

ALAPORAN DESAIN WEBSITE

Dosen Pengampu:

Vearen Dika Sofirudin, S.pd., M.ed.

Disusun Oleh:

Nama: Adinda Putri Utami

NIM: K3524002

Kelas: B

A. Judul

Penerapan Web Storage dan Dark Mode pada Desain Website

B. Tujuan Praktikum

Praktikum ini bertujuan agar kita dapat memahami penerapan *Web Storage* khususnya Local Storage pada website. Mengimplementasikan fitur penyimpanan data pengguna seperti nama atau preferensi tema. Mempelajari cara menerapkan Dark Mode menggunakan CSS dan JavaScript. Meningkatkan kemampuan dalam mengelola file HTML, CSS, dan JavaScript pada sebuah projek sederhana. Mengembangkan kemampuan dasar pengembangan website interaktif.

C. Alat dan Bahan

1. Laptop / Komputer
2. Vscode (text editor)
3. Web browser (chrome)
4. File html
5. Internet

D. Ringkasan Materi

Materi pada praktikum ini membahas beberapa konsep penting dalam pemrograman website:

1. HTML (HyperText Markup Language)
Digunakan untuk menyusun struktur halaman seperti heading, paragraf, tombol, dan elemen lainnya.
2. CSS (Cascading Style Sheets)
Digunakan untuk memberikan gaya (style) pada halaman website seperti warna, ukuran, posisi, dan tampilan dark mode.
3. JavaScript
Digunakan untuk memberikan interaksi pada website, seperti:

- Menyimpan data ke Local Storage
 - Mengubah tampilan halaman secara dinamis
 - Mengambil kembali data pengguna
4. Web Storage (Local Storage)

Merupakan fitur penyimpanan data pada browser yang memungkinkan website menyimpan informasi secara permanen (hingga dihapus manual).

Local Storage digunakan untuk:

- Menyimpan nama pengguna yang diinput
- Menyimpan status dark mode agar tetap aktif meski halaman direfresh

5. Implementasi Dark Mode

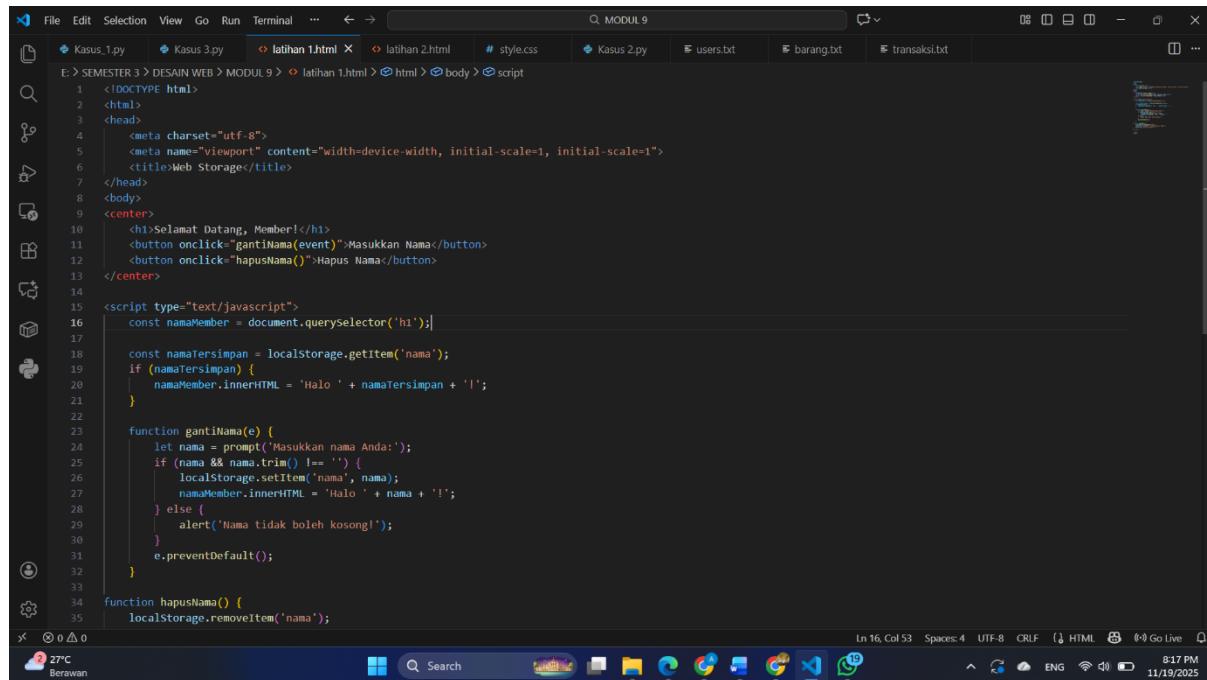
Dark Mode diimplementasikan menggunakan kombinasi:

- CSS untuk gaya warna terang/gelap
- JavaScript untuk toggle tema
- Local Storage untuk menyimpan pilihan tema pengguna

E. Program Python

1. Latihan 1

Input Program:



```

File Edit Selection View Go Run Terminal ... ↵ → O MODUL 9
Kasus_1.py Kasus_3.py latihan_1.html latihan_2.html # style.css Kasus_2.py users.txt barang.txt transaksi.txt

E: > SEMESTER 3 > DESAIN WEB > MODUL 9 > latihan_1.html > html > body > script
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, initial-scale=1">
6      <title>Web Storage</title>
7  </head>
8  <body>
9  <center>
10     <h1>Selamat Datang, Member!</h1>
11     <button onclick="gantiNama(event)">Masukkan Nama</button>
12     <button onclick="hapusNama()">Hapus Nama</button>
13  </center>
14
15  <script type="text/javascript">
16      const namaMember = document.querySelector('h1');
17
18      const namatersimpan = localStorage.getItem('nama');
19      if (namatersimpan) {
20          namaMember.innerHTML = 'Halo ' + namatersimpan + '!';
21      }
22
23      function gantiNama(e) {
24          let nama = prompt('Masukkan nama Anda:');
25          if (nama && nama.trim() != '') {
26              localStorage.setItem('nama', nama);
27              namaMember.innerHTML = 'Halo ' + nama + '!';
28          } else {
29              alert('Nama tidak boleh kosong!');
30          }
31          e.preventDefault();
32      }
33
34      function hapusNama() {
35          localStorage.removeItem('nama');

```

Ln 16, Col 53 Spaces: 4 UTF-8 CRLF ⌂ HTML ⌂ Go Live ⌂

27°C Berawan 8:17 PM 11/19/2025

```
31 |     e.preventDefault();
32 |
33 |
34 function hapusNama() {
35     localStorage.removeItem('nama');
36     namaMember.innerHTML = "Selamat Datang, Member!";
37     alert("Nama berhasil dihapus!");
38 }
39 </script>
40
41 </body>
42 </html>
```

Screenshot copied to clipboard
Automatically saved to screenshots folder.

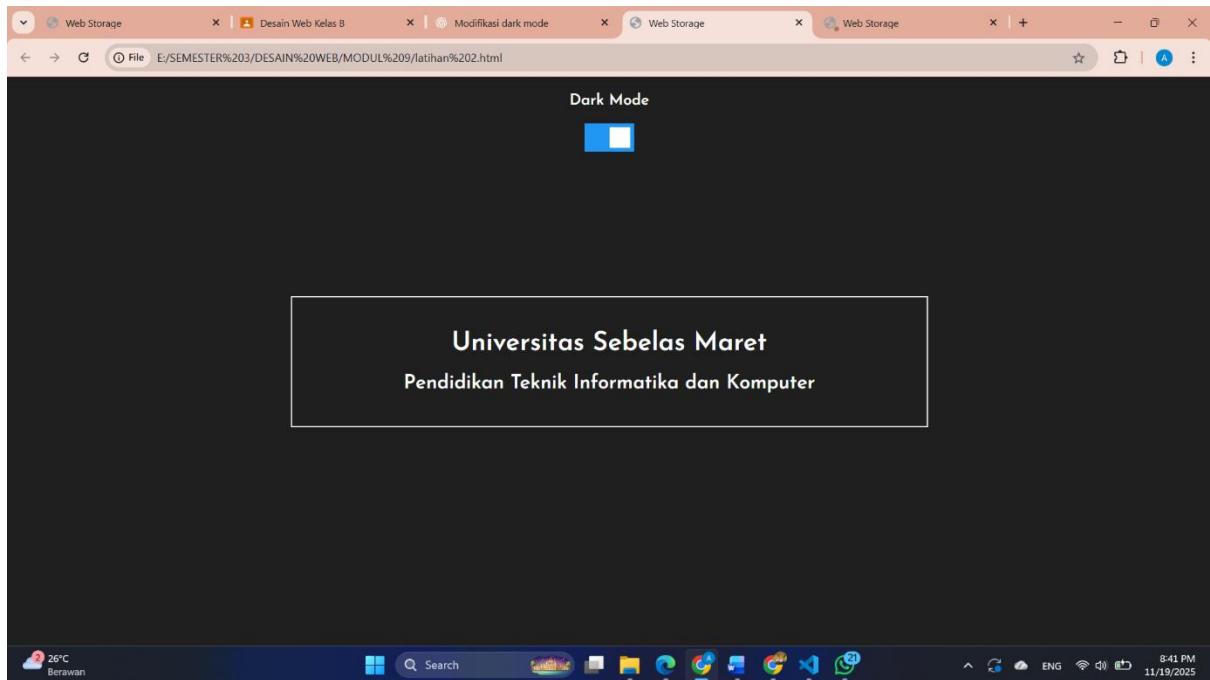
Snipping Tool

Markup and share

In 16, Col 53 Spaces: 4 UTF-8 CRLF Line HTML If I Go Live 8:17 PM ENG 11/19/2025

Output Program:





2. Latihan 2

Input Program:

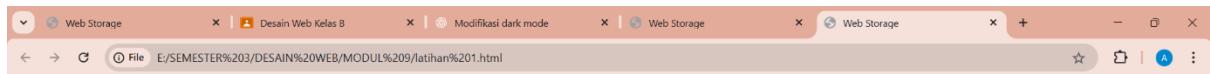
A screenshot of a code editor window titled 'MODUL 9'. The editor displays two files: 'latihan 2.html' (HTML) and '# style.css' (CSS). The HTML file contains the structure of a dark mode website, including meta tags for charset and viewport, a title, and a style link to the CSS file. The CSS file defines styles for the body (font-family: 'Josefin Sans', sans-serif; background-color: black; color: white; transition: all 0.3s ease;) and a dark body class (background-color: #1f1f1f; color: white;). It also includes styles for .tulisan (border-color: white;) and .label (display: block; text-align: center; margin-top: 20px;). The code editor interface includes a sidebar with file navigation, a status bar at the bottom, and a taskbar at the very bottom.

```

36     width: 50%;  
37     position: absolute;  
38     top: 50%;  
39     left: 50%;  
40     transform: translate(-50%, -50%);  
41     padding: 20px;  
42   }  
43  
44   .switch {  
45     position: relative;  
46     display: inline-block;  
47     width: 60px;  
48     height: 34px;  
49   }  
50  
51   .switch input {  
52     opacity: 0;  
53     width: 0;  
54     height: 0;  
55   }  
56  
57   .slider {  
58     position: absolute;  
59     cursor: pointer;  
60     top: 0;  
61     left: 0;  
62     right: 0;  
63     bottom: 0;  
64     background-color: #ccc;  
65     transition: .4s;  
66   }  
67  
68   .slider:before {  
69     position: absolute;  
70     content: "";  
71     height: 26px;  
72     width: 26px;  
73     left: 4px;  
74     bottom: 4px;  
75     background-color: white;  
76     transition: .4s;  
77   }  
78  
79   input:checked + .slider {  
80     background-color: #2196F3;  
81   }  
82  
83   input:checked + .slider:before {  
84     transform: translateX(26px);  
85   }  
86 </style>  
87 </head>  
88 <body>  
89   <div class="tulisan">  
90     <h1>Universitas Sebelas Maret</h1>  
91     <h2>Pendidikan Teknik Informatika dan Komputer</h2>  
92   </div>  
93  
94   <div class="label">  
95     <h3>Dark Mode</h3>  
96     <label class="switch">  
97       <input type="checkbox" id="toggle">  
98       <span class="slider"></span>  
99     </label>  
100   </div>  
101  
102 <script>  
103   const toggle = document.getElementById("toggle");  
104  
105   if (localStorage.getItem("tema") === "dark") {  
106     document.body.classList.add("dark");  
107     toggle.checked = true;  
108   }  
109  
110   toggle.addEventListener("change", function() {  
111     if (this.checked) {  
112       document.body.classList.add("dark");  
113       localStorage.setItem("tema", "dark");  
114     } else {  
115       document.body.classList.remove("dark");  
116       localStorage.setItem("tema", "light");  
117     }  
118   });  
119 </script>  
120 </body>  
121 </html>

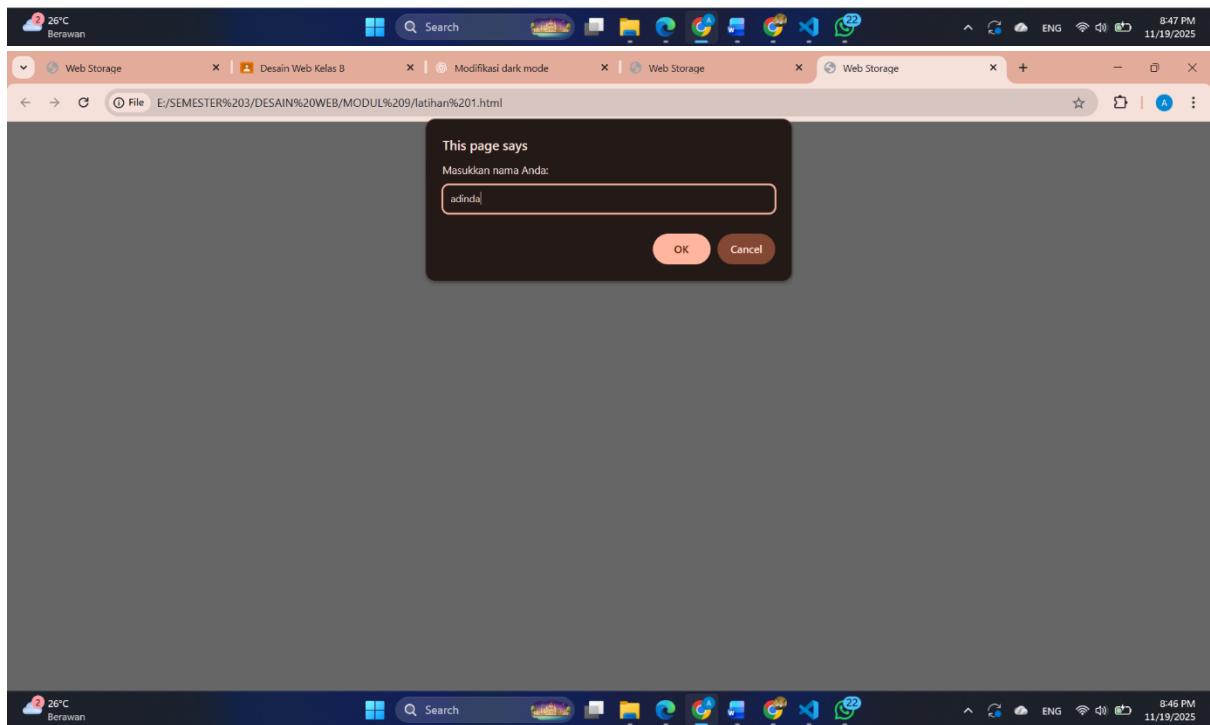
```

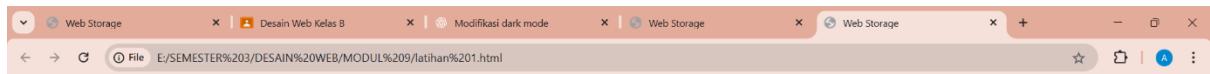
Output Program:



Selamat Datang, Member!

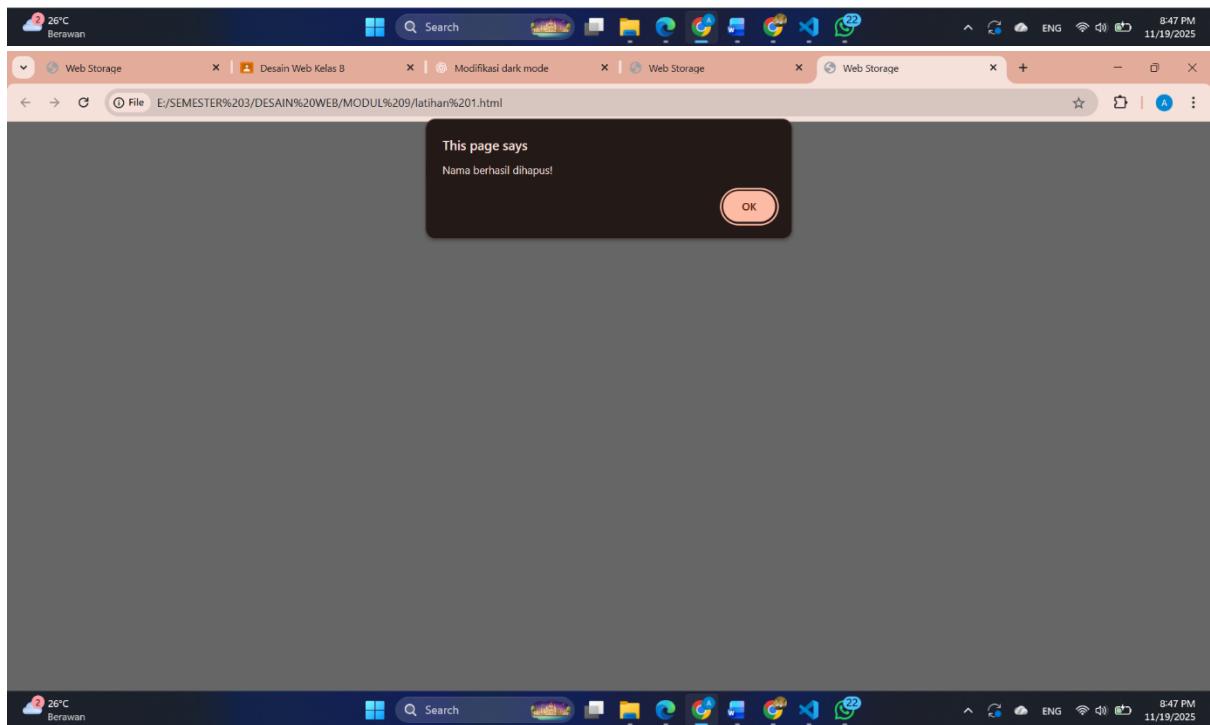
[Masukkan Nama] [Hapus Nama]

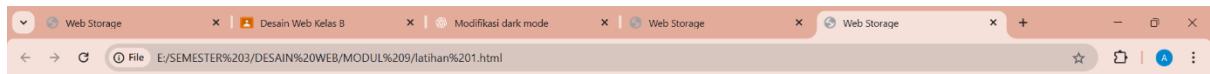




Halo adinda!

[Masukkan Nama] [Hapus Nama]





Selamat Datang, Member!

[Masukkan Nama] [Hapus Nama]



F. Langkah

A. Praktikum Latihan1

1. Buka Visual Studio Code. Kemudian membuat file baru bernama latihan 1.html
2. Ketik struktur dasar html yang diawali dengan berikut dan tambahkan judul halaman dalam tak <title>. Simpan dan lanjutkan ke selanjutnya.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, initial-scale=1">
    <title>Web Storage</title>
</head>
<body>
```

3. Bagian tampilan (interface) didalam <body>Buat elemen <h1> berisi teks “Selamat Datang, Member!”. Tambahkan tombol Masukkan Nama dengan atribut dan tambahkan tombol Hapus Nama dengan atribut:

```
<center>
    <h1>Selamat Datang, Member!</h1>
    <button onclick="gantiNama(event)">Masukkan Nama</button>
    <button onclick="hapusNama()">Hapus Nama</button>
</center>
```

4. Tambahkan tag <script> dibawah elemen html dan buat variabel namaMember untuk mengambil elemen <h1>. Variabel ini digunakan untuk mengubah teks salam sesuai nama yang dimasukkan.

```
<script type="text/javascript">
    const namaMember = document.querySelector('h1');
```

5. Mengecek data local storage, gunakan perintah localStorage.getItem('nama') untuk melihat apakah ada data nama yang tersimpan. Jika ada data, tampilkan teks “Halo [nama]!” di <h1>.

```
const namaterimpan = localStorage.getItem('nama');
if (namaterimpan) {
    namaMember.innerHTML = 'Halo ' + namaterimpan + '!';
}
```

6. Fungsi gantiNama(), Ketika tombol Masukkan Nama diklik, browser menampilkan prompt. Pengguna memasukkan nama pada prompt. Program

mengecek apakah nama tidak kosong menggunakan. Jika valid: Simpan nama ke Local Storage menggunakan localStorage.setItem('nama', nama), Ubah teks <h1> menjadi “Halo [nama]!”. Jika nama kosong, tampilkan alert “Nama tidak boleh kosong!”. Gunakan e.preventDefault() untuk mencegah reload.

```

22
23
24
25
26
27
28
29
30
31
32
33
    function gantiNama(e) {
        let nama = prompt('Masukkan nama Anda:');
        if (nama && nama.trim() !== '') {
            localStorage.setItem('nama', nama);
            namaMember.innerHTML = 'Halo ' + nama + '!';
        } else {
            alert('Nama tidak boleh kosong!');
        }
        e.preventDefault();
    }

```

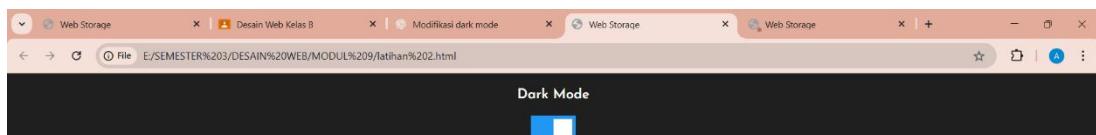
- Fungsi hapusNama(), Hapus data nama dari Local Storage dengan kode dibawah ini. Kemudian kembalikan teks <h1> menjadi “Selamat Datang, Member!”. Tampilkan alert bertuliskan “Nama berhasil dihapus!”.

```

33
34     function hapusNama() {
35         localStorage.removeItem('nama');
36         namaMember.innerHTML = "Selamat Datang, Member!";
37         alert("Nama berhasil dihapus!");
38     }
39
40
41
42

```

- Jalankan program dan lihat hasilnya.



B. Praktikum Latihan2

1. Membuat file HTML dengan struktur dasar: <!DOCTYPE html>, <html>, <head>, dan <body>. Menambahkan elemen <meta> untuk pengaturan karakter dan responsive layout. Memberikan judul halaman menggunakan <title>Web Storage</title>.



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Web Storage</title>
```

2. Pengaturan tampilan dasar, Pada bagian <style>, Mengimpor font "Josefin Sans" dari Google Fonts. Mengatur tampilan default website background putih, teks hitam, animasi transisi warna. Menata elemen .tulisan agar berada di tengah halaman. Menata tampilan komponen switch (tombol toggle).



```
<style type="text/css">
@import url('https://fonts.googleapis.com/css2?family=Josefin+Sans:wght@300;400;500;600&display=swap');
```

```
body {
    font-family: 'Josefin sans', sans-serif;
    background-color: white;
    color: black;
    transition: all 0.3s ease;
}

.tulisan {
    width: 50%;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    padding: 20px;
}
```

3. Kemudian membuat dark mode, body.dark Mengubah background menjadi hitam (#1f1f1f) dan teks menjadi putih. body.dark .tulisan Mengubah border agar sesuai dengan mode gelap. Pengaturan warna slider dalam kondisi checked: Slider berubah menjadi biru ketika diaktifkan. Dan tombol putih pindah ke kanan sebagai penanda aktif.



```
body.dark {
    background-color: #1f1f1f;
    color: white;
}

body.dark .tulisan {
    border-color: white;
}

.label {
    display: block;
    text-align: center;
    margin-top: 20px;
}
```

```

36     width: 50%;  
37     position: absolute;  
38     top: 50%;  
39     left: 50%;  
40     transform: translate(-50%, -50%);  
41     padding: 20px;  
42   }  
43  
44   .switch {  
45     position: relative;  
46     display: inline-block;  
47     width: 60px;  
48     height: 34px;  
49   }  
50  
51   .switch input {  
52     opacity: 0;  
53     width: 0;  
54     height: 0;  
55   }  
56  
57   .slider {  
58     position: absolute;  
59     cursor: pointer;  
60     top: 0;  
61     left: 0;  
62     right: 0;  
63     bottom: 0;  
64     background-color: #ccc;  
65     transition: .4s;  
66   }  
67  
68     .slider:before {  
69       position: absolute;  
70       content: "";  
71       height: 26px;  
72       width: 26px;  
73       left: 4px;  
74       bottom: 4px;  
75       background-color: white;  
76       transition: .4s;  
77     }  
78  
79     input:checked + .slider {  
80       background-color: #2196F3;  
81     }  
82  
83     input:checked + .slider:before {  
84       transform: translateX(26px);  
85     }  
86   
```

4. Konten utama pada website, Pada bagian <body> terdapat elemen:

a. **Div .tulisan**, berisi:

<h1>Universitas Sebelas Maret</h1>

<h2>Pendidikan Teknik Informatika dan Komputer</h2>

Bagian ini adalah konten utama yang ditampilkan di tengah halaman.

b. **Div .label** berisi:

Judul kecil “Dark Mode”

Elemen switch untuk mengubah tema

Tombol ini akan digunakan user untuk mengaktifkan dark mode.

```

88 <body>  
89  
90   <div class="tulisan">  
91     <h1>Universitas Sebelas Maret</h1>  
92     <h2>Pendidikan Teknik Informatika dan Komputer</h2>  
93   </div>  
94  
95   <div class="label">  
96     <h3>Dark Mode</h3>  
97     <label class="switch">  
98       <input type="checkbox" id="toggle">  
99       <span class="slider"></span>  
100 
```

5. Bagian JavaScript: Mengambil Elemen Toggle

Pada awal script: const toggle = document.getElementById("toggle"); Bagian ini berfungsi untuk mengambil elemen checkbox (switch) agar bisa digunakan dalam logika JavaScript.

```

102 <script>  
103   const toggle = document.getElementById("toggle");  
104  
105 
```

6. Mengecek tema di local storage, yaitu mengecek apakah sebelumnya user memilih dark mode.

Jika iya: Tambahkan class dark pada body.

Jadikan toggle dalam posisi *checked*. Hal ini membuat dark mode tetap aktif meskipun halaman direfresh.

```
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
```



7. Event listener toggle, Ketika toggle diklik, fungsi change dijalankan.

Jika toggle aktif (checked):

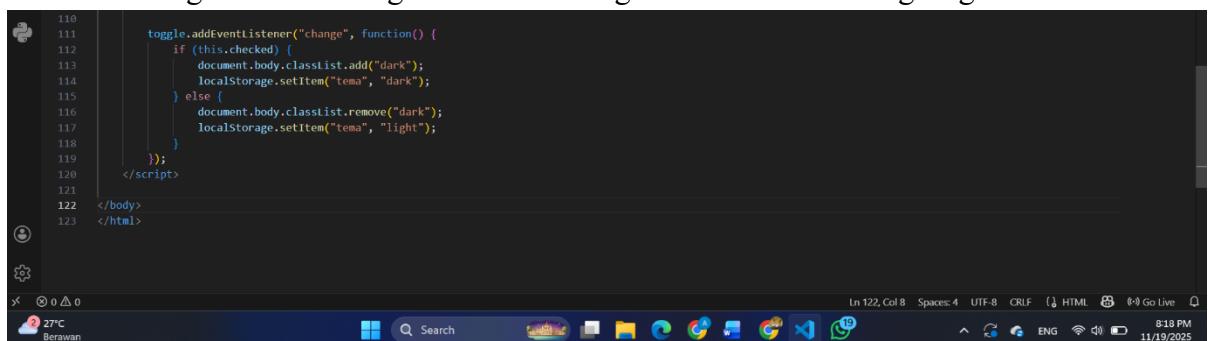
Tambahkan class .dark, Simpan preferensi tema = “dark”

Jika toggle dimatikan:

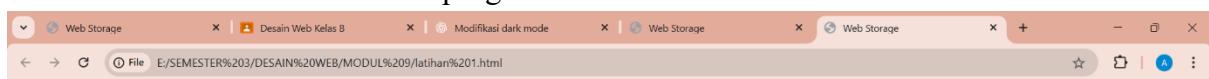
Hapus class .dark, Simpan preferensi tema = “light”

Bagian ini memungkinkan user mengubah tema secara langsung.

```
110
111
112
113
114
115
116
117
118
119
120
121
122
123
```

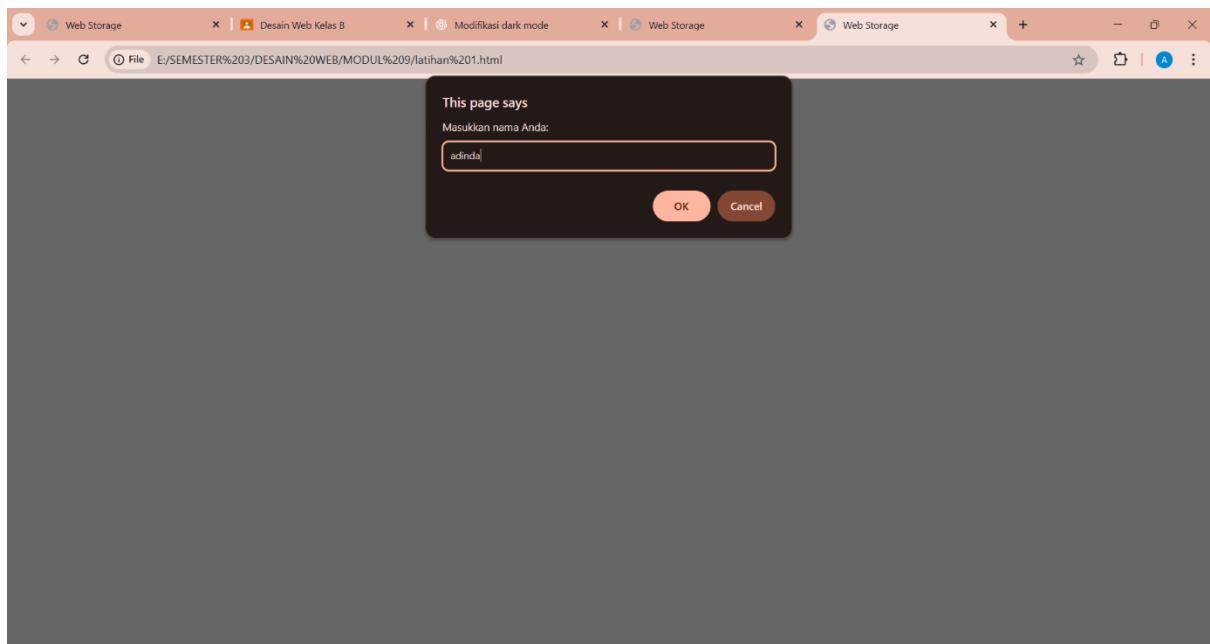


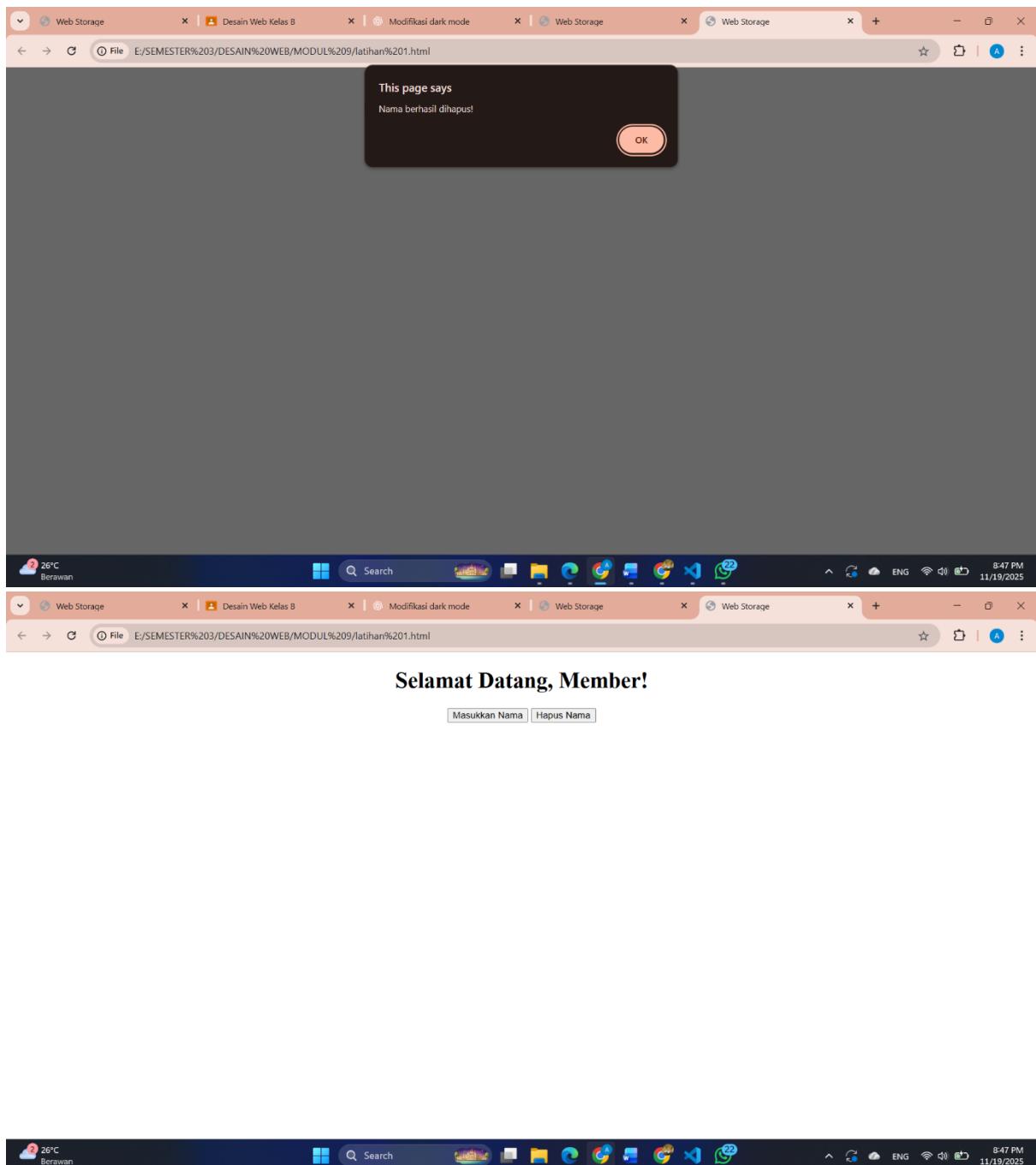
8. Jalankan dan liat hasil program.



Selamat Datang, Member!







G. Kesimpulan

Berdasarkan praktikum yang sudah dilakukan dapat disimpulkan bahwa Local Storage dapat digunakan untuk menyimpan data pengguna secara permanen tanpa harus menggunakan database. JavaScript berperan penting untuk mengolah data, mengubah tampilan, serta membuat website lebih interaktif. Penerapan fitur sederhana seperti input nama atau dark mode dapat meningkatkan pemahaman mengenai interaksi front-end. Praktikum ini membantu kita untuk mempelajari dan memahami dasar pengembangan website menggunakan HTML, CSS, dan JavaScript.