

LAPORAN DESAIN ANALISIS ALGORITMA
PROGRAM 20 SOLVER



Disusun Oleh :

- | | |
|-------------------------------|------------|
| 1. Nadzira Rifqi Amin Rinawan | (L0123104) |
| 2. Muhammad Hafiz Dicky Putra | (L0123096) |

Dosen Pengajar :

Fajar Muslim S.T., M.T.

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET
2024

Permasalahan

20 Solver adalah program yang dirancang untuk menemukan semua kemungkinan kombinasi operasi matematika yang dapat menghasilkan nilai 20. Program ini menggunakan empat angka yang dimasukkan oleh pengguna, serta berbagai kombinasi dari empat operasi aritmatika dasar (penjumlahan +, pengurangan -, perkalian *, pembagian /), dan juga tanda kurung () untuk menentukan urutan operasi. Program ini menguji setiap kombinasi yang memungkinkan di antara angka-angka dan operasi tersebut.

Implementasi

Langkah-langkah menjalankan program :

1. Tekan Link dibawah ini.
<https://20solver.pythonanywhere.com/>
2. Masukkan 4 digit angka pada kolom yang tersedia.
3. Tekan “Enter” atau kotak “Hitung” untuk menjalankan program

Pengujian Program

Analisa Kode

HTML

Index.html

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

<!DOCTYPE html>

Deklarasi DOCTYPE untuk memberitahu browser bahwa dokumen ini adalah HTML5

<html lang="en">

Dokumen HTML yang digunakan adalah bahasa Inggris (lang="en").

```
<head>
  <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Input Angka</title>
      <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
```

<meta charset="UTF-8">

Mengatur encoding karakter halaman ke UTF-8 untuk mendukung karakter dari berbagai bahasa.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Mengatur tampilan halaman agar responsif, menyesuaikan skala berdasarkan ukuran layar perangkat pengguna.

<title>Input Angka</title>

Judul halaman web yang ditampilkan pada tab browser. Di sini, judulnya adalah "Input Angka"

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

Menyertakan file CSS eksternal melalui fungsi “url_for” biasanya digunakan dalam framework untuk mengakses file statis.

<link rel="stylesheet" href="styles.css">

Menyertakan file CSS eksternal bernama “styles.css” yang kemungkinan digunakan untuk memberikan gaya pada halaman ini. Ini adalah penautan langsung ke file CSS di direktori root atau relatif.

```
<body>
```

<body>

Bagian utama dari halaman yang memuat konten yang terlihat oleh pengguna.

```
<header>
    <h1>20 Solver</h1>
</header>
```

```
<header>
```

Bagian header dari halaman, biasanya digunakan untuk menyimpan elemen judul atau navigasi.

<h1>20 Solver</h1>

Heading utama (H1) yang menampilkan judul "20 Solver" yang mana adalah judul yang ditampilkan di website.

```
<main>
```

<main>

Bagian utama dari halaman, yang berisi konten dari suatu website.

```
<form action="/calculate" method="POST">
```

<form action="/calculate" method="POST">: Membuat form yang akan mengirim data input ke server di URL “/calculate” menggunakan metode POST. Metode POST digunakan untuk mengirimkan data yang dilakukan dibalik layar.

```
<div class="container">
    <div class="title"><h3>Masukkan 4 Angka</h3></div>
```

<div class="container">: Elemen pembungkus untuk mengelompokkan seluruh elemen form, Kelas “container” digunakan untuk memberi tata letak pada elemen-elemen di dalamnya.

<div class="title"><h3>Masukkan 4 Angka</h3></div>: Menampilkan "Masukkan 4 Angka", memberikan instruksi tentang apa yang harus user lakukan.

```
<div class="details">
    <div class="input">
```

<div class="input">: Elemen pembungkus untuk setiap input angka, digunakan untuk pengaturan tata letak.

<div class="details">: digunakan untuk memberikan gaya tertentu melalui CSS, seperti margin, padding, atau tata letak pada elemen-elemen di dalamnya.

```
<div class="input">
    <span class="input_title">Angka
Pertama</span>
    <input type="text" id="number1"
name="number1" placeholder="Masukkan angka pertama">
</div>
<div class="input">
    <span class="input_title">Angka Kedua</span>
    <input type="text" id="number2"
name="number2" placeholder="Masukkan angka kedua">
</div>
<div class="input">
    <span class="input_title">Angka Ketiga</span>
```

```

                                <input type="text" id="number3"
name="number3" placeholder="Masukkan angka ketiga">
                                </div>
                                <div class="input">
                                    <span class="input_title">Angka
Keempat</span>
                                    <input type="text" id="number4"
name="number4" placeholder="Masukkan angka keempat">

```

<div class="input"> : Elemen pembungkus untuk setiap input angka, digunakan untuk pengaturan tata letak.

**** : Label untuk input, menjelaskan jenis angka yang harus dimasukkan ("Angka Pertama" hingga "Angka Keempat")

<input type="text"> : Elemen input untuk menerima masukan teks dari pengguna. "id" dan "name" digunakan untuk identifikasi elemen dan pengiriman data ke server. "placeholder" memberikan contoh atau instruksi tambahan dalam kotak input.

```

<div class="submit"><input type="submit" value="Hitung"></div>

```

<div class="submit"> : Pembungkus untuk tombol submit, digunakan untuk mengatur gaya atau posisi tombol.

<input type="submit"> : Tombol yang mengirimkan form saat ditekan. Nilai tombol yang ditampilkan adalah "Hitung" yang akan menampilkan hasil dari perhitungan program.

```

</form>
    </div>
</main>
</body>
</html>

```

Kode diatas adalah untuk menutup struktur HTML yang telah dibuka sebelumnya.

result.html

```

<head>
    <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Hasil Perhitungan</title>
        <link rel="stylesheet" href="{ { url_for('static',
filename='style1.css') } }">
</head>

```

<title>Hasil Perhitungan</title>: Menentukan judul halaman yang akan ditampilkan di tab browser, judulnya adalah "Hasil Perhitungan".

<link rel="stylesheet" href="{{ url_for('static', filename='style1.css') }}">: Menghubungkan file CSS eksternal bernama "style1.css" yang digunakan untuk memberikan gaya pada halaman ini. Fungsi "{{ url_for('static', filename='style1.css') }}" digunakan untuk menghasilkan URL ke file statis.

```
<div class="title">
    <p>Terdapat {{num_solutions}} solusi yang ditemukan :
</p>

    <a class="links" href="/">Kembali ke Input</a>
</div>

<div>
```

<div class="title">: Elemen ini berfungsi untuk menyusun bagian judul yang berisi informasi tentang jumlah solusi yang ditemukan. Kelas "<"title">" digunakan untuk memberikan gaya khusus melalui CSS.

<p> Terdapat {{num_solutions}} solusi yang ditemukan :

Menampilkan jumlah solusi yang ditemukan, di mana "{{num_solutions}}" adalah variabel template yang berasal dari backend. Variabel ini akan diganti dengan nilai sebenarnya saat halaman dirender di server.

Kembali ke Input: tautan untuk kembali ke halaman input. Atribut "href=/" menunjuk ke halaman utama dan kelas "links" dapat digunakan untuk memberikan gaya tertentu pada tautan.

```
<ul class="solutions-list">
    {% for solution in solutions %}
    <li>{{ solution }}</li>
    {% endfor %}
</ul>
```

Bagian ini menampilkan daftar solusi yang ditemukan

<ul class="solutions-list">: Membuat daftar tak berurut (unordered list) untuk menampung solusi. Kelas "solutions-list" digunakan untuk memberikan gaya pada daftar.

{% for solution in solutions %}: Ini adalah loop di dalam template engine (seperti Jinja2) yang mengiterasi setiap item di dalam variabel solutions. Setiap solusi akan ditempatkan di dalam elemen , yang mewakili item dalam daftar.

{{ solution }}: Merupakan variabel yang akan menampilkan solusi individual dari daftar solutions.

CSS

```
*{
    margin: 0;
    padding: 0;
```

```
    box-sizing: border-box;
    font-family: 'Poppins', sans-serif;
}
```

“*” adalah global setting

Mengatur font utama halaman menjadi *Poppins* dengan font fallback *sans-serif* jika font utama tidak tersedia.

```
body {
    background:rgb(79, 79, 130);
    min-height: 100vh;
    align-items: center;
    justify-content: center;
    padding: 0 20px;
    flex-direction: column;
}
```

“body” halaman utama berwarna ungu kebiruan, halaman yang ditampilkan mencakup seluruh layar, elemen-elemen diletak di tengah layar, konten untuk ditampilkan dalam susunan vertikal, padding horizontal sebesar 20px ditambahkan di kiri dan kanan.

```
header {
    height: 100px;
    width: 100%;
    text-align: center;
    padding: 20px 0;
}
```

Tinggi header adalah 100px, teks diletakkan dipusat header, menambahkan jarak (padding) atas dan bawah sebesar 20px di dalam header.

```
h1 {
    color: gold;
    padding: 25px;
}
```

Mengatur warna teks elemen <h1> menjadi emas, lalu menambahkan jarak (padding) di sekitar teks heading, membuatnya terlihat lebih lebar dan berjarak dari elemen lain

```
.container{
    position: relative;
    width: 100%;
    background: #fff;
    padding: 25px 30px;
    border-radius: 8px;
    box-shadow: 0 0 15px rgba(0,0,0,.1);
}
```

Berwarna latar belakang putih, padding sebesar 25px di bagian atas dan bawah, serta 30px di kiri dan kanan, memberikan ruang di dalam container, sudutnya dibuat melengkung, lalu ditambahkan sedikit efek shadow

```
.container .title {  
    font-size: 20px;  
    font-weight: 500;  
    position: relative;  
    margin-bottom: 25px;  
}
```

Ukuran font 20px dan ketebalan font 500 agar terlihat jelas kalau itu adalah judul dari container, posisi elemen diatur relatif agar dapat digunakan bersama elemen lain.

```
.solutions-list{  
    display: grid;  
    grid-template-columns: repeat(5, 1fr); /* Membuat 5 kolom */  
    gap: 10px;  
    list-style-type: none; /* Menghilangkan bullet points */  
    padding: 0;  
}
```

Elemen daftar diubah menjadi dalam bentuk grid, membuat 5 kolom yang berukuran sama, bullet points default dihilangkan dari elemen list.

```
.solutions-list li {  
    border: 1px solid #000; /* Untuk membuat solusi lebih terlihat */  
    padding: 5px;  
    text-align: center;  
}
```

Teks diletakkan pada pusat setiap item daftar, ditambah sedikit border hitam disetiap item daftar, lalu diberi jarak sebesar 5px

```
.details .input .input_title{  
    display: block;  
    font-weight: 500;  
    margin-bottom: 5px;  
}
```

Mengganti display input pada 'index.html', menebalkan font sebesar 500, dan memberikan margin ke bawah sebesar 5px


```
.details .input input{
  height: 45px;
  width: 100%;
  outline: none;
  border-radius: 5px;
  border: 1px solid #ccc;
  padding-left: 15px;
  font-size: 16px;
  border-bottom-width: 2px;;
  transition: all 0.3s ease;
}
```

Mengganti ukuran input box dengan tinggi 45px dan lebar 100%, serta memberikan border dengan radius 5px, ketebalan border 1px dan 2px pada bagian bawah. memberikan padding ke kiri sebesar 15px pada input box agar placeholder-nya lebih enak dilihat. serta memberikan transisi ease dengan durasi 0.3s

```
form .submit input {
  height: 100%;
  width: 100%;
  outline: none;
  color: white;
  border: none;
  font-size: 18px;
  font-weight: 500;
  background:rgb(79, 79, 130);
  padding: 10px;
}
```

Tombol submit menggunakan warna latar belakang yang sama dengan tema utama halaman dan teks putih untuk kontras, dengan font sebesar 18px dengan ketebalan 500

```
.links {
  text-decoration: none;
  font-size: 15px;
}
```

Menghilangkan underscore pada link “Kembali ke input”, serta mengatur ukuran font sebesar 15px

```
@media (max-width: 584px) {
  .container{
    max-width: 100%;
  }

  form .details .input{
```

```

margin-bottom: 15px;
width: 100%;
}

```

Pada layar dengan lebar kurang dari 584px, container diatur untuk menempati lebar penuh layar agar kompetibel pada layar handphone

PYTHON

```

1  from flask import Flask, request, render_template
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def index():
7      return render_template('index.html')
8
9  @app.route('/calculate', methods=['POST'])

```

Kode ini bertujuan untuk mengimpor Flask agar bisa membangun aplikasi web, serta menghubungkan antara file python dan html.

request: Digunakan untuk menangani permintaan HTTP dan data dari formulir (form).

render_template: Untuk merender template HTML (file .html) dan menampilkan halaman web.

@app.route('/'): Menangani rute utama (root), yaitu halaman awal (index).

index(): untuk merender template file index.html yang merupakan halaman utama dimana pengguna akan memasukkan angka.

@app.route('/calculate', methods=['POST']): Rute ini menangani permintaan POST ketika pengguna memasukkan angka di form dan menekan tombol submit.

```

10 def calculate():
11     try:
12         # Mendapatkan input angka dari form
13         number1 = int(request.form['number1'])
14         number2 = int(request.form['number2'])
15         number3 = int(request.form['number3'])
16         number4 = int(request.form['number4'])
17
18         # Memasukkan angka
19         numbers = [number1, number2, number3, number4]
20         solutions = solve_for_20_unique(numbers)
21
22         # Hitung jumlah solusi
23         num_solutions = len(solutions)
24
25         # Kirimkan jumlah solusi dan solusi ke template
26         return render_template('result.html', solutions=solutions, num_solutions=num_solutions)
27
28     except ValueError:
29         return "Pastikan semua input adalah angka!"

```

request.form[]: Mengambil nilai input dari form yang dikirim pengguna. Dalam kasus ini, empat angka dimasukkan oleh pengguna.

Fungsi `solve_for_20_unique(numbers)` menghitung solusi (kombinasi operasi) untuk menghasilkan angka 20. Dimana numbers didapatkan dari `syntax request.form[]` sebelumnya. Hasilnya kemudian dikirim ke template `result.html` untuk ditampilkan ke pengguna.

```
31 # Fungsi yang menghitung hasil
32 def solve_for_20_unique(numbers):
33     import itertools
34     ops = {'+': Lambda x, y: x + y, '-': Lambda x, y: x - y, '*': Lambda x, y: x * y, '/': Lambda x, y: x / y if y != 0 else None}
35
36     expressions = [
37         "{a} {op1} {b} {op2} {c} {op3} {d}",
38         "({a} {op1} {b}) {op2} ({c} {op3} {d})",
39         "(({a} {op1} {b}) {op2} {c}) {op3} {d}",
40         "({a} {op1} ({b} {op2} {c})) {op3} {d}",
41         "{a} {op1} (({b} {op2} {c}) {op3} {d})",
42         "{a} {op1} ({b} {op2} ({c} {op3} {d}))"
43     ]
44
45     unique_solutions = set()
46
47     for perm_numbers in itertools.permutations(numbers):
48         for perm_ops in itertools.product(ops.keys(), repeat=3):
49             for expr_template in expressions:
50                 try:
51                     expr = expr_template.format(
52                         a=perm_numbers[0], b=perm_numbers[1], c=perm_numbers[2], d=perm_numbers[3],
53                         op1=perm_ops[0], op2=perm_ops[1], op3=perm_ops[2]
54                     )
55                     result = eval(expr)
56                     if result == 20:
57                         formatted_expr = expr.replace(" ", "")
58                         unique_solutions.add(f"{expr} = 20")
59                 except ZeroDivisionError:
60                     pass
61     return unique_solutions
```

```
31 # Fungsi yang menghitung hasil
32 def solve_for_20_unique(numbers):
33     import itertools
34     ops = {'+': Lambda x, y: x + y, '-': Lambda x, y: x - y, '*': Lambda x, y: x * y, '/': Lambda x, y: x / y if y != 0 else None}
35
```

itertools: Modul ini digunakan untuk menghasilkan semua kombinasi angka dan operasi.

ops: Dictionary yang menyimpan operasi matematika (+, -, *, /) sebagai fungsi lambda.

Fungsi lambda: Setiap operasi dijalankan secara dinamis tergantung dari input pengguna.

```
36     expressions = [
37         "{a} {op1} {b} {op2} {c} {op3} {d}",
38         "({a} {op1} {b}) {op2} ({c} {op3} {d})",
39         "(({a} {op1} {b}) {op2} {c}) {op3} {d}",
40         "({a} {op1} ({b} {op2} {c})) {op3} {d}",
41         "{a} {op1} (({b} {op2} {c}) {op3} {d})",
42         "{a} {op1} ({b} {op2} ({c} {op3} {d}))"
43     ]
44
45     unique_solutions = set()
46
```

expression: mendefinisikan semua bentuk operasi matematika jika disisipkan tanda kurung ()

unique_solutions = set(): menyimpan solusi unik yang ditemukan di fungsi `solve_for_20_unique`

```

47     for perm_numbers in itertools.permutations(numbers):
48         for perm_ops in itertools.product(ops.keys(), repeat=3):
49             for expr_template in expressions:
50                 try:
51                     expr = expr_template.format(
52                         a=perm_numbers[0], b=perm_numbers[1], c=perm_numbers[2], d=perm_numbers[3],
53                         op1=perm_ops[0], op2=perm_ops[1], op3=perm_ops[2]
54                     )
55                     result = eval(expr)
56                     if result == 20:
57                         formatted_expr = expr.replace(" ", "")
58                         unique_solutions.add(f"{expr} = 20")
59                 except ZeroDivisionError:
60                     pass
61     return unique_solutions

```

itertools.permutations(numbers): Menghasilkan semua permutasi dari angka-angka yang diberikan.

itertools.product(ops.keys(), repeat=3): Menghasilkan semua kombinasi operasi (+, -, *, /) yang mungkin.

expr_template.format: ekspresi matematika yang dihasilkan diformat menggunakan angka dan operator dari iterasi sebelumnya. a=perm_numbers[0], b=perm_numbers[1]...: Ini mengisi template dengan angka-angka dari hasil permutasi perm_numbers dan operator dari perm_ops.

eval(expr): Mengeksekusi string ekspresi matematika sebagai kode Python dan menghitung hasilnya.

Jika hasil ekspresi sama dengan 20, maka ekspresi tersebut ditambahkan ke dalam set unique_solutions.

ZeroDivisionError: Mengabaikan kasus pembagian dengan nol.

```

63  ✓ if __name__ == '__main__':
64      |     app.run(debug=True)

```

kondisi pengecekan dalam Python yang memastikan bahwa kode di bawahnya hanya dijalankan jika file Python tersebut dieksekusi secara langsung, bukan diimpor sebagai modul dalam file lain.

Tampilan Program

- Tampilan awal program

20 Solver

Masukkan 4 Angka

Angka Pertama:

Angka Kedua:

Angka Ketiga:

Angka Keempat:

Hitung

- Tampilan website jika program menemukan solusi

20 Solver

Terdapat 12 solusi yang ditemukan :
[Kembali ke Input](#)

$(1 - 6) * (4 - 8) = 20$	$(1 + (6 / 4)) * 8 = 20$	$((1 + 6) * 4) - 8 = 20$	$(8 - 4) * (6 - 1) = 20$	$8 * (1 + (6 / 4)) = 20$
$(4 - 8) * (1 - 6) = 20$	$(6 - 1) * (8 - 4) = 20$	$(4 * (6 + 1)) - 8 = 20$	$8 * ((6 / 4) + 1) = 20$	$(4 * (1 + 6)) - 8 = 20$
$((6 / 4) + 1) * 8 = 20$	$((6 + 1) * 4) - 8 = 20$			

- Tampilan jika program tidak menemukan hasil dari perhitungan

20 Solver

Terdapat 0 solusi yang ditemukan :
[Kembali ke Input](#)

- Tampilan jika user memasukan huruf bukan angka

Pastikan semua input adalah angka!

Peran Anggota Kelompok

1. Nadzira Rifqi Amin Rinawan : Membuat algoritma perhitungan menggunakan python, membuat repository, mendeploy website dan membantu membuat laporan dokumentasi.
2. Muhammad Hafiz Dicky Putra : Membantu membuat algoritma, mendeploy website dan membuat laporan dokumentasi.