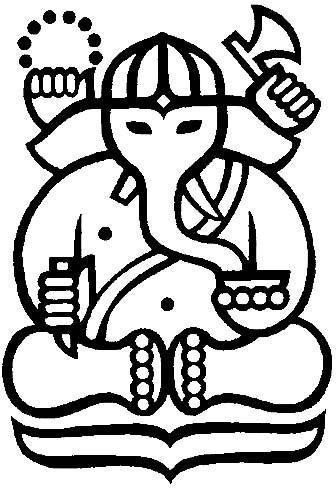
LAPORAN TUGAS KECIL 1

IF2211 STRATEGI ALGORITMA

PENYELESAIAN PERMAINAN KARTU 24 DENGAN ALGORITMA BRUTE FORCE

Disusun untuk memenuhi tugas mata kuliah Strategi Algoritma

pada Semester 2 (dua) Tahun Akademik 2022/2023.



Oleh:

|  |  |
| --- | --- |
| Muhamad Aji Wibisono | 13521095 |

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2022**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc125226425)

[BAB I PENDAHULUAN 3](#_Toc125226426)

[1.1 Latar Belakang 3](#_Toc125226427)

[BAB II ALGORITMA 4](#_Toc125226428)

[A. Penjelasan Algoritma Brute Force 4](#_Toc125226429)

[B. Source Program dalam Bahasa Java 4](#_Toc125226430)

[BAB III EKSPERIMEN 5](#_Toc125226431)

[LAMPIRAN 6](#_Toc125226432)

# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Algoritma Brute Force adalah algoritma berupa metode lempang yang menyelesaikan permasalahan dengan mengandalkan kekuatan komputasi dan tidak menggunakan teknik – teknik lain untuk meningkatkan kemangkusan. Secara sederhana, algoritma Brute Force akan mencoba semua kemungkinan solusi untuk suatu masalah dan hanya berhenti jika menemukan solusi yang tepat. Sebagai contoh diberikan permasalahan berupa kunci untuk gembok dengan empat digit angka, algoritma Brute Force akan menyelesaikan masalah tersebut dengan mencoba angka dari 0001 hingga 9999 sampai akhirnya berhasil.

Algoritma Brute Force mempunyai kelebihan yaitu mudah untuk dipikirkan serta diimplementasikan dengan kekurangan yaitu waktu yang diperlukan dan keterandalan berlebih pada kekuatan komputasi mesin..

Permainan 24 adalah permainan kartu dengan pemain mengambil empat kartu secara acak lalu mencari cara dengan operasi tambah, kurang, kali, dan bagi sehingga mendapatkan nilai 24. Untuk laporan ini permainan nilai kartu untuk permainan 24 yang tidak berupa angka adalah sebagai berikut: kartu A (as) bernilai 1, J (jack) bernilai 11, Q (queen) bernilai 12, dan K (king) bernilai 13.

Pada laporan ini dipaparkan program yang telah dibuat untuk menyelesaikan permasalahan permainan 24 dengan algoritma Brute Force. Rincian dari program yang dipaparkan terdapat pada bab – bab selanjutnya.

# 

# BAB II ALGORITMA

## 2.1. Penjelasan Algoritma Brute Force

Pada Algoritma Brute Force yang digunakan ada beberapa langkah yang dilakukan:

1. Menentukan permutasi dengan elemen sama pada kartu untuk urutan pengoperasian kartu. Hal ini dilakukan dengan mencari permutasi urutan kartu dengan indeks 0 1 2 3. Dari permutasi tersebut didapatkan urutan nilai kartu untuk urutan tersebut, lalu urutan tersebut dimasukkan kepada sebuah array yang mengingat nilai yang telah diambil jika nilai – nilai kartu belum terdapat pada array tersebut.
2. Menentukan cara pengoperasian yang dapat dilakukan. Hal ini dilakukan dengan looping operator tambah kurang kali dan bagi serta tanda kurung pada operasi tersebut. Untuk tambah kurang kali bagi dapat dilakukan dengan looping, sementara untuk tanda kurung penulis memilih untuk melakukan hard-code pada setiap kemungkinan kurung yang memungkinkan.
3. Mencoba semua kemungkinan pada permutasi kartu dan cara pengoperasian yang dapat dilakukan dengan looping untuk tiap permutasi kartu dan masing – masing operator, yaitu operator untuk kartu 0 dan 1, kartu 1 dan 2, dan kartu 2 dan 3.
4. Jika ada kombinasi kartu dan cara pengoperasian yang menghasilkan nilai 24 program akan membuat string yang sesuai dari operasi tersebut dan digabungkan pada string yang diperlukan untuk mengembalikan hasil akhir.
5. Untuk skema kurung penulis hanya memperbolehkan satu skema kurung untuk tiap kombinasi operator dan angka karena pengecekan untuk tiap kurung dilakukan dari yang paling sederhana sampai rumit, sehingga jika pengecekan awal sudah didapat nilai 24, untuk pengecekan selanjutnya skema kurung hanya menambah kurung dalam tulisan saja dan tidak mengubah cara pengoperasian.
6. String hasil akan displit untuk tiap kombinasi yang memungkinkan dan dapat disajikan pada pengguna.

## 2.2. Source Program dalam Bahasa Java

Penulis berusaha untuk mengaplikasikan paradigma pemrograman berorientasi objek pada pengimplementasian program sebagai usaha mempelajari paradigma tersebut. Sebab itu, program akan dibagi sebagai beberapa kelas yang tertulis pada folder ./src/

### Card.java

Kelas ini adalah kelas yang menyimpan informasi empat kartu yang didapat serta kombinasi yang dapat dihasilkan dari kartu – kartu tersebut.

Adapula source code adalah sebagai berikut:

### Op.java

Kelas ini adalah kelas yang menyimpan informasi operasi yang dapat dilakukan. Kelas ini juga mengandung algoritma utama yang digunakan untuk menyelesaikan permainan kartu 24.

### InputHandler.java

Kelas ini adalah kelas yang digunakan untuk memvalidasi input dari user sesuai input yang diperlukan oleh program. Di dalamnya menggunakan library java yaitu scanner untuk menerima input dari user.

Terdapat method untuk kelas ini:

1. Int Choice(int bottomLimit, int upperLimit)

Menerima argumen untuk batas bawah dan bawah atas. Kelas akan menunggu sampai pengguna memberikan nilai yang di antara dua nilai tersebut lalu mengembalikannya.

I.S:

F.S:

1. String[] SpacedWords(int wordnum)

Menerima argumen untuk jumlah kata yang diperlukan. Kelas akan menunggu sampai pengguna memberikan jumlah kata string yang dipisahkan oleh tanda spasi sesuai dengan argument yang diberikan.

I.S:

F.S:

1. String StringLine()

Menerima input dari user pada satu baris dan mengembalikan string tersebut.

I.S:

F.S:

### Interface.java

Kelas ini adalah kelas yang menggunakan kelas lainnya, yaitu Card, Op, dan InputHandler. Saat program dijalankan, kelas ini akan menyajikan user interface berupa tulisan dan command line. Kelas ini juga menghubungkan input user yang diterima dari InputHandler kepada aksi yang perlu dilakukan oleh kelas Card serta Objek.

Terdapat 1 method untuk kelas ini:

### Main.java

Kelas ini berupa program yang dijalankan. Pada pengimplementasiannya kelas ini hanya memanggil kelas lain yaitu interface yang akhirnya menjalankan program secara keseluruhan.

Terdapat 1 method untuk kelas ini:

# BAB III EKSPERIMEN

## 3.1. Hasil Program

### 3.1.1. test1.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
| A picture containing text  Description automatically generated | Keluaran hasil jika memberikan 4 kartu berbeda |

3.1.2. test2.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
| Shape  Description automatically generated with medium confidence | Keluaran hasil jika memberikan 4 kartu yang sama |

### 3.1.3. test3.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
| Shape  Description automatically generated with medium confidence | Keluaran hasil jika memberikan 4 angka yang tidak memiliki solusi |

### 3.1.4. test4.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Keluaran hasil jika memberikan 4 kartu yang bernilai huruf |

### 3.1.5. test5.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
| Shape  Description automatically generated with medium confidence | Keluaran hasil jika memilih kartu secara random |

### 3.1.6. test6.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
| A picture containing text  Description automatically generated | Keluaran hasil lain jika memilih kartu secara random |

### 3.1.7. Contoh saat memilih output ke file

|  |  |
| --- | --- |
| Gambar | Keterangan |
| Shape  Description automatically generated with medium confidence | Contoh 3.1.2. dengan keluaran dipilih untuk pada file |

## 3.2. Pembahasan

Pada eksperimen yang dilakukan oleh penulis, jika dibandingkan dengan situs referensi yang terdapat pada spesifikasi tugas kecil (<http://24solver.us-west-2.elasticbeanstalk.com/>) memiliki solusi yang lebih sedikit. Hal ini disebabkan program yang dibuat dalam laporan ini tidak mengecek skema kurung lain setelah didapatkan sebuah solusi dengan satu buah skema tanda kurung, selain itu, skema tanda kurung yang dicek oleh program juga berbeda dengan web referensi. Jika skema tanda kurung disamakan dengan mengubah baris 51 hingga 83 pada file ./src/Op.java dengan komen dari baris 93 hingga 104 pada file yang sama akan didapatkan hasil yang sama persis dengan yang dihasilkan oleh web referensi.

Penulis melakukan pengurangan terhadap tanda kurung karena konvensi *precedence* pertambahan dan perkalian sudah dilakukan terlebih dahulu pada fungsi eval3 dan eval4 yang terdapat pada file ./src/Op.java. Akan tetapi, penulis tidak memastikan dengan lebih lanjut bahwa semua kemungkinan telah tercakup sepenuhnya. Selain itu, urutan pengecekan kurung juga belum dipastikan untuk mendahulukan skema kurung yang paling sederhana. Dengan demikian, penulis masih menyimpan skema tanda kurung dari web referensi pada baris komen yang telah disebutkan sebelumnya dan program dapat diganti untuk mengeluarkan keluaran yang sama persis seperti web referensi.

# LAMPIRAN

**Repository Github**

Berikut adalah link repository GitHub untuk program penulis.

<https://github.com/MuhamadAjiW/Basogoreng12rebuan>