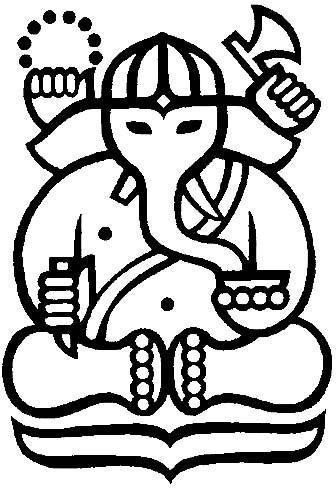
LAPORAN TUGAS KECIL 1

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](#_Toc125329048)

[BAB I PENDAHULUAN 3](#_Toc125329049)

[1.1. Latar Belakang 3](#_Toc125329050)

[BAB II ALGORITMA 4](#_Toc125329051)

[2.1. Penjelasan Algoritma Brute Force 4](#_Toc125329052)

[2.2. Source Program dalam Bahasa Java 5](#_Toc125329053)

[BAB III EKSPERIMEN 11](#_Toc125329054)

[3.1. Hasil Program 11](#_Toc125329055)

[3.2. Pembahasan 15](#_Toc125329056)

[LAMPIRAN 17](#_Toc125329057)

# 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

### Card.java

|  |
| --- |
| import java.util.Random;  import java.util.Vector;  public class Card {      InputHandler inputHandler = new InputHandler();      float[] Val = new float[4];      Vector<float[]> cbmVector = new Vector<float[]>(1);      public void readCards(){          String[] vals = new String[4];          Boolean valid = false;          System.out.println("Masukkan 4 kombinasi kartu: ");          while (!valid){              vals = inputHandler.SpacedWords(4);              try{                  for(byte i = 0; i < 4; i++){                      switch(vals[i]){                          case "A":                              Val[i] = 1;                              break;                          case "J":                              Val[i] = 11;                              break;                          case "Q":                              Val[i] = 12;                              break;                          case "K":                              Val[i] = 13;                              break;                          default:                              Val[i] = Integer.parseInt(vals[i]);                              if (Val[i] < 2 || Val[i] > 10){                                  throw new Exception("Ada kartu < 2 atau > 10 (Gunakan A, J, Q, K)");                              }                              break;                      }                  }                  valid = true;              } catch (Exception e){                  System.out.println("Input tidak valid!");                  System.out.println(e+"\n");              }          }      }      public void randomCards(){          Random randomNum = new Random();          System.out.println("Kartu yang didapat: ");          int temp;          for(byte i = 0; i < 4; i++){              temp = randomNum.nextInt(13) + 1;              Val[i] =((float)temp);              switch(temp){                  case 1:                  System.out.print("A ");                  break;                  case 11:                  System.out.print("J ");                  break;                  case 12:                  System.out.print("Q ");                  break;                  case 13:                  System.out.print("K ");                  break;                  default:                  System.out.print(String.format("%d ", temp));                  break;              }          }          System.out.print("\n");      }      boolean sameContent(float[] arr1, float[] arr2){          boolean retval = true;          byte i = 0;          if(arr1.length != arr2.length){              retval = false;          }          while(retval && i < arr1.length){              if(arr1[i] != arr2[i]){                  retval = false;              } else{                  i++;              }          }          return retval;      }      boolean VectorContains(float[] arr){          boolean retval = false;          byte i;          i = 0;          while(!retval && i < cbmVector.size()){              if(sameContent(arr, cbmVector.get(i))){                  retval = true;              }              i++;          }          return retval;      }      void ValPermutation(int index){          for(byte i = 0; i < 4; i++){              for(byte j = 0; j < 4; j++){                  if(j!=i){                      for(byte k = 0; k < 4; k++){                          if(k!=i && k!=j){                              for (byte l = 0; l < 4; l++){                                  if(l!=i && l!=j && l!=k){                                      float[] getter = {Val[i], Val[j], Val[k], Val[l]};                                      if(!VectorContains(getter)){                                          cbmVector.add(getter);                                      }                                  }                              }                          }                      }                  }              }          }      }      public void generateCbm(){          ValPermutation(0);      }  } |

### Op.java

|  |
| --- |
| import java.io.BufferedWriter;  import java.io.FileWriter;  import java.util.Vector;  public class Op {      char[] Oper = {'+', '-', '\*', '/'};      Vector<String> retval = new Vector<String>(1);      static float eval2 (float a, char op, float b){          if(op == '+'){              return a + b;          }          else if(op == '-'){              return a - b;          }          else if(op == '\*'){              return a \* b;          }          else return a / b;      }      static float eval3 (float a, char op1, float b, char op2, float c){          if((op2 == '\*' || op2 == '/')){              return eval2(a, op1, eval2(b, op2, c));          }          else return eval2(eval2(a, op1, b), op2, c);      }      public static float eval4 (float a, char op1, float b, char op2, float c, char op3, float d){          if((op1 == '+' || op1 == '-') && (op2 == '+' || op2 == '-') && (op3 == '\*' || op3 == '/')){              return eval2(eval2(a, op1, b), op2, eval2(c, op3, d));          }          else if((op1 == '+' || op1 == '-') && (op2 == '\*' || op2 == '/') && (op3 == '+' || op3 == '-')){              return eval2(eval2(a, op1, eval2(b, op2, c)), op3, d);          }          else if((op1 == '+' || op1 == '-') && (op2 == '\*' || op2 == '/') && (op3 == '\*' || op3 == '/')){              return eval2(a, op1, eval2(eval2(b, op2, c), op3, d));          }          else if((op1 == '\*' || op1 == '/') && (op2 == '+' || op2 == '-') && (op3 == '\*' || op3 == '/')){              return eval2(eval2(a, op1, b), op2, eval2(c, op3, d));          }          else return eval2(eval2(eval2(a, op1, b), op2, c), op3, d);      }      public void getmodels(Vector<float[]> Cbm){          byte i, j, k, l;              for(i = 0; i < Cbm.size(); i++){                  for(j = 0; j < 4; j++){                      for(k = 0; k < 4; k++){                          for(l = 0; l < 4; l++){                              if (eval4(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("%.0f %c %.0f %c %.0f %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval3(eval2(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1]), Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("(%.0f %c %.0f) %c %.0f %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval3(Cbm.get(i)[0], Oper[j], eval2(Cbm.get(i)[1], Oper[k], Cbm.get(i)[2]), Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("%.0f %c (%.0f %c %.0f) %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval3(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], eval2(Cbm.get(i)[2], Oper[l], Cbm.get(i)[3])) == 24){                                  retval.add(String.format("%.0f %c %.0f %c (%.0f %c %.0f)", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(eval2(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1]), Oper[k], eval2(Cbm.get(i)[2], Oper[l], Cbm.get(i)[3])) == 24){                                  retval.add(String.format("(%.0f %c %.0f) %c (%.0f %c %.0f)", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(eval3(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2]), Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("(%.0f %c %.0f %c %.0f) %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(eval2(eval2(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1]), Oper[k], Cbm.get(i)[2]), Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("((%.0f %c %.0f) %c %.0f) %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(eval2(Cbm.get(i)[0], Oper[j], eval2(Cbm.get(i)[1], Oper[k], Cbm.get(i)[2])), Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("(%.0f %c (%.0f %c %.0f)) %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(Cbm.get(i)[0], Oper[j], eval3(Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3])) == 24){                                  retval.add(String.format("%.0f %c (%.0f %c %.0f %c %.0f)", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(Cbm.get(i)[0], Oper[j], eval2(eval2(Cbm.get(i)[1], Oper[k], Cbm.get(i)[2]), Oper[l], Cbm.get(i)[3])) == 24){                                  retval.add(String.format("%.0f %c ((%.0f %c %.0f) %c %.0f)", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              else if (eval2(Cbm.get(i)[0], Oper[j], eval2(Cbm.get(i)[1], Oper[k], eval2(Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]))) == 24){                                  retval.add(String.format("%.0f %c (%.0f %c (%.0f %c %.0f))", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              /\*                              if (eval2(eval2(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1]), Oper[k], eval2(Cbm.get(i)[2], Oper[l], Cbm.get(i)[3])) == 24){                                  retval.add(String.format("(%.0f %c %.0f) %c (%.0f %c %.0f)", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              if (eval2(eval2(eval2(Cbm.get(i)[0], Oper[j], Cbm.get(i)[1]), Oper[k], Cbm.get(i)[2]), Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("((%.0f %c %.0f) %c %.0f) %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              if (eval2(eval2(Cbm.get(i)[0], Oper[j], eval2(Cbm.get(i)[1], Oper[k], Cbm.get(i)[2])), Oper[l], Cbm.get(i)[3]) == 24){                                  retval.add(String.format("(%.0f %c (%.0f %c %.0f)) %c %.0f", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              if (eval2(Cbm.get(i)[0], Oper[j], eval2(eval2(Cbm.get(i)[1], Oper[k], Cbm.get(i)[2]), Oper[l], Cbm.get(i)[3])) == 24){                                  retval.add(String.format("%.0f %c ((%.0f %c %.0f) %c %.0f)", Cbm.get(i)[0], Oper[j], Cbm.get(i)[1], Oper[k], Cbm.get(i)[2], Oper[l], Cbm.get(i)[3]));                              }                              \*/                          }                      }                  }              }      }      public void printResult(){          if(retval.size() == 0){              System.out.println("Tidak ada solusi");          }          else{              String output;              System.out.println(String.format("Terdapat %d solusi:", retval.size()));              for(int i = 0; i < retval.size(); i++){                  output = retval.get(i).replace("11", "J").replace("12", "Q").replace("13", "K").replace("10", "T").replace("1", "A").replace("T", "10");                  System.out.println(output);              }          }      }      public void fileResult(String filename){          String fileOutput = filename + ".txt";          if(retval.size() == 0){              try{                  BufferedWriter bw = new BufferedWriter(new FileWriter("./test/" + fileOutput));                  bw.write("Tidak ada solusi");                  bw.newLine();                  bw.flush();                  bw.close();                  System.out.println("\n\n\nFile berhasil disimpan pada /test/" + fileOutput);              } catch(Exception e){                  System.out.println(e);              }          }          else{              String output;              try{                  BufferedWriter bw = new BufferedWriter(new FileWriter("./test/" + fileOutput));                  bw.write(String.format("Terdapat %d solusi:", retval.size()));                  bw.newLine();                  for(int i = 0; i < retval.size(); i++){                      output = retval.get(i).replace("11", "J").replace("12", "Q").replace("13", "K").replace("10", "T").replace("1", "A").replace("T", "10");                      bw.write(output);                      bw.newLine();                  }                  bw.flush();                  bw.close();                  System.out.println("\n\n\nFile berhasil disimpan pada /test/" + fileOutput);              } catch(Exception e){                  System.out.println(e);              }          }        }  } |

### InputHandler.java

|  |
| --- |
| import java.util.Scanner;  public class InputHandler{      static Scanner in = new Scanner(System.in);      String inputLine;      String[] parsedInput;      int Choice(int bottomLimit, int upperLimit){          int retval = bottomLimit - 1;          do{              inputLine = in.nextLine();              parsedInput = inputLine.split(" ");              if(parsedInput.length == 1){                  try {                      retval = Integer.parseInt(parsedInput[0]);                  } catch (NumberFormatException e) {                      retval = bottomLimit-1;                  }                  if (retval < bottomLimit || retval > upperLimit) {                      System.out.println("Input tidak valid");                  }              }              else{                  retval = bottomLimit - 1;                  System.out.println("Jumlah input tidak valid");              }          } while (retval < bottomLimit || retval > upperLimit);          return retval;      }      String[] SpacedWords(int wordnum){          boolean valid = false;          do{              inputLine = in.nextLine();              parsedInput = inputLine.split(" ");              if(parsedInput.length == wordnum){                  valid = true;              }              else{                  System.out.println("Jumlah input tidak valid");              }          } while (!valid);          return parsedInput;      }      String StringLine(){          return in.nextLine();      }  } |

### Interface.java

|  |
| --- |
| public class Interface{      public void run(){          boolean running = true;          String filename;          int input;          long startTime;          long finishTime;          while (running){                  input = 0;                  startTime = 0;                  finishTime = 0;                  filename = "";                  Card combination = new Card();                  Op operation = new Op();                  InputHandler inputHandler = new InputHandler();                  System.out.println("24 Solver");                  System.out.println("Pilih cara:");                  System.out.println("1. Input manual");                  System.out.println("2. Randomly generated");                  System.out.println("3. Exit");                  input = inputHandler.Choice(1, 3);                  switch(input){                      case 1:                      combination.readCards();                      break;                      case 2:                      combination.randomCards();                      break;                      case 3:                      running = false;                      break;                  }                    if (running == true){                      startTime = System.nanoTime();                      combination.generateCbm();                      operation.getmodels(combination.cbmVector);                      finishTime = System.nanoTime() - startTime;                        System.out.println("Selesai");                      System.out.println("Waktu eksekusi (milisekon): " + (double)finishTime /1000000);                      System.out.println("Pilih output:");                      System.out.println("1. Terminal");                      System.out.println("2. File");                      input = inputHandler.Choice(1, 2);                      switch(input){                          case 1:                          operation.printResult();                          break;                            case 2:                          System.out.println("Masukkan nama file: ");                          filename = inputHandler.StringLine();                          operation.fileResult(filename);                          break;                      }                      System.out.println("\n\n");                  }          }            System.out.println("Program Selesai");      }  } |

### Main.java

|  |
| --- |
| public class Main {      public static void main(String[] args) {          Interface solver24 = new Interface();          solver24.run();      }  } |

# BAB III EKSPERIMEN

## 3.1. Hasil Program

### 3.1.1. test1.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Keluaran hasil jika memberikan 4 kartu berbeda |

3.1.2. test2.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Keluaran hasil jika memberikan 4 kartu yang sama |

### 3.1.3. test3.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | 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 |
|  | Keluaran hasil jika memilih kartu secara random |

### 3.1.6. test6.txt

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Keluaran hasil lain jika memilih kartu secara random |

### 3.1.7. Contoh saat memilih output ke file

|  |  |
| --- | --- |
| Gambar | Keterangan |
|  | Contoh 3.1.6. 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. Jika skema tanda kurung disamakan dengan mengubah baris 49 hingga 81 pada file ./src/Op.java dengan komen dari baris 82 hingga 95 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/Tucil1Stima>