

LAPORAN DOKUMENTASI TUGAS SOLVER
DESAIN ANALISIS & ALGORITMA



DISUSUN OLEH:

1. Mario Valentino Ardhana (L0123079)
2. Nabil Riano Zaky (L0123103)

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET

2024

Pendahuluan

Program 20 Solver adalah aplikasi berbasis web yang memungkinkan pengguna memasukkan empat angka, lalu menghitung dan menampilkan semua ekspresi matematika yang dapat dibentuk menggunakan angka tersebut dan operator aritmatika dasar (+, -, *, /). Program ini bertujuan untuk menemukan kombinasi ekspresi yang menghasilkan nilai 20. Berikut adalah penjelasan detail dari setiap bagian program beserta analisis berdasarkan baris kode.

Struktur Kode

1. HTML (Baris 1-114)

```
<!DOCTYPE html>
<html lang="en">
```

Baris ini mendefinisikan bahwa dokumen ini adalah dokumen HTML5 dan menetapkan bahasa dokumen menjadi "en" (bahasa Inggris).

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>20 Solver</title>
  <style>
```

Pada bagian <head>, dokumen ini mengatur metadata, termasuk karakter yang digunakan (UTF-8), membuat dokumen responsif pada perangkat mobile dengan tag viewport, dan memberikan judul "20 Solver" pada halaman.

```
body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 20px;
  background-color: #f9f6f2; /* Light beige */
  color: #3e2723; /* Dark brown */
}
h2 {
  text-align: center;
  color: #5d4037; /* Medium brown */
}
input {
```

```

        width: 50px;
        text-align: center;
        border: 2px solid #5d4037; /* Medium brown */
        border-radius: 5px;
        padding: 5px;
        margin: 5px;
    }
    table {
        width: 100%;
        border: 1px solid #3e2723; /* Dark brown */
        border-collapse: collapse;
        margin-top: 20px;
    }
    th, td {
        border: 1px solid #3e2723; /* Dark brown */
        padding: 10px;
        text-align: left;
    }
    th {
        background-color: #8d6e63; /* Light brown */
        color: white;
    }
    button {
        background-color: #5d4037; /* Medium brown */
        color: white;
        border: none;
        border-radius: 5px;
        padding: 10px;
        font-size: 16px;
        cursor: pointer;
        margin-top: 10px;
        display: block;
        width: 100%;
    }
    button:hover {
        background-color: #3e2723; /* Darker brown */
    }
    #noSolution {
        color: red;
        display: none;
    }
    #jumlahSolusi {
        font-weight: bold;
        text-align: center;
    }
</style>
</head>
<body>

```

Lines 8-67: Bagian ini berisi definisi CSS untuk mengatur tampilan halaman. Berikut adalah beberapa komponen utama:

- Lines 8-14 : Mengatur tampilan dasar seperti font, margin, padding, warna latar belakang, dan warna teks.
- Lines 15-18: Mengatur tampilan judul dengan warna tertentu dan perataan tengah.
- Lines 16-26: Mendefinisikan gaya input angka, dengan ukuran tertentu dan penataan agar terlihat rapi.
- Lines 27-42 : Mengatur tampilan tabel, dengan lebar penuh, border untuk tabel, dan style untuk header (th) serta sel tabel (td).
- Lines 43-57 : Mengatur gaya tombol, termasuk warna, ukuran, dan efek hover.
- Lines 58-61 : Mengatur tampilan pesan jika tidak ada solusi dengan warna merah dan menyembunyikannya secara default.
- Lines 62-65 : Mengatur tampilan jumlah solusi dengan penekanan pada teks tebal.

```
<h2>20 Solver</h2>
```

Menampilkan judul utama halaman di bagian tengah.

```
<!-- Tabel untuk input 4 angka -->
<table>
  <tr>
    <th>Angka 1</th>
    <th>Angka 2</th>
    <th>Angka 3</th>
    <th>Angka 4</th>
  </tr>
  <tr>
    <td><input type="number" id="angka1"></td>
    <td><input type="number" id="angka2"></td>
    <td><input type="number" id="angka3"></td>
    <td><input type="number" id="angka4"></td>
  </tr>
</table>
```

Bagian ini membuat tabel input untuk memasukkan empat angka.

- Lines 73-78 : Mendefinisikan struktur tabel dengan empat kolom untuk input angka.
- Lines 79-84 : Membuat input untuk setiap angka (1 hingga 4) dengan tipe number.

```
<button onclick="solve()">Hitung Kemungkinan</button>
```

Tombol ini menjalankan fungsi solve() saat ditekan oleh pengguna. Fungsi ini bertanggung jawab untuk menghitung semua kemungkinan dari input angka.

```
<!-- Tabel untuk menampilkan hasil -->
<table id="resultTable" style="display:none;">
  <thead>
    <tr>
      <th>Ekspresi</th>
      <th>Hasil</th>
    </tr>
  </thead>
  <tbody id="resultBody"></tbody>
</table>

<!-- Tabel untuk menampilkan solusi -->
<table id="solutionTable" style="display:none;">
  <thead>
    <tr>
      <th>Solusi yang menghasilkan 20</th>
    </tr>
  </thead>
  <tbody id="solutionBody"></tbody>
</table>
```

- Lines 90-98 : Membuat tabel kosong yang akan menampilkan hasil kemungkinan ekspresi dan hasilnya. Tabel ini tersembunyi (display:none) dan hanya akan muncul saat ada hasil untuk ditampilkan.
- Lines 101-108 : Membuat tabel kosong untuk menampilkan solusi yang menghasilkan angka 20, yang juga tersembunyi dan akan ditampilkan hanya jika ada solusi.

```
<!-- Menampilkan jumlah solusi -->
<p id="jumlahSolusi" style="display:none;"></p>
```

```
<p id="noSolution" style="display:none;">Angka yang di input  
tidak akan menghasilkan nilai 20.</p>  
  
<script>
```

Bagian ini menampilkan pesan yang menunjukkan jumlah solusi atau pesan bahwa tidak ada solusi. Secara default, kedua elemen ini tersembunyi.

2. JavaScript (Baris 116-214)

```
// Fungsi untuk menghasilkan permutasi dari array  
function permutasi(arr) {  
    if (arr.length === 0) return [[]];  
    const result = [];  
    for (let i = 0; i < arr.length; i++) {  
        const rest = [...arr.slice(0, i), ...arr.slice(i +  
1)];  
        for (let perm of permutasi(rest)) {  
            result.push([arr[i], ...perm]);  
        }  
    }  
    return result;  
}
```

Fungsi `permutasi(arr)` bertugas untuk menghitung semua permutasi dari array `arr` yang berisi angka.

- Lines 119-120 : Jika array kosong, mengembalikan array kosong.
- Lines 121-126 : Melakukan iterasi pada array untuk mendapatkan permutasi. Array `rest` menyimpan semua elemen kecuali elemen saat ini, dan `result` menyimpan hasil akhir dari semua permutasi.

```
// Fungsi untuk menghitung semua kemungkinan ekspresi  
function hitungKemungkinan(angka) {  
    const operators = ['+', '-', '*', '/'];  
    const allExpressions = [];  
    const angkaPermutasi = permutasi(angka);  
  
    for (let perm of angkaPermutasi) {  
        for (let op1 of operators) {  
            for (let op2 of operators) {  
                for (let op3 of operators) {  
                    const expressions = [  

```

```

        `(${perm[0]} ${op1} ${perm[1]})
    ${op2} (${perm[2]} ${op3} ${perm[3]})`,
        `(${perm[0]} ${op1} (${perm[1]}
    ${op2} ${perm[2]})) ${op3} ${perm[3]}`,
        `(${perm[0]} ${op1} ((${perm[1]}
    ${op2} ${perm[2]}) ${op3} ${perm[3]})`,
        `(${perm[0]} ${op1} ${perm[1]})
    ${op2} ${perm[2]} ${op3} ${perm[3]}`,
        `(${perm[0]} ${op1} ${perm[1]} ${op2}
    ${perm[2]} ${op3} ${perm[3]}`
    ];

    for (let expr of expressions) {
        try {
            const hasil = eval(expr);
            if (isFinite(hasil)) {
                allExpressions.push({ expr,
hasil });
            }
        } catch (e) {
            console.error(`Error in
expression: ${expr} - ${e}`);
            continue;
        }
    }
}
}
}
}

return allExpressions;
}

```

Fungsi hitungKemungkinan(angka) bertugas untuk menghasilkan semua kemungkinan ekspresi matematika dengan menggunakan empat angka yang diberikan serta operator +, -, *, dan /.

- Lines 131-133 : Mendefinisikan operator matematika yang digunakan dan menyiapkan array kosong allExpressions untuk menyimpan hasil ekspresi.
- Lines 134 : Memanggil fungsi permutasi(angka) untuk mendapatkan semua permutasi dari empat angka.
- Lines 136-156: Menjalankan iterasi bersarang untuk mencoba semua kombinasi operator dan permutasi angka. Pada setiap kombinasi, ekspresi dibangun,

dievaluasi dengan `eval()`, dan jika hasilnya valid, dimasukkan ke dalam `allExpressions`.

- Line 164: Mengembalikan semua ekspresi valid.

```
// Fungsi untuk menghitung dan menampilkan hasil
function solve() {
    const angka1 = document.getElementById('angka1').value;
    const angka2 = document.getElementById('angka2').value;
    const angka3 = document.getElementById('angka3').value;
    const angka4 = document.getElementById('angka4').value;

    // Ambil input dari pengguna dan validasi
    const angkaInput = [angka1, angka2, angka3,
angka4].map(Number);

    if (angkaInput.some(isNaN) || angkaInput.length !== 4) {
        alert("Harap masukkan 4 angka yang valid!");
        return;
    }

    // Hapus hasil sebelumnya
    document.getElementById('resultBody').innerHTML = "";
    document.getElementById('solutionBody').innerHTML = "";
    document.getElementById('jumlahSolusi').style.display =
'none';
    document.getElementById('noSolution').style.display =
'none';

    // Hitung kemungkinan
    const kemungkinan = hitungKemungkinan(angkaInput);

    // Cari solusi yang menghasilkan angka 20
    const targetHasil = 20;
    const solusi = kemungkinan.filter(item => item.hasil ===
targetHasil);

    // Tampilkan solusi jika ada
    const solutionTable =
document.getElementById('solutionTable');
    const solutionBody =
document.getElementById('solutionBody');
    solutionTable.style.display = solusi.length > 0 ? 'table'
: 'none';

    solusi.forEach(item => {
        const row = `|  |
| --- |
| ${item.expr} = ${targetHasil}</td></tr>`;     }); } |

```



```

        solutionBody.innerHTML += row;
    });

    // Tampilkan jumlah solusi
    const jumlahSolusi =
document.getElementById('jumlahSolusi');
    jumlahSolusi.textContent = `Jumlah solusi yang
menghasilkan 20: ${solusi.length}`;
    jumlahSolusi.style.display = solusi.length > 0 ? 'block'
: 'none';

    // Tampilkan pesan jika tidak ada solusi
    const noSolution = document.getElementById('noSolution');
    noSolution.style.display = solusi.length === 0 ? 'block'
: 'none';
    }
</script>
</body>
</html>

```

Fungsi solve() bertanggung jawab untuk mengambil input pengguna, menghitung kemungkinan ekspresi, dan menampilkan hasil.

- Lines 168-172 : Mengambil input dari pengguna dan mengonversi ke tipe angka.
- Lines 175-180: Memvalidasi input dan menghentikan eksekusi jika input tidak valid (bukan angka atau kurang dari 4 angka).
- Lines 183-186 : Membersihkan hasil sebelumnya dari tabel dan menyembunyikan elemen yang tidak perlu.
- Lines 189: Memanggil fungsi hitungKemungkinan() untuk mendapatkan semua ekspresi.
- Lines 192-193: Menyaring hasil untuk menemukan ekspresi yang menghasilkan nilai 20.
- Lines 196-203: Jika ada solusi, menampilkan solusi di tabel solutionTable dan menambahkannya ke tabel solutionBody.
- Lines 206-213: Menampilkan jumlah solusi jika ada, atau menampilkan pesan bahwa tidak ada solusi jika tidak ditemukan.

Kesimpulan

Program "20 Solver" memungkinkan pengguna untuk memasukkan empat angka dan menghitung semua kemungkinan ekspresi matematika dengan menggunakan operator dasar, mencari ekspresi yang menghasilkan nilai 20. Program ini menggunakan metode permutasi dan validasi dengan `eval()` untuk menghitung hasil ekspresi dan menampilkan solusi.

Pembagian Tugas

Mario : Pembuatan file index, tampilan web, dan laporan

Nabil : Pembuatan file index, deployment, laporan