

PEMBAHASAN KODE SCRIPT.JS (JAVASCRIPT)

Fungsi `appendToDisplay(value)`

Fungsi `appendToDisplay(value)` digunakan untuk menambahkan `nilai` (angka atau operator) ke tampilan kalkulator yang ditampilkan di elemen input dengan `id="display"`.

Penjelasan Fungsi:

```
function appendToDisplay(value) {  
    document.getElementById('display').value += value;  
}
```

- `value`:
Parameter yang diterima oleh fungsi ini adalah nilai yang ingin ditambahkan ke tampilan. Ini bisa berupa angka atau `operator` (seperti `+`, `-`, `*`, dll.), yang berasal dari tombol yang ditekan oleh pengguna.
- `document.getElementById('display')`:
Ini digunakan untuk memilih elemen dengan `id="display"`, yang dalam hal ini adalah elemen `input` (di HTML `<input type="text" class="display" id="display" disabled>`). Elemen input ini menampilkan hasil atau ekspresi matematika yang sedang diketik oleh pengguna.

- `.value`:
Mengakses properti `.value` dari elemen input memungkinkan kita untuk membaca atau mengubah teks yang ada di dalam input tersebut. Pada awalnya, nilai `value` ini bisa kosong atau berisi angka/hasil perhitungan sebelumnya.
- `+=`:
Operator ini digunakan untuk menambahkan nilai yang ada ke nilai yang sudah ada sebelumnya. Jika nilai `value` yang diterima adalah angka `5`, maka angka `5` akan ditambahkan ke nilai yang sudah ada di display `.value`. Misalnya, jika sebelumnya ada `3`, setelah menekan tombol `5`, hasilnya akan menjadi `35`.

Contoh Penggunaan:

Jika tombol yang ditekan adalah `7`, maka `appendToDisplay('7')` akan menambahkan angka `7` ke dalam input. Jika sebelumnya input sudah berisi angka `5`, maka setelah `appendToDisplay('7')`, nilai tampilan akan menjadi `57`.

Inti Fungsinya:

Fungsi ini memungkinkan pengguna untuk mengklik tombol angka atau operator pada kalkulator dan secara langsung menambahkannya ke tampilan kalkulator tanpa menghapus nilai yang sudah ada di sana.

Fungsi `clearDisplay()`

Fungsi `clearDisplay()` digunakan untuk menghapus nilai atau teks yang sedang ditampilkan di kalkulator. Dalam konteks kalkulator ini, fungsi ini akan membersihkan tampilan pada elemen input yang memiliki `id="display"`, yang merupakan tempat hasil perhitungan atau input pengguna ditampilkan.

Penjelasan Fungsi:

```
function clearDisplay() {  
    document.getElementById('display').value = '';  
}
```

- `document.getElementById('display')`:
Ini digunakan untuk memilih elemen HTML dengan `id="display"`. Elemen ini adalah input field (seperti `<input type="text" id="display">`) tempat hasil atau ekspresi matematika ditampilkan.
- `.value`:
Properti `.value` digunakan untuk mengakses atau mengubah `nilai` (isi) dari elemen input. Ketika kita mengubah nilai `value` dari elemen input, itu akan memperbarui apa yang dilihat oleh pengguna di kalkulator.
- `= ''`:
Menetapkan nilai `value` menjadi string `kosong` (`''`). Ini menghapus semua teks yang ada di dalam input `display`, yang berarti tampilan kalkulator akan kosong setelah fungsi ini dipanggil.

Contoh Penggunaan:

Jika tombol "C" pada kalkulator ditekan, fungsi `clearDisplay()` akan dijalankan. Sebelumnya jika tampilan kalkulator menunjukkan nilai `123+45`, setelah fungsi ini dijalankan, nilai tampilan akan dihapus dan input kalkulator akan menjadi kosong.

Tujuan:

Fungsi ini berfungsi untuk menghapus input atau hasil sebelumnya, memberikan pengguna kesempatan untuk memulai perhitungan baru tanpa harus merefresh halaman atau menekan tombol satu per satu untuk menghapus input. Biasanya, tombol "C" (Clear) di kalkulator akan memanggil fungsi ini untuk mereset kalkulator.

Fungsi `calculateResult()`

Fungsi `calculateResult()` adalah fungsi JavaScript yang digunakan untuk menghitung ekspresi matematika yang dimasukkan ke dalam elemen input dengan ID `"display"`. Fungsi ini menangani beberapa operator khusus, seperti pangkat (^) dan pembagian bulat (//), serta menghindari kesalahan perhitungan dengan menggunakan try-catch.

Kode Lengkap

```
function calculateResult() {  
    try {  
        let expression = document.getElementById('display').value  
            .replace(/\^/g, '**') // Ubah ^ menjadi ** untuk pangkat  
            .replace(/\//g, '/'); // Ubah // menjadi /  
  
        let result = new Function('return ' + expression)();  
        if (document.getElementById('display').value.includes('//')) {  
            result = Math.floor(result); // Jika "/" digunakan, buat hasilnya menjadi integer  
        }  
        document.getElementById('display').value = result;  
    } catch {  
        document.getElementById('display').value = 'Error';  
    }  
}
```

Penjelasan Detail Kode

```
function calculateResult() {  
  try {  
    let expression = document.getElementById('display').value  
      .replace(/\^/g, '**') // Ubah ^ menjadi ** untuk operasi pangkat  
      .replace(/\//g, '/'); // Ubah // menjadi / untuk operasi pembagian
```

- Mengambil ekspresi matematika yang ada di dalam elemen input dengan `id="display"`.
- Mengganti simbol `^` dengan `**` karena JavaScript menggunakan `**` untuk operasi pangkat, sedangkan `^` adalah operator bitwise `XOR`.
- Mengganti `//` dengan `/` karena JavaScript tidak memiliki operator pembagian bulat `//` seperti di Python. Namun, ada bagian kode yang akan menangani ini setelah hasil perhitungan.

```
let result = new Function('return ' + expression)();
```

- Menggunakan `new Function()` untuk mengeksekusi ekspresi matematika.
 - `new Function('return ' + expression)()` membuat fungsi baru yang mengembalikan hasil ekspresi yang diberikan.
 - Ini mirip dengan `eval()`, tetapi lebih aman karena hanya mengevaluasi ekspresi yang diberikan sebagai string.

```
if (document.getElementById('display').value.includes('/')) {  
    result = Math.floor(result); // Jika "/" digunakan, buat hasilnya menjadi  
integer  
}
```

- Menangani pembagian bulat (/):
Jika input yang diketik pengguna mengandung "/", hasil pembagian akan dibulatkan ke bawah menggunakan `Math.floor()`.

```
document.getElementById('display').value = result;  
} catch {  
    document.getElementById('display').value = 'Error';  
}  
}
```

- Menampilkan hasil ke dalam elemen input `"display"`.
- Menangani kesalahan dengan try-catch:
- Jika ada kesalahan saat menghitung `ekspresi` (misalnya pengguna memasukkan input yang tidak valid), maka tampilan akan diubah menjadi `"Error"`.

Contoh Implementasi dalam HTML

```
<input type="text" id="display" readonly>
<button onclick="appendToDisplay('2')">2</button>
<button onclick="appendToDisplay('+')">+</button>
<button onclick="appendToDisplay('3')">3</button>
<button onclick="calculateResult()">=</button>
```

Contoh Penggunaannya

Input	Hasil
2+3	5
2^3	8 (karena ^ diubah menjadi **)
7//2	3 (pembagian bulat, karena Math.floor(3.5))

Fungsi `square()`

Fungsi `square()` adalah fungsi JavaScript yang digunakan untuk menghitung kuadrat dari nilai yang ada di elemen input dengan ID `"display"`.

Kode Lengkap

```
function square() {  
    let display = document.getElementById('display');  
    display.value = Math.pow(parseFloat(display.value), 2);  
}
```

Penjelasan Kode

```
function square() {  
    let display = document.getElementById('display');
```

- Mengambil elemen input dengan `id="display"` yang digunakan sebagai tempat menampilkan angka.

```
display.value = Math.pow(parseFloat(display.value), 2);
```

- Mengambil nilai dari input, lalu mengubahnya menjadi tipe angka menggunakan `parseFloat()`.

- Menghitung kuadrat dari nilai tersebut dengan fungsi `Math.pow(x, 2)`, yang artinya x^2 .
- Memasukkan hasil kuadrat kembali ke dalam input sehingga tampilan diperbarui.

Contoh Implementasi dalam HTML

```
<input type="text" id="display" value="5" readonly>  
<button onclick="square()">x2</button>
```

Cara Kerja

- Jika pengguna memasukkan angka **5** ke dalam input.
- Saat tombol x^2 ditekan, fungsi `square()` akan mengubah 5 menjadi $5^2 = 25$.
- Input akan diperbarui menjadi **25**.

Penjelasan Fungsi `powerN()`

Fungsi `powerN()` adalah fungsi JavaScript yang digunakan untuk menambahkan simbol `"^"` (pangkat) ke dalam elemen input dengan ID `"display"`. Simbol ini nantinya bisa digunakan oleh fungsi lain untuk melakukan perhitungan pangkat.

Kode Lengkap dan Penjelasanannya

```
function powerN() {  
    let display = document.getElementById('display');  
    display.value += '^'; // Menambahkan simbol ^ agar bisa digunakan sebagai pangkat  
}
```

- Mengambil elemen input dengan `id="display"` untuk mendapatkan nilai yang sedang ditampilkan.
- Menambahkan simbol `"^"` ke dalam nilai input tanpa menghapus angka yang sudah ada sebelumnya.
- Misalnya, jika tampilan saat ini adalah `"2"`, setelah fungsi ini dipanggil, tampilan akan menjadi `"2^"`.

Contoh Implementasi dalam HTML

```

<input type="text" id="display" readonly>
<button onclick="appendToDisplay('2')">2</button>
<button onclick="powerN()">^</button>
<button onclick="appendToDisplay('3')">3</button>
<button onclick="calculateResult()">=</button>

```

- Jika pengguna menekan 2, lalu ^, lalu 3, tampilan menjadi "2^3".
- Saat tombol = ditekan, fungsi `calculateResult()` harus menangani ^ dan mengubahnya menjadi ** agar JavaScript dapat menghitung hasilnya.

Fungsi Pendukung (`calculateResult()`)

Agar fungsi `powerN()` bisa bekerja dengan benar dalam kalkulator, kita bisa menggunakan fungsi `calculateResult()` yang mengganti ^ dengan ** (operator pangkat di JavaScript):

```

function calculateResult() {
  try {
    let expression = document.getElementById('display').value
      .replace(/\^/g, '**'); // Mengubah ^ menjadi ** untuk operasi pangkat

    let result = new Function('return ' + expression)();
    document.getElementById('display').value = result;
  } catch {
    document.getElementById('display').value = 'Error';
  }
}

```

Hasil yang diharapkan

Input	Tampilan Setelah powerN()	Hasil Setelah =
2 ^ 3	"2^3"	8
5 ^ 2	"5^2"	25

Fungsi `powerN()` hanya menambahkan simbol "^" ke dalam input, tidak melakukan perhitungan langsung. Untuk menyelesaikan operasi pangkat, diperlukan fungsi lain yang menangani konversi ^ ke ** agar JavaScript dapat mengolahnya dengan benar.

Penjelasan Fungsi `squareRoot()`

Fungsi `squareRoot()` adalah fungsi JavaScript yang digunakan untuk menghitung akar kuadrat dari nilai yang ada di elemen input dengan ID `"display"`.

Kode Lengkap

```
function squareRoot() {  
    let display = document.getElementById('display');  
    display.value = Math.sqrt(parseFloat(display.value));  
}
```

Penjelasan Detail Kode

```
function squareRoot() {  
    let display = document.getElementById('display');
```

- Mengambil elemen input dengan `id="display"` yang berisi angka yang ingin dihitung akar kuadratnya.

```
display.value = Math.sqrt(parseFloat(display.value));
```

- Mengubah nilai input menjadi angka dengan `parseFloat(display.value)`, karena input dari elemen HTML berbentuk string.
- Menggunakan `Math.sqrt()` untuk menghitung akar kuadrat dari angka tersebut.
- Menampilkan hasil perhitungan di elemen input.

Contoh Implementasi dalam HTML

```
<input type="text" id="display" value="9" readonly>  
<button onclick="squareRoot()">√</button>
```

Cara Kerjanya

- Jika pengguna memasukkan angka **9** ke dalam input.
- Saat tombol √ ditekan, fungsi `squareRoot()` akan mengubah 9 menjadi $\sqrt{9} = 3$.
- Input akan diperbarui menjadi **3**.

Hasil yang diharapkan

Input	Hasil
9	3
16	4
2.25	1.5

Penjelasan Fungsi `percentage()`

Fungsi `percentage()` adalah fungsi JavaScript yang digunakan untuk mengubah angka dalam input menjadi bentuk persen dengan cara membagi angka tersebut dengan `100`.

Kode Lengkap

```
function percentage() {  
    let display = document.getElementById('display');  
    display.value = parseFloat(display.value) / 100;  
}
```

Penjelasan Detail Kode

```
function percentage() {  
    let display = document.getElementById('display');
```

- Mengambil elemen input dengan `id="display"`, yang berisi angka yang ingin dikonversi menjadi persen.

```
display.value = parseFloat(display.value) / 100;
```

- Mengonversi nilai input dari string menjadi angka menggunakan `parseFloat(display.value)`, karena nilai dari elemen input biasanya berbentuk string.
- Membagi angka dengan `100` untuk mendapatkan bentuk persen dalam desimal.
- Mengupdate kembali nilai input dengan hasil perhitungan.

Contoh Implementasi dalam HTML

```
<input type="text" id="display" value="50" readonly>  
<button onclick="percentage()">%</button>
```

Cara Kerja

- Jika pengguna memasukkan angka **50** ke dalam input.
- Saat tombol % ditekan, fungsi **percentage()** akan mengubah 50 menjadi **50 / 100 = 0.5**.
- Input akan diperbarui menjadi **0.5**.

Hasil yang diharapkan

Input	Hasil Setelah %
50	0.5
25	0.25
200	2

Penjelasan Fungsi `modulus()`

Fungsi `modulus()` adalah fungsi JavaScript yang digunakan untuk menambahkan simbol "%" ke dalam elemen input dengan ID `"display"`. Simbol ini nantinya dapat digunakan untuk operasi `modulus` dalam perhitungan.

Penjelasan Detail Kode

```
function modulus() {  
    let display = document.getElementById('display');  
    display.value += '%'; // Menambahkan simbol % ke input  
}
```

- Mengambil elemen input dengan `id="display"`, yang digunakan sebagai tempat menampilkan angka.
- Menambahkan simbol "%" ke dalam nilai input tanpa menghapus angka yang sudah ada sebelumnya.
- Misalnya, jika tampilan saat ini adalah `"10"`, setelah fungsi ini dipanggil, tampilan akan menjadi `"10%"`.

Fungsi `modulus()` dalam Kalkulator

Dalam JavaScript, operator `modulus (%)` digunakan untuk mendapatkan sisa hasil bagi dari suatu pembagian. Namun, dalam matematika umum, simbol % sering digunakan untuk persentase.

Karena fungsi ini hanya menambahkan "%" ke tampilan, perlu ada fungsi tambahan yang akan menangani perhitungan **modulus** saat tombol = ditekan.

Cara menggunakan dalam HTML

```
<input type="text" id="display" readonly>
<button onclick="appendToDisplay('10')">10</button>
<button onclick="modulus()">%</button>
<button onclick="appendToDisplay('3')">3</button>
<button onclick="calculateResult()">=</button>
```

- Jika pengguna menekan 10, lalu %, lalu 3, tampilan menjadi **"10%3"**.
- Saat tombol = ditekan, **fungsi calculateResult()** harus menangani % agar dikonversi ke operasi modulus (% dalam JavaScript).

Menggunakan calculateResult() untuk Menghitung Modulus

Agar simbol % bisa dihitung sebagai operasi modulus, kita perlu menyesuaikan fungsi `calculateResult()`:

```
function calculateResult() {
  try {
    let expression = document.getElementById('display').value
      .replace(/%/g, '%'); // Memastikan % tetap sebagai operator modulus

    let result = new Function('return ' + expression)();
    document.getElementById('display').value = result;
  } catch {
    document.getElementById('display').value = 'Error';
  }
}
```

Sekarang, jika pengguna mengetik **10%3**, hasilnya akan menjadi **1**, karena **10 % 3 = 1** (sisanya dari 10 dibagi 3).

Hasil yang diharapkan

Input	Tampilan Setelah <code>modulus()</code>	Hasil Setelah <code>=</code>
10 % 3	"10%3"	1
15 % 4	"15%4"	3

Fungsi `integerDivision()`

Fungsi `integerDivision()` adalah fungsi JavaScript yang digunakan untuk menambahkan simbol `"//"` ke dalam elemen input dengan ID `"display"`. Simbol ini nantinya akan digunakan untuk melakukan pembagian bulat (integer division).

Penjelasan Detail Kode

```
function integerDivision() {  
    let display = document.getElementById('display');  
    display.value += '//'; // Menambahkan "//" ke input  
}
```

- Mengambil elemen input dengan `id="display"`, yang digunakan untuk menampilkan angka dalam kalkulator.
- Menambahkan simbol `"//"` ke dalam nilai input tanpa menghapus angka yang sudah ada sebelumnya.
- Misalnya, jika tampilan saat ini adalah `"10"`, setelah fungsi ini dipanggil, tampilan akan menjadi `"10//"`.

Apa Itu Pembagian Bulat (`//`)?

- Dalam bahasa pemrograman seperti **Python**, operator `//` digunakan untuk **pembagian bulat**, yaitu pembagian yang hasilnya selalu berupa **bilangan bulat** (dibulatkan ke bawah).

Contoh dalam bahasa pemrograman python

```
10 // 3 # Hasil: 3
15 // 4 # Hasil: 3
```

- Namun, JavaScript tidak memiliki operator `//` secara langsung. Oleh karena itu, kita harus menggantinya dengan `Math.floor(a / b)` agar hasilnya tetap bilangan bulat

Fungsi Pendukung: calculateResult()

Agar `//` dapat digunakan sebagai operator pembagian bulat dalam kalkulator, kita harus menyesuaikan fungsi `calculateResult()`:

```
function calculateResult() {
  try {
    let expression = document.getElementById('display').value
      .replace(/\^/g, '**') // Mengubah ^ menjadi ** untuk operasi pangkat
      .replace(/\/\//g, '/'); // Sementara mengganti "//" menjadi "/"

    let result = new Function('return ' + expression)();

    // Jika input mengandung "//", ubah hasilnya menjadi pembagian bulat
    if (document.getElementById('display').value.includes('//')) {
      result = Math.floor(result); // Membulatkan hasil ke bawah
    }

    document.getElementById('display').value = result;
  } catch {
    document.getElementById('display').value = 'Error';
  }
}
```

Bagaimana Cara Kerjanya?

- Mengubah "//" menjadi "/" sementara agar JavaScript dapat mengolahnya.
- Menghitung hasil ekspresi menggunakan **new Function()**.
- Jika input mengandung "//", hasilnya dibulatkan ke bawah dengan **Math.floor(result)**.

Contoh Implementasi dalam HTML

```
<input type="text" id="display" readonly>
<button onclick="appendToDisplay('10')">10</button>
<button onclick="integerDivision()">//</button>
<button onclick="appendToDisplay('3')">3</button>
<button onclick="calculateResult()">=</button>
```

Cara Kerja

1. Jika pengguna menekan **10**, lalu **//**, lalu **3**, tampilan menjadi "10//3".
2. Saat tombol = ditekan, fungsi `calculateResult()` mengubah "10//3" menjadi **Math.floor(10 / 3) = 3**.
3. Input akan diperbarui menjadi **3**.

Hasil yang diharapkan

Input	Tampilan Setelah integerDivision()	Hasil Setelah =
10 // 3	"10//3"	3
15 // 4	"15//4"	3
9 // 2	"9//2"	4