**Laporan Praktikum Membuat Buku Favorit  
Desain Web**

**Kelas A**

****

Oleh :

Adjie Firmansyah

4523210004

Dosen :

# **Adi Wahyu Pribadi, S.Si., M.Kom**

**S1 Teknik Informatika**

**Universitas Pancasila**

**index.html**

<!DOCTYPE html>

<html lang="id">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Aplikasi Buku Favorit</title>

<link rel="stylesheet" href="sytle.css" />

</head>

<body>

<h1>Aplikasi Buku Favorit</h1>

<!-- Input Nama Pengguna -->

<div id="session-container">

<input type="text" id="namaPengguna" placeholder="Masukkan Nama Anda" />

<button id="btnSimpanNama">Simpan Nama</button>

<h3 id="salamPengguna"></h3>

</div>

<!-- Form Tambah Buku -->

<form id="form-tambah-buku">

<input type="text" id="judul" placeholder="Judul Buku" required />

<input type="text" id="penulis" placeholder="Penulis" required />

<input type="number" id="tahun" placeholder="Tahun Terbit" required />

<button type="submit">Tambah Buku</button>

</form>

<!-- Daftar Buku -->

<h2>Daftar Buku</h2>

<div id="daftar-buku"></div>

<!-- Buku Favorit -->

<h2>Buku Favorit</h2>

<div id="buku-favorit"></div>

<script src="script.js"></script>

</body>

</html>

**Penjelasan program HTML**

**Bagian Head:**

* meta charset="UTF-8": Menentukan encoding karakter menjadi UTF-8.
* meta http-equiv="X-UA-Compatible" content="IE=edge": Menginstruksikan browser untuk menggunakan mode kompatibilitas terbaru.
* meta name="viewport": Mengatur tampilan agar sesuai dengan ukuran layar perangkat (responsive).
* title: Judul halaman web adalah "Aplikasi Buku Favorit".
* link: Menyertakan file CSS eksternal bernama sytle.css (namanya tampaknya salah ketik, seharusnya style.css) untuk styling halaman web.

**Bagian Body:**

* **Header (h1)**: Menampilkan judul aplikasi, yaitu "Aplikasi Buku Favorit".
* **Session Pengguna**:
  + Terdapat input teks untuk memasukkan nama pengguna (input type="text" id="namaPengguna").
  + Sebuah tombol untuk menyimpan nama pengguna (button id="btnSimpanNama").
  + Setelah nama disimpan, sebuah salam atau pesan selamat datang akan ditampilkan pada elemen h3 id="salamPengguna".
* **Form Tambah Buku**:
  + Formulir ini digunakan untuk menambahkan buku ke daftar. Terdapat tiga input:
    - input type="text" id="judul": Untuk memasukkan judul buku.
    - input type="text" id="penulis": Untuk memasukkan nama penulis buku.
    - input type="number" id="tahun": Untuk memasukkan tahun terbit buku.
  + Tombol "Tambah Buku" akan menambah buku ke daftar saat formulir dikirim.
* **Daftar Buku**:
  + Elemen div id="daftar-buku" akan menampilkan daftar buku yang sudah ditambahkan.
* **Buku Favorit**:
  + Elemen div id="buku-favorit" menampilkan daftar buku yang ditandai sebagai favorit oleh pengguna.

**Script:**

* File JavaScript eksternal script.js digunakan untuk menjalankan interaksi pada halaman, seperti menyimpan nama pengguna, menambahkan buku, dan menampilkan daftar buku serta buku favorit.

**script.js**

// Definisi Class Buku

class Buku {

constructor(judul, penulis, tahun) {

this.judul = judul;

this.penulis = penulis;

this.tahun = tahun;

}

tampilkanInfo() {

return `${this.judul} oleh ${this.penulis} (${this.tahun})`;

}

}

// Array untuk menyimpan daftar buku dan buku favorit

let daftarBuku = [];

let bukuFavorit = [];

// Mengambil elemen DOM

const formTambahBuku = document.getElementById('form-tambah-buku');

const divDaftarBuku = document.getElementById('daftar-buku');

const divBukuFavorit = document.getElementById('buku-favorit');

const btnSimpanNama = document.getElementById('btnSimpanNama');

const salamPengguna = document.getElementById('salamPengguna');

// Event Listener untuk Form Tambah Buku

formTambahBuku.addEventListener('submit', function (e) {

e.preventDefault();

tambahBuku();

});

// Fungsi untuk menambahkan buku ke daftar

function tambahBuku() {

const judul = document.getElementById('judul').value;

const penulis = document.getElementById('penulis').value;

const tahun = document.getElementById('tahun').value;

// Validasi input

if (judul === '' || penulis === '' || tahun === '') {

alert('Semua kolom harus diisi!');

return;

}

const bukuBaru = new Buku(judul, penulis, tahun);

daftarBuku.push(bukuBaru);

simpanDaftarBuku(); // Simpan daftar buku setelah menambahkan buku baru

tampilkanDaftarBuku();

formTambahBuku.reset();

}

// Fungsi untuk menyimpan daftar buku ke Local Storage

function simpanDaftarBuku() {

localStorage.setItem('daftarBuku', JSON.stringify(daftarBuku));

}

// Fungsi untuk menampilkan daftar buku

function tampilkanDaftarBuku() {

divDaftarBuku.innerHTML = '';

daftarBuku.forEach((buku, index) => {

const divBuku = document.createElement('div');

divBuku.classList.add('buku');

divBuku.innerHTML = `

<p>${buku.tampilkanInfo()}</p>

<button onclick="tambahKeFavorit(${index})">Tambah ke Favorit</button>

<button onclick="hapusBuku(${index})">Hapus</button>

`;

divDaftarBuku.appendChild(divBuku);

});

}

// Fungsi untuk menambahkan buku ke favorit

function tambahKeFavorit(index) {

const buku = daftarBuku[index];

// Cek apakah buku sudah ada di favorit

const sudahAda = bukuFavorit.some(favBuku => {

return favBuku.judul === buku.judul &&

favBuku.penulis === buku.penulis &&

favBuku.tahun === buku.tahun;

});

if (sudahAda) {

alert('Buku ini sudah ada di daftar favorit!');

return;

}

bukuFavorit.push(buku);

simpanBukuFavorit();

tampilkanBukuFavorit();

}

// Fungsi untuk menyimpan buku favorit ke Local Storage

function simpanBukuFavorit() {

localStorage.setItem('bukuFavorit', JSON.stringify(bukuFavorit));

}

// Fungsi untuk menampilkan buku favorit

function tampilkanBukuFavorit() {

divBukuFavorit.innerHTML = '';

bukuFavorit.forEach((buku, index) => {

const divBuku = document.createElement('div');

divBuku.classList.add('buku');

divBuku.innerHTML = `

<p>${buku.tampilkanInfo()}</p>

<button onclick="hapusDariFavorit(${index})">Hapus</button>

`;

divBukuFavorit.appendChild(divBuku);

});

}

// Fungsi untuk menghapus buku dari favorit

function hapusDariFavorit(index) {

bukuFavorit.splice(index, 1);

simpanBukuFavorit();

tampilkanBukuFavorit();

}

// Fungsi untuk menghapus daftar buku

function hapusBuku(index) {

daftarBuku.splice(index, 1);

simpanDaftarBuku();

tampilkanDaftarBuku();

}

// Event Listener untuk tombol simpan nama pengguna

btnSimpanNama.addEventListener('click', function () {

const nama = document.getElementById('namaPengguna').value;

if (nama === '') {

alert('Masukkan nama Anda!');

return;

}

sessionStorage.setItem('namaPengguna', nama);

tampilkanNamaPengguna();

document.getElementById('namaPengguna').value = '';

});

// Fungsi untuk menampilkan nama pengguna

function tampilkanNamaPengguna() {

const nama = sessionStorage.getItem('namaPengguna');

if (nama) {

salamPengguna.textContent = `Selamat datang, ${nama}!`;

} else {

salamPengguna.textContent = '';

}

}

// Memuat data saat halaman dimuat

window.onload = function () {

// Memuat daftar buku dari Local Storage

if (localStorage.getItem('daftarBuku')) {

const storedBooks = JSON.parse(localStorage.getItem('daftarBuku'));

// Rekonstruksi objek buku menjadi instance dari kelas Buku

daftarBuku = storedBooks.map(book => {

return new Buku(book.judul, book.penulis, book.tahun);

});

tampilkanDaftarBuku();

}

// Memuat buku favorit dari Local Storage

if (localStorage.getItem('bukuFavorit')) {

const storedFavorites = JSON.parse(localStorage.getItem('bukuFavorit'));

// Rekonstruksi objek buku menjadi instance dari kelas Buku

bukuFavorit = storedFavorites.map(book => {

return new Buku(book.judul, book.penulis, book.tahun);

});

tampilkanBukuFavorit();

}

// Menampilkan nama pengguna dari Session Storage

tampilkanNamaPengguna();

};

**Penjelasan program JavaScript**

#### **Class Buku**

* class Buku: Sebuah kelas yang merepresentasikan sebuah buku dengan properti judul, penulis, dan tahun.
* tampilkanInfo(): Metode untuk mengembalikan string yang menampilkan informasi buku dalam format "Judul oleh Penulis (Tahun)".

#### **2. Menyimpan Daftar Buku dan Buku Favorit**

* Dua array daftarBuku dan bukuFavorit digunakan untuk menyimpan buku yang ditambahkan pengguna dan buku yang ditandai sebagai favorit.

#### **3. Elemen DOM**

* Beberapa elemen HTML diambil menggunakan getElementById() untuk digunakan dalam logika JavaScript, seperti formTambahBuku (form untuk menambah buku), divDaftarBuku (untuk menampilkan daftar buku), dan divBukuFavorit (untuk menampilkan daftar buku favorit).

#### **4. Event Listener**

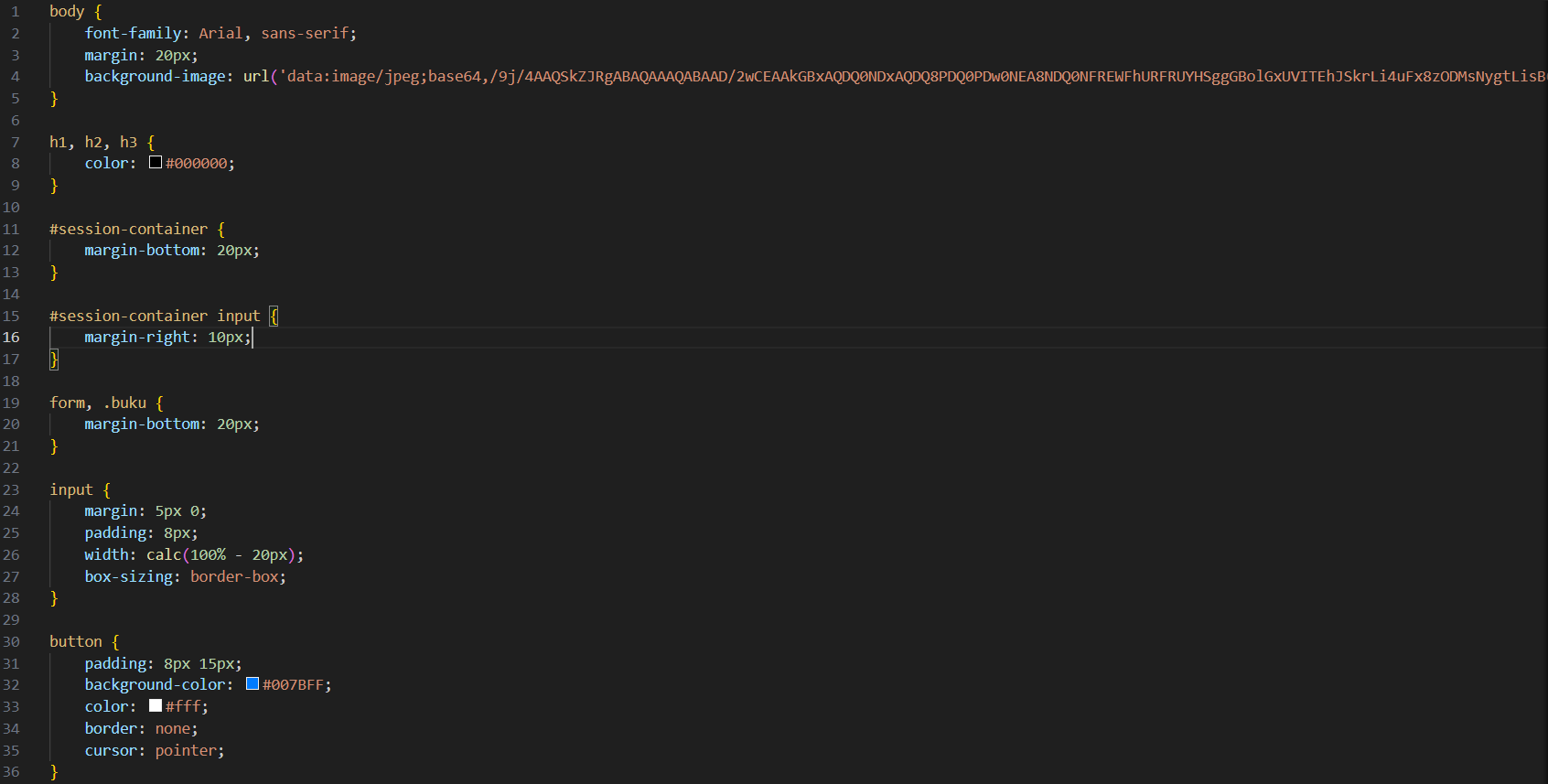
* *Tambah Buku*: Saat pengguna menekan tombol "Tambah Buku", form disubmit dan buku baru ditambahkan ke daftarBuku. Program juga menvalidasi input untuk memastikan bahwa semua kolom (judul, penulis, tahun) telah diisi.
* *Simpan Nama Pengguna*: Saat pengguna memasukkan nama dan menekan tombol "Simpan Nama", nama pengguna disimpan dalam *Session Storage* dan ditampilkan salam selamat datang di halaman.

#### **5. Fungsi Utama:**

* **tambahBuku()**: Fungsi ini mengambil nilai dari input pengguna (judul, penulis, tahun), membuat objek Buku baru, lalu menambahkannya ke array daftarBuku. Data buku kemudian disimpan ke *Local Storage* dan daftar buku diperbarui.
* **tampilkanDaftarBuku()**: Fungsi ini menampilkan seluruh buku dalam daftarBuku di halaman. Setiap buku ditampilkan dengan dua tombol:
  + *Tambah ke Favorit*: Untuk menambahkan buku ke dalam daftar favorit.
  + *Hapus*: Untuk menghapus buku dari daftar.
* **tambahKeFavorit()**: Fungsi untuk menambahkan buku dari daftarBuku ke bukuFavorit. Program mengecek apakah buku tersebut sudah ada di favorit untuk menghindari duplikasi.
* **tampilkanBukuFavorit()**: Fungsi untuk menampilkan buku-buku favorit di area "Buku Favorit". Setiap buku di favorit memiliki tombol "Hapus" untuk mengeluarkannya dari daftar favorit.
* **hapusDariFavorit()** dan **hapusBuku()**: Masing-masing berfungsi untuk menghapus buku dari daftar favorit atau daftar buku secara keseluruhan.
* **simpanDaftarBuku()** dan **simpanBukuFavorit()**: Kedua fungsi ini digunakan untuk menyimpan array daftarBuku dan bukuFavorit ke dalam *Local Storage* agar data tetap tersimpan meskipun halaman direfresh.
* **tampilkanNamaPengguna()**: Menampilkan nama pengguna yang telah disimpan di *Session Storage* sebagai salam "Selamat datang".

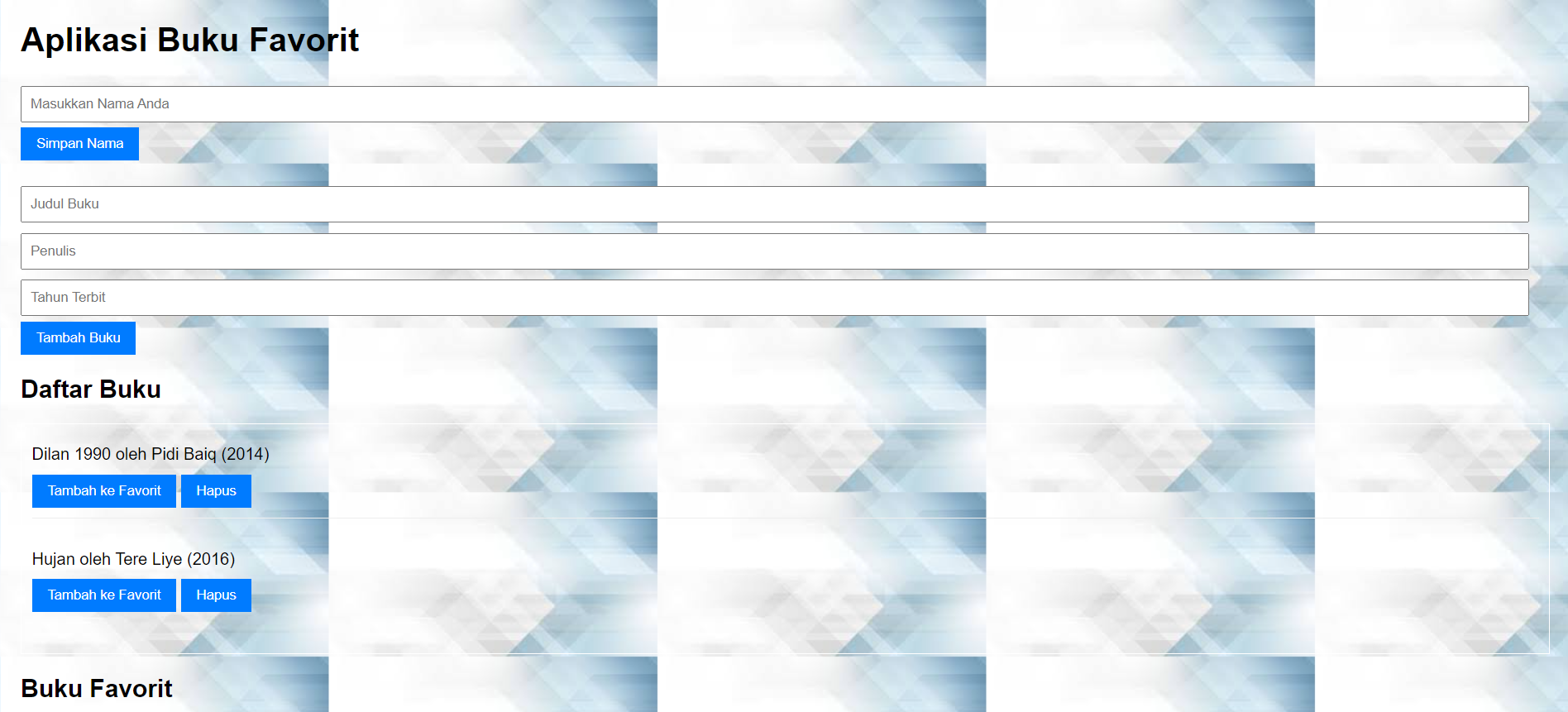
#### **6. Memuat Data Saat Halaman Dibuka**

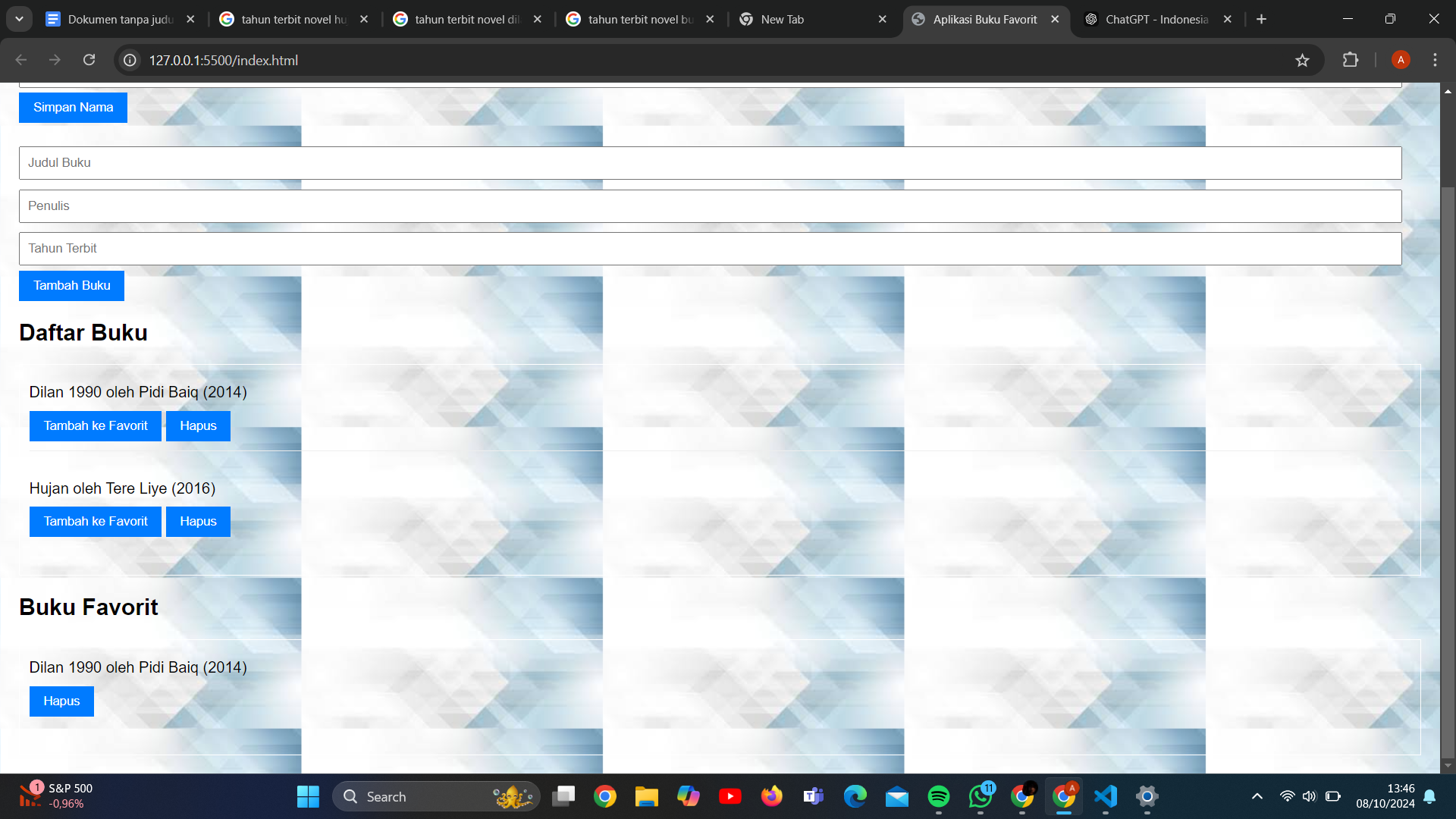
* Ketika halaman dimuat ulang, aplikasi memuat kembali data yang sudah tersimpan di *Local Storage* (daftar buku dan buku favorit) dan merekonstruksi objek Buku. Selain itu, nama pengguna juga dimuat dari *Session Storage* dan ditampilkan.

**style.css**  
  


Tampilan HTML



Menambahkan buku  


Menandai buku favorit  


Memasukan nama pengguna  
