

# WEB FASHION STORE



## 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.

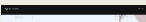
Web Fashion Store merupakan platform digital yang dirancang untuk memudahkan penjual dan pembeli dalam berinteraksi serta melakukan transaksi secara online. Dalam modul ini, akan dibahas berbagai aspek pengembangan web, mulai dari desain tampilan, implementasi fitur, hingga integrasi sistem yang mendukung 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 Februari 2025

**PENULIS**

 <b>Unlabeled</b>	Bidang Keahlian: <b>Teknologi Informasi</b>	Lab: <b>SE - 209</b>	Tanggal: <b>Sept. 2023</b>	<div>TI</div>	<div>SE</div>	<div>04</div>	<div>VT</div>	Page 1 - 27
	Program Keahlian: TI-SE-TE	<b>Universal Internet of Things</b>	Rev.: 06/11/2024	Dibuat oleh: <b>Meliani   Eko</b>				

# 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 mempelajari 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.

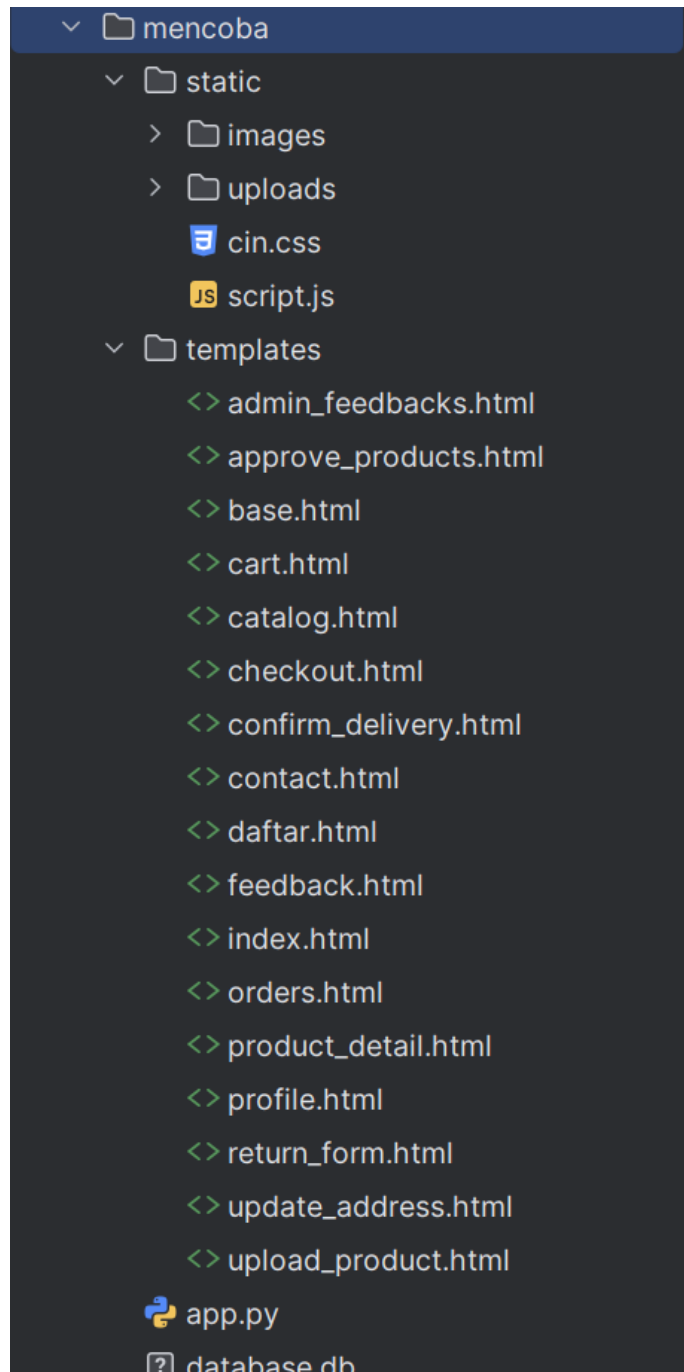
---

## 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:

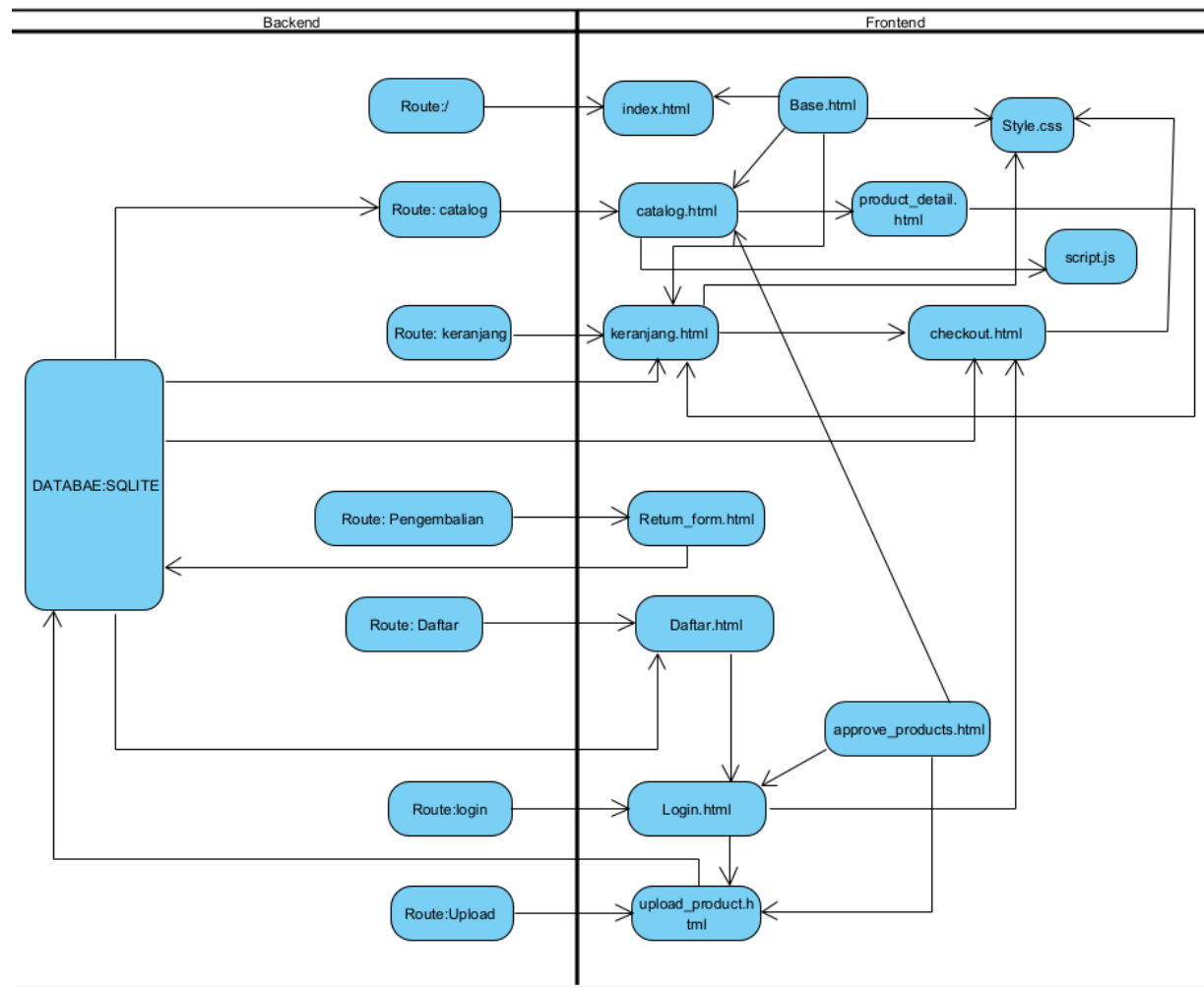


Setiap folder memiliki fungsinya masing-masing:

- 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:

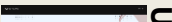
Flask akan bekerja sebagai Backend yang menangani input data dan disimpan ke database; HTML, CSS, dan JavaScript akan menjadi tampilan utama website; SQLite3 berperan sebagai database, yang menyimpan data-data yang telah dimasukkan oleh pengguna.

## 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**.

	Bidang Keahlian: <b>Teknologi Informasi</b>	Lab: <b>SE - 209</b>	Tanggal: <b>Sept. 2023</b>	TI	SE	04	VT	Page <b>7 - 27</b>
	Program Keahlian: <b>TI-SE-TF</b>	<b>Universal</b> <b>Internet of Things</b>	Rev.: <b>06/11/2024</b>	Dibuat oleh: <b>Meliani   Eko</b>				



Flask adalah **microframework** berbasis Python yang sederhana dan mudah dipelajari. Dengan Flask, kita bisa membuat 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:

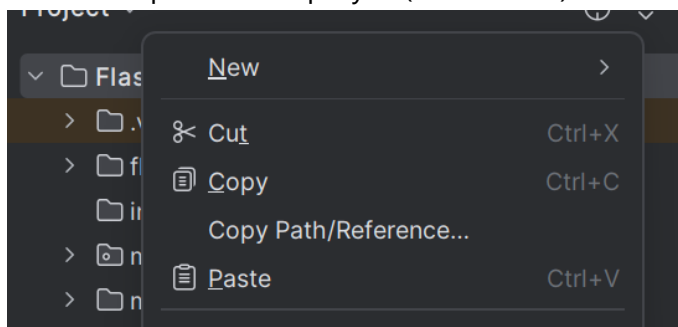
```
Pip install flask
```

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

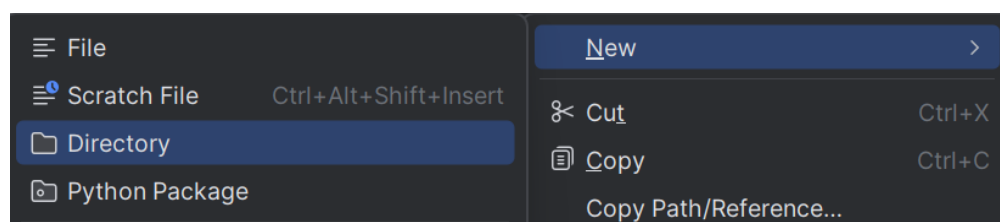
### 3.3. Membuat File Di Pycharm

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

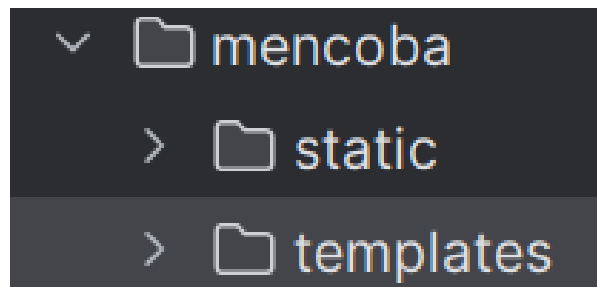
1. Klik kanan pada folder proyek (folder anda).



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

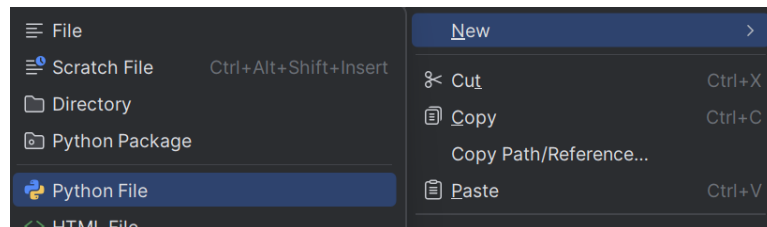


3. Terus klik kanan yang directory yang anda buat.
  - Buat nama folder " templates " untuk file HTML.
  - Buat nama folder " static " untuk file CSS, JAVASCRIPT, dan Gambar.

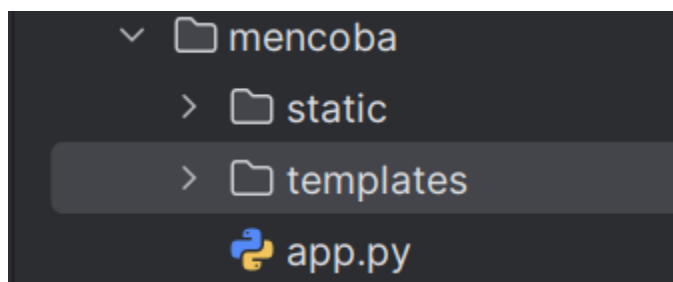


4. Untuk Membuat file app.py (file utama)


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



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



### 3.4.Membangun Backend dengan Flask Python

 <b>Unl</b>	Bidang Keahlian: <b>Teknologi Informasi</b>	Lab: <b>SE - 209</b>	Tanggal: <b>Sept. 2023</b>	TI	SE	04	VT	Page 9 - 27
	Program Keahlian: <b>TI-SE-TF</b>	<b>Universal Internet of Things</b>	Rev.: <b>06/11/2024</b>	Dibuat oleh: <b>Meliani   Eko</b>				

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, flash
import sqlite3
import os
from werkzeug.security import generate_password_hash, check_password_hash
from werkzeug.utils import secure_filename
```

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

2. Koneksi ke database

```
def get_db_connection():
    conn = sqlite3.connect('database.db')
    conn.row_factory = sqlite3.Row
    return conn
```

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

3. Buat tabel dalam database jika belum tersedia

```
with sqlite3.connect("database.db") as conn:
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS returns (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT NOT NULL,
            order_id TEXT NOT NULL,
            product_name TEXT NOT NULL,
            reason TEXT NOT NULL,
            details TEXT NOT NULL,
            image_url TEXT,
            created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        )
    """)
```

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

4. Buat route untuk halaman utama

```
@app.route('/')
def index():
    return render_template(template_name_or_list='index.html')
```

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

```
200 @app.route('/checkout', methods=['GET', 'POST'])
201 def checkout():
202     user_id = session.get('user_id')
203     if not user_id:
204         flash(message="Silakan login sebelum melakukan checkout!", category="warning")
205         return redirect(url_for('contact'))
206
207     cart_items = session.get('cart', {})
208     if isinstance(cart_items, list):
209         cart_items = {product_id: 1 for product_id in cart_items}
210
211     conn = get_db_connection()
212     products_in_cart = []
213     total_price = 0
214
215     for product_id, quantity in cart_items.items():
216         product = conn.execute('SELECT * FROM products WHERE id = ?', (product_id,)).fetchone()
217         if product:
218             products_in_cart.append({
219                 'id': product['id'],
220                 'name': product['name'],
221                 'price': product['price'],
222                 'quantity': quantity,
223                 'subtotal': product['price'] * quantity
224             })
225     total_price = sum(product['subtotal'] for product in products_in_cart)
226     conn.close()
227
228     if request.method == 'POST':
229         name = request.form.get('name', '').strip()
230         address = request.form.get('address', '').strip()
231         phone = request.form.get('phone', '').strip()
232         payment_method = request.form.get('payment_method', '').strip() # Pilih metode pembayaran
233
234         print(f"Checkout Data: user_id={user_id}, name={name}, address={address}, phone={phone}, total_price={total_price}")
235
236         if not name or not address or not phone or not payment_method:
237             flash(message="Harap isi semua informasi pengiriman dan pilih metode pembayaran!", category="danger")
238             return redirect(url_for('checkout'))
239
240     def checkout():
241         conn = get_db_connection()
242         cursor = conn.cursor()
243
244         # Simpan pesanan dengan metode pembayaran
245         cursor.execute('INSERT INTO orders (user_id, name, address, phone, total_price, payment_method)
246             VALUES (?, ?, ?, ?, ?, ?)')
247         order_id = cursor.lastrowid
248
249         print(f"Order ID yang dibuat: {order_id}")
250
251         for item in products_in_cart:
252             cursor.execute('INSERT INTO order_items (order_id, product_id, quantity, price)
253                 VALUES (?, ?, ?, ?)')
254             parameters = (order_id, item['id'], item['quantity'], item['price'])
255
256         conn.commit()
257         print("Pesanan berhasil disimpan")
258         flash(message=f"Pesanan berhasil dengan metode pembayaran: {payment_method}", category="success")
259         session.pop('cart', None)
260         return redirect(url_for('orders'))
261
262     except sqlite3.Error as e:
263         conn.rollback()
264         print(f"Error saat menyimpan pesanan: {str(e)}")
265         flash(message=f"Terjadi kesalahan saat menyimpan pesanan: {str(e)}", category="danger")
266
267     finally:
268         conn.close()
269
270     return render_template(template_name_or_list='checkout.html', products=products_in_cart, total_price=total_price,
271
272
273 @app.route('/orders')
274 def orders():
275     if 'user_id' not in session:
276         flash(message="Silakan login untuk melihat pesanan Anda!", category="warning")
```

## 5. 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.

## 6. Halaman upload untuk menambahkan barang

```

@app.route('/upload_product', methods=['GET', 'POST'])
def upload_product():
    if 'user_id' not in session:
        flash(message="Anda harus login untuk mengupload produk.", category="warning")
        return redirect(url_for('contact'))

    if request.method == 'POST':
        user_id = session['user_id']
        name = request.form['name']
        description = request.form['description']
        price = request.form['price']
        image = request.files['image']

        # Cek apakah file gambar valid
        if image and allowed_file(image.filename):
            filename = secure_filename(image.filename)
            image_path = os.path.join(UPLOAD_FOLDER, filename)
            image.save(image_path)

            image_url = image_path.replace(os.sep, _new= '/')

            # Simpan produk ke database
            conn = get_db_connection()
            conn.execute('INSERT INTO products (name, description, price, image_url, status, user_id)
            VALUES (?, ?, ?, ?, ?, ?)')
            conn.commit()
            conn.close()

            flash(message="Produk berhasil di-upload dan sedang menunggu persetujuan.", category="success")
            return redirect(url_for('catalog')) # Redirect ke halaman katalog

        else:
            flash(message="Format gambar tidak valid atau tidak ada gambar yang diupload.", category="danger")
            return redirect(url_for('upload_product'))

    return render_template('upload_product.html')

```

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

#### ➤ Menjalankan program

```

if __name__ == '__main__':
    init_db()
    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.

## 3.7 Kesimpulan

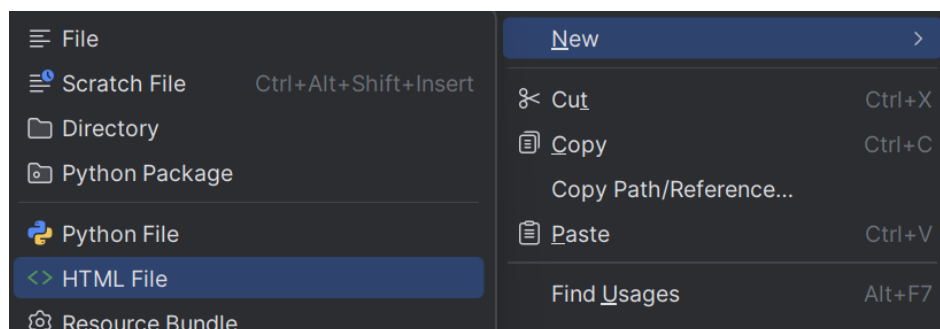
Pada bab ini, kita telah mempelajari dasar-dasar penggunaan Flask untuk membangun backend sebuah website. Mulai dari instalasi, routing, penggunaan template, hingga koneksi dengan database SQLite. Dengan memahami dasar-dasar ini, kita bisa mulai membuat aplikasi web yang lebih kompleks menggunakan Flask.

## 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

```

1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
7   <link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
8   <title>{{ title }}</title>
9 </head>
10 <body>
11   <nav class="navbar navbar-expand-lg navbar-dark">
12     <div class="container-fluid">
13       <a class="navbar-brand" href="/">Fashion Store</a>
14       <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar">
15         <span class="navbar-toggler-icon"></span>
16       </button>
17       <div class="collapse navbar-collapse">
18         <ul class="navbar-nav ml-auto">
19           <li class="nav-item">
20             <a class="nav-link" href="{{ url_for('logout') }}">Logout</a>
21           </li>
22           <li class="nav-item">
23             <a class="nav-link" href="{{ url_for('contact') }}">Login</a>
24           </li>
25         </ul>
26       </div>
27     </div>
28   </nav>
29
30   <div class="container mt-4">
31     <div class="block content">
32       <div class="block content">
33
34     </div>
35   </div>
36
37   <div class="container mt-5">
38     <div class="container">
39       <div class="footer-wrapper">
40         <div class="footer-col">
41           <h4>Our Company</h4>
42           <ul>
43             <li><a href="/">Home</a></li>
44             <li><a href="/cart">Keranjang</a></li>
45             <li><a href="/catalog">Catalog</a></li>
46             <li><a href="/upload_product">Upload</a></li>
47             <li><a href="/profile">Profile</a></li>
48           </ul>
49           <div class="if session.get('is_admin')">
50             <li><a href="/approve_products">Approve Products</a></li>
51           </div>
52         </div>
53         <div class="footer-col">
54           <h4>Get Help</h4>
55           <ul>
56             <li><a href="/return_form">pengembalian</a></li>
57           </ul>
58         </div>
59       </div>
60       <div class="footer-bottom text-center">
61         <p>© 2025 Fashion Store. All Rights Reserved.</p>
62       </div>
63     </div>
64   </div>
65
66   <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

```

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

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
<title>{{ title }}</title>

```

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

Fashion Store

Login

## OUR COMPANY

Home

Keranjang

Catalog

Upload

Profile

## GET HELP

[pengembailan](#)



## CSS

```
body {
    background-color: {{ bg_color }};
    color: {{ text_color }};
}

.card {
    background-color: #2c2c2c;
    color: #ffffff;
}

.navbar {
    background-color: #121212 !important;
}

.btn-primary {
    background-color: #1e88e5;
    border-color: #1e88e5;
}
```

```
.footer {
    background-color: #1a1a1a;
    color: #fff;
    padding: 50px 0 20px;
}

.footer-wrapper {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    gap: 20px;
}

.footer-col {
    flex: 1 1 220px;
    margin-bottom: 20px;
}

.footer-col h4 {
    font-size: 18px;
    text-transform: uppercase;
    margin-bottom: 20px;
    font-weight: 600;
    position: relative;
}

.footer-col h4::before {
    content: "";
    position: absolute;
    left: 0;
    bottom: -8px;
    width: 50px;
    height: 3px;
    background-color: goldenrod;
}

.footer-col ul {
    list-style: none;
    padding: 0;
}

.footer-col ul li {
    margin-bottom: 10px;
}

.footer-col ul li a {
    color: #bbb;
    text-decoration: none;
    font-size: 15px;
    transition: 0.3s;
}
```

```
.footer-col ul li a:hover {
    color: #fff;
    padding-left: 5px;
}

.social-links a {
    display: inline-block;
    font-size: 20px;
    color: #bbb;
    width: 40px;
    height: 40px;
    line-height: 40px;
    text-align: center;
    border-radius: 50%;
    background: rgba(255, 255, 255, 0.1);
    margin-right: 10px;
    transition: 0.3s;
}

.social-links a:hover {
    background: goldenrod;
    color: #1a1a1a;
}
```

```
.footer-bottom {
    border-top: 1px solid #333;
    padding-top: 20px;
    margin-top: 20px;
    font-size: 14px;
}

@media (max-width: 768px) {
    .footer-wrapper {
        flex-direction: column;
        align-items: center;
    }

    .footer-col {
        text-align: center;
    }
}
```

## JAVASCRIPT

```
document.addEventListener("DOMContentLoaded", function () {  
    // Auto-hide flash messages  
    let flashMessages = document.querySelectorAll(".flash-message");  
    flashMessages.forEach(msg => {  
        setTimeout(() => {  
            msg.style.transition = "opacity 0.5s";  
            msg.style.opacity = "0";  
            setTimeout(() => msg.remove(), 500);  
        }, 3000);  
    });  
  
    // Konfirmasi sebelum menghapus produk dari keranjang  
    let removeButtons = document.querySelectorAll(".remove-from-cart");  
    removeButtons.forEach(btn => {  
        btn.addEventListener("click", function (event) {  
            let confirmDelete = confirm("Apakah Anda yakin ingin menghapus produk ini dari keranjang?");  
            if (!confirmDelete) {  
                event.preventDefault();  
            }  
        });  
    });  
  
    // Validasi form checkout  
    let checkoutForm = document.querySelector("#checkout-form");  
    if (checkoutForm) {  
        checkoutForm.addEventListener("submit", function (event) {  
            let name = document.querySelector("#name").value.trim();  
            let address = document.querySelector("#address").value.trim();  
            let phone = document.querySelector("#phone").value.trim();  
  
            if (name === "" || address === "" || phone === "") {  
                alert("Harap lengkapi semua bidang sebelum melanjutkan checkout.");  
                event.preventDefault();  
            }  
        });  
    }  
});
```

### 4.3. pembayaran.html

- Output pembayaran.html

## Checkout

• Hoodie Oversized Hoodie Oversized Rp75,000

**Total: Rp75,000**

Nama Lengkap


Alamat Pengiriman

Nomor Telepon

Pilih Metode Pembayaran:

Transfer Bank

[Bayar Sekarang](#)



Hoodie Oversized

Hoodie nyaman dengan desain oversized.

**Rp 75000.0**

Hapus

Checkout

# HTML

```

1  {% extends "base.html" %}
2
3  {% block content %}
4  <h1>Checkout</h1>
5
6  <!-- Menampilkan Produk dalam Keranjang -->
7  <div class="cart-items">
8    {% if products %}
9      <ul>
10       {% for product in products %}
11         <li>
12           
13           <span>{{ product['name'] }}</span>
14           <span>Rp{{ '{:,.0f}'.format(product['price']) }}</span>
15         </li>
16       {% endfor %}
17     </ul>
18     <hr>
19     <p><strong>Total: Rp{{ '{:,.0f}'.format(total_price) }}</strong></p>
20   {% else %}
21     <p>Keranjang Anda kosong!</p>
22   {% endif %}
23 </div>
24
25 <!-- Form Checkout -->
26 <form method="POST" enctype="multipart/form-data">
27   <div class="form-group">
28     <label for="name">Nama Lengkap</label>
29     <input type="text" class="form-control" id="name" name="name" required>
30   </div>
31   <div class="form-group">
32     <label for="address">Alamat Pengiriman</label>
33     <textarea class="form-control" id="address" name="address" rows="3" required></textarea>
34   </div>
35   <div class="form-group">
36     <label for="phone">Nomor Telepon</label>
37     <input type="text" class="form-control" id="phone" name="phone" required>
38   </div>
39
40   <!-- Pilihan Metode Pembayaran -->
41   <div class="form-group">
42     <label for="payment_method">Pilih Metode Pembayaran</label>
43     <select class="form-control" name="payment_method" id="payment_method" required>
44       <option value="bank_transfer">Transfer Bank</option>
45       <option value="wallet">E-Wallet</option>
46       <option value="credit_card">Kartu Kredit</option>
47     </select>
48   </div>
49
50   <button type="submit" class="btn btn-success">Bayar Sekarang</button>
51 </form>
52
53 {% endblock %}

```

```

1  {% extends "base.html" %}
2
3  {% block content %}
4  <div class="container mt-4">
5    <h1>Keranjang Belanja</h1>
6
7    {% if products %}
8    <div class="row">
9      {% for product in products %}
10       <div class="col-12 col-md-6 col-lg-4 mb-4">
11         <div class="card">
12           
13           <div class="card-body">
14             <h5 class="card-title">{{ product['name'] }}</h5>
15             <p class="card-text">{{ product['description'] }}</p>
16             <p><strong>Rp {{ product['price'] }}</strong></p>
17             <a href="{{ url_for('remove_from_cart', product_id=product['id']) }}" class="btn btn-danger">
18               Hapus
19             </a>
20           </div>
21         </div>
22       </div>
23     {% endfor %}
24   </div>
25
26   <a href="{{ url_for('checkout') }}" class="btn btn-warning mt-3">
27     Checkout
28   </a>
29   {% else %}
30   <p>Keranjang Anda kosong!</p>
31   {% endif %}
32 </div>
33 {% endblock %}

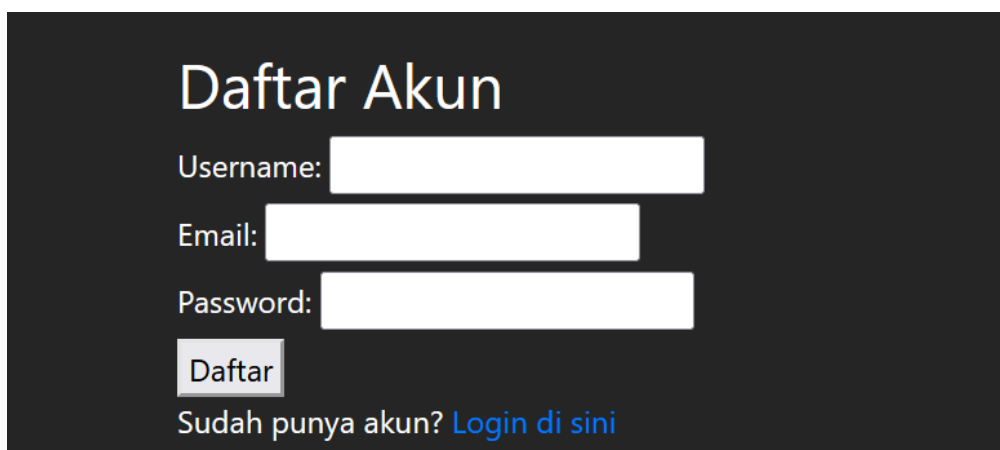
```

## CSS

```
/* Konten di atas overlay */
.container {
  z-index: 2;
  color: white;
  position: relative;
  max-width: 100%;
  padding: 0;
  margin: 0;
}
```

### 4.4. Menambahkan barang

- Masuk ke daftar terlebih dahulu jika memang belum memiliki akun

A dark-themed registration form titled "Daftar Akun". It contains four input fields: "Username:", "Email:", "Password:", and a "Daftar" button. Below the button is a link "Sudah punya akun? [Login di sini](#)".

Daftar Akun

Username:

Email:

Password:

Sudah punya akun? [Login di sini](#)

kode daftar.html

```

1  {% extends "base.html" %}
2
3  {% block content %}
4      <div class="container">
5          <h2>Daftar Akun</h2>
6          <form method="POST">
7              <label for="username">Username:</label>
8              <input type="text" id="username" name="username" required>
9              <br>
10             <label for="email">Email:</label>
11             <input type="email" id="email" name="email" required>
12             <br>
13             <label for="password">Password:</label>
14             <input type="password" id="password" name="password" required>
15             <br>
16             <button type="submit">Daftar</button>
17         </form>
18
19         <p class="login-link">Sudah punya akun? <a href="{ url_for('contact') }">Login di sini</a></p>
20     </div>
21
22 {% endblock %}

```

- Masuk ke halaman login

Masukkan username dan password ya ada buat di halaman daftar tadi

Kode contact.html

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

{% block content %}
{% with messages = get_flashed_messages(with_categories=true) %}
    {% if messages %}
        <div class="alert alert-{{ messages[0][0] }}">
            {{ messages[0][1] }}
        </div>
    {% endif %}
{% endwith %}

<div class="container">
    <h2>Login</h2>
    <form method="POST">
        <div class="input-container">
            <input type="text" id="username" name="username" placeholder=" " required>
            <label for="username">Username</label>
        </div>

        <div class="input-container">
            <input type="password" id="password" name="password" placeholder=" " required>
            <label for="password">Password</label>
        </div>

        <button type="submit">Login</button>
    </form>

    <p class="register-link">Belum punya akun? <a href="{{ url_for('daftar') }}">Daftar di sini</a></p>
</div>

{% endblock %}
```

- Masuk ke halaman Upload  
Untuk menambahkan produk yang di ingin kan



## Upload Produk

**Nama Produk**

**Deskripsi Produk**

**Harga Produk**

**Gambar Produk**

Tidak ada berkas dipilih.


Kode upload\_product.html

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

{% block title %}Upload Produk{% endblock %}

{% block content %}
<div class="container mt-5">
  <div class="card shadow-lg p-4">
    <h1 class="text-center mb-4">Upload Produk</h1>
    <form method="POST" enctype="multipart/form-data">
      <div class="mb-3">
        <label for="name" class="form-label">Nama Produk</label>
        <input type="text" class="form-control" id="name" name="name" placeholder="Masukkan nama produk" required>
      </div>
      <div class="mb-3">
        <label for="description" class="form-label">Deskripsi Produk</label>
        <textarea class="form-control" id="description" name="description" rows="3" placeholder="Masukkan deskripsi produk" required></textarea>
      </div>
      <div class="mb-3">
        <label for="price" class="form-label">Harga Produk</label>
        <input type="number" class="form-control" id="price" name="price" placeholder="Masukkan harga produk, Contoh: 100000" required>
      </div>
      <div class="mb-3">
        <label for="image" class="form-label">Gambar Produk</label>
        <input type="file" class="form-control" id="image" name="image" accept="image/*" required>
      </div>
      <div class="text-center">
        <button type="submit" class="btn btn-primary">Upload Produk</button>
      </div>
    </form>
  </div>
</div>
{% endblock %}
```

## DAFTAR PUSTAKA

	Bidang Keahlian:	Lab:	Tanggal:	TI	SE	04	VT	Page 24 - 27
	Teknologi Informasi	SE - 209	Sept. 2023					
	Program Keahlian:	Universal	Rev.:	Dibuat oleh:				
	TI-SE-TE	Internet of Things	06/11/2024	Meliani   Eko				

- 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)**