

# **Praktikum Desain Web - A**

*“Tugas 05”*



**Nama:**

Zidan Wali Arubusman/ 4523210001

**Dosen Pengampu:**

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

**PROGRAM STUDI TEKNIK INFORMATIKA  
UNIVERSITAS PANCASILA  
2024/2025**

- A. ) Berikan penjelasan dari setiap kode HTML dan JavaScript di atas  
Jawab :

```
<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="styles.css" />
</head>
```

## HTML

Bagian `<head>` berisi metadata yang memberi tahu browser tentang karakteristik halaman:

- `<meta charset="UTF-8" />`: Menentukan pengkodean karakter yang digunakan, yaitu UTF-8, yang mendukung hampir semua karakter dalam bahasa apapun.
- `<meta http-equiv="X-UA-Compatible" content="IE=edge" />`: Menginstruksikan agar halaman web ini di-render menggunakan standar browser terbaru.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0" />`: Mengatur tampilan responsif, memastikan halaman bisa tampil baik di perangkat mobile dan tablet.
- `<title>Aplikasi Buku Favorit</title>`: Menetapkan judul halaman yang akan ditampilkan di tab browser.
- `<link rel="stylesheet" href="styles.css" />`: Menyertakan file CSS eksternal `styles.css` untuk memberi gaya pada tampilan halaman

```
<div id="session-container">
  <input type="text" id="namaPengguna" placeholder="Masukkan Nama
Anda" />
  <button id="btnSimpanNama">Simpan Nama</button>
  <h3 id="salamPengguna"></h3>
</div>
```

Ini adalah sebuah div dengan ID `session-container` yang berisi:

- Input teks untuk memasukkan nama pengguna, dengan placeholder **"Masukkan Nama Anda"**.
- Tombol **Simpan Nama** yang kemungkinan digunakan untuk menyimpan nama pengguna.
- Elemen heading `<h3>` dengan ID `salamPengguna` yang mungkin akan digunakan untuk menampilkan salam kepada pengguna setelah nama disimpan.

```

<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 ini digunakan untuk menambah buku ke daftar, dengan tiga input:

- Input untuk **judul buku**.
- Input untuk **penulis**.
- Input untuk **tahun terbit** dalam format angka. Setiap input ini bersifat wajib (**required**), dan tombol **Tambah Buku** akan mengirimkan form ini.

## 1. Definisi Class Buku

javascript

```

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**: Mendefinisikan kelas **Buku** dengan properti **judul**, **penulis**, dan **tahun**.
- **tampilkanInfo()**: Metode ini mengembalikan informasi buku dalam format string yang rapi, berisi judul, penulis, dan tahun terbit.

## 2. Variabel untuk Menyimpan Data Buku

javascript

```

let daftarBuku = [];
let bukuFavorit = [];

```

- **daftarBuku**: Array ini menyimpan daftar buku yang ditambahkan pengguna.

- **bukuFavorit**: Array ini menyimpan daftar buku yang ditandai sebagai favorit oleh pengguna.

### 3. Pengambilan Elemen DOM

javascript

```
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');
```

Elemen-elemen HTML diambil menggunakan `getElementById` dan disimpan dalam variabel untuk memudahkan manipulasi DOM.

### 4. Event Listener untuk Form Tambah Buku

javascript

```
formTambahBuku.addEventListener('submit', function (e) {
    e.preventDefault();
    tambahBuku();
});
```

- **Event Listener ini** menangani saat formulir "Tambah Buku" dikirim. Fungsi `e.preventDefault()` mencegah reload halaman, dan memanggil fungsi `tambahBuku()` untuk menambahkan buku ke daftar.

### 5. Fungsi untuk Menambahkan Buku

javascript

```
function tambahBuku() {
    const judul = document.getElementById('judul').value;
    const penulis = document.getElementById('penulis').value;
    const tahun = document.getElementById('tahun').value;

    if (judul === '' || penulis === '' || tahun === '') {
        alert('Semua kolom harus diisi!');
        return;
    }

    const bukuBaru = new Buku(judul, penulis, tahun);
    daftarBuku.push(bukuBaru);
    simpanDaftarBuku();
}
```

```

    tampilkanDaftarBuku();
    formTambahBuku.reset();
}

```

- **Fungsi ini** mengambil nilai dari input pengguna, memvalidasinya, lalu membuat instance baru dari **Buku** dan menambahkannya ke **daftarBuku**.
- Setelah buku ditambahkan, daftar buku disimpan di Local Storage dan ditampilkan pada halaman.

## 6. Fungsi untuk Menyimpan dan Menampilkan Daftar Buku

javascript

```

function simpanDaftarBuku() {
    localStorage.setItem('daftarBuku', JSON.stringify(daftarBuku));
}

```

```

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);
    });
}

```

- **simpanDaftarBuku**: Menyimpan daftar buku ke Local Storage dengan format JSON.
- **tampilkanDaftarBuku**: Menampilkan daftar buku pada halaman dengan menambahkan elemen HTML untuk setiap buku yang disimpan.

## 7. Menambah Buku ke Favorit

javascript

```

function tambahKeFavorit(index) {
    const buku = daftarBuku[index];

    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 ini mengambil buku dari `daftarBuku` berdasarkan indeks, lalu mengecek apakah buku tersebut sudah ada di `bukuFavorit`. Jika belum, buku ditambahkan ke favorit.

## 8. Fungsi Menyimpan dan Menampilkan Buku Favorit

javascript

```

function simpanBukuFavorit() {
    localStorage.setItem('bukuFavorit',
JSON.stringify(bukuFavorit));
}

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);
    });
}

```

- **simpanBukuFavorit**: Menyimpan daftar buku favorit ke Local Storage.
- **tampilkanBukuFavorit**: Menampilkan daftar buku favorit di halaman.

## 9. Menghapus Buku dari Favorit

javascript

```
function hapusDariFavorit(index) {
    bukuFavorit.splice(index, 1);
    simpanBukuFavorit();
    tampilkanBukuFavorit();
}
```

- Menghapus buku dari `bukuFavorit` berdasarkan indeks dan memperbarui daftar di Local Storage dan halaman.

## 10. Event Listener untuk Menyimpan Nama Pengguna

javascript

```
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 = '';
});
```

- Menyimpan nama pengguna ke `sessionStorage` dan menampilkan pesan selamat datang.

## 11. Menampilkan Nama Pengguna

javascript

```
function tampilkanNamaPengguna() {
    const nama = sessionStorage.getItem('namaPengguna');

    if (nama) {
        salamPengguna.textContent = `Selamat datang, ${nama}!`;
    } else {
        salamPengguna.textContent = '';
    }
}
```

```
}
```

- Mengambil nama pengguna dari `sessionStorage` dan menampilkannya di halaman.

## 12. Memuat Data dari Local Storage saat Halaman Dimuat

javascript

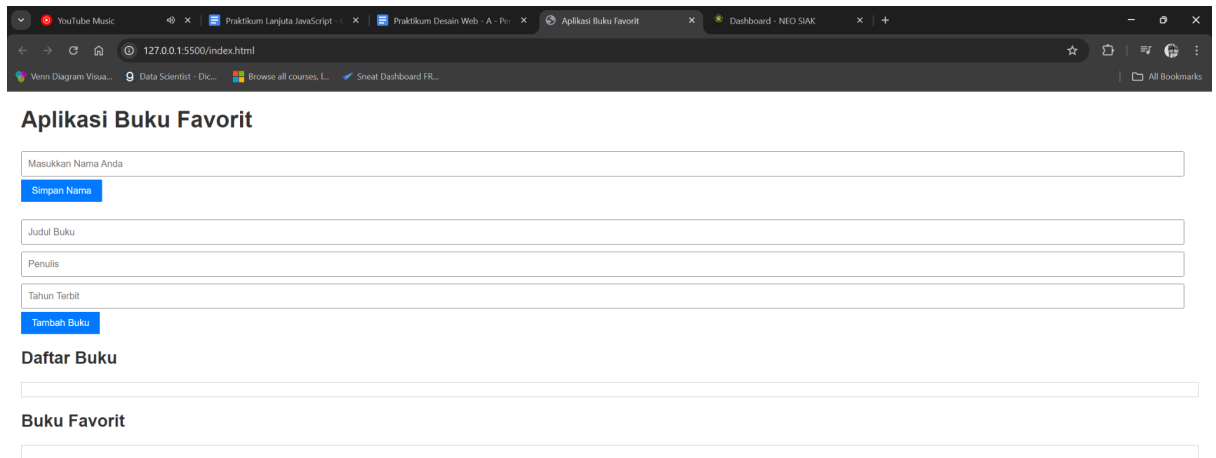
```
window.onload = function () {  
    if (localStorage.getItem('daftarBuku')) {  
        const storedBooks =  
JSON.parse(localStorage.getItem('daftarBuku'));  
        daftarBuku = storedBooks.map(book => new Buku(book.judul,  
book.penulis, book.tahun));  
        tampilkanDaftarBuku();  
    }  
  
    if (localStorage.getItem('bukuFavorit')) {  
        const storedFavorites =  
JSON.parse(localStorage.getItem('bukuFavorit'));  
        bukuFavorit = storedFavorites.map(book => new  
Buku(book.judul, book.penulis, book.tahun));  
        tampilkanBukuFavorit();  
    }  
  
    tampilkanNamaPengguna();  
};
```

- Saat halaman dimuat, data buku dan buku favorit diambil dari Local Storage, dan nama pengguna ditampilkan dari `sessionStorage`.

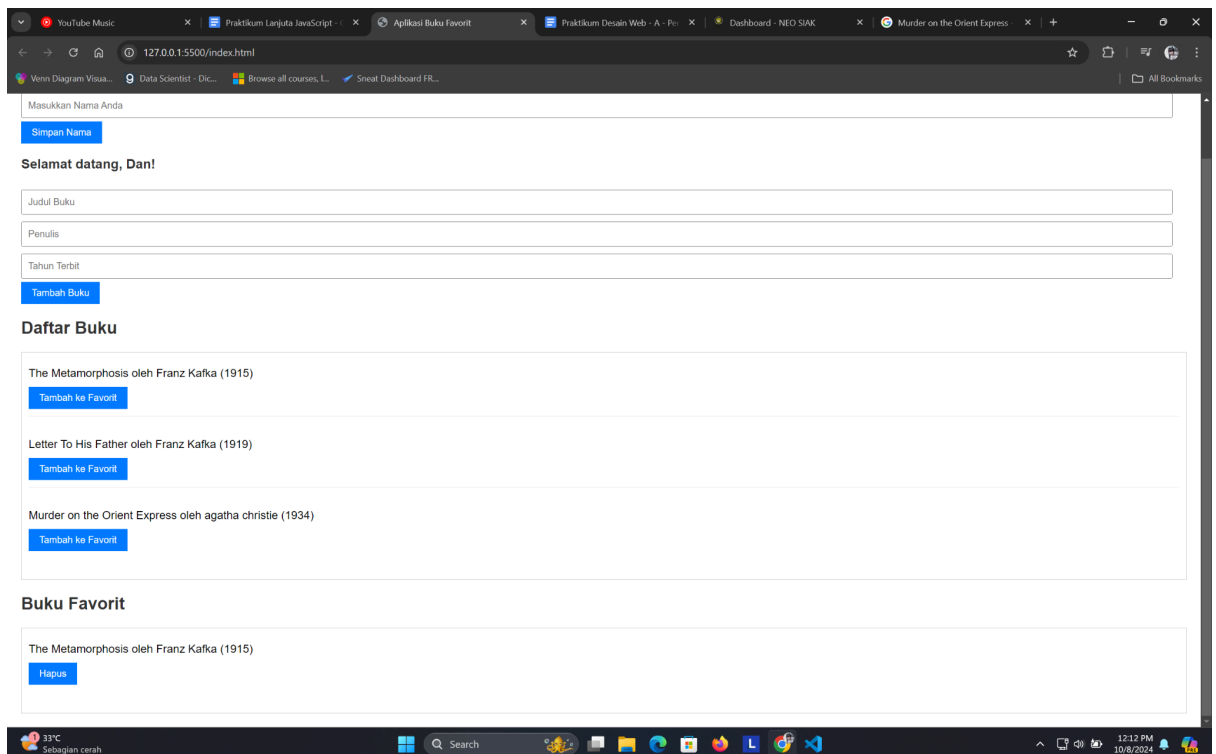


Laporan Tangkapan layar dari setiap ujicoba di atas:

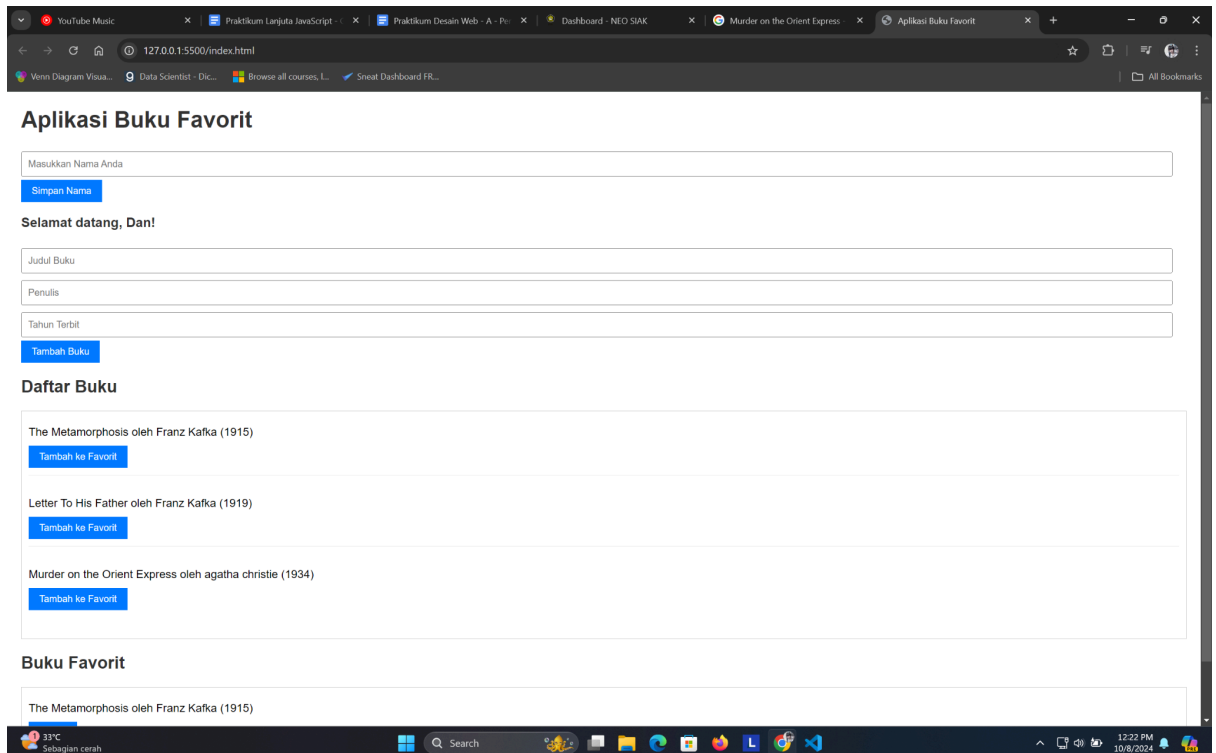
1. ) Buka index.html di browser



2. ) Tambahkan beberapa buku & Tandai beberapa buku sebagai favorit



### 3. Refresh halaman dan pastikan buku favorit tetap ada



### 4. Tutup tab atau browser, lalu buku kembali dan cek apakah nama pengguna masih ada

