

**LAPORAN TUGAS KECIL 1 IF2211 STRATEGI
ALGORITMA
PENYELESAIAN PERMAINAN KARTU 24 DENGAN
ALGORITMA BRUTE FORCE**



Oleh:

Muhammad Raihan Iqbal 13518134

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023**

Bab 1

Algoritma Brute Force

Berikut adalah langkah-langkah algoritma Brute Force yang digunakan untuk menyelesaikan kasus pada permainan ini:

1. Membuat susunan semi-permutasi (belum mengatur apabila ada nilai yang sama sehingga dapat muncul solusi ganda) dari empat buah nilai yang ada.
2. Melakukan perhitungan terhadap keempat nilai tersebut dengan menggunakan kombinasi operator yang ada (tambah, kali, kurang, bagi).
3. Apabila sebuah perhitungan mendapatkan hasil 24, maka tahapan perhitungan tersebut akan dimasukkan ke dalam list.

Bab 2

Source Program

Berikut adalah kode program yang dibuat:

```
import java.util.*;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

public class Main {
    public static float operations(float var1, char
operator, float var2) {
        switch (operator) {
            case '+':
                return (float)var1 + (float)var2;
            case '*':
                return (float)var1 * (float)var2;
            case '-':
                return (float)var1 - (float)var2;
            default:
                return (float)var1 / (float)var2;
        }
    }

    public static ArrayList<String> solutionChecker(int
var1, int var2, int var3, int var4) {
        ArrayList<String> solutions = new ArrayList<String>();
        char[] operators = {'+', '*', '-', '/'};
        int[][] permutations = {
            { var1, var2, var3, var4, },
            { var1, var2, var4, var3, },
            { var1, var3, var2, var4, },
            { var1, var3, var4, var2, },
            { var1, var4, var2, var3, },
            { var1, var4, var3, var2, },
            { var2, var1, var3, var4, },
            { var2, var1, var4, var3, },
            { var2, var3, var1, var4, },
            { var2, var3, var4, var1, },
            { var2, var4, var1, var3, },
            { var2, var4, var3, var1, },
            { var3, var1, var2, var4, },
            { var3, var1, var4, var2, },
            { var3, var2, var1, var4, },
            { var3, var2, var4, var1, },
            { var3, var4, var1, var2, },
            { var3, var4, var2, var1, },
            { var4, var1, var2, var3, },
        }
```

```

        { var4, var1, var3, var2, },
        { var4, var2, var1, var3, },
        { var4, var2, var3, var1, },
        { var4, var3, var1, var2, },
        { var4, var3, var2, var1, },
    };

    long startTime = System.currentTimeMillis();

    for (char i : operators) {
        for (char j : operators) {
            for (char k : operators) {
                for (int l = 0; l < 24; l++) {
                    int a = permutations[l][0];
                    int b = permutations[l][1];
                    int c = permutations[l][2];
                    int d = permutations[l][3];

                    // mengecek hasil operasi
                    if (operations(operations(operations(a, i, b),
j, c), k, d) == 24) { // ((var opr var) opr var) opr var
                        solutions.add("(" + "(" + a + i + b + ")" +
j + c + ")" + k + d);
                    }
                    if (operations(operations(a, i, b), j,
operations(c, k, d)) == 24) { // (var opr var) opr (var
opr var)
                        solutions.add("(" + a + i + b + ")" + j +
 "(" + c + k + d + ")");
                    }
                    if (operations(operations(a, i, operations(b,
j, c)), k, d) == 24) { // (var opr (var opr var)) opr var
                        solutions.add("(" + a + i + "(" + b + j + c
+ ")" + k + d + ")");
                    }
                    if (operations(a, i, operations(operations(b,
j, c), k, d)) == 24) { // var opr ((var opr var) opr var)
                        solutions.add(a + i + "(" + "(" + b + j + c
+ ")" + k + d + ")");
                    }
                    if (operations(a, i, operations(b, j,
operations(c, k, d))) == 24) { // var opr (var opr (var
opr var))
                        solutions.add(a + i + "(" + b + j + "(" + c
+ k + d + ")" + ")");
                    }
                }
            }
        }
    }

    long endTime = System.currentTimeMillis();

```

```

        long duration = endTime - startTime;
        solutions.add(Long.toString(duration));

        return solutions;
    }
    public static void main(String[] args) {
        while(true) {
            Scanner sc= new Scanner(System.in);
            System.out.println("");
            System.out.println("Opsi peng-input-an 4 buah
kartu:");
            System.out.println("1. Input sendiri");
            System.out.println("2. Generate secara random");
            System.out.println("0. Keluar");
            System.out.println("");
            System.out.print("Masukkan pilihan input: ");
            int chosenOption = Integer.parseInt(sc.nextLine());
            if (chosenOption == 0) {
                System.out.println("Terima kasih telah menggunakan
program ini!");
                break;
            }
            int[] cardArray = new int[4];
            boolean check = true;
            switch (chosenOption) {
                case 1:
                    System.out.print("Masukkan 4 buah kartu: ");
                    String cardInput = sc.nextLine();
                    String[] tempCardArray = cardInput.split(" ");
                    if (tempCardArray.length != 4) {
                        System.out.println("Masukan tidak sesuai");
                        check = false;
                        break;
                    }
                    for (int i = 0; i < tempCardArray.length; i++) {
                        System.out.println(tempCardArray[i]);
                    }
                    for (int i = 0; i < tempCardArray.length; i++) {
                        if (tempCardArray[i] == "A") {
                            cardArray[i] = 1;
                        }
                        else if (tempCardArray[i] == "J") {
                            cardArray[i] = 11;
                        }
                        else if (tempCardArray[i] == "Q") {
                            cardArray[i] = 12;
                        }
                        else if (tempCardArray[i] == "K") {
                            cardArray[i] = 13;
                        }
                        // else if (Integer.parseInt(tempCardArray[i])
> 13 || Integer.parseInt(tempCardArray[i]) < 1) {

```

```

        //    System.out.println("Masukan tidak
sesuai");
        //    break;
        // }
        else {
            try {
                int temp =
Integer.parseInt(tempCardArray[i]);
                if (temp > 13 || temp < 1) {
                    throw new Error();
                }
                cardArray[i] = temp;
            } catch (Exception e) {
                System.out.println("Masukan tidak
sesuai");
                check = false;
                break;
            }
        }
        for (int i = 0; i < cardArray.length; i++) {
            System.out.print(cardArray[i]);
        }
        System.out.println("");
        break;
    case 2:
        int minRange = 1;
        int maxRange = 13;
        for (int i = 0; i < 4; i++) {
            cardArray[i] = (int)Math.floor(Math.random() *
(maxRange - minRange + 1) + minRange);
        }

        System.out.println("4 kartu yang dihasilkan: ");
        for (int i = 0; i < cardArray.length; i++) {
            if (cardArray[i] == 1) {
                System.out.print("A ");
            }
            else if (cardArray[i] == 11) {
                System.out.print("J ");
            }
            else if (cardArray[i] == 12) {
                System.out.print("Q ");
            }
            else if (cardArray[i] == 13) {
                System.out.print("K ");
            }
            else {
                System.out.print(cardArray[i] + " ");
            }
        }
        System.out.println("");

```

```

        break;
    }
    if (check) {
        System.out.println("");
        int var1 = cardArray[0];
        int var2 = cardArray[1];
        int var3 = cardArray[2];
        int var4 = cardArray[3];
        ArrayList<String> solutions =
solutionChecker(var1, var2, var3, var4);
        if (solutions.size()-1 > 0) {
            System.out.println(solutions.size()-1 + "
solutions found");
            for (int i = 0; i < solutions.size()-1; i++) {
                System.out.println(solutions.get(i));
            }
        }
        else {
            System.out.println("Tidak ada solusi");
        }
        System.out.println("Waktu eksekusi program: " +
solutions.get(solutions.size()-1) + " milisekon");

        System.out.println("Apakah anda ingin menyimpan
solusi? ");
        System.out.println("1. Ya");
        System.out.println("2. Tidak");
        System.out.print("Masukkan pilihan penyimpanan:
");
        int fileSave = Integer.parseInt(sc.nextLine());
        if (fileSave == 1) {
            System.out.print("Nama file yang diinginkan: ");
            String fileName = sc.nextLine();
            try {
                File myObj = new File(fileName + ".txt");
                if (myObj.createNewFile()) {
                    FileWriter writeContent = new
FileWriter(fileName + ".txt");
                    for (int i = 0; i < solutions.size()-1; i++)
{
                        writeContent.write(solutions.get(i)+"\n");
                    }
                    writeContent.close();
                    System.out.println("File berhasil dibuat: "
+ myObj.getName());
                } else {
                    System.out.println("File dengan nama
tersebut telah ada.");
                }
            } catch (IOException e) {
                System.out.println("Terdapat error.");
                e.printStackTrace();
            }
        }
    }
}

```

```
        }  
    }  
}  
else {  
    System.out.println("");  
    System.out.println("Masukan tidak sesuai");  
}  
// sc.close();  
// break;  
}  
}  
}
```


Bab 3

Tangkapan Layar Hasil Program

Contoh 1. Output disimpan dalam file selfinput1.txt

```
Opsi peng-input-an 4 buah kartu:
1. Input sendiri
2. Generate secara random
0. Keluar

Masukkan pilihan input: 1
Masukkan 4 buah kartu: 7 5 5 7
7
5
5
7
7557

144 solutions found
((7+5)+5)+7
(7+5)+(5+7)
(7+(5+5))+7
```

.....

```
(7-5)*(7+5)
(7-5)*(5+7)
Waktu eksekusi program: 1 milisekon
```

Contoh 2. Output disimpan dalam file selfinput2.txt

```
Opsi peng-input-an 4 buah kartu:
1. Input sendiri
2. Generate secara random
0. Keluar

Masukkan pilihan input: 1
Masukkan 4 buah kartu: 9 5 7 6

6 solutions found
49(9*(7-5))
49((7-5)*9)
(9*(7-5)+6)
((7-5)*9)+6
51(9*(5-7))
51((5-7)*9)
Waktu eksekusi program: 1 milisekon
```

Contoh 3. Output disimpan dalam file selfinput3.txt

```
Opsi peng-input-an 4 buah kartu:
1. Input sendiri
2. Generate secara random
0. Keluar

Masukkan pilihan input: 1
Masukkan 4 buah kartu: 9 10 8 8

Tidak ada solusi
Waktu eksekusi program: 1 milisekon
```

Contoh 4. Output disimpan dalam file autogenerate1.txt

```
Opsi peng-input-an 4 buah kartu:
1. Input sendiri
2. Generate secara random
0. Keluar

Masukkan pilihan input: 2
4 kartu yang dihasilkan:
K 9 2 2

Tidak ada solusi
Waktu eksekusi program: 1 milisekon
```

Contoh 5. Output disimpan dalam file autogenerate2.txt

```
Opsi peng-input-an 4 buah kartu:
1. Input sendiri
2. Generate secara random
0. Keluar

Masukkan pilihan input: 2
4 kartu yang dihasilkan:
8 10 6 J

2 solutions found
(10*(11-8)-6)
((11-8)*10)-6
Waktu eksekusi program: 0 milisekon
```

Contoh 6. Output disimpan dalam file autogenerate3.txt

```
Opsi peng-input-an 4 buah kartu:
1. Input sendiri
2. Generate secara random
0. Keluar

Masukkan pilihan input: 2
4 kartu yang dihasilkan:
10 7 3 6

16 solutions found
53((7*6)/3)
53(7*(6/3))
53((6*7)/3)
53(6*(7/3))
53((7/3)*6)
53((6/3)*7)
53(7/(3/6))
53(6/(3/7))
((7*6)/3)+10
(7*(6/3)+10)
((6*7)/3)+10
(6*(7/3)+10)
((7/3)*6)+10
((6/3)*7)+10
(7/(3/6)+10)
(6/(3/7)+10)
Waktu eksekusi program: 1 milisekon
```

Bab 4

Tautan Repositori

Berikut adalah tautan repositori dari program yang telah dibuat:

https://github.com/raihaniqbal24/Tucil1_13518134

Check List Kelengkapan

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