

PANDUAN MEMBUAT PROGRAM KALKULATOR WEB

Pendahuluan

Pada Modul ini, Anda akan diarahkan untuk membuat sebuah kalkulator analog berbasis web dengan fitur riwayat perhitungan dan mode tema gelap. Modul ini mencakup langkah-langkah untuk memanfaatkan file HTML, CSS, dan JavaScript bersama dengan pemrograman PHP untuk memproses data.

Tujuan

Membuat sebuah kalkulator analog berbasis web yang memiliki fitur:

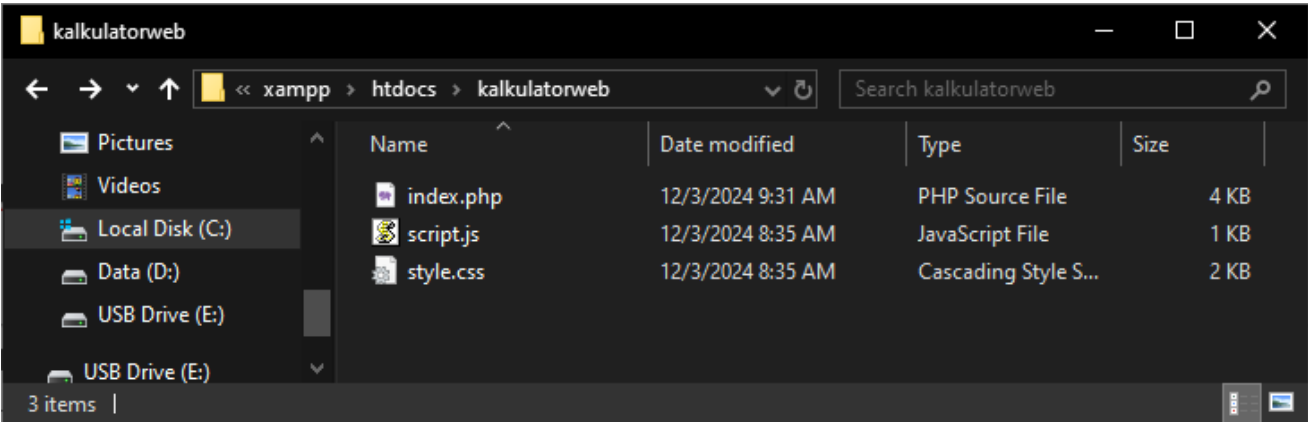
- Tampilan responsif.
- Mode tema terang dan gelap.
- Riwayat perhitungan yang dapat dihapus.

Alat dan Bahan

1. **Alat:**
 - Komputer dengan editor teks (misalnya VS Code, Sublime Text, atau Notepad++).
 - Browser untuk pengujian (misalnya Google Chrome, Firefox, atau Edge).
 - Server lokal (misalnya XAMPP atau WAMP).
2. **Bahan:**
 - File HTML, CSS, dan JavaScript.
 - Library Bootstrap 5 (dapat diakses melalui CDN).
 - PHP untuk logika pemrosesan.

Langkah 1: Persiapan Lingkungan

1. Buat Struktur Folder:



- index.php (file utama untuk antarmuka)
 - style.css (file untuk styling)
 - script.js (file untuk interaksi tema gelap)
2. **Unduh Library:**
 - Gunakan CDN Bootstrap 5 untuk styling.
 - Gunakan tag <link> di file HTML untuk menghubungkan Bootstrap dan file CSS tambahan.

Langkah 2: Membuat Dokumen HTML

Buka file **index.php** dan tuliskan tag tag dasar dokumen HTML Berikut

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title></title>
</head>
<body>

</body>
</html>
```

Kode ini mempersiapkan halaman web yang menggunakan Bootstrap untuk tampilan dan menyediakan tempat untuk menambahkan gaya CSS khusus, serta menetapkan halaman web sebagai kalkulator dengan Riwayat, Lengkapi bagaian tag head berikut :

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Kalkulator Analog dengan Riwayat</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="style.css">
</head>
```

Penejelasan Kode Program :

- <html lang="en">
Penjelasan: Tag <html> menandai awal dokumen HTML. Atribut lang="en" menunjukkan bahwa dokumen menggunakan Bahasa Inggris sebagai bahasa utama.
- <head>
Penjelasan: Tag <head> adalah bagian dokumen yang berisi metadata, seperti judul, encoding, dan tautan ke file eksternal.
- <meta charset="UTF-8">
Penjelasan: Menentukan karakter encoding yang digunakan adalah UTF-8, yang mendukung berbagai jenis karakter termasuk huruf internasional dan simbol khusus.
- <meta name="viewport" content="width=device-width, initial-scale=1.0">
Penjelasan: Mengatur halaman agar responsif, dengan menyesuaikan lebar tampilan sesuai perangkat yang digunakan dan memberikan skala awal 1.0.
- <title>Kalkulator Analog dengan Riwayat</title>
Penjelasan: Menentukan judul halaman web yang akan ditampilkan pada tab browser atau bookmark.
- <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
Penjelasan: Menghubungkan halaman dengan file CSS Bootstrap yang diambil dari CDN (Content Delivery Network) untuk memanfaatkan fitur dan gaya dari framework tersebut.
- <link rel="stylesheet" href="style.css">
Penjelasan: Menghubungkan halaman dengan file style.css yang berisi gaya kustom untuk halaman web.

Langkah 4: Membuat Tampilan Kalkulator

Disini kita akan membuat container untuk membuat tampilan kalkulator, Lengkapi bagaian tag body sebagai berikut :

```
<body>
<div class="container mt-5">
  <div class="calculator bg-light">
    <div class="card">
      <div class="text-end mb-1">
        <!-- Toggle switch untuk mode gelap -->
        <label class="form-check-label me-0 col-form-label-sm">Thema</label>
        <div class="form-check form-switch d-inline-block">
          <input class="form-check-input" type="checkbox" id="themeToggle">
        </div>
      </div>
      <div class="card-header text-center bg-primary text-white">
        <h4>Kalkulator Analog</h4>
      </div>
      <div class="card-body">
        <form method="post">
          <!-- Tampilan Display -->
          <input type="text" class="form-control mb-3 display" name="display" id="display" readonly
            value="<?php echo isset($_POST['display']) ? $_POST['display'] : ''; ?>">

          <!-- Tombol-Tombol -->
          <div class="row g-2 mb-3">
            <?php
            $buttons = [
              ['7', '8', '9', '/'],
              ['4', '5', '6', '*'],
              ['1', '2', '3', '-'],
              ['0', 'C', '=', '+']
            ];
            foreach ($buttons as $row) {
              foreach ($row as $btn) {
                $class = is_numeric($btn) ? 'btn-number' : ($btn === 'C' || $btn === '=' ? 'btn-clear' : 'btn-operator');
                echo '<div class="col-3">';
                echo '<button type="submit" name="button" value=" ' . $btn . '" class="btn ' . $class . ' w-100">' . $btn . '</button>';
                echo '</div>';
              }
            }
            ?>
          </div>

          <!-- Riwayat Perhitungan -->
          <textarea class="form-control mb-3" name="history" rows="5" readonly><?php
            echo isset($_POST['history']) ? $_POST['history'] : '';
          ?></textarea>

          <!-- Tombol Clear History -->
          <button type="submit" name="clear_history" class="btn btn-clear-history w-100">Clear History</button>
        </form>
      </div>
    </div>
  </div>
</div>
```

Penjelasan :

- <body>
Penjelasan: Tag <body> adalah bagian utama dokumen HTML yang berisi konten halaman yang terlihat oleh pengguna.
- <div class="container mt-5">

Penjelasan: Membuat elemen kontainer dengan margin atas sebesar 5 (kelas mt-5) untuk memberi jarak dari atas halaman. Kontainer ini memuat seluruh kalkulator.

- `<div class="calculator bg-light">`

Penjelasan: Membungkus elemen kalkulator dengan latar belakang warna terang (kelas bg-light).

- `<div class="card">`

Penjelasan: Membuat elemen kartu sebagai komponen utama untuk antarmuka kalkulator.

- `<div class="text-end mb-1">`

Penjelasan: Elemen pembungkus untuk pengaturan mode gelap dengan teks rata kanan (text-end) dan margin bawah kecil (mb-1).

- `<label class="form-check-label me-0 col-form-label-sm">Thema</label>`

Penjelasan: Label untuk tombol switch tema dengan teks kecil.

- `<div class="form-check form-switch d-inline-block">
<input class="form-check-input" type="checkbox" id="themeToggle">
</div>`

Penjelasan: Membuat toggle switch untuk mengaktifkan mode tema gelap. Input tipe checkbox memiliki ID themeToggle.

- `<div class="card-header text-center bg-primary text-white">
<h4>Kalkulator Analog</h4>
</div>`

Penjelasan: Bagian header kartu dengan teks tengah (text-center), latar biru (bg-primary), dan teks putih (text-white). Judul kalkulator ditampilkan dalam tag <h4>.

- `<div class="card-body">`

Penjelasan: Area utama dalam kartu untuk menempatkan elemen-elemen kalkulator seperti display, tombol, dan riwayat.

- `<form method="post">`

Penjelasan: Membuka tag form yang menggunakan metode POST untuk mengirim data ke server.

- `<input type="text" class="form-control mb-3 display" name="display" id="display" readonly
value="<?php echo isset($_POST['display']) ? $_POST['display'] : ''; ?>">`

Penjelasan: Elemen input teks sebagai layar kalkulator. Atribut readonly memastikan input ini hanya menampilkan data. Nilai diambil dari input sebelumnya menggunakan PHP.

- `<div class="row g-2 mb-3">`

Penjelasan: Membuat grid untuk tombol-tombol kalkulator dengan jarak antar-elemen (g-2) dan margin bawah (mb-3).

- ```
<?php
$buttons = [
 ['7', '8', '9', '/'],
 ['4', '5', '6', '*'],
 ['1', '2', '3', '-'],
 ['0', 'C', '=', '+']
];
foreach ($buttons as $row) {
 foreach ($row as $btn) {
 $class = is_numeric($btn) ? 'btn-number' : ($btn === 'C' || $btn === '=' ? 'btn-clear' : 'btn-operator');
 echo '<div class="col-3">';
 echo '<button type="submit" name="button" value="' . $btn . '" class="btn ' . $class . ' w-100">' . $btn . '</button>';
 echo '</div>';
 }
}
?>
```

**Penjelasan:** PHP digunakan untuk membuat tombol secara dinamis. Tombol diklasifikasikan sebagai angka (btn-number), operator (btn-operator), atau tombol aksi seperti C dan = (btn-clear). Semua tombol dimasukkan ke dalam elemen grid (col-3) dengan lebar penuh (w-100).

- `<textarea class="form-control mb-3" name="history" rows="5" readonly><?php  
echo isset($_POST['history']) ? $_POST['history'] : "";  
?></textarea>`

**Penjelasan:** Elemen `<textarea>` digunakan untuk menampilkan riwayat perhitungan. Data diambil dari input sebelumnya menggunakan PHP dan ditampilkan dalam kotak teks.

- `<button type="submit" name="clear_history" class="btn btn-clear-history w-100">Clear History</button>`

**Penjelasan:** Tombol untuk menghapus riwayat perhitungan. Saat diklik, mengirimkan data dengan nama `clear_history` ke server.

Langkah 5: Membuat Logika Perhitungan dan Riwayat

Masih didalam tag body kita akan menamahkan kode untuk logika perhitungan dan Riwayat, didalam Kode ini menjelaskan logika backend dan riwayat6 frontend untuk sebuah kalkulator berbasis web dengan riwayat perhitungan :

```
<?php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $current = $_POST['display'] ?? '';
 $button = $_POST['button'] ?? '';
 $history = $_POST['history'] ?? '';

 if (isset($_POST['clear_history'])) {
 $history = '';
 } elseif ($button === 'C') {
 $current = '';
 } elseif ($button === '=') {
 try {
 $result = eval("return $current;");
 $history .= $current . ' = ' . $result . PHP_EOL;
 $current = $result;
 } catch (Exception $e) {
 $current = 'Error';
 }
 } else {
 $current .= $button;
 }

 echo "<script>
 document.getElementById('display').value = '$current';
 document.getElementsByName('history')[0].value = '$history';
 </script>";
}
?>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script src="script.js"></script>
</body>
</html>
```

Penjelasan Kode Program :

- <?php  
if (\$\_SERVER['REQUEST\_METHOD'] === 'POST') {  
**Penjelasan:** Mengecek apakah metode request adalah POST. Hal ini memastikan kode hanya dijalankan saat form dikirimkan.
- \$current = \$\_POST['display'] ?? '';  
\$button = \$\_POST['button'] ?? '';  
\$history = \$\_POST['history'] ?? '';  
**Penjelasan:** Mengambil data dari input form:
  - o \$current adalah nilai dari layar kalkulator (display).
  - o \$button adalah nilai tombol yang ditekan.
  - o \$history adalah teks riwayat perhitungan.
- if (isset(\$\_POST['clear\_history'])) {  
\$history = '';  
}  
**Penjelasan:** Jika tombol "Clear History" ditekan, maka riwayat dihapus dengan mengosongkan variabel \$history.
- elseif (\$button === 'C') {  
\$current = '';  
}  
**Penjelasan:** Jika tombol C ditekan, nilai pada layar (\$current) dihapus.

```
- elseif ($button === '=') {
 try {
 $result = eval("return $current;");
 $history .= $current . ' = ' . $result . PHP_EOL;
 $current = $result;
 } catch (Exception $e) {
 $current = 'Error';
 }
}
```

**Penjelasan:**

- o Jika tombol = ditekan, string di layar (\$current) dievaluasi menggunakan fungsi eval.
- o Hasilnya ditambahkan ke riwayat dengan format "ekspresi = hasil".
- o Jika terjadi kesalahan (contohnya sintaks ekspresi tidak valid), layar akan menampilkan Error.

```
- else {
 $current .= $button;
}
```

**Penjelasan:** Jika tombol selain C atau = ditekan, nilai tombol ditambahkan ke string \$current.

```
- echo "<script>
 document.getElementById('display').value = '$current';
 document.getElementsByName('history')[0].value = '$history';
</script>";
```

**Penjelasan:**

- o Menambahkan kode JavaScript untuk memperbarui nilai layar (display) dan riwayat (history) secara langsung di sisi klien.
- o Ini memungkinkan pembaruan UI tanpa memuat ulang halaman sepenuhnya.

```
- <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
```

**Penjelasan:** Memuat file JavaScript Bootstrap untuk mendukung fitur interaktif seperti toggle switch.

```
- <script src="script.js"></script>
```

**Penjelasan:** Memuat file JavaScript eksternal bernama script.js. File ini digunakan untuk fungsionalitas tambahan pada kalkulator.

```
- </body>
</html>
```

**Penjelasan:** Menutup tag <body> dan <html>, menandai akhir dokumen.

Kode ini mengatur desain responsif dan estetis untuk kalkulator berbasis web, termasuk dukungan untuk **mode terang** dan **mode gelap**. Buka file **style.css** dan tuliskan kode berikut :

```
/* Gaya dasar untuk kalkulator */
.calculator {
 max-width: 400px;
 margin: auto;
 border-radius: 10px;
 overflow: hidden;
 box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
}

.display {
 font-size: 1.5rem;
 height: 3rem;
 text-align: right;
}

.btn-number {
 background-color: #f8f9fa;
 color: #212529;
}

.btn-number:hover {
 background-color: #e9ecef;
}

.btn-operator {
 background-color: #ffc107;
 color: #212529;
}

.btn-operator:hover {
 background-color: #ffcd39;
}

.btn-clear, .btn-clear-history {
 background-color: #dc3545;
 color: #fff;
}

.btn-clear:hover, .btn-clear-history:hover {
 background-color: #c82323;
}

/* Gaya mode gelap */
body.dark-mode {
 background-color: #212529;
 color: #f8f9fa;
}

body.dark-mode .card {
 background-color: #343a40;
}

body.dark-mode .btn-number {
 background-color: #495057;
 color: #f8f9fa;
}

body.dark-mode .btn-number:hover {
 background-color: #6c757d;
}

body.dark-mode .btn-operator {
 background-color: #ffc107;
 color: #212529;
}

body.dark-mode .btn-clear, body.dark-mode .btn-clear-history {
 background-color: #c82323;
}
```



Berikut penjelasan singkat untuk tiap baris kode:

- `.calculator { ... }`  
Menetapkan gaya untuk elemen dengan kelas `.calculator`, seperti batas maksimal lebar 400px, margin otomatis (pusatkan elemen), sudut membulat, dan bayangan kotak.
- `max-width: 400px;`  
Membatasi lebar maksimum elemen kalkulator menjadi 400px.
- `margin: auto;`  
Memusatkan elemen dengan margin otomatis di kiri dan kanan.
- `border-radius: 10px;`  
Membulatkan sudut elemen sebesar 10px.
- `overflow: hidden;`  
Menyembunyikan konten yang melampaui batas elemen.
- `box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);`  
Menambahkan bayangan pada kotak dengan offset horizontal 0px, offset vertikal 4px, dan blur 10px, serta warna bayangan setengah transparan.
- `.display { ... }`  
Menetapkan gaya untuk elemen dengan kelas `.display`, seperti ukuran font dan tinggi baris.
- `font-size: 1.5rem;`  
Mengatur ukuran font menjadi 1.5rem (relative to root element).
- `height: 3rem;`  
Menetapkan tinggi elemen display menjadi 3rem.
- `text-align: right;`  
Mengatur teks agar rata kanan di dalam elemen.
- `.btn-number { ... }`  
Menetapkan gaya untuk tombol angka, dengan latar belakang terang dan warna teks gelap.
- `background-color: #ff8f9fa;`  
Memberikan latar belakang tombol angka warna terang.
- `color: #212529;`  
Memberikan warna teks tombol angka dengan warna gelap.
- `.btn-number:hover { ... }`  
Menetapkan efek hover (saat kursor di atas tombol) pada tombol angka, dengan latar belakang sedikit lebih gelap.
- `background-color: #e9ecef;`  
Mengubah latar belakang tombol angka saat hover menjadi lebih gelap.
- `.btn-operator { ... }`  
Menetapkan gaya untuk tombol operator dengan latar belakang kuning dan teks gelap.
- `background-color: #ffc107;`  
Memberikan latar belakang tombol operator berwarna kuning.
- `color: #212529;`  
Memberikan warna teks tombol operator dengan warna gelap.
- `.btn-operator:hover { ... }`  
Menetapkan efek hover pada tombol operator dengan latar belakang lebih terang.
- `background-color: #ffcd39;`  
Mengubah latar belakang tombol operator saat hover menjadi lebih terang.
- `.btn-clear, .btn-clear-history { ... }`  
Menetapkan gaya untuk tombol hapus dan tombol hapus riwayat dengan latar belakang merah dan teks putih.
- `background-color: #dc3545;`  
Memberikan latar belakang merah pada tombol hapus dan tombol hapus riwayat.
- `color: #fff;`  
Memberikan warna teks putih pada tombol hapus dan tombol hapus riwayat.
- `.btn-clear:hover, .btn-clear-history:hover { ... }`  
Menetapkan efek hover pada tombol hapus dan tombol hapus riwayat dengan latar belakang merah lebih gelap.
- `background-color: #c82333;`  
Mengubah latar belakang tombol hapus dan tombol hapus riwayat menjadi merah lebih gelap saat hover.
- `body.dark-mode { ... }`  
Menetapkan gaya untuk mode gelap, dengan latar belakang gelap dan teks terang.
- `background-color: #212529;`  
Memberikan latar belakang gelap pada halaman saat mode gelap aktif.
- `color: #ff8f9fa;`  
Memberikan warna teks terang pada halaman saat mode gelap aktif.

- `body.dark-mode .card { ... }`  
Menetapkan gaya untuk elemen dengan kelas `.card` dalam mode gelap, memberikan latar belakang lebih gelap.
- `background-color: #343a40;`  
Mengatur latar belakang elemen `.card` dalam mode gelap menjadi lebih gelap.
- `body.dark-mode .btn-number { ... }`  
Menetapkan gaya untuk tombol angka dalam mode gelap dengan latar belakang abu-abu dan teks terang.
- `background-color: #495057;`  
Memberikan latar belakang abu-abu gelap pada tombol angka di mode gelap.
- `color: #f8f9fa;`  
Memberikan warna teks terang pada tombol angka di mode gelap.
- `body.dark-mode .btn-number:hover { ... }`  
Menetapkan efek hover pada tombol angka di mode gelap dengan latar belakang lebih terang.
- `background-color: #6c757d;`  
Mengubah latar belakang tombol angka saat hover di mode gelap menjadi lebih terang.
- `body.dark-mode .btn-operator { ... }`  
Menetapkan gaya untuk tombol operator dalam mode gelap dengan latar belakang kuning dan teks gelap.
- `background-color: #ffc107;`  
Memberikan latar belakang kuning pada tombol operator di mode gelap.
- `color: #212529;`  
Memberikan warna teks gelap pada tombol operator di mode gelap.
- `body.dark-mode .btn-clear, body.dark-mode .btn-clear-history { ... }`  
Menetapkan gaya untuk tombol hapus dan tombol hapus riwayat dalam mode gelap dengan latar belakang merah gelap.
- `background-color: #c82333;`  
Memberikan latar belakang merah gelap pada tombol hapus dan tombol hapus riwayat di mode gelap.

Kode ini menjelaskan bagaimana cara mengubah tema (mode gelap atau terang) pada halaman web dengan menggunakan JavaScript, dan menyimpan pilihan tema pengguna menggunakan localStorage. Secara keseluruhan, kode ini bertujuan untuk menyediakan fungsionalitas toggle theme (beralih tema) yang disimpan dan dipulihkan secara otomatis berdasarkan preferensi pengguna sebelumnya. Buka file **script.js** dan tuliskan kode berikut :

```
const themeToggle = document.getElementById('themeToggle');
const currentTheme = localStorage.getItem('theme');

if (currentTheme === 'dark') {
 document.body.classList.add('dark-mode');
 themeToggle.checked = true;
}

themeToggle.addEventListener('change', () => {
 if (themeToggle.checked) {
 document.body.classList.add('dark-mode');
 localStorage.setItem('theme', 'dark');
 } else {
 document.body.classList.remove('dark-mode');
 localStorage.setItem('theme', 'light');
 }
});
```

Berikut adalah penjelasan untuk tiap baris kode JavaScript tersebut:

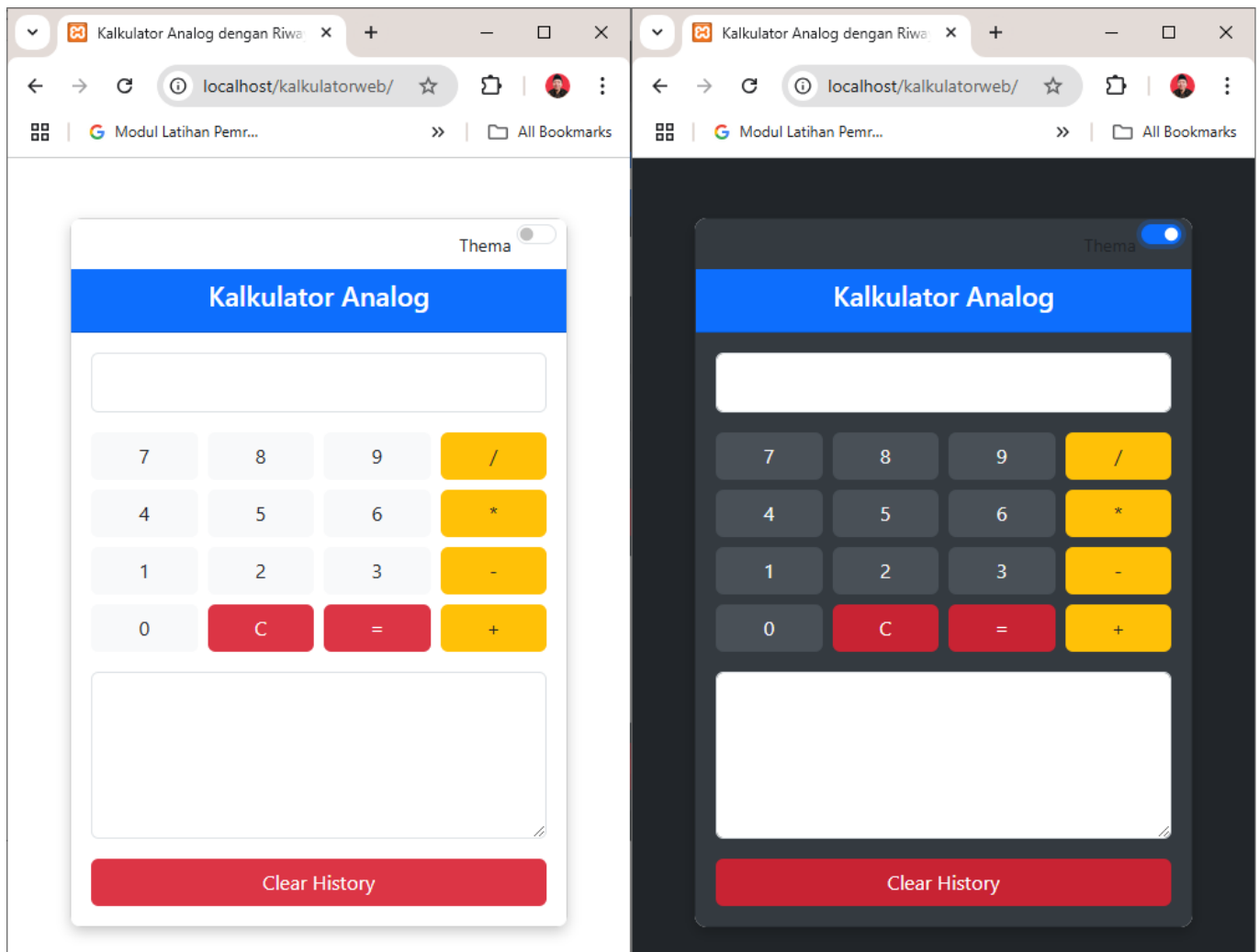
- `const themeToggle = document.getElementById('themeToggle');`  
Mendapatkan elemen HTML dengan ID themeToggle (misalnya, sebuah checkbox atau switch) dan menyimpannya dalam variabel themeToggle.
- `const currentTheme = localStorage.getItem('theme');`  
Mengambil nilai dari localStorage dengan key theme, yang menyimpan tema yang digunakan sebelumnya (dark atau light), dan menyimpannya dalam variabel currentTheme.
- `if (currentTheme === 'dark') {`  
Memeriksa apakah tema yang disimpan di localStorage adalah 'dark'.
- `document.body.classList.add('dark-mode');`  
Jika tema saat ini adalah 'dark', menambahkan kelas dark-mode ke elemen <body>, yang akan menerapkan gaya untuk mode gelap.
- `themeToggle.checked = true;`  
Mengatur properti checked pada elemen themeToggle menjadi true, menandakan bahwa switch atau checkbox diaktifkan, sesuai dengan tema gelap.
- `}`  
Menutup blok if yang memeriksa tema.
- `themeToggle.addEventListener('change', () => {`  
Menambahkan event listener untuk mendeteksi perubahan status pada elemen themeToggle (misalnya, saat switch diubah).
- `if (themeToggle.checked) {`  
Memeriksa apakah elemen themeToggle dalam keadaan tercentang (aktif) setelah perubahan.
- `document.body.classList.add('dark-mode');`  
Jika elemen themeToggle tercentang, menambahkan kelas dark-mode ke elemen <body>, yang mengaktifkan mode gelap.
- `localStorage.setItem('theme', 'dark');`  
Menyimpan 'dark' di localStorage dengan key theme untuk mengingat preferensi tema pengguna.
- `} else {`  
Jika elemen themeToggle tidak tercentang (mode terang), blok kode ini akan dijalankan.

- `document.body.classList.remove('dark-mode');`  
Menghapus kelas `dark-mode` dari elemen `<body>`, yang akan mematikan mode gelap dan mengembalikannya ke mode terang.
- `localStorage.setItem('theme', 'light');`  
Menyimpan 'light' di `localStorage` dengan key `theme` untuk mengingat bahwa pengguna memilih mode terang.
- `}`  
Menutup blok `else`.
- `});`  
Menutup event listener untuk perubahan status pada elemen `themeToggle`.

## Langkah 8: Integrasi dan Uji Coba

1. **Hubungkan File:**
  - o Pastikan semua file (HTML, CSS, JS) terhubung dengan benar.
2. **Uji Fungsionalitas:**
  - o Periksa semua tombol kalkulator.
  - o Pastikan mode gelap berjalan sesuai harapan.
  - o Validasi riwayat perhitungan tampil dengan benar.

## Hasil Akhir



Kalkulator analog Anda siap digunakan dengan fitur:

- Tampilan responsif.
- Mode tema terang dan gelap.
- Riwayat perhitungan.

### Catatan

- Pastikan semua file disimpan dalam direktori yang sesuai.
- Gunakan XAMPP atau server lokal lainnya untuk menguji PHP.