

Laporan Praktikum



Dosen Pengampu :

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

Disusun oleh :

Mufti Fazli (4523210067)

Matakuliah :

Pratikum Desain Web_A

Hari dan tanggal :

Selasa 8 Oktober 2024

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS PANCASILA 2024

1. <!DOCTYPE html>

<html lang="id"> **berfungsi untuk mendeklarasikan variable**

2. <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="style.css" />

</head>

= Berfungsi untuk membuat title di bagian heading

3. <body>

<h1>Aplikasi Buku Favorit</h1> = **untuk membuat namanya**

<!-- Input Nama Pengguna --> = **berfungsi untuk menginput 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> = **Elemen div yang membungkus input dan tombol untuk menyimpan nama pengguna.**

<!-- 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>
```

Bagian body ini berfungsi untuk membuat bagian isinya yang di dalam body

2. // 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 = []; = Array untuk menyimpan daftar buku dan buku favorit
```

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>
        `;
        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();

}

```

// 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 => {
```

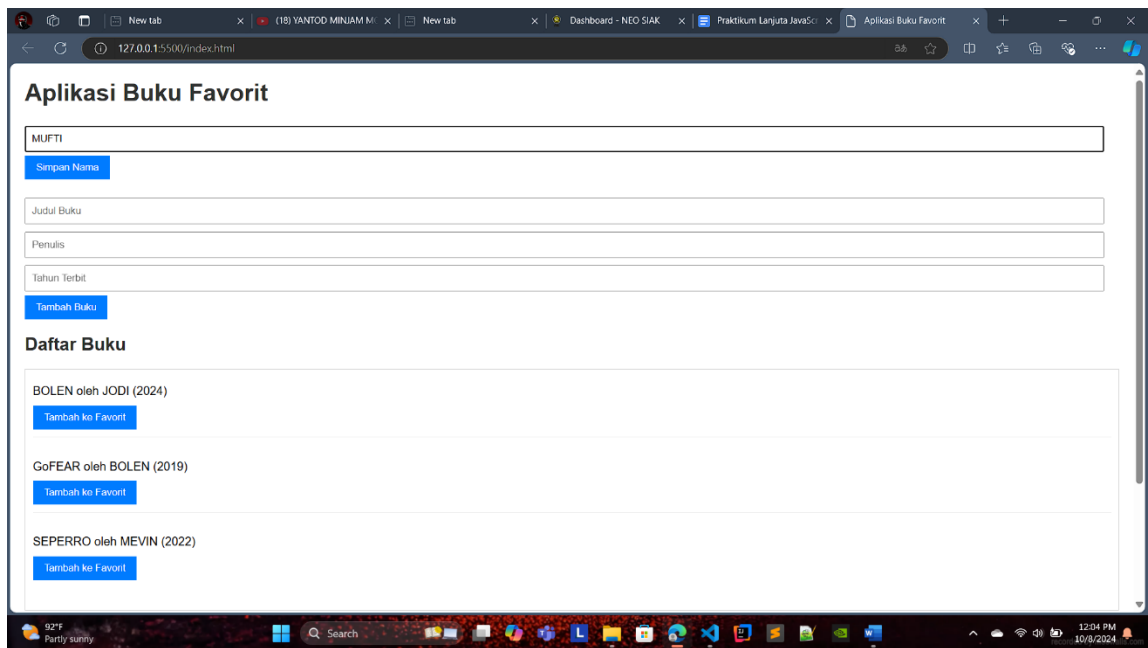


```
        return new Buku(book.judul, book.penulis, book.tahun);  
    });  
    tampilkanBukuFavorit();  
}
```

Menampilkan nama pengguna dari Session Storage =

```
tampilkanNamaPengguna();  
};
```

1. MEMBUAT NAMA USER 'MUFTI'



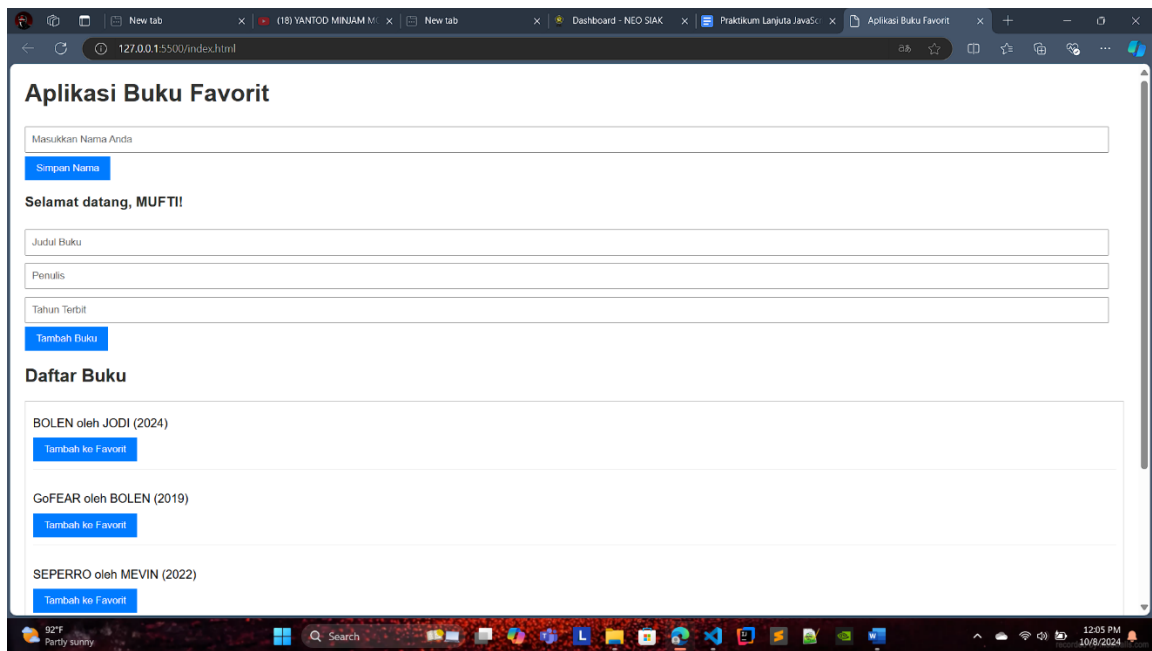
The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5500/index.html'. The page title is 'Aplikasi Buku Favorit'. The form contains the following fields and buttons:

- Username:** A text input field containing 'MUFTI' with a blue 'Simpan Nama' button below it.
- Judul Buku:** A text input field.
- Penulis:** A text input field.
- Tahun Terbit:** A text input field.
- Tambah Buku:** A blue button below the input fields.

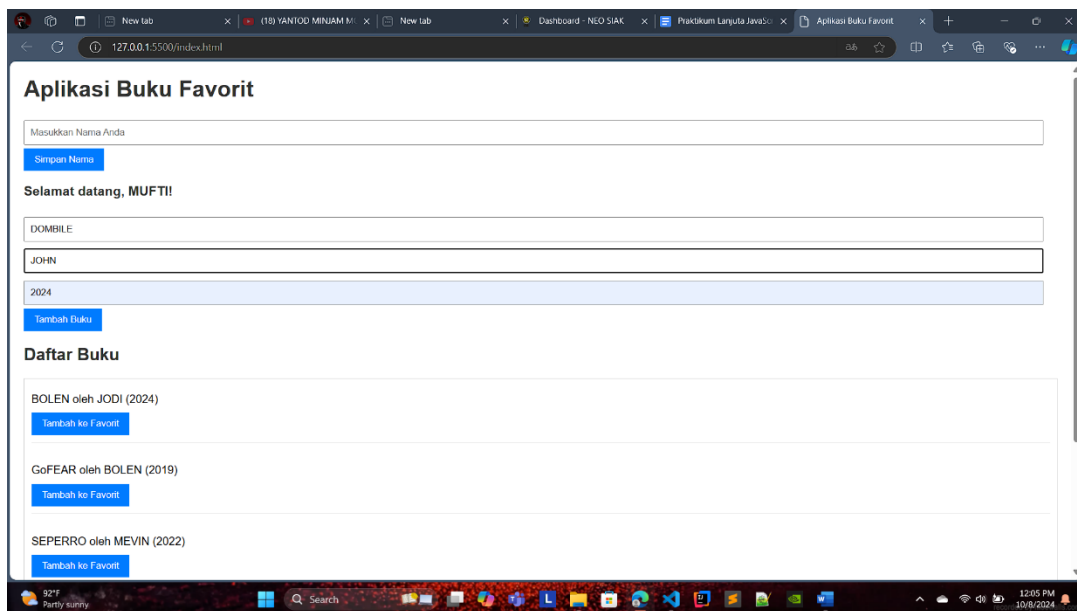
Below the form is a section titled 'Daftar Buku' containing a list of books with 'Tambah ke Favorit' buttons:

- BOLEN oleh JODI (2024) with a blue 'Tambah ke Favorit' button.
- GoFEAR oleh BOLEN (2019) with a blue 'Tambah ke Favorit' button.
- SEPERRO oleh MEVIN (2022) with a blue 'Tambah ke Favorit' button.

2. MENYIMPAN DAN MNAMPILKAN NAMA MUFTI



3.MENAMBAHKAN /MEMBUAT DAFTAR BUKU



4. MENAMBAHKAN BUKU BOLEN KE DALAM FAVORIT

