

LATIHAN MODUL 6

JAVASCRIPT

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



disusun oleh :
Natasya Luthfia Ramadhani
(K3524063)

Prodi Pendidikan Teknik Informatika dan Komputer
Fakultas Keguruan dan Ilmu Pendidikan
Universitas Sebelas Maret
2025

A. JUDUL LAPORAN

“JAVASCRIPT DAN DOM MANIPULATION”

B. TUJUAN PRAKTIKUM

Berdasarkan Modul 6, tujuan praktikum ini adalah :

1. Mahasiswa memahami struktur penulisan (sintaks) dasar JavaScript.
2. Mahasiswa mampu mengimplementasikan fungsi dasar JavaScript (Input/Output) dan operator aritmatika.
3. Mahasiswa mampu mengimplementasikan konsep Document Object Model (DOM) Manipulation untuk membuat interaksi dinamis pada halaman web.

C. RINGKASAN MATERI

Praktikum ini menggunakan JavaScript (JS) sebagai bahasa *client-side scripting* untuk membuat halaman web menjadi interaktif. Konsep utama yang diterapkan adalah DOM Manipulation dan Event Handling.

- DOM Manipulation: JS digunakan untuk mengakses, mengubah, dan memperbarui elemen-elemen HTML (misalnya mengubah teks hasil kalkulator atau mengubah gaya/style CSS pada paragraf).
- Event Handling: Menggunakan *event* seperti onclick pada tombol untuk memicu fungsi JS, yang menjadi dasar interaksi pengguna.
- State Management: Di Latihan 1, variabel digunakan untuk melacak status perhitungan; di Latihan 2, digunakan untuk melacak status *toggle (on/off)* dari efek *highlight*.

D. LANGKAH KERJA

I. LATIHAN 1: KALKULATOR INTERAKTIF DENGAN TOMBOL

Langkah kerja diuraikan berdasarkan implementasi kode HTML (Struktur dan Desain) dan JavaScript (Logika).

Langkah Kerja:

1. Membuat file baru bernama latihan1.html.
2. Menulis struktur dasar HTML dan tag `<script>`.
3. Menerapkan fungsi `prompt()` untuk meminta dua input bilangan.
4. Menggunakan fungsi `parseFloat()` untuk mengonversi input (yang awalnya berupa String) menjadi bilangan desimal agar operasi aritmatika berjalan benar.
5. Melakukan empat operasi aritmatika.

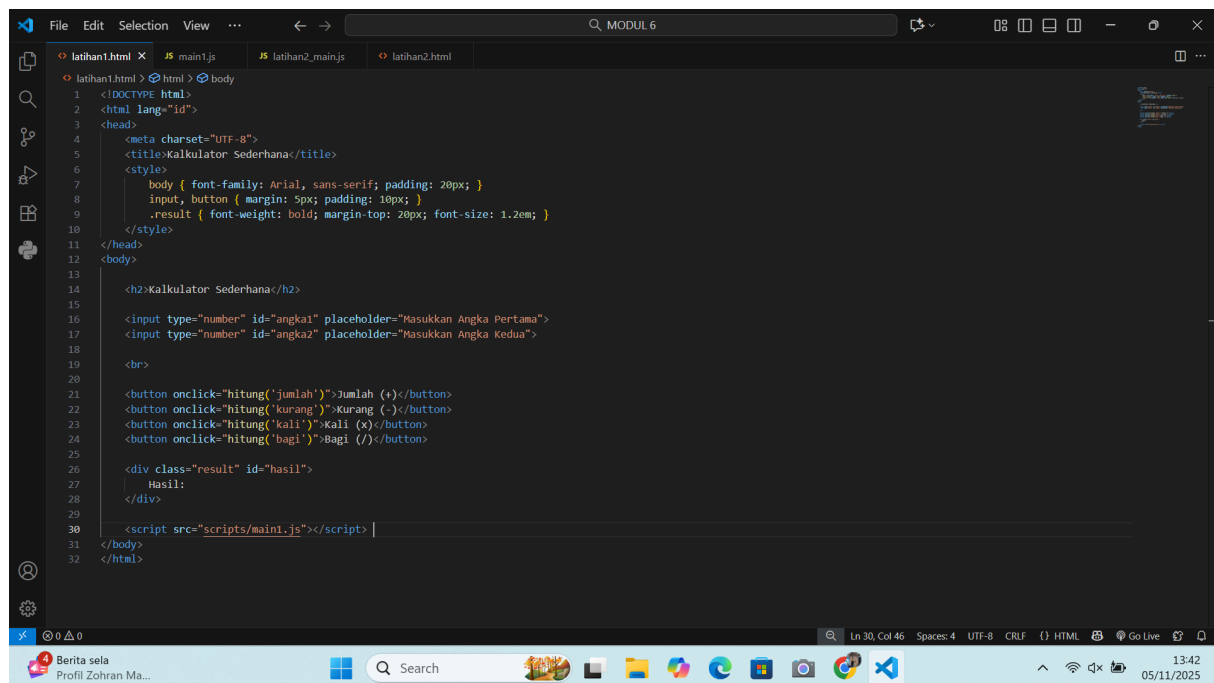
- Menampilkan hasil menggunakan `alert()` dan `document.getElementById().innerHTML` untuk output di halaman web.

Pembuatan Struktur HTML dan Desain Antarmuka Tujuan Langkah :

Membuat *layout* kalkulator 4x4, mendefinisikan variabel warna CSS (Cream-Cokelat Tua yang gelap), dan mengatur tautan fungsi JS pada setiap tombol.

1. Kode Program (latihan1.html)

Menunjukkan seluruh struktur HTML, termasuk pengaturan `<style>` yang mendefinisikan palet warna gelap pada `:root`, dan *grid* tombol 4x4.



```
1 <!DOCTYPE html>
2 <html lang="id">
3 <head>
4   <meta charset="UTF-8">
5   <title>Kalkulator Sederhana</title>
6   <style>
7     body { font-family: Arial, sans-serif; padding: 20px; }
8     input, button { margin: 5px; padding: 10px; }
9     .result { font-weight: bold; margin-top: 20px; font-size: 1.2em; }
10  </style>
11 </head>
12 <body>
13   <h2>Kalkulator Sederhana</h2>
14
15   <input type="number" id="angka1" placeholder="Masukkan Angka Pertama">
16   <input type="number" id="angka2" placeholder="Masukkan Angka Kedua">
17
18   <br>
19
20   <button onclick="hitung('jumlah')">Jumlah (+)</button>
21   <button onclick="hitung('kurang')">Kurang (-)</button>
22   <button onclick="hitung('kali')">Kali (x)</button>
23   <button onclick="hitung('bagi')">Bagi (/)</button>
24
25   <div class="result" id="hasil">
26     Hasil:
27   </div>
28
29   <script src="scripts/main1.js"></script>
30
31 </body>
32 </html>
```

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Kalkulator Sederhana</title>
  <style>
    body { font-family: Arial, sans-serif; padding: 20px; }
    input, button { margin: 5px; padding: 10px; }
    .result { font-weight: bold; margin-top: 20px; font-size: 1.2em; }
  </style>
</head>
<body>

  <h2>Kalkulator Sederhana</h2>

  <input type="number" id="angka1" placeholder="Masukkan Angka Pertama">
  <input type="number" id="angka2" placeholder="Masukkan Angka Kedua">

  <br>

  <button onclick="hitung('jumlah')">Jumlah (+)</button>
  <button onclick="hitung('kurang')">Kurang (-)</button>
```

```

<button onclick="hitung('kali')">Kali (x)</button>
<button onclick="hitung('bagi')">Bagi (/)</button>

<div class="result" id="hasil">
  Hasil:
</div>

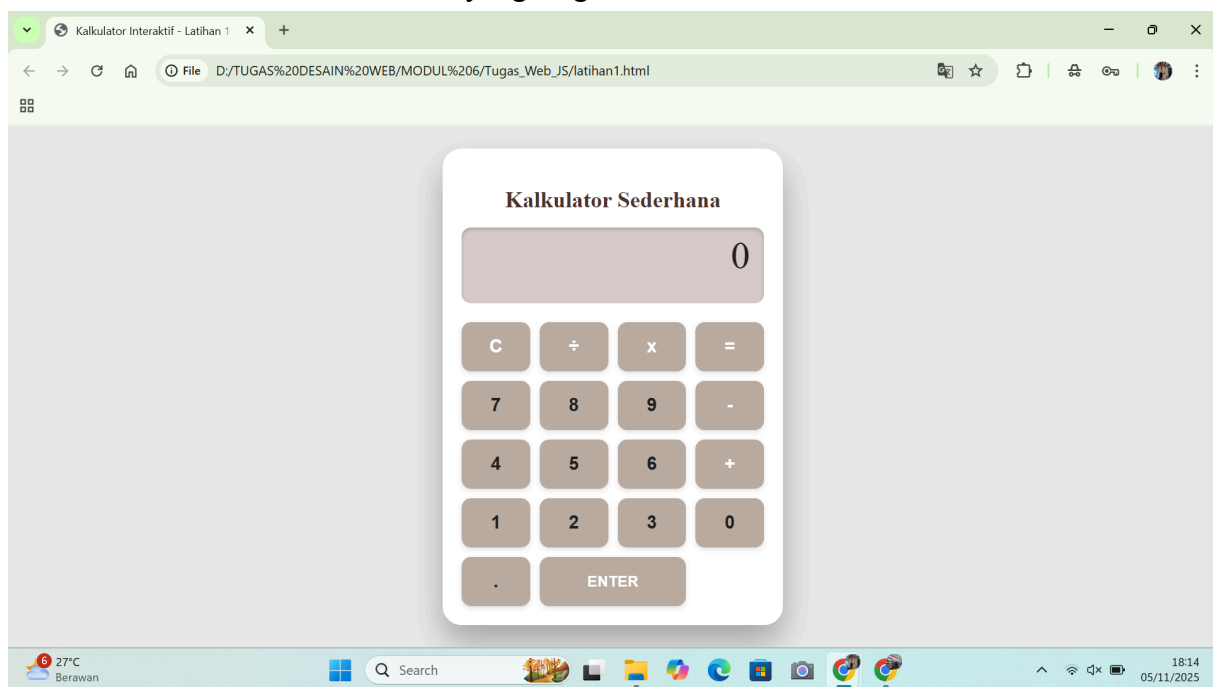
<script src="scripts/main1.js"></script>
</body>
</html>

```

Penjelasan: Kode ini menggunakan `prompt()` untuk mendapatkan `bil1` dan `bil2`. Setelah dikonversi menggunakan `parseFloat()`, hasil perhitungan (jumlah, kurang, kali, bagi) disimpan dalam variabel. Hasil kemudian ditampilkan melalui `alert()` dan dimasukkan ke dalam elemen `div` ber-ID `hasil-output` di badan dokumen HTML.

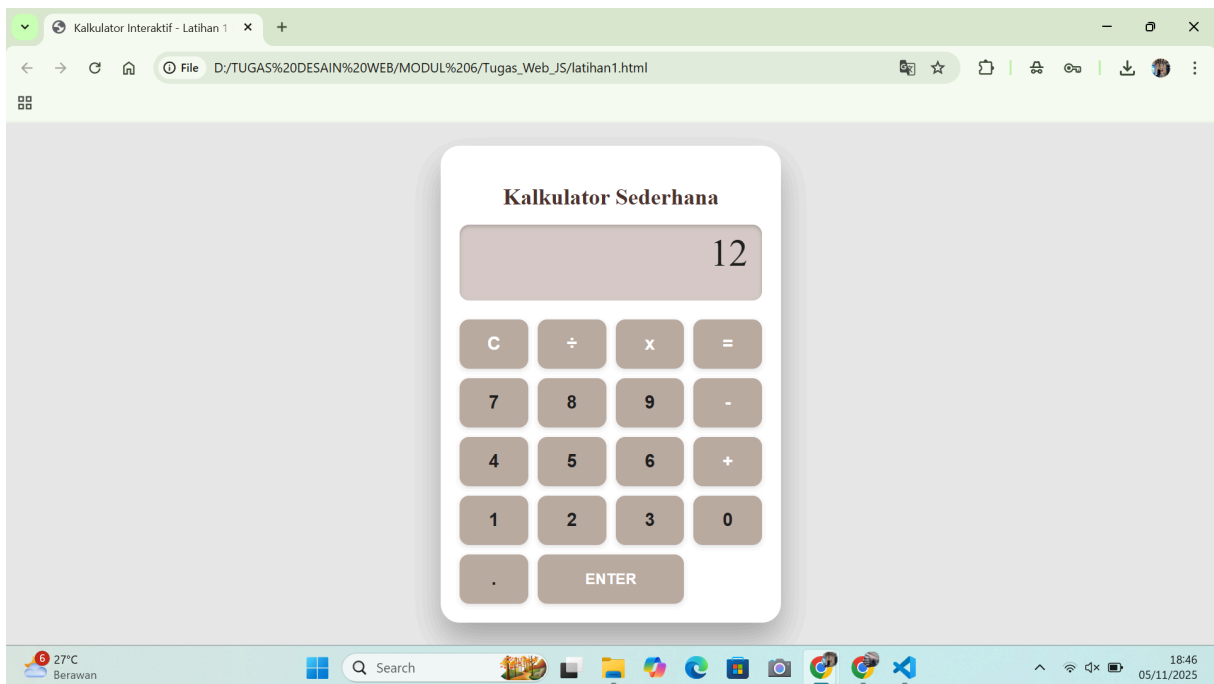
2. Tampilan Hasil Akhir Latihan 1

Menampilkan hasil kalkulator yang sudah berjalan di *browser* (preview), menunjukkan desain akhir Cream-Cokelat Tua yang elegan.

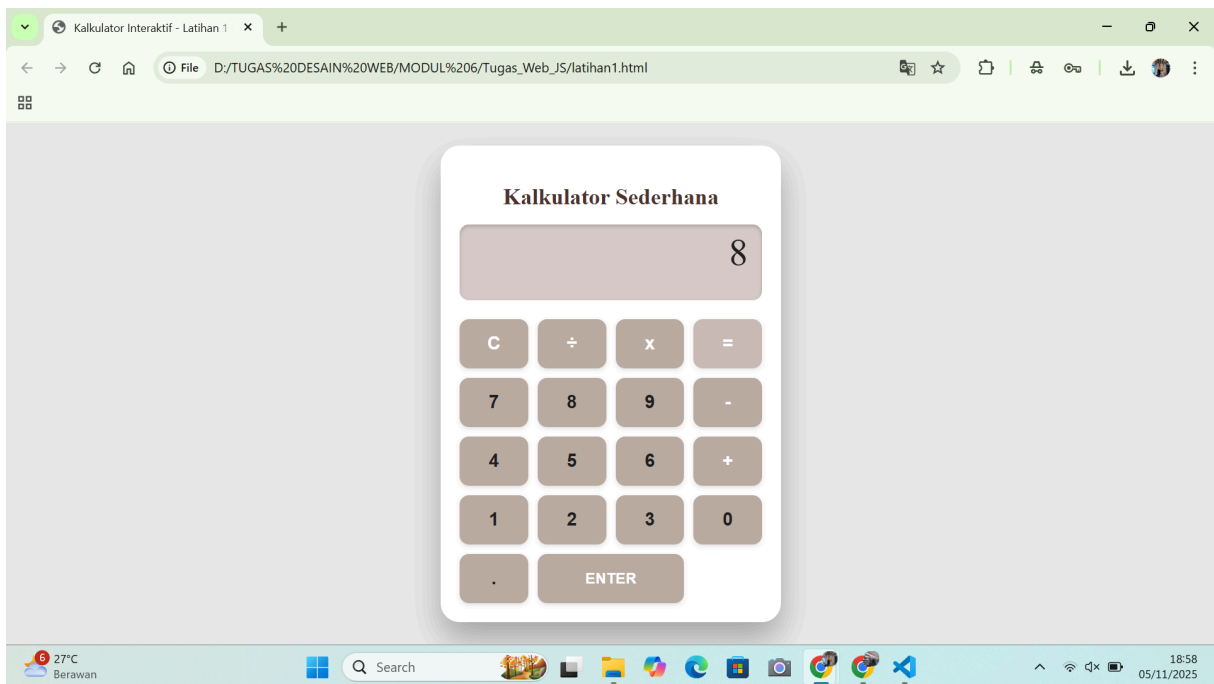


Penjelasan: Fungsi `prompt()` dieksekusi pertama kali, meminta pengguna memasukkan nilai, yang akan disimpan sebagai variabel.

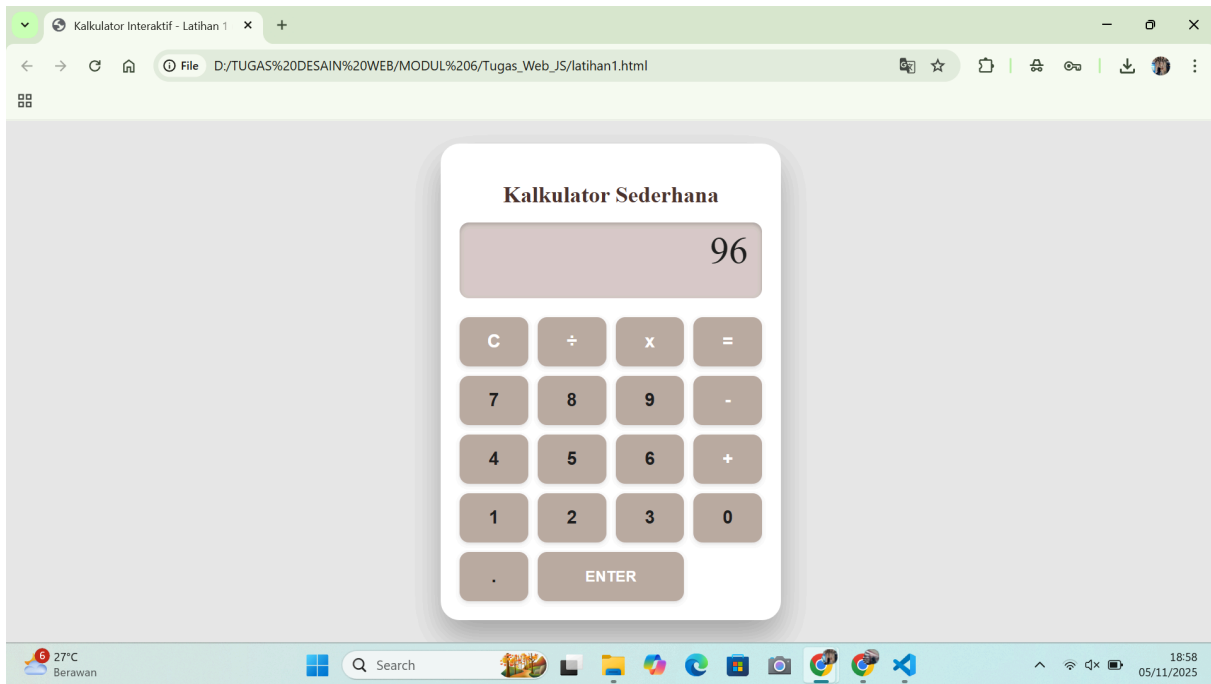
Uji coba Kalkulator sederhana dengan memasukkan bilangan pertama, contohnya : 12



kemudian klik bebas (+, :, -, x) disini saya memilih x dan masukkan bilangan kedua : 8



Kemudian klik = dan hasilnya :



Penjelasan : disini kalkulator sederhana memiliki fungsi yang sama seperti kalkulator yang ada pada umumnya di handphone, dan kalkulator mekanik. C memiliki fungsi untuk membersihkan angka bilangan yang ada pada layar kalkulator.

3. Implementasi Logika JavaScript Kalkulator

Tujuan Langkah: Mengimplementasikan logika kalkulator interaktif yang menangani input tombol, pemilihan operator, dan perhitungan bertingkat.

Kode JS (scripts/main1.js) - Bagian State dan Fungsi. Kode JS (scripts/main1.js) - Fungsi appendOperator dan performCalculation :

```

1  // hitung
2  * Fungsi untuk melakukan perhitungan kalkulator
3  * @param (string) operasi - jenis operasi: 'jumlah', 'kurang', 'kali', atau 'bagi'
4  */
5  function hitung(operasi) {
6      // Mengambil nilai dari input dan mengubahnya menjadi angka (floating point)
7      const angka1 = parseFloat(document.getElementById('angka1').value);
8      const angka2 = parseFloat(document.getElementById('angka2').value);
9      let hasil;
10
11      // Memeriksa apakah input valid
12      if (isNaN(angka1) || isNaN(angka2)) {
13          document.getElementById('hasil').innerHTML = "Hasil: **Masukkan dua angka yang valid!**";
14          return;
15      }
16
17      // Melakukan perhitungan berdasarkan operasi
18      switch (operasi) {
19          case 'jumlah':
20              hasil = angka1 + angka2;
21              break;
22          case 'kurang':
23              hasil = angka1 - angka2;
24              break;
25          case 'kali':
26              hasil = angka1 * angka2;
27              break;
28          case 'bagi':
29              if (angka2 === 0) {
30                  document.getElementById('hasil').innerHTML = "Hasil: **Tidak bisa dibagi nol!**";
31                  return;
32              }
33              hasil = angka1 / angka2;
34              break;
35          default:
36              hasil = "Operasi tidak valid";
37      }
38
39      // Menampilkan hasil pada elemen div#hasil menggunakan innerHTML
40      document.getElementById('hasil').innerHTML = "Hasil ${operasi}: **${hasil}**";
41  }

```

```

/**
 * Fungsi untuk melakukan perhitungan kalkulator
 * @param {string} operasi - Jenis operasi: 'jumlah', 'kurang', 'kali', atau 'bagi'
 */
function hitung(operasi) {
    // Mengambil nilai dari input dan mengubahnya menjadi angka (floating point)
    const angka1 = parseFloat(document.getElementById('angka1').value);
    const angka2 = parseFloat(document.getElementById('angka2').value);
    let hasil;

    // Memeriksa apakah input valid
    if (isNaN(angka1) || isNaN(angka2)) {
        document.getElementById('hasil').innerHTML = "Hasil: **Masukkan dua angka yang valid!**";
        return;
    }

    // Melakukan perhitungan berdasarkan operasi
    switch (operasi) {
        case 'jumlah':
            hasil = angka1 + angka2;
            break;
        case 'kurang':
            hasil = angka1 - angka2;
            break;
        case 'kali':
            hasil = angka1 * angka2;
            break;
        case 'bagi':
            if (angka2 === 0) {
                document.getElementById('hasil').innerHTML = "Hasil: **Tidak bisa dibagi nol!**";
                return;
            }
            hasil = angka1 / angka2;
            break;
        default:
            hasil = "Operasi tidak valid";
    }

    // Menampilkan hasil pada elemen div#hasil menggunakan innerHTML
    document.getElementById('hasil').innerHTML = `Hasil ${operasi}: **${hasil}**`;
}

```

II. LATIHAN 2: HIGHLIGHT TEKS DINAMIS

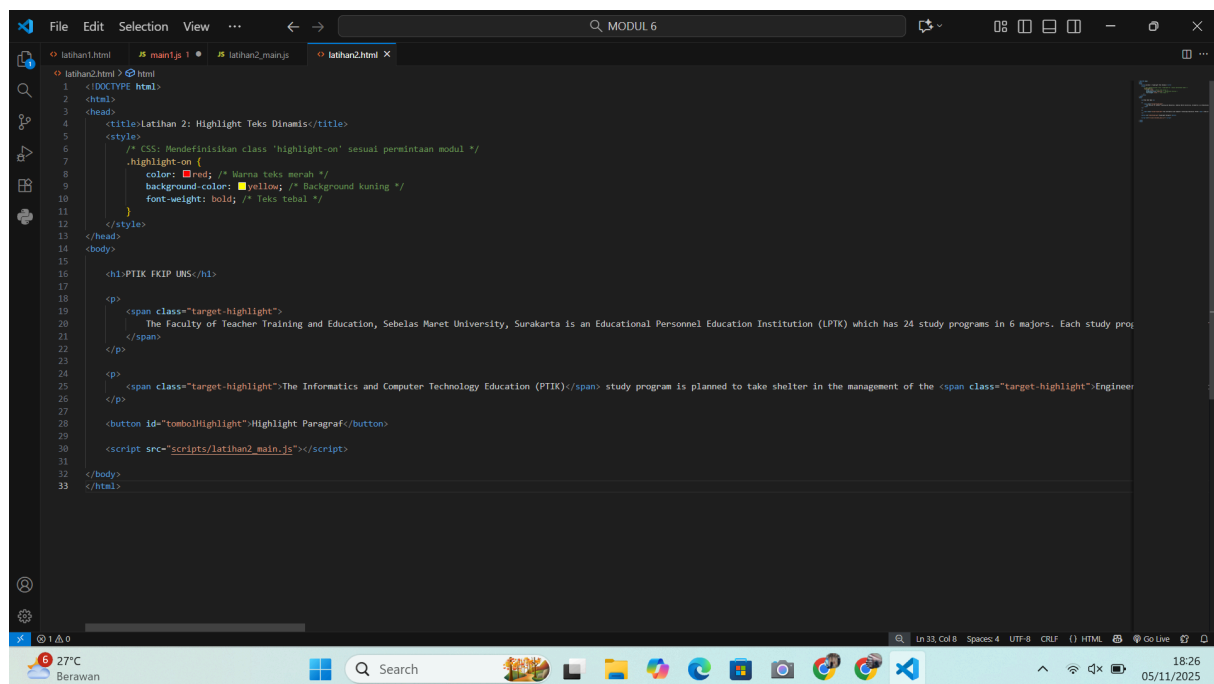
Tugas: Langkah kerja diuraikan berdasarkan implementasi kode HTML (Struktur Paragraf) dan JavaScript (Logika *Toggle Class*).

LANGKAH KERJA:

1. Membuat file baru bernama latihan2.html.
2. Menulis struktur HTML dan CSS internal untuk mendefinisikan *class* highlight-on (merah, kuning, tebal).
3. **Modifikasi HTML Kritis:** Membungkus setiap bagian teks yang ingin di-highlight dengan tag ``.
4. Di dalam `<script>`, menggunakan `document.querySelectorAll(".target-highlight")` untuk mendapatkan daftar semua elemen yang perlu diubah.
5. Mendefinisikan fungsi `toggleHighlight()` yang menggunakan *loop* (`forEach`) pada daftar elemen tersebut dan menerapkan `classList.toggle("highlight-on")`.
6. Menghubungkan fungsi `toggleHighlight()` ke event click pada tombol.

1. Pembuatan Struktur HTML dan Class CSS

Tujuan Langkah: Membuat paragraf target dan tombol pemicu, serta membuat *class* CSS khusus untuk efek *highlight* (latar belakang kuning, teks merah, teks tebal).



```
<!DOCTYPE html>
<html>
<head>
  <title>Latihan 2: Highlight Teks Dinamis</title>
  <style>
    /* CSS: Mendefinisikan class 'highlight-on' sesuai permintaan modul */
    .highlight-on {
      color: red; /* Warna teks merah */
      background-color: yellow; /* Background kuning */
      font-weight: bold; /* Teks tebal */
    }
  </style>
</head>
<body>
```



```

<h1>PTIK FKIP UNS</h1>

<p>
  <span class="target-highlight">
    The Faculty of Teacher Training and Education, Sebelas Maret
    University, Surakarta is an Educational Personnel Education Institution (LPTK)
    which has 24 study programs in 6 majors. Each study program has its own
    characteristics in producing superior, strong and intelligent educational staff.
  </span>
</p>

<p>
  <span class="target-highlight">The Informatics and Computer Technology
  Education (PTIK)</span> study program is planned to take shelter in the management
  of the <span class="target-highlight">Engineering and Vocational Education
  (PTK)</span> department. This placement revises our previous statement, where the
  PTIK study program is under the P.MIPA department. PTIK is a study program that
  will produce graduates who are <span class="target-highlight">prioritized to teach
  in vocational/vocational programs</span>. So, it would be better if PTIK was under
  the PTK department which oversees vocational programs such as <span
  class="target-highlight">Building Engineering Education</span>, and <span
  class="target-highlight">Mechanical Engineering Education</span>. To meet the needs
  of lecturers in the field of information technology that cannot be met by the PTK
  department, PTIK will conduct <span class="target-highlight">resource sharing with
  the mathematics study program</span>.
</p>

<button id="tombolHighlight">Highlight Paragraf</button>

<script src="scripts/latihan2_main.js"></script>

</body>
</html>

```

2. Implementasi Logika JavaScript (*Toggle Class*)

Tujuan Langkah: Mengimplementasikan fungsi JS untuk menambahkan atau menghapus *class highlight* pada paragraf target setiap kali tombol diklik (*toggle*).

```
File Edit Selection View ... MODUL 6
latihan1.html main1.js latihan2_main.js latihan2.html
scripts > scripts > latihan2_main.js > toggleHighlight
1 /*
2  * Fungsi untuk melakukan toggle highlight pada semua elemen
3  * dengan class 'target-highlight' dan mengubah teks tombol.
4  */
5 function toggleHighlight() {
6     // 1. Dapatkan SEMUA elemen yang ditargetkan menggunakan class
7     // document.querySelectorAll mengembalikan NodeList (mirip array) dari semua elemen yang cocok.
8     const targetTeksList = document.querySelectorAll(".target-highlight");
9
10    // 2. Dapatkan elemen tombol
11    const tombol = document.getElementById("tombolHighlight");
12
13    // Lakukan perulangan (loop) pada setiap elemen yang ditemukan
14    targetTeksList.forEach(element => {
15        // classList.toggle akan menambah class 'highlight-on' jika belum ada, dan menghapusnya jika sudah ada.
16        element.classList.toggle("highlight-on");
17    });
18
19    // Mengubah teks tombol secara dinamis berdasarkan status highlight
20    // Kita cek status dari elemen pertama saja (targetTeksList[0])
21    if (targetTeksList.length > 0 && targetTeksList[0].classList.contains("highlight-on")) {
22        tombol.textContent = "Hapus Highlight";
23    } else {
24        tombol.textContent = "Highlight Paragraf";
25    }
26
27    // 3. Menghubungkan fungsi 'toggleHighlight' ke event 'click' pada tombol
28    // Ini adalah cara yang disarankan (menggunakan addEventListener) daripada menggunakan onclick="" di HTML.
29    document.addEventListener("DOMContentLoaded", () => {
30        const tombol = document.getElementById("tombolHighlight");
31        if (tombol) {
32            tombol.addEventListener('click', toggleHighlight);
33        }
34    });
35 }
```

```
/**
 * Fungsi untuk melakukan toggle highlight pada semua elemen
 * dengan class 'target-highlight' dan mengubah teks tombol.
 */
function toggleHighlight() {
    // 1. Dapatkan SEMUA elemen yang ditargetkan menggunakan class
    // document.querySelectorAll mengembalikan NodeList (mirip array) dari semua
    // elemen yang cocok.
    const targetTeksList = document.querySelectorAll(".target-highlight");

    // 2. Dapatkan elemen tombol
    const tombol = document.getElementById("tombolHighlight");

    // Lakukan perulangan (loop) pada setiap elemen yang ditemukan
    targetTeksList.forEach(element => {
        // classList.toggle akan menambah class 'highlight-on' jika belum ada, dan
        // menghapusnya jika sudah ada.
        element.classList.toggle("highlight-on");
    });

    // Mengubah teks tombol secara dinamis berdasarkan status highlight
    // Kita cek status dari elemen pertama saja (targetTeksList[0])
    if (targetTeksList.length > 0 &&
targetTeksList[0].classList.contains("highlight-on")) {
        tombol.textContent = "Hapus Highlight";
    } else {
        tombol.textContent = "Highlight Paragraf";
    }
}

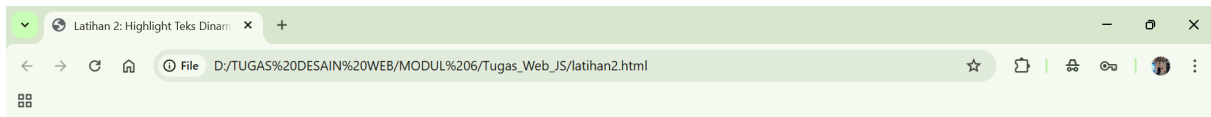
// 3. Menghubungkan fungsi 'toggleHighlight' ke event 'click' pada tombol
```

```
// Ini adalah cara yang disarankan (menggunakan addEventListener) daripada
menggunakan onclick="" di HTML.
document.addEventListener('DOMContentLoaded', () => {
    const tombol = document.getElementById("tombolHighlight");
    if (tombol) {
        tombol.addEventListener('click', toggleHighlight);
    }
});
```

Penjelasan: HTML menggunakan `` dengan *class* `target-highlight` sebagai penanda. JavaScript kemudian menggunakan `querySelectorAll` untuk mengambil semua penanda tersebut dan method `forEach` bersama `classList.toggle` untuk memanipulasi gaya (CSS) dari setiap target secara serentak.

3. Tampilan Hasil Akhir Latihan 2 (Preview) - Highlight AKTIF

Menampilkan hasil di *browser* (preview) ketika tombol sudah diklik dan *highlight* telah aktif (paragraf berubah menjadi latar belakang kuning, teks merah, dan tebal).

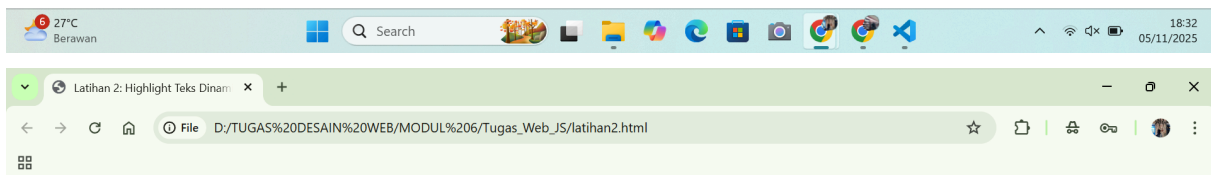


PTIK FKIP UNS

The Faculty of Teacher Training and Education, Sebelas Maret University, Surakarta is an Educational Personnel Education Institution (LPTK) which has 24 study programs in 6 majors. Each study program has its own characteristics in producing superior, strong and intelligent educational staff.

The Informatics and Computer Technology Education (PTIK) study program is planned to take shelter in the management of the Engineering and Vocational Education (PTK) department. This placement revises our previous statement, where the PTIK study program is under the P.MIPA department. PTIK is a study program that will produce graduates who are prioritized to teach in vocational/vocational programs. So, it would be better if PTIK was under the PTK department which oversees vocational programs such as Building Engineering Education, and Mechanical Engineering Education. To meet the needs of lecturers in the field of information technology that cannot be met by the PTK department, PTIK will conduct resource sharing with the mathematics study program.

Highlight Paragraf



PTIK FKIP UNS

The Faculty of Teacher Training and Education, Sebelas Maret University, Surakarta is an Educational Personnel Education Institution (LPTK) which has 24 study programs in 6 majors. Each study program has its own characteristics in producing superior, strong and intelligent educational staff.

The Informatics and Computer Technology Education (PTIK) study program is planned to take shelter in the management of the Engineering and Vocational Education (PTK) department. This placement revises our previous statement, where the PTIK study program is under the P.MIPA department. PTIK is a study program that will produce graduates who are prioritized to teach in vocational/vocational programs. So, it would be better if PTIK was under the PTK department which oversees vocational programs such as Building Engineering Education, and Mechanical Engineering Education. To meet the needs of lecturers in the field of information technology that cannot be met by the PTK department, PTIK will conduct resource sharing with the mathematics study program.

Hapus Highlight



E. KESIMPULAN

Praktikum ini berhasil membuktikan kemampuan JavaScript sebagai bahasa skrip *client-side* untuk membuat halaman web yang interaktif.

1. **Latihan 1** menunjukkan bahwa JavaScript dapat digunakan untuk menerima input dari pengguna (prompt()), melakukan perhitungan aritmatika, dan menampilkan output dalam berbagai format (alert() dan document.getElementById().innerHTML). Konversi tipe data menggunakan parseFloat() adalah langkah krusial untuk memastikan operasi matematika berjalan akurat.

2. **Latihan 2** menegaskan pentingnya DOM Manipulation. Dengan menggunakan `document.querySelectorAll()` untuk menargetkan banyak elemen dan `classList.toggle()`, kita dapat secara efisien mengubah gaya (CSS) halaman web berdasarkan aksi pengguna (klik tombol), menciptakan interaksi *toggle* (mengaktifkan/menonaktifkan) *highlight* secara dinamis. Kemampuan ini menjadi dasar dalam pengembangan antarmuka pengguna yang responsif.