**Paket**

**1**

**APLIKASI KALKULATOR**

### LAPORAN HASIL UKK 2025

### SMK 1 SIMPANGKATIS

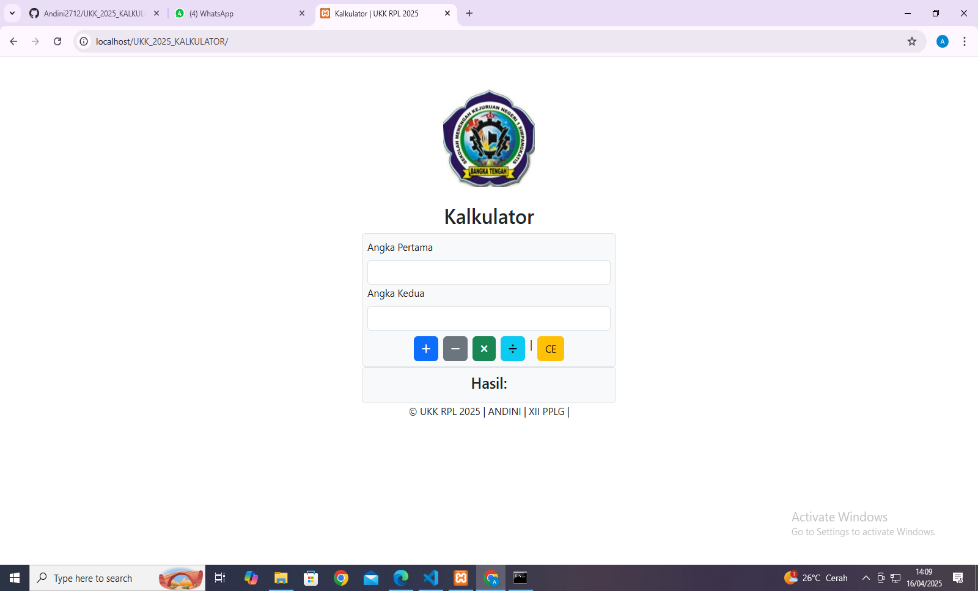
### ANDINI XII PPLG

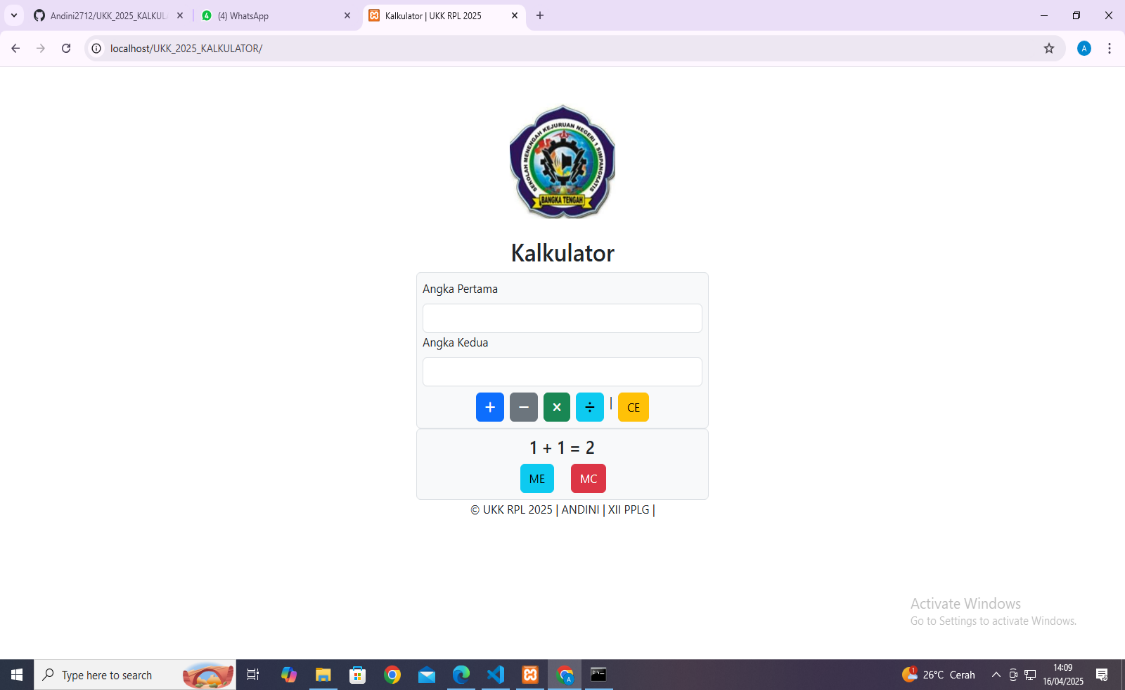


**PEMBAHASAN**

1. **Rancangan Desain Aplikasi**

Berikut adalah rancangan layar sistem yang dibangun.

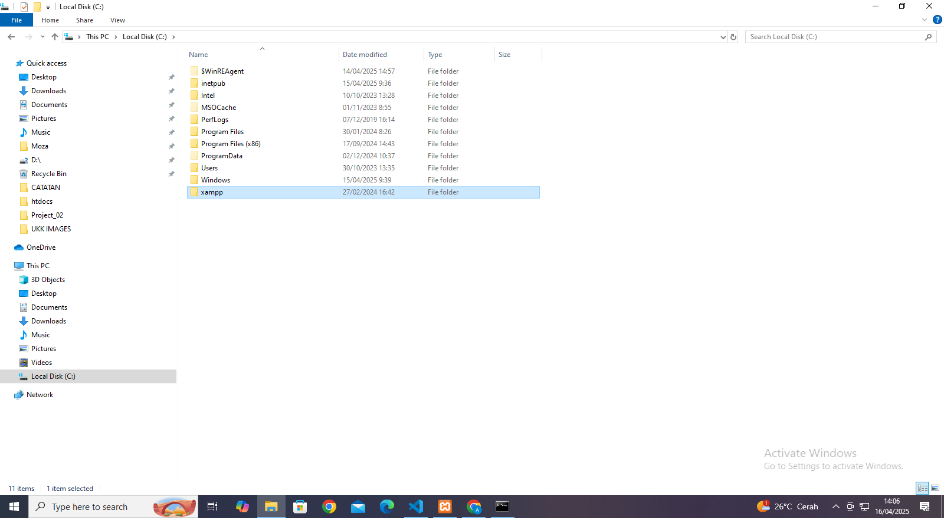




1. Tahap Implementasi merupakan tahap penerapan dari hasil analisis dan desain sistem yang telah dibuat sebelumnya.

Berikut langkah-langkah pembuatan aplikasi diskon.

1. Buatlah folder dengan nama ukk\_2025\_kalkulator yang disimpan didalam folder htdcos (C:\xampp\htdocs).



1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8">
5. <meta name="viewport" content="width=device-width, initial-scale=1">
6. <title>Kalkulator | UKK RPL 2025</title>
7. <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8. <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css">
9. <style type="text/css">
10. .logo {
11. width: 150px;
12. max-width: 100%;
13. height: auto;
14. margin: 0 auto;
15. }
16. </style>
17. </head>
18. <body>
19. <?php session\_start(); ?>
20. <div class="container mt-5">
21. <div class="text-center mb-4">
22. <img src="logo.jpg" alt="logo" class="logo">
23. </div>
24. <h2 class="text-center">Kalkulator</h2>
25. <div class="row justify-content-center">
26. <div class="col-md-4">
27. <form method="POST" class="p-2 border rounded bg-light">
28. <label class="form-label">Angka Pertama</label>
29. <input type="text" name="angka1" class="form-control" value="<?php echo isset($\_SESSION['memory']) ? $\_SESSION['memory'] : ''; ?>" required>
30. <label class="form-label">Angka Kedua</label>
31. <input type="text" name="angka2" class="form-control" required>
32. <div class="d-flex justify-content-center gap-2 mt-2">
33. <button type="submit" class="btn btn-primary" name="operator" value="+" title="Tambah"><i class="fas fa-plus"></i></button>
34. <button type="submit" class="btn btn-secondary" name="operator" value="-" title="Kurang"><i class="fas fa-minus"></i></button>
35. <button type="submit" class="btn btn-success" name="operator" value="\*" title="Kali"><i class="fas fa-xmark"></i></button>
36. <button type="submit" class="btn btn-info" name="operator" value="/" title="Bagi"><i class="fas fa-divide"></i></button>
37. |
38. <button type="reset" class="btn btn-warning" title="Clear Entry">CE</button>
39. </div>
40. </form>
41. <div class="p-2 border rounded bg-light">
42. <h4 class="text-center">
43. <?php
44. if (isset($\_POST['operator'])) {
45. $angka1 = $\_POST['angka1'];
46. $angka2 = $\_POST['angka2'];
47. $operator = $\_POST['operator'];
49. if (!preg\_match('/^-?\d+([,]\d+)?$/', $angka1) || !preg\_match('/^-?\d+([,]\d+)?$/', $angka2)) {
50. echo "<script>alert('Input harus berupa angka dengan koma sebagai desimal')</script>";
51. } elseif ($operator == '/' && $angka2 == '0') {
52. echo "<script>alert('Tidak dapat membagi dengan nol')</script>";
53. } else {
54. $angka1 = floatval(str\_replace(',', '.', $angka1));
55. $angka2 = floatval(str\_replace(',', '.', $angka2));
57. switch ($operator) {
58. case '+':
59. $hasil = $angka1 + $angka2;
60. break;
61. case '-':
62. $hasil = $angka1 - $angka2;
63. break;
64. case '\*':
65. $hasil = $angka1 \* $angka2;
66. break;
67. case '/':
68. $hasil = $angka1 / $angka2;
69. break;
70. default:
71. echo "Operator tidak valid";
72. break;
73. }
75. echo "$angka1 $operator $angka2 = " . str\_replace('.', ',', $hasil);
76. $\_SESSION['memory'] = str\_replace('.', ',', $hasil);
77. }
78. } else {
79. echo "Hasil: ";
80. }
81. ?>
82. </h4>
83. <div class="row mt-2">
84. <div class="col-6 text-end">
85. <?php if (!empty($hasil)) : ?>
86. <form method="POST">
87. <button type="submit" name="memory" class="btn btn-info" title="Memory Entry">ME</button>
88. </form>
89. <?php endif; ?>
90. </div>
91. <div class="col-6 text-start">
92. <?php if (isset($hasil) && $hasil !== null): ?>
93. <form method="POST">
94. <button type="submit" name="resetmemory" class="btn btn-danger" title="Memory Clear">MC</button>
95. </form>
96. <?php endif; ?>
97. </div>
98. </div>
99. </div>
101. <?php
102. if (isset($\_POST['resetmemory'])) {
103. unset($\_SESSION['memory']);
104. }
105. ?>
106. </div>
107. </div>
108. </div>
109. <p class="text-center">&copy; UKK RPL 2025 | ANDINI | XII PPLG |</p>
110. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
111. </body>
112. </html>

1. Apa tujuan utama dari project ini?

Untuk membuat aplikasi kalkulator sederhana berbasis web yang dapat menjumlah, mengurangi, mengalikan, dan membagi dua angka.

**2. Bahasa pemrograman apa saja yang kamu gunakan di sini?**

HTML, CSS (via Bootstrap), JavaScript, dan PHP.

#### 🧩 ****Struktur Data****

**17. Apa struktur data utama dalam kode ini?**

Variabel-variabel sederhana (angka1, angka2, operator, hasil) — ini bentuk data primitif (float/string).

### ****"Apa struktur data yang digunakan dalam program ini?"****

✅ **Jawaban:**

Struktur data yang digunakan dalam program ini adalah **tipe data primitif**, khususnya **angka (float)** dan **string**. Karena kalkulator ini hanya memproses dua angka dan operator, maka tidak perlu pakai array, object, atau struktur data kompleks lainnya.

### ****"Apa saja variabel yang digunakan dan fungsinya?"****

✅ **Jawaban:**

Berikut daftar variabel penting dalam program dan fungsinya:

| **Variabel** | **Fungsi** |
| --- | --- |
| $angka1 | Menyimpan nilai angka pertama dari input pengguna |
| $angka2 | Menyimpan nilai angka kedua dari input pengguna |
| $operator | Menyimpan simbol operasi yang dipilih pengguna (+, -, \*, /) |
| $hasil | Menyimpan hasil perhitungan dari $angka1 dan $angka2 sesuai operator |
| $\_POST[...] | Menyimpan data input yang dikirim dari form (metode POST) |

perintah kedua pas di tanya dapat memperbaiki program penguji minta di perubahan codingan seperti method penjumlahan sepereti di +,-, /,x

###### ChatGPT said:

Mantap! Kalau penguji minta **perubahan codingan ke bentuk method** (fungsi) untuk operasi +, -, \*, dan /, itu berarti kamu harus **membuat fungsi terpisah** untuk setiap operasi, lalu panggil fungsi tersebut di dalam switch.

Tujuannya:

* Biar kode lebih **terstruktur**
* Lebih **mudah dibaca**
* Menunjukkan pemahaman kamu soal **modularisasi program (fungsi/metode)**

### 📌 Penjelasan:

* Fungsi tambah, kurang, kali, dan bagi dibuat untuk memproses angka.
* switch sekarang lebih bersih dan hanya memanggil fungsi.
* Validasi pembagian dengan nol tetap dicek di luar fungsi agar aman.
* Tetap kompatibel dengan fitur **riwayat perhitungan** sebelumnya.

**"Jelaskan konsep data dan struktur data yang digunakan dalam aplikasi kalkulator ini!"**

### 1. ****Konsep Data****

* **Data input**: berasal dari pengguna melalui form (angka1 dan angka2) dengan tipe data **string**, kemudian diubah menjadi **float** (angka desimal).
* **Operator**: data yang dikirim dari tombol seperti +, -, \*, /, bertipe **string**.
* **Hasil**: hasil perhitungan, bertipe **float** atau **string** (kalau error).

### 2. ****Struktur Data yang Digunakan****

* **Variabel sederhana**:
  + $angka1, $angka2, $operator, $hasil → menyimpan nilai input & hasil
* **Array session**:
  + $\_SESSION['history'] → struktur data berupa **array**, menyimpan beberapa hasil perhitungan sebelumnya.
  + Setiap elemen dalam array adalah **string** hasil perhitungan, misalnya "5 + 3 = 8".

### 3. ****Fungsi / Modularisasi****

* Perhitungan menggunakan **fungsi-fungsi terpisah** seperti tambah(), kurang(), dll.
* Fungsi ini membantu mengorganisir kode berdasarkan **konsep pemrosesan data modular**.

## 🧩 Bonus (kalau ditanya lebih lanjut):

### "Apa manfaat menggunakan struktur data array di sini?"

Struktur data array (seperti $\_SESSION['history']) memudahkan untuk menyimpan dan menampilkan **riwayat perhitungan**, yang bisa digunakan untuk fitur seperti "Memory Entry" (ME) dan "Memory Clear" (MC). Ini membuat program lebih interaktif dan fleksibel.

**"Jelaskan bagian-bagian tampilan (UI) dari aplikasi kalkulator ini dan hasil eksekusinya ketika dijalankan!"**

## ✅ Contoh Jawaban yang Bagus dan Lengkap:

### 1. ****Identifikasi UI (User Interface) – Rancangan Tampilan****

Tampilan kalkulator ini dirancang menggunakan **Bootstrap**, dan terdiri dari beberapa elemen:

* **Logo**:
  + Berada di bagian atas, menggunakan elemen <img> dengan class logo.
* **Judul**:
  + <h2 class="text-center">Kalkulator Sederhana</h2>
* **Form Input**:
  + **Angka Pertama**: <input type="text" name="angka1" ...>
  + **Angka Kedua**: <input type="text" name="angka2" ...>
* **Tombol Operator**:
  + Terdapat 4 tombol untuk operasi dasar:
    - Tambah + (btn-primary)
    - Kurang - (btn-secondary)
    - Kali \* (btn-success)
    - Bagi / (btn-info)
  + Menggunakan ikon dari **Font Awesome**
* **Tombol CE (Clear Entry)**:
  + <button type="reset" ... class="btn btn-warning">CE</button>
* **Tombol ME / MC**:
  + ME (Memory Entry): Menyimpan hasil ke input
  + MC (Memory Clear): Menghapus hasil/riwayat
* **Output Hasil**:
  + Ditampilkan di bawah tombol dalam elemen <h4>:

php

CopyEdit

echo "$angka1 $operator $angka2 = $hasil";

* **Riwayat Perhitungan** (kalau fitur ini ditambahkan):
  + Menggunakan <ul> untuk menampilkan $\_SESSION['history']

### 2. ****Identifikasi Hasil Eksekusi****

Saat pengguna mengisi angka dan memilih operator:

* Program akan memproses input, lalu menampilkan hasil perhitungan.
* Contoh:
  + Input: angka1 = 5, angka2 = 2, operator = \*
  + Output: 5 \* 2 = 10

Jika input tidak valid:

* Akan muncul alert() menggunakan JavaScript:
  + "Input harus berupa angka"
  + "Tidak dapat membagi dengan nol"

## 🎯 Ringkasan

"UI pada aplikasi ini terdiri dari form input, tombol operator, dan hasil output. Setiap elemen punya fungsi interaktif yang mendukung proses perhitungan, dan hasil eksekusi akan langsung ditampilkan ke layar atau disimpan sebagai riwayat."

## Penjelasan Baris per Baris:

| **Baris** | **Penjelasan** |
| --- | --- |
| if (isset($\_POST['operator'])) | Mengecek apakah tombol operator ditekan. |
| $angka1 = $\_POST['angka1']; | Mengambil input angka pertama dari form. |
| $angka2 = $\_POST['angka2']; | Mengambil input angka kedua dari form. |
| $operator = $\_POST['operator']; | Mengambil operator (+, -, \*, /) dari tombol yang ditekan. |
| str\_replace(',', '.', ...) | Mengubah koma jadi titik supaya bisa diproses sebagai desimal. |
| if (!is\_numeric(...)) | Validasi: pastikan input berupa angka. |
| elseif ($operator == '/' && $angka2 == 0) | Validasi: tidak boleh membagi dengan nol. |
| switch($operator) | Mengeksekusi operasi berdasarkan operator. |
| case '+': ... | Operasi penjumlahan. |
| case '-': ... | Operasi pengurangan. |
| case '\*': ... | Operasi perkalian. |
| case '/': ... | Operasi pembagian. |
| default: | Jika operator tidak dikenal. |
| echo "$angka1 $operator $angka2 = $hasil"; | Menampilkan hasil perhitungan. |
| else { echo "Hasil : "; } | Default tampilan awal saat belum ada perhitungan. |

### ****1. Apa fungsi dari method POST di form?****

**Jawaban:** Method POST digunakan untuk mengirim data input dari form ke server secara tersembunyi (tidak terlihat di URL). Di sini, digunakan untuk mengirim nilai angka1, angka2, dan operator ke file PHP untuk diproses perhitungannya.

### ****2. Mengapa input angka perlu divalidasi?****

**Jawaban:** Untuk mencegah error saat proses perhitungan. Misalnya, jika user memasukkan huruf, atau membagi dengan nol. Validasi memastikan input hanya berupa angka yang valid.

### ****3. Jelaskan penggunaan**** isset() ****di PHP ini.****

**Jawaban:** isset() digunakan untuk memeriksa apakah suatu variabel telah diset (diisi). Misalnya isset($\_POST['operator']) mengecek apakah tombol operator sudah diklik, agar kalkulasi baru dijalankan setelah ada input.

### ****4. Mengapa digunakan**** str\_replace(',', '.', ...)****?****

**Jawaban:** Untuk mengantisipasi pengguna yang menggunakan tanda koma , sebagai desimal (contoh: 3,14). PHP hanya mengenali titik . sebagai desimal, jadi harus diubah agar tidak error.

### ****5. Jelaskan struktur logika program ini!****

**Jawaban:**

* User memasukkan dua angka.
* Memilih operator.
* Server menerima input via POST.
* Validasi dilakukan.
* Perhitungan dilakukan menggunakan switch.
* Hasil ditampilkan di layar.

### ****6. Apa manfaat menggunakan**** switch ****dalam perhitungan?****

**Jawaban:** Switch membuat kode lebih terstruktur dan mudah dibaca saat memproses kondisi berdasarkan operator. Lebih rapi dibanding if-else berulang.

### ****7. Apa perbedaan tombol**** CE****,**** ME****, dan**** MC****?****

**Jawaban:**

* **CE (Clear Entry)**: Menghapus input saat ini.
* **ME (Memory Entry)**: Menyimpan hasil ke input untuk digunakan lagi.
* **MC (Memory Clear)**: Menghapus hasil yang tersimpan sebelumnya.

### ****8. Apa tujuan penggunaan session (jika kamu pakai history)?****

**Jawaban:** Session ($\_SESSION) digunakan untuk menyimpan riwayat perhitungan sementara, selama browser belum ditutup. Ini berguna untuk fitur memori atau history.

### ****9. Bagaimana kamu menghindari error saat pembagian dengan nol?****

**Jawaban:** Dengan menambahkan pengecekan khusus:

php

CopyEdit

if ($operator == '/' && $angka2 == 0) {

alert("Tidak bisa dibagi 0");

}

Hal ini mencegah sistem crash dan memberi notifikasi ke user.

### ****10. Kalau ingin ubah ke OOP (Object Oriented Programming), apa yang dilakukan?****

**Jawaban:** Saya akan membuat **class Kalkulator**, lalu buat method tambah(), kurang(), kali(), dan bagi(). Lalu instance objek digunakan untuk memanggil method tersebut. Ini membuat kode lebih terstruktur dan scalable.

CATATAN PENJELASAN PROYEK UKK RPL 2025 - KALKULATOR SEDERHANA

1. Deskripsi Umum:

Aplikasi ini adalah kalkulator sederhana berbasis web yang menggunakan HTML, CSS (Bootstrap),

JavaScript untuk validasi input, dan PHP sebagai backend untuk menghitung operasi matematika.

2. Alur Kerja Program:

- User menginput dua angka ke form HTML.

- User memilih operator (tambah, kurang, kali, bagi).

- Data dikirim melalui method POST ke server.

- PHP menerima data, memvalidasi, lalu menghitung sesuai operator.

- Hasil ditampilkan ke user, dengan opsi ME (memory entry) dan MC (memory clear).

3. Struktur Data:

- Data input berupa string (karena dari form), lalu dikonversi menjadi float menggunakan str\_replace

(mengubah koma ke titik) dan fungsi aritmatika.

- Variabel yang digunakan antara lain: $angka1, $angka2, $operator, $hasil.

4. Validasi & Error Handling:

- JavaScript membatasi input hanya angka, titik, atau koma.

- PHP memvalidasi ulang agar tidak terjadi pembagian nol atau input non-numerik.

- Alert ditampilkan menggunakan JavaScript jika error ditemukan.

5. Tampilan (User Interface):

- Menggunakan Bootstrap agar responsif dan rapi.

- Menggunakan ikon dari Font Awesome untuk tombol operator.

- Logo sekolah ditampilkan di bagian atas untuk branding.

6. Fitur Tombol:

- Tombol operator (+, -, \*, /) untuk menghitung.

- CE (Clear Entry): Reset form input.

- ME (Memory Entry): Menyimpan hasil ke input pertama.

- MC (Memory Clear): Menghapus hasil perhitungan sebelumnya.

7. Tools yang Digunakan:

- Bootstrap CDN: Untuk tampilan responsif.

- Font Awesome: Untuk ikon tombol.

- JavaScript: Validasi input real-time.

- PHP: Backend kalkulasi dan validasi hasil.

8. Coding Guidelines & Best Practices:

- Menggunakan pemisahan antara tampilan (HTML), logika (PHP), dan validasi (JS).

- Penggunaan isset() dan str\_replace() untuk keamanan dan keakuratan.

- Switch-case untuk mempermudah proses logika operator.

- Menghindari hardcode, dan menangani error dengan alert.

9. Security Aspek (Dasar):

- Validasi dua sisi: client-side (JS) & server-side (PHP).

- Cek numeric dengan is\_numeric().

- Mencegah pembagian nol dengan pengecekan kondisi.

10. Catatan Tambahan:

- Tidak menggunakan database karena ini aplikasi sederhana.

- Dapat dikembangkan lebih lanjut menjadi kalkulator ilmiah atau dengan riwayat.

**anjutan catatan ini menjelaskan praktik terbaik (best practices), penanganan error (error handling), dan**

**aspek keamanan (security) dalam kode kalkulator sederhana. Ini penting untuk menjawab pertanyaan**

**penguji secara teknis dan profesional.**

**1. Best Practices (Praktik Terbaik)**

**Beberapa praktik terbaik yang diterapkan dalam kode:**

**- Menggunakan `required` di input form agar tidak kosong.**

**- Validasi input dengan JavaScript dan PHP.**

**- Menggunakan `switch-case` untuk perhitungan lebih terstruktur daripada if-else panjang.**

**- Struktur kode HTML rapi dengan Bootstrap.**

**- Menggunakan `isset()` untuk menghindari error saat POST belum dikirim.**

**Jawaban ujian:**

**"Saya terapkan best practice seperti validasi input, pakai switch-case biar rapi, dan selalu cek data pakai**

**isset sebelum dipakai."**

**2. Error Handling**

**Penanganan error yang ada di kode:**

**- Validasi angka: jika input bukan angka, muncul alert dari JavaScript atau PHP.**

**- Pembagian dengan nol: dicek dengan kondisi `if ($angka2 == 0)` dan tampilkan alert.**

**- Tombol reset (CE dan MC) membantu menghindari user menginput data lama.**

**Jawaban ujian:**

**"Ada validasi angka dan pembagian nol. Kalau bukan angka atau angka kedua nol saat bagi, akan muncul**

**peringatan supaya tidak crash."**

SOAL & JAWABAN YANG BIASANYA DITANYAKAN PENGUJI UKK  
1. Apa fungsi dari baris ini?  
<input type="text" name="angka1" class="form-control" required  
oninput="validateNumberInput(this)">  
Baris ini membuat input form untuk 'Angka Pertama'.  
- type="text" -> Input teks  
- name="angka1" -> Dikirim ke PHP pakai $\_POST['angka1']  
- oninput="validateNumberInput(this)" -> Memanggil fungsi JS untuk validasi angka  
2. Jelaskan fungsi JavaScript ini:  
function validateNumberInput(input) {  
input.value = input.value.replace(/[^0-9,\.]/g, '');  
input.value = input.value.replace(',', '.');  
}  
Fungsi ini mencegah karakter selain angka, titik, dan koma dimasukkan oleh user. Lalu koma diganti  
titik agar bisa dihitung sebagai desimal di PHP.  
3. Apa fungsi dari baris berikut?  
if (isset($\_POST['operator'])) {  
Untuk mengecek apakah form sudah disubmit dengan salah satu operator (tambah, kurang, kali,  
bagi).  
4. Apa tujuan dari ini?  
$angka1 = str\_replace(',', '.', $angka1);  
Mengganti koma menjadi titik agar PHP bisa membaca nilai desimal.  
5. Bagaimana kamu menangani error seperti pembagian dengan nol?  
elseif ($operator == '/' && $angka2 == 0) {  
echo "<script>alert('Tidak dapat membagi dengan Nol')</script>";

Baris ini mencegah aplikasi crash jika user membagi angka dengan 0.  
6. Kenapa kamu pakai switch-case daripada if-else untuk operator?  
switch ($operator) {  
case '+': $hasil = $angka1 + $angka2; break;  
...  
}  
Switch-case lebih rapi dan efisien untuk banyak kondisi dari 1 variabel.  
7. Apa maksud dari ini?  
<input type="hidden" name="hasil" value="<?php echo $hasil ?>">  
Untuk menyimpan hasil ke dalam form secara tersembunyi.  
8. Kenapa pakai isset()?  
<?php echo isset($\_POST['hasil']) ? $\_POST['hasil'] : '' ?>  
Untuk mencegah error jika variabel belum ada nilainya.  
9. Kenapa pakai method POST, bukan GET?  
POST lebih aman - data tidak terlihat di URL.  
10. Apa yang terjadi saat user klik tombol CE?  
<button type="reset" name="reset" class="btn btn-warning">CE</button>  
Form akan dikosongkan (di-reset).  
11. Apa fungsi dari ME dan MC di aplikasi kamu?  
- ME (Memory Entry): Menyimpan hasil ke input angka pertama.  
- MC (Memory Clear): Menghapus hasil terakhir.  
12. Kenapa logo sekolah kamu masukkan di bagian atas halaman?  
Sebagai branding dan mempercantik UI aplikasi.  
13. Apa yang terjadi kalau user input huruf ke input angka?  
Akan difilter oleh JavaScript dan tidak diterima oleh PHP juga

#### ****1. Struktur Data yang Digunakan:****

| **Struktur Data** | **Penjelasan** |
| --- | --- |
| **Array ($\_POST)** | Digunakan untuk menyimpan data yang dikirim dari form HTML. Contohnya: $\_POST['angka1'], $\_POST['operator']. Ini adalah array asosiatif. |
| **String** | Input dari user awalnya berupa string. Misalnya $\_POST['angka1'] adalah string, meskipun terlihat angka. |
| **Primitive (Scalar)** | Setelah validasi dan konversi, data string diubah menjadi **tipe data numerik** (float atau integer), yang termasuk struktur data sederhana (primitive). |
| **Switch-case (control structure)** | Ini bukan struktur data, tapi bagian dari struktur kendali logika dalam program untuk memilih perhitungan berdasarkan operator. |

#### 📌 ****2. Tipe Data yang Digunakan:****

| **Tipe Data** | **Digunakan untuk** |
| --- | --- |
| **String** | Menyimpan input awal dari user: $\_POST['angka1'] dan $\_POST['angka2']. |
| **Float / Double** | Setelah string diubah (str\_replace()), tipe data ini digunakan untuk menyimpan angka desimal seperti 3.14. |
| **Integer** | Untuk angka tanpa desimal (misalnya: 3, 7). PHP bisa otomatis mengubah dari string ke int jika tidak ada titik. |
| **Boolean** | Dalam validasi, misalnya isset() mengembalikan nilai boolean true atau false. |
| **Null (kadang)** | Jika suatu variabel belum di-set, dia akan bernilai null. Ini terjadi kalau misalnya $\_POST['hasil'] belum dikirim. |