

LAPORAN TUGAS KECIL 1

IF2211 Strategi Algoritma

Permainan Kartu 24

Menyusun banyaknya kemungkinan angka menjadi 24 dengan Algoritma *Brute Force*



Oleh:

Agsha Athalla Nurkareem

13521027

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
1.1. Pengertian Algoritma Brute Force	3
1.2. Penggunaan Algoritma Brute Force dalam program	3
BAB II.....	5
2.1 File splashScreen.cpp	5
2.2 File 24Solver.cpp	5
BAB III	13
3.1 Hasil Pengujian 1	13
3.2 Hasil Pengujian 2	14
3.3 Hasil Pengujian 3	15
3.4 Hasil Pengujian 4	16
3.5 Hasil Pengujian 5	16
3.6 Hasil Pengujian 6	17
BAB 4	20
4.1 Tautan <i>repository</i> GitHub	20
4.2 Cek list.....	20

BAB I

Algoritma Brute Force

1.1. Pengertian Algoritma Brute Force

Algoritma brute force merupakan suatu pendekatan yang lempang atau *straightforward* untuk memecahkan suatu masalah. Biasanya, Algoritma Brute Force memecahkan masalah dengan cara yang sangat sederhana, langsung dan dengan cara yang jelas. Walaupun Algoritma Brute Force bukan merupakan satu-satunya metode yang efektif untuk memecahkan suatu permasalahan, tetapi hampir semua permasalahan dapat diselesaikan dengan metode ini.

Algoritma brute force umumnya tidak "cerdas". Hal ini disebabkan karena algoritma ini membutuhkan volume komputasi yang besar serta waktu yang biasanya cukup lama untuk menyelesaikan suatu permasalahan. Hal ini yang membuat algoritma ini juga disebut dengan algoritma naif. Algoritma brute force lebih cocok untuk permasalahan yang memiliki input yang kecil, karena implementasinya mudah dan sederhana sehingga algoritma ini sering digunakan sebagai pembandingan dengan algoritma lain yang lebih cepat dan efektif.

1.2. Penggunaan Algoritma Brute Force dalam program

Permainan kartu 24 merupakan salah satu cara bermain yang menggunakan kartu remi sebagai medianya. Permainan ini akan dimulai dengan hadirnya 4 kartu yang telah dikocok sebelumnya. Kartu tersebut diantaranya terdiri dari kartu As yang bernilai satu, kartu 2 sampai 10, kartu Jack yang bernilai 11, kartu Queen yang bernilai 12, dan kartu King yang bernilai 13. Pemain diminta untuk membuat suatu persamaan matematika sederhana dengan operasi sederhana untuk membentuk keempat kartu tersebut menjadi angka 24. Operasi yang diizinkan hanyalah penjumlahan, pengurangan, perkalian, dan pembagian. Pemain diperbolehkan untuk memakai kurung untuk memudahkan pencarian.

Penulis menggunakan bahasa pemrograman C++ untuk memecahkan permainan ini, kemungkinan dari persamaan matematika itu dapat dipecahkan dengan algoritma brute force. Pada program, mulanya meminta pengguna untuk memasukkan angka 2-3, huruf A (As), J (jack), dan Q (queen) sesuai dengan peraturan sebanyak empat kali. Pengguna juga dapat meminta program untuk membuat masukan secara acak. Karena pada hal ini tipe masukannya tidak selalu *integer* maupun *character*, penulis menggunakan tipe data *string* terlebih dahulu untuk menyamakan masukan. Setelah proses validasi terhadap kartu, apabila tidak sesuai maka program akan meminta masukan kembali. Jika proses sudah valid, maka program akan mengubah data tersebut menjadi bertipe *float* untuk kemudian akan dimasukkan ke dalam algoritma brute force.

Penulis membuat suatu *array* yang berisi kemungkinan angka-angka yang dapat disusun dengan empat masukan tadi, dalam hal ini dapat menggunakan permutasi untuk membuat kemungkinan urutan kartu tersebut. Penulis juga membuat suatu *array* yang berisi kemungkinan operasi apa saja selain penggunaan kurung yang dapat di sisipkan di antara keempat kartu tersebut sehingga penulis dapat membuat perulangan dari *array* tersebut. Setelah program selesai membuat kedua *array* tersebut, penulis membuat suatu perulangan di dalam perulangan untuk melakukan operasi di dalam perulangan tersebut. Penulis juga memasukkan kemungkinan bentuk kurung yang dapat dipakai dalam persamaan. Setiap kemungkinan kemudian akan dilakukan operasi matematikanya dalam bentuk bilangan dengan desimal. Apabila suatu kemungkinan mendapat hasil 24.0, maka program akan menghitung banyaknya kemungkinan yang ada dan memasukkan bentuk persamaan tersebut ke dalam suatu list yang kemudian list tersebut dapat disimpan dalam bentuk *file* txt.

BAB II

Source Code dalam Bahasa C++

2.1 File splashScreen.cpp

```
#include <iostream>

void splashScreen(){
    std::cout <<"                                     "<< std::endl;
    std::cout <<"   _ | | ____ _(_) | |__ \\| || | "<<
std::endl;
    std::cout <<" /   \\ / _' | | / _ \\ | | __|   _) | | | _ "<<
std::endl;
    std::cout <<" /  //\\ \\ \\ ( _ | |   <  _/ | | | _ / _/_|_ _|"<<
std::endl;
    std::cout <<"\\ /      \\ \\ \\ _ , _ | \\ \\ _ | | _ \\ | | _ _|"<<
std::endl;
    std::cout << std::endl;
    std::cout <<"          -----"<< std::endl;
    std::cout <<"          Nama : Agsha Athalla Nurkareem"<< std::endl;
    std::cout <<"          NIM : 13521027"<< std::endl;
    std::cout <<"          Mata Kuliah : IF2211"<< std::endl;
    std::cout <<"          -----"<< std::endl;
    std::cout <<" "<< std::endl;
    std::cout <<" "<< std::endl;
}
```

2.2 File 24Solver.cpp

```
#include <iostream>
#include <random>
#include <string>
#include <fstream>
#include "splashScreen.cpp"
#include <vector>
#include <bits/stdc++.h>
using namespace std;

float b[25], c[25], d[25], e[25];
vector <string> forPrint;
string hasil;

bool tepat(string a){
```

```

        if(a == "A" || a == "2" || a == "3" || a == "4" || a == "5" || a == "6" ||
a == "7" || a == "8" || a == "9" || a == "10" || a == "J" || a == "Q" || a ==
"K"){
            return true;
        }
        else{
            return false;
        }
    }
}

float ubah(string a){
    if (a == "A"){
        return 1.0;
    }
    else if (a == "J"){
        return 11.0;
    }
    else if (a == "Q"){
        return 12.0;
    }
    else if (a == "K"){
        return 13.0;
    }
    else{
        return stof(a);
    }
    return 0;
}

int randomGenerator(){
    float random = rand() % 13 + 1;
    return random;
}

void addArray(float k, float L, float m, float n, int &N){
    int i;
    bool found;
    found = false;
    i = 0;
    if (k==L && k==m && k==n){
        found = false;
    }
    while (i <= N && found == false){
        if (k == b[i] && L == c[i] && m == d[i] && n == e[i]){
            found = true;
        }
        else{

```

```

        i += 1;
    }
}
if (found == false){
    N = N+1;
    b[N] = k;
    c[N] = l;
    d[N] = m;
    e[N] = n;
}
}

float operasi(float a, char ops, float b){
    if (ops == '+'){
        return a+b;
    }
    else if (ops == '-'){
        return a-b;
    }
    else if (ops == '*'){
        return a*b;
    }
    else if (ops == '/'){
        return a/b;
    }
    return 0;
}

void writeFile(vector<string> forPrint, string nama, int w, int x, int y, int
z, double waktu, int count){
    ofstream file("./test/"+ nama + ".txt");
    file << "Daftar kartu : " << w << " " << x << " " << y << " " << z <<
endl;
    file << " _____" << endl;
    file << count << " Solution(s) found." << endl;
    file << "Time required: " << waktu << " seconds" << endl;
    file << " _____" << endl;
    file << "Jawaban: " << endl;
    if (count > 0){
        for (int i = 0; i < forPrint.size(); i++){
            file << forPrint[i] << endl;
        }
    }
    file.close();
}

void buatKombinasi(float w, float x, float y, float z, int &N){
    float siapPermutasi[] = {w, x, y, z};

```

```

    int ukuran = 4;
    for (int i=0; i<ukuran; i++){
        for (int j=0; j<ukuran; j++){
            for (int k=0; k<ukuran; k++){
                for (int l=0; l<ukuran; l++){
                    if (i != j && i != k && i != l && j != k && j != l && k !=
1){
                        addArray(siapPermutasi[i], siapPermutasi[j],
siapPermutasi[k], siapPermutasi[l], N);
                    }
                }
            }
        }
    }
}

void menu(){
    int N = 0;
    int pilihan;
    cout <<"Anda dapat memasukkan 4 angka (2-10) dan huruf (A,J,K,Q) atau
membuatnya secara random" << endl;
    cout <<"1. memasukkan angka/huruf" << endl;
    cout <<"2. membuat angka/huruf secara random" << endl;
    cout <<"_____ " << endl;
    cout <<"Masukkan pilihan anda : ";
    cin >> pilihan;
    while (pilihan != 1 && pilihan != 2){
        cout << "_____ " << endl;
        cout << "Masukkan angka 1 atau 2" << endl;
        cout <<"Masukkan pilihan anda : ";
        cin >> pilihan;
    }
    string a1, a2, a3, a4;
    float w,x,y,z;
    if(pilihan == 1){
        cout << "Masukkan 4 angka/huruf" << endl;
        cin >> a1 >> a2 >> a3 >> a4;
        while(tepat(a1) == false || tepat(a2) == false || tepat(a3) == false
|| tepat(a4) == false){
            cout << "_____ " << endl;
            cout << "Masukan salah!" << endl;
            cout << "Masukkan 4 angka/huruf" << endl;
            cin >> a1 >> a2 >> a3 >> a4;
        }
        w = ubah(a1);
        x = ubah(a2);
        y = ubah(a3);
        z = ubah(a4);
    }
}

```



```

    }
    else if(pilihan == 2){
        srand(time(0));
        w = rand() % 13 + 1;
        x = rand() % 13 + 1;
        y = rand() % 13 + 1;
        z = rand() % 13 + 1;
    }
    cout << "anda memasukkan angka [" << w << " " << x << " " << y << " " << z
<< "]" << endl;
    buatKombinasi(w,x,y,z,N);

    string ops[64] = {"+++", "++-", "++*", "++/", "+-+", "+--", "+-*", "+-/",
"+++", "+*- ", "+**", "+*/", "+/+ ", "+/- ", "+/*", "+//", "-++", "-+-", "-+*",
"-+/", "--+", "---", "--*", "--/", "-*+", "-*- ", "-**", "-*/", "-/+ ", "-/- ",
"-/*", "-//", "*++", "*+-", "*+*", "*+/", "*-+", "*--", "*-*", "*-/", "***",
***-", "****", "***/", "**/+ ", "**/- ", "**/*", "**//", "/++", "/+-", "/+*", "/+/",
"/-+", "/--", "/-*", "/-/ ", /*+", /*- ", /***", /*/*", //+", //- ", //**",
///"};
    //pewaktu
    auto start = chrono::high_resolution_clock::now();
    int count = 0;
    //kita mulai algoritmanya
    for (int i=1;i<=N;i++){
        for (int j=0; j<64;j++){
            //(b+c)+d)+e
            float uji1 = operasi(operasi(operasi(b[i], ops[j][0], c[i]),
ops[j][1], d[i]), ops[j][2], e[i]);
            if(abs(uji1-24) == 0){
                hasil = "(" + to_string(int(b[i])) + ops[j][0] +
to_string(int(c[i])) + ")" + ops[j][1] + to_string(int(d[i])) + ")" +
ops[j][2] + to_string(int(e[i])) + "";
                forPrint.push_back(hasil);
                count += 1;}
            // (b+(c+d))+e
            float uji2 = operasi(operasi(b[i], ops[j][0], operasi(c[i],
ops[j][1], d[i])), ops[j][2], e[i]);
            if(abs(uji2-24) == 0){
                hasil = "(" + to_string(int(b[i])) + ops[j][0] + "(" +
to_string(int(c[i])) + ops[j][1] + to_string(int(d[i])) + ")" + ops[j][2] +
to_string(int(e[i])) + "";
                forPrint.push_back(hasil);
                count += 1;}
            // b+((c+d)+e)
            float uji3 = operasi(b[i], ops[j][0], operasi(operasi(c[i],
ops[j][1], d[i]), ops[j][2], e[i]));
            if(abs(uji3-24) == 0){

```

```

        hasil = to_string(int(b[i])) + ops[j][0] + "(" +
to_string(int(c[i])) + ops[j][1] + to_string(int(d[i])) + ")" + ops[j][2] +
to_string(int(e[i])) + " ";
        forPrint.push_back(hasil);
        count += 1;}
    // b+(c+(d+e))
    float uji4 = operasi(b[i], ops[j][0], operasi(c[i], ops[j][1],
operasi(d[i], ops[j][2], e[i])));
    if(abs(uji4-24) == 0){
        hasil = to_string(int(b[i])) + ops[j][0] + "(" +
to_string(int(c[i])) + ops[j][1] + "(" + to_string(int(d[i])) + ops[j][2] +
to_string(int(e[i])) + "));";
        forPrint.push_back(hasil);
        count += 1;}
    // (b+c)+(d+e)
    float uji5 = operasi(operasi(b[i], ops[j][0], c[i]), ops[j][1],
operasi(d[i], ops[j][2], e[i]));
    if(abs(uji5-24) == 0){
        hasil = "(" + to_string(int(b[i])) + ops[j][0] +
to_string(int(c[i])) + ")" + ops[j][1] + "(" + to_string(int(d[i])) +
ops[j][2] + to_string(int(e[i])) + "));";
        forPrint.push_back(hasil);
        count += 1;}

    }
}
//print hasil
cout << " _____" << endl;

if (count == 0) {
    cout << "Tidak ada solusi" << endl;
}
else{
    cout << "Telah ditemukan " << count << " solusi" << endl;
    cout << " _____" << endl;
    cout << "Hasil pencarian: " << endl;
    for (size_t i = 0; i < forPrint.size(); i++) {
        cout << forPrint[i] << endl;
    }
}

cout << " _____" << endl;

//pewaktu
auto end1 = chrono::high_resolution_clock::now();
double time_taken = chrono::duration_cast<chrono::nanoseconds>(end1 -
start).count();
time_taken *= 1e-9;

```

```

cout << "Waktu pencarian jawaban : " << time_taken << setprecision(9);
cout << " detik" << endl;

char pil;
cout << "" << endl;
cout << "Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n):
";
cin >> pil;

if (pil == 'y') {
    cout << "Masukkan nama file: ";
    string namaFile;
    cin >> namaFile;
    writeFile(forPrint, namaFile, w, x, y, z, time_taken, count);
    cout << "File " << namaFile << ".txt berhasil disimpan!" << endl;

}

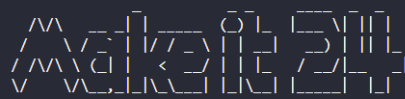
//pewaktu
auto end2 = chrono::high_resolution_clock::now();
double time_taken2 = chrono::duration_cast<chrono::nanoseconds>(end2 -
start).count();
time_taken2 *= 1e-9;
cout << "Waktu eksekusi program : " << time_taken2 << setprecision(9);
cout << " detik" << endl;
}

int main(){
    system("CLS");
    char restart;
    auto mulai = chrono::high_resolution_clock::now();
    do{
        forPrint.clear();
        splashScreen();
        menu();
        cout << "Apakah anda ingin mencoba kembali? (y/n): ";
        cin >> restart;
        system("CLS");
    } while(restart == 'y');
    splashScreen();
    auto selesai = chrono::high_resolution_clock::now();
    double waktu = chrono::duration_cast<chrono::nanoseconds>(selesai -
mulai).count();
    waktu *= 1e-9;
    cout << "Waktu program berjalan : " << waktu << setprecision(9);
    cout << " detik" << endl;
    cout << "Terima kasih telah menggunakan program ini" << endl;
    cout << " _____ " << endl;

```

```
getchar();  
return 0;  
}
```


3.2 Hasil Pengujian 2



Nama : Agsha Athalla Nurkareem
NIM : 13521027
Mata Kuliah : IF2211

Anda dapat memasukkan 4 angka (2-10) dan huruf (A,J,K,Q) atau membuatnya secara random
1. memasukkan angka/huruf
2. membuat angka/huruf secara random

Masukkan pilihan anda : 1
Masukkan 4 angka/huruf
11 12 13 1

Masukan salah!
Masukkan 4 angka/huruf
2 7 4 9
anda memasukkan angka [2 7 4 9]

Telah ditemukan 40 solusi

Hasil pencarian:

2*((7-4)+9)
2*(7-(4-9))
2*((7+9)-4)
2*(7+(9-4))
((2*4)+7)+9
(2*4)+(7+9)
((2*4)+9)+7
(2*4)+(9+7)
2*((9+7)-4)
2*(9+(7-4))
2*((9-4)+7)
2*(9-(4-7))
(7+(2*4))+9
7+((2*4)+9)

(7+(4*2))+9
7+((4*2)+9)
((7-4)+9)*2
(7-(4-9))*2
7+(9+(2*4))
(7+9)+(2*4)
7+(9+(4*2))
(7+9)+(4*2)
((7+9)-4)*2
(7+(9-4))*2
((4*2)+7)+9
(4*2)+(7+9)
((4*2)+9)+7
(4*2)+(9+7)
(9+(2*4))+7
9+((2*4)+7)
9+(7+(2*4))
(9+7)+(2*4)
9+(7+(4*2))
(9+7)+(4*2)
((9+7)-4)*2
(9+(7-4))*2
(9+(4*2))+7
9+((4*2)+7)
((9-4)+7)*2
(9-(4-7))*2

Waktu pencarian jawaban : 0.0091987 detik

Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n): y
Masukkan nama file: Hasil2
File Hasil2.txt berhasil disimpan!
Waktu eksekusi program : 8.9497343 detik
Apakah anda ingin mencoba kembali? (y/n):

3.3 Hasil Pengujian 3

/M _ | _ _ _ _ O _ | _ _ _ _
 / \ / _ | / / \ | _ | _ _ | |
 /M \ C | | < _ | | | / / | _
 V _ \ , | \ \ | | \ | _ _ | |

Nama : Agsha Athalla Nurkareem
NIM : 13521027
Mata Kuliah : IF2211

Anda dapat memasukkan 4 angka (2-10) dan huruf (A,J,K,Q) atau membuatnya secara random

1. memasukkan angka/huruf
2. membuat angka/huruf secara random

Masukkan pilihan anda : 1

Masukkan 4 angka/huruf

A 7 3 Q

anda memasukkan angka [1 7 3 12]

Telah ditemukan 8 solusi

Hasil pencarian:

$$((7-1)/3)*12$$
$$(7-1)/(3/12)$$
$$((7-1)*12)/3$$
$$(7-1)*(12/3)$$
$$(12 * (7 - 1)) / 3$$
$$12 * ((7-1)/3)$$
$$(12/3) * (7-1)$$
$$12 / (3 / (7 - 1))$$

Waktu pencarian jawaban : 0.0020517 detik

```
Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n): y
```

```
Masukkan nama file: Hasil3
```

```
File Hasil3.txt berhasil disimpan!
```

Waktu eksekusi program : 7.9566559 detik

Apakah anda ingin mencoba kembali? (y/n):

3.4 Hasil Pengujian 4

```

      /\_ _-| |____ O_| |___\|| |
    / \_/ \_| |/_/\ || |__ )|| |
  // \ \ C| | < _/ | | | /_/ | |
V   V\_,_| |\_\_| | \_| | ____ | |

-----
Nama : Agsha Athalla Nurkareem
NIM : 13521027
Mata Kuliah : IF2211
-----

Anda dapat memasukkan 4 angka (2-10) dan huruf (A,J,K,Q) atau membuatnya secara random
1. memasukkan angka/huruf
2. membuat angka/huruf secara random

-----
Masukkan pilihan anda : 2
anda memasukkan angka [7 8 9 2]

-----
Telah ditemukan 4 solusi

-----
Hasil pencarian:
((7+9)*2)-8
((9+7)*2)-8
(2*(7+9))-8
(2*(9+7))-8

-----
Waktu pencarian jawaban : 0.0011222 detik

Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n): y
Masukkan nama file: Hasil4
File Hasil4.txt berhasil disimpan!
Waktu eksekusi program : 6.9466722 detik
Apakah anda ingin mencoba kembali? (y/n): 

```

3.5 Hasil Pengujian 5

```

/\ \ _ _ _ | | _ _ _ O _ | | _ _ _ \ | | |
/ \ \ / _ _ | | / / _ \ | | _ | | _ _ _ \ | | |
/ \ \ \ C | | < _ | | | _ | / _ | _ _ _ |
V _ V \ _ _ | | \ _ _ | | \ _ _ | _ _ _ | _ _

-----
Nama : Agsha Athalla Nurkareem
NIM : 13521027
Mata Kuliah : IF2211
-----

Anda dapat memasukkan 4 angka (2-10) dan huruf (A,J,K,Q) atau membuatnya secara random
1. memasukkan angka/huruf
2. membuat angka/huruf secara random

-----
Masukkan pilihan anda : 2
anda memasukkan angka [11 11 13 5]

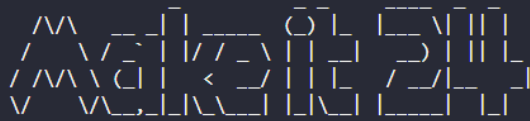
-----
Tidak ada solusi

-----
Waktu pencarian jawaban : 0.001129 detik

Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n): y
Masukkan nama file: Hasil5
File Hasil5.txt berhasil disimpan!
Waktu eksekusi program : 91.6415341 detik
Apakah anda ingin mencoba kembali? (y/n):

```


3.6 Hasil Pengujian 6



Nama : Agsha Athalla Nurkareem
NIM : 13521027
Mata Kuliah : IF2211

Anda dapat memasukkan 4 angka (2-10) dan huruf (A,J,K,Q) atau membuatnya secara random

1. memasukkan angka/huruf
2. membuat angka/huruf secara random

Masukkan pilihan anda : 2
anda memasukkan angka [3 7 10 4]

Telah ditemukan 140 solusi

Hasil pencarian:

((3+7)+10)+4
(3+(7+10))+4
3+((7+10)+4)
3+(7+(10+4))
(3+7)+(10+4)
((3-7)+10)*4
(3-(7-10))*4
((3+7)+4)+10
(3+(7+4))+10
3+((7+4)+10)
3+(7+(4+10))
(3+7)+(4+10)
(3-7)*(4-10)
((3+10)+7)+4
(3+(10+7))+4
3+((10+7)+4)
3+(10+(7+4))
(3+10)+(7+4)
((3+10)-7)*4
(3+(10-7))*4

$((3+10)-7)*4$
 $(3+(10-7))*4$
 $((3+10)+4)+7$
 $(3+(10+4))+7$
 $3+((10+4)+7)$
 $3+(10+(4+7))$
 $(3+10)+(4+7)$
 $((3+4)+7)+10$
 $(3+(4+7))+10$
 $3+((4+7)+10)$
 $3+(4+(7+10))$
 $(3+4)+(7+10)$
 $((3+4)+10)+7$
 $(3+(4+10))+7$
 $3+((4+10)+7)$
 $3+(4+(10+7))$
 $(3+4)+(10+7)$
 $((7+3)+10)+4$
 $(7+(3+10))+4$
 $7+((3+10)+4)$
 $7+(3+(10+4))$
 $(7+3)+(10+4)$
 $(7-3)*(10-4)$
 $((7+3)+4)+10$
 $(7+(3+4))+10$
 $7+((3+4)+10)$
 $7+(3+(4+10))$
 $(7+3)+(4+10)$
 $((7+10)+3)+4$
 $(7+(10+3))+4$
 $7+((10+3)+4)$
 $7+(10+(3+4))$
 $(7+10)+(3+4)$
 $((7+10)+4)+3$
 $(7+(10+4))+3$
 $7+((10+4)+3)$
 $7+(10+(4+3))$
 $(7+10)+(4+3)$
 $((7+4)+3)+10$
 $(7+(4+3))+10$
 $7+((4+3)+10)$
 $7+(4+(3+10))$
 $(7+4)+(3+10)$

$(10+3)+(4+7)$
 $((10+7)+3)+4$
 $(10+(7+3))+4$
 $10+((7+3)+4)$
 $10+(7+(3+4))$
 $(10+7)+(3+4)$
 $((10-7)+3)*4$
 $(10-(7-3))*4$
 $((10+7)+4)+3$
 $(10+(7+4))+3$
 $10+((7+4)+3)$
 $10+(7+(4+3))$
 $(10+7)+(4+3)$
 $((10+4)+3)+7$
 $(10+(4+3))+7$
 $10+((4+3)+7)$
 $10+(4+(3+7))$
 $(10+4)+(3+7)$
 $((10+4)+7)+3$
 $(10+(4+7))+3$
 $10+((4+7)+3)$
 $10+(4+(7+3))$
 $(10+4)+(7+3)$
 $(10-4)*(7-3)$
 $((4+3)+7)+10$
 $(4+(3+7))+10$
 $4+((3+7)+10)$
 $4+(3+(7+10))$
 $(4+3)+(7+10)$
 $4*((3-7)+10)$
 $4*(3-(7-10))$
 $((4+3)+10)+7$
 $(4+(3+10))+7$
 $4+((3+10)+7)$
 $4+(3+(10+7))$
 $(4+3)+(10+7)$
 $4*((3+10)-7)$
 $4*(3+(10-7))$
 $((4+7)+3)+10$
 $(4+(7+3))+10$
 $4+((7+3)+10)$
 $4+(7+(3+10))$
 $(4+7)+(3+10)$

```

((4+7)+10)+3
(4+(7+10))+3
4+((7+10)+3)
4+(7+(10+3))
(4+7)+(10+3)
((4+10)+3)+7
(4+(10+3))+7
4+((10+3)+7)
4+(10+(3+7))
(4+10)+(3+7)
(4-10)*(3-7)
4*((10+3)-7)
4*(10+(3-7))
((4+10)+7)+3
(4+(10+7))+3
4+((10+7)+3)
4+(10+(7+3))
(4+10)+(7+3)
4*((10-7)+3)
4*(10-(7-3))

-----
Waktu pencarian jawaban : 0.0673143 detik

Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n): y
Masukkan nama file: Hasil6
File Hasil6.txt berhasil disimpan!
Waktu eksekusi program : 505.58679 detik
Apakah anda ingin mencoba kembali? (y/n): 

```

BAB 4

LAMPIRAN

4.1 Tautan *repository* GitHub

https://github.com/agshaathalla/Tucil1_13521027

4.2 Cek list

Poin	Ya	Tidak
Program berhasil dikompilasi tanpa kesalahan	✓	
Program berhasil <i>running</i>	✓	
Program dapat membaca <i>input</i> / <i>generate</i> sendiri dan memberikan luaran	✓	
Solusi yang diberikan program memenuhi (berhasil mencapai 24)	✓	
Program dapat menyimpan solusi dalam <i>file</i> teks	✓	