LAPORAN PRAKTIKUM DESAIN WEB

MODUL 9 WEB STORAGE

Dalam rangka memenuhi tugas mata kuliah Praktikum Desain Website

Dosen pengampu: Vearen Dika Sofirudin, S.Pd., M.Ed.

Disusun oleh:

Nama : Muhammad Irfan Maulana

NIM : K3524059

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN

UNIVERSITAS SEBELAS MARET

SURAKARTA

2025

A. Tujuan Praktikum

Tujuan dari praktikum ini adalah berfokus pada pemahaman konsep, tujuan kedua pada implementasi teknis, dan tujuan ketiga pada aplikasi praktis untuk meningkatkan kualitas aplikasi web.

1. Memahami Konsep dan Implementasi Web Storage
2. Mengimplementasikan Fitur Persistent State dengan localStorage
3. Meningkatkan User Experience Melalui Data Persistence

B. Ringkasan Materi

Web Storage adalah salah satu fitur penting dalam pengembangan aplikasi web modern yang memungkinkan aplikasi web untuk menyimpan data secara lokal di dalam browser pengguna. Teknologi ini merupakan bagian dari spesifikasi HTML5 yang dikembangkan oleh World Wide Web Consortium (W3C) sebagai solusi penyimpanan data yang lebih baik dibandingkan dengan metode tradisional seperti cookies. Web Storage memberikan kemampuan kepada developer untuk menyimpan data dalam bentuk pasangan key-value (kunci-nilai) dengan kapasitas yang jauh lebih besar, yaitu sekitar 5-10 MB per domain, dibandingkan dengan cookies yang hanya memiliki kapasitas maksimal 4 KB. Dengan adanya Web Storage, aplikasi web dapat menyimpan informasi pengguna, preferensi, status aplikasi, dan data lainnya secara persisten tanpa perlu mengirimkan data tersebut ke server setiap kali melakukan request HTTP.

Web Storage dirancang dengan tujuan untuk meningkatkan performa aplikasi web dan memberikan pengalaman pengguna yang lebih baik. Berbeda dengan cookies yang secara otomatis dikirimkan ke server pada setiap HTTP request, data yang disimpan dalam Web Storage hanya dapat diakses melalui JavaScript di sisi klien dan tidak akan dikirimkan ke server secara otomatis. Hal ini membuat Web Storage menjadi pilihan yang lebih efisien untuk menyimpan data yang tidak perlu dikomunikasikan dengan server, seperti preferensi tema tampilan, pengaturan bahasa, status form yang belum selesai diisi, data cache, atau informasi session pengguna. Dengan demikian, bandwidth jaringan dapat dihemat dan loading time aplikasi web menjadi lebih cepat karena mengurangi jumlah data yang harus ditransfer antara client dan server.

Web Storage terbagi menjadi dua jenis utama yang memiliki karakteristik dan kegunaan yang berbeda, yaitu localStorage dan sessionStorage. localStorage adalah jenis penyimpanan yang bersifat permanen atau persistent, dimana data yang disimpan akan

tetap ada meskipun browser ditutup, komputer dimatikan, atau bahkan setelah berhari-hari kemudian. Data dalam localStorage hanya akan hilang jika secara eksplisit dihapus melalui kode JavaScript, dibersihkan oleh pengguna melalui pengaturan browser, atau ketika melakukan clear browsing data. Karakteristik persistent ini membuat localStorage sangat cocok digunakan untuk menyimpan preferensi pengguna yang bersifat jangka panjang, seperti pilihan tema (light mode atau dark mode), pengaturan bahasa, data login (dengan catatan bukan password), riwayat pencarian, atau data aplikasi yang perlu tersimpan untuk sesi-sesi berikutnya.

Di sisi lain, sessionStorage adalah jenis penyimpanan yang bersifat sementara atau temporary, dimana data hanya akan tersimpan selama tab atau window browser masih terbuka. Ketika pengguna menutup tab atau window tersebut, semua data yang tersimpan dalam sessionStorage akan otomatis terhapus. Setiap tab atau window memiliki sessionStorage yang independen, artinya data yang disimpan di satu tab tidak dapat diakses dari tab lain meskipun membuka URL yang sama. Hal ini membuat sessionStorage ideal untuk menyimpan data yang hanya relevan untuk satu sesi browsing tertentu, seperti data form multi-step yang sedang diisi, state aplikasi sementara, token autentikasi temporary, atau data wizard yang membutuhkan beberapa langkah untuk diselesaikan. Perbedaan fundamental antara localStorage dan sessionStorage terletak pada durasi dan scope penyimpanan data, sehingga developer harus memilih jenis storage yang tepat sesuai dengan kebutuhan aplikasi yang sedang dikembangkan.

Implementasi Web Storage dalam aplikasi web sangat mudah dan straightforward. Untuk menyimpan data, developer cukup memanggil localStorage.setItem('namaKey', 'nilaiData') atau sessionStorage.setItem('namaKey', 'nilaiData'). Sebagai contoh, jika ingin menyimpan preferensi dark mode pengguna, dapat menggunakan kode localStorage.setItem('darkMode', 'enabled'). Untuk mengambil data yang telah disimpan, gunakan method getItem() dengan sintaks localStorage.getItem('namaKey'), yang akan mengembalikan nilai data jika key ditemukan atau null jika key tidak ada. Ketika ingin menghapus data tertentu, gunakan removeItem() dengan sintaks localStorage.removeItem('namaKey'), sedangkan untuk menghapus semua data sekaligus dapat menggunakan localStorage.clear(). Semua operasi ini bersifat synchronous, artinya akan dieksekusi secara langsung dan memblokir eksekusi kode berikutnya hingga operasi selesai, sehingga perlu diperhatikan bahwa operasi yang terlalu banyak atau kompleks dapat mempengaruhi performa aplikasi.

Meskipun Web Storage sangat berguna dan mudah digunakan, terdapat beberapa aspek keamanan yang harus diperhatikan oleh developer. Web Storage tidak memiliki enkripsi built-in, sehingga semua data yang disimpan dapat dengan mudah diakses dan dibaca oleh siapa saja yang memiliki akses ke browser pengguna. Oleh karena itu, sangat tidak disarankan untuk menyimpan data sensitif seperti password, token autentikasi yang bersifat permanen, informasi kartu kredit, data pribadi yang confidential, atau informasi rahasia lainnya dalam Web Storage. Data dalam Web Storage juga rentan terhadap serangan Cross-Site Scripting (XSS), dimana jika aplikasi web memiliki celah keamanan XSS, attacker dapat menjalankan script berbahaya untuk mencuri atau memanipulasi data yang tersimpan dalam Web Storage. Untuk mengurangi risiko ini, developer harus selalu melakukan sanitasi input pengguna, menggunakan Content Security Policy (CSP), dan mengimplementasikan best practices keamanan web lainnya.

Aplikasi web progresif (Progressive Web Apps) sering memanfaatkan Web Storage sebagai bagian dari strategi offline-first, dimana data penting disimpan secara lokal untuk memungkinkan aplikasi tetap berfungsi meskipun tidak ada koneksi internet. Web Storage juga digunakan untuk implementasi caching sederhana, menyimpan hasil API call yang jarang berubah untuk mengurangi jumlah request ke server dan mempercepat loading time. Dalam konteks e-commerce, Web Storage dapat digunakan untuk menyimpan shopping cart sementara, riwayat produk yang dilihat, atau wishlist pengguna. Untuk aplikasi analytics, Web Storage berguna untuk menyimpan user ID, session tracking, atau data behaviour pengguna sebelum dikirim ke server analytics. Developer juga sering menggunakan Web Storage untuk menyimpan state aplikasi single-page application (SPA), token JWT untuk autentikasi (dengan pertimbangan keamanan yang matang), atau sebagai fallback storage ketika teknologi penyimpanan yang lebih advanced seperti IndexedDB tidak tersupport oleh browser pengguna. Dengan memahami karakteristik, kelebihan, dan keterbatasan Web Storage, developer dapat membuat keputusan yang tepat dalam merancang arsitektur penyimpanan data aplikasi web yang efisien, aman, dan user-friendly.

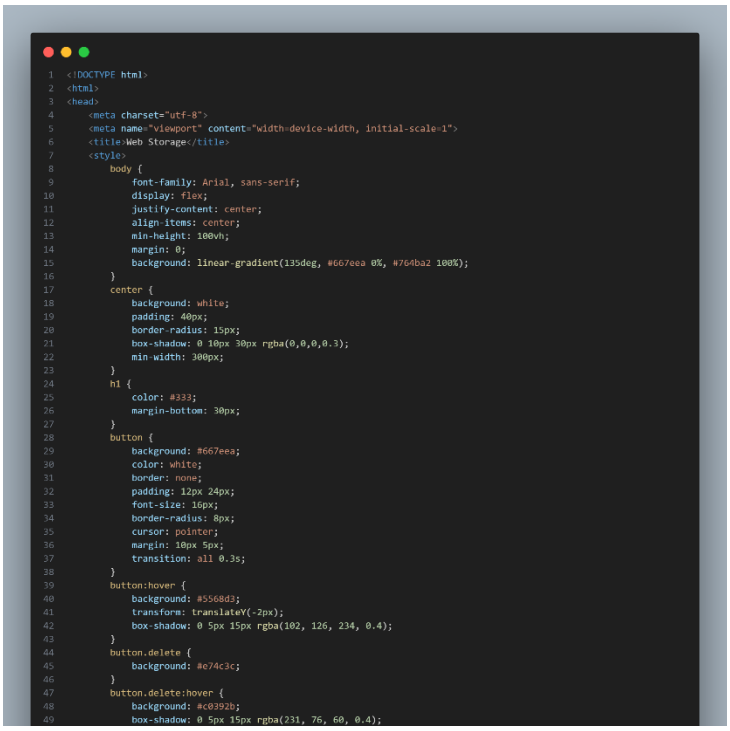
C. Latihan

Buatlah sebuah halaman web sederhana dengan memanfaatkan Web Storage yang berisi :

- Tambahkan sebuah tombol yang dapat menghapus nama dari member!
- Modifikasi javascript sehingga data nama member tetap akan ada selama belum dihapus!

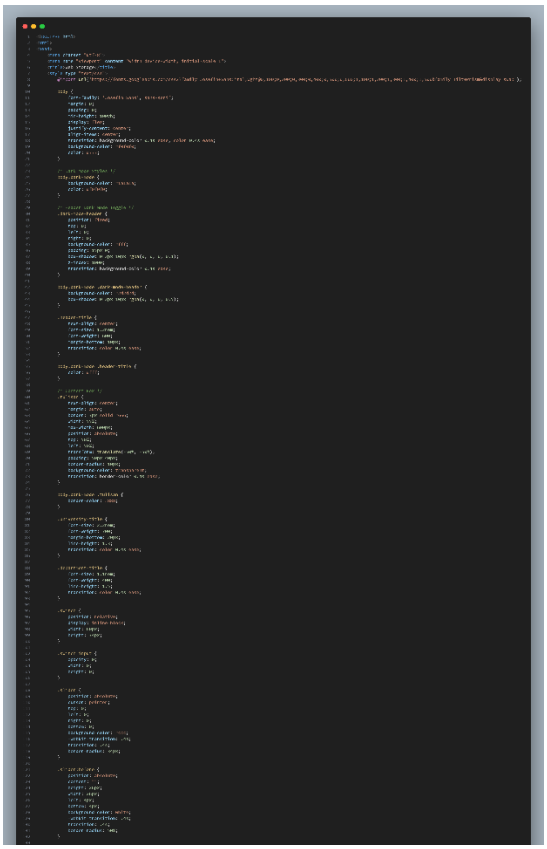
- Menambahkan css dan javascript yang tepat dengan memanfaatkan local storage sehingga fungsi dark mode dapat berjalan ketika toggle dark mode dijalankan

Kode HTML dan CSS :



A

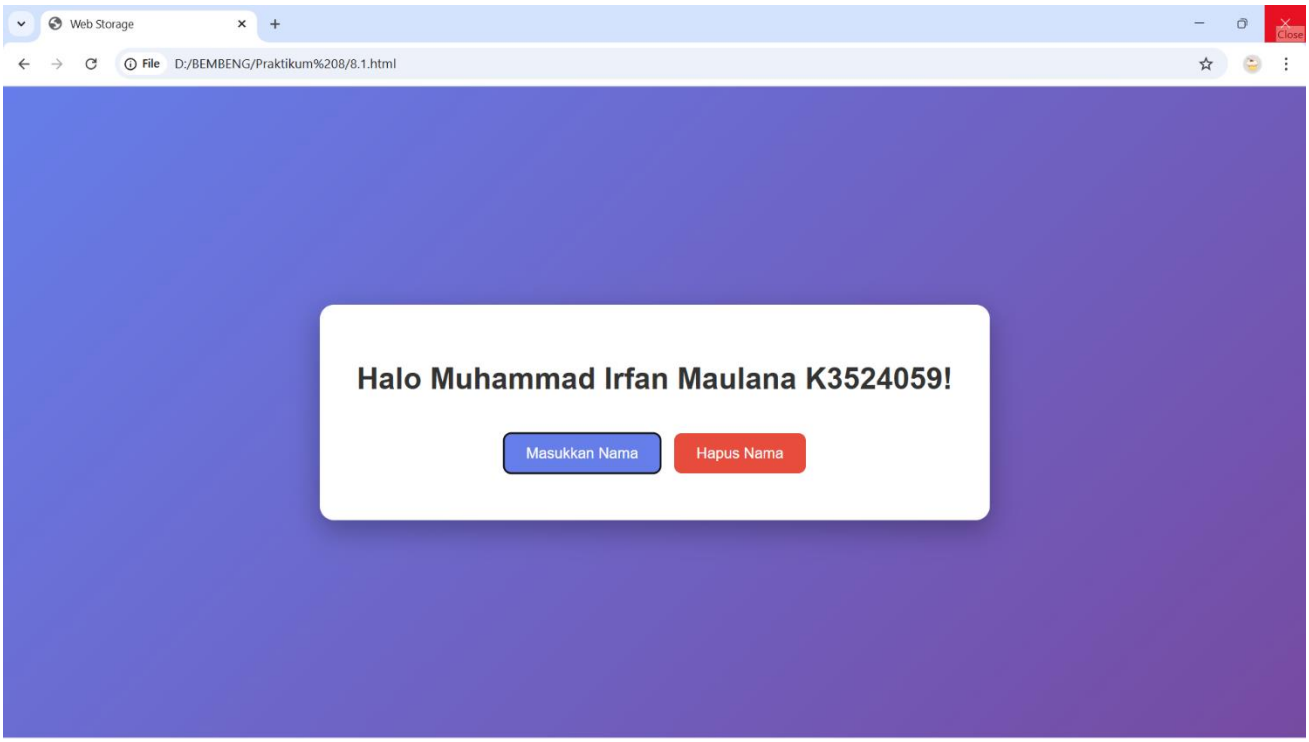
```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Web Storage</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            margin: 0;
            background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
        }
        center {
            background: white;
            padding: 40px;
            border-radius: 15px;
            box-shadow: 0 10px 30px rgba(0,0,0,0.3);
            min-width: 300px;
        }
        h1 {
            color: #333;
            margin-bottom: 30px;
        }
        button {
            background: #667eea;
            color: white;
            border: none;
            padding: 12px 24px;
            font-size: 16px;
            border-radius: 8px;
            cursor: pointer;
            margin: 10px 5px;
            transition: all 0.3s;
        }
        button:hover {
            background: #5568d3;
            transform: translateY(-2px);
            box-shadow: 0 5px 15px rgba(102, 126, 234, 0.4);
        }
        button.delete {
            background: #e74c3c;
        }
        button.delete:hover {
            background: #c0392b;
            box-shadow: 0 5px 15px rgba(231, 76, 60, 0.4);
        }
    </style>
</head>
<body>
    <center>
        <h1 id="namaMember">Selamat Datang, Member!</h1>
        <button onclick="gantiNama(event)">Masukkan Nama</button>
        <button class="delete" onclick="hapusNama(event)" id="btnHapus" style="display:none;">Hapus Nama</button>
    </center>

    <script type="text/javascript">
        const namaMember = document.querySelector('#namaMember');
        const btnHapus = document.querySelector('#btnHapus');

        // Mengecek apakah sudah ada nama tersimpan di localStorage
        const namaTersimpan = localStorage.getItem('nama');
        if (namaTersimpan) {
            namaMember.innerHTML = 'Halo ' + namaTersimpan + '!';
            btnHapus.style.display = 'inline-block'; // Tampilkan tombol hapus
        }

        function gantiNama(e) {
            e.preventDefault();

            let nama = prompt('Masukkan nama Anda:');
            if (nama && nama.trim() !== '') {
                // Simpan nama ke localStorage
                localStorage.setItem('nama', nama);
                namaMember.innerHTML = 'Halo ' + nama + '!';
                btnHapus.style.display = 'inline-block'; // Tampilkan tombol hapus
            } else {
                alert('Nama tidak boleh kosong!!');
            }
        }

        function hapusNama(e) {
            e.preventDefault();

            // Konfirmasi sebelum menghapus
            if (confirm('Apakah Anda yakin ingin menghapus nama?')) {
                // Hapus nama dari localStorage
                localStorage.removeItem('nama');
                namaMember.innerHTML = 'Selamat Datang, Member!';
                btnHapus.style.display = 'none'; // Sembunyikan tombol hapus
            }
        }
    </script>
</body>
</html>
```



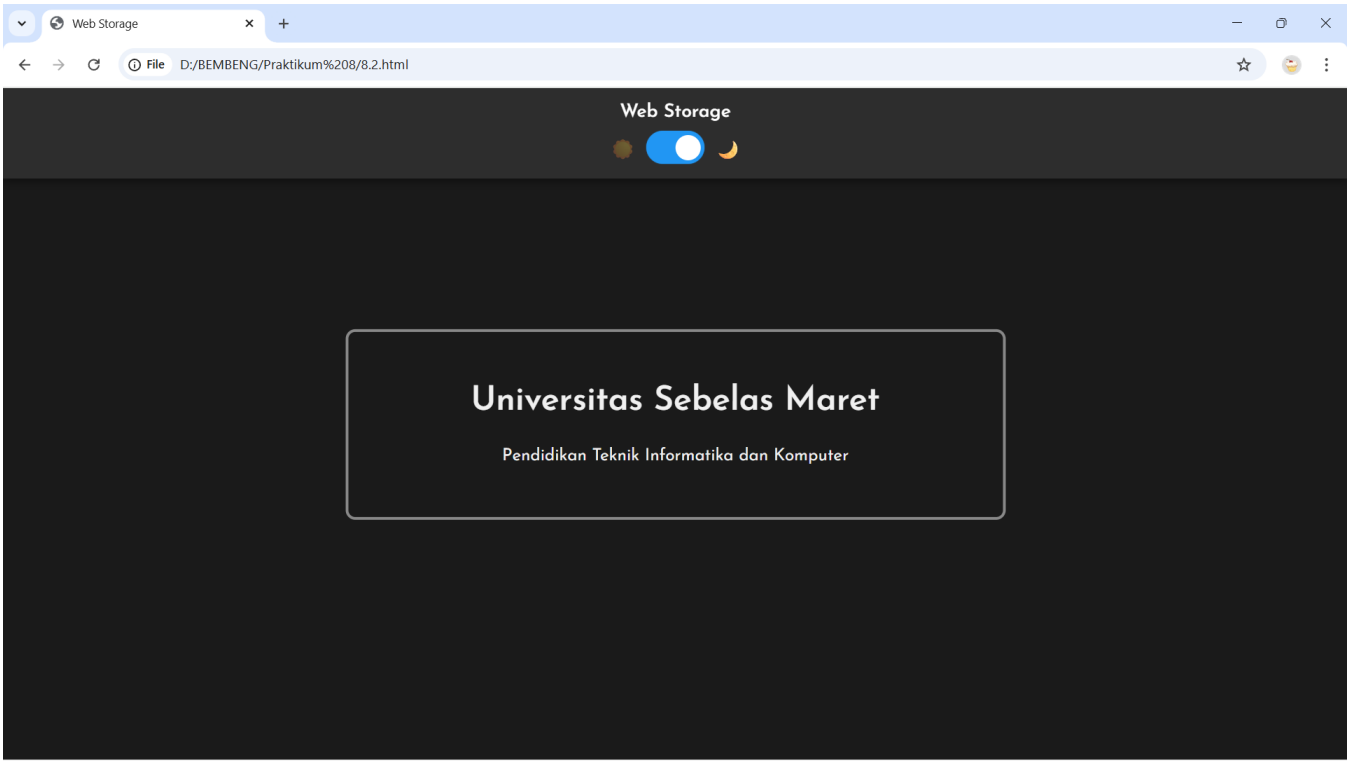
B

Hasil :

Hasil 1



Hasil 2



D. Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa Web Storage merupakan teknologi penyimpanan data lokal yang sangat powerful dan efisien dalam pengembangan aplikasi web modern. Web Storage menyediakan dua jenis penyimpanan yaitu localStorage untuk data yang bersifat permanen dan sessionStorage untuk data yang bersifat sementara, dengan kapasitas yang jauh lebih besar (5-10 MB) dibandingkan cookies tradisional (4 KB). Melalui praktikum ini, telah berhasil diimplementasikan fitur dark mode toggle yang memanfaatkan localStorage untuk menyimpan preferensi pengguna secara permanen. Aplikasi yang dibuat mampu mengingat pilihan mode tampilan pengguna, sehingga ketika membuka kembali aplikasi di kemudian hari, preferensi tersebut akan langsung diterapkan tanpa perlu mengatur ulang, membuktikan bahwa Web Storage sangat efektif untuk meningkatkan user experience dengan memberikan pengalaman yang lebih personal dan konsisten.

Namun demikian, penggunaan Web Storage juga harus memperhatikan aspek keamanan dan limitasi yang ada. Data yang disimpan dalam Web Storage tidak terenkripsi dan dapat diakses oleh siapa saja yang memiliki akses ke browser, sehingga tidak boleh digunakan untuk menyimpan data sensitif seperti password atau informasi kartu kredit. Selain itu, Web Storage memiliki keterbatasan kapasitas dan bersifat synchronous yang dapat mempengaruhi performa jika tidak digunakan dengan bijak. Dengan memahami kelebihan dan keterbatasan Web Storage, developer dapat membuat keputusan yang tepat dalam merancang sistem penyimpanan data yang aman, efisien, dan sesuai dengan kebutuhan aplikasi web yang dikembangkan.