**LAPORAN PRAKTIKUM**

**JAVASCRIPT – MEMBUAT BUKU FAVORIT**

****

Disusun oleh:

Nama : Daffa Yassar Ahmad (4523210032)

Mata Kuliah : Praktikum Desain Web – A

Tanggal : 8 Oktober 2024

Dosen Pengampu:

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

**PROGRAM SARJANA TEKNIK INFORMATIKA**

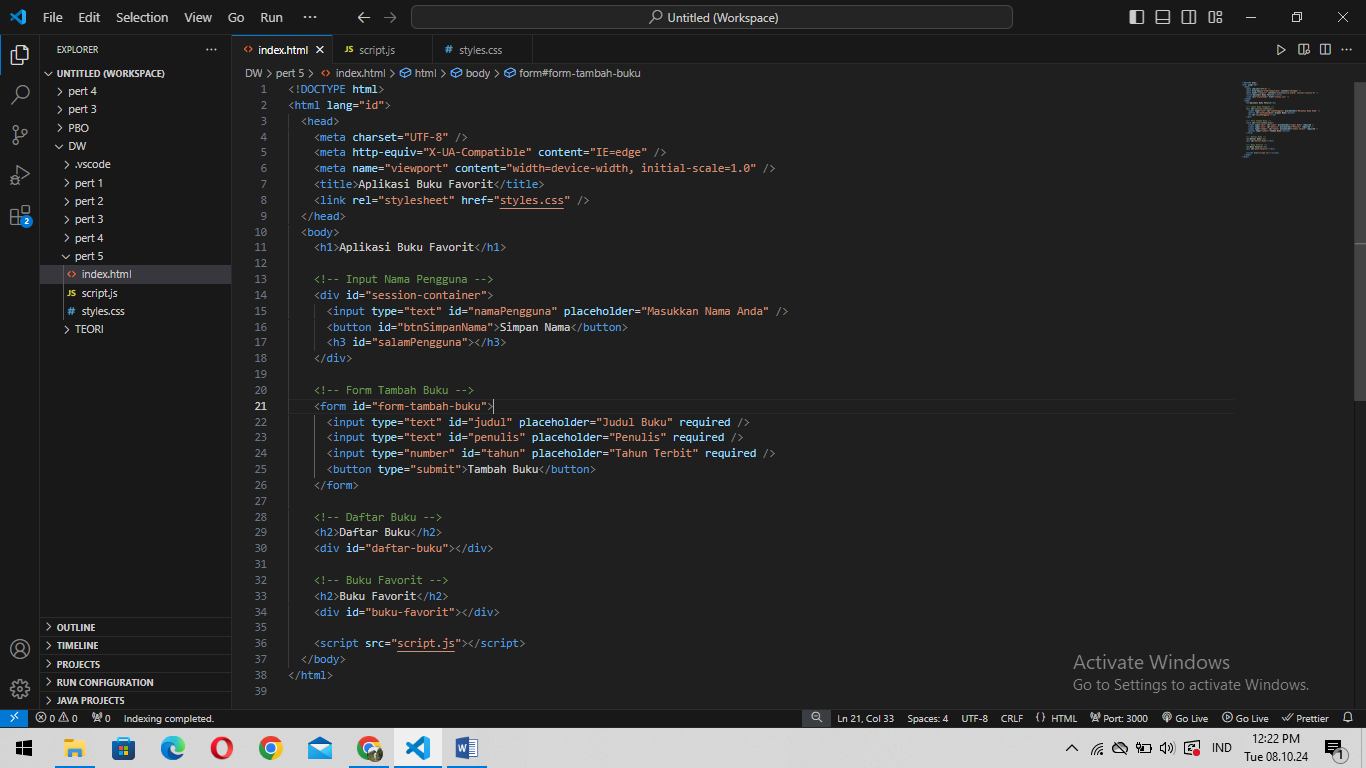
**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024**

index.html



Penjelasan Setiap Kode :

<h1>Aplikasi Buku Favorit</h1>

Heading utama dalam halaman dalam menyatakan nama aplikasi

 <div id="session-container">

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

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

      <h3 id="salamPengguna"></h3>

    </div>

Container untuk input nama pengguna, tombol simpan, dan ucapan salam.

<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>

Form untuk menginput teks untuk memasukkan nama pengguna dengan placeholder yang memberi instruksi.

<h2>Daftar Buku</h2>

    <div id="daftar-buku"></div>

Div kosong yang akan diisi dengan daftar buku secara dinamis menggunakan JavaScript.

 <!-- Buku Favorit -->

    <h2>Buku Favorit</h2>

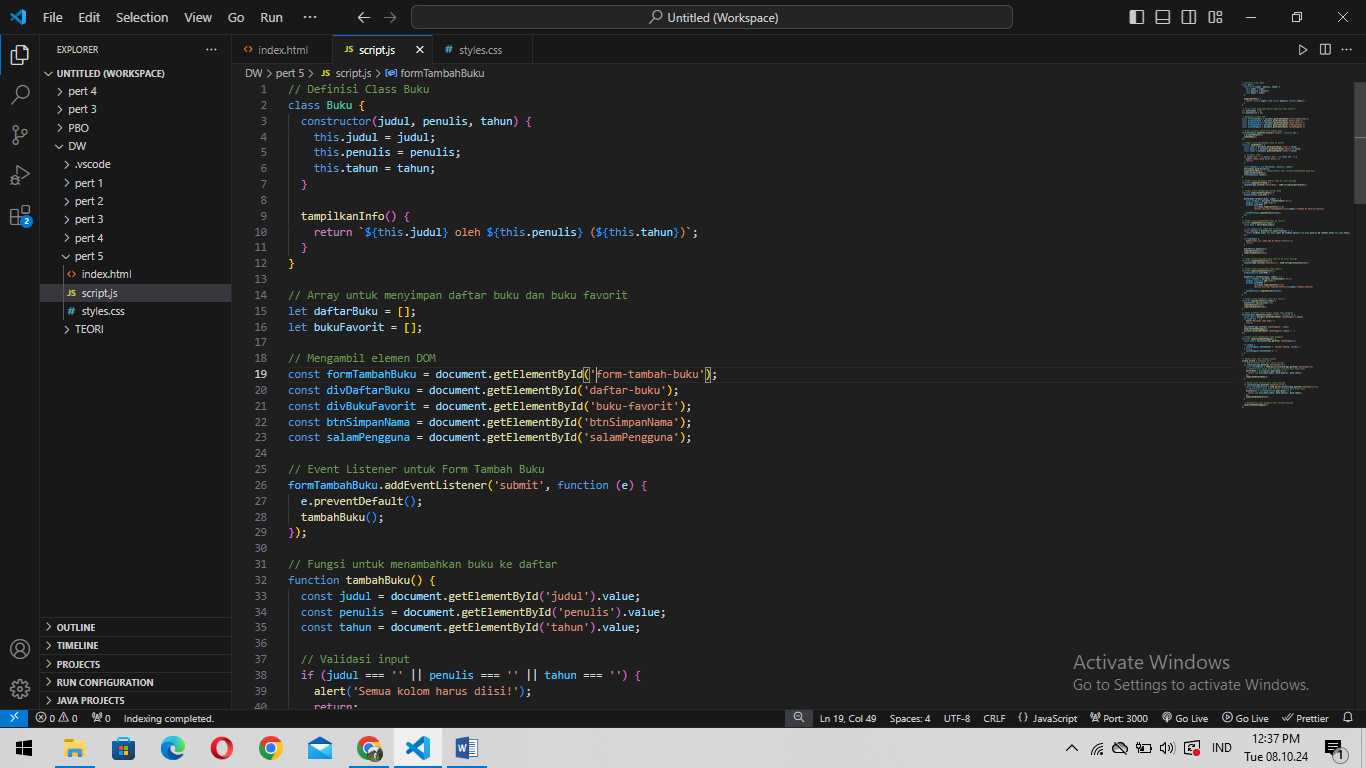
    <div id="buku-favorit"></div>

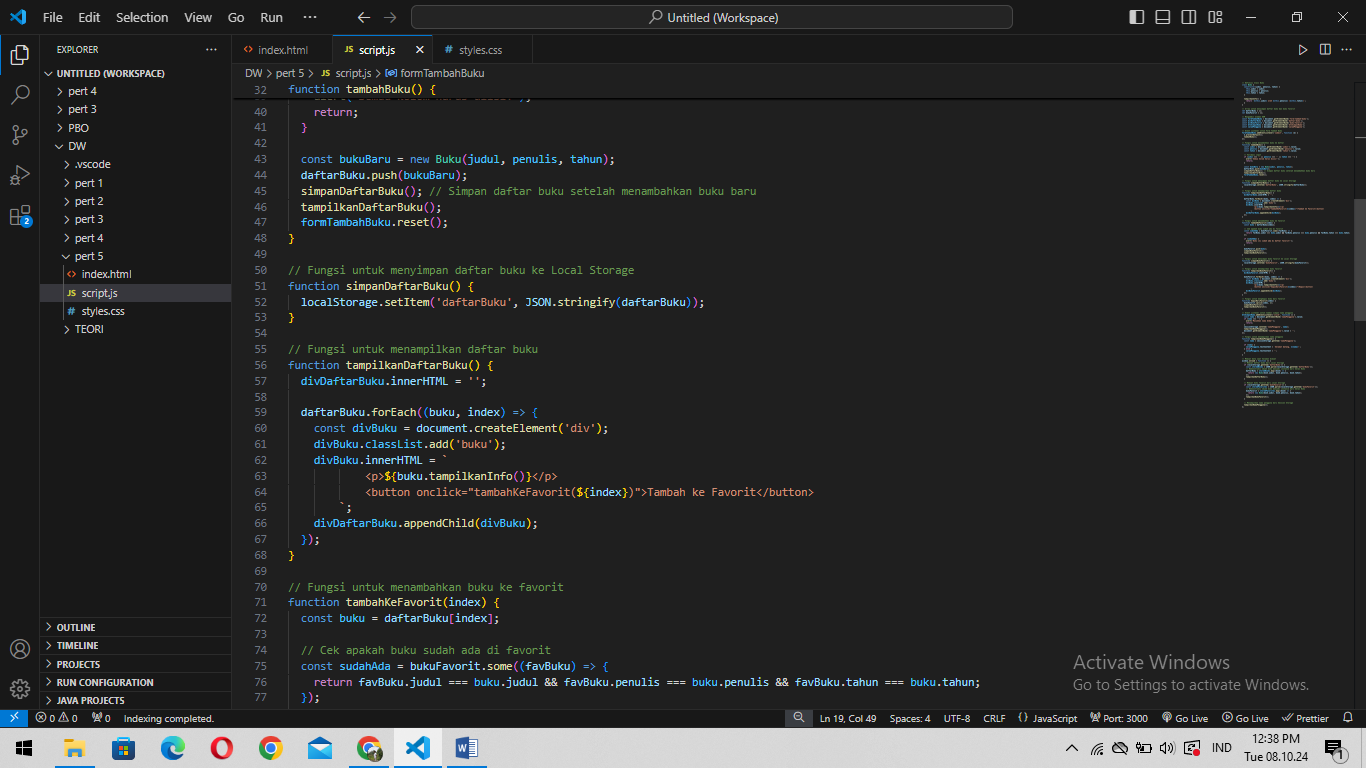
Div kosong yang akan diisi dengan Buku Favorit secara dinamis menggunakan JavaScript.

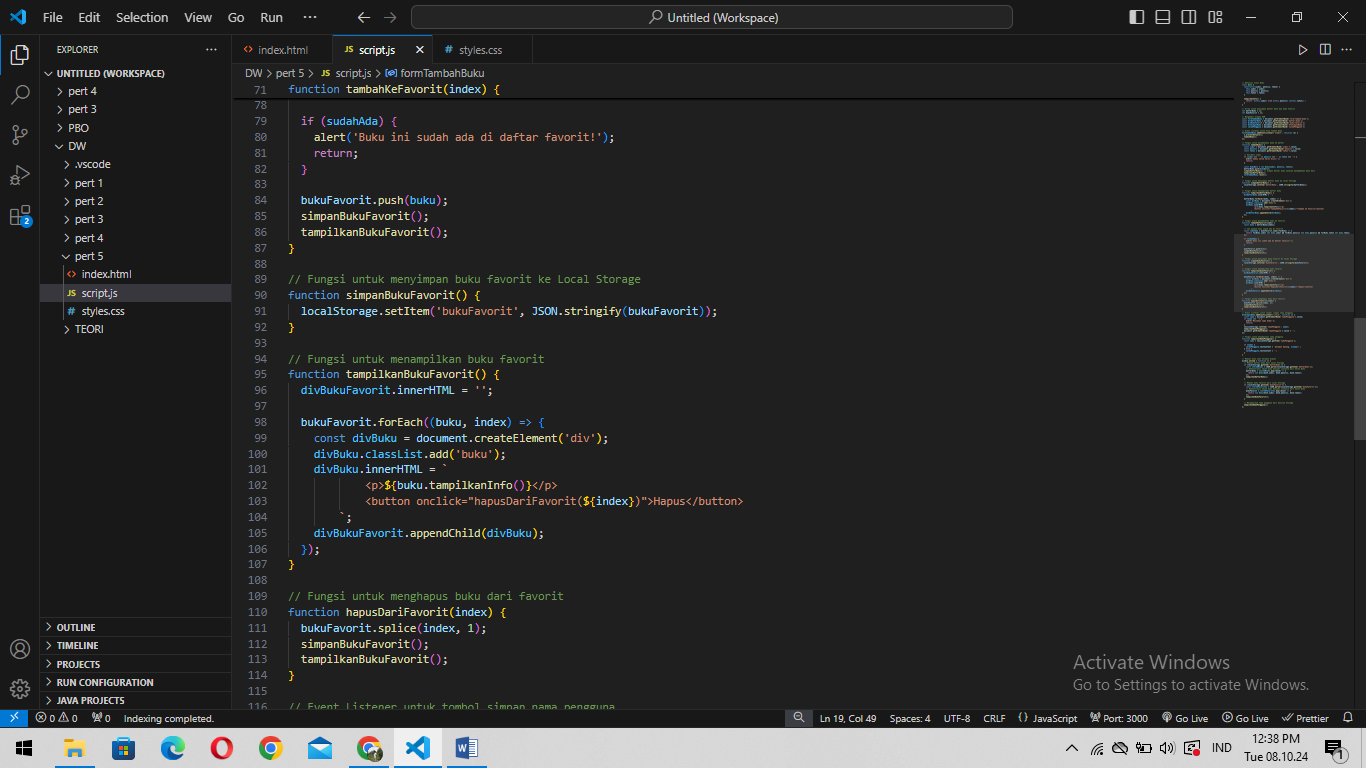
<script src="script.js"></script>

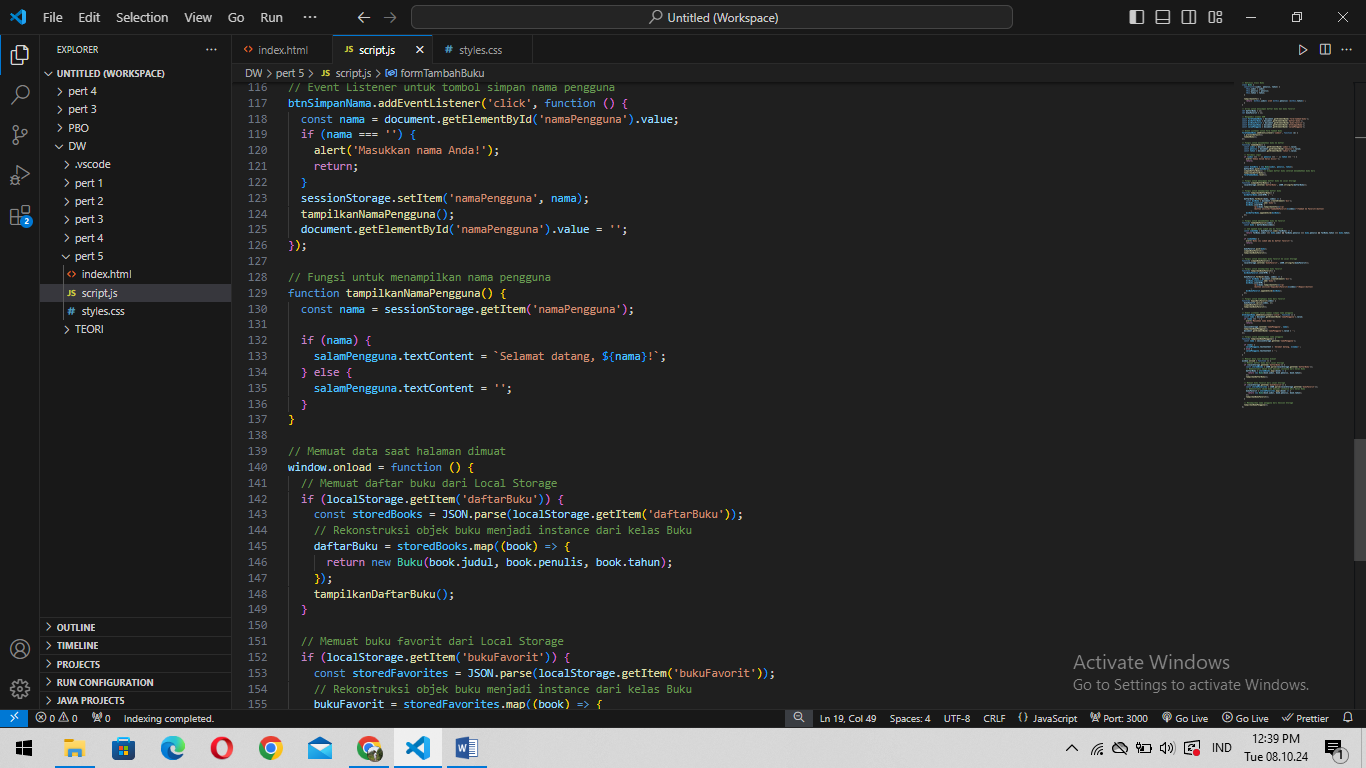
Digunakan untuk mengubungkan ke file script.js dan menjalankan fungsi-fungsi logika interaktif

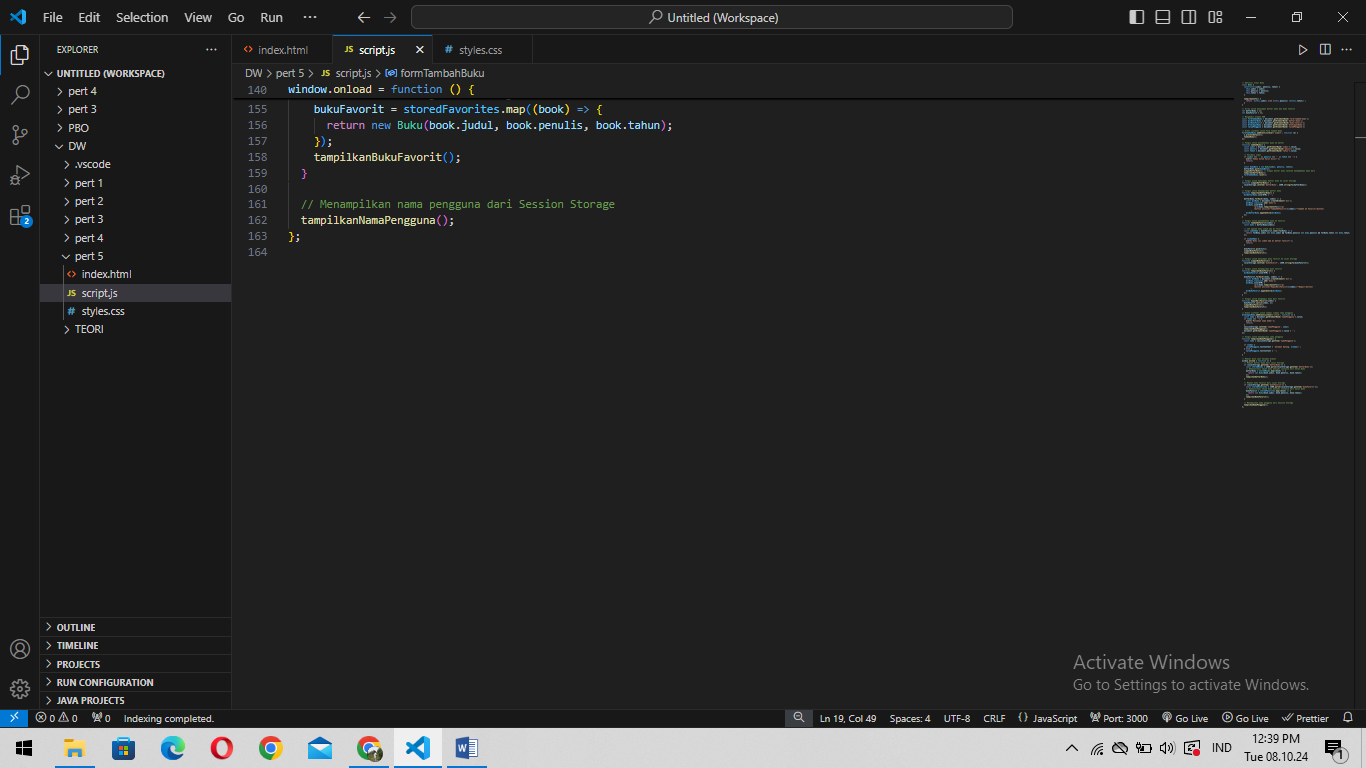
Script.js











Penjelasan Kode :

class Buku {

  constructor(judul, penulis, tahun) {

    this.judul = judul;

    this.penulis = penulis;

    this.tahun = tahun;

  }

  tampilkanInfo() {

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

  }

}

**Class Buku**: Ini adalah blueprint untuk membuat objek buku, yang berisi properti judul, penulis, dan tahun.

* **Constructor**: Digunakan untuk membuat objek baru dari class Buku dengan nilai properti judul, penulis, dan tahun.
* **Method tampilkanInfo()**: Mengembalikan string yang menggabungkan informasi buku, berupa judul, penulis, dan tahun terbit.

let daftarBuku = [];

let bukuFavorit = [];

 **daftarBuku**: Menyimpan buku-buku yang ditambahkan oleh pengguna.

 **bukuFavorit**: Menyimpan buku-buku favorit yang dipilih dari daftar buku.

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');

 **formTambahBuku**: Mengambil elemen form tambah buku.

 **divDaftarBuku**: Mengambil elemen div tempat menampilkan daftar buku.

 **divBukuFavorit**: Mengambil elemen div tempat menampilkan buku favorit.

 **btnSimpanNama**: Mengambil elemen tombol untuk menyimpan nama pengguna.

 **salamPengguna**: Mengambil elemen h3 yang digunakan untuk menampilkan ucapan selamat kepada pengguna.

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

  e.preventDefault();

  tambahBuku();

});

 Menambahkan event listener untuk menangani aksi "submit" dari form tambah buku.

 e.preventDefault() mencegah halaman reload ketika form disubmit, lalu memanggil fungsi tambahBuku()

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 tambahBuku()**: Mengambil nilai judul, penulis, dan tahun dari form.

* Melakukan validasi: Jika ada kolom kosong, menampilkan alert.
* Jika valid, membuat instance baru dari class Buku dan menambahkannya ke array daftarBuku.
* Menyimpan daftar buku ke Local Storage dengan simpanDaftarBuku().
* Menampilkan daftar buku dengan tampilkanDaftarBuku().
* Mengosongkan form menggunakan formTambahBuku.reset().

function simpanDaftarBuku() {

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

}

**simpanDaftarBuku()**: Menyimpan daftarBuku dalam format JSON ke Local Storage.

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>

        `;

    divDaftarBuku.appendChild(divBuku);

  });

}

**tampilkanDaftarBuku()**: Mengosongkan konten elemen divDaftarBuku, lalu membuat elemen HTML untuk setiap buku yang ada di array daftarBuku, dan menampilkannya di halaman.

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();

}

**tambahKeFavorit()**: Menambahkan buku ke array bukuFavorit.

* Cek apakah buku sudah ada di daftar favorit dengan some(). Jika sudah ada, tampilkan alert dan hentikan proses.
* Jika belum, tambahkan buku ke bukuFavorit, simpan dengan simpanBukuFavorit(), dan tampilkan dengan tampilkanBukuFavorit().

function simpanBukuFavorit() {

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

}

**simpanBukuFavorit()**: Menyimpan bukuFavorit dalam format JSON ke Local Storage.

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);

  });

}

**tampilkanBukuFavorit()**: Sama seperti tampilkanDaftarBuku(), namun ini untuk menampilkan buku yang telah ditambahkan ke favorit.

function hapusDariFavorit(index) {

  bukuFavorit.splice(index, 1);

  simpanBukuFavorit();

  tampilkanBukuFavorit();

}

**hapusDariFavorit()**: Menghapus buku dari bukuFavorit berdasarkan indeks, menyimpan perubahan, dan memperbarui tampilan.

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 = '';

});

Ketika tombol "Simpan Nama" diklik, nama pengguna disimpan di sessionStorage dan ditampilkan di halaman.

function tampilkanNamaPengguna() {

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

  if (nama) {

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

  } else {

    salamPengguna.textContent = '';

  }

}

**tampilkanNamaPengguna()**: Menampilkan nama pengguna yang tersimpan di sessionStorage pada elemen salamPengguna.

**TAMPILAN**

