

**LAPORAN TUGAS MATA KULIAH
DESAIN & ANALISIS ALGORITMA**

“20 GAME SOLVER”

Dosen Pengampu:
Fajar Muslim S.T., M.T.



Disusun Oleh :

Mutia Rahman (L0123101)

Oktavia Suci Rahmadhani (L0123110)

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

2024

A. Penjabaran Masalah

Program "20 Game Solver" adalah sebuah program yang bertujuan untuk mencari semua kemungkinan kombinasi operasi aritmatika (penjumlahan, pengurangan, perkalian, dan pembagian) dan mengelompokkannya menggunakan tanda kurung, yang dapat menghasilkan nilai 20. Program ini mengambil pendekatan berbasis metode brute force, yang berarti program akan mencoba semua kemungkinan kombinasi dari empat angka yang diberikan dan operasi matematis yang bisa diterapkan.

Permainan ini mirip dengan permainan kartu 24 yang menggunakan empat angka dari kartu remi dan bertujuan untuk menghasilkan nilai 24. Pada permainan tersebut, pemain diharuskan menggunakan operasi dasar matematika (+, -, *, /) untuk menemukan kombinasi yang menghasilkan angka target. Namun, dalam program ini, target yang digunakan adalah 20, bukan 24.

Masalah yang Diselesaikan: Tujuan dari program ini adalah mencari kombinasi operasi aritmatika yang melibatkan empat angka masukan sehingga dapat menghasilkan angka 20. Angka-angka yang dimasukkan oleh pengguna bisa berupa angka acak atau angka tertentu yang dipilih sendiri. Program ini akan mencoba semua kombinasi operasi pada angka-angka tersebut untuk menghasilkan nilai 20. Jika solusi ditemukan, program akan menampilkannya, jika tidak, program akan memberi tahu bahwa tidak ada solusi yang ditemukan.

link : <https://20-gamesolver-mutvia.vercel.app/>

B. Implementasi Kode

Berikut adalah implementasi kode untuk program tersebut:

```
<!DOCTYPE html>
<html lang="en">
<head></head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Number Puzzle Solver</title>
  <style>
    body {
      background-color: #f0e6ff;
      font-family: 'Arial', sans-serif;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      min-height: 100vh;
      margin: 0;
      padding: 20px;
      box-sizing: border-box;
    }
    h1, h2 {
```

```
        text-align: center;
        color: #4a0e78;
    }
    #puzzleForm {
        display: flex;
        flex-direction: column;
        align-items: center;
        width: 100%;
        max-width: 300px;
        background-color: #ffffff;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    .input-group {
        display: flex;
        flex-direction: column;
        align-items: center;
        margin-bottom: 15px;
        width: 100%;
    }
    label {
        margin-bottom: 5px;
        color: #6c3483;
    }
    input[type="number"] {
        width: 100%;
        padding: 10px;
        text-align: center;
        border: 2px solid #d2b4de;
        border-radius: 5px;
        transition: border-color 0.3s;
    }
    input[type="number"]:focus {
        border-color: #8e44ad;
        outline: none;
    }
    button {
        margin-top: 10px;
        padding: 10px 20px;
        background-color: #8e44ad;
        color: white;
```

```

        border: none;
        border-radius: 5px;
        cursor: pointer;
        transition: background-color 0.3s;
    }
    button:hover {
        background-color: #6c3483;
    }
    #result {
        margin-top: 20px;
        text-align: center;
        background-color: #e8daef;
        padding: 15px;
        border-radius: 5px;
        color: #4a0e78;
    }
</style>
</head>
<body>
    <h1>20 GAME SOLVER</h1>
    <form id="puzzleForm">
        <input type="number" id="number1" placeholder="Angka
1"><br><br>
        <input type="number" id="number2" placeholder="Angka
2"><br><br>
        <input type="number" id="number3" placeholder="Angka
3"><br><br>
        <input type="number" id="number4" placeholder="Angka
4"><br><br>
        <button type="submit">Cari Solusi</button>
    </form>

    <h2>Hasil:</h2>
    <div id="result"></div>

    <script>
        // Class Chunk untuk menyimpan angka dan representasi
teksnya
        class Chunk {
            constructor(number, text = null) {
                this.total = number;

```

```

        this.text = text === null ? number.toString() :
text;
    }
}

    // Operasi penjumlahan, perkalian, pengurangan, dan
pembagian
    function add(a, b) {
        // Mengurutkan teks agar tidak ada duplikasi
        const sortedText = [a.text, b.text].sort().join(" +
");
        return new Chunk(a.total + b.total,
`${sortedText}`);
    }

    function subtract(a, b) {
        return new Chunk(a.total - b.total, `${a.text} -
${b.text}`);
    }

    function multiply(a, b) {
        // Mengurutkan teks agar tidak ada duplikasi
        const sortedText = [a.text, b.text].sort().join(" *
");
        return new Chunk(a.total * b.total,
`${sortedText}`);
    }

    function divide(a, b) {
        if (b.total === 0) return null; // Menghindari
pembagian dengan 0
        return new Chunk(a.total / b.total, `${a.text} /
${b.text}`);
    }

    const operations = [add, subtract, multiply, divide];

    // Fungsi rekursif untuk mencoba semua kombinasi
operasi dan angka
    function operate(chunks, target) {
        // Jika hanya ada satu chunk, periksa apakah sudah
mencapai target

```

```

        if (chunks.length === 1) {
            if (Math.abs(chunks[0].total - target) < 0.000001)
            {
                return new Set([chunks[0].text]); // Return
solusi dalam Set
            }
            return new Set();
        }

        let solutions = new Set();
        for (let i = 0; i < chunks.length; i++) {
            for (let j = 0; j < chunks.length; j++) {
                if (i !== j) {
                    // Buat chunk yang tersisa setelah mengambil
dua angka
                    const remainingChunks = chunks.filter((_, idx)
=> idx !== i && idx !== j);
                    for (let operation of operations) {
                        const result = operation(chunks[i],
chunks[j]);
                        if (result) {
                            // Lakukan operasi rekursif dengan hasil
operasi
                            const subSolutions = operate([result,
...remainingChunks], target);
                            subSolutions.forEach(solution =>
solutions.add(solution));
                        }
                    }
                }
            }
        }
        return solutions;
    }

    // Fungsi utama untuk menyelesaikan permainan dan
mencari solusi
    function solve20(numbers) {
        const chunks = numbers.map(num => new Chunk(num));
        return operate(chunks, 20);
    }

```

```

        // Event listener untuk menangani form submit

document.getElementById("puzzleForm").addEventListener("submit", function (event) {
    event.preventDefault();

    // Ambil angka dari form
    const number1 =
parseFloat(document.getElementById("number1").value);
    const number2 =
parseFloat(document.getElementById("number2").value);
    const number3 =
parseFloat(document.getElementById("number3").value);
    const number4 =
parseFloat(document.getElementById("number4").value);

    // Buat array angka
    const numbers = [number1, number2, number3, number4];

    // Panggil fungsi solve20 untuk mencari solusi
    const solutions = solve20(numbers);
    const resultDiv = document.getElementById("result");

    // Tampilkan solusi di halaman
    if (solutions.size > 0) {
        const solutionCount = solutions.size;
        const solutionText =
Array.from(solutions).join('<br>');
        resultDiv.innerHTML = `<p>Found ${solutionCount}
solution${solutionCount > 1 ? 's' : ''}:</p>${solutionText}`;
    } else {
        resultDiv.innerHTML = "No solution found";
    }
});
</script>
</body>
</html>

```

C. Pengujian Program

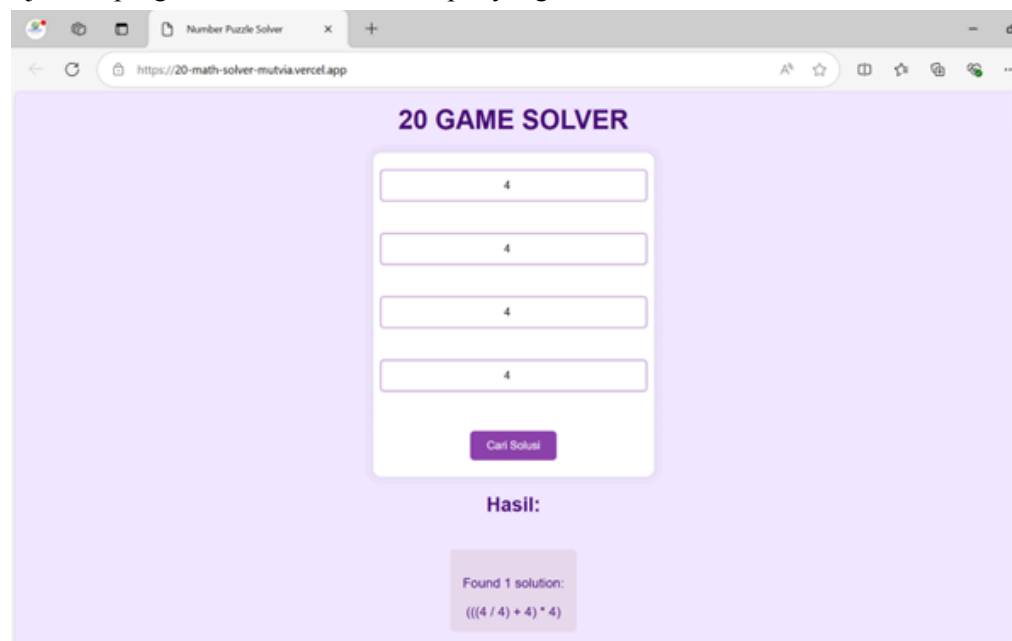
- Uji coba program untuk kombinasi input yang tidak memiliki solusi



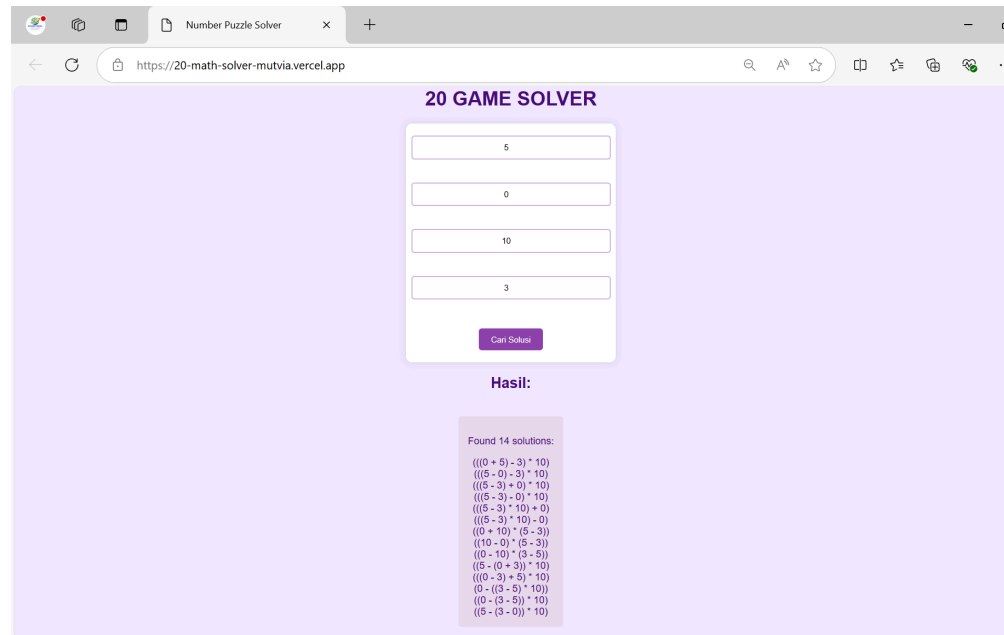
- Uji coba program untuk kombinasi input yang memiliki solusi



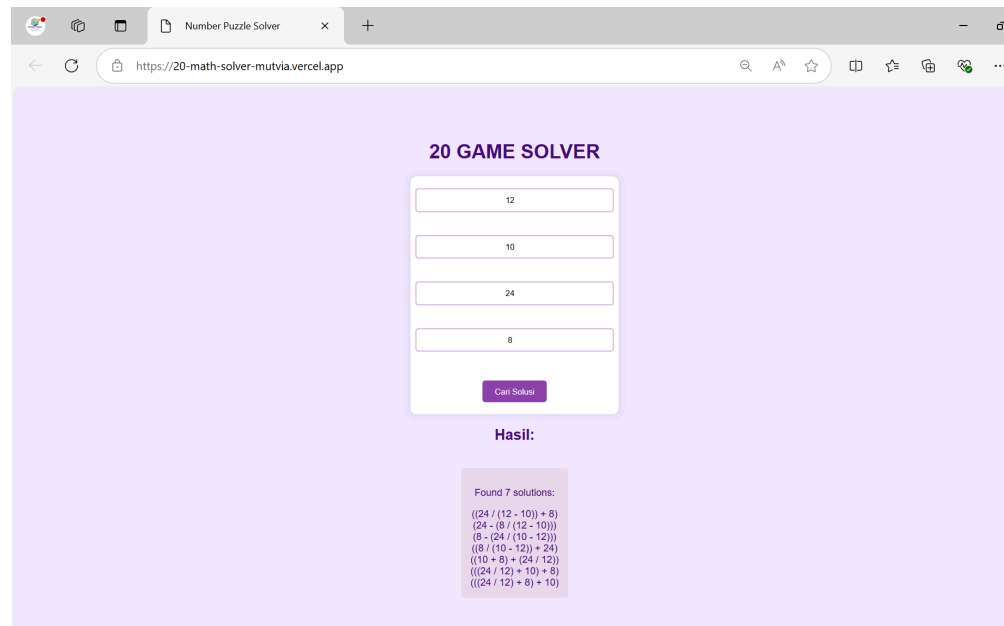
- Uji coba program untuk kombinasi input yang sama



- Uji coba program untuk input yang menghasilkan pembagian nol



- Uji coba program untuk input yang berdigit dua



	Ya	Tidak
Program berhasil dikompilasi tanpa kesalahan	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Program berhasil running	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Program dapat membaca input dan memberikan output	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Solusi yang diberikan program memenuhi (berhasil mencapai 20)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

D. Kesimpulan

Setelah dilakukan pengujian menyeluruh, program "20 Game Solver" menunjukkan performa yang cukup baik. Program mampu menampilkan pesan kesalahan ketika kombinasi input yang dimasukkan tidak memiliki solusi, memastikan pengguna mendapat umpan balik yang jelas. Selain itu, program dapat menangani input angka dua digit tanpa masalah dan mampu menghindari error saat terjadi pembagian dengan nol, dengan mengabaikan operasi yang tidak valid. Pengujian juga menunjukkan bahwa program dapat memproses input dengan nilai yang sama tanpa kesalahan atau duplikasi hasil. Program berhasil dikompilasi tanpa kesalahan dan berjalan dengan lancar di lingkungan eksekusi yang sesuai. Proses pembacaan input dan pengeluaran output dilakukan dengan benar, memberikan solusi yang sesuai dengan target yang diharapkan. Setiap solusi yang diberikan oleh program unik dan tidak ada duplikasi, meskipun ada variasi dalam urutan operasi. Secara keseluruhan, program ini berfungsi dengan baik dan dapat diandalkan untuk menghasilkan solusi yang akurat dan tepat bagi pengguna tanpa kesalahan atau inkonsistensi.

E. Referensi

Farhansyah, R. (2023). 24-Game-Solver. GitHub. Retrieved September 22, 2024, from <https://github.com/rifqifarhansyah/24-Game-Solver>

Munir, R. (n.d.). IF2211 Strategi Algoritma - Semester II Tahun 2023/2024. Informatika. Retrieved September 22, 2024, from <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2023-2024/stima23-24.htm>

F. Pembagian Tugas

Nama	Tugas
Mutia Rahman	<ul style="list-style-type: none">● Mencari referensi untuk laporan● Mengembangkan dan mengimplementasikan kode● Membuat dan mendeploy website● Membuat laporan● Membuat dan mengedit video demo
Oktavia Suci Rahmadhani	<ul style="list-style-type: none">● Mencari referensi kode● Mengembangkan dan mengimplementasikan kode● Menguji dan mengoptimalkan kode● Membuat laporan● Membuat video demo