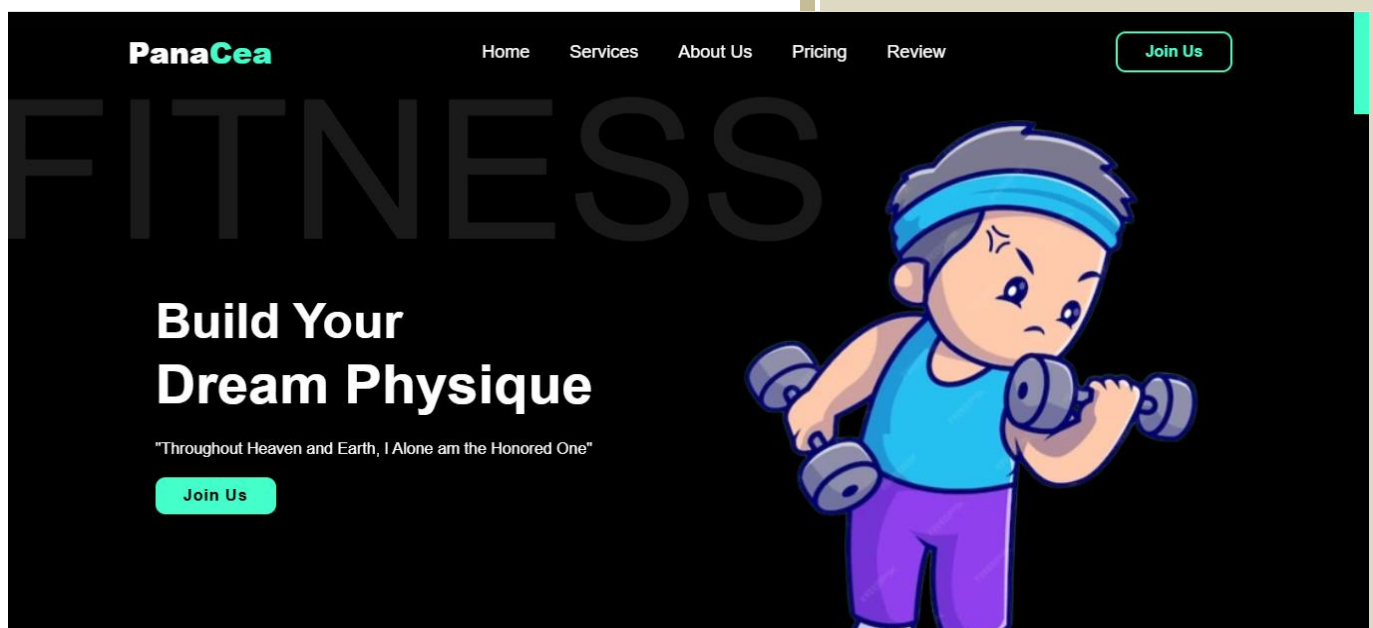


PanaCea Website



KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, modul pembelajaran **Pengembangan Website** ini dapat diselesaikan dengan baik. Modul ini dirancang sebagai **teaching aids** yang bertujuan untuk mendukung proses transfer pengetahuan dan teknologi di bidang Teknik Informatika, serta menjawab kebutuhan pembelajaran yang sesuai dengan perkembangan Revolusi Industri 4.0.

Laporan ini disusun untuk memberikan pemahaman tentang proses pengembangan website, mulai dari perancangan tampilan (frontend), pengelolaan data di server (backend), hingga penggunaan database sebagai penyimpanan informasi. Selain itu, pembahasan juga mencakup teknologi yang sering digunakan dalam pembuatan website modern, seperti HTML, CSS, JavaScript, Python, dan framework pendukung lainnya.

Kami berharap, modul pembelajaran ini dapat menjadi referensi yang bermanfaat bagi para pengajar, siswa, dan praktisi teknologi dalam memahami dan menerapkan konsep Pengembangan Website secara lebih mendalam. Kami juga mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dalam penyusunan modul ini.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun agar tulisan ini dapat lebih bermanfaat di masa mendatang.


Semoga modul Pengembangan Website dapat memberikan manfaat yang sebesar-besarnya bagi dunia pendidikan, serta mampu berkontribusi dalam pengembangan kompetensi di bidang teknologi.

Malang, 25 Februari 2025

PENULIS

BAB I PENDAHULUAN

1.1. Latar Belakang

	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: 27 Feb. 2025	<div>TI</div>	<div>SE</div>	<div>04</div>	<div>VT</div>	Page 1 - 19
	Program Keahlian: TI-SE-TE	Universal Internet of Things	Rev.: 06/03/2025	Dibuat oleh: EKO-Jagad-Rully				

Di era digital saat ini, kebutuhan akan aplikasi web yang ringan, cepat, dan efisien semakin meningkat. Banyak bisnis, institusi, dan individu membutuhkan solusi berbasis web untuk berbagai keperluan, seperti manajemen data, sistem informasi, absensi, hingga e-commerce. Untuk memenuhi kebutuhan tersebut, kombinasi teknologi Flask, HTML, CSS, JavaScript, dan SQLite menjadi pilihan yang ideal.

Flask dipilih karena merupakan framework berbasis Python yang minimalis dan fleksibel. Dibandingkan dengan framework yang lebih besar seperti Django, Flask lebih ringan dan mudah digunakan, sehingga cocok untuk pengembangan proyek kecil hingga menengah. Kemudahan dalam mengatur rute, mengelola request, dan mengintegrasikan database menjadikannya pilihan yang efisien bagi pengembang.

HTML, CSS, dan JavaScript menjadi fondasi utama dalam pengembangan antarmuka pengguna. HTML digunakan untuk membangun struktur halaman, CSS bertanggung jawab dalam memperindah tampilan serta memastikan desain responsif, sementara JavaScript memberikan interaktivitas dan dinamika pada halaman web. Kombinasi ketiga teknologi ini memungkinkan pembuatan antarmuka yang menarik, responsif, dan mudah digunakan oleh pengguna.

SQLite digunakan sebagai database karena sifatnya yang ringan dan tidak memerlukan server terpisah. Dengan pendekatan ini, aplikasi dapat berjalan dengan lebih sederhana tanpa perlu konfigurasi database yang kompleks. SQLite sangat cocok untuk aplikasi berskala kecil hingga menengah yang membutuhkan penyimpanan data tanpa perlu sistem manajemen database yang besar.

Penggunaan teknologi ini memberikan berbagai keunggulan, seperti kemudahan dalam pembelajaran dan pengembangan, proses deployment yang cepat, serta efisiensi dalam penggunaan sumber daya. Dengan menggabungkan teknologi ini, pengembang dapat membangun website yang fungsional, menarik, dan mudah dikelola sesuai dengan kebutuhan pengguna.


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. Mengajarkan cara menghubungkan Flask dengan database SQLite.
4. Memberikan tampilan menarik menggunakan HTML, CSS, dan JavaScript.

1.3. Manfaat

Dengan mempelajari modul ini, pembaca diharapkan dapat:

	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: 27 Feb. 2025	TI	SE	04	VT	Page 2 - 19
	Program Keahlian: TI-SE-TE	Universal Internet of Things	Rev.: 06/03/2025	Dibuat oleh: EKO-Jagad-Rully				

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

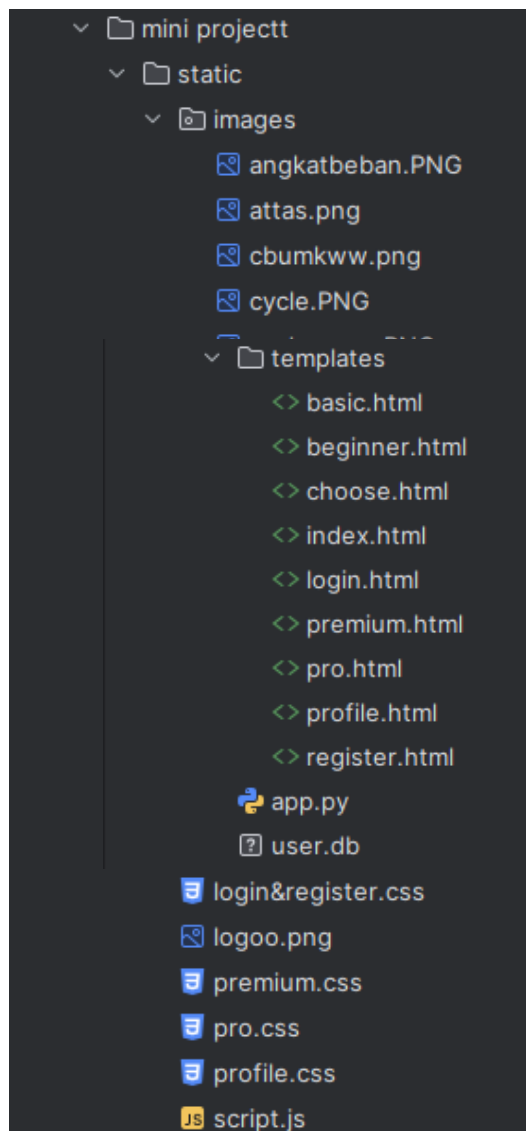
Modul ini disusun menjadi beberapa bab dengan sistematika sebagai berikut:

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** - Membuat tampilan utama website yang berinteraksi secara langsung dengan pengguna menggunakan HTML, CSS, dan JavaScript.

BAB II STRUKTUR PROYEK FLASK

2.1. Struktur Direktori Proyek

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

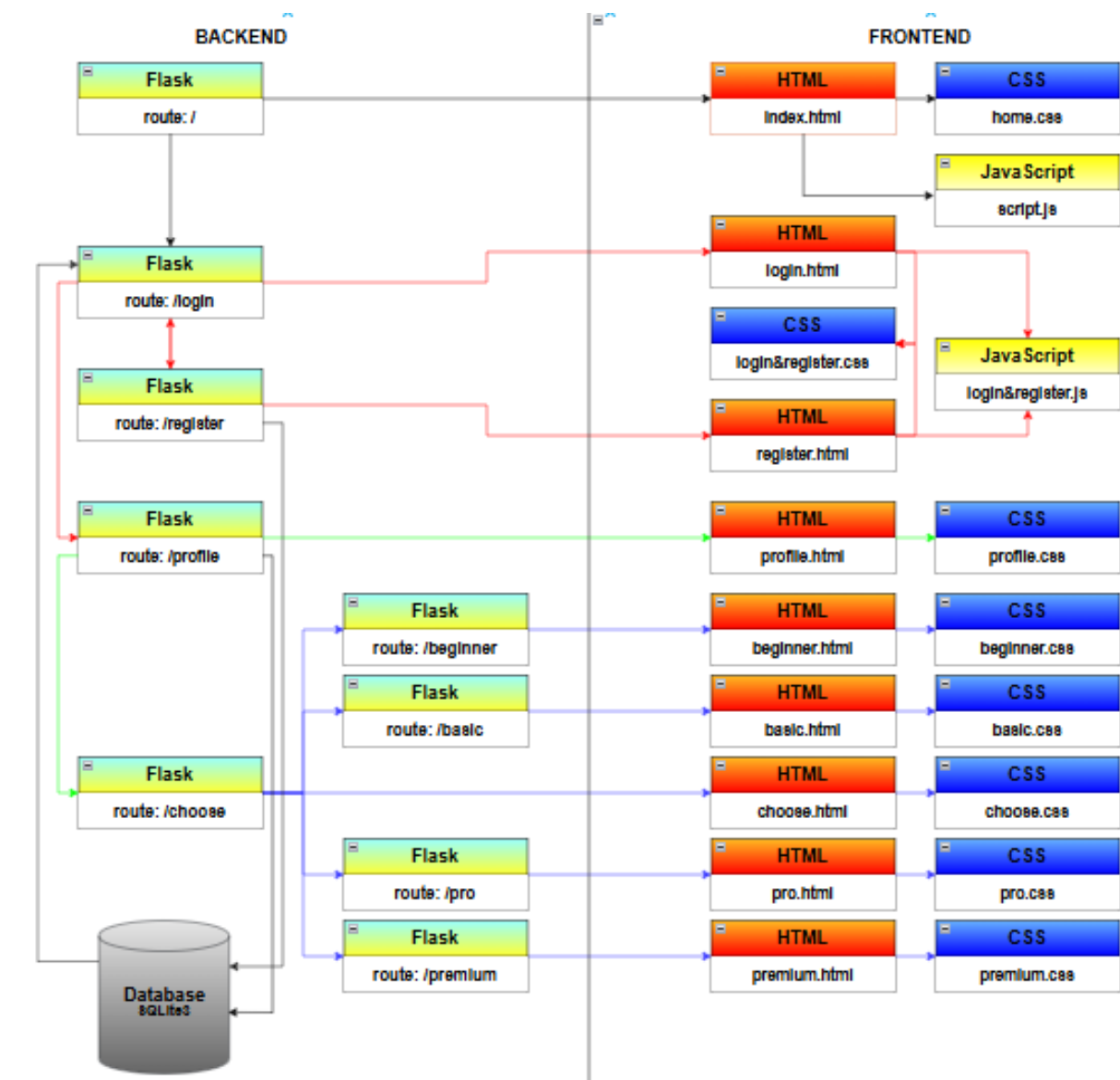


Setiap folder memiliki fungsinya masing-masing:

- static/ Folder: Menyimpan file static seperti CSS, JavaScript, dan gambar.
- templates/ Folder: Menyimpan file HTML yang akan ditampilkan oleh Flask.
- app.py: File utama yang menjalankan website Flask, menangani request, dan menghubungkan ke database.
- user.db: Database SQLite untuk Menyimpan data pribadi pengguna.

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.

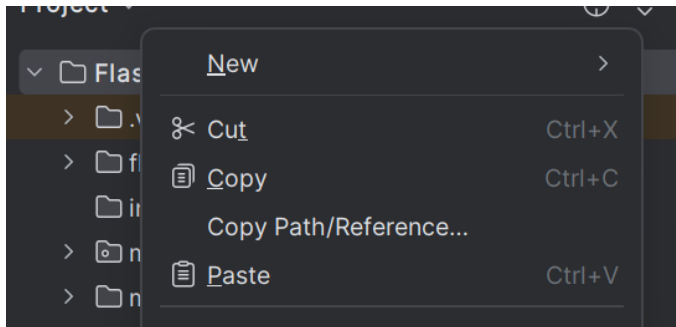
BAB III

BACKEND DENGAN MICROFRAMEWORK FLASK PYTHON

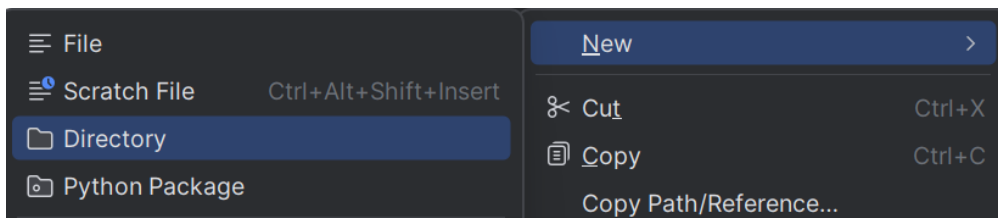
3.1. Membuat Direktori dan File Di Pycharm

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

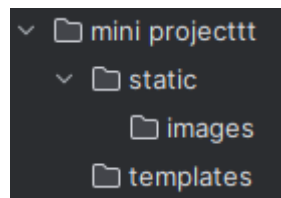
1. Klik kanan pada folder utama Anda.



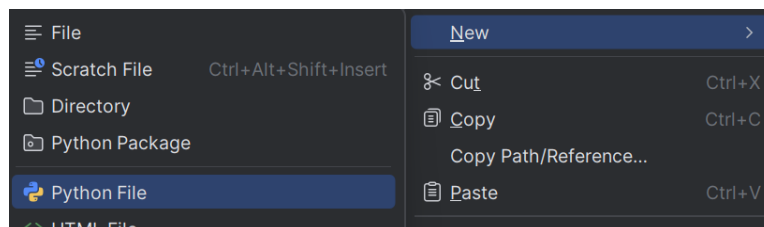
2. Klik **new** => **Directory** buat nama bebas, misalnya mini projectt.



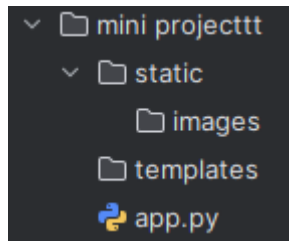
3. Klik kanan pada directory yang anda buat.
 - Buat nama folder “ templates ” untuk file HTML.
 - Buat nama folder “ static ” untuk file CSS, JavaScript serta “images” untuk Gambar.



4. Buat juga file Flask dengan nama misalnya app.py.
 - Klik kanan pada directory mini projecttt.
 - Klik **new** => **Python File**.



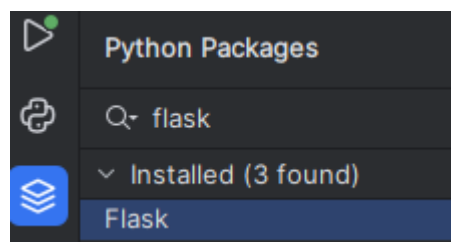
5. Ketik nama file : “ app.py ”, lalu klik **enter** .



3.2. Membangun Backend dengan Flask Python

Berikut adalah cara memulai membangun Backend Website dengan Flask :

1. Install Packages Flask di bagian kiri bawah pada logo kotak tumpuk tiga. Pada menu tersebut, ketikkan Flask lalu Install.



2. Ketik kode di bawah ini. Kode tersebut mengimpor beberapa modul dan fungsi dari Flask, modul sqlite3 di Python, serta inisialisasi aplikasi Flask.

```
from flask import Flask, render_template, request, redirect, url_for, jsonify
import sqlite3

app = Flask(__name__)
```

3. Buat database dan tabel. Tabel yang dibuat akan disimpan ke database dan berisi id, name, email, dan password. Tutup database dengan conn.close().

```
def create_db():
    usage
    conn = sqlite3.connect('user.db')
    c = conn.cursor()
    c.execute('''CREATE TABLE IF NOT EXISTS user (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL,
        email TEXT UNIQUE NOT NULL,
        password TEXT NOT NULL)''')
    conn.commit()
    conn.close()

create_db()
```

4. Buat route sebagai halaman utama. Pada halaman ini akan berisi pengenalan tentang website PanaCea, dimana route akan menampilkan frontend dari index.html.


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

5. Buat register & login menu untuk pengguna baru. Data pengguna akan otomatis tersimpan ke database user yang telah dibuat sebelumnya.

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

        conn = sqlite3.connect('user.db')
        c = conn.cursor()

        try:
            c.execute(sql="SELECT * FROM user WHERE email = ?", parameters=(email,))

            if c.fetchone():
                flash(message='Email is already registered!', category='error')
            else:
                c.execute(sql="INSERT INTO user (name, email, password) VALUES (?, ?, ?)", parameters=(name, email, password))
                conn.commit()
                flash(message='Register successful!', category='success')
                return redirect(url_for('login'))
        except sqlite3.IntegrityError:
            flash(message='Error : Failed to save data', category='error')
        finally:
            conn.close()
    return render_template('register.html')
```

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

        with sqlite3.connect('user.db') as conn:
            c = conn.cursor()
            c.execute(sql="SELECT * FROM user WHERE email = ? AND password = ?", parameters=(email, password))
            user = c.fetchone()

            if user:
                return redirect(url_for('profile'))
            else:
                flash(message='Invalid Email or Password!', category='error')
                return redirect(url_for('login'))

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

6. Jalankan website dengan perintah berikut.

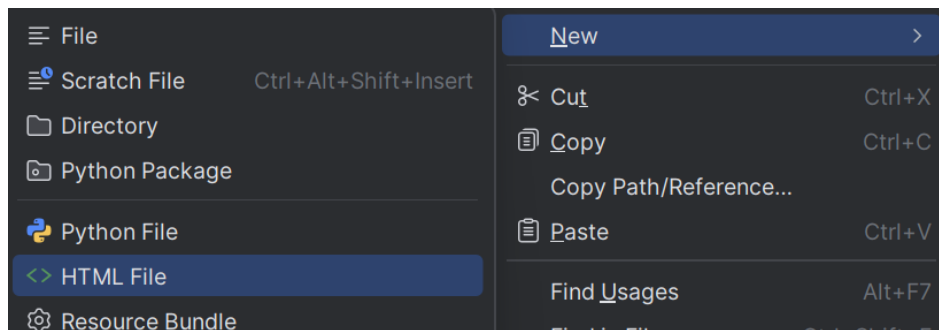
```
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

BAB IV

FRONTEND DENGAN HTML CSS DAN JAVASCRIPT

4.1. Membuat Stuktur Halaman HTML

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



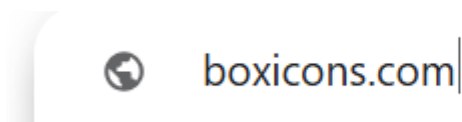
4.2. Code index.html

Dalam HTML, ada dua bagian utama yaitu <head> dan <body>.

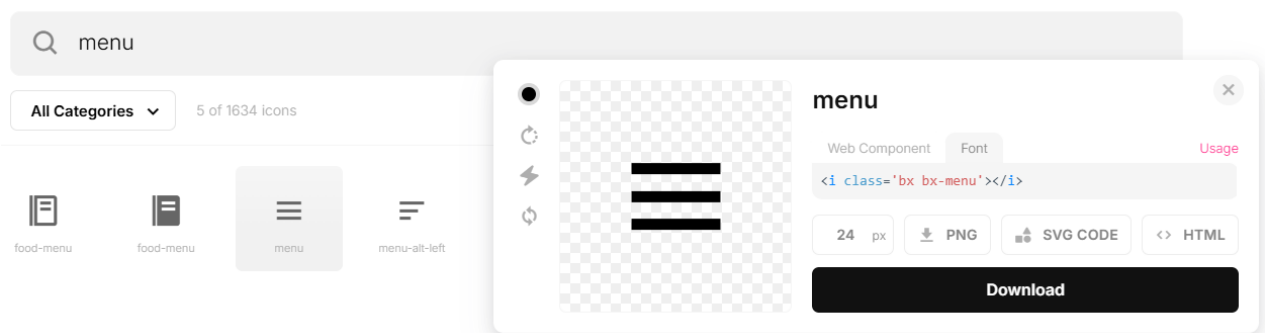
Kode pada bagian <head> index.html biasanya berupa link yang dapat ditulis sebagai berikut.

4.3. Ikon dari Boxicons.com

1. Cari di browser website “ boxicons.com ”



2. Pada bagian berikut ketik “menu” dan klik “menu”. Pergi ke Font lalu copy link dengan klik link tersebut. Simpan di <body> HTML.



```
<i class='bx bx-menu'></i>
```

3. Pergi ke “Usage” pada bagian kanan atas. Scroll down sampai “Usage as a Font” lalu copy. Dan letakkan pada bagian <head> di HTML.

[Github](#)[Usage](#)[Download v2.1.4](#)

Usage as a Font

Import the CSS

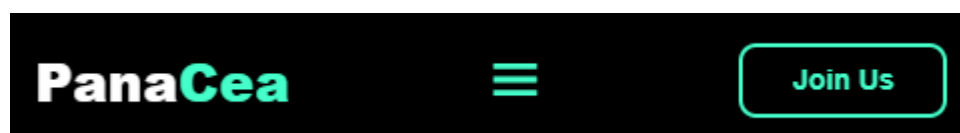
Copy-paste the stylesheet [link](#) into your [head](#) to load our CSS

```
<link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
```

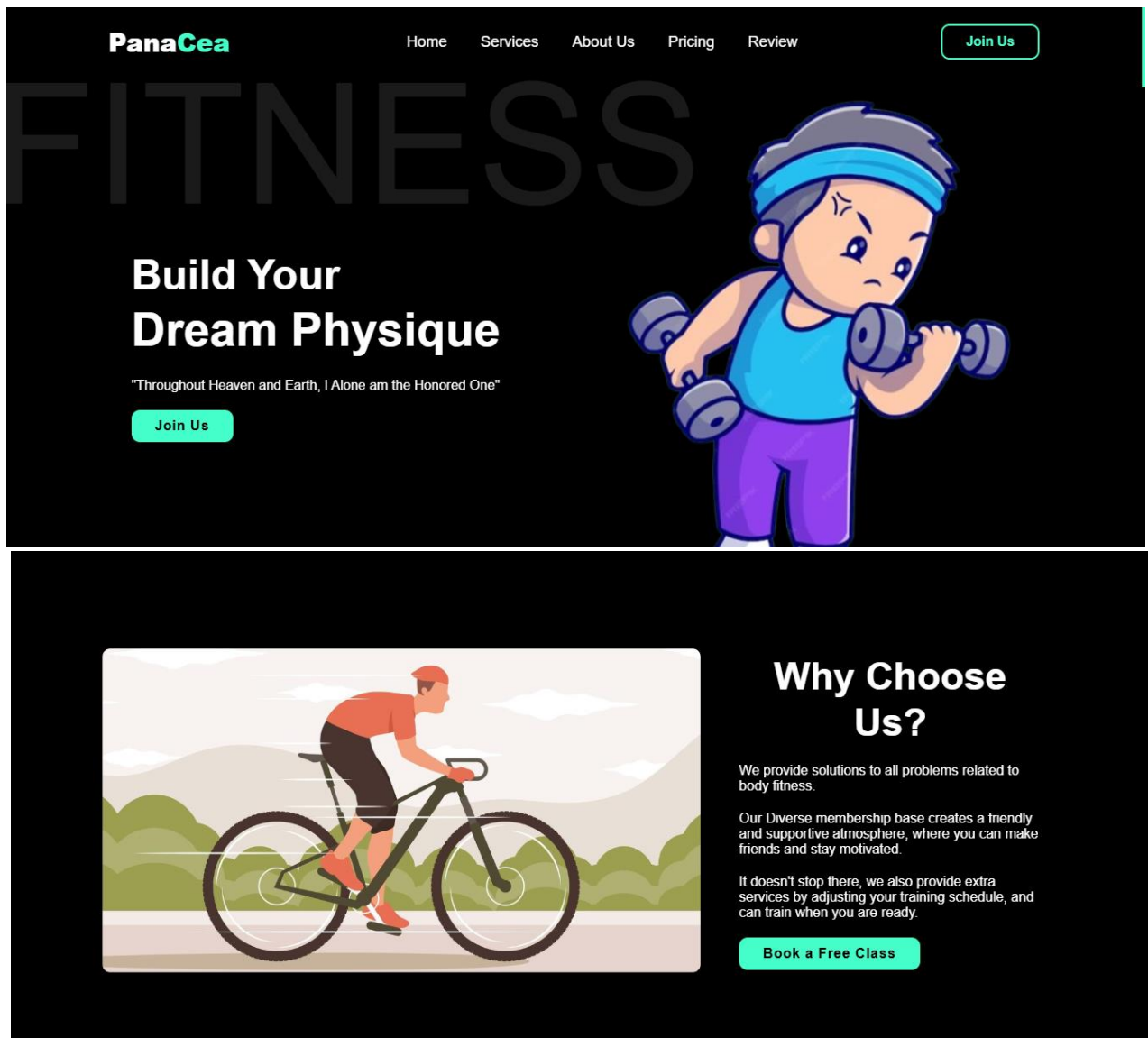
Sehingga HTML akan menjadi seperti ini.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css' rel='stylesheet'>
  <title>PanaCea</title>
  <link rel="icon" type="image/png" href="../static/logoo.png">
  <link rel="stylesheet" href="../static/home.css">
</head>
```

Berikut adalah hasil outpunya.



4.4. Output dan Input home.html



HTML

```
<header>
  <a href="#home" class="logo">Pana<span>Cea</span></a>

  <div class='bx bx-menu' id="menu-icon"></div>

  <ul class="navbar">
    <li><a href="#home">Home</a></li>
    <li><a href="#services">Services</a></li>
    <li><a href="#about">About Us</a></li>
    <li><a href="#pricing">Pricing</a></li>
    <li><a href="#review">Review</a></li>
  </ul>

  <div class="top-btn">
    <a href="{ { url_for('login') } }" class="nav-btn">Join Us</a>
  </div>

</header>
<section class="home" id="home">
  <div class="home-content">

    <h3>Build Your</h3>
    <h1>Dream Physique</h1>
    <h3><span class="multiple-text"></span></h3>

    <p>"Throughout Heaven and Earth, I Alone am the Honored One"</p>

    <a href="{ { url_for('login') } }" class="btn">Join Us</a>

  </div>

  <div class="home-img">
    
  </div>

</section>
```

CSS

```

section {
  min-height: 100vh;
  padding: 10rem 8% 2rem;
}

header {
  position: fixed;
  width: 100%;
  top: 0;
  right: 0;
  z-index: 1000;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 2rem 9%;
  background: rgba(0, 0, 0, 0.5);
  backdrop-filter: blur(10px);
  transition: all 0.5s ease;
}

.navbar {
  display: flex;
}

.navbar a {
  font-size: 1.8rem;
  font-weight: 500;
  color: var(--text-color);
  margin-left: 4rem;
  transition: all 0.5s ease;
  border-bottom: 3px solid transparent;
}

.navbar a:hover,
.navbar a.active {
  color: var(--main-color);
  border-bottom: 3px solid var(--main-color);
}

```

```

.home {
  display: flex;
  justify-content: center;
  align-items: center;
}

.home-content h3 {
  font-size: 5rem;
  font-weight: 700;
}

span {
  color: var(--main-color);
}

.home-content span {
  font-size: 4rem;
  font-weight: 550;
}

.home-content h1 {
  font-size: 5.6rem;
  font-weight: 700;
  line-height: 1.3;
}

.home-img img {
  width: 45vw;
}

.home-img::before {
  content: 'FITNESS';
  position: absolute;
  top: 10%;
  left: 30%;
  transform: translateX(-50%);
  font-size: 20rem;
  font-weight: 400;
  line-height: 20rem;
  color: var(--text-color);
  opacity: 0.1;
  z-index: -1;
}

.home-content p {
  font-size: 1.6rem;
  margin-bottom: 2rem;
  margin-top: 2rem;
}

```



JavaScript

```
let menu = document.querySelector('#menu-icon');
let navbar = document.querySelector('.navbar');

menu.onclick = () => {
  menu.classList.toggle('bx-x');
  navbar.classList.toggle('active');
}

window.onscroll = () => {
  menu.classList.remove('bx-x');
  navbar.classList.remove('active');
}

const typed = new Typed('.multiple-text', {
  strings: ['Physical Fitness', 'Weight Gain', 'Strength Training', 'Fat Lose', 'Weightlifting', 'Endurance'],
  typeSpeed: 60,
  backSpeed: 60,
  backDelay: 1000,
  loop: true,
});
```

4.4. Output dan Input register.html & login.html

HTML Register

```
<!-- Form -->
<div class="container">
  <h2>Masukan Menu Anda</h2>
  <form id="pembayaran">
    <label for="nama">Nama :</label>
    <input type="text" id="nama" placeholder="Masukan Nama Anda" required>

    <label for="menu">Nama Menu: </label>
    <input type="text" id="menu" readonly>

    <label for="harga">Harga: </label>
    <input type="text" id="harga" readonly>

    <label for="jumlah">Jumlah: </label>
    <input type="number" id="jumlah" min="1" max="5" placeholder="Masukkan jumlah barang" required>

    <div class="shop-btn-container">
      <button type="button" class="btn" onclick="redirectToStatus()">Pesan</button>
    </div>
  </form>
</div>
```

Login

```
<div class="container">
  <h1>Login</h1>
  {% with messages = get_flashed_messages(with_categories=true) %}
  {% if messages %}
    {% for category, message in messages %}
      <div style="...">
        {{ message }}
      </div>
    {% endfor %}
  {% endif %}
  {% endwith %}
  <br>
  <form method="POST" action="/login">
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" autocomplete="off" required>

    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <label onclick="togglePassword()" style="...">Show</label>
    <button type="submit">Login</button>
  </form>
  <p>Don't have an account? <a href="{{ url_for('register') }}">Register here</a>.</p>
</div>
```


CSS

```
.container {
  background-color: #000;
  padding: 2rem;
  border: 1px solid transparent;
  border-radius: 10px;
  box-shadow: 0 0 5px var(--main-color);
  width: 100%;
  max-width: 400px;
  text-align: center;
  transition: background 0.3s, border 0.3s;
}

h1 {
  color: #fff;
  margin-bottom: 1.5rem;
  font-size: 2rem;
}

form {
  display: flex;
  flex-direction: column;
}

label {
  text-align: left;
  margin-bottom: 0.5rem;
  color: #ccc;
}

input[type="text"],
input[type="email"],
input[type="password"] {
  padding: 0.75rem;
  margin-bottom: 1rem;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 1rem;
  outline: none;
  transition: border 0.3s;
}

input[type="text"]:focus,
input[type="email"]:focus,
input[type="password"]:focus {
  border-color: #007bff;
}

.dark-mode {
  background-color: var(--bg-dark);
  color: var(--text-dark);
}

.dark-mode .container {
  background-color: #2a2a2a;
  border-color: var(--border-dark);
}
```

Kode lebih lengkapnya ada di GitHub


<https://github.com/Rullyy32>

JavaScript

```
function togglePassword() {
  let passwordInput = document.getElementById("password");
  let label = document.querySelector("label[onclick]");

  if (passwordInput.type === "password") {
    passwordInput.type = "text";
    label.textContent = "Hide";
  } else {
    passwordInput.type = "password";
    label.textContent = "Show";
  }
}
```

DAFTAR PUSTAKA

	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: 27 Feb. 2025	<div>TI</div>	<div>SE</div>	<div>04</div>	<div>VT</div>	Page 16 - 19
	Program Keahlian: TI-SE-TE	Universal Internet of Things	Rev.: 06/03/2025	Dibuat oleh: EKO-Jagad-Rully				

- [1] <https://boxicons.com/>
- [2] <https://www.youtube.com/@WebDevCampus/videos>
- [3] <https://www.freepik.com/>
- [4] <https://app.diagrams.net/>

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

	Bidang Keahlian: Teknologi Informasi	Lab: SE - 209	Tanggal: 27 Feb. 2025	<div>TI</div>	<div>SE</div>	<div>04</div>	<div>VT</div>	Page 18 - 19
	Program Keahlian: TI-SE-TE	Universal Internet of Things	Rev.: 06/03/2025	Dibuat oleh: EKO-Jagad-Rully				