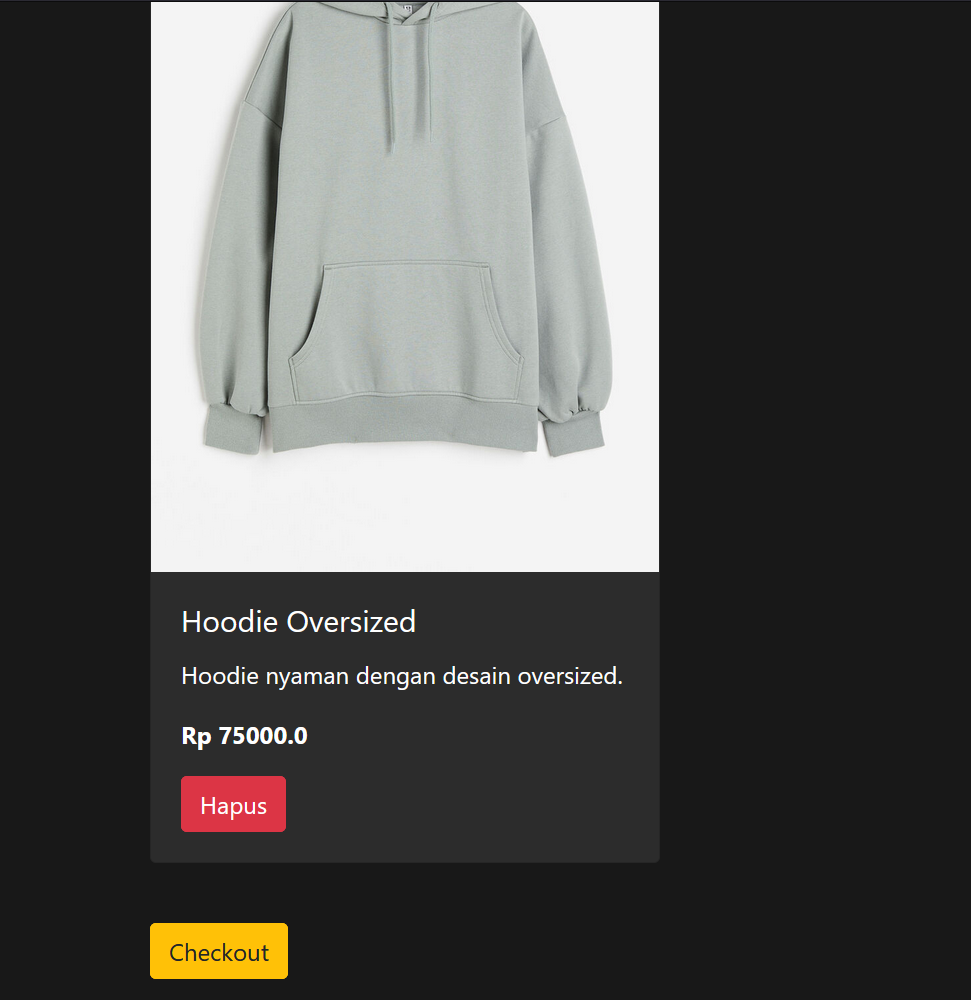


|  |
| --- |
| CSS |
| JAVASCRIPT |

**4.3. pembayaran.html**

* Output pembayaran.html





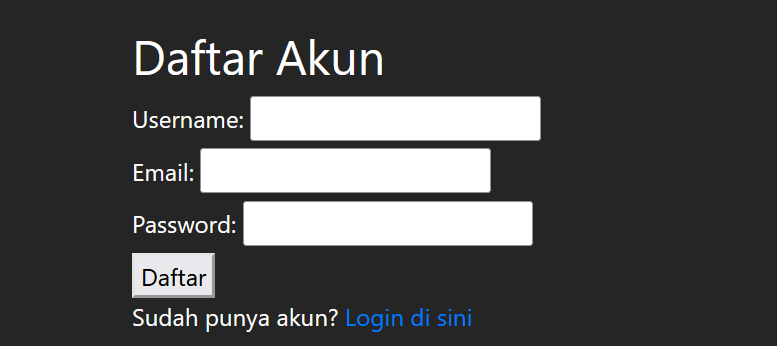
|  |
| --- |
| **HTML** |

|  |
| --- |
|  |

|  |
| --- |
| **CSS** |

**4.4. Menambahkan barang**

* Masuk ke daftar terlebih dahulu jika memang belum memiliki akun

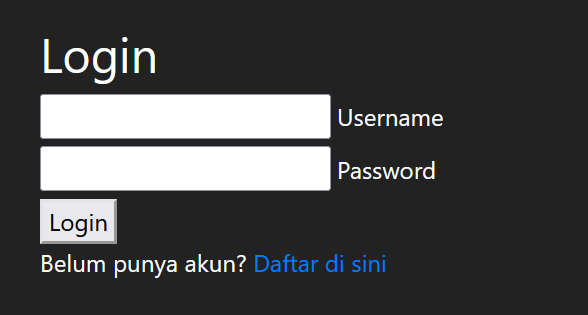


kode daftar.html

|  |
| --- |
|  |

* Masuk ke halaman login

Masukkan username dan password ya ada buat di halaman daftar tadi

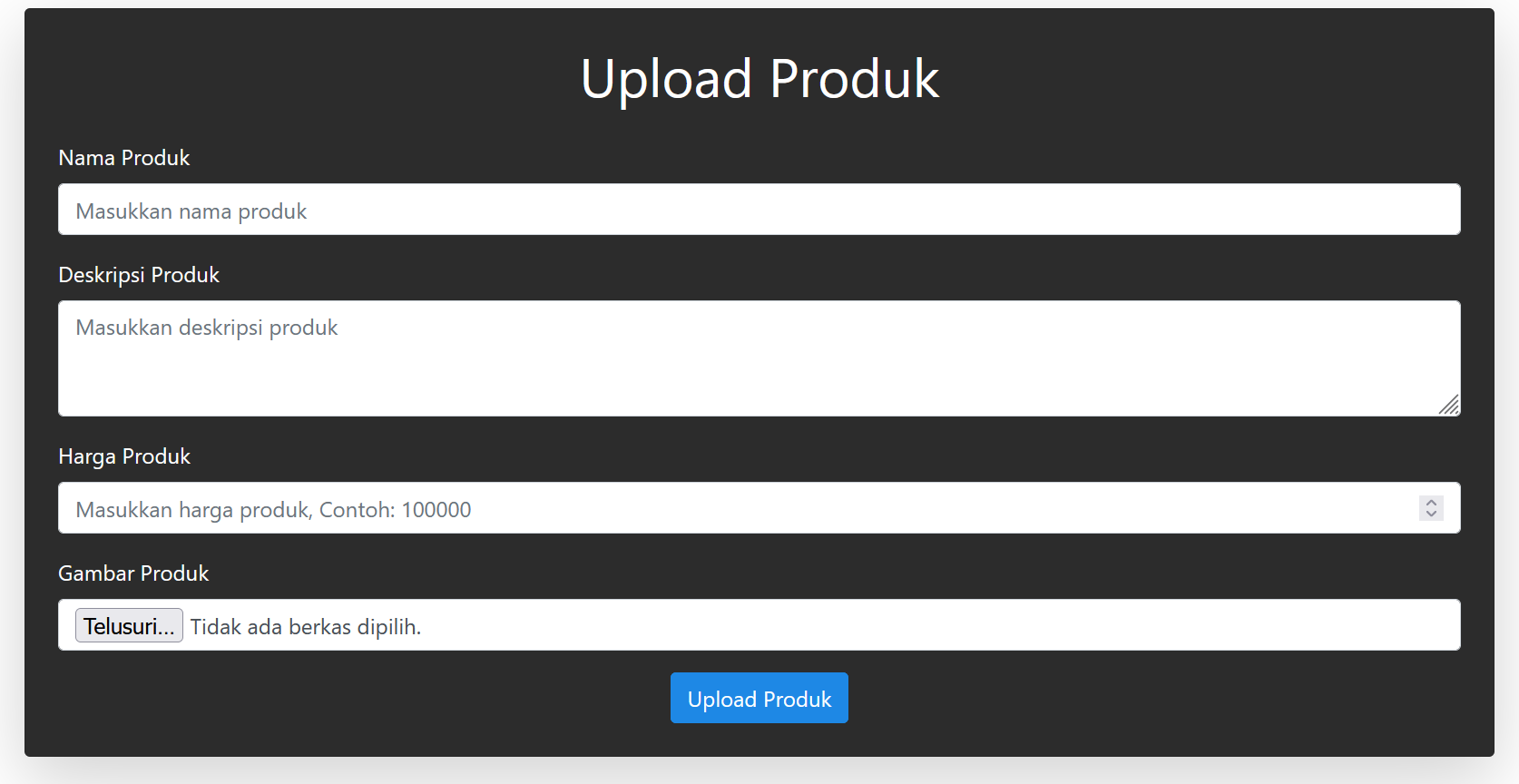


Kode contact.html

|  |
| --- |
|  |

* Masuk ke halaman Upload

Untuk menambahkan produk yang di ingin kan



Kode upload\_product.html

|  |
| --- |
|  |

**DAFTAR PUSTAKA**

* W3Schools. "HTML, CSS, and JavaScript Tutorials". [Online] Available at: https://www.w3schools.com/
* Flask Documentation. "Flask Web Framework". [Online] Available at: https://flask.palletsprojects.com/
* Bootstrap. "Official Documentation". [Online] Available at: https://getbootstrap.com/

**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**](mailto:ekovedc@gmail.com)