Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Dibuat sebagai salah satu luaran Tugas Kecil 1

Matthew Mahendra
13521007

Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

Daftar Isi

BAB 1

Latar Belakang

1.1 Permainan Kartu 24

Permainan kartu 24 merupakan suatu permainan yang menggunakan satu set kartu remi. Dari 52 kartu yang ada, diambil 4 yang berbeda secara acak. Dari keempat kartu yang diambil, pemain harus mencari cara agar dapat menghasilkan angka 24 dari operasi matematika pada 4 kartu tersebut.

Operasi matematika yang digunakan adalah operasi penjumlahan, pengurangan, perkalian, pembagian, dan sifat distributif. Nilai pada setiap kartu menjadi seperti demikian,

- Kartu As = 1
- Kartu 2 s.d. 10 = Angka itu sendiri
- Kartu Jack = 11
- Kartu Queen = 12
- Kartu King = 13

1.2 Algoritma Brute Force

Algoritma *brute force* merupakan sebuah algoritma yang menggunakan prinsip *straightforward* untuk menyelesaikan suatu permasalahan. Algoritma ini dapat menyelesaikan hampir semua permasalahan komputasi, namun memiliki kekurangan dalam hal kompleksitas ruang dan waktu untuk kasus-kasus yang besar.

BAB 2

Hasil

2.1 Alur Program dan Aplikasi Algoritma Brute Force

Pada permainan kartu 24, algoritma brute force digunakan sebagai berikut,

1. Dari kartu yang dimasukkan oleh pengguna atau dihasilkan secara otomatis oleh komputer, dipikirkan semua kemungkinan operasi kartu 24 yang dapat digunakan

2. Operasi-operasi tersebut antara lain:

   ```
   (a op b) op (c op d); (a op b op c) op d; ((a op b) op c) op d); (a
       op (b op c)) op d); a op (b op c op d); a op ((b op c) op d); a
       op (b op (c op d)); a op (b op c) op d; a op b op (c op d)
   ```

   dengan op adalah operator matematika; a,b,c,d bilangan 1-13

3. Untuk setiap operasi matematika, dilakukan permutasi dari a,b,c, dan d serta op yang terdiri dari (+, -, *, /). Untuk urutan bilangan terdapat 24 cara penyusunan. Untuk urutan operasi terdapat 24 cara penyusunan juga. Dengan permutasi terhadap operasi-operasi, ada 5184 operasi yang harus diujikan

4. Dari hasil operasi matematika yang diujikan, jika hasilnya adalah 24, maka dianggap permutasi dari bilangan dan operasi matematika adalah benar dan didaftarkan pada hasil yang valid

2.2 Source Code

Program dibuat menggunakan bahasa Java versi 19.0.1. Dibuat 4 file yaitu Input.java, Main.java, Output.java, dan TwentyFour.java.

Input.java berisi kelas untuk melakukan input secara manual ataupun secara otomatis. Main.java merupakan program utama (driver). Output.java berisi kelas untuk melakukan output ke layar, dalam hal ini mengembalikan angka 11-13 menjadi J,Q,K, dan 1 menjadi A serta untuk menyimpan hasil ke suatu file. TwentyFour.java berisi program untuk mencari permutasi operasi matematika yang menghasilkan angka 24 dari empat kartu yang telah diinput.

2.2.1 Input.java

```
import java.util.Scanner;
import java.util.Random;

public class Input{
    int k1,k2,k3,k4;
```

```
    Scanner sc = new Scanner(System.in);

    public void inputManual(){
        System.out.println("Masukkan Kartu Anda");
        /* Membaca input */
        String input_k1 = sc.next();
        String input_k2 = sc.next();
        String input_k3 = sc.next();
        String input_k4 = sc.next();

        /* Convert untuk k1 */
        k1 = stringToInt(input_k1);

        /* Convert untuk k2 */
        k2 = stringToInt(input_k2);

        /* Convert untuk k3 */
        k3 = stringToInt(input_k3);

        /* Convert untuk k4 */
        k4 = stringToInt(input_k4);

        if (k1 == -1000 | k2 == -1000 | k3 == -1000 | k4 == -1000){
            System.out.println("Masukkan salah! Ulangi pemasukan!");
            inputManual();
        }
    }

    public void inputAuto(){
        Random rand = new Random();

        /* Generate Random */
        k1 = rand.nextInt(13) + 1;
        k2 = rand.nextInt(13) + 1;
        k3 = rand.nextInt(13) + 1;
        k4 = rand.nextInt(13) + 1;

        /* Just incase tidak boleh ada duplikat kartu yang sama
        if(k1 == k2 | k1 == k3 | k1 == k4 | k2 == k3 | k2 == k4 | k3 ==
            k4)
        {
            inputAuto();
        }
        */

    }

    public int stringToInt(String c){
        int k;
        if(c.equals("A")){
            k = 1;
        } else if( c.equals("J")){
```

```
            k = 11;
        } else if ( c.equals("Q")){
            k = 12;
        } else if ( c.equals("K")){
            k = 13;
        }else if (c.equals("2")){
            k = 2;
        }else if (c.equals("3")){
            k = 3;
        }else if (c.equals("4")){
            k = 4;
        }else if (c.equals("5")){
            k = 5;
        }else if (c.equals("6")){
            k = 6;
        }else if (c.equals("7")){
            k = 7;
        }else if (c.equals("8")){
            k = 8;
        }else if (c.equals("9")){
            k = 9;
        }else if (c.equals("10")){
            k = 10;
        }else{
            /* FLAG ERROR */
            k = -1000;
        }
        return k;
    }

    public static void main(String[] args){
        Input in = new Input();
        in.inputManual();
        System.out.println(in.k1);
        System.out.println(in.k2);
        System.out.println(in.k3);
        System.out.println(in.k4);
    }
}
```

2.2.2 Main.java

```
import java.util.Scanner;

class Main {
    public static void main(String[] args){
        Output ot = new Output();
        Scanner sc = new Scanner(System.in);
        Input in = new Input();

        System.out.println(" ================ ");
```

```
        System.out.println("  24 CARD GAME ");
        System.out.println(" ================ ");

        System.out.println("Pilih masukan kartu: ");
        System.out.println("1 untuk Manual, 2 untuk Auto");

        int x = sc.nextInt();

        while(x != 1 & x != 2)
        {
            System.out.println("Masukan salah! Silakan ulangi");
            x = sc.nextInt();
        }

        if(x == 1){
            in.inputManual();
        }else{
            in.inputAuto();
        }

        System.out.println("YOUR CARDS: ");
        System.out.print(ot.writeBackCards(in.k1) + " ");
        System.out.print(ot.writeBackCards(in.k2) + " ");
        System.out.print(ot.writeBackCards(in.k3) + " ");
        System.out.println(ot.writeBackCards(in.k4));

        TwentyFour tf = new TwentyFour(in.k1, in.k2, in.k3, in.k4);

        tf.findTwentyFour();

        sc.close();
    }
}
```

2.2.3 Output.java

```
import java.util.Scanner;
import java.util.Random;

public class Input{
    int k1,k2,k3,k4;
    Scanner sc = new Scanner(System.in);

    public void inputManual(){
        System.out.println("Masukkan Kartu Anda");
        /* Membaca input */
        String input_k1 = sc.next();
        String input_k2 = sc.next();
        String input_k3 = sc.next();
        String input_k4 = sc.next();
```

```
        /* Convert untuk k1 */
        k1 = stringToInt(input_k1);

        /* Convert untuk k2 */
        k2 = stringToInt(input_k2);

        /* Convert untuk k3 */
        k3 = stringToInt(input_k3);

        /* Convert untuk k4 */
        k4 = stringToInt(input_k4);

        if (k1 == -1000 | k2 == -1000 | k3 == -1000 | k4 == -1000){
            System.out.println("Masukkan salah! Ulangi pemasukan!");
            inputManual();
        }
    }

    public void inputAuto(){
        Random rand = new Random();

        /* Generate Random */
        k1 = rand.nextInt(13) + 1;
        k2 = rand.nextInt(13) + 1;
        k3 = rand.nextInt(13) + 1;
        k4 = rand.nextInt(13) + 1;

        /* Just incase tidak boleh ada duplikat kartu yang sama
        if(k1 == k2 | k1 == k3 | k1 == k4 | k2 == k3 | k2 == k4 | k3 ==
            k4)
        {
            inputAuto();
        }
        */

    }

    public int stringToInt(String c){
        int k;
        if(c.equals("A")){
            k = 1;
        } else if( c.equals("J")){
            k = 11;
        } else if ( c.equals("Q")){
            k = 12;
        } else if ( c.equals("K")){
            k = 13;
        }else if (c.equals("2")){
            k = 2;
        }else if (c.equals("3")){
            k = 3;
        }else if (c.equals("4")){
```

```
            k = 4;
        }else if (c.equals("5")){
            k = 5;
        }else if (c.equals("6")){
            k = 6;
        }else if (c.equals("7")){
            k = 7;
        }else if (c.equals("8")){
            k = 8;
        }else if (c.equals("9")){
            k = 9;
        }else if (c.equals("10")){
            k = 10;
        }else{
            /* FLAG ERROR */
            k = -1000;
        }
        return k;
    }

    public static void main(String[] args){
        Input in = new Input();
        in.inputManual();
        System.out.println(in.k1);
        System.out.println(in.k2);
        System.out.println(in.k3);
        System.out.println(in.k4);
    }
}
```

2.2.4 TwentyFour.java

```
import java.util.Scanner;
public class TwentyFour {
    Output out = new Output();

    int[] arrAwal = new int[4];
    public int found = 0;
    String operationText;

    /* CONSTRUCTOR */
    public TwentyFour(int a, int b, int c, int d){
        arrAwal[0] = a;
        arrAwal[1] = b;
        arrAwal[2] = c;
        arrAwal[3] = d;
    }

    /* METHODS */
    public void findTwentyFour(){
        Scanner opt = new Scanner(System.in);
```

```
double[][] arrKartu = new double[24][4];
int n = 0;
for (int i = 0; i < 4; i++){
    for (int j = 0; j < 4; j++){
        for (int k = 0; k < 4; k++){
            for (int l = 0; l < 4 ; l++){
                if ( i == j | i == k | i == l | j == k | j == l
                    | k == l)
                {
                    /* PASS */
                }
                else
                {
                    arrKartu[n][0] = arrAwal[i];
                    arrKartu[n][1] = arrAwal[j];
                    arrKartu[n][2] = arrAwal[k];
                    arrKartu[n][3] = arrAwal[l];
                    n++;
                }
            }
        }
    }
}

long start = System.currentTimeMillis();

for (int i = 0; i < 24; i++){
    operations(arrKartu[i]);
}

long end = System.currentTimeMillis();

long dur = end-start;

System.out.println("RESULTS: " + found + " combinations");
System.out.println("================================");

if (found == 0){
    System.out.println("NO RESULTS");
}else{
    for (int i = 0; i < out.results.size(); i++){
        System.out.println(out.results.get(i));
    }
}

System.out.println("================================");
System.out.println("Runtime " + dur + "ms");

boolean option_bool = false;

while(!option_bool){
```

```
            System.out.println("Do you want to save the results? (Y/N)"
               );

            String option = opt.nextLine();

            if(option.equals("Y") | option.equals("y")){
                option_bool = true;
                out.writeToFile(found, dur, out.results, arrAwal);
                System.out.println("File is saved with filename " + out
                   .fileName);
            }else if(option.equals("N") | option.equals("n")){
                option_bool = true;
            }else{
                System.out.println("Masukan salah! Silakan ulangi.");
            }
        }

        opt.close();
    }

    public void operations(double[] arr){
        /* OPERASI DENGAN TANDA KURUNG */

        /* (a operator b) operator (c operator d) */
        for(int i = 0; i < 4; i++){
            double ab, cd;
            String abText = "(";
            ab = operate(arr[0], i, arr[1]);
            abText += makeStringFromOp(arr[0], i, arr[1]);
            abText += ")";
            for (int j = 0 ; j < 4; j++){
                cd = operate(arr[2], j, arr[3]);
                String cdText = "(";
                cdText += makeStringFromOp(arr[2], j, arr[3]) + ")";
                for (int k = 0; k < 4; k++){
                    if(operate(ab, k, cd) == 24){
                        operationText = concateOperationsString(abText,
                            k, cdText);
                        found();
                    }
                }
            }
        }

        /* ((a operator b) operator c) operator d */
        for (int i = 0; i < 4; i++){
            double ab;
            String abText = "(";
            ab = operate(arr[0], i, arr[1]);
            abText += makeStringFromOp(arr[0], i, arr[1]) + ")";
            for (int j = 0; j < 4 ; j++){
                double c;
```

```
            c = operate(ab, j, arr[2]);
            String cText = "(" + concateStringOp(abText, j, arr[2])
                + ")";
            for (int k = 0; k<4 ;k++){
                if(operate(c, k, arr[3]) == 24){
                    operationText = concateStringOp(cText, k, arr
                       [3]);
                    found();
                }
            }
        }
    }

    /* (a operator (b operator c)) operator d */
    for (int i = 0; i < 4; i++){
        double bc;
        bc = operate(arr[1], i, arr[2]);
        String bcText = "(" + makeStringFromOp(arr[1], i, arr[2]) +
            ")";
        for (int j = 0; j < 4 ; j++){
            double abc;
            abc = operate(arr[0], j, bc);
            String abcText = "(" + concateOpString(arr[0], j,
               bcText) +")";
            for (int k = 0; k<4; k++){
                if(operate(abc, k, arr[3]) == 24){
                    operationText = concateStringOp(abcText, k, arr
                       [3]);
                    found();
                }
            }
        }
    }

    /* a operator (b operator c) operator d */
    for (int i = 0; i < 4 ; i++){
        double bc;
        bc = operate(arr[1], i, arr[2]);
        String bcText = "(" + makeStringFromOp(arr[1], i, arr[2]) +
            ")";
        for (int j = 0; j < 4 ; j++){
            for(int k = 0; k < 4 ;k++){
                if((j >= 2 & k < 2) | (j >= 2 & k >=2) | (j < 2 & k
                    < 2)){
                    double abc;
                    abc = operate(arr[0], j, bc);
                    String abcText = concateOpString(arr[0], j,
                       bcText);
                    if(operate(abc, k, arr[3])==24 & j != 2){
                        operationText = concateStringOp(abcText, k,
                            arr[3]);
                        found();
```

```
                }
            }else{
                double bcd;
                bcd = operate(bc, k, arr[3]);
                String bcdText = concateStringOp(bcText, k, arr
                    [3]);
                if(operate(arr[0], j, bcd)==24){
                    operationText = concateOpString(arr[0], j,
                        bcdText);
                    found();
                }
            }
        }
    }
}

/* a operator b operator (c operator d) */
for (int i = 0; i < 4; i++){
    double cd;
    cd = operate(arr[2], i, arr[3]);
    String cdText = "(" + makeStringFromOp(arr[2], i, arr[3]) +
        ")";
    for (int j = 0; j < 4; j++){
        for (int k = 0; k < 4; k++){
            if((j >= 2 & k < 2) | (j>=2 & k >= 2) | (j < 2 & k
                < 2)){
                double ab;
                ab = operate(arr[0], j, arr[1]);
                String abText = makeStringFromOp(arr[0], j, arr
                    [1]);
                if(operate(ab, k, cd) == 24){
                    operationText = concateOperationsString(
                        abText, k, cdText);
                    found();
                }
            }else{
                double bcd;
                bcd = operate(arr[1], k,cd);
                String bcdText = concateOpString(arr[1], k,
                    cdText);
                if(operate(arr[0], j, bcd) == 24){
                    operationText = concateOpString(arr[0], j,
                        bcdText);
                    found();
                }
            }
        }
    }
}

/* a operator (b operator (c operator d)) */
for (int i = 0; i < 4; i++){
```

```
            double cd;
            cd = operate(arr[2], i, arr[3]);
            String cdText = "("+makeStringFromOp(arr[2], i, arr[3])+")"
               ;
            for (int j = 0; j < 4; j++){
                double bcd = operate(arr[1], j, cd);
                String bcdText = "(" + concateOpString(arr[1], j,
                   cdText) + ")";
                for (int k = 0; k < 4; k++){
                    if(operate(arr[0], k, bcd) == 24){
                        operationText = concateOpString(arr[0], k,
                           bcdText);
                        found();
                    }
                }
            }
        }

        /* a operator ((b operator c) operator d) */
        for (int i = 0; i < 4; i++){
            double bc = operate(arr[1], i, arr[2]);
            String bcText = "(" + makeStringFromOp(arr[1], i, arr[2]) +
                ")";
            for (int j = 0; j < 4; j++){
                double bcd = operate(bc, j, arr[3]);
                String bcdText = "(" + concateStringOp(bcText, j, arr
                   [3]) + ")";
                for (int k = 0; k < 4 ; k++){
                    if(operate(arr[0], k,bcd) == 24){
                        operationText = concateOpString(arr[0], k,
                           bcdText);
                        found();
                    }
                }
            }
        }

        /* (a operate b operate c) operate d */
        for (int i =0; i < 4; i++){
            for (int j = 0; j < 4; j++){
                double temp;
                String tempText;
                if ((i >= 2 & j < 2) | (i < 2 & j < 2) | (i >= 2 & j >=
                    2)){
                    temp = operate(arr[0], i, arr[1]);
                    temp = operate(temp, j, arr[2]);
                }else{
                    temp = operate(arr[1],j, arr[2]);
                    temp = operate(arr[0], i, temp);
                }
                tempText = "(" + makeStringFromOp(arr[0],i, arr[1]);
                tempText = concateStringOp(tempText, j, arr[2]) + ")";
```

```
                for (int k = 0; k < 4; k++){
                    if(operate(temp, k, arr[3]) == 24){
                        operationText = concateStringOp(tempText, k,
                           arr[3]);
                        found();
                    }
                }
            }
        }

        /* a operate (b operate c operate d) */
        for (int i =0; i < 4; i++){
            for (int j = 0; j < 4; j++){
                double bc, cd, bcd;
                String tempText;
                if ((i >= 2 & j < 2) | (i < 2 & j < 2) | (i >= 2 & j >=
                    2)){
                    bc = operate(arr[1], i, arr[2]);
                    bcd = operate(bc, j, arr[3]);
                }else{
                    cd = operate(arr[2],j, arr[3]);
                    bcd = operate(arr[1], i, cd);
                }
                tempText = "(" + makeStringFromOp(arr[1],i, arr[2]);
                tempText = concateStringOp(tempText, j, arr[3]) + ")";
                for (int k = 0; k < 4; k++){
                    if(operate(arr[0], k, bcd) == 24){
                        operationText = concateOpString(arr[0], k,
                           tempText);
                        found();
                    }
                }
            }
        }


    }

    public double operate (double a, int op, double b){
        double ret = 0;
        if (op == 0){
            ret = a + b;
        } else if (op == 1){
            ret = a - b;
        } else if (op == 2){
            ret = a * b;
        } else if (op == 3){
            ret = a / b;
        }
        return ret;
    }
```

```
public String concateOperationsString(String a, int op, String b){
    String text;
    text = "";
    text += a;
    if(op == 0){
        text += " + ";
    }else if(op == 1){
        text += " - ";
    }else if(op == 2){
        text += " * ";
    }else if(op == 3){
        text += " / ";
    }
    text+=b;
    return text;
}

public String concateStringOp(String a, int op, double b){
    String text;
    text = "";
    text += a;
    if(op == 0){
        text += " + ";
    }else if(op == 1){
        text += " - ";
    }else if(op == 2){
        text += " * ";
    }else if(op == 3){
        text += " / ";
    }
    text+=b;
    return text;
}

public String concateOpString(double a, int op, String b){
    String text;
    text = "";
    text += a;
    if(op == 0){
        text += " + ";
    }else if(op == 1){
        text += " - ";
    }else if(op == 2){
        text += " * ";
    }else if(op == 3){
        text += " / ";
    }
    text+=b;
    return text;
}

public String makeStringFromOp(double a, int op, double b){
```

```
        String text;
        text = "";
        text += a;
        if(op == 0){
            text += " + ";
        }else if(op == 1){
            text += " - ";
        }else if(op == 2){
            text += " * ";
        }else if(op == 3){
            text += " / ";
        }
        text+=b;
        return text;
    }

    public void found(){
    /* WRITE KE OUTPUT
        * Periksa apakah elemen yang ditambahkan
        * sudah ada atau belum pada array
    */
        if(!out.results.contains(operationText)){
            /* Tambahkan ke array */
            out.addResults(operationText);

            /* Tambah Jumlah found */
            found++;
        }
    }
}
```

2.3 Contoh Penyelesaian Kasus

2.3.1 Kartu A A A A

```
 ================
  24 CARD GAME
 ================
Pilih masukan kartu:
1 untuk Manual, 2 untuk Auto
1
Masukkan Kartu Anda
A A A A
YOUR CARDS:
A A A A
RESULTS: 0 combinations
==================================
NO RESULTS
==================================
Runtime 35ms
Do you want to save the results? (Y/N)
Y
PLEASE ENTER YOUR FILENAME:
A A A A
File is saved with filename A A A A.txt
```

Gambar 2.1: Masukan dan Luaran untuk Kartu 'A A A A'

2.3.2 Kartu 6 2 J 6

```
 ================
  24 CARD GAME
 ================
Pilih masukan kartu:
1 untuk Manual, 2 untuk Auto
1
Masukkan Kartu Anda
6 2 J 6
YOUR CARDS:
6 2 J 6
RESULTS: 8 combinations
==================================
((6.0 / 6.0) + 11.0) * 2.0
(6.0 / 6.0 + 11.0) * 2.0
2.0 * ((6.0 / 6.0) + 11.0)
2.0 * (6.0 / 6.0 + 11.0)
2.0 * (11.0 + (6.0 / 6.0))
2.0 * (11.0 + 6.0 / 6.0)
(11.0 + (6.0 / 6.0)) * 2.0
(11.0 + 6.0 / 6.0) * 2.0
==================================
Runtime 36ms
Do you want to save the results? (Y/N)
Y
PLEASE ENTER YOUR FILENAME:
6 2 J 6
File is saved with filename 6 2 J 6.txt
```

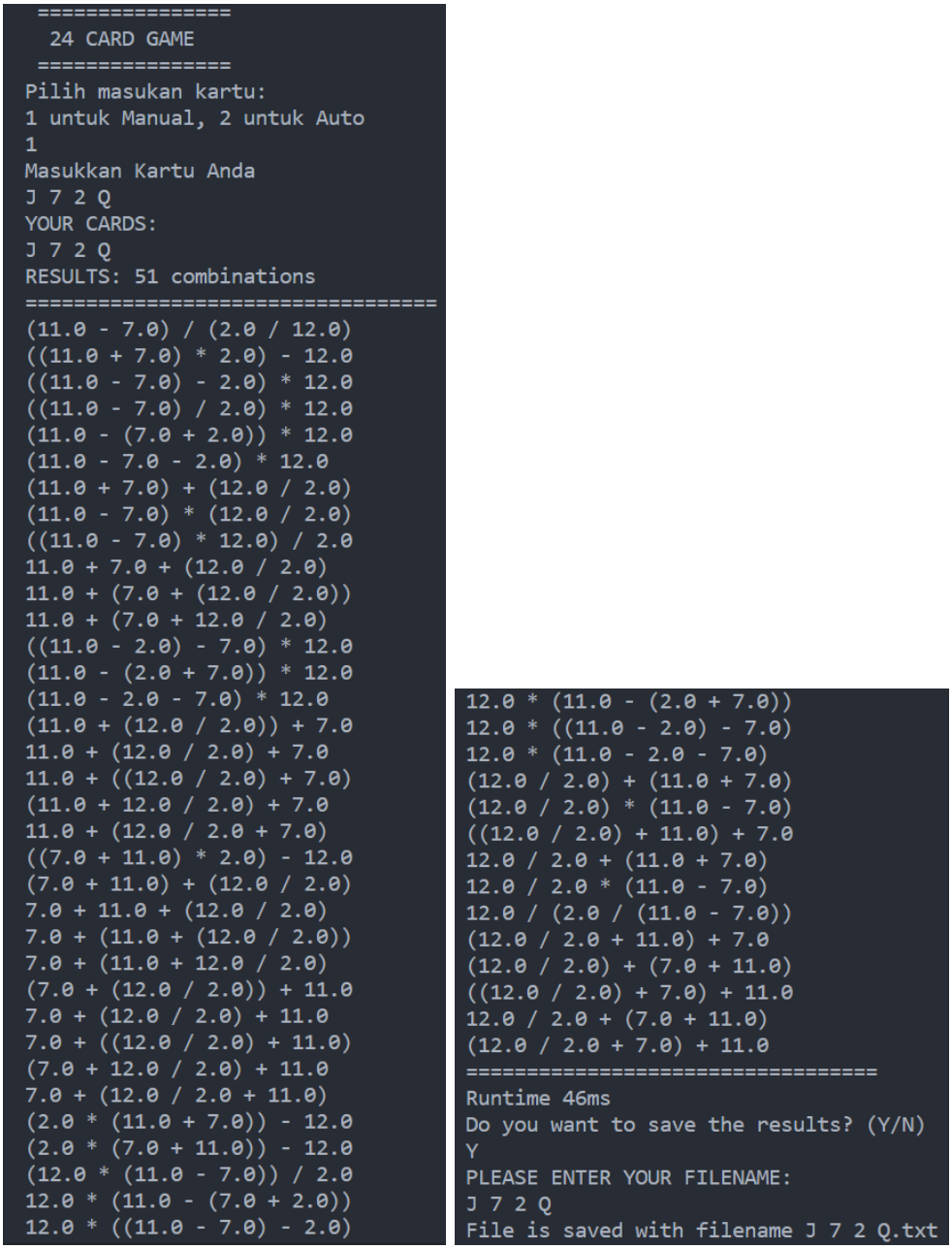
Gambar 2.2: Masukan dan Luaran untuk Kartu '6 2 J 6'

2.3.3 Kartu 7 5 8 8

```
  ================
   24 CARD GAME
  ================
Pilih masukan kartu:
1 untuk Manual, 2 untuk Auto
1
Masukkan Kartu Anda
7 5 8 8
YOUR CARDS:
7 5 8 8
RESULTS: 28 combinations
==================================
((7.0 - 5.0) * 8.0) + 8.0
(7.0 + 8.0) / (5.0 / 8.0)
((7.0 + 8.0) / 5.0) * 8.0
(7.0 + 8.0) * (8.0 / 5.0)
((7.0 + 8.0) * 8.0) / 5.0
(8.0 + 7.0) / (5.0 / 8.0)
((8.0 + 7.0) / 5.0) * 8.0
(8.0 * (7.0 - 5.0)) + 8.0
8.0 + (7.0 - 5.0) * 8.0
8.0 + ((7.0 - 5.0) * 8.0)
(8.0 + 7.0) * (8.0 / 5.0)
((8.0 + 7.0) * 8.0) / 5.0
(8.0 * (7.0 + 8.0)) / 5.0
8.0 * ((7.0 + 8.0) / 5.0)
(8.0 / 5.0) * (7.0 + 8.0)
8.0 - (5.0 - 7.0) * 8.0
8.0 / 5.0 * (7.0 + 8.0)
8.0 / (5.0 / (7.0 + 8.0))
8.0 - ((5.0 - 7.0) * 8.0)
(8.0 / 5.0) * (8.0 + 7.0)
8.0 / 5.0 * (8.0 + 7.0)
8.0 / (5.0 / (8.0 + 7.0))
(8.0 * (8.0 + 7.0)) / 5.0
8.0 + 8.0 * (7.0 - 5.0)
8.0 + (8.0 * (7.0 - 5.0))
8.0 * ((8.0 + 7.0) / 5.0)
8.0 - 8.0 * (5.0 - 7.0)
8.0 - (8.0 * (5.0 - 7.0))
==================================
Runtime 28ms
Do you want to save the results? (Y/N)
Y
PLEASE ENTER YOUR FILENAME:
7 5 8 8
File is saved with filename 7 5 8 8.txt
```

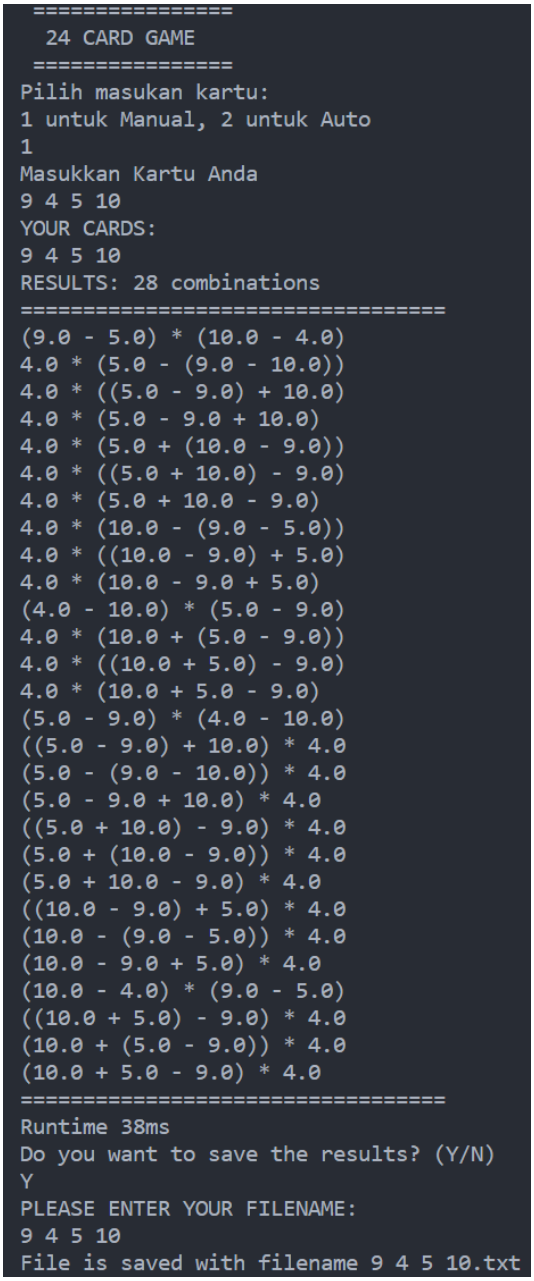
Gambar 2.3: Masukan dan Luaran untuk Kartu ’7 5 8 8’

2.3.4 Kartu J 7 2 Q



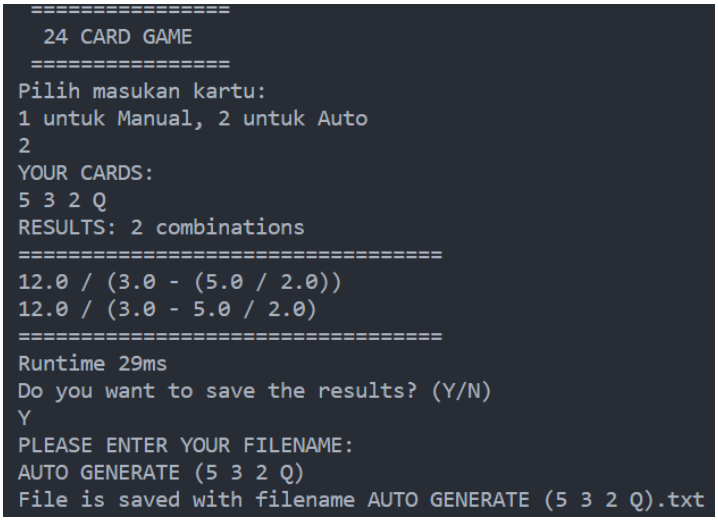
Gambar 2.4: Masukan dan Luaran untuk Kartu 'J 7 2 Q'

2.3.5 Kartu 9 4 5 10



Gambar 2.5: Masukan dan Luaran untuk Kartu ’9 4 5 10’

2.3.6 Kartu Auto Generate



Gambar 2.6: Masukan dan Luaran untuk Kartu yang dihasilkan Otomatis

BAB 3

Simpulan

3.1 Simpulan

Dari tugas kecil ini, dapat disimpulkan hal-hal sebagai berikut,

1. Algoritma Brute Force dapat diaplikasikan pada berbagai permasalahan untuk mendapatkan hasil yang diinginkan
2. Permainan Kartu 24 dapat menggunakan algoritma brute force dalam penyelesaiaannya
3. Brute force tidak harus selalu di hard-code, tetapi dapat ditentukan pola-pola yang dapat dihasilkan.

Lampiran A

Pranala Github

Tugas ini sudah dipublikasi pada Github dengan pranala `https://github.com/MHEN2606/Tucil1_13521007`

Lampiran B

Check 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	✓	

Tabel B.1: Tabel Check List