**PROJECT DESAIN & ANALISIS ALGORITMA**

Laporan Problem 20 Solver

Disusun untuk Memenuhi Tugas Mata Kuliah Desain & Analisis Algoritma

Dosen Pengampu:

**Fajar Muslim, S. T., M. T.,**

****

Disusun Oleh :

1. Najwa Adlina Yanida (L0123105)

2. Nazila Nur Hanifa Ramadhani (L0123109)

**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA**

**UNIVERSITAS SEBELAS MARET**

**2024**

**BAB 1**

**PENDAHULUAN**

**A. Latar Belakang**

Project "20 Solver" adalah sebuah web berbasis coding yang dirancang untuk menyelesaikan tantangan matematika, di mana pengguna harus menemukan kombinasi tepat dari empat angka dan berbagai operasi matematika yang menghasilkan angka 20. Project ini menggunakan algoritma brute force. Dimana brute force adalah sebuah metode yang mengandalkan pencarian menyeluruh melalui semua kemungkinan kombinasi angka dan operasi untuk menemukan solusi. Dalam hal ini, algoritma akan mencoba semua kombinasi dari empat angka yang diberikan serta variasi dari operasi aritmatika dasar, seperti penjumlahan, pengurangan, perkalian, dan pembagian, hingga ditemukan solusi yang hasil akhirnya adalah 20. Dalam project web di mana pengguna dapat dengan mudah memasukkan angka yang ingin diuji, melihat hasil perhitungan, serta memahami proses pencarian solusi oleh sistem. Dengan menggunakan bahasa pemrograman seperti JavaScript dan kerangka kerja seperti HTML dan CSS.

Secara keseluruhan, project "20 Solver" tidak hanya bertujuan untuk menyelesaikan tantangan matematis, tetapi juga berfungsi sebagai latihan dalam penerapan algoritma brute force, manipulasi data, serta pengembangan aplikasi web secara menyeluruh.

**BAB 2**

**TEORI SINGKAT**

1. **Definisi Algoritma *Brute Force***

Algoritma *brute force* merupakan salah satu pendekatan dasar dalam pemecahan masalah di bidang komputasi, di mana solusi ditemukan dengan cara mencoba semua kemungkinan secara sistematis hingga solusi yang benar ditemukan. Metode ini sangat intuitif karena tidak memerlukan logika atau strategi optimasi yang rumit. Sebagai gantinya, brute force mengandalkan eksplorasi menyeluruh terhadap setiap kemungkinan yang tersedia, tanpa mempertimbangkan apakah solusi tersebut efisien atau tidak. Oleh karena itu, meskipun sederhana, algoritma ini sering kali dianggap sebagai pendekatan "*brute*" karena kurangnya kecerdasan dalam proses pencariannya.

Salah satu kelebihan utama dari algoritma brute force adalah kepastiannya dalam menemukan solusi. Jika suatu masalah memiliki solusi yang mungkin, algoritma ini pasti akan menemukannya karena ia memeriksa semua kemungkinan tanpa terkecuali. Inilah mengapa brute force bisa sangat berguna dalam situasi di mana tidak ada metode alternatif yang lebih efisien atau ketika ruang solusi relatif kecil sehingga setiap kemungkinan bisa dieksplorasi dalam waktu yang wajar. Misalnya, dalam masalah pencarian angka atau kata sandi, *brute force* akan mencoba semua kombinasi angka, huruf, atau simbol hingga menemukan yang cocok.

Namun, kelemahan utama dari pendekatan ini adalah ketidakefisienannya, terutama ketika berhadapan dengan masalah yang memiliki ruang solusi yang besar. Semakin banyak kemungkinan yang harus dicoba, semakin lama waktu yang dibutuhkan oleh algoritma untuk menemukan solusi. Misalnya, dalam kasus pencarian kata sandi yang terdiri dari delapan karakter, jika setiap karakter bisa berupa huruf besar, huruf kecil, angka, atau simbol, jumlah kombinasi yang mungkin akan sangat besar, sehingga waktu yang diperlukan oleh brute force untuk memeriksa semua kemungkinan akan menjadi sangat panjang. Inilah sebabnya mengapa dalam dunia nyata, algoritma brute force jarang digunakan untuk masalah yang kompleks, kecuali jika masalah tersebut dapat diselesaikan dengan ruang solusi yang sangat terbatas.

Secara keseluruhan, meskipun brute force mungkin bukan solusi yang paling elegan atau efisien, kesederhanaannya membuatnya mudah dipahami dan diimplementasikan, serta dapat diandalkan dalam menemukan solusi yang benar jika waktu dan sumber daya tidak menjadi kendala.

1. **Penerapan Algoritma *Brute Force***

Penerapan algoritma brute force dalam masalah menemukan empat angka dan operasi matematika yang menghasilkan nilai 20 adalah contoh sederhana dari bagaimana pendekatan ini bekerja. Dalam skenario ini, kita diminta untuk menyusun empat angka menggunakan operasi matematika seperti penjumlahan, pengurangan, perkalian, atau pembagian agar hasil akhirnya sama dengan 20. Algoritma brute force akan mencoba semua kemungkinan susunan angka dan kombinasi operasi secara sistematis hingga menemukan solusi yang benar.

Langkah pertama adalah mencoba setiap permutasi dari empat angka tersebut. Misalnya, jika kita memiliki empat angka, seperti 2, 4, 5, dan 10, algoritma brute force akan mengatur angka-angka ini dalam setiap urutan yang mungkin. Setelah menguji setiap urutan angka, algoritma brute force kemudian akan mencoba semua kemungkinan kombinasi operasi matematika di antara angka-angka tersebut. Algoritma dapat mencoba penjumlahan, pengurangan, perkalian, atau pembagian antara setiap pasangan angka.

Algoritma brute force kemudian akan menghitung hasil dari setiap kombinasi angka dan operasi tersebut dan membandingkannya dengan target hasil, yaitu 20. Jika hasilnya sama dengan 20, algoritma akan berhenti dan menampilkan kombinasi yang tepat. Jika tidak, algoritma akan terus mencoba hingga semua kemungkinan diuji.

Kelemahan dari brute force adalah pendekatan ini sering kali tidak efisien, terutama jika ruang solusi sangat besar. Semakin banyak jumlah angka atau operasi, maka waktu yang diperlukan untuk mencoba semua kombinasi akan meningkat. Oleh karena itu, brute force biasanya hanya cocok untuk masalah dengan ruang solusi terbatas, di mana kesederhanaan dan keandalan lebih diutamakan daripada efisiensi waktu.

1. **Contoh Pengimplementasian Algoritma *Brute Force***

Contoh input yang dimasukkan:

2 4 5 10

Dan goal:

20

Proses dan letak *Brute Force:*

1. Permutasi Angka:

* Algoritma akan mencoba semua kombinasi urutan dari angka yang diberikan. Dalam hal ini, terdapat 24 kemungkinan urutan (4! = 24). Menggunakan 4! dikarenakan angka yang diinput berjumlah 4.
* Contoh urutan yang akan dicoba: 2 4 5 10, 2 4 10 5, 4 2 5 10, dst.

1. Kombinasi Operasi:

* Setelah menentukan urutan angka, algoritma akan mencoba semua kemungkinan kombinasi operasi aritmatika (penjumlahan, pengurangan, perkalian, pembagian) untuk angka-angka tersebut.
* Contoh operasi yang akan dicoba: 2 + 4 + 5 + 10, 2 \* 4 + 5 \* 10, 10 - 2 + 4 \* 5, dst.

1. Evaluasi Ekspresi:

* Untuk setiap kombinasi urutan dan operasi, algoritma akan membangun ekspresi matematika sebagai string dan menggunakan eval() untuk menghitung hasilnya.
* Jika hasil dari ekspresi tersebut sama dengan 20, maka kombinasi tersebut disimpan sebagai solusi.

**BAB 3**

**KODE DAN PENGGUNAAN**

**Source Code**

* solver.js

|  |
| --- |
|  |

* Solver.html

|  |
| --- |
|  |

**Contoh penggunaan 20 Game Solver**

|  |
| --- |
|  |

Gambar 1.1 dan 1.2 Contoh permainan 20 Solver

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Hasil output dari program menunjukkan beberapa kombinasi angka dan operasi matematika yang menghasilkan nilai target 20, sesuai dengan angka yang diberikan oleh pengguna (2, 4, 5, dan 10). |

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Hasil output dari program menunjukkan beberapa kombinasi angka dan operasi matematika yang menghasilkan nilai target 20, sesuai dengan angka yang diberikan oleh pengguna (2, 4, 5, dan 6). |

**BAB 4**

**KESIMPULAN**

Proyek 20 solver ini menunjukkan bagaimana algoritma brute force dapat digunakan untuk menyelesaikan masalah pencarian kombinasi angka dan operasi aritmatika yang menghasilkan nilai target, yaitu 20. Algoritma ini bekerja dengan mencoba semua kemungkinan kombinasi dari urutan angka dan operasi matematika secara sistematis, tanpa mengabaikan satu pun kemungkinan. Meskipun brute force dijamin akan menemukan solusi jika ada, kelemahan utamanya terletak pada ketidakefisienannya, terutama ketika ruang solusi besar. Namun, dalam kasus ini, brute force cocok karena ruang solusi yang relatif kecil, memungkinkan pencarian solusi dalam waktu yang wajar. Proyek ini juga membantu mempraktikkan penerapan algoritma brute force pada aplikasi web dengan memanfaatkan JavaScript, HTML, dan CSS.