LAPORAN TUGAS KECIL 1

IF2211/Strategi Algoritma

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force

Muhammad Farrel Danendra Rachim

13521048

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

1. Algoritma Brute Force

Algoritma *brute force* adalah pendekatan yang lempang untuk memecahkan suatu persoalan. Algoritma ini memecahkan persoalan dengan cara yang sangat sederhana dan *to the point*, biasanya dengan enumerasi semua kemungkinan solusi dan mengambil solusi terbaik. Pada umumnya, algoritma *brute force* tidak mangkus, karena membutuhkan volume komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Oleh karena itu, algoritma *brute force* lebih cocok digunakan untuk persoalan dengan ukuran masukan (n) kecil. Walaupun begitu, hampir semua persoalan dapat diselesaikan dengan algoritma *brute force*, dan pendekatan ini dijamin akan memperoleh solusi apabila solusi tersebut ada.

Permainan “Kartu 24” adalah sebuah permainan kartu remi yang memanfaatkan operasi aritmatika yang bertujuan mencari berbagai macam cara untuk mengubah 4 buah angka random sehingga mendapatkan hasil akhir sejumlah 24. As (A) bernilai 1, Jack bernilai 11, Queen bernilai 12, King bernilai 13, sedangkan kartu bilangan memiliki nilai dari bilangan itu sendiri. Permainan ini menggunakan operasi penjumlahan, pengurangan, perkalian, pembagian, dan tanda kurung, serta memperhatikan urutan nilai keempat nilai kartu, urutan operator, dan pengelompokan kurung yang mungkin.

Penulis menggunakan algoritma *brute force* untuk menyelesaikan permasalahan kartu 24, dengan langkah-langkah berikut:

1. Menerima input dari pengguna yang telah divalidasi berupa empat buah kartu (tidak harus berbeda semua), atau meng-generate empat buah kartu secara random.
2. Input a, b, c, d ditampung pada fungsi final24. Fungsi ini mengimplementasikan fungsi lain bernama findsolution24 yang berguna untuk menyimpan string-string berupa solusi yang mungkin dari sebuah urutan angka a, b, c, d tertentu, agar memperoleh angka 24 (untuk operasi yang melibatkan pembagian, angka-angka tersebut diubah terlebih dahulu menjadi float agar dilakukan real division).
3. Fungsi final24 berguna mengimplementasikan findsolution24 dengan permutasi nilai a, b, c, dan d yang berbeda-beda dengan memperhatikan apakah ada nilai variabel yang sama untuk menghindari adanya duplikat.
4. Hasil akhir dari fungsi final24 berupa array of string yang berisi solusi dari semua kemungkinan urutan angka, operator, dan pengelompokan kurung. Isi array tersebut akan ditampilkan ke layar, dan pengguna bisa menyimpan hasil solusi tersebut ke dalam file .txt.

2. Source Code

Program ini terbagi menjadi 3 file: operation.java, utils.java, dan main.java.

a. operation.java
    i. Daftar library: Penulis memakai library Random untuk mengacak angka, Scanner untuk mengambil input dari pengguna, dan IO untuk menulis File.

```java
import java.util.Random;
import java.util.Scanner;
import java.io.FileWriter;
import java.io.IOException;
```

ii. print functions: mem-print berbagai tipe data

```java
public static void print(String str) {
    // Menulis str tanpa newline
    System.out.print(str);
}
public static void println() {
    // Menulis new line
    System.out.println();
}

public static void printlnstr(String str) {
    // Menulis str dengan newline
    System.out.println(str);
}

public static void printlnint(int integer) {
    // Menulis str bertipe integer denangan new line
    System.out.println(integer);
}

public static void printlndou(double dou) {
    // Menulis str bertipe double denangan new line
    System.out.println(dou);
}

public static void printf(String str, Object... args) {
    // Menulis str dengan beberapa format argumen
    System.out.printf(str, args);
}

public static void printstrarray(String[] liststring) {
    // Mencetak semua elemen dalam array of strings
    for (int i=0; i < liststring.length; i++) {
        printlnstr(liststring[i]);
    }
}
```

iii. int pilihanMenu(String s): Menerima input karakter dan mengembalikan true jika input berupa karakter s

```java
public static boolean pilihanMenu(String s) {
    // Menerima input pilihan berupa karakter s
    String input = "";
    Scanner scanner = new Scanner(System.in);
    print(str: "> ");
    input = scanner.nextLine();
    if (input.equals(s)) {
        return true;
    } else {
        return false;
    }
}
```

iv. double[] convertTodouble(int a, int b, int c, int d): Mengubah angka a, b, c, d menjadi double.

```java
public static double[] convertTodouble(int a, int b, int c, int d) {
    // Mengubah angka a, b, c, d menjadi double
    double[] arraydouble = new double[4];
    arraydouble[0] = (double) a;
    arraydouble[1] = (double) b;
    arraydouble[2] = (double) c;
    arraydouble[3] = (double) d;
    return arraydouble;
}
```

v. String intToString(int a): Mengubah angka 1-13 menjadi kartu yang berkorespondensi dengan nilainya

```java
public static String intToString(int a) {
    // Mengubah angka 1-13 menjadi kartu yang berkorespondensi dengan nilainya
    String result = "";
    if (a == 1) {
        result = "A";
    } else if (a == 11) {
        result = "J";
    } else if (a == 12) {
        result = "Q";
    } else if (a == 13) {
        result = "K";
    } else {
        result = String.valueOf(a);
    }
    return result;
}
```

vi. String[] concatStr(String[] list, String a): Append elemen sebagai index terakhir array

```java
public static String[] concatStr(String[] list, String a) {
    // Append elemen sebagai index terakhir array
    String[] new_arr = new String[list.length + 1];
    for (int i = 0; i < list.length; i++) {
        new_arr[i] = list[i];
    }
    new_arr[new_arr.length - 1] = a;
    return new_arr;

}
```

vii. String[] mergeArrays(String[] list1, String[] list2): Meng-konkat dua buah array

```java
public static String[] mergeArrays(String[] list1, String[] list2) {
    // Meng-konkat dua buah array
    String[] newArray = new String[list1.length + list2.length];
    int index = 0;
    for (String i : list1) {
        newArray[index] = i;
        index++;
    }
    for (String i : list2) {
        newArray[index] = i;
        index++;
    }
    return newArray;
}
```

viii. int[] RandomNumbers(): Meng-generate angka 1-13 dengan urutan acak dalam sebuah array

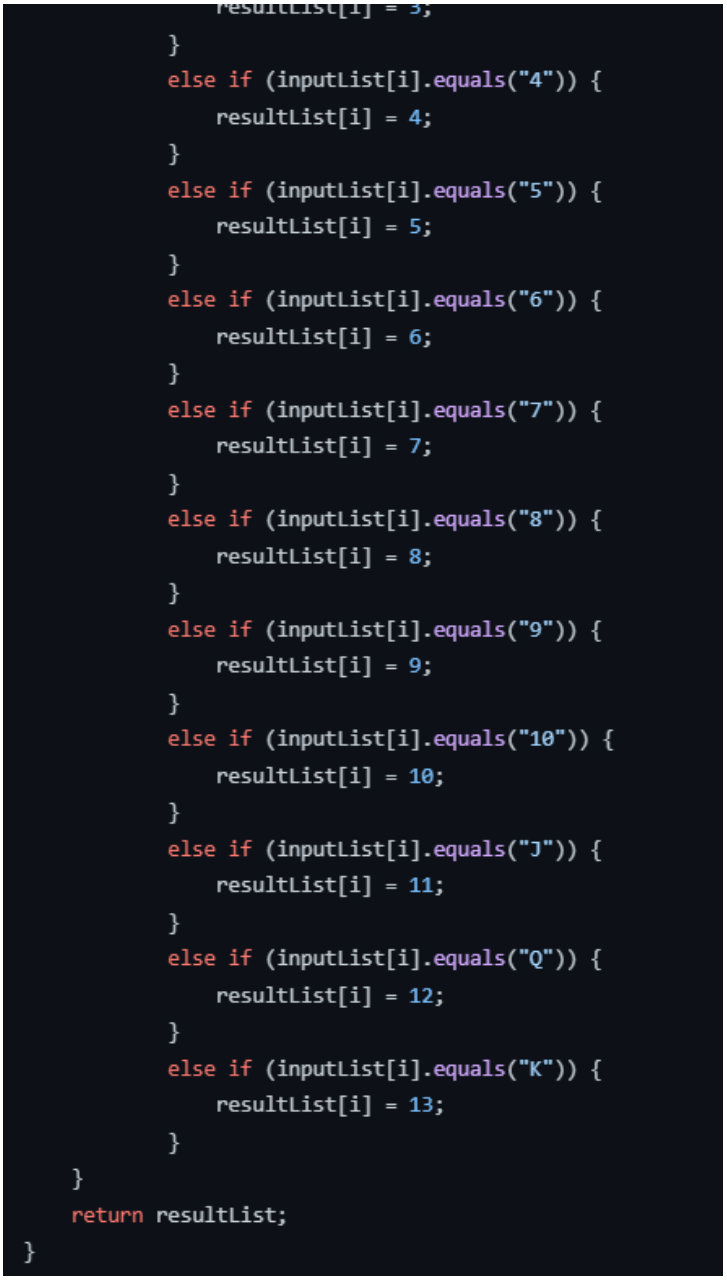
```java
public static int[] RandomNumbers() {
    // Meng-generate angka 1-13 dengan urutan acak dalam sebuah array
    int[] newlist = new int[4];
    Random rand = new Random();
    for (int i=0; i < 4; i++) {
        newlist[i] = rand.nextInt(1, 13);
    }
    return newlist;
}
```

ix. boolean isSubset(String[] list1, String[] list2): Menentukan apakah list2 merupakan subset list1

```java
public static boolean isSubset(String[] list1, String[] list2) {
    // Menentukan apakah list2 merupakan subset list1
    int i = 0;
            int j = 0;
            for (i = 0; i < list2.length; i++) {
                for (j = 0; j < list1.length; j++) {
                    if (list2[i].equals(list1[j])) {
                        break;
                    }
                }
                if (j == list1.length)
                    return false;
            }
            return true;
}
```

x. int[] manualInput(): Input pilihan kartu manual dari pengguna, output nilai dari kartu tersebut (sudah divalidasi)

```java
public static int[] manualInput() {
    // Input pilihan kartu manual dari pengguna, output nilai dari kartu tersebut (sudah divalidasi)
    boolean validInput = false;
    Scanner sc = new Scanner(System.in);
    String[] cardSet = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};
    String userInput = "";
    while (!validInput) {
        print("Masukkan pilihan kartu: ");
        print("> ");
        userInput = sc.nextLine();
        String[] inputList = userInput.split(" ");
        if (inputList.length != 4) {
            printlnstr("Perlu dimasukkan 4 kartu.");
        } else {
            if (isSubset(cardSet, inputList)) {
                validInput = true;
            } else {
                printlnstr("Masukan tidak valid, coba lagi.");
            }
        }
    }
    String[] inputList = userInput.split(" ");
    int[] resultList = new int[4];
    for (int i = 0; i < 4; i++) {
            if (inputList[i].equals("A")) {
                resultList[i] = 1;
            }
            else if (inputList[i].equals("2")) {
                resultList[i] = 2;
            }
            else if (inputList[i].equals("3")) {
                resultList[i] = 3;
            }
            else if (inputList[i].equals("4")) {
                resultList[i] = 4;
            }
            else if (inputList[i].equals("5")) {
                resultList[i] = 5;
```



```
                resultList[i] = 3;
            }
            else if (inputList[i].equals("4")) {
                resultList[i] = 4;
            }
            else if (inputList[i].equals("5")) {
                resultList[i] = 5;
            }
            else if (inputList[i].equals("6")) {
                resultList[i] = 6;
            }
            else if (inputList[i].equals("7")) {
                resultList[i] = 7;
            }
            else if (inputList[i].equals("8")) {
                resultList[i] = 8;
            }
            else if (inputList[i].equals("9")) {
                resultList[i] = 9;
            }
            else if (inputList[i].equals("10")) {
                resultList[i] = 10;
            }
            else if (inputList[i].equals("J")) {
                resultList[i] = 11;
            }
            else if (inputList[i].equals("Q")) {
                resultList[i] = 12;
            }
            else if (inputList[i].equals("K")) {
                resultList[i] = 13;
            }
        }
        return resultList;
    }
```

xi. boolean writeSolutionToFile(String directory, String[] solutionList, String a, String b, String c, String d): Tulis solusi ke file .txt

```java
public static boolean writeSolutionToFile(String directory, String[] solutionList, String a, String b, String c, String d) {
    // Tulis solusi ke File
    try {
        FileWriter writer = new FileWriter(directory);
        writer.write("Kartu: " + a + " " + b + " " + c + " " + d + "\n");
        if (solutionList.length == 0) {
            writer.write("Tidak ada solusi.\n");
        } else {
            writer.write("Banyak solusi: " + solutionList.length + "\n");
            for (int i=0; i < solutionList.length; i++) {
                writer.write(solutionList[i] + "\n");
            }
        }
        writer.close();
        println("Hasil sudah situliskan kepada file.");
        return true;
    } catch (IOException e) {
        // TODO: handle exception
        println("Something went wrong. Try again.");
        return false;
    }
}
```

b. operation.java

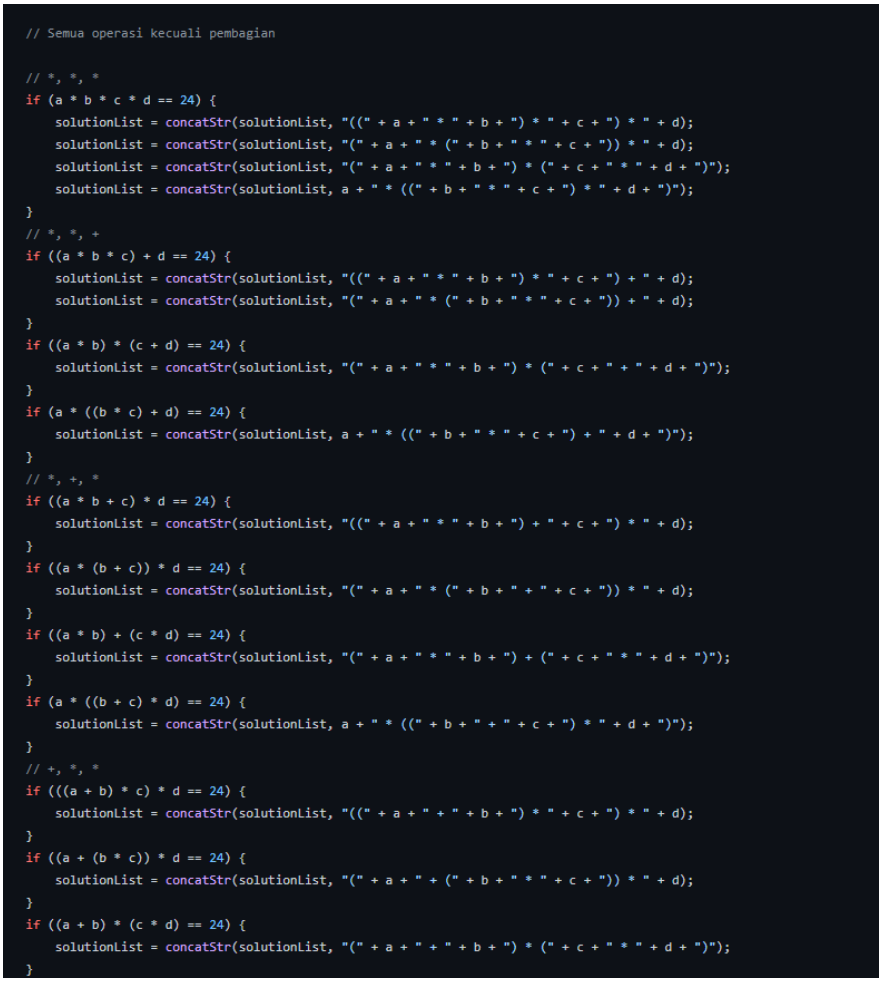
i. String[] findsolution24(int a, int b, int c, int d)

Penjumlahan dan pengurangan saja:

```java
public class operation extends utils {
    public static String[] findsolution24(int a, int b, int c, int d) {
        String[] solutionList = {};

        // Penjumlahan & pengurangan

        // +, +, +
        if (a + b + c + d == 24) {
            solutionList = concatStr(solutionList, "((" + a + " + " + b + ") + " + c + ") + " + d);
            solutionList = concatStr(solutionList, "(" + a + " + (" + b + " + " + c + ")) + " + d);
            solutionList = concatStr(solutionList,"(" + a + " + " + b + ") + (" + c + " + " + d + ")");
            solutionList = concatStr(solutionList, a + " + ((" + b + " + " + c + ") + " + d + ")");
        }
        // +, +, -
        if (a + b + c - d == 24) {
            solutionList = concatStr(solutionList, "((" + a + " + " + b + ") + " + c + ") - " + d);
            solutionList = concatStr(solutionList, "(" + a + " + (" + b + " + " + c + ")) - " + d);
            solutionList = concatStr(solutionList, "(" + a + " + " + b + ") + (" + c + " - " + d + ")");
            solutionList = concatStr(solutionList, a + " + ((" + b + " + " + c + ") - " + d + ")");
        }
        // +, -, +
        if (a + b - c + d == 24) {
            solutionList = concatStr(solutionList, "((" + a + " + " + b + ") - " + c + ") + " + d);
            solutionList = concatStr(solutionList, "(" + a + " + (" + b + " - " + c + ")) + " + d);
            solutionList = concatStr(solutionList, "(" + a + " + " + b + ") - (" + c + " - " + d + ")");
            solutionList = concatStr(solutionList, a + " + ((" + b + " - " + c + ") + " + d + ")");
        }
        // -, +, +
        if (a - b + c + d == 24) {
            solutionList = concatStr(solutionList, "((" + a + " - " + b + ") + " + c + ") + " + d);
            solutionList = concatStr(solutionList, "(" + a + " - (" + b + " - " + c + ")) + " + d);
            solutionList = concatStr(solutionList, "(" + a + " - " + b + ") + (" + c + " + " + d + ")");
            solutionList = concatStr(solutionList, a + " - ((" + b + " - " + c + ") - " + d + ")");
        }
        // +, -, -
        if (a + b - c - d == 24) {
            solutionList = concatStr(solutionList, "((" + a + " + " + b + ") - " + c + ") - " + d);
            solutionList = concatStr(solutionList, "(" + a + " + (" + b + " - " + c + ")) - " + d);
            solutionList = concatStr(solutionList, "(" + a + " + " + b + ") - (" + c + " + " + d + ")");
            solutionList = concatStr(solutionList, a + " + ((" + b + " - " + c + ") - " + d + ")");
```

```
// -, +, +
if (a - b + c + d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " - " + b + ") + " + c + ") + " + d);
    solutionList = concatStr(solutionList, "(" + a + " - (" + b + " - " + c + ")) + " + d);
    solutionList = concatStr(solutionList, "(" + a + " - " + b + ") + (" + c + " + " + d + ")");
    solutionList = concatStr(solutionList, a + " - ((" + b + " - " + c + ") - " + d + ")");
}
// +, -, -
if (a + b - c - d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " + " + b + ") - " + c + ") - " + d);
    solutionList = concatStr(solutionList, "(" + a + " + (" + b + " - " + c + ")) - " + d);
    solutionList = concatStr(solutionList, "(" + a + " + " + b + ") - (" + c + " + " + d + ")");
    solutionList = concatStr(solutionList, a + " + ((" + b + " - " + c + ") - " + d + ")");
}
// -, +, -
if (a - b + c - d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " - " + b + ") + " + c + ") - " + d);
    solutionList = concatStr(solutionList, "(" + a + " - (" + b + " - " + c + ")) - " + d);
    solutionList = concatStr(solutionList, "(" + a + " - " + b + ") + (" + c + " - " + d + ")");
    solutionList = concatStr(solutionList, a + " - ((" + b + " - " + c + ") + " + d + ")");

}
// -, -, +
if (a - b - c + d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " - " + b + ") - " + c + ") + " + d);
    solutionList = concatStr(solutionList, "(" + a + " - (" + b + " + " + c + ")) + " + d);
    solutionList = concatStr(solutionList, "(" + a + " - " + b + ") - (" + c + " - " + d + ")");
    solutionList = concatStr(solutionList, a + " - ((" + b + " + " + c + ") - " + d + ")");
}
// -, -, -
if (a - b - c - d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " - " + b + ") - " + c + ") - " + d);
    solutionList = concatStr(solutionList, "(" + a + " - (" + b + " + " + c + ")) - " + d);
    solutionList = concatStr(solutionList, "(" + a + " - " + b + ") - (" + c + " + " + d + ")");
    solutionList = concatStr(solutionList, a + " - ((" + b + " + " + c + ") + " + d + ")");

}
```

Semua operasi kecuali pembagian:



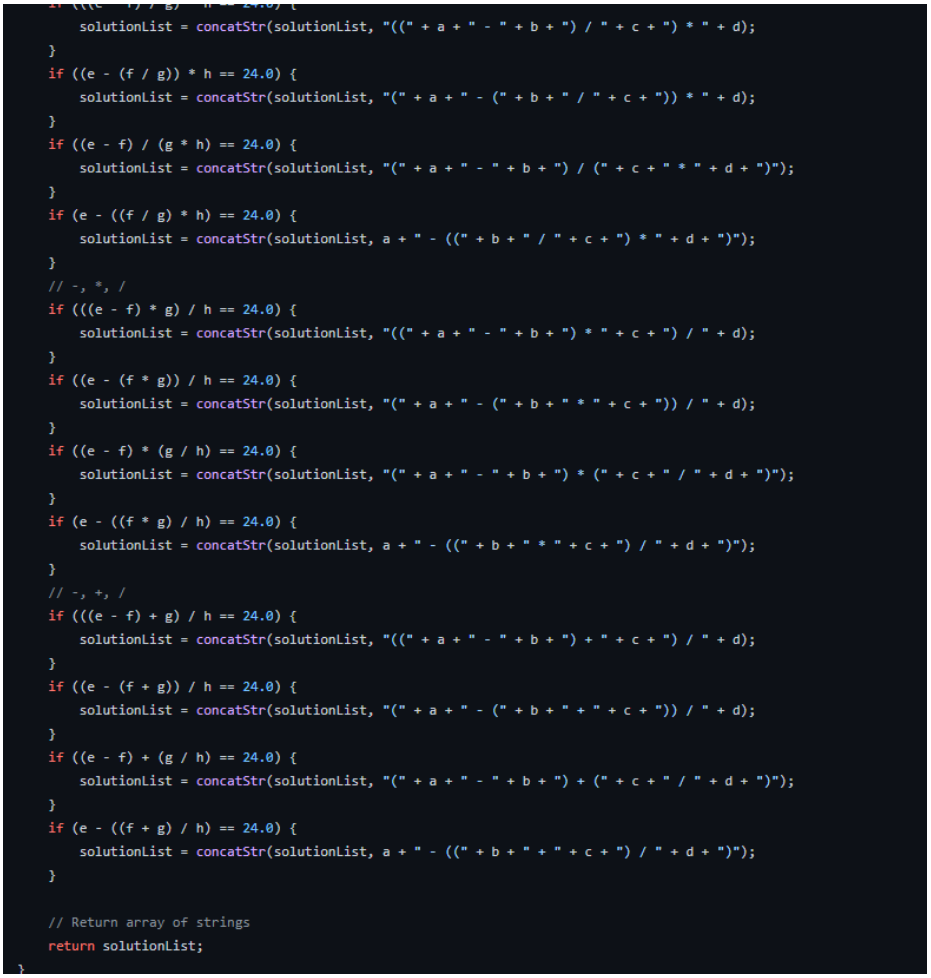
```
// Semua operasi kecuali pembagian

// *, *, *
if (a * b * c * d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " * " + b + ") * " + c + ") * " + d);
    solutionList = concatStr(solutionList, "(" + a + " * (" + b + " * " + c + ")) * " + d);
    solutionList = concatStr(solutionList, "(" + a + " * " + b + ") * (" + c + " * " + d + ")");
    solutionList = concatStr(solutionList, a + " * ((" + b + " * " + c + ") * " + d + ")");
}
// *, *, +
if ((a * b * c) + d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " * " + b + ") * " + c + ") + " + d);
    solutionList = concatStr(solutionList, "(" + a + " * (" + b + " * " + c + ")) + " + d);
}
if ((a * b) * (c + d) == 24) {
    solutionList = concatStr(solutionList, "(" + a + " * " + b + ") * (" + c + " + " + d + ")");
}
if (a * ((b * c) + d) == 24) {
    solutionList = concatStr(solutionList, a + " * ((" + b + " * " + c + ") + " + d + ")");
}
// *, +, *
if ((a * b + c) * d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " * " + b + ") + " + c + ") * " + d);
}
if ((a * (b + c)) * d == 24) {
    solutionList = concatStr(solutionList, "(" + a + " * (" + b + " + " + c + ")) * " + d);
}
if ((a * b) + (c * d) == 24) {
    solutionList = concatStr(solutionList, "(" + a + " * " + b + ") + (" + c + " * " + d + ")");
}
if (a * ((b + c) * d) == 24) {
    solutionList = concatStr(solutionList, a + " * ((" + b + " + " + c + ") * " + d + ")");
}
// +, *, *
if (((a + b) * c) * d == 24) {
    solutionList = concatStr(solutionList, "((" + a + " + " + b + ") * " + c + ") * " + d);
}
if ((a + (b * c)) * d == 24) {
    solutionList = concatStr(solutionList, "(" + a + " + (" + b + " * " + c + ")) * " + d);
}
if ((a + b) * (c * d) == 24) {
    solutionList = concatStr(solutionList, "(" + a + " + " + b + ") * (" + c + " * " + d + ")");
}
```

dst.

Operasi yang termasuk pembagian:

```
// Operasi termasuk pembagian
// a, b, c, d harus diubah menjadi double lalu diubah lagi menjadi int
double[] flVersion = convertTodouble(a, b, c, d);
double e = flVersion[0];
double f = flVersion[1];
double g = flVersion[2];
double h = flVersion[3];
// +, +, /
if (((e + f) + g) / h == 24.0) {
    solutionList = concatStr(solutionList, "((" + a + " " + b + ") + " + c + ") / " + d);
    solutionList = concatStr(solutionList, "(" + a + " + (" + b + " " + c + ")) / " + d);
}
if ((e + f) + (g / h) == 24.0) {
    solutionList = concatStr(solutionList, "(" + a + " " + b + ") + (" + c + " / " + d + ")");
}
if (e + ((f + g) / h) == 24.0) {
    solutionList = concatStr(solutionList, a + " + ((" + b + " " + c + ") / " + d + ")");
}
// +, /, +
if (((e + f) / g) + h == 24.0) {
    solutionList = concatStr(solutionList, "((" + a + " " + b + ") / " + c + ") + " + d);
}
if ((e + (f / g)) + h == 24.0) {
    solutionList = concatStr(solutionList, "(" + a + " + (" + b + " / " + c + ")) + " + d);
}
if ((e + f) / (g + h) == 24.0) {
    solutionList = concatStr(solutionList, "(" + a + " + " + b + ") / (" + c + " + " + d + ")");
}
if (e + ((f / g) + h) == 24.0) {
    solutionList = concatStr(solutionList, a + " + ((" + b + " / " + c + ") + " + d + ")");
}
// /, +, +
if (((e / f) + g) + h == 24.0) {
    solutionList = concatStr(solutionList, "((" + a + " / " + b + ") + " + c + ") + " + d);
}
if ((e / (f + g)) + h == 24.0) {
    solutionList = concatStr(solutionList, "(" + a + " / (" + b + " " + c + ")) + " + d);
}
```

dst.



```
        solutionList = concatStr(solutionList, "((" + a + " - " + b + ") / " + c + ") * " + d);
    }
    if ((e - (f / g)) * h == 24.0) {
        solutionList = concatStr(solutionList, "(" + a + " - (" + b + " / " + c + ")) * " + d);
    }
    if ((e - f) / (g * h) == 24.0) {
        solutionList = concatStr(solutionList, "(" + a + " - " + b + ") / (" + c + " * " + d + ")");
    }
    if (e - ((f / g) * h) == 24.0) {
        solutionList = concatStr(solutionList, a + " - ((" + b + " / " + c + ") * " + d + ")");
    }
    // -, *, /
    if (((e - f) * g) / h == 24.0) {
        solutionList = concatStr(solutionList, "((" + a + " - " + b + ") * " + c + ") / " + d);
    }
    if ((e - (f * g)) / h == 24.0) {
        solutionList = concatStr(solutionList, "(" + a + " - (" + b + " * " + c + ")) / " + d);
    }
    if ((e - f) * (g / h) == 24.0) {
        solutionList = concatStr(solutionList, "(" + a + " - " + b + ") * (" + c + " / " + d + ")");
    }
    if (e - ((f * g) / h) == 24.0) {
        solutionList = concatStr(solutionList, a + " - ((" + b + " * " + c + ") / " + d + ")");
    }
    // -, +, /
    if (((e - f) + g) / h == 24.0) {
        solutionList = concatStr(solutionList, "((" + a + " - " + b + ") + " + c + ") / " + d);
    }
    if ((e - (f + g)) / h == 24.0) {
        solutionList = concatStr(solutionList, "(" + a + " - (" + b + " + " + c + ")) / " + d);
    }
    if ((e - f) + (g / h) == 24.0) {
        solutionList = concatStr(solutionList, "(" + a + " - " + b + ") + (" + c + " / " + d + ")");
    }
    if (e - ((f + g) / h) == 24.0) {
        solutionList = concatStr(solutionList, a + " - ((" + b + " + " + c + ") / " + d + ")");
    }

    // Return array of strings
    return solutionList;
}
```

ii. String[] final24(int a, int b, int c, int d)

```java
public static String[] final24(int a, int b, int c, int d) {
    String[] fullSolutionList = {};
    if (a != b && a != c && a != d && b != c && b != d && c != d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, c, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, a, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, d, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, a, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, d, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, d, a, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, d, b, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, c, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, c, a, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, c, b, a));
    }
    else if (a == b && a != c && a != d && b != c && b != d && c != d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, c, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, d, a, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, c, a, b));
    }
    else if (a != b && a == c && a != d && b != c && b != d && c != d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
```

```
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, c, a, b));
    }
    else if (a != b && a == c && a != d && b != c && b != d && c != d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, c, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, c, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, a, c));
    }
    else if (a != b && a != c && a == d && b != c && b != d && c != d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, c, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, a, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, b, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, d, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, a, d));
    } else if (a != b && a != c && a != d && b == c && b != d && c != d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, c, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, d, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, a, d));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, d, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, b, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, b, c, a));
```

```
} else if (a != b && a != c && a != d && b != c && b == d && c != d) {
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, b, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, c, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, d, c));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, a, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, d, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, a, c));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, d, c, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, b, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, a, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, d, a));
} else if (a != b && a != c && a != d && b != c && b != d && c == d) {
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, b, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, d, b));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, c, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, a, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, c, d, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, b, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, d, b));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, a, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, b, d, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, d, a, b));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, d, b, a));
} else if (a == b && a == c && a != d) {
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, c, d));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, d, c));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, d, b, c));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(d, a, b, c));
} else if (a != b && a == c && a == d) {
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, a, b));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, b, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, a, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, a, a));
}
else if (a == b && a != c && a == d) {
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, a, c));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, c, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, a, a));
    fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, a, a));
} else if (a != b && b == c && b == d) {
```

```
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, a, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, a, a));
    }
    else if (a == b && a != c && a == d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, a, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, a, a));
    } else if (a != b && b == c && b == d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, b, b, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, b, a, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, b, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, b, b));
    } else if (a == b && a == c && a == d) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, a, a));
    } else if (a == b && c == d && a != c) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, c, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, a, c));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, c, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, c, a, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, c, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(c, a, a, c));
    } else if (a == c && b == d && a != b) {
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, a, b, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, a, b));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(a, b, b, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, b, a, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, b, a));
        fullSolutionList = mergeArrays(fullSolutionList, findsolution24(b, a, a, b));
    }
    return fullSolutionList;
}
```

c. main.java
    i. Proses input

```java
import java.util.Scanner;

public class main extends operation {
    public static void main(String[] args) throws Exception {
        boolean notRunning = false;
        println("================================================================================");
        println("Selamat Datang Pada Permainan Kartu 24!");
        println("================================================================================");
        while (!notRunning) {
            int[] newArray = new int[4];
            println();
            println("Apakah Anda ingin meng-input angka secara manual atau diacak secara random?");
            println("Press m for manual, press any other key for auto");
            boolean pilihan = pilihanMenu("m");
            if (pilihan) {
                newArray = manualInput();
            } else {
                newArray = RandomNumbers();
            }
            String a = intToString(newArray[0]);
            String b = intToString(newArray[1]);
            String c = intToString(newArray[2]);
            String d = intToString(newArray[3]);
            int e = newArray[0];
            int f = newArray[1];
            int g = newArray[2];
            int h = newArray[3];
            println();
            printf("Kartu yang Anda punya adalah: %s %s %s %s", a, b, c, d);
            println();
            println();
```

ii. Algoritma brute force

```java
long startTime = System.nanoTime();

String[] finalResults = final24(e, f, g, h);

long endTime = System.nanoTime();
```

iii. Output dan simpan solusi ke file .txt

```
        if (finalResults.length != 0) {
            printf("Banyak solusi adalah: %d", finalResults.length);
            println();
            printlnstr("Berikut daftar solusinya:");
            printstrarray(finalResults);
        } else {
            printlnstr("Tidak ada solusi.");
        }
        println();

        printlnstr("Apakah Anda ingin menyimpan file di .txt?");
        printlnstr("Press y for yes, press any other key for no");
        boolean pilihanfile = pilihanMenu("y");
        boolean finishedWriting = false;
        if (pilihanfile) {
            while (!finishedWriting) {
                printlnstr("Masukkan nama file yang ingin dituliskan hasilnya:");
                print("> ");
                Scanner sc = new Scanner(System.in);
                String directory = sc.next();
                finishedWriting = writeSolutionToFile("test/" + directory, finalResults, a, b, c, d);
            }
        }

        println();

        printlnstr("Apakah Anda ingin tetap bermain atau keluar?");
        printlnstr("Press y if you want to stay, press any other key if you want to quit");
        boolean pilihanlagi = pilihanMenu("y");
        if (pilihanlagi == false) {
            printlnstr("================================================================================");
            printlnstr("Terima kasih sudah bermain, sampai jumpa lagi!");
            printlnstr("================================================================================");
            println();
            long duration = (endTime - startTime);
            println();
            printlnstr("Waktu eksekusi program terbaru: " + duration + " nanosecond" );
            println();
            notRunning = true;
        }

    }
```

3. Input dan Output

a. Menu Input

i. Input manual

```
================================================================================
Selamat Datang Pada Permainan Kartu 24!
================================================================================

Apakah Anda ingin meng-input angka secara manual atau diacak secara random?
Press m for manual, press any other key for auto
> m
Masukkan pilihan kartu: > 3 4 5 6

Kartu yang Anda punya adalah: 3 4 5 6
```

ii. Random number generator

```
Apakah Anda ingin meng-input angka secara manual atau diacak secara random?
Press m for manual, press any other key for auto
> j

Kartu yang Anda punya adalah: 3 9 10 A
```

b. Save File
   i. Iya

```
Apakah Anda ingin menyimpan file di .txt?
Press y for yes, press any other key for no
> y
Masukkan nama file yang ingin dituliskan hasilnya:
> datatest9.txt
Hasil sudah dituliskan kepada file.
```

   ii. Tidak

```
Apakah Anda ingin menyimpan file di .txt?
Press y for yes, press any other key for no
> d

Apakah Anda ingin tetap bermain atau keluar?
Press y if you want to stay, press any other key if you want to quit
```

c. Keluar atau tidak
   i. Tetap bermain

```
Apakah Anda ingin tetap bermain atau keluar?
Press y if you want to stay, press any other key if you want to quit
> y

Apakah Anda ingin meng-input angka secara manual atau diacak secara random?
Press m for manual, press any other key for auto
```

   ii. Keluar

```
Apakah Anda ingin tetap bermain atau keluar?
Press y if you want to stay, press any other key if you want to quit
> j
================================================================================
Terima kasih sudah bermain, sampai jumpa lagi!
================================================================================


Waktu eksekusi program terbaru: 1379700 nanosecond
```

d. datatest1.txt

```
Kartu yang Anda punya adalah: 3 4 5 6

Banyak solusi adalah: 12
Berikut daftar solusinya:
(3 - (4 - 5)) * 6
((3 - 4) + 5) * 6
((3 + 5) - 4) * 6
(3 + (5 - 4)) * 6
((5 + 3) - 4) * 6
(5 + (3 - 4)) * 6
(5 - (4 - 3)) * 6
((5 - 4) + 3) * 6
6 * ((3 - 4) + 5)
6 * ((3 + 5) - 4)
6 * ((5 + 3) - 4)
6 * ((5 - 4) + 3)
```

test > datatest1.txt

```
1  Kartu: 3 4 5 6
2  Banyak solusi: 12
3  (3 - (4 - 5)) * 6
4  ((3 - 4) + 5) * 6
5  ((3 + 5) - 4) * 6
6  (3 + (5 - 4)) * 6
7  ((5 + 3) - 4) * 6
8  (5 + (3 - 4)) * 6
9  (5 - (4 - 3)) * 6
10 ((5 - 4) + 3) * 6
11 6 * ((3 - 4) + 5)
12 6 * ((3 + 5) - 4)
13 6 * ((5 + 3) - 4)
14 6 * ((5 - 4) + 3)
15
```

e. datatest2.txt

```
Kartu yang Anda punya adalah: J Q Q Q

Banyak solusi adalah: 18
Berikut daftar solusinya:
(12 + 12) * (12 - 11)
(12 + (12 / 12)) + 11
12 + ((12 / 12) + 11)
((12 / 12) + 12) + 11
(12 / 12) + (12 + 11)
(12 + 12) / (12 - 11)
12 + ((12 - 11) * 12)
(12 * (12 - 11)) + 12
((12 / 12) + 11) + 12
(12 / 12) + (11 + 12)
(12 / (12 - 11)) + 12
12 - ((11 - 12) * 12)
((12 - 11) * 12) + 12
(12 - 11) * (12 + 12)
(12 + 11) + (12 / 12)
(11 + 12) + (12 / 12)
(11 + (12 / 12)) + 12
11 + ((12 / 12) + 12)
```

test > datatest2.txt

```
1  Kartu: J Q Q Q
2  Banyak solusi: 18
3  (12 + 12) * (12 - 11)
4  (12 + (12 / 12)) + 11
5  12 + ((12 / 12) + 11)
6  ((12 / 12) + 12) + 11
7  (12 / 12) + (12 + 11)
8  (12 + 12) / (12 - 11)
9  12 + ((12 - 11) * 12)
10 (12 * (12 - 11)) + 12
11 ((12 / 12) + 11) + 12
12 (12 / 12) + (11 + 12)
13 (12 / (12 - 11)) + 12
14 12 - ((11 - 12) * 12)
15 ((12 - 11) * 12) + 12
16 (12 - 11) * (12 + 12)
17 (12 + 11) + (12 / 12)
18 (11 + 12) + (12 / 12)
19 (11 + (12 / 12)) + 12
20 11 + ((12 / 12) + 12)
21
```

f. datatest3.txt

```
Kartu yang Anda punya adalah: 10 10 10 10

Tidak ada solusi.
```

test > datatest3.txt

```
1   Kartu: 10 10 10 10
2   Tidak ada solusi.
3
```

g. datatest4.txt

```
Kartu yang Anda punya adalah: 4 5 4 5

Banyak solusi adalah: 10
Berikut daftar solusinya:
(4 + (4 / 5)) * 5
4 * ((5 - 4) + 5)
((4 / 5) + 4) * 5
4 * ((5 + 5) - 4)
((5 + 5) - 4) * 4
(5 + (5 - 4)) * 4
(5 * 5) - (4 / 4)
(5 - (4 - 5)) * 4
((5 - 4) + 5) * 4
5 * ((4 / 5) + 4)
```

test > datatest4.txt

```
1   Kartu: 4 5 4 5
2   Banyak solusi: 10
3   (4 + (4 / 5)) * 5
4   4 * ((5 - 4) + 5)
5   ((4 / 5) + 4) * 5
6   4 * ((5 + 5) - 4)
7   ((5 + 5) - 4) * 4
8   (5 + (5 - 4)) * 4
9   (5 * 5) - (4 / 4)
10  (5 - (4 - 5)) * 4
11  ((5 - 4) + 5) * 4
12  5 * ((4 / 5) + 4)
```

h. datatest5.txt

```
Kartu yang Anda punya adalah: 4 Q 8 5

Banyak solusi adalah: 40
Berikut daftar solusinya:
4 + ((12 - 8) * 5)
(4 - 12) * (5 - 8)
4 - ((8 - 12) * 5)
(4 * (8 - 5)) + 12
((4 * 5) + 12) - 8
(4 * 5) + (12 - 8)
(4 * 5) - (8 - 12)
((4 * 5) - 8) + 12
(12 - 4) * (8 - 5)
((12 - 4) - 5) * 8
(12 + (4 * 5)) - 8
12 + ((4 * 5) - 8)
(12 - (4 + 5)) * 8
(12 - 8) + (4 * 5)
12 + ((8 - 5) * 4)
((12 - 8) * 5) + 4
(12 - 8) + (5 * 4)
((12 - 5) - 4) * 8
(12 + (5 * 4)) - 8
12 + ((5 * 4) - 8)
(12 - (5 + 4)) * 8
12 - ((5 - 8) * 4)
8 * ((12 - 4) - 5)
8 * ((12 - 5) - 4)
((8 * 5) - 4) - 12
(8 * 5) - (4 + 12)
((8 - 5) * 4) + 12
(8 - 5) * (12 - 4)
((8 * 5) - 12) - 4
(8 * 5) - (12 + 4)
((5 * 4) + 12) - 8
(5 * 4) + (12 - 8)
(5 * 4) - (8 - 12)
((5 * 4) - 8) + 12
(5 * (12 - 8)) + 4
(5 - 8) * (4 - 12)
((5 * 8) - 4) - 12
(5 * 8) - (4 + 12)
((5 * 8) - 12) - 4
(5 * 8) - (12 + 4)
```

test > datatest5.txt

```
1  Kartu: 4 Q 8 5
2  Banyak solusi: 40
3  4 + ((12 - 8) * 5)
4  (4 - 12) * (5 - 8)
5  4 - ((8 - 12) * 5)
6  (4 * (8 - 5)) + 12
7  ((4 * 5) + 12) - 8
8  (4 * 5) + (12 - 8)
9  (4 * 5) - (8 - 12)
10 ((4 * 5) - 8) + 12
11 (12 - 4) * (8 - 5)
12 ((12 - 4) - 5) * 8
13 (12 + (4 * 5)) - 8
14 12 + ((4 * 5) - 8)
15 (12 - (4 + 5)) * 8
16 (12 - 8) + (4 * 5)
17 12 + ((8 - 5) * 4)
18 ((12 - 8) * 5) + 4
19 (12 - 8) + (5 * 4)
20 ((12 - 5) - 4) * 8
21 (12 + (5 * 4)) - 8
22 12 + ((5 * 4) - 8)
23 (12 - (5 + 4)) * 8
24 12 - ((5 - 8) * 4)
25 8 * ((12 - 4) - 5)
26 8 * ((12 - 5) - 4)
27 ((8 * 5) - 4) - 12
28 (8 * 5) - (4 + 12)
29 ((8 - 5) * 4) + 12
30 (8 - 5) * (12 - 4)
31 ((8 * 5) - 12) - 4
32 (8 * 5) - (12 + 4)
33 ((5 * 4) + 12) - 8
34 (5 * 4) + (12 - 8)
35 (5 * 4) - (8 - 12)
36 ((5 * 4) - 8) + 12
37 (5 * (12 - 8)) + 4
38 (5 - 8) * (4 - 12)
39 ((5 * 8) - 4) - 12
40 (5 * 8) - (4 + 12)
41 ((5 * 8) - 12) - 4
42 (5 * 8) - (12 + 4)
43
```

i. datatest6.txt

```
Kartu yang Anda punya adalah: 10 8 9 10

Tidak ada solusi.
```

test > datatest6.txt

```
1  Kartu: 10 8 9 10
2  Tidak ada solusi.
3
```

j. datatest7.txt

```
Kartu yang Anda punya adalah: 3 2 J 10

Tidak ada solusi.
```

test > datatest7.txt

```
1   Kartu: 3 2 J 10
2   Tidak ada solusi.
3
```

k. datatest8.txt

```
Kartu yang Anda punya adalah: 3 9 10 A

Banyak solusi adalah: 4
Berikut daftar solusinya:
(3 * (10 + 1)) - 9
(3 * (1 + 10)) - 9
((10 + 1) * 3) - 9
((1 + 10) * 3) - 9
```

test > datatest8.txt

```
1   Kartu: 3 9 10 A
2   Banyak solusi: 4
3   (3 * (10 + 1)) - 9
4   (3 * (1 + 10)) - 9
5   ((10 + 1) * 3) - 9
6   ((1 + 10) * 3) - 9
7
```

l. datatest9.txt

```
Kartu yang Anda punya adalah: Q 9 9 4

Banyak solusi adalah: 6
Berikut daftar solusinya:
(9 - (12 - 9)) * 4
((9 - 12) + 9) * 4
((9 + 9) - 12) * 4
(9 + (9 - 12)) * 4
4 * ((9 - 12) + 9)
4 * ((9 + 9) - 12)
```

test > datatest9.txt

```
1   Kartu: Q 9 9 4
2   Banyak solusi: 6
3   (9 - (12 - 9)) * 4
4   ((9 - 12) + 9) * 4
5   ((9 + 9) - 12) * 4
6   (9 + (9 - 12)) * 4
7   4 * ((9 - 12) + 9)
8   4 * ((9 + 9) - 12)
9
```

4. Referensi dan Lampiran

- https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/MakalahStima-2016-038.pdf
- http://24solver.us-west-2.elasticbeanstalk.com/
- https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf

Link repository: https://github.com/BreezyDR/Tucil1_13521048

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 teks	v	