Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2022/2023

**Penyelesaian Permainan Kartu 24 dengan Algoritma *Brute Force***

Disusun oleh:

Kenneth Dave Bahana 13521145

Logo

Description automatically generated

**PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG**

**2023**

1. **Penjelasan Algoritma *Brute Force***

Algoritma *Brute Force* merupakan algoritma dengan pendekatan yang lempang(*straightforward*) untuk memecahkan suatu persoalan. Algoritma ini bersifat luas karena dapat berupa berbagai algoritma yang sifatnya sederhana dan caranya bersifat jelas dipahami manusia karena ide pembentukan algoritma ini didasarkan apa yang terlintas dalam pemikiran ketika suatu program ingin dibuat.

Pendekatan utama dari algoritma *Brute Force* adalah dengan menguji seluruh kemungkinan dalam mendapatkan hasil yang memenuhi syarat sebagai solusi dari suatu permasalahan. Pendekatan ini yang menyebabkan algoritma *Brute Force* seringkali bersifat tidak efisien dikarenakan algoritmanya tidak mengutamakan efisiensi dan kecepatan eksekusi program, namun mengutamakan program dapat berjalan dengan cara yang secara *straightforward* dan optimal sehingga algoritma ini dapat juga disebut algoritma naif (naïve algorithm) dikarenakan menggunakan usaha untuk menguji segala kemungkinan dalam pembentukkan algoritmanya, dibandingkan dengan memikirkan pengujian cara yang efisien atau membutuhkan jumlah tahapan yang lebih sedikit dalam algoritmanya sehingga tidak perlu menguji setiap kemungkinan.

Variasi Algoritma *Brute Force* yang digunakan dalam pembuatan program ini adalah konsep permutasi. Pada permainan 24 ini, untuk mencari solusi yang menghasilkan nilai 24 dari suatu empat kartu, dibutuhkan empat permutasi posisi dari keempat nilai kartu tersebut, serta kombinasi dari operator yang digunakan antara keempat nilai tersebut. Variasi untuk menguji seluruh kemungkinan dengan menggunakan permutasi didasarkan dengan urutan perhitungan yaitu dimulai dengan kartu pertama dengan kedua, atau dimulai dengan kartuma pertama dengan ketiga, dan seterusnya hingga seluruh variasi terpenuhi. Setiap variasi pengurutan nilai suatu kartu yang dihitung terlebih dahulu ditandai dengan pengurungan, dimana pengurungan tersebut mengindikasikan pengurutan perhitungan setiap nilai pada kartu dan diuji semua serta disimpan setiap nilai yang pada hitungan akhirnya menghasilkan nilai 24. Pada pengujian tersebut, perlu dibuat beberapa Batasan yang tidak memungkinkan, yaitu apabila terdapat operasi pertambahan ataupun pengurangan yang menghasilkan 0 tidak boleh menjadi penyebut dari operasi pembagian selanjutnya karena akan menghasilakn nilai yang tidak terdefinisi dan menyebabkan program error. Secara algoritma *brute force*, permutasi dilakukan dengan melakukan *for loop* sejumlah angka serta operasi yang dilakukan menggunakan 5 variasi pengurutan perhitungan yang merupakan seluruh variasi kemungkinan urutan perhitungan dari keempat nilai kartu.

1. ***Source Code* dengan Bahasa C++**

***Text

Description automatically generated***

Text

Description automatically generated

Text

Description automatically generated

***Text

Description automatically generated***

***Text

Description automatically generated***

***Text

Description automatically generated***

***Text

Description automatically generated***

***Text

Description automatically generated***

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

1. ***Screenshot* dari *Input* dan *Output* dari program**
2. Testcase dengan *Input* *user*
3. Input pertama

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A screen shot of a computer

Description automatically generated with low confidence

1. Input kedua (tidak memiliki solusi)

Text

Description automatically generated

1. Input ketiga (apabila tidak tepat)

Text

Description automatically generated

1. Testcase dengan nilai *random generator*
2. Testcase 1Text

   Description automatically generatedA picture containing text

   Description automatically generatedShape

   Description automatically generated with low confidence
3. Testcase 2

Text

Description automatically generated

1. **Daftar Pustaka**

Repository github: <https://github.com/kenndave/Tucil1_13521145>

1. ***Checklist***

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa kesalahan | **✓** |  |
| 1. Program berhasil *running* | **✓** |  |
| 1. Program berhasil membaca *input* / *generate* sendiri dan memberikan luaran | **✓** |  |
| 1. Solusi yang diberikan program memenuhi (berhasil mencapai 24) | **✓** |  |
| 1. Program dapat menyimpan solusi dalam file teks | **✓** |  |