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

Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force

Henry Anand Septian Radityo – 13521004 – K03

I. Algoritma

Algoritma Brute Force bekerja dengan mencoba semua kemungkinan yang ada untuk mendapatkan hasil yang sesuai. Algoritma ini dapat digunakan untuk hampir semua permasalahan namun memiliki kekurangan yaitu keefektifan dari program brute force seringkali terbilang cukup rendah. Dalam implementasi algoritma Brute Force untuk menyelesaikan permainan kartu 24, maka dilakukan semua konfigurasi kartu dan juga kemungkinan dari operasi yang mungkin.

Perhitungan dari semua kemungkinan untuk permainan kartu 24 terdiri dari dua kemungkinan permutasi yaitu permutasi pada kartu dan permutasi pada operator. Untuk kemungkinan dari 4 kartu maka dapat dilakukan perhitungan dengan rumus permutasi yaitu 24 kemungkinan yang didapat dari banyaknya cara untuk meletakkan 4 kartu ke dalam 4 tempat dengan menggunakan perulangan bersarang maka cara mencari posisi 4 kartu tersebut dapat terselesaikan. Perlu diperhatikan pula adanya kartu yang berulang apabila terdapat kartu dengan angka yang sama akan mengurangi kombinasi 4 kartu. Selain itu untuk menghitung kemungkinan operasi juga dapat menggunakan permutasi yaitu 64 kemungkinan dan hal ini juga dapat diselesaikan dengan menggunakan perulangan bersarang. Dikarenakan terdapat 4 kemungkinan operator (+,-,*,/) yang ada di 3 tempat dan operator yang dapat berulang maka.

$$P_{kartu} = \frac{4!}{(4-4)!} = 4! = 24$$

$$P_{operasi} = 4 \times 4 \times 4 = 4^3 = 64$$

Setelah itu terdapat beberapa kemungkinan untuk penggunaan kurung, penggunaan kurung untuk 4 angka adalah dua kurung dan terdapat beberapa kemungkinan untuk penggunaan kurung diantaranya

a. A B C D
b. (A B) C D
c. A (B C) D
d. A B (C D)
e. (A B C) D
f. A (B C D)
g. (A B) (C D)
h. (A (B C)) D
i. ((A B) C) D
j. A (B (C D))
k. A ((B C) D)

Setelah diketahui konfigurasi angka, operator dan juga kurung maka selanjutnya semua kemungkinan untuk membentuk angka 24 dapat dihitung dengan aturan perkalian. Hasil dari aturan perkaliannya adalah $24 \times 64 \times 11 = 16896$ kemungkinan, hasil ini merupakan kemungkinan terbanyak dan dapat berkurang apabila terdapat kartu yang bernilai sama.

II. Source Code

File main.java

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

class UserInput
{
    public static void makeMatrix(double[][] matrix)
    {
        int idx = 0;
        for (int a = 0; a<4 ; a++)
        {
            for (int b = 0 ; b<4 ; b++)
            {
                for (int c = 0 ; c<4 ; c++)
                {
                    for (int d = 0 ; d<4 ; d++)
                    {
                        if (isNotSame(a, b, c, d))
                        {
                            matrix[idx][0] = a;
                            matrix[idx][1] = b;
                            matrix[idx][2] = c;
                            matrix[idx][3] = d;
                            idx++;
                        }
                    }
                }
            }
        }
    }
    public static boolean isNotSame(int a,int b, int c,int d)
    {
        if ((a==b) || (a==c) || (a==d) || (b==c) || (b==d) || (c==d))
        {
            return false;
        }
        return true;
    }
    public static boolean allZero(int[] arr)
    {
        if (arr[0]==0 && arr[1]==0 && arr[2]==0)
        {
            return true;
        }
        else
```

```
        {
            return false;
        }
    }
    public static int[][] kurung(int[] arrOpr)
    {
        int[][] temp = new int[11][3];
        for(int i = 0; i<11; i++)
        {
            for(int j = 0 ; j<3; j++)
            {
                if (arrOpr[j] == 0 || arrOpr[j]==1)
                {
                    temp[i][j] = 1;
                }
                else if (arrOpr[j]==2 || arrOpr[j]==3)
                {
                    temp[i][j] = 3;
                }
            }
            temp[1][0]+= 10;
            temp[2][1]+= 10;
            temp[3][2]+= 10;
            temp[4][0]+= 10;
            temp[4][1]+= 10;
            temp[5][1]+= 10;
            temp[5][2]+= 10;
            temp[6][0]+= 10;
            temp[6][2]+= 10;
            temp[7][0]+= 10;
            temp[7][1]+= 20;
            temp[8][0]+= 20;
            temp[8][1]+= 10;
            temp[9][1]+= 20;
            temp[9][2]+= 10;
            temp[10][1]+= 10;
            temp[10][2]+= 20;
        }
        return temp;
    }
    public static boolean isNotExist(double[][] matrix, double[] arr)
    {
        boolean r = true;
        for (int i = 0 ; i<24 ; i++)
        {
            if (matrix[i][0] == arr[0] & matrix[i][1] == arr[1] &
matrix[i][2] == arr[2] & matrix[i][3] == arr[3])
            {
                r = false;
            }
        }
        return r;
    }
    public static double count(int[] arrOpr, double[] arrNumber, int[] temp)
    {
        double[] tempNum = new double[4];
        for (int i = 0 ; i<4; i++)
```

```
        {
            tempNum[i] = arrNumber[i];
        }
        int max = 0;
        int idmax = 0;
        double temp1 = 0;
        double temp2 = 0;
        double temp3 = 0;
        while (!allZero(temp))
        {
            for (int i = 0; i<3; i++)
            {
                if(temp[i]>max)
                {
                    max = temp[i];
                    idmax = i;
                }
            }
            if (idmax == 0)
            {
                //cari apakah angka kedua sudah pernah kehitung
                if (tempNum[1]==-1)
                {
                    temp1 = operator.calc(tempNum[0],arrOpr[idmax],temp2);
                    temp2 = temp1;
                    temp1 = 0;
                }
                else
                {
                    temp1 = operator.calc(tempNum[0], arrOpr[idmax],
tempNum[1]);
                }
                tempNum[0] = -1;
                tempNum[1] = -1;
                temp[0] = 0;
            }
            else if (idmax == 1)
            {
                //kemungkinan ada temp1 atau temp3 atau keduanya
                if (tempNum[1]==-1 && tempNum[2]==-1)
                {
                    temp2 = operator.calc(temp1,arrOpr[idmax],temp3);
                    temp1 = 0;
                    temp3 = 0;
                }
                else if (tempNum[1] == -1)
                {
                    temp2 = operator.calc(temp1,arrOpr[idmax],tempNum[2]);
                    temp1 = 0;
                }
                else if (tempNum[2]==-1)
                {
                    temp2 = operator.calc(tempNum[1],arrOpr[idmax],temp3);
                    temp3 = 0;
                }
                else
                {
```

```
                    temp2 =
operator.calc(tempNum[1],arrOpr[idmax],tempNum[2]);
                }
                temp[1] = 0;
                tempNum[1] = -1;
                tempNum[2] = -1;
            }
            else if (idmax==2)
            {
                if (tempNum[2]==-1)
                {
                    temp3 = operator.calc(temp2,arrOpr[idmax],tempNum[3]);
                    temp2 = temp3;
                    temp3 = 0;
                }
                else
                {
                    temp3 =
operator.calc(tempNum[2],arrOpr[idmax],tempNum[3]);
                }
                temp[2] = 0;
                tempNum[2] = -1;
                tempNum[3] = -1;
            }
            max = 0;
        }
        if (temp1 != 0 )
        {
            return temp1;
        }
        else if (temp2 != 0)
        {
            return temp2;
        }
        else
        {
            return temp3;
        }
    }
    public static void main(String[] args)
    {
        int idx = 0;
        int arr[]={0,0,0,0};
        boolean valid = false;
        Scanner sc =new Scanner(System.in);
        System.out.println("                              .g8\"\"\"bgd
                                  ");
        System.out.println("                                  .dP'     `M
                                ");
        System.out.println("
pd*\"*b.      .AM       dM'       `   .6\"Yb.  `7MMpMMMb.pMMMb.  .gP\"Ya  ,pP
\"Ybd         ");
        System.out.println("(O)   j8     AVMM       MM          8)   MM    M
M    MM    MM ,M'   Yb 8I   `\"          ");
        System.out.println("    ,;j9   ,W'
MM       MM.    `7MMF',pm9MM    MM    MM    MM 8M\"\"\"\"\"\"\"\"\"\"
`YMMMa.          ");
```

```
        System.out.println(" ,-='     ,W'    MM        `Mb.      MM
8M    MM     MM     MM     MM YM.     , L.    I8          ");
        System.out.println("Ammmmmmm AmmmmmMMmm        `\"bmmmdPY
`Moo9^Yo..JMML  JMML  JMML.`Mbmmd' M9mmmP'          ");
        System.out.println("                MM
                                      ");
        System.out.println("                MM
                                      ");
        boolean menuValid = false;
        String[] kartu = new String[4];
        while (!menuValid)
        {
            System.out.println("================================");
            System.out.println("1. Masukkan 4 kartu");
            System.out.println("2. Random 4 kartu");
            System.out.println("================================");
            System.out.print("Masukkan Menu : ");
            int menu = sc.nextInt();
            if (menu==1)
            {
                System.out.println("Masukkan 4 kartu");
                while(!valid)
                {
                    while (idx<4)
                    {
                        String input = sc.next();
                        if (input.equals("A"))
                        {
                            arr[idx] = 1;
                            kartu[idx] = "A";
                        }
                        else if (input.equals("2"))
                        {
                            arr[idx] = 2;
                            kartu[idx] = "2";
                        }
                        else if (input.equals("3"))
                        {
                            arr[idx] = 3;
                            kartu[idx] = "3";
                        }
                        else if (input.equals("4"))
                        {
                            arr[idx] = 4;
                            kartu[idx] = "4";
                        }
                        else if (input.equals("5"))
                        {
                            arr[idx] = 5;
                            kartu[idx] = "5";
                        }
                        else if (input.equals("6"))
                        {
                            arr[idx] = 6;
                            kartu[idx] = "6";
                        }
                        else if (input.equals("7"))
```

```
                        {
                            arr[idx] = 7;
                            kartu[idx] = "7";
                        }
                        else if (input.equals("8"))
                        {
                            arr[idx] = 8;
                            kartu[idx] = "8";
                        }
                        else if (input.equals("9"))
                        {
                            arr[idx] = 9;
                            kartu[idx] = "9";
                        }
                        else if (input.equals("10"))
                        {
                            arr[idx] = 10;
                            kartu[idx] = "10";
                        }
                        else if (input.equals("J"))
                        {
                            arr[idx] = 11;
                            kartu[idx] = "J";
                        }
                        else if (input.equals("Q"))
                        {
                            arr[idx] = 12;
                            kartu[idx] = "Q";
                        }
                        else if (input.equals("K"))
                        {
                            arr[idx] = 13;
                            kartu[idx] = "K";
                        }
                        idx++;
                        if (arr[3]!=0)
                        {
                            valid = true;
                        }
                    }
                    if (!valid)
                    {
                        System.out.println("Masukan tidak valid, silahkan
ulangi");
                        idx = 0;
                    }
                }
                menuValid = true;
            }
            else if (menu == 2)
            {
                int min = 1;
                int max = 13;
                for (int i = 0 ; i <4 ; i++)
                {
                    int random_int = (int)Math.floor(Math.random() * (max -
min + 1) + min);
```

```
                arr[i] = random_int;
                kartu[i] = operator.converter(random_int);
            }
            menuValid = true;
            System.out.println("List kartu");
            for (int i = 0 ; i < 4 ; i++)
            {
                System.out.print(kartu[i]+ " ");
            }
            System.out.println();
        }
        else
        {
            System.out.println("Masukan salah, silahkan ulangi!");
        }
    }
    /*Matrix to Config */
    // CONFIG -> Konfigurasi dari semua susunan bilangan yang mungkin
    double start = System.currentTimeMillis();
    double[][] config = new double[24][4];
    makeMatrix(config);
    double[][] displayNum = new double[24][4];
    makeMatrix(displayNum);
    double[] tconfig = new double[4];
    for (int i = 0 ; i < 24; i++)
    {
        for (int j = 0; j<4; j++)
        {
            if (config[i][j]==0)
            {
                tconfig[j] = arr[0];
            } else if (config[i][j]==1)
            {
                tconfig[j] = arr[1];
            } else if (config[i][j]==2)
            {
                tconfig[j] = arr[2];
            }
            else if (config[i][j]==3)
            {
                tconfig[j] = arr[3];
            }
        }
        if (isNotExist(config, tconfig))
        {
            for (int j = 0 ; j<4 ; j++)
            {
                if (config[i][j]==0)
                {
                    config[i][j] = arr[0];
                } else if (config[i][j]==1)
                {
                    config[i][j] = arr[1];
                } else if (config[i][j]==2)
                {
                    config[i][j] = arr[2];
                }
```

```
                    else if (config[i][j]==3)
                    {
                        config[i][j] = arr[3];
                    }
                }
            }
            else
            {
                for (int j = 0 ; j<4 ; j++)
                {
                    config[i][j] = 0;
                }
            }
        }


        // Matriks Operator

        int[][] optr = new int[64][3];
        operator.opr(optr);
        int jml = 0;
        List<String> ArrRes = new ArrayList<String>();
        System.out.println("================================");
        for(int i = 0; i<24 ; i++)
        {
            for (int j = 0 ; j<64; j++)
            {
                int[][] temp = kurung(optr[j]);
                for (int l = 0; l<11 ; l++)
                {
                    double res = count(optr[j],config[i],temp[l]);
                    int[][] displayOp = optr;
                    char[] tempOp = new char[3];
                    for (int k = 0 ; k< 3; k++)
                    {
                        if (displayOp[j][k]==0)
                        {
                            tempOp[k] = '+';
                        }
                        else if (displayOp[j][k]==1)
                        {
                            tempOp[k] = '-';
                        }
                        else if (displayOp[j][k]==2)
                        {
                            tempOp[k] = '*';
                        }
                        else if (displayOp[j][k]==3)
                        {
                            tempOp[k] = '/';
                        }
                    }
                    if (res==24 & l==0)
                    {
                        System.out.println(config[i][0] + " " + tempOp[0] +
" " +config[i][1] + " " + tempOp[1] + " " + config[i][2] + " " + tempOp[2] +
" " + config[i][3]);
```

```
                        ArrRes.add(config[i][0] + " " + tempOp[0] + " "
+config[i][1] + " " + tempOp[1] + " " + config[i][2] + " " + tempOp[2] + " "
+ config[i][3]);
                    }
                    else if (res == 24 & l==1)
                    {
                        System.out.println("("+config[i][0] + " " +
tempOp[0] + " " + config[i][1] + ")" + " " + tempOp[1] + " " + config[i][2]
+ " " + tempOp[2] + " " + config[i][3]);
                        ArrRes.add("("+config[i][0] + " " + tempOp[0] + " "
+ config[i][1] + ")" + " " + tempOp[1] + " " + config[i][2] + " " +
tempOp[2] + " " + config[i][3]);
                    }
                    else if (res==24 & l==2)
                    {
                        System.out.println(config[i][0] + " " + tempOp[0] +
" " +"(" +config[i][1] + " " + tempOp[1] + " " + config[i][2]+ ")" + " " +
tempOp[2] + " " + config[i][3]);
                        ArrRes.add(config[i][0] + " " + tempOp[0] + " " +"("
+config[i][1] + " " + tempOp[1] + " " + config[i][2]+ ")" + " " + tempOp[2]
+ " " + config[i][3]);
                    }
                    else if (res==24 & l==3)
                    {
                        System.out.println(config[i][0] + " " + tempOp[0] +
" " +config[i][1] + " " + tempOp[1] + " " + "(" + config[i][2] + " " +
tempOp[2] + " " + config[i][3] + ")");
                        ArrRes.add(config[i][0] + " " + tempOp[0] + " "
+config[i][1] + " " + tempOp[1] + " " + "(" + config[i][2] + " " + tempOp[2]
+ " " + config[i][3] + ")");
                    }
                    else if (res==24 & l==4)
                    {
                        System.out.println("("+config[i][0] + " " +
tempOp[0] + " " +config[i][1] + " " + tempOp[1] + " " + config[i][2] + ")" +
" " + tempOp[2] + " " + config[i][3]);
                        ArrRes.add("("+config[i][0] + " " + tempOp[0] + " "
+config[i][1] + " " + tempOp[1] + " " + config[i][2] + ")" + " " + tempOp[2]
+ " " + config[i][3]);
                    }
                    else if (res==24 & l==5)
                    {
                        System.out.println(config[i][0] + " " + tempOp[0] +
" " +"("+config[i][1] + " " + tempOp[1] + " " + config[i][2] + " " +
tempOp[2] + " " + config[i][3] + ")");
                        ArrRes.add(config[i][0] + " " + tempOp[0] + " "
+"("+config[i][1] + " " + tempOp[1] + " " + config[i][2] + " " + tempOp[2] +
" " + config[i][3] + ")");
                    }
                    else if (res==24 & l==6)
                    {
                        System.out.println("("+config[i][0] + " " +
tempOp[0] + " " +config[i][1]+ ")" + " " + tempOp[1] + " " + "(" +
config[i][2] + " " + tempOp[2] + " " + config[i][3] + ")");
                        ArrRes.add("("+config[i][0] + " " + tempOp[0] + " "
+config[i][1]+ ")" + " " + tempOp[1] + " " + "(" + config[i][2] + " " +
tempOp[2] + " " + config[i][3] + ")");
```

```
                    }
                    else if (res==24 & l==7)
                    {
                        System.out.println("("+config[i][0] + " " +
tempOp[0] + " " +"("+config[i][1] + " " + tempOp[1] + " "  + config[i][2] +
"))" +" " + tempOp[2] + " " + config[i][3]);
                        ArrRes.add("("+config[i][0] + " " + tempOp[0] + " "
+"("+config[i][1] + " " + tempOp[1] + " "  + config[i][2] + "))" +" " +
tempOp[2] + " " + config[i][3]);
                    }
                    else if (res==24 & l==8)
                    {
                        System.out.println(("("+config[i][0] + " " +
tempOp[0] + " " +config[i][1] + ")" +" " + tempOp[1] + " "  + config[i][2] +
")" +" " + tempOp[2] + " " + config[i][3]);
                        ArrRes.add(("("+config[i][0] + " " + tempOp[0] + " "
+config[i][1] + ")" +" " + tempOp[1] + " "  + config[i][2] + ")" +" " +
tempOp[2] + " " + config[i][3]);
                    }
                    else if (res==24 & l==9)
                    {
                        System.out.println(config[i][0] + " " + tempOp[0] +
" " + "(("+config[i][1] + " " + tempOp[1] + " "  + config[i][2]  +")" +" " +
tempOp[2] + " " + config[i][3] + ")");
                        ArrRes.add(config[i][0] + " " + tempOp[0] + " " +
"(("+config[i][1] + " " + tempOp[1] + " "  + config[i][2]  +")" +" " +
tempOp[2] + " " + config[i][3] + ")");
                    }
                    else if (res==24 & l==10)
                    {
                        System.out.println(config[i][0] + " " + tempOp[0] +
" " + "("+config[i][1] + " " + tempOp[1] + " "  + "("+config[i][2]  +" " +
tempOp[2] + " " + config[i][3] + "))");
                        ArrRes.add(config[i][0] + " " + tempOp[0] + " " +
"("+config[i][1] + " " + tempOp[1] + " "  + "("+config[i][2]  +" " +
tempOp[2] + " " + config[i][3] + "))");
                    }
                    if (res==24)
                    {
                        jml++;
                    }
                }
            }
        }
        System.out.println("================================");
        double end = System.currentTimeMillis();
        System.out.println("Total Time : " + (end-start)/1000 + " s" );
        System.out.println("TOTAL = " + jml);
        System.out.print("Apakah Anda ingin menyimpan jawaban ? (1/0) ");
        int save = sc.nextInt();
        if (save == 1)
        {
            System.out.print("Masukkan nama file : ");
            String name = sc.next();
            try (FileWriter fWriter = new FileWriter(
                "./test/" + name))
            {
```

```
                fWriter.write("==========================" + "\n");
                fWriter.write("LIST KARTU" + "\n");
                for (int i = 0 ; i<4; i++)
                {
                    fWriter.write(kartu[i] + " ");
                }
                fWriter.write("\n"+"==========================" + "\n");
                for(int i = 0 ; i<ArrRes.size();i++)
                {
                    try
                    {
                        fWriter.write(ArrRes.get(i) + "\n");
                    }
                    catch (IOException e)
                    {
                        System.out.print(e.getMessage());
                    }
                }
                fWriter.write("==========================" + "\n");
                fWriter.write("Total = " + jml);
            }
            catch (IOException e)
            {
                System.out.print(e.getMessage());
            }
        }
    }
}
```

File operator.java

```
public class operator {
    public static void opr(int[][] matrix)
    {
        int idx = 0 ;
        for (int i = 0 ; i < 4 ; i++)
        {
            for (int j = 0 ; j < 4 ; j++)
            {
                for (int k = 0 ; k < 4 ; k++)
                {
                    matrix[idx][0] = i;
                    matrix[idx][1] = j;
                    matrix[idx][2] = k;
                    idx++;
                }
            }
        }
    }
    public static double calc(double a, int opr, double b)
    {
        if (opr == 0)
        {
```

```
            return a+b;
        } else if (opr == 1)
        {
            return (a-b);
        }
        else if (opr == 2)
        {
            return a*b;
        }
        else
        {
            return a/b;
        }
    }
    public static String converter(int a)
    {
        if(a==1)
        {
            return "A";
        }
        else if(a == 2)
        {
            return "2";
        }
        else if(a == 3)
        {
            return "3";
        }
        else if(a == 4)
        {
            return "4";
        }
        else if(a == 5)
        {
            return "5";
        }
        else if(a == 6)
        {
            return "6";
        }
        else if(a == 7)
        {
            return "7";
        }
        else if(a == 8)
        {
            return "8";
        }
        else if(a == 9)
```

```
        {
            return "9";
        }
        else if(a == 10)
        {
            return "10";
        }
        else if(a == 11)
        {
            return "J";
        }
        else if(a == 12)
        {
            return "Q";
        }
        else
        {
            return "K";
        }
    }
}
```

III. Uji Kasus

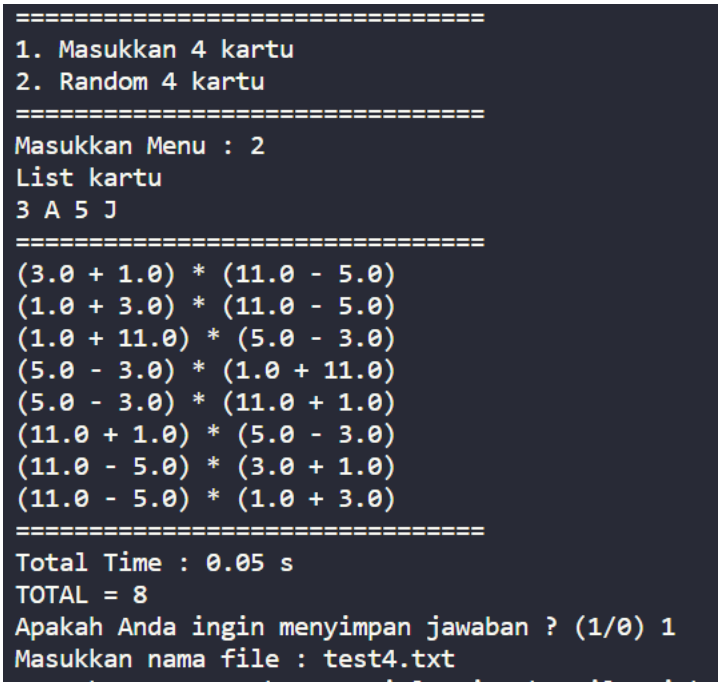
Gambar 1. Test Case I

```
================================
1. Masukkan 4 kartu
2. Random 4 kartu
================================
Masukkan Menu : 2
List kartu
7 9 6 8
================================
6.0 / (9.0 - 7.0) * 8.0
(6.0 / (9.0 - 7.0)) * 8.0
6.0 / ((9.0 - 7.0) / 8.0)
6.0 * 8.0 / (9.0 - 7.0)
(6.0 * 8.0) / (9.0 - 7.0)
6.0 * (8.0 / (9.0 - 7.0))
8.0 / (9.0 - 7.0) * 6.0
(8.0 / (9.0 - 7.0)) * 6.0
8.0 / ((9.0 - 7.0) / 6.0)
8.0 * 6.0 / (9.0 - 7.0)
(8.0 * 6.0) / (9.0 - 7.0)
8.0 * (6.0 / (9.0 - 7.0))
================================
Total Time : 0.045 s
TOTAL = 12
Apakah Anda ingin menyimpan jawaban ? (1/0) 1
Masukkan nama file : test2.txt
```

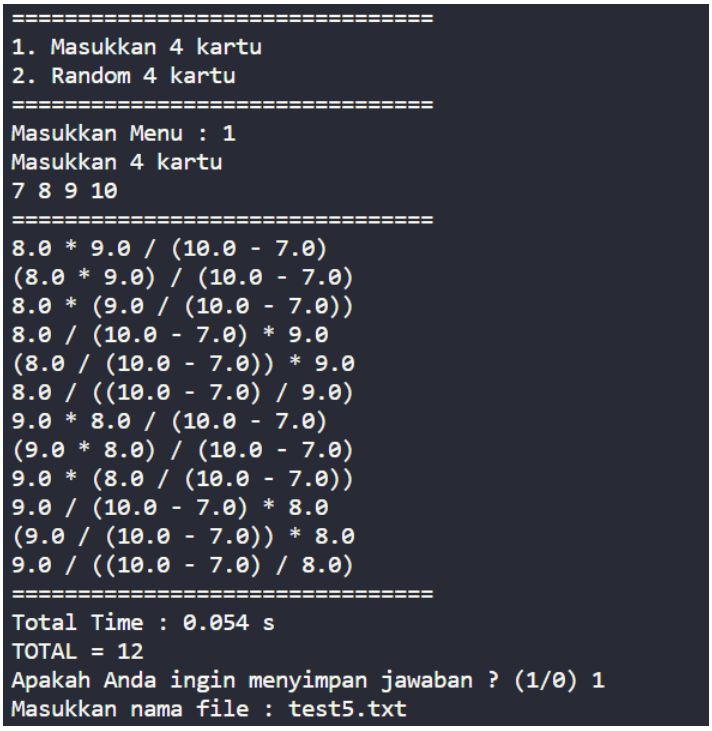
Gambar 2. Test Case II

```
================================
Masukkan Menu : 1
Masukkan 4 kartu
3 4 5 6
================================
(3.0 - 4.0 + 5.0) * 6.0
((3.0 - 4.0) + 5.0) * 6.0
(3.0 - (4.0 - 5.0)) * 6.0
(3.0 + 5.0 - 4.0) * 6.0
(3.0 + (5.0 - 4.0)) * 6.0
((3.0 + 5.0) - 4.0) * 6.0
(5.0 + 3.0 - 4.0) * 6.0
(5.0 + (3.0 - 4.0)) * 6.0
((5.0 + 3.0) - 4.0) * 6.0
(5.0 - 4.0 + 3.0) * 6.0
((5.0 - 4.0) + 3.0) * 6.0
(5.0 - (4.0 - 3.0)) * 6.0
6.0 * (3.0 - 4.0 + 5.0)
6.0 * ((3.0 - 4.0) + 5.0)
6.0 * (3.0 - (4.0 - 5.0))
6.0 * (3.0 + 5.0 - 4.0)
6.0 * ((3.0 + 5.0) - 4.0)
6.0 * (3.0 + (5.0 - 4.0))
6.0 * (5.0 + 3.0 - 4.0)
6.0 * ((5.0 + 3.0) - 4.0)
6.0 * (5.0 + (3.0 - 4.0))
6.0 * (5.0 - 4.0 + 3.0)
6.0 * ((5.0 - 4.0) + 3.0)
6.0 * (5.0 - (4.0 - 3.0))
================================
Total Time : 0.07 s
TOTAL = 24
Apakah Anda ingin menyimpan jawaban ? (1/0) 1
Masukkan nama file : test3.txt
```

Gambar 3. Test Case III



Gambar 4. Test Case IV



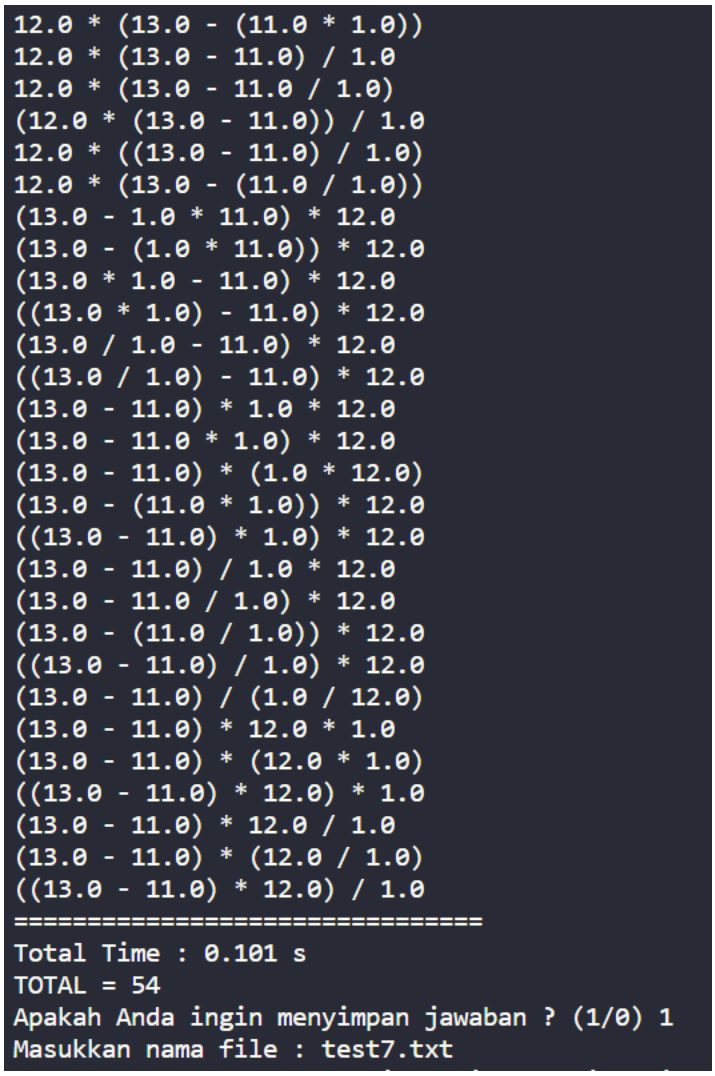
Gambar 5. Test Case V

```
================================
1. Masukkan 4 kartu
2. Random 4 kartu
================================
Masukkan Menu : 2
List kartu
9 K 9 K
================================
================================
Total Time : 0.018 s
TOTAL = 0
Apakah Anda ingin menyimpan jawaban ? (1/0) 1
Masukkan nama file : test6.txt
```

Gambar 6. Test Case VI

```
================================
Masukkan Menu : 1
Masukkan 4 kartu
A J Q K
================================
1.0 * 12.0 * (13.0 - 11.0)
(1.0 * 12.0) * (13.0 - 11.0)
1.0 * (12.0 * (13.0 - 11.0))
1.0 * (13.0 - 11.0) * 12.0
(1.0 * 13.0 - 11.0) * 12.0
(1.0 * (13.0 - 11.0)) * 12.0
((1.0 * 13.0) - 11.0) * 12.0
1.0 * ((13.0 - 11.0) * 12.0)
12.0 * 1.0 * (13.0 - 11.0)
12.0 * (1.0 * 13.0 - 11.0)
(12.0 * 1.0) * (13.0 - 11.0)
12.0 * ((1.0 * 13.0) - 11.0)
12.0 * (1.0 * (13.0 - 11.0))
12.0 / 1.0 * (13.0 - 11.0)
(12.0 / 1.0) * (13.0 - 11.0)
12.0 / (1.0 / (13.0 - 11.0))
12.0 * (13.0 - 1.0 * 11.0)
12.0 * (13.0 - (1.0 * 11.0))
12.0 * (13.0 * 1.0 - 11.0)
12.0 * ((13.0 * 1.0) - 11.0)
12.0 * (13.0 / 1.0 - 11.0)
12.0 * ((13.0 / 1.0) - 11.0)
12.0 * (13.0 - 11.0) * 1.0
12.0 * (13.0 - 11.0 * 1.0)
(12.0 * (13.0 - 11.0)) * 1.0
12.0 * ((13.0 - 11.0) * 1.0)
12.0 * (13.0 - (11.0 * 1.0))
12.0 * (13.0 - 11.0) / 1.0
12.0 * (13.0 - 11.0 / 1.0)
```

Gambar 7. Test Case VII



Gambar 8. Test Case VII

IV. Source Code

https://github.com/henryanandsr/Tucil1_13521004

V. Cek List

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