LAPORAN TUGAS KECIL 01 STRATEGI ALGORITMA: PENYELESAIAN PERMAINAN KARTU 24 DENGAN ALGORITMA BRUTE FORCE

Disusun oleh:

13521012 – Haikal Ardzi Shofiyyurrohman

INSTITUT TEKNOLOGI BANDUNG

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

2023

BAB I
ALGORITMA BRUTE FORCE DAN IMPLEMENTASINYA PADA PERMAINAN KARTU 24

1.1. Algoritma *Brute Force*

Algoritma *brute force* adalah suatu pendekatan yang *straightforward* untuk memecahkan suatu masalah, biasanya algoritma *brute force* didasarkan pada pernyataan persoalan (*problem statement*) dan definisi/konsep yang dilibatkan. Algoritma *brute force* memecahkan persoalan dengan sangat sederhana, langsung, dan jelas caranya.

Algoritma *brute force* umumnya tidak "cerdas" dan tidak efisien, karena ia membutuhkan volume dan komputasi yang besar dan waktu yang lama dalam penyelesaiannya. Algoritma *brute force* lebih cocok untuk persoalan yang ukuran masukannya (n) kecil. Meskipun bukan metode *problem solving* yang efisien, namun hampir semua persoalan dapat diselesainkan dengan algoritma *brute force*.

1.2 Permainan Kartu 24

Permainan kartu 24 adalah permainan kartu aritmatika dengan tujuan mencari cara untuk mengubah 4 buah angka random sehingga mendapatkan hasil akhir sejumlah 24. Permainan 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 (×), divisi (/) dan tanda kurung (()).

1.3 Algoritma Penyelesaian Permainan Kartu 24 dengan Pendekatan Brute Force

Program Permainan Kartu 24 menggunakan algoritma *brute force* untuk mendapatkan semua kemungkinan solusi yang ada. Adapun langkah-langkah untuk mendapatkan semua solusi tersebut ialah sebagai berikut:

1. Program meminta pengguna untuk memilih antara melakukan input 4 kartu secara mandiri atau dipilih secara acak oleh program.
2. Program akan mencari semua kemungkinan urutan kartu dengan permutasi dan dilakukan.
3. Setiap urutan kartu yang didapat akan dihitung kemungkinannya untuk menjadi 24 dengan semua kemungkinan kombinasi operasi sebagai berikut:
   (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
   op adalah operator dari aritmatika (+,-,*,/).
4. Untuk mendapatkan semua kemungkinan solusi, dicek dengan setiap kemungkinan kombinasi operasi serta permutasi angka menghasilkan angka tepat 24. Jika didapatkan angka 24 maka bentuk solusi akan dimasukkan ke dalam list.
5. Lalu dilakukan penghapusan duplikasi dari setiap elemen list.
6. Setelahnya, ditampilkan jumlah banyaknya solusi, semua solusi yang didapatkan, dan waktu estimasi

BAB II

SOURCE CODE DALAM BAHASA JAVA

2.1. BruteForce.java

```
import java.util.ArrayList;
import java.util.List;
import java.util.Collections;

public class BruteForce {

    public static List<Integer> toInteger(Object[] cardValue){
        List<Integer> newCard = new ArrayList<>(4);
        for (int i = 0; i < cardValue.length; i++) {
            newCard.add(i, (Integer) cardValue[i]);
        }
        return newCard;
    }

    public static void permutasi(List<List<Integer>> allPermutation,
List<Integer> cardValue, int j){
        for (int i = j; i < cardValue.size(); i++) {
            Collections.swap(cardValue, i, j);
            permutasi(allPermutation, cardValue, j+1);
            Collections.swap(cardValue, j, i);
        }
        if (j == cardValue.size()-1){
            allPermutation.add(toInteger(cardValue.toArray()));
        }
    }

    public static List<List<Integer>> removeIntDuplicates(List<List<Integer>>
list){
        // Create a new ArrayList
        List<List<Integer>> newList = new ArrayList<List<Integer>>();

        // Traverse through the first list
        for (int i = 0; i < list.size(); i++) {
            if (!newList.contains(list.get(i))) {
                newList.add(list.get(i));
            }
        }

        // return the new list
        return newList;
    }
```

```
    public static List<List<String>> removeStringDuplicates(List<List<String>>
list){
        // Create a new ArrayList
        List<List<String>> newList = new ArrayList<List<String>>();

        // Traverse through the first list
        for (int i = 0; i < list.size(); i++) {
            if (!newList.contains(list.get(i))) {
                newList.add(list.get(i));
            }
        }

        // return the new list
        return newList;
    }

    public static void printHasilPlusMin(List<String> result,
List<Integer>cardValue, String OPBRACKET1, String OPBRACKET2, String
OPBRACKET3, int a, int b, int c, int d){
        result.add("(("+a+OPBRACKET1+b+")"+OPBRACKET2+c+")"+OPBRACKET3+d);
        result.add("("+a+OPBRACKET1+"("+b+OPBRACKET2+c+"))"+OPBRACKET3+d);
        result.add("("+a+OPBRACKET1+b+")"+OPBRACKET2+"("+c+OPBRACKET3+d+")");
        result.add(a+OPBRACKET1+"(("+b+OPBRACKET2+c+")"+OPBRACKET3+d+")");
    }

    public static List<String> getAllResult(List<Integer> cardValue) {
        //Kamus
        String[] opp={"+","-","*","/"};
        List<String> allResult  = new ArrayList<>();
        List<List<Integer>> allPermutation = new ArrayList<List<Integer>>();

        //Algoritma
        permutasi(allPermutation, cardValue, 0);
        allPermutation=removeIntDuplicates(allPermutation);

        for (int i = 0; i < allPermutation.size(); i++) {
            int a=allPermutation.get(i).get(0);
            int b=allPermutation.get(i).get(1);
            int c=allPermutation.get(i).get(2);
            int d=allPermutation.get(i).get(3);
            //
            if(a+b+c+d==24){
                printHasilPlusMin(allResult, cardValue, opp[0], opp[0],
opp[0], a, b, c, d);
            }
            if(a+b+c-d==24){
                printHasilPlusMin(allResult, cardValue, opp[0], opp[0],
opp[1], a, b, c, d);
```

```
            }
            if(a+b-c+d==24){
                printHasilPlusMin(allResult, cardValue, opp[0], opp[1],
opp[0], a, b, c, d);
            }
            if(a-b+c+d==24){
                printHasilPlusMin(allResult, cardValue, opp[1], opp[0],
opp[0], a, b, c, d);
            }
            if(a+b-c-d==24){
                printHasilPlusMin(allResult, cardValue, opp[0], opp[1],
opp[1], a, b, c, d);
            }
            if(a-b+c-d==24){
                printHasilPlusMin(allResult, cardValue, opp[1], opp[0],
opp[1], a, b, c, d);
            }
            if(a-b-c+d==24){
                printHasilPlusMin(allResult, cardValue, opp[1], opp[1],
opp[0], a, b, c, d);
            }
            if(((a*b)+c)+d==24){
                allResult.add("(("+a+"*"+b+")+"+c+")+"+d);
            }
            if((a*(b+c))+d==24){
                allResult.add("("+a+"*("+b+"))+"+c+"+"+d+"");
            }
            if((a*b)+(c+d)==24){
                allResult.add("(("+a+"*"+b+")+("+c+"+"+d+")");
            }
            if(a*(b+(c+d))==24){
                allResult.add(""+a+"*("+b+"+("+c+"+"+d+"))");
            }
            if(((a+b)+c)*d==24){
                allResult.add("(("+a+"+"+b+")+"+c+")*"+d);
            }
            if((a+(b+c))*d==24){
                allResult.add("("+a+"+("+b+"))+"+c+"*"+d+"");
            }
            if((a+b)+(c*d)==24){
                allResult.add("(("+a+"+"+b+")+("+c+"*"+d+")");
            }
            if(a+(b+(c*d))==24){
                allResult.add(""+a+"+("+b+"+("+c+"*"+d+"))");
            }
            if(((a+b)+c)/d==24){
                allResult.add("(("+a+"+"+b+")+"+c+")/"+d);
            }
```

```
            if((a+(b+c))/d==24){
                allResult.add("("+a+"+("+b+"))+"+c+"/"+d+"");
            }
            if((a+b)+(c/d)==24){
                allResult.add("(("+a+"+"+b+")+("+c+"/"+d+")");
            }
            if(a+(b+(c/d))==24){
                allResult.add(""+a+"+"+b+"+("+c+"(/"+d+"))");
            }
            if(((a+b)-c)*d==24){
                allResult.add("(("+a+"+"+b+")-"+c+")*"+d);
            }
            if((a+(b-c))*d==24){
                allResult.add("("+a+"+("+b+"))-"+c+"*"+d+"");
            }
            if((a+b)-(c*d)==24){
                allResult.add("(("+a+"+"+b+")-("+c+"*"+d+")");
            }
            if(a+(b-(c*d))==24){
                allResult.add(""+a+"+("+b+"-("+c+"*"+d+"))");
            }
            if(((a+b)-c)/d==24){
                allResult.add("(("+a+"+"+b+")-"+c+")/"+d);
            }
            if((a+(b-c))/d==24){
                allResult.add("("+a+"+("+b+"))-"+c+"/"+d+"");
            }
            if((a+b)-(c/d)==24){
                allResult.add("(("+a+"+"+b+")-("+c+"/"+d+")");
            }
            if(a+(b-(c/d))==24){
                allResult.add(""+a+"+("+b+"-("+c+"/"+d+"))");
            }
            if(((a+b)*c)+d==24){
                allResult.add("(("+a+"+"+b+")*"+c+")+"+d);
            }
            if((a+(b*c))+d==24){
                allResult.add("("+a+"+("+b+"))*"+c+"+"+d+"");
            }
            if((a+b)*(c+d)==24){
                allResult.add("(("+a+"+"+b+")*("+c+"+"+d+")");
            }
            if(a+(b*(c+d))==24){
                allResult.add(""+a+"+("+b+"*("+c+"+"+d+"))");
            }
            if(((a+b)*c)-d==24){
                allResult.add("(("+a+"+"+b+")*"+c+")-"+d);
            }
```

```
            if((a+(b*c))-d==24){
                allResult.add("("+a+"+("+b+"))*"+c+"-"+d+"");
            }
            if((a+b)*(c-d)==24){
                allResult.add(("+a+"+"+b+")*("+c+"-"+d+")");
            }
            if(a+(b*(c-d))==24){
                allResult.add(""+a+"+("+b+"*("+c+"-"+d+"))");
            }
            if(((a+b)*c)*d==24){
                allResult.add(("+a+"+"+b+")*"+c+")*"+d);
            }
            if((a+(b*c))*d==24){
                allResult.add("("+a+"+("+b+"))*"+c+"*"+d+"");
            }
            if((a+b)*(c*d)==24){
                allResult.add(("+a+"+"+b+")*("+c+"*"+d+")");
            }
            if(a+(b*(c*d))==24){
                allResult.add(""+a+"+("+b+"*("+c+"*"+d+"))");
            }
            if(((a+b)*c)/d==24){
                allResult.add(("+a+"+"+b+")*"+c+")/"+d);
            }
            if((a+(b*c))/d==24){
                allResult.add("("+a+"+("+b+"))*"+c+"/"+d+"");
            }
            if((a+b)*(c/d)==24){
                allResult.add(("+a+"+"+b+")*("+c+"/"+d+")");
            }
            if(a+(b*(c/d))==24){
                allResult.add(""+a+"+("+b+"*("+c+"/"+d+"))");
            }
            if(((a+b)/c)+d==24){
                allResult.add(("+a+"+"+b+")/"+c+")+"+d);
            }
            if((a+(b/c))+d==24){
                allResult.add("("+a+"+("+b+"))/"+c+"+"+d+"");
            }
            if((a+b)/(c+d)==24){
                allResult.add(("+a+"+"+b+")/("+c+"+"+d+")");
            }
            if(a+(b/(c+d))==24){
                allResult.add(""+a+"+("+b+"/("+c+"+"+d+"))");
            }
            if(((a+b)/c)-d==24){
                allResult.add(("+a+"+"+b+")/"+c+")-"+d);
            }
```

```
            if((a+(b/c))-d==24){
                allResult.add("("+a+"+("+b+"))/"+c+"-"+d+"");
            }
            if(c-d!=0){
                if((a+b)/(c-d)==24){
                    allResult.add(("+a+"+"+b+")/("+c+"-"+d+")");
                }
                if(a+(b/(c-d))==24){
                    allResult.add(""+a+"+("+b+"/("+c+"-"+d+"))");
                }
            }
            if(((a+b)/c)*d==24){
                allResult.add(("+a+"+"+b+")/"+c+")*"+d);
            }
            if((a+(b/c))*d==24){
                allResult.add("("+a+"+("+b+"))/"+c+"*"+d+"");
            }
            if((a+b)/(c*d)==24){
                allResult.add(("+a+"+"+b+")/("+c+"*"+d+")");
            }
            if(a+(b/(c*d))==24){
                allResult.add(""+a+"+("+b+"/("+c+"*"+d+"))");
            }
            if(((a+b)/c)/d==24){
                allResult.add(("+a+"+"+b+")/"+c+")/"+d);
            }
            if((a+(b/c))/d==24){
                allResult.add("("+a+"+("+b+"))/"+c+"/"+d+"");
            }
            if(c>=d&&a+b>=(c/d)){
                if((a+b)/(c/d)==24){
                    allResult.add(("+a+"+"+b+")/("+c+"/"+d+")");
                }
            }
            if(c>=d&&b>c/d){
                if(a+(b/(c/d))==24){
                    allResult.add(""+a+"+("+b+"/("+c+"/"+d+"))");
                }
            }
            if(((a+b)-c)*d==24){
                allResult.add(("+a+"+"+b+")-"+c+")*"+d);
            }
            if((a+(b-c))*d==24){
                allResult.add("("+a+"+("+b+"))-"+c+"*"+d+"");
            }
            if((a+b)-(c*d)==24){
                allResult.add(("+a+"+"+b+")-("+c+"*"+d+")");
            }
```

```
            if(a+(b-(c*d))==24){
                allResult.add(""+a+"+("+b+"-("+c+"*"+d+"))");
            }
            if(((a+b)-c)/d==24){
                allResult.add("(("+a+"+"+b+")-"+c+")/"+d);
            }
            if((a+(b-c))/d==24){
                allResult.add("("+a+"+("+b+"))-"+c+"/"+d+"");
            }
            if((a+b)-(c/d)==24){
                allResult.add("(("+a+"+"+b+")-("+c+"/"+d+")");
            }
            if(a+(b-(c/d))==24){
                allResult.add(""+a+"+("+b+"-("+c+"/"+d+"))");
            }
            if(((a-b)-c)*d==24){
                allResult.add("(("+a+"-"+b+")-"+c+")*"+d);
            }
            if((a-(b-c))*d==24){
                allResult.add("("+a+"-("+b+"))-"+c+"*"+d+"");
            }
            if((a-b)-(c*d)==24){
                allResult.add("(("+a+"-"+b+")-("+c+"*"+d+")");
            }
            if(a-(b-(c*d))==24){
                allResult.add(""+a+"-("+b+"-("+c+"*"+d+"))");
            }
            if(((a-b)*c)+d==24){
                allResult.add("(("+a+"-"+b+")*"+c+")+"+d);
            }
            if((a-(b*c))+d==24){
                allResult.add("("+a+"-("+b+"))*"+c+"+"+d+"");
            }
            if((a-b)*(c+d)==24){
                allResult.add("(("+a+"-"+b+")*("+c+"+"+d+")");
            }
            if(a-(b*(c+d))==24){
                allResult.add(""+a+"-("+b+"*("+c+"+"+d+"))");
            }
            if(((a-b)*c)+d==24){
                allResult.add("(("+a+"-"+b+")*"+c+")+"+d);
            }
            if((a-(b*c))+d==24){
                allResult.add("("+a+"-("+b+"))*"+c+"+"+d+"");
            }
            if((a-b)*(c+d)==24){
                allResult.add("(("+a+"-"+b+")*("+c+"+"+d+")");
            }
```

```
            if(a-(b*(c+d))==24){
                allResult.add(""+a+"-("+b+"*("+c+"+"+d+"))");
            }
            if(((a-b)*c)-d==24){
                allResult.add("(("+a+"-"+b+")*"+c+")-"+d);
            }
            if((a-(b*c))-d==24){
                allResult.add("("+a+"-("+b+"))*"+c+"-"+d+"");
            }
            if((a-b)*(c-d)==24){
                allResult.add("(("+a+"-"+b+")*("+c+"-"+d+")");
            }
            if(a-(b*(c-d))==24){
                allResult.add(""+a+"-("+b+"*("+c+"-"+d+"))");
            }
            if(((a-b)*c)*d==24){
                allResult.add("(("+a+"-"+b+")*"+c+")*"+d);
            }
            if((a-(b*c))*d==24){
                allResult.add("("+a+"-("+b+"))*"+c+"*"+d+"");
            }
            if((a-b)*(c*d)==24){
                allResult.add("(("+a+"-"+b+")*("+c+"*"+d+")");
            }
            if(a-(b*(c*d))==24){
                allResult.add(""+a+"-("+b+"*("+c+"*"+d+"))");
            }
            if(((a-b)*c)/d==24){
                allResult.add("(("+a+"-"+b+")*"+c+")/"+d);
            }
            if((a-(b*c))/d==24){
                allResult.add("("+a+"-("+b+"))*"+c+"/"+d+"");
            }
            if((a-b)*(c/d)==24){
                allResult.add("(("+a+"-"+b+")*("+c+"/"+d+")");
            }
            if(a-(b*(c/d))==24){
                allResult.add(""+a+"-("+b+"*("+c+"/"+d+"))");
            }
            if(((a-b)/c)+d==24){
                allResult.add("(("+a+"-"+b+")/"+c+")+"+d);
            }
            if((a-(b/c))+d==24){
                allResult.add("("+a+"-("+b+"))/"+c+"+"+d+"");
            }
            if((a-b)/(c+d)==24){
                allResult.add("(("+a+"-"+b+")/("+c+"+"+d+")");
            }
```

```
            if(a-(b/(c+d))==24){
                allResult.add(""+a+"-("+b+"/("+c+"+"+d+"))");
            }
            if(((a-b)/c)-d==24){
                allResult.add("(("+a+"-"+b+")/"+c+")-"+d);
            }
            if((a-(b/c))-d==24){
                allResult.add("("+a+"-("+b+"))/"+c+"-"+d+"");
            }
            if(c-d!=0){
                if((a-b)/(c-d)==24){
                    allResult.add("(("+a+"-"+b+")/("+c+"-"+d+")");
                }
                if(a-(b/(c-d))==24){
                    allResult.add(""+a+"-("+b+"/("+c+"-"+d+"))");
                }
            }
            if(((a-b)/c)*d==24){
                allResult.add("(("+a+"-"+b+")/"+c+")*"+d);
            }
            if((a-(b/c))*d==24){
                allResult.add("("+a+"-("+b+"))/"+c+"*"+d+"");
            }
            if((a-b)/(c*d)==24){
                allResult.add("(("+a+"-"+b+")/("+c+"*"+d+")");
            }
            if(a-(b/(c*d))==24){
                allResult.add(""+a+"-("+b+"/("+c+"*"+d+"))");
            }
            if(((a*b)+c)+d==24){
                allResult.add("(("+a+"*"+b+")+"+c+")+"+d);
            }
            if((a*(b+c))+d==24){
                allResult.add("("+a+"*("+b+"))+"+c+"+"+d+"");
            }
            if((a*b)+(c+d)==24){
                allResult.add("(("+a+"*"+b+")+("+c+"+"+d+")");
            }
            if(a*(b+(c+d))==24){
                allResult.add(""+a+"*("+b+"+("+c+"+"+d+"))");
            }
            if(((a*b)+c)-d==24){
                allResult.add("(("+a+"*"+b+")+"+c+")-"+d);
            }
            if((a*(b+c))-d==24){
                allResult.add("("+a+"*("+b+"))+"+c+"-"+d+"");
            }
            if((a*b)+(c-d)==24){
```

```
            allResult.add("(("+a+"*"+b+")+("+c+"-"+d+")");
        }
        if(a*(b+(c-d))==24){
            allResult.add(""+a+"*("+b+"+("+c+"-"+d+"))");
        }
        if(((a*b)+c)*d==24){
            allResult.add("(("+a+"*"+b+")+"+c+")*"+d);
        }
        if((a*(b+c))*d==24){
            allResult.add("("+a+"*("+b+"))+"+c+"*"+d+"");
        }
        if((a*b)+(c*d)==24){
            allResult.add("(("+a+"*"+b+")+("+c+"*"+d+")");
        }
        if(a*(b+(c*d))==24){
            allResult.add(""+a+"*("+b+"+("+c+"*"+d+"))");
        }
        if(((a*b)+c)/d==24){
            allResult.add("(("+a+"*"+b+")+"+c+")/"+d);
        }
        if((a*(b+c))/d==24){
            allResult.add("("+a+"*("+b+"))+"+c+"/"+d+"");
        }
        if((a*b)+(c/d)==24){
            allResult.add("(("+a+"*"+b+")+("+c+"/"+d+")");
        }
        if(a*(b+(c/d))==24){
            allResult.add(""+a+"*("+b+"+("+c+"/"+d+"))");
        }
        if(((a*b)-c)+d==24){
            allResult.add("(("+a+"*"+b+")-"+c+")+"+d);
        }
        if((a*(b-c))+d==24){
            allResult.add("("+a+"*("+b+"))-"+c+"+"+d+"");
        }
        if((a*b)-(c+d)==24){
            allResult.add("(("+a+"*"+b+")-("+c+"+"+d+")");
        }
        if(a*(b-(c+d))==24){
            allResult.add(""+a+"*("+b+"-("+c+"+"+d+"))");
        }
        if(((a*b)-c)-d==24){
            allResult.add("(("+a+"*"+b+")-"+c+")-"+d);
        }
        if((a*(b-c))-d==24){
            allResult.add("("+a+"*("+b+"))-"+c+"-"+d+"");
        }
        if((a*b)-(c-d)==24){
```

```
            allResult.add("(("+a+"*"+b+")-("+c+"-"+d+")");
        }
        if(a*(b-(c-d))==24){
            allResult.add(""+a+"*("+b+"-("+c+"-"+d+"))");
        }
        if(((a*b)-c)*d==24){
            allResult.add("(("+a+"*"+b+")-"+c+")*"+d);
        }
        if((a*(b-c))*d==24){
            allResult.add("("+a+"*("+b+"))-"+c+"*"+d+"");
        }
        if((a*b)-(c*d)==24){
            allResult.add("(("+a+"*"+b+")-("+c+"*"+d+")");
        }
        if(a*(b-(c*d))==24){
            allResult.add(""+a+"*("+b+"-("+c+"*"+d+"))");
        }
        if(((a*b)-c)/d==24){
            allResult.add("(("+a+"*"+b+")-"+c+")/"+d);
        }
        if((a*(b-c))/d==24){
            allResult.add("("+a+"*("+b+"))-"+c+"/"+d+"");
        }
        if((a*b)-(c/d)==24){
            allResult.add("(("+a+"*"+b+")-("+c+"/"+d+")");
        }
        if(a*(b-(c/d))==24){
            allResult.add(""+a+"*("+b+"-("+c+"/"+d+"))");
        }
        if(((a*b)*c)+d==24){
            allResult.add("(("+a+"*"+b+")*"+c+")+"+d);
        }
        if((a*(b*c))+d==24){
            allResult.add("("+a+"*("+b+"))*"+c+"+"+d+"");
        }
        if((a*b)*(c+d)==24){
            allResult.add("(("+a+"*"+b+")*("+c+"+"+d+")");
        }
        if(a*(b*(c+d))==24){
            allResult.add(""+a+"*("+b+"*("+c+"+"+d+"))");
        }
        if(((a*b)*c)-d==24){
            allResult.add("(("+a+"*"+b+")*"+c+")-"+d);
        }
        if((a*(b*c))-d==24){
            allResult.add("("+a+"*("+b+"))*"+c+"-"+d+"");
        }
        if((a*b)*(c-d)==24){
```

```
                allResult.add("(("+a+"*"+b+")*("+c+"-"+d+")");
            }
            if(a*(b*(c-d))==24){
                allResult.add(""+a+"*("+b+"*("+c+"-"+d+"))");
            }
            if(((a*b)*c)*d==24){
                allResult.add("(("+a+"*"+b+")*"+c+")*"+d);
            }
            if((a*(b*c))*d==24){
                allResult.add("("+a+"*("+b+"))*"+c+"*"+d+"");
            }
            if((a*b)*(c*d)==24){
                allResult.add("(("+a+"*"+b+")*("+c+"*"+d+")");
            }
            if(a*(b*(c*d))==24){
                allResult.add(""+a+"*("+b+"*("+c+"*"+d+"))");
            }
            if(((a*b)*c)/d==24){
                allResult.add("(("+a+"*"+b+")*"+c+")/"+d);
            }
            if((a*(b*c))/d==24){
                allResult.add("("+a+"*("+b+"))*"+c+"/"+d+"");
            }
            if((a*b)*(c/d)==24){
                allResult.add("(("+a+"*"+b+")*("+c+"/"+d+")");
            }
            if(a*(b*(c/d))==24){
                allResult.add(""+a+"*("+b+"*("+c+"/"+d+"))");
            }
            if(((a*b)/c)+d==24){
                allResult.add("(("+a+"*"+b+")/"+c+")+"+d);
            }
            if((a*(b/c))+d==24){
                allResult.add("("+a+"*("+b+"))/"+c+"+"+d+"");
            }
            if((a*b)/(c+d)==24){
                allResult.add("(("+a+"*"+b+")/("+c+"+"+d+")");
            }
            if(a*(b/(c+d))==24){
                allResult.add(""+a+"*("+b+"/("+c+"+"+d+"))");
            }
            if(((a*b)/c)-d==24){
                allResult.add("(("+a+"*"+b+")/"+c+")-"+d);
            }
            if((a*(b/c))-d==24){
                allResult.add("("+a+"*("+b+"))/"+c+"-"+d+"");
            }
            if(c-d!=0){
```

```
            if((a*b)/(c-d)==24){
                allResult.add("(("+a+"*"+b+")/("+c+"-"+d+")");
            }

            if(a*(b/(c-d))==24){
                allResult.add(""+a+"*("+b+"/("+c+"-"+d+"))");
            }
        }
        if(((a*b)/c)*d==24){
            allResult.add("(("+a+"*"+b+")/"+c+")*"+d);
        }
        if((a*(b/c))*d==24){
            allResult.add("("+a+"*("+b+"))/"+c+"*"+d+"");
        }
        if((a*b)/(c*d)==24){
            allResult.add("(("+a+"*"+b+")/("+c+"*"+d+")");
        }
        if(a*(b/(c*d))==24){
            allResult.add(""+a+"*("+b+"/("+c+"*"+d+"))");
        }
        if(((a*b)/c)/d==24){
            allResult.add("(("+a+"*"+b+")/"+c+")/"+d);
        }
        if((a*(b/c))/d==24){
            allResult.add("("+a+"*("+b+"))/"+c+"/"+d+"");
        }
        if(c>=d&&a*b>=c/d){
            if((a*b)/(c/d)==24){
                allResult.add("(("+a+"*"+b+")/("+c+"/"+d+")");
            }
        }
        if(c>=d&&b>=c/d){
            if(a*(b/(c/d))==24){
                allResult.add(""+a+"*("+b+"/("+c+"/"+d+"))");
            }
        }
        if(((a/b)/c)+d==24){
            allResult.add("(("+a+"/"+b+")/"+c+")+"+d);
        }
        if((b/c)>0){
            if((a/(b/c))+d==24){
                allResult.add("("+a+"/("+b+"))/"+c+"+"+d+"");
            }
        }
        if((a/b)/(c+d)>0){
            if((a/b)/(c+d)==24){
                allResult.add("(("+a+"/"+b+")/("+c+"+"+d+")");
            }
```

```
        }
        if(a>=b&&b>=c+d){
            if(a/(b/(c+d))==24){
                allResult.add(""+a+"/("+b+"/("+c+"+"+d+"))");
            }
        }
        if(((a/b)+c)+d==24){
            allResult.add(("+a+"/"+b+")+"+c+")+"+d);
        }
        if((a/(b+c))+d==24){
            allResult.add("("+a+"/("+b+"))+"+c+"+"+d+"");
        }
        if((a/b)+(c+d)==24){
            allResult.add(("+a+"/"+b+")+("+c+"+"+d+")");
        }
        if(a/(b+(c+d))==24){
            allResult.add(""+a+"/("+b+"+("+c+"+"+d+"))");
        }
        if(((a/b)+c)-d==24){
            allResult.add(("+a+"/"+b+")+"+c+")-"+d);
        }
        if((a/(b+c))-d==24){
            allResult.add("("+a+"/("+b+"))+"+c+"-"+d+"");
        }
        if((a/b)+(c-d)==24){
            allResult.add(("+a+"/"+b+")+("+c+"-"+d+")");
        }
        if(a>=b+(c-d)&&b+(c-d)>0){
            if(a/(b+(c-d))==24){
                allResult.add(""+a+"/("+b+"+("+c+"-"+d+"))");
            }
        }
        if(((a/b)+c)*d==24){
            allResult.add(("+a+"/"+b+")+"+c+")*"+d);
        }
        if((a/(b+c))*d==24){
            allResult.add("("+a+"/("+b+"))+"+c+"*"+d+"");
        }
        if((a/b)+(c*d)==24){
            allResult.add(("+a+"/"+b+")+("+c+"*"+d+")");
        }
        if(a/(b+(c*d))==24){
            allResult.add(""+a+"/("+b+"+("+c+"*"+d+"))");
        }
        if(((a/b)+c)/d==24){
            allResult.add(("+a+"/"+b+")+"+c+")/"+d);
        }
        if((a/(b+c))/d==24){
```

```
            allResult.add("("+a+"/("+b+"))+"+c+"/"+d+"");
        }
        if((a/b)+(c/d)==24){
            allResult.add(("+a+"/"+b+")+("+c+"/"+d+")");
        }
        if(a/(b+(c/d))==24){
            allResult.add(""+a+"/("+b+"+("+c+"/"+d+"))");
        }
        if(((a/b)-c)+d==24){
            allResult.add(("+a+"/"+b+")-"+c+")+"+d);
        }
        if(b-c!=0){
            if((a/(b-c))+d==24){
                allResult.add("("+a+"/("+b+"))-"+c+"+"+d+"");
            }
        }
        if((a/b)-(c+d)==24){
            allResult.add(("+a+"/"+b+")-("+c+"+"+d+")");
        }
        if(b-(c+d)!=0){
            if(a/(b-(c+d))==24){
                allResult.add(""+a+"/("+b+"-("+c+"+"+d+"))");
            }
        }
        if(((a/b)-c)*d==24){
            allResult.add(("+a+"/"+b+")-"+c+")*"+d);
        }
        if(b-c!=0){
            if((a/(b-c))*d==24){
                allResult.add("("+a+"/("+b+"))-"+c+"*"+d+"");
            }
        }
        if((a/b)-(c*d)==24){
            allResult.add(("+a+"/"+b+")-("+c+"*"+d+")");
        }
        if(b-(c*d)!=0){
            if(a/(b-(c*d))==24){
                allResult.add(""+a+"/("+b+"-("+c+"*"+d+"))");
            }
        }
        if(((a/b)*c)+d==24){
            allResult.add(("+a+"/"+b+")*"+c+")+"+d);
        }
        if((a/(b*c))+d==24){
            allResult.add("("+a+"/("+b+"))*"+c+"+"+d+"");
        }
        if((a/b)*(c+d)==24){
            allResult.add(("+a+"/"+b+")*("+c+"+"+d+")");
```

```
            }
            if(a/(b*(c+d))==24){
                allResult.add(""+a+"/("+b+"*("+c+"+"+d+"))");
            }
            if(((a/b)*c)-d==24){
                allResult.add(("+a+"/"+b+")*"+c+")-"+d);
            }
            if((a/(b*c))-d==24){
                allResult.add("("+a+"/("+b+"))*"+c+"-"+d+"");
            }
            if(c-d!=0){
                if((a/b)*(c-d)==24){
                    allResult.add(("+a+"/"+b+")*("+c+"-"+d+")");
                }
                if(a/(b*(c-d))==24){
                    allResult.add(""+a+"/("+b+"*("+c+"-"+d+"))");
                }
            }
            if(((a/b)*c)*d==24){
                allResult.add(("+a+"/"+b+")*"+c+")*"+d);
            }
            if((a/(b*c))*d==24){
                allResult.add("("+a+"/("+b+"))*"+c+"*"+d+"");
            }
            if((a/b)*(c*d)==24){
                allResult.add(("+a+"/"+b+")*("+c+"*"+d+")");
            }
            if(a/(b*(c*d))==24){
                allResult.add(""+a+"/("+b+"*("+c+"*"+d+"))");
            }
            if(((a/b)*c)/d==24){
                allResult.add(("+a+"/"+b+")*"+c+")/"+d);
            }
            if((a/(b*c))/d==24){
                allResult.add("("+a+"/("+b+"))*"+c+"/"+d+"");
            }
            if((a/b)*(c/d)==24){
                allResult.add(("+a+"/"+b+")*("+c+"/"+d+")");
            }
            if(c>d&&a>b*(c/d)){
                if(a/(b*(c/d))==24){
                    allResult.add(""+a+"/("+b+"*("+c+"/"+d+"))");
                }
            }
            if(((a/b)/c)*d==24){
                allResult.add(("+a+"/"+b+")/"+c+")*"+d);
            }
            if(a>b&&b>c){
```

```
                if((a/(b/c))*d==24){
                    allResult.add("("+a+"/("+b+"))/"+c+"*"+d+"");
                }
            }
            if(a>b&&(a/b>c*d)){
                if((a/b)/(c*d)==24){
                    allResult.add(("+a+"/"+b+")/("+c+"*"+d+")");
                }
            }
            if(a>b&&b>c*d){
                if(a/(b/(c*d))==24){
                    allResult.add(""+a+"/("+b+"/("+c+"*"+d+"))");
                }
            }
        }
        return allResult;
    }
}
```

2.2. File_Writer.java

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.List;

public class File_Writer extends Input {
    public static boolean isWriteResult(){
        int pilihan;
        boolean condition;
        while (true) {
            System.out.println("Apakah hasil mau disimpan ke dalam file?");
            System.out.println("1. YA");
            System.out.println("2. TIDAK");
            System.out.print(">> ");
            pilihan = scan.nextInt();

            if (pilihan == 1) {
                condition = true;
                break;

            } else if (pilihan == 2) {
                condition = false;
                break;

            } else {
                System.out.println("Input tidak valid!");
            }
```

```
        }
        return condition;
    }

    public static void writeResult(String[] input, List<String> result) {
        try {
            System.out.print("Input nama file: ");
            String fileName = scan.next();
            String address = "../test/"+fileName+".txt";
            FileWriter writer = new FileWriter(address);
            writer.write("Kartu Anda: \n");
            writer.append(input[0]);
            for (int i = 1; i < input.length; i++) {
                writer.append(" "+input[i]);
            }
            writer.append("\nDiperoleh "+result.size()+" solusi:\n");
            for (int j = 0; j < result.size(); j++) {
                if(j!=result.size()-1){
                    writer.append(result.get(j)+"\n");
                } else if (j==result.size()-1){
                    writer.append(result.get(j));
                }
            }

            writer.close();
        } catch (IOException e) {
            System.out.println("Error Found");
        }
    }
}
```

2.3. Function.java

```
import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class Function {
    public static int converterInput (String inputManual) {
        int convert=0;
        if (inputManual.equals("A")){
            convert=1;
        } else if (inputManual.equals("2")){
            convert=2;
        } else if (inputManual.equals("3")){
            convert=3;
        } else if (inputManual.equals("4")){
            convert=4;
```

```
        } else if (inputManual.equals("5")){
            convert=5;
        } else if (inputManual.equals("6")){
            convert=6;
        } else if (inputManual.equals("7")){
            convert=7;
        } else if (inputManual.equals("8")){
            convert=8;
        } else if (inputManual.equals("9")){
            convert=9;
        } else if (inputManual.equals("10")){
            convert=10;
        } else if (inputManual.equals("J")){
            convert=11;
        } else if (inputManual.equals("Q")){
            convert=12;
        } else if (inputManual.equals("K")){
            convert=13;
        }
        return convert;
    }

    public static String converterOutput (Integer autoInput) {
        String convert="";
        if (autoInput==1){
            convert="A";
        } else if (autoInput==2){
            convert="2";
        } else if (autoInput==3){
            convert="3";
        } else if (autoInput==4){
            convert="4";
        } else if (autoInput==5){
            convert="5";
        } else if (autoInput==6){
            convert="6";
        } else if (autoInput==7){
            convert="7";
        } else if (autoInput==8){
            convert="8";
        } else if (autoInput==9){
            convert="9";
        } else if (autoInput==10){
            convert="10";
        } else if (autoInput==11){
            convert="J";
        } else if (autoInput==12){
            convert="Q";
```

```
        } else if (autoInput==13){
            convert="K";
        }
        return convert;
    }

    public static void printStringArray(String[] array){
        for (int i = 0; i < 4; i++) {
            if(i!=3){
                System.out.printf(array[i]+" ");
            } else {
                System.out.println(array[i]);
            }
        }
    }

    public static int getRandomElement(List<Integer> list)
    {
        Random rand = new Random();
        return list.get(rand.nextInt(list.size()));
    }

    public static int randomCard(){
        //List
        List<Integer> cardDeck = new ArrayList<>(13);
        cardDeck.add(1);
        cardDeck.add(2);
        cardDeck.add(3);
        cardDeck.add(4);
        cardDeck.add(5);
        cardDeck.add(6);
        cardDeck.add(7);
        cardDeck.add(9);
        cardDeck.add(9);
        cardDeck.add(10);
        cardDeck.add(11);
        cardDeck.add(12);
        cardDeck.add(13);
        //Array
        int selectedCard;

        //Algorithm
        selectedCard=getRandomElement(cardDeck);

        return selectedCard;
    }


}
```

2.4. Input.java

```
import java.util.Scanner;

public class Input extends Menu{
    static Scanner scan = new Scanner(System.in);

    public static int choice(){
        int pilihan = scan.nextInt();
        return pilihan;
    }

    public static String choiceString(){
        String pilihan = scan.next();
        return pilihan;
    }

    public static String manualInput(){
        String inputManual;
        inputManual=scan.next();
        return inputManual;
    }
}
```

2.5. Main.java

```
import java.util.ArrayList;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        //variables
        long startTime;
        long endTime;
        int opsiProgram;
        List<Integer>cardValue = new ArrayList<>(4);
        String[] userCard = new String[4];
        List<String>allResult = new ArrayList<>();
        boolean condition=false;

        //Deck of Cards
        List<String> deckCard = new ArrayList<>();
        deckCard.add("A");
        deckCard.add("2");
        deckCard.add("3");
        deckCard.add("4");
        deckCard.add("5");
        deckCard.add("6");
        deckCard.add("7");
```

```
        deckCard.add("8");
        deckCard.add("9");
        deckCard.add("10");
        deckCard.add("J");
        deckCard.add("Q");
        deckCard.add("K");

        //Algoritma
        Menu.splash();
        Menu.menu();
        opsiProgram=Input.choice();
        while(true){
            if(opsiProgram==1){
                do{
                    condition=false;
                    cardValue.removeAll(cardValue);
                    for (int i = 0; i < userCard.length; i++) {
                        userCard[i]=Input.manualInput();
                        if(deckCard.indexOf(userCard[i])==-1){
                            condition=true;
                        }
                        cardValue.add(i,
Function.converterInput(userCard[i]));
                    }
                    if(condition==true){
                        Menu.errorMessage();
                    }
                }while(condition==true);
                System.out.println("Kartu Anda: ");
                Function.printStringArray(userCard);
                break;
            } else if (opsiProgram==2){
                for (int i = 0; i < userCard.length; i++) {
                    cardValue.add(Function.randomCard());
                }
                for (int i = 0; i < cardValue.size(); i++) {
                    userCard[i]=Function.converterOutput(cardValue.get(i));
                }
                System.out.println("Kartu Anda: ");
                Function.printStringArray(userCard);
                break;
            } else {
                Menu.errorMessage();
                Menu.menu();
                opsiProgram=Input.choice();
            }
        }
        startTime = System.currentTimeMillis();
```

```
        allResult.addAll(BruteForce.getAllResult(cardValue));
        if(allResult.size()==0){
            System.out.println("TIdak ada solusi");
        }else{
            System.out.println("Diperoleh "+allResult.size()+" solusi:");
            for (int i = 0; i < allResult.size(); i++) {
                System.out.println(allResult.get(i));
            }
        }
        endTime = System.currentTimeMillis();
        System.out.println("Time execution: "+(endTime-startTime)+" ms");
        if(allResult.size()!=0){
            if(File_Writer.isWriteResult()==true){
                File_Writer.writeResult(userCard, allResult);
            }
        }
    }
}
```

2.6. Menu.java

```
public class Menu {
    public static void splash(){
        System.out.println("  ####       ###              ####     ##     ####
##   #####");
        System.out.println("
##  ##     ####             ##  ##   ####     ##  ##   ## ##");
        System.out.println("     ##    ##
##           ##       ##  ##    ##  ##   ##  ##");
        System.out.println("   ###    ##  ##            ##       ##  ##    ###
##    ##  ##");
        System.out.println("  ##      #######           ##       ######    ##
##    ##  ##");
        System.out.println("
##  ##       ##             ##  ##  ##  ##    ##  ##   ## ##");
        System.out.println(" ######       ##              ####   ##  ##   ####
##  #####");
        System.out.println("                  ####     ##     ##   ##  #######
");
        System.out.println("                 ##  ##   ####    ###
###   ##   ");
        System.out.println("                ##       ##  ##   #######   ## ");
        System.out.println("                ##       ##  ##   #######   #####"
);
        System.out.println("                ##  ###  ######   ## # ##   ## ");
        System.out.println("                 ##  ##  ##  ##   ##   ##   ##   "
);
```

```
        System.out.println("                  #####  ##  ##   ##   ##  #######
");
        System.out.println("Selamat datang di 24 Card Game!!");
    }

    public static void menu(){
        System.out.println("1. Pilih kartu secara manual");
        System.out.println("2. Dipilih otomatis oleh Game");
    }
    public static void errorMessage(){
        System.out.println("Input salah!!!");
        System.out.println("Masukan input yang benar!!");
    }
}
```

BAB III

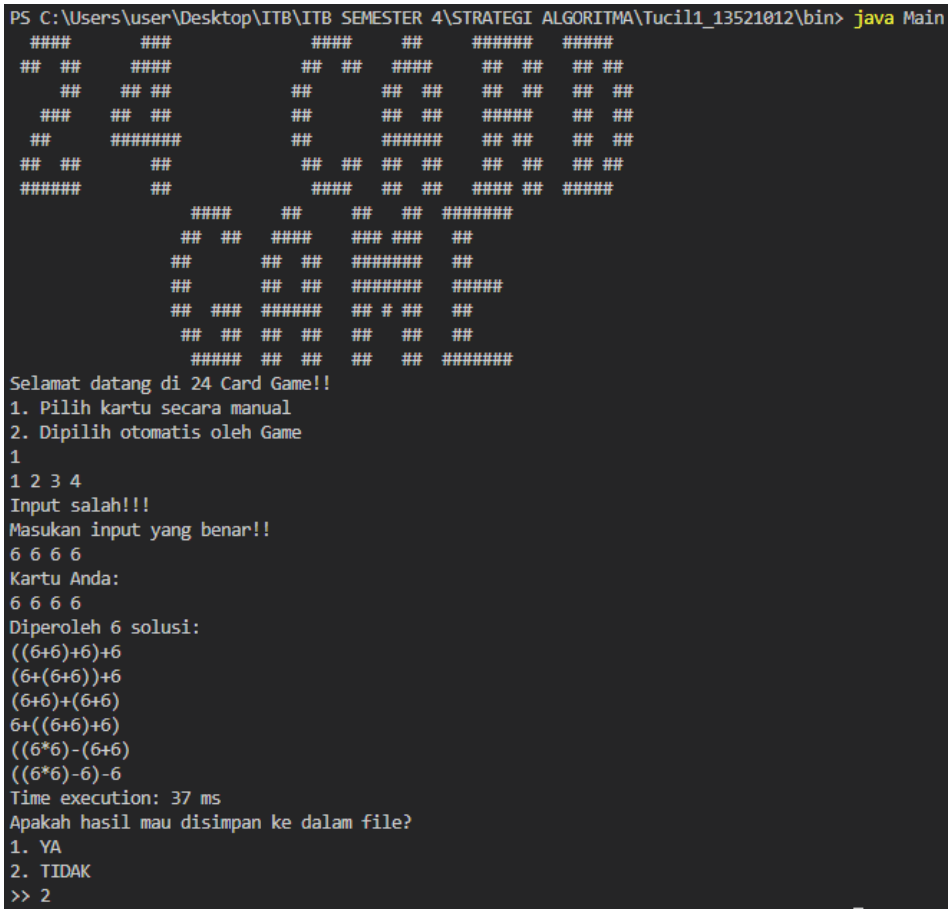
CONTOH INPUT DAN OUTPUT PROGRAM SERTA LINK REPOSITORY

3.1. Contoh input dan output

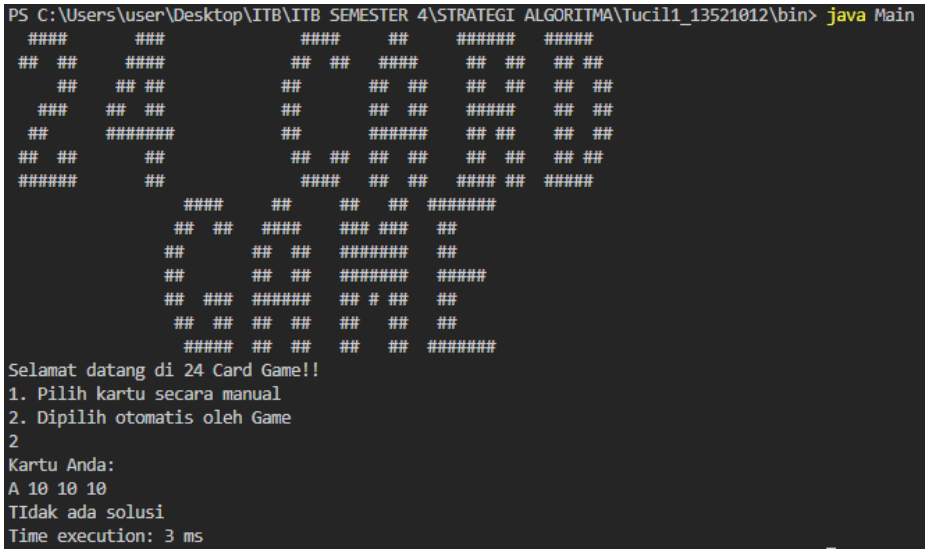
3.1.1. Contoh input dan output 1.

```
PS C:\Users\user\Desktop\ITB\ITB SEMESTER 4\STRATEGI ALGORITMA\Tucil1_13521012\bin> java Main
  ####       ###              ####     ##     ######   #####
 ##  ##     ####            ##  ##   ####     ##  ##   ## ##
    ##    ## ##            ##       ##  ##    ##  ##   ##  ##
   ###    ##  ##           ##       ##  ##    #####    ##  ##
  ##      #######          ##       ######    ## ##    ##  ##
 ##  ##       ##            ##  ##  ##  ##    ##  ##   ## ##
 ######       ##              ####   ##  ##   #### ##  #####
                ####     ##     ##   ##  #######
               ##  ##   ####    ### ###   ##
              ##       ##  ##   #######   ##
              ##       ##  ##   #######   #####
              ##  ###  ######   ## # ##   ##
               ##  ##  ##  ##   ##   ##   ##
                #####  ##  ##   ##   ##  #######
Selamat datang di 24 Card Game!!
1. Pilih kartu secara manual
2. Dipilih otomatis oleh Game
2
Kartu Anda:
4 9 2 A
Diperoleh 26 solusi:
4*(9-(2+1))
((4+9)-1)*2
(4+(9))-1*2
((4+9)-1)*2
(4+(9))-1*2
4*(9-(1+2))
4+(2*(9+1))
4+(2*(1+9))
(4-(1))-9*2
((9+4)-1)*2
(9+(4))-1*2
((9+4)-1)*2
(9+(4))-1*2
((9-2)-1)*4
((9+1)*2)+4
((9-1)-2)*4
(9-(1))-4*2
2*(9+(4-1))
(2*(9))+1+4
(2*(9))+1+4
2*(9-(1-4))
2*(4+(9-1))
2*(4-(1-9))
(2*(1))+9+4
(2*(1))+9+4
((1+9)*2)+4
Time execution: 32 ms
Apakah hasil mau disimpan ke dalam file?
1. YA
2. TIDAK
>> 1
Input nama file: test3
```

3.1.2. Contoh input dan output 2.



3.1.3. Contoh input dan output 3.



3.1.4. Contoh input dan output 4.

```
PS C:\Users\user\Desktop\ITB\ITB SEMESTER 4\STRATEGI ALGORITMA\Tucil1_13521012\bin> java Main
  ####       ###             ####     ##     ######   #####
 ##  ##     ####            ##  ##   ####     ##  ##   ## ##
    ##    ## ##           ##       ##  ##    ##  ##   ##  ##
   ###    ##  ##           ##       ##  ##    #####    ##  ##
  ##      #######          ##       ######    ## ##    ##  ##
 ##  ##       ##            ##  ##  ##  ##    ##  ##   ## ##
 ######       ##              ####   ##  ##   #### ##  #####
                ####     ##     ##   ##  #######
               ##  ##   ####    ### ###   ##
              ##       ##  ##   #######   ##
              ##       ##  ##   #######   #####
              ##  ###  ######   ## # ##   ##
               ##  ##  ##  ##   ##   ##   ##
                #####  ##  ##   ##   ##  #######
Selamat datang di 24 Card Game!!
1. Pilih kartu secara manual
2. Dipilih otomatis oleh Game
2
Kartu Anda:
9 7 Q 5
Diperoleh 10 solusi:
((9*7)/5)+12
((9+5)*12)/7
((9+5)/7)*12
((7*9)/5)+12
(12*(9))+5/7
(12*(5))+9/7
12*(5/(9-7))
((5+9)*12)/7
((5+9)/7)*12
(5/(9))-7*12
Time execution: 33 ms
Apakah hasil mau disimpan ke dalam file?
1. YA
2. TIDAK
>> 2
```

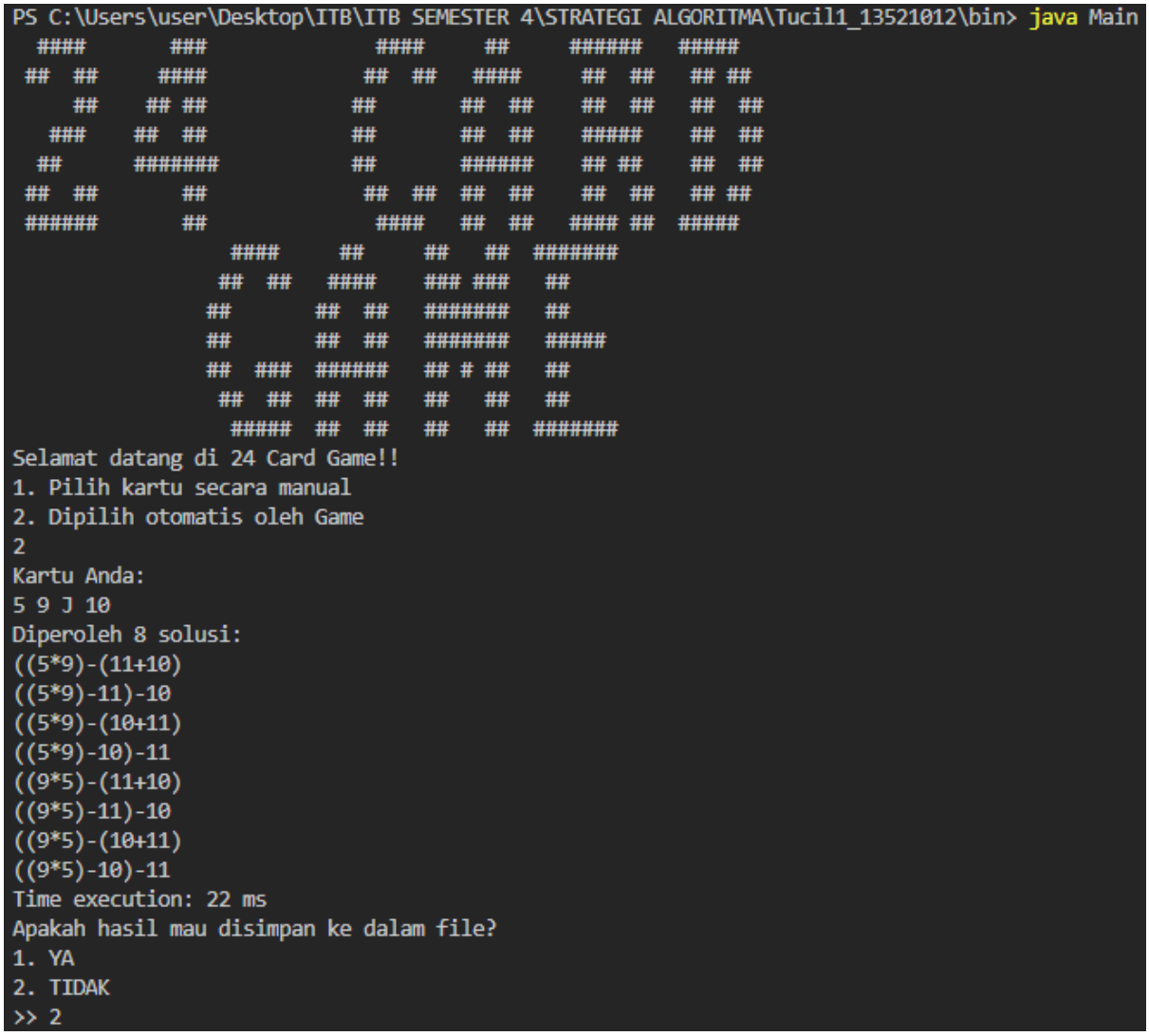
3.1.5. Contoh input dan output 5.

((1*4)*2)*3
(1*(4))*2*3
((1*4)*(2*3)
1*(4*(2*3))
((2+1)+3)*4
(2+(1))+3*4
((2*1)*3)*4
(2*(1))*3*4
((2*1)*(3*4)
2*(1*(3*4))
((2/1)*3)*4
((2/1)*(3*4)
((2*1)*4)*3
(2*(1))*4*3
((2*1)*(4*3)
2*(1*(4*3))
((2/1)*4)*3
((2/1)*(4*3)
((2+3)+1)*4
(2+(3))+1*4
((2*3)*1)*4
(2*(3))*1*4
((2*3)*(1*4)
2*(3*(1*4))
((2*3)/1)*4
(2*(3))/1*4
((2*3)*4)*1
(2*(3))*4*1
((2*3)*(4*1)
2*(3*(4*1))
((2*3)*4)/1
(2*(3))*4/1
((2*3)*(4/1)
2*(3*(4/1))
((2+4)*(3+1)
((2*4)*3)*1
(2*(4))*3*1
((2*4)*(3*1)
2*(4*(3*1))
((2*4)*3)/1
(2*(4))*3/1
((2*4)*(3/1)
2*(4*(3/1))
((2+4)*(1+3)
((2*4)*1)*3
(2*(4))*1*3
((2*4)*(1*3)
2*(4*(1*3))
((2*4)/1)*3
(2*(4))/1*3
((3+2)+1)*4
(3+(2))+1*4
((3*2)*1)*4
(3*(2))*1*4
((3*2)*(1*4)

3*(2*(1*4))
((3*2)/1)*4
(3*(2))/1*4
((3*2)*4)*1
(3*(2))*4*1
((3*2)*(4*1)
3*(2*(4*1))
((3*2)*4)/1
(3*(2))*4/1
((3*2)*(4/1)
3*(2*(4/1))
((3+1)+2)*4
(3+(1))+2*4
((3+1)*(2+4)
((3*1)*2)*4
(3*(1))*2*4
((3*1)*(2*4)
3*(1*(2*4))
((3/1)*2)*4
((3/1)*(2*4)
((3+1)*(4+2)
((3*1)*4)*2
(3*(1))*4*2
((3*1)*(4*2)
3*(1*(4*2))
((3/1)*4)*2
((3/1)*(4*2)
((3*4)*1)*2
(3*(4))*1*2
((3*4)*(1*2)
3*(4*(1*2))
((3*4)/1)*2
(3*(4))/1*2
((3*4)*2)*1
(3*(4))*2*1
((3*4)*(2*1)
3*(4*(2*1))
((3*4)*2)/1
(3*(4))*2/1
((3*4)*(2/1)
3*(4*(2/1))
4*(2+(3+1))
((4+2)*(3+1)
4*(2+(3+1))
((4*2)*3)*1
(4*(2))*3*1
((4*2)*(3*1)
4*(2*(3*1))
((4*2)*3)/1
(4*(2))*3/1
((4*2)*(3/1)
4*(2*(3/1))
4*(2+(1+3))
((4+2)*(1+3)
4*(2+(1+3))

```
((4*2)*1)*3
(4*(2))*1*3
((4*2)*(1*3)
4*(2*(1*3))
((4*2)/1)*3
(4*(2))/1*3
4*(3+(2+1))
4*(3+(2+1))
((4*3)*2)*1
(4*(3))*2*1
((4*3)*(2*1)
4*(3*(2*1))
((4*3)*2)/1
(4*(3))*2/1
((4*3)*(2/1)
4*(3*(2/1))
4*(3+(1+2))
4*(3+(1+2))
((4*3)*1)*2
(4*(3))*1*2
((4*3)*(1*2)
4*(3*(1*2))
((4*3)/1)*2
(4*(3))/1*2
4*(1+(3+2))
4*(1+(3+2))
((4*1)*3)*2
(4*(1))*3*2
((4*1)*(3*2)
4*(1*(3*2))
((4/1)*3)*2
((4/1)*(3*2)
4*(1+(2+3))
4*(1+(2+3))
((4*1)*2)*3
(4*(1))*2*3
((4*1)*(2*3)
4*(1*(2*3))
((4/1)*2)*3
((4/1)*(2*3)
Time execution: 89 ms
Apakah hasil mau disimpan ke dalam file?
1. YA
2. TIDAK
>> 1
Input nama file: test4
```

3.1.6. Contoh input dan output 6.



3.2. Link Repository

https://github.com/AlphaThrone/Tucil1_13521012.git

LAMPIRAN

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	✓	

REFERENSI

- *https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tucil1-Stima-2023.pdf*
- *https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf*