

LAPORAN HASIL APLIKASI UKK

Oleh:

AMANDA BR TAMBUNAN

NIS. 22161002



SMK PERMATA HARAPAN

JURUSAN REKAYASA PERANGKAT LUNAK

2025

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Standard Calculator</title>
<style>

```

Yang pertama kita membuat kode seperti ini di bagian awal dari halaman web untuk sebuah **kalkulator standar**, yang kemungkinan akan ditambahkan elemen lain seperti tombol dan input angka di bagian selanjutnya.

```

.calculator-container {
  display: flex;
  justify-content: center;
  align-items: flex-start;
  gap: 30px;
  margin-top: 30px;
}

.calculator, .history {
  display: flex;
  flex-direction: column;
  gap: 10px;
  background: linear-gradient(135deg, #d1b3e0, #f8c6d8);
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
  width: 400px;
  height: 550px;
  color: white;
}

```

Kemudian selanjutnya, ada CSS yang mengatur tampilan kalkulator dan riwayat perhitungan dalam sebuah halaman web. Berikut penjelasannya:

1. `.calculator-container` → Mengatur wadah utama yang berisi kalkulator dan riwayat perhitungan:
 - `display: flex;` → Mengatur elemen agar sejajar secara horizontal.
 - `justify-content: center;` → Posisikan elemen ke tengah secara horizontal.
 - `align-items: flex-start;` → Posisi elemen sejajar ke atas.
 - `gap: 30px;` → Jarak antara kalkulator dan riwayat 30px.
 - `margin-top: 30px;` → Memberi jarak dari atas sebesar 30px.
2. `.calculator, .history` → Mengatur tampilan kotak kalkulator dan riwayat perhitungan:
 - `display: flex; flex-direction: column;` → Menyusun elemen dalam bentuk vertikal.
 - `gap: 10px;` → Jarak antar elemen di dalamnya 10px.
 - `background: linear-gradient(135deg, #d1b3e0, #f8c6d8);` → Warna gradasi ungu ke pink.
 - `padding: 20px;` → Ruang kosong di dalam kotak 20px.

- border-radius: 10px; → Membuat sudut kotak melengkung.
- box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2); → Menambahkan bayangan agar tampak lebih elegan.
- width: 400px; height: 550px; → Ukuran kotak 400px lebar dan 550px tinggi.
- color: white; → Warna teks putih agar kontras dengan latar belakang.

```
.calculator {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 10px;
}

.display {
  grid-column: span 4;
  background: linear-gradient(135deg, #e2c6f2, #fad0e6);
  border-radius: 5px;
  padding: 20px;
  font-size: 32px;
  text-align: right;
  overflow: hidden;
  white-space: nowrap;
}
```

Lalu ada CSS yang mengatur tampilan kalkulator, khususnya tata letak tombol dan layar tampilan hasil perhitungan. Berikut penjelasannya:

1. .calculator → Mengatur tampilan wadah tombol kalkulator:

- display: grid; → Menggunakan grid layout untuk menata tombol.
- grid-template-columns: repeat(4, 1fr); → Membuat 4 kolom dengan ukuran yang sama.
- gap: 10px; → Jarak antar tombol 10px.

2. .display → Mengatur tampilan layar kalkulator (tempat menampilkan angka & hasil):

- grid-column: span 4; → Layar ini memanjang ke 4 kolom, jadi selebar kalkulator.
- background: linear-gradient(135deg, #e2c6f2, #fad0e6); → Warna gradasi ungu muda ke pink agar estetik.
- border-radius: 5px; → Membuat sudut layar sedikit melengkung.
- padding: 20px; → Memberi ruang kosong di dalam layar agar teks tidak terlalu mepet.
- font-size: 32px; → Ukuran teks besar agar mudah dibaca.
- text-align: right; → Angka rata kanan, seperti kalkulator pada umumnya.
- overflow: hidden; → Jika teks panjang, akan tersembunyi agar tidak keluar dari layar.
- white-space: nowrap; → Mencegah teks turun ke baris baru

```

.calculator button {
  font-size: 22px;
  padding: 20px;
  border: none;
  border-radius: 5px;
  background: linear-gradient(135deg, #b277d6, #d48cb3);
  color: white;
  cursor: pointer;
  transition: background 0.2s ease, transform 0.1s ease;
  box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);
}

.calculator button:active {
  background: linear-gradient(135deg, #9b5fc1, #c0789c);
  transform: scale(0.95);
}

.calculator button.equal {
  background: linear-gradient(135deg, #f37ba3, #f8bbd0);
  font-weight: bold;
}

.history h3 {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 10px;
}

.history .entry {
  font-size: 18px;
  margin: 5px 0;
  padding: 5px;
  background: rgba(255, 255, 255, 0.2);
  border-radius: 5px;
  color: white;
}

```

Kemudian mengatur tampilan tombol kalkulator dan bagian riwayat perhitungan:

1. .calculator button → Mengatur tampilan tombol kalkulator:

- Teks besar (22px), padding luas (20px) agar nyaman ditekan.
- Tombol tanpa border, sudut melengkung (border-radius: 5px).
- Warna gradasi ungu ke pink, teks putih.
- Efek saat ditekan (:active) → Tombol sedikit mengecil & warnanya lebih gelap.
- Tombol "=" (.equal) → Warna lebih mencolok & teks tebal.

2. .history → Mengatur tampilan riwayat perhitungan:

- Judul (h3) → Rapi dengan teks sejajar kiri & kanan.
- Riwayat (.entry) → Teks 18px, latar belakang transparan, sudut melengkung, warna teks putih.

```
#delete-history {
  background: none;
  border: none;
  color: white;
  font-size: 18px;
  cursor: pointer;
}
```

Selanjutnya, Kode di atas mengatur tampilan tombol hapus riwayat (#delete-history) pada kalkulator:

- background: none; → Tidak ada latar belakang (transparan).
- border: none; → Tanpa garis tepi (border).
- color: white; → Warna teks putih agar kontras dengan latar belakang.
- font-size: 18px; → Ukuran teks 18px agar mudah dibaca.
- cursor: pointer; → Mengubah kursor menjadi tangan (pointer) saat diarahkan ke tombol.

```
</style>
</head>
<body>
  <div class="calculator-container">
    <div class="calculator">
      <div class="display" id="display">0</div>
      <button class="special">C</button>
      <button class="backspace">&larr;</button>
      <button>/</button>
      <button>*</button>
      <button>7</button>
      <button>8</button>
      <button>9</button>
      <button>-</button>
      <button>4</button>
      <button>5</button>
      <button>6</button>
      <button>+</button>
      <button>1</button>
      <button>2</button>
      <button>3</button>
      <button>0</button>
      <button>.</button>
      <button class="equal">=</button>
    </div>
    <div class="history" id="history">
      <h3>
        History
        <button id="delete-history">🗑️</button>
      </h3>
      <p id="no-history">No history available.</p>
    </div>
  </div>
```

Lalu ada kode untuk mengatur tampilan tombol hapus riwayat (#delete-history) pada kalkulator:

- background: none; → Tidak ada latar belakang (transparan).
- border: none; → Tanpa garis tepi (border).
- color: white; → Warna teks putih agar kontras dengan latar belakang.
- font-size: 18px; → Ukuran teks 18px agar mudah dibaca.
- cursor: pointer; → Mengubah kursor menjadi tangan (pointer) saat diarahkan ke tombol.

```

<script>
const display = document.getElementById('display');
const historyContainer = document.getElementById('history');
const buttons = document.querySelectorAll('.calculator button');
let currentExpression = '';

function updateHistoryUI() {
  const history = JSON.parse(localStorage.getItem('history')) || [];
  historyContainer.innerHTML = '<h3>History <button id="delete-history">🗑️</button></h3>';
  if (history.length === 0) {
    historyContainer.innerHTML += '<p id="no-history">No history available.</p>';
  } else {
    history.forEach(entry => {
      const div = document.createElement('div');
      div.classList.add('entry');
      div.textContent = `${entry.expression} = ${entry.result}`;
      historyContainer.appendChild(div);
    });
  }
  document.getElementById('delete-history').addEventListener('click', clearHistory);
}

```

Kemudian ada kode ada JavaScript yang mengatur tampilan layar kalkulator dan riwayat perhitungan:

1. Mengambil elemen HTML

- display → Layar kalkulator.
- historyContainer → Wadah riwayat perhitungan.
- buttons → Semua tombol kalkulator.
- currentExpression → Menyimpan ekspresi matematika yang sedang diketik.

2. Fungsi updateHistoryUI()

- Mengambil riwayat dari localStorage.
- Menampilkan judul "History" dengan tombol hapus (🗑️).
- Jika tidak ada riwayat → tampilkan teks "No history available."
- Jika ada riwayat → tambahkan hasil perhitungan ke daftar.
- Tombol hapus (delete-history) akan menghapus semua riwayat saat diklik.

```

buttons.forEach(button => {
  button.addEventListener('click', () => {
    const value = button.textContent;
    if (value === 'C') {
      display.textContent = '0';
      currentExpression = '';
    } else if (value === '=') {
      try {
        const result = eval(currentExpression);
        display.textContent = result;
        addHistory(currentExpression, result);
        currentExpression = result.toString();
      } catch {
        display.textContent = 'Error';
      }
    } else if (value === '+') {
      currentExpression = currentExpression.slice(0, -1);
      display.textContent = currentExpression || '0';
    } else {
      currentExpression += value;
      display.textContent = currentExpression;
    }
  });
});

```

Lalu ada Kode JavaScript yang mengatur fungsi tombol kalkulator:

1. Mendeteksi Klik Tombol → Setiap tombol kalkulator akan menjalankan kode saat diklik.

2. Fungsi Tiap Tombol:

- C (Clear) → Menghapus semua input & mengatur layar ke "0".
- = (Hasil) → Menghitung ekspresi dengan eval(), menampilkan hasil, dan menyimpan ke riwayat.
- ← (Backspace) → Menghapus satu karakter terakhir dari input.
- Angka & Operator → Ditambahkan ke ekspresi saat ditekan.

```
function addHistory(expression, result) {  
  const history = JSON.parse(localStorage.getItem('history')) || [];  
  history.push({ expression, result });  
  localStorage.setItem('history', JSON.stringify(history));  
  updateHistoryUI();  
}  
  
function clearHistory() {  
  localStorage.removeItem('history');  
  updateHistoryUI();  
}  
  
updateHistoryUI();  
</script>  
</body>  
</html>
```

Lalu Kode di atas adalah JavaScript yang mengatur penyimpanan dan penghapusan riwayat perhitungan kalkulator menggunakan localStorage.

1. addHistory(expression, result) → Menyimpan Riwayat

- Mengambil riwayat lama dari localStorage (jika ada).
- Menambahkan ekspresi & hasil terbaru ke dalam riwayat.
- Menyimpan kembali data ke localStorage dalam format JSON.
- Memperbarui tampilan riwayat dengan updateHistoryUI().

2. clearHistory() → Menghapus Riwayat

- Menghapus semua riwayat dari localStorage.
- Memperbarui tampilan agar kosong.

3. updateHistoryUI();

- Dipanggil saat halaman dimuat untuk menampilkan riwayat yang sudah tersimpan.

```

<?php

namespace App\Controllers;

class Home extends BaseController
{
    public function standard()
    {
        echo view('header');

        echo view('standard');
        echo view('footer');
    }
}

```

Ini adalah Kode PHP (CodeIgniter 4) yang mengatur Controller untuk menampilkan halaman kalkulator standar.

1. namespace App\Controllers; → Menentukan bahwa file ini ada di dalam folder Controllers di aplikasi CodeIgniter.
2. class Home extends BaseController → Membuat Controller bernama Home, yang mewarisi fitur dari BaseController.
3. public function standard() → Fungsi yang dipanggil saat pengguna mengakses halaman kalkulator standar.
4. echo view('header'); → Menampilkan file header.php (biasanya berisi bagian atas halaman).
5. echo view('standard'); → Menampilkan file standard.php (kemungkinan tampilan kalkulator standar).
6. echo view('footer'); → Menampilkan file footer.php (biasanya berisi bagian bawah halaman).

