

TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA
Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force



Disusun oleh:
13521155 - Kandida Edgina Gunawan

INSTITUT TEKNOLOGI BANDUNG
2023

DAFTAR ISI

DAFTAR ISI	2
BAB 1 : Deskripsi Masalah daan Algoritma	3
Algoritma Brute Force	3
Permainan Kartu 24	3
Algoritma Penyelesaian Permainan Kartu 24 dengan Pendekatan Brute Force	4
BAB 2 : Implementasi Algoritma dalam Bahasa Java	6
Main.java	6
App.java	6
outputFile.java	6
BAB 3 : Source Code Program	8
Repository Program	8
Source Code Program	8
BAB 4 : Masukan dan Luaran Program	13
Masukan dan Luaran Program	13

BAB 1

DESKRIPSI MASALAH DAN ALGORITMA

1.1. Algoritma Brute Force

Algoritma Brute Force adalah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu persoalan. Pendekatan Brute Force umumnya identik dengan melakukan enumerasi pada semua kemungkinan solusi yang ada, kemudian menerapkan batasan-batasan tertentu (bila diperlukan) sehingga kemudian terpilih lah solusi yang kita inginkan. Algoritma Brute Force biasanya didasarkan pada pernyataan pada persoalan (*problem statement*) dan definisi/konsep yang dilibatkan. Sifat-sifat dari algoritma Brute Force, yaitu sangat sederhana, langsung, dan jelas. Algoritma Brute Force juga dapat diterapkan untuk memecahkan sebagian besar masalah.

Kendati demikian, algoritma Brute Force umumnya tidak “cerdas” dan tidak mangkus karena ia membutuhkan volume komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Oleh karena itu, algoritma Brute Force lebih cocok untuk diaplikasikan pada persoalan yang ukuran masukannya kecil. Beberapa persoalan yang dapat diselesaikan dengan algoritma Brute Force, yaitu mencari elemen terbesar dan terkecil dari sebuah senarai, mencari elemen dengan nilai tertentu pada sebuah senarai, dan sebagainya.

1.2. Permainan Kartu 24

Permainan kartu 24 adalah permainan kartu aritmatika dengan tujuan mencari cara untuk mengubah 4 buah angka *random* sehingga mendapatkan hasil akhir sejumlah 24. Permainan ini cukup menarik perhatian banyak orang karena permainan ini dapat mengasah kemampuan berhitung secara cepat dan akurat. Permainan ini biasanya dimainkan dengan menggunakan kartu remi. Kartu remi terdiri dari 52 kartu yang terbagi menjadi empat suit (sekop, hati, keriting, dan wajik) yang masing-masing terdiri dari 13 kartu (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). Yang perlu diperhatikan hanyalah nilai kartu yang didapat (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). As bernilai 1, Jack bernilai 11, Queen bernilai 12, King bernilai 13, sedangkan kartu bilangan memiliki nilai dari bilangan itu sendiri. Pada awal permainan moderator atau salah satu pemain mengambil 4 kartu dari dek yang sudah dikocok secara random. Permainan berakhir ketika pemain berhasil menemukan solusi untuk membuat kumpulan nilainya menjadi

24. Pengubahan nilai tersebut dapat dilakukan menggunakan operasi dasar matematika penjumlahan (+), pengurangan (-), perkalian (\times), divisi (/) dan tanda kurung (). Tiap kartu harus digunakan tepat sekali dan urutan penggunaannya bebas.

1.3. Algoritma Penyelesaian Permainan Kartu 24 dengan Pendekatan Brute Force

Untuk menyelesaikan permainan kartu 24, penulis menggunakan pendekatan Brute Force. Berikut langkah-langkah penyelesaian kartu 24 dengan algoritma Brute Force.

1. Menerima masukan dari pengguna berupa pilihan cara men-*generate* kartu (*generate* dari masukan manual pengguna atau *generate* secara otomatis).
2. Menerima masukan 4 kartu dari pengguna jika memilih cara manual atau men-*generate* 4 kartu secara acak dengan menggunakan bantuan *library* pada *python*
3. Tampung 4 kartu tersebut di dalam struktur data ArrayList.
4. Memasukkan operator-operator yang digunakan (" $*$ ", " $/$ ", " $+$ ", " $-$ ", " $()$ ") ke dalam list
5. Melakukan iterasi untuk menemukan kemungkinan-kemungkinan solusi yang ada dengan menemukan hasil permutasi pada urutan masukan kartu, urutan penggunaan operator, serta kemungkinan peletakan kurung pada solusi. Hal ini dapat dilakukan dengan melakukan iterasi dengan for loop pada array yang berisi operator serta masukan kartu.
6. Kemungkinan peletakan kurung untuk 4 masukan bilangan dipersempit menjadi 5 bentuk kemungkinan, yaitu
 - a. Bentuk : $((x1 + x2) + x3) + x4$
 - b. Bentuk : $(x1 + x2) + (x3 + x4)$
 - c. Bentuk : $(x1 + (x2 + x3)) + x4$
 - d. Bentuk : $x1 + ((x2 + x3) + x4)$
 - e. Bentuk : $x1 + (x2 + (x3 + x4))$+ disini merupakan operator yang dapat pula berupa -, *, atau /

Kemungkinan peletakan kurung untuk melingkupi 3 bilangan seperti $(x1 + x2 + x3) + x4$ dihilangkan karena bentuknya kongruen dengan bentuk a ataupun c.

7. Pada saat melakukan iterasi, masukkan formula yang menghasilkan bilangan 24 ke dalam sebuah set. Elemen-elemen dari set inilah yang merupakan solusi dari permainan angka 24.

8. Melakukan iterasi pada elemen-elemen dari set untuk menghitung jumlah solusi yang ada dengan menampilkan solusi-solusi tersebut ke layar.

BAB 2

IMPLEMENTASI ALGORITMA DALAM BAHASA JAVA

Dalam pembuatan program ini, penulis menggunakan bahasa pemrograman Java. Struktur dari program ini terbagi menjadi 3 *file*, yaitu Main.java, App.java, dan outputFile.java.

2.1. Main.java

File ini berisi kode yang digunakan untuk menggabungkan fungsi-fungsi yang berada pada *file* lain dan menjalankan program utama.

2.2. App.java

Fungsi	Deskripsi
<pre>public static double operation(double num1, double num2, double opcode)</pre>	Fungsi ini berguna untuk memberikan hasil dari operasi 2 buah bilangan dengan suatu operator sesuai dengan kode opcode yang dimasukkan pengguna.
<pre>public static Set hitung(double x1, double x2, double x3, double x4, Set<String> formula)</pre>	Fungsi ini berisi Brute Force Algorithm yang digunakan untuk menyelesaikan persoalan game 24 ini. Algoritma Brute Force di sini berupa enumerasi kemungkinan penempatan input bilangan, operator, serta tanda kurung yang mungkin dapat menghasilkan bilangan 24.

2.3. outputFile.java

Fungsi	Deskripsi
<pre>public static void createFile()</pre>	Fungsi ini berguna untuk membuat sebuah file kosong di dalam folder test dengan

	nama yang diinputkan user
<pre> public static void solutionFile(Set<String> solutionFormula, Integer numOfSol, ArrayList<Integer> cards) </pre>	<p>Fungsi yang digunakan untuk menuliskan solusi dari 24 card game ke dalam file.</p>

BAB 3

SOURCE CODE PROGRAM

3.1. Link Github Repository

Link Repository dari Tugas Kecil Strategi Algoritma yang pertama ini dapat dilihat pada https://github.com/kandidagunawan/Tucil1_13521155

3.2. Source Code Program

3.2.1. Main.java

```
import java.util.Scanner;
import java.util.Set;
import java.util.ArrayList;
import java.util.HashSet;

public class Main extends App {
    Run | Debug
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Selamat datang di 24 game solver!");

        Boolean start = true;
        while (start) {
            System.out.println("Silahkan memilih jenis input yang akan kamu masukkan:");
            System.out.println("1. Generate input dari user");
            System.out.println("2. Generate input secara acak");
            System.out.print("Pilihan anda (1/2): ");
            String choice = scan.nextLine();
            while ((!choice.equals(anObject: "1") && !choice.equals(anObject: "2"))) {
                System.out.println("Masukan salah!");
                System.out.print("Pilihan anda (1/2): ");
                choice = scan.nextLine();
            }
            ArrayList<Integer> cards = new ArrayList<>();
            String input;
            int num;
            int numOfCards = 0;
            int numOfSpaces = 0;
            int numOfSolutions = 0;
            Boolean incorrect;
            String currCard = "";
            incorrect = false;
```



```

if (choice.equals(anObject: "1")) {
    do {
        System.out.println("Masukkan input dengan contoh format sebagai berikut: A 8 9 Q");
        input = scan.nextLine();
        numOfSpaces = 0;

        for (char curr : input.toCharArray()) {
            if (numOfSpaces > 3) {
                System.out.println(numOfSpaces);
                incorrect = true;
                System.out.println(" ");
                cards = new ArrayList<>();
                break;
            }
            if (curr == ' ') {
                numOfSpaces++;
                if ((currCard.equals(anObject: "A")) || (currCard.equals(anObject: "2")) || currCard.equals(anObject: "3")
                    || currCard.equals(anObject: "4")
                    || currCard.equals(anObject: "5")
                    || currCard.equals(anObject: "6") || currCard.equals(anObject: "7") || currCard.equals(anObject: "8")
                    || currCard.equals(anObject: "9")
                    || currCard.equals(anObject: "10") || currCard.equals(anObject: "J") || currCard.equals(anObject: "Q")
                    || currCard.equals(anObject: "K")) {
                    incorrect = false;
                    if (currCard.equals(anObject: "A")) {
                        cards.add(0: 1);
                    } else if (currCard.equals(anObject: "J")) {
                        cards.add(0: 11);
                    } else if (currCard.equals(anObject: "Q")) {
                        cards.add(0: 12);
                    } else if (currCard.equals(anObject: "K")) {
                        cards.add(0: 13);
                    } else {
                        cards.add(Integer.parseInt(currCard));
                    }
                    currCard = "";
                } else {
                    incorrect = true;
                    currCard = "";
                    cards = new ArrayList<>();
                    break;
                }
            } else {
                currCard += curr;
            }
        }
    }
    if (numOfSpaces != 3) {
        incorrect = true;
        currCard = "";
        cards = new ArrayList<>();
    } else {
        if ((currCard.equals(anObject: "A")) || (currCard.equals(anObject: "2")) || currCard.equals(anObject: "3")
            || currCard.equals(anObject: "4")
            || currCard.equals(anObject: "5")
            || currCard.equals(anObject: "6") || currCard.equals(anObject: "7") || currCard.equals(anObject: "8")
            || currCard.equals(anObject: "9")
            || currCard.equals(anObject: "10") || currCard.equals(anObject: "J") || currCard.equals(anObject: "Q")
            || currCard.equals(anObject: "K")) {
            incorrect = false;
        } else {
            incorrect = true;
            currCard = "";
            cards = new ArrayList<>();
        }
    }
    if (incorrect == false) {
        if (currCard.equals(anObject: "A")) {
            cards.add(0: 1);
        } else if (currCard.equals(anObject: "J")) {
            cards.add(0: 11);
        } else if (currCard.equals(anObject: "Q")) {
            cards.add(0: 12);
        } else if (currCard.equals(anObject: "K")) {
            cards.add(0: 13);
        } else {
            cards.add(Integer.parseInt(currCard));
        }
        currCard = "";
    }
}
}

```

```

        if (incorrect) {
            System.out.println(x: "Masukan tidak sesuai!");
        }

    } while (incorrect == true);

}

else if (choice.equals(anObject: "2")) {
    int random;
    for (int i = 1; i <= 4; i++) {
        random = (int) Math.floor(Math.random() * 13 + 1);
        System.out.println("Kartu ke-" + i + " yang terpilih adalah " + random);
        cards.add(random);
    }
}

long startTime = System.currentTimeMillis();
Set<String> formula = new HashSet<String>();

for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        for (int k = 0; k < 4; k++) {
            for (int l = 0; l < 4; l++) {
                if ((i != j) && (i != k) && (i != l) && (j != k) && (j != l) && (k != l)) {
                    formula = App.hitung(cards.get(i), cards.get(j), cards.get(k), cards.get(l),
                        formula);
                }
            }
        }
    }
}

long finishTime = System.currentTimeMillis();
for (String f : formula) {
    numOfSolutions++;
}

System.out.println(numOfSolutions + " solutions found");
for (String f : formula) {
    System.out.println(f);
}

long duration = finishTime - startTime;
System.out.println("Time taken: " + duration + " ms");
System.out.println(x: "Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)");
String isfileOut = scan.nextLine();
while (!isfileOut.equals(anObject: "y") && !isfileOut.equals(anObject: "n")) {
    System.out.println(x: "Masukan salah!");
    System.out.println(x: "Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)");
    isfileOut = scan.nextLine();
}

if (isfileOut.equals(anObject: "y")) {
    outputFile.solutionFile(formula, numOfSolutions, cards);
}

System.out.println(x: "Apakah kamu ingin keluar dari program ini? (y/n)");
String end = scan.nextLine();
while (!end.equals(anObject: "y") && !end.equals(anObject: "n")) {
    System.out.println(x: "Masukan salah!");
    System.out.println(x: "Apakah kamu ingin keluar dari program ini? (y/n)");
    end = scan.nextLine();
}

if (end.equals(anObject: "y")) {
    start = false;
} else {
    start = true;
}

}

}

```

3.2.2. App.java

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Set;

public class App {
    public static double operation(double num1, double num2, double opcode) {
        if (opcode == 0) {
            return (num1 + num2);
        } else if (opcode == 1) {
            return (num1 - num2);
        } else if (opcode == 2) {
            return (num1 * num2);
        } else {
            if (num2 == 0) {
                return -99999;
            } else {
                return (num1 / num2);
            }
        }
    }
}
```

```
public static Set hitung(double x1, double x2, double x3, double x4, Set<String> formula) {
    String[] operators = { "+", "-", "*", "/" };
    double hasil = 0;
    double temp = 0;
    int count = 0;
    String op1, op2, op3;
    String formulaString = "";
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            for (int k = 0; k < 4; k++) {
                for (int l = 0; l < 5; l++) {
                    switch (l) {
                        case 0:
                            // bentuk : ((2+4)+(5)+6
                            hasil = operation(x1, x2, i);
                            hasil = operation(hasil, x3, j);
                            hasil = operation(hasil, x4, k);
                            formulaString = "(" + (int) x1 + operators[i] + (int) x2 + ")" + operators[j]
                                + (int) x3 + ")"
                                + operators[k] + (int) x4;
                            break;
                        case 1:
                            // bentuk : (2+4)+(5+6)
                            hasil = operation(x1, x2, i);
                            temp = operation(x3, x4, k);
                            hasil = operation(hasil, temp, j);
                            formulaString = "(" + (int) x1 + operators[i] + (int) x2 + ")" + operators[j] + "("
                                + (int) x3
                                + operators[k] + (int) x4 + ")";
                            break;
                        case 2:
                            // bentuk : (2+(4+5))+6
                            temp = operation(x2, x3, j);
                            hasil = operation(x1, temp, i);
                            hasil = operation(hasil, x4, k);
                            formulaString = "(" + (int) x1 + operators[i] + "(" + (int) x2 + operators[j] + (int) x3
                                + ")"
                                + operators[k] + (int) x4;
                            break;
                        case 3:
                            // bentuk : 2+((4+5)+6)
                            temp = operation(x2, x3, j);
                            hasil = operation(temp, x4, k);
                            hasil = operation(x1, hasil, i);
                            formulaString = (int) x1 + operators[i] + "(" + (int) x2 + operators[j] + (int) x3
                                + ")" + operators[k]
                                + (int) x4 + ")";
                            break;
                        case 4:
                            // bentuk : 2+(4+(5+6))
                            hasil = operation(x3, x4, k);
                            hasil = operation(x2, hasil, j);
                            hasil = operation(x1, hasil, i);
                            formulaString = (int) x1 + operators[i] + "(" + (int) x2 + operators[j] + "(" + (int) x3
                                + operators[k]
                                + (int) x4 + ")";
                            break;
                    }
                }
            }
        }
    }
    if (hasil == 24) {
        formula.add(formulaString);
    }
}

return formula;
}
```

3.2.3. outputFile.java

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.Set;

public class outputFile {
    private static String filename;
    static Scanner scan = new Scanner(System.in);

    public static void createFile() {
        try {
            System.out.println("Masukkan nama file: ");
            filename = scan.nextLine();
            File solution = new File("test\\" + filename);
            while (!solution.createNewFile()) {
                System.out.println("Nama file yang kamu inginkan sudah ada!");
                System.out.println("Coba masukkan nama file yang lain: ");
                filename = scan.nextLine();
                solution = new File("test/" + filename);
            }
            System.out.println("File created: " + solution.getName());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void solutionFile(Set<String> solutionFormula, Integer numOfSol, ArrayList<Integer> cards) {
        try {
            createFile();
            FileWriter writer = new FileWriter("test/" + filename);
            writer.write("SOLUSI DARI 24 GAME SOLVER \n");
            writer.write(
                "Masukan : " + cards.get(index: 0) + " " + cards.get(index: 1) + " " + cards.get(index: 2) + " " + cards.get(index: 3) + "\n");
            writer.write(numOfSol + " solutions found" + "\n");
            for (String f : solutionFormula) {
                writer.write(f + "\n");
            }
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

BAB 4

MASUKAN DAN LUARAN PROGRAM

1. test1.txt

```
Selamat datang di 24 game solver!  
Silahkan memilih jenis input yang akan kamu masukkan:  
1. Generate input dari user  
2. Generate input secara acak  
Pilihan anda (1/2): 1  
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q  
J Q K 3  
0 solutions found  
Time taken: 8 ms  
Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)  
y  
Masukkan nama file:  
test1.txt  
File created: test1.txt  
Apakah kamu ingin keluar dari program ini? (y/n)  
n
```

```
SOLUSI DARI 24 GAME SOLVER  
Masukan : 11 12 13 3  
0 solutions found
```

2. test2.txt

```
Silahkan memilih jenis input yang akan kamu masukkan:  
1. Generate input dari user  
2. Generate input secara acak  
Pilihan anda (1/2): 1  
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q  
2 8 10 A  
6 solutions found  
10-(2*(1-8))  
(2*(8-1))+10  
((8-1)*2)+10  
10-((1-8)*2)  
10+((8-1)*2)  
10+(2*(8-1))  
Time taken: 21 ms  
Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)  
y  
Masukkan nama file:  
test2.txt  
File created: test2.txt  
Apakah kamu ingin keluar dari program ini? (y/n)  
n
```

```
SOLUSI DARI 24 GAME SOLVER
```

```
Masukan : 2 8 10 1
```

```
6 solutions found
```

```
 $10 - (2 * (1 - 8))$ 
```

```
 $(2 * (8 - 1)) + 10$ 
```

```
 $((8 - 1) * 2) + 10$ 
```

```
 $10 - ((1 - 8) * 2)$ 
```

```
 $10 + ((8 - 1) * 2)$ 
```

```
 $10 + (2 * (8 - 1))$ 
```

3. test3.txt

```
Silahkan memilih jenis input yang akan kamu masukkan:
```

```
1. Generate input dari user
```

```
2. Generate input secara acak
```

```
Pilihan anda (1/2): 2
```

```
Kartu ke-1 yang terpilih adalah 13
```

```
Kartu ke-2 yang terpilih adalah 8
```

```
Kartu ke-3 yang terpilih adalah 12
```

```
Kartu ke-4 yang terpilih adalah 13
```

```
0 solutions found
```

```
Time taken: 8 ms
```

```
Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)
```

```
y
```

```
Masukkan nama file:
```

```
test3.txt
```

```
File created: test3.txt
```

```
Apakah kamu ingin keluar dari program ini? (y/n)
```

```
n
```

```
SOLUSI DARI 24 GAME SOLVER
```

```
Masukan : 13 8 12 13
```

```
0 solutions found
```

4. test4.txt

```
Silahkan memilih jenis input yang akan kamu masukkan:  
1. Generate input dari user  
2. Generate input secara acak  
Pilihan anda (1/2): 2  
Kartu ke-1 yang terpilih adalah 9  
Kartu ke-2 yang terpilih adalah 13  
Kartu ke-3 yang terpilih adalah 1  
Kartu ke-4 yang terpilih adalah 7  
8 solutions found  
(13-9)*(7-1)  
(9-13)*(1-7)  
(7-9)*(1-13)  
(1-13)*(7-9)  
(9-7)*(13-1)  
(1-7)*(9-13)  
(13-1)*(9-7)  
(7-1)*(13-9)  
Time taken: 8 ms  
Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)  
y  
Masukkan nama file:  
test4.txt  
File created: test4.txt
```

SOLUSI DARI 24 GAME SOLVER

Masukan : 9 13 1 7

8 solutions found

(13-9)*(7-1)

(9-13)*(1-7)

(7-9)*(1-13)

(1-13)*(7-9)

(9-7)*(13-1)

(1-7)*(9-13)

(13-1)*(9-7)

(7-1)*(13-9)

5. test 5.txt

```
Silahkan memilih jenis input yang akan kamu masukkan:
1. Generate input dari user
2. Generate input secara acak
Pilihan anda (1/2): 1
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
8 9 6 4
36 solutions found
6+((9/4)*8)
((9-8)*6)*4
(4*(9-8))*6
((9-8)*4)*6
6+((9*8)/4)
(8*(9/4))+6
((8*9)/4)+6
(9/(4/8))+6
6+((8/4)*9)
(4/(9-8))*6
((8/4)*9)+6
4*((9-8)*6)
(8/(4/9))+6
(6*(9-8))*4
6+(8*(9/4))
(6*4)*(9-8)
6+(9/(4/8))
6+(8/(4/9))
(9-8)*(4*6)
(9-8)*(6*4)
6+(9*(8/4))
4/((9-8)/6)
6*((9-8)*4)
((9*8)/4)+6
(6/(9-8))*4
6/((9-8)/4)
(4*6)*(9-8)
(4*6)/(9-8)
4*(6*(9-8))
((9/4)*8)+6
4*(6/(9-8))
(9*(8/4))+6
6*(4/(9-8))
(6*4)/(9-8)
6+((8*9)/4)
6*(4*(9-8))
Time taken: 6 ms
```


SOLUSI DARI 24 GAME SOLVER

Masukan : 8 9 6 4

36 solutions found

$6 + ((9/4) * 8)$

$((9-8) * 6) * 4$

$(4 * (9-8)) * 6$

$((9-8) * 4) * 6$

$6 + ((9 * 8) / 4)$

$(8 * (9/4)) + 6$

$((8 * 9) / 4) + 6$

$(9 / (4/8)) + 6$

$6 + ((8/4) * 9)$

$(4 / (9-8)) * 6$

$((8/4) * 9) + 6$

$4 * ((9-8) * 6)$

$(8 / (4/9)) + 6$

$(6 * (9-8)) * 4$

$6 + (8 * (9/4))$

$(6 * 4) * (9-8)$

$6 + (9 / (4/8))$

$6 + (8 / (4/9))$

$(9-8) * (4 * 6)$

$(9-8) * (6 * 4)$

$6 + (9 * (8/4))$

$4 / ((9-8) / 6)$

$6 * ((9-8) * 4)$

$((9 * 8) / 4) + 6$

$(6 / (9-8)) * 4$

$6 / ((9-8) / 4)$

$(4 * 6) * (9-8)$

$(4 * 6) / (9-8)$

$4 * (6 * (9-8))$

$((9/4) * 8) + 6$

$4 * (6 / (9-8))$

$(9 * (8/4)) + 6$

$6 * (4 / (9-8))$

$(6 * 4) / (9-8)$

$6 + ((8 * 9) / 4)$

$6 * (4 * (9-8))$

```

Silahkan memilih jenis input yang akan kamu masukkan:
1. Generate input dari user
2. Generate input secara acak
Pilihan anda (1/2): 2
Kartu ke-1 yang terpilih adalah 9
Kartu ke-2 yang terpilih adalah 3
Kartu ke-3 yang terpilih adalah 9
Kartu ke-4 yang terpilih adalah 10
8 solutions found
((9+9)-10)*3
3*((9+9)-10)
3*(9+(9-10))
3*((9-10)+9)
3*(9-(10-9))
((9-10)+9)*3
(9-(10-9))*3
(9+(9-10))*3
Time taken: 15 ms
Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)
y
Masukkan nama file:
test6.txt
File created: test6.txt

```

SOLUSI DARI 24 GAME SOLVER

```

Masukan : 9 3 9 10
8 solutions found
((9+9)-10)*3
3*((9+9)-10)
3*(9+(9-10))
3*((9-10)+9)
3*(9-(10-9))
((9-10)+9)*3
(9-(10-9))*3
(9+(9-10))*3

```

7. Test7.txt

```
Pilihan anda (1/2): 1
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
9 3 3 2
42 solutions found
(3+(2*9))+3
9+(3*(3+2))
(2*9)+(3+3)
9+((3+2)*3)
3+(3*(9-2))
((9-3)+2)*3
((2+9)-3)*3
9+(3*(2+3))
3+(3+(2*9))
((2*9)+3)+3
3*((9+2)-3)
3-(3*(2-9))
3+((9-2)*3)
(3*(9-2))+3
3*(9-(3-2))
(9-(3-2))*3
((2+3)*3)+9
((9-2)*3)+3
9+((2+3)*3)
3*(2-(3-9))
((9+2)-3)*3
((9*2)+3)+3
((2-3)+9)*3
3*(9+(2-3))
(3*(3+2))+9
3*((2+9)-3)
3+((9*2)+3)
3*((9-3)+2)
3-((2-9)*3)
(2-(3-9))*3
(3+(9*2))+3
(9*2)+(3+3)
3*(2+(9-3))
(2+(9-3))*3
3*((2-3)+9)
(9+(2-3))*3
3+(3+(9*2))
(3+3)+(2*9)
(3+3)+(9*2)
3+((2*9)+3)
((3+2)*3)+9
(3*(2+3))+9
Time taken: 8 ms
```

SOLUSI DARI 24 GAME SOLVER

Masukan : 9 3 3 2

42 solutions found

$$(3+(2*9))+3$$

$$9+(3*(3+2))$$

$$(2*9)+(3+3)$$

$$9+((3+2)*3)$$

$$3+(3*(9-2))$$

$$((9-3)+2)*3$$

$$((2+9)-3)*3$$

$$9+(3*(2+3))$$

$$3+(3+(2*9))$$

$$((2*9)+3)+3$$

$$3*((9+2)-3)$$

$$3-(3*(2-9))$$

$$3+((9-2)*3)$$

$$(3*(9-2))+3$$

$$3*(9-(3-2))$$

$$(9-(3-2))*3$$

$$((2+3)*3)+9$$

$$((9-2)*3)+3$$

$$9+((2+3)*3)$$

$$3*(2-(3-9))$$

$$((9+2)-3)*3$$

$$((9*2)+3)+3$$

$$((2-3)+9)*3$$

$$3*(9+(2-3))$$

$$(3*(3+2))+9$$

$$3*((2+9)-3)$$

$$3+((9*2)+3)$$

$$3*((9-3)+2)$$

$$3-((2-9)*3)$$

$$(2-(3-9))*3$$

$$(3+(9*2))+3$$

$$(9*2)+(3+3)$$

$$3*(2+(9-3))$$

$$(2+(9-3))*3$$

$$3*((2-3)+9)$$

$$(9+(2-3))*3$$

$$3+(3+(9*2))$$

$$(3+3)+(2*9)$$

$$(3+3)+(9*2)$$

$$3+((2*9)+3)$$

$$((3+2)*3)+9$$

$$(3*(2+3))+9$$

8. Test8.txt

```
Pilihan anda (1/2): 1
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
Q 7 4 3
30 solutions found
((7*12)/4)+3
((12/4)*7)+3
((12*7)/4)+3
(7/(4/12))+3
3+(7/(4/12))
(12/(3/7))-4
(7*(12/4))+3
(7/(3/12))-4
(7*(12/3))-4
3+(7*(12/4))
(4*7)-(12/3)
3+((7/4)*12)
((7/4)*12)+3
((7/3)*12)-4
(12*(7/4))+3
((12/3)*7)-4
((7*12)/3)-4
((12*7)/3)-4
3+(12/(4/7))
(3*7)+(12/4)
3+(12*(7/4))
(12*(7/3))-4
(7*3)+(12/4)
3+((12*7)/4)
(7*4)-(12/3)
(12/(4/7))+3
3+((7*12)/4)
(12/4)+(3*7)
(12/4)+(7*3)
3+((12/4)*7)
```

9. Test9.txt

```
Silahkan memilih jenis input yang akan kamu masukkan:
1. Generate input dari user
2. Generate input secara acak
Pilihan anda (1/2): 2
Kartu ke-1 yang terpilih adalah 9
Kartu ke-2 yang terpilih adalah 9
Kartu ke-3 yang terpilih adalah 9
Kartu ke-4 yang terpilih adalah 13
0 solutions found
Time taken: 0 ms
```

```
SOLUSI DARI 24 GAME SOLVER
Masukan : 9 9 9 13
0 solutions found
```

10. test10.txt

Silahkan memilih jenis input yang akan kamu masukkan:

1. Generate input dari user
2. Generate input secara acak

Pilihan anda (1/2): 2

Kartu ke-1 yang terpilih adalah 4

Kartu ke-2 yang terpilih adalah 10

Kartu ke-3 yang terpilih adalah 4

Kartu ke-4 yang terpilih adalah 5

10 solutions found

$4 + ((10 - 5) * 4)$

$4 + (4 * (10 - 5))$

$4 - ((5 - 10) * 4)$

$((10 - 5) * 4) + 4$

$((10 / 5) + 4) * 4$

$(4 * (10 - 5)) + 4$

$4 * (4 + (10 / 5))$

$(4 + (10 / 5)) * 4$

$4 * ((10 / 5) + 4)$

$4 - (4 * (5 - 10))$

Time taken: 7 ms

SOLUSI DARI 24 GAME SOLVER

Masukan : 4 10 4 5

10 solutions found

$4 + ((10 - 5) * 4)$

$4 + (4 * (10 - 5))$

$4 - ((5 - 10) * 4)$

$((10 - 5) * 4) + 4$

$((10 / 5) + 4) * 4$

$(4 * (10 - 5)) + 4$

$4 * (4 + (10 / 5))$

$(4 + (10 / 5)) * 4$

$4 * ((10 / 5) + 4)$

$4 - (4 * (5 - 10))$

11. HANDLING INPUT ERROR

Silahkan memilih jenis input yang akan kamu masukkan:

1. Generate input dari user
2. Generate input secara acak

Pilihan anda (1/2): 4

Masukan salah!

Pilihan anda (1/2): dafjlds

Masukan salah!

Pilihan anda (1/2):

Masukan salah!

Pilihan anda (1/2): 1

Masukkan input dengan contoh format sebagai berikut: A 8 9 Q

```

Silahkan memilih jenis input yang akan kamu masukkan:
1. Generate input dari user
2. Generate input secara acak
Pilihan anda (1/2): 1
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
6 7 A 8
0 solutions found
Time taken: 5 ms
Apakah kamu ingin menyimpan solusi dalam bentuk file? (y/n)
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
ASDKALFJA
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
3 4 5
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
4 5
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
3 18 7 6 5
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
5 12 A Q
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
A Q 7 8
20 solutions found
(12-8)*(7-1)
((1-7)+8)*12
(7-1)*(12-8)
12*(8+(1-7))
12*(8-(7-1))
((1+8)-7)*12
12*(1+(8-7))
12*((8-7)+1)
12*((1-7)+8)
12*(1-(7-8))
(8+(1-7))*12
(8-12)*(1-7)
12*((8+1)-7)
(1-(7-8))*12
((8-7)+1)*12
(8-(7-1))*12
(1+(8-7))*12
(1-7)*(8-12)
12*((1+8)-7)
((8+1)-7)*12
Time taken: 2 ms

```

```
Apakah kamu ingin keluar dari program ini? (y/n)
adjsflkajd
Masukan salah!
Apakah kamu ingin keluar dari program ini? (y/n)
N
Masukan salah!
Apakah kamu ingin keluar dari program ini? (y/n)
Y
Masukan salah!
Apakah kamu ingin keluar dari program ini? (y/n)
n
Silahkan memilih jenis input yang akan kamu masukkan:
1. Generate input dari user
2. Generate input secara acak
Pilihan anda (1/2): █
```

```
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
damsflka
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
A 7 8
Masukan tidak sesuai!
Masukkan input dengan contoh format sebagai berikut: A 8 9 Q
A 2 1234 4
Masukan tidak sesuai!
```

LAMPIRAN

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