

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2022/2023

**Penyelesaian Permainan Kartu 24
dengan Algoritma Brute Force**



Hobert Anthony Jonatan
13521079

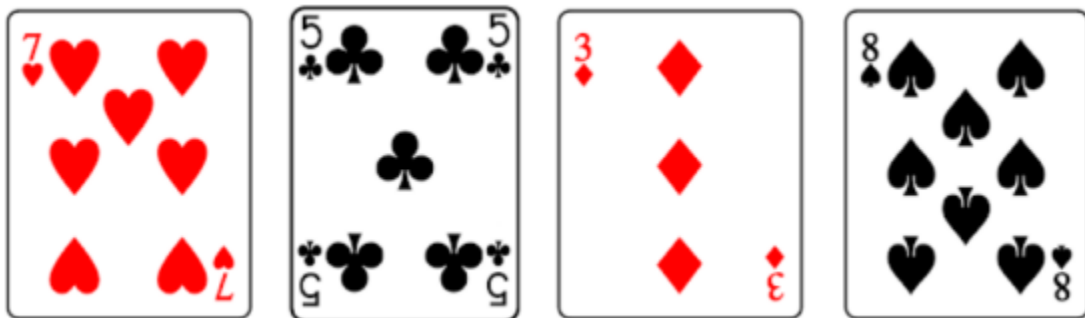
PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

DAFTAR ISI

| | |
|---|-----------|
| BAB 1 : DESKRIPSI MASALAH | 3 |
| BAB 2 : PENERAPAN ALGORITMA BRUTE FORCE | 4 |
| BAB 3 : IMPLEMENTASI PROGRAM DENGAN JAVA | 5 |
| BAB 4 : EKSPERIMEN | 13 |
| LAMPIRAN | 22 |

BAB I : DESKRIPSI MASALAH

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 menarik cukup banyak peminat dikarenakan dapat meningkatkan kemampuan berhitung serta mengasah otak agar dapat berpikir dengan cepat dan akurat. Permainan Kartu 24 biasa 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. (Paragraf di atas dikutip dari sini: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah2016/MakalahStima-2016-038.pdf>).



Gambar 1. Permainan Kartu 24

BAB 2 : PENERAPAN ALGORITMA BRUTE FORCE

Brute Force adalah sebuah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu masalah, biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *Brute Force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*) atau paling mudah terpikir (sangat intuitif).

Berikut adalah langkah-langkah penyelesaian permainan kartu 24 dengan algoritma *Brute Force* yang saya desain.

Langkah 1

Meninjau seluruh kombinasi ekspresi aritmatika yang dapat dibentuk dari 4 operan dan 3 operator, terdapat 5 ekspresi utama yang mewakili seluruh ekspresi yang dapat dibentuk dari 4 operan dan 3 operator.

Langkah 2

Melakukan looping untuk membentuk seluruh susunan permutasi dari 4 operan dan diikuti dengan penempatan 3 operator lainnya, dalam kasus ini operator bisa dipakai berulang tetapi operan hanya bisa sekali saja. Susun seluruh kombinasi berbeda dan evaluasi nilai yang dihasilkan dari ekspresi tersebut.

Langkah 3

Tinjau seluruh ekspresi yang dihasilkan dan bandingkan hasil ekspresi aritmatikanya dengan angka 24, apabila ekspresi menghasilkan 24 maka simpan ekspresi tersebut dalam array dan tambah nilai counter dari jumlah solusi.

Langkah 4

Tampilkan seluruh solusi

BAB 3 : IMPLEMENTASI PROGRAM DENGAN JAVA

Algoritma yang telah dijabarkan pada Bab 2 diimplementasikan dengan program dalam bahasa Java. Seluruh program dimuat pada satu file saja, yaitu Main.java karena permasalahan yang diselesaikan cukup singkat dan program yang dibuat juga tidak terlalu besar sehingga masih wajar untuk dimuat hanya dalam satu file. File Main.java berisi seluruh fungsi yang diperlukan untuk menyelesaikan permainan kartu 24 dan juga program utama untuk menyelesaikan permainan ini.

Berikut adalah source code dari program yang telah dibuat.

```
import java.util.*;
import java.io.*;

public class Main{
    private static InputStreamReader streamReader = new InputStreamReader(System.in);
    private static BufferedReader bufferedReader = new BufferedReader(Main.streamReader);

    public static int converter(String x){
        int num;
        char y = x.charAt(index: 0);
        int index = y;
        if(index == 65){           // A
            num = 1;
        }else if(index == 74){    // J
            num = 11;
        }else if(index== 81){     // Q
            num = 12;
        }else if(index == 75){   // K
            num = 13;
        }else{
            num = Integer.parseInt(x);
        }
        return num;
    }
}
```

Gambar 2. Source Code (1)

```

public static boolean validator(String y){
    boolean hasil = false;
    char x = y.charAt(index: 0);
    int index = x;
    if(index == 65){           // A
        hasil = true;
    }else if(index == 74){    // J
        hasil = true;
    }else if(index == 81){    // Q
        hasil = true;
    }else if(index == 75){    // K
        hasil = true;
    }else{
        try{
            if(Integer.parseInt(y) > 0 && Integer.parseInt(y) < 14){
                hasil = true;
            }
        }catch(NumberFormatException ex){
            System.out.println(x: "Salah masukan");
        }
    }
    return hasil;
}

public static int[] generateRandom(){
    Random rand = new Random();
    int num1 = rand.nextInt((13-1)+1)+1;
    int num2 = rand.nextInt((13-1)+1)+1;
    int num3 = rand.nextInt((13-1)+1)+1;
    int num4 = rand.nextInt((13-1)+1)+1;
    int[] result = {num1, num2, num3, num4};
    return result;
}

```

Gambar 3. Source Code (2)

```

public static void showGeneratedCard(int[] num){
    System.out.print(s: "Kartu random: ");
    for(int i:num){
        if(i == 13){
            System.out.print(s: "K");
        }else if(i == 12){
            System.out.print(s: "Q");
        }else if(i == 11){
            System.out.print(s: "J");
        }else if(i == 1){
            System.out.print(s: "A");
        }else{
            System.out.print(i);
        }
        System.out.print(s: " ");
    }
    System.out.println();
}

public static int[] getFromUser() {
    boolean inputValid = false;
    System.out.print(s: "Masukkan 4 kartu: ");
    String input = "test";
    try{
        input = bufferedReader.readLine();
    }catch(IOException e){
        e.printStackTrace();
        System.out.println(x: "Ada kesalahan dalam masukan");
    }
    String[] hasil = input.split(regex: " ");
    int len = hasil.length;
    if(len == 4){
        inputValid = validator(hasil[0]) && validator(hasil[1]) && validator(hasil[2]) && validator(hasil[3]);
    }
}

```

Gambar 4. Source Code(3)

```

    int num1 = converter(hasil[0]);
    int num2 = converter(hasil[1]);
    int num3 = converter(hasil[2]);
    int num4 = converter(hasil[3]);

    int[] result = {num1, num2, num3, num4};
    return result;
}

public static double operate(double num1, double num2, String op){
    double result = 0;
    if(op == "+"){
        result = num1 + num2;
    }else if(op == "-"){
        result = num1 - num2;
    }else if(op == "*"){
        result = num1 * num2;
    }else if(op == "/"){
        result = num1 / num2;
    }
    return result;
}

```

Gambar 5. Source Code(4)

```

public static void solve(int[] num) throws IOException{

    long begin = System.currentTimeMillis();
    List<String> solution=new ArrayList<String>();
    int goal = 24;
    int count_solution = 0;
    String[] op = {"+", "-", "*", "/"};
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < 4; j++){
            if(j != i){
                for(int k = 0; k < 4; k++){
                    if(k != i && k != j){
                        for(int l = 0; l < 4; l++){
                            if(l != i && l != j && l != k){
                                for(int m = 0; m < 4; m++){
                                    for(int n = 0; n < 4; n++){
                                        for(int o = 0; o < 4; o++){

                                            double a = num[i];
                                            double b = num[j];
                                            double c = num[k];
                                            double d = num[l];

                                            String op1 = op[m];
                                            String op2 = op[n];
                                            String op3 = op[o];

```

Gambar 6. Source Code(5)


```

String sol;
// variasi 1 : ((A op B) op C) op D
if(operate(operate(operate(a,b,op1),c,op2),d,op3) == goal){
    count_solution++;
    sol = String.format(format: "((%d %s %d) %s %d) %s %d",num[i],op1,num[j],op2,num[k],op3,num[l]);
    solution.add(sol);
}

// variasi 2 : (A op (B op C)) op D
if(operate(operate(a,operate(b,c,op2),op1),d,op3) == goal){
    count_solution++;
    sol = String.format(format: "(%d %s (%d %s %d)) %s %d",num[i],op1,num[j],op2,num[k],op3,num[l]);
    solution.add(sol);
}

// variasi 3 : A op ((B op C) op D)
if(operate(a,operate(operate(b,c,op2),d,op3),op1) == goal){
    count_solution++;
    sol = String.format(format: "%d %s ((%d %s %d) %s %d)",num[i],op1,num[j],op2,num[k],op3,num[l]);
    solution.add(sol);
}

// variasi 4 : A op (B op (C op D))
if(operate(a,operate(b,operate(c,d,op3),op2),op1) == goal){
    count_solution++;
    sol = String.format(format: "%d %s (%d %s (%d %s %d))",num[i],op1,num[j],op2,num[k],op3,num[l]);
    solution.add(sol);
}

// variasi 5 : (A op B) op (C op D)

```

Gambar 7. Source Code(6)

```

// variasi 5 : (A op B) op (C op D)

if(operate(operate(a,b,op1),operate(c,d,op2),op3) == goal){
    count_solution++;
    sol = String.format(format: "(%d %s %d) %s (%d %s %d)",num[i],op1,num[j],op2,num[k],op3,num[l]);
    solution.add(sol);
}

```

Gambar 8. Source Code(7)

```

    }
    if(count_solution == 0){
        System.out.println();
        System.out.println(x: "Tidak ada solusi yang ditemukan untuk kombinasi angka tersebut");
        System.out.println();
    }else{
        System.out.println();
        System.out.printf(format: "Terdapat %d solusi",count_solution);
        System.out.println();
        for(String sols:solution){
            System.out.println(sols);
        }

        long end = System.currentTimeMillis();
        long time = end - begin;
        System.out.println();
        System.out.println("Waktu Eksekusi: " + time + "ms");

        String filename = "testing"; // temporary name
        int jawab = 0;
        System.out.print(s: "Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): ");
        try{
            jawab = Integer.parseInt(bufferedReader.readLine());
        }catch(IOException e){
            e.printStackTrace();
        }

        if(jawab == 1){
            System.out.print(s: "Masukkan nama file (tanpa .txt): ");
            filename = bufferedReader.readLine();
            FileWriter writer = new FileWriter("Tucil1_13521079/test/" + filename + ".txt");
            int len = solution.size();
            for(int i = 0; i< len; i++){
                writer.write(solution.get(i) + "\n");
            }
            writer.close();
        }
    }
}

```

Gambar 9. Source Code(8)

```

public static void printMenu(){
    System.out.println();
    System.out.println(x: "#####");
    System.out.println(x: "  1. Input manual ");
    System.out.println(x: "  2. Generate 4 kartu secara acak ");
    System.out.println(x: "  0. Keluar");
    System.out.println();
}

public static int askInput() throws IOException{
    int action = 3;

    while(action < 0 || action > 2){
        System.out.print(s: "Pilih (1/2/0): ");
        try{
            action = Integer.parseInt(bufferedReader.readLine());
        }catch(IOException e){
            System.out.println(x: "Masukkan salah");
        }
    }
    return action;
}

Run | Debug
public static void main(String[] args) throws IOException {
    System.out.println();
    System.out.println(x: "#####");
    System.out.println(x: "                          Selamat datang di program 24 Solver!");
    System.out.println(x: "Anda dapat melihat solusi untuk permainan kartu 24 dengan program ini.");
    printMenu();
    int aksi = askInput();
    int[] kartu;
    while(aksi != 0){
        if(aksi == 1){
            kartu = getFromUser();

```

Gambar 10. Source Code(9)

```

    }else{
        kartu = generateRandom();
        showGeneratedCard(kartu);
    }
    solve(kartu);
    System.out.println();
    System.out.println();

    printMenu();
    aksi = askInput();
}

System.out.println();
System.out.println();
System.out.println(x: "#####");
System.out.println(x: "Terima kasih telah menggunakan program ini");
System.out.println(x: "#####");
}
}

```

Gambar 11. Source Code (10)

BAB 4 : EKSPERIMEN

Berikut adalah tampilan program dan beberapa contoh kasus saat program dijalankan.

```
#####  
#####  
                Selamat datang di program 24 Solver!  
Anda dapat melihat solusi untuk permainan kartu 24 dengan program ini.  
#####  
  
#####  
    1. Input manual  
    2. Generate 4 kartu secara acak  
    0. Keluar  
  
Pilih (1/2/0): 
```

Gambar 12. Tampilan Program

```
#####
1. Input manual
2. Generate 4 kartu secara acak
0. Keluar

Pilih (1/2/0): 1
Masukkan 4 kartu: 4 5 6 7

Terdapat 20 solusi
4 * ((5 - 6) + 7)
4 * (5 - (6 - 7))
4 * ((5 + 7) - 6)
4 * (5 + (7 - 6))
4 * ((7 + 5) - 6)
4 * (7 + (5 - 6))
4 * ((7 - 6) + 5)
4 * (7 - (6 - 5))
((5 - 6) + 7) * 4
(5 - (6 - 7)) * 4
((5 + 7) - 6) * 4
(5 + (7 - 6)) * 4
(5 + 7) - (6 * 4)
(6 - 4) + (5 * 7)
(6 - 4) + (7 * 5)
((7 + 5) - 6) * 4
(7 + (5 - 6)) * 4
(7 + 5) - (6 * 4)
((7 - 6) + 5) * 4
(7 - (6 - 5)) * 4

Waktu Eksekusi: 15ms
Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1
Masukkan nama file (tanpa .txt): test1
Berhasil menyimpan solusi ke dalam file!
```

Gambar 13. Test Case (1)

```
Pilih (1/2/0): 2
Kartu random: 6 9 Q 8
```

```
Terdapat 32 solusi
```

```
((6 + 9) - 12) * 8
(6 + (9 - 12)) * 8
((6 - 12) + 9) * 8
(6 - (12 - 9)) * 8
((9 + 6) - 12) * 8
(9 + (6 - 12)) * 8
((9 / 6) * 8) + 12
(9 / (6 / 8)) + 12
((9 - 12) + 6) * 8
(9 - (12 - 6)) * 8
((9 * 8) / 6) + 12
(9 * (8 / 6)) + 12
12 + ((9 / 6) * 8)
12 + (9 / (6 / 8))
12 + ((9 * 8) / 6)
12 + (9 * (8 / 6))
12 + ((8 / 6) * 9)
12 + (8 / (6 / 9))
12 + ((8 * 9) / 6)
12 + (8 * (9 / 6))
8 * ((6 + 9) - 12)
8 * (6 + (9 - 12))
((8 / 6) * 9) + 12
(8 / (6 / 9)) + 12
8 * ((6 - 12) + 9)
8 * (6 - (12 - 9))
8 * ((9 + 6) - 12)
8 * (9 + (6 - 12))
((8 * 9) / 6) + 12
(8 * (9 / 6)) + 12
8 * ((9 - 12) + 6)
8 * (9 - (12 - 6))
```

```
Waktu Eksekusi: 9ms
```

```
Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1
```

Gambar 14. Test Case (2)

```
Waktu Eksekusi: 9ms
Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1
Masukkan nama file (tanpa .txt): test2
Berhasil menyimpan solusi ke dalam file!
```

Gambar 15. Lanjutan SS Test Case (2)

```
Pilih (1/2/0): 1
Masukkan 4 kartu: 7 8 5 9

Terdapat 24 solusi
((7 + 5) - 9) * 8
(7 + (5 - 9)) * 8
((7 - 9) + 5) * 8
(7 - (9 - 5)) * 8
8 * ((7 + 5) - 9)
8 * (7 + (5 - 9))
8 * ((7 - 9) + 5)
8 * (7 - (9 - 5))
8 * ((5 + 7) - 9)
8 * (5 + (7 - 9))
(8 * 5) + (7 - 9)
((8 * 5) - 7) - 9
(8 * 5) + (9 - 7)
8 * ((5 - 9) + 7)
((8 * 5) - 9) - 7
8 * (5 - (9 - 7))
((5 + 7) - 9) * 8
(5 + (7 - 9)) * 8
(5 * 8) + (7 - 9)
((5 * 8) - 7) - 9
(5 * 8) + (9 - 7)
((5 * 8) - 9) - 7
((5 - 9) + 7) * 8
(5 - (9 - 7)) * 8

Waktu Eksekusi: 7ms
Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1
Masukkan nama file (tanpa .txt): test3
Berhasil menyimpan solusi ke dalam file!
```

Gambar 16. Test Case (3)

Pilih (1/2/0): 2
Kartu random: Q 5 J 8

Terdapat 36 solusi

$$\begin{aligned} &(12 * (5 + 11)) / 8 \\ &12 * ((5 + 11) / 8) \\ &12 * ((5 - 11) + 8) \\ &12 * (5 - (11 - 8)) \\ &12 * ((5 + 8) - 11) \\ &12 * (5 + (8 - 11)) \\ &(12 * (11 + 5)) / 8 \\ &12 * ((11 + 5) / 8) \\ &12 * ((8 + 5) - 11) \\ &12 * (8 + (5 - 11)) \\ &(12 / 8) + (5 * 11) \\ &12 / (8 / (5 + 11)) \\ &(12 - 8) - (11 * 5) \\ &12 * ((8 - 11) + 5) \\ &12 * (8 - (11 - 5)) \\ &(12 / 8) + (11 * 5) \\ &12 / (8 / (11 + 5)) \\ &((5 + 11) * 12) / 8 \\ &(5 + 11) / (12 * 8) \\ &((5 + 11) / 8) * 12 \\ &(5 + 11) / (8 / 12) \\ &((5 - 11) + 8) * 12 \\ &(5 - (11 - 8)) * 12 \\ &(5 - 11) - (8 * 12) \\ &((5 + 8) - 11) * 12 \\ &(5 + (8 - 11)) * 12 \\ &((11 + 5) * 12) / 8 \\ &(11 + 5) / (12 * 8) \\ &(11 - 5) - (12 * 8) \\ &((11 + 5) / 8) * 12 \\ &(11 + 5) / (8 / 12) \\ &(8 - 12) - (5 * 11) \\ &((8 + 5) - 11) * 12 \\ &(8 + (5 - 11)) * 12 \\ &((8 - 11) + 5) * 12 \\ &(8 - (11 - 5)) * 12 \end{aligned}$$

Gambar 17. Test Case (4)

```
Waktu Eksekusi: 11ms
Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1
Masukkan nama file (tanpa .txt): test4
Berhasil menyimpan solusi ke dalam file!
```

Gambar 18. Lanjutan Test Case 4

```
#####
1. Input manual
2. Generate 4 kartu secara acak
0. Keluar

Pilih (1/2/0): 1
Masukkan 4 kartu: 1 2 3 4
Masukan Anda salah!
```

Gambar 19. Test Case input salah

Silakan Masukkan kembali 4 kartu: A J Q K

Terdapat 32 solusi

$(1 * 12) - (13 * 11)$
 $1 * (12 * (13 - 11))$
 $((1 * 13) - 11) * 12$
 $(1 * (13 - 11)) * 12$
 $1 * ((13 - 11) * 12)$
 $(12 * 1) - (13 * 11)$
 $12 * ((1 * 13) - 11)$
 $12 * (1 * (13 - 11))$
 $(12 / 1) - (13 * 11)$
 $12 / (1 / (13 - 11))$
 $12 * (13 - (1 * 11))$
 $12 * ((13 * 1) - 11)$
 $12 * ((13 / 1) - 11)$
 $(12 * (13 - 11)) * 1$
 $12 * ((13 - 11) * 1)$
 $12 * (13 - (11 * 1))$
 $(12 * (13 - 11)) / 1$
 $12 * ((13 - 11) / 1)$
 $12 * (13 - (11 / 1))$
 $(13 - (1 * 11)) * 12$
 $((13 * 1) - 11) * 12$
 $((13 / 1) - 11) * 12$
 $((13 - 11) * 1) * 12$
 $(13 - (11 * 1)) * 12$
 $(13 - 11) * (1 * 12)$
 $((13 - 11) / 1) * 12$
 $(13 - (11 / 1)) * 12$
 $(13 - 11) / (1 / 12)$
 $((13 - 11) * 12) * 1$
 $(13 - 11) * (12 * 1)$
 $((13 - 11) * 12) / 1$
 $(13 - 11) / (12 * 1)$

Waktu Eksekusi: 9ms

Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1

Masukkan nama file (tanpa .txt): test5

Gambar 20. Test Case (5)

Pilih (1/2/0): 1
Masukkan 4 kartu: A 8 9 Q

Terdapat 48 solusi
 $((1 - 8) + 9) * 12$
 $(1 - (8 - 9)) * 12$
 $(1 * 8) - (12 * 9)$
 $1 * (8 * (12 - 9))$
 $((1 + 9) - 8) * 12$
 $(1 + (9 - 8)) * 12$
 $((1 * 12) - 9) * 8$
 $(1 * (12 - 9)) * 8$
 $1 * ((12 - 9) * 8)$
 $(8 * 1) - (12 * 9)$
 $8 * ((1 * 12) - 9)$
 $8 * (1 * (12 - 9))$
 $(8 / 1) - (12 * 9)$
 $8 / (1 / (12 - 9))$
 $8 * (12 - (1 * 9))$
 $8 * ((12 * 1) - 9)$
 $8 * ((12 / 1) - 9)$
 $(8 * (12 - 9)) * 1$
 $8 * ((12 - 9) * 1)$
 $8 * (12 - (9 * 1))$
 $(8 * (12 - 9)) / 1$
 $8 * ((12 - 9) / 1)$
 $8 * (12 - (9 / 1))$
 $((9 + 1) - 8) * 12$
 $(9 + (1 - 8)) * 12$
 $((9 - 8) + 1) * 12$
 $(9 - (8 - 1)) * 12$
 $12 * ((1 - 8) + 9)$
 $12 * (1 - (8 - 9))$
 $(12 - (1 * 9)) * 8$
 $12 * ((1 + 9) - 8)$
 $12 * (1 + (9 - 8))$
 $((12 * 1) - 9) * 8$
 $((12 / 1) - 9) * 8$

Gambar 21. Test Case (6)

```
12 * ((9 + 1) - 8)
12 * (9 + (1 - 8))
((12 - 9) * 8) * 1
(12 - 9) * (8 * 1)
((12 - 9) * 8) / 1
(12 - 9) / (8 * 1)
12 * ((9 - 8) + 1)
12 * (9 - (8 - 1))

Waktu Eksekusi: 9ms
Apakah anda ingin menyimpan solusi ke dalam file? (1:yes/0:no): 1
Masukkan nama file (tanpa .txt): test6
Berhasil menyimpan solusi ke dalam file!
```

Gambar 22. Lanjutan Test Case 6

```
Pilih (1/2/0): 0

#####
Terima kasih telah menggunakan program ini
#####
```

Gambar 23. Tampilan Saat Keluar

LAMPIRAN

Link Repository GitHub :

https://github.com/HobertJ/Tucil1_13521079

| Poin | Ya | Tidak |
|--|----|-------|
| 1. Program berhasil dikompilasi tanpa kesalahan. | | |
| 2. Program berhasil <i>running</i> . | | |
| 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. | | |