

Tugas Kecil 1 IF2211 Strategi Algoritma
24 Game Solver dengan Algoritma Brute Force



Disusun oleh:

13521127 – Marcel Ryan Antony

INSTITUT TEKNOLOGI BANDUNG

2022

DAFTAR ISI

DAFTAR ISI

BAB 1	3
DESKRIPSI MASALAH DAN ALGORITMA	3
1.1 Algoritma Brute Force.....	3
1.2 Permainan Kartu 24	3
1.3 Algoritma 24 Game Solver dengan Algoritma Brute Force.....	3
BAB 2	5
IMPLEMENTASI ALGORITMA DALAM BAHASA C++	5
BAB 3	7
SOURCE CODE PROGRAM	7
3.1 Repository Program.....	7
3.2 Source Code Program.....	7
BAB 4	19
EKSPERIMEN.....	19
REFERENSI	28
LAMPIRAN.....	29
Checklist Fitur.....	29

BAB 1

DESKRIPSI MASALAH DAN ALGORITMA

1.1 Algoritma Brute Force

Algoritma *Brute Force* adalah sebuah algoritma yang memecahkan suatu persoalan algoritmik dengan menggunakan pendekatan yang lempeng (*straightforward*). Algoritma *Brute Force* sendiri didasarkan pada pernyataan pada persoalan algoritmik (*problem statement*) dan juga definisi/konsep yang dilibatkan pada persoalan algoritmik tersebut.

Ciri khas dari algoritma dengan pendekatan *Brute Force* adalah cara algoritma ini memecahkan persoalan yaitu dengan sangat sederhana, langsung, dan jelas. Persoalan yang diselesaikan dengan pendekatan *Brute Force* biasanya akan dilakukan dengan mengenumerasi segala kemungkinan solusi, dan pada akhirnya akan mengambil solusi terbaik apabila pada persoalan tersebut terdapat solusi. Algoritma *Brute Force* sendiri sudah pasti akan menghasilkan sebuah solusi apabila solusi tersebut ada. Namun, karena pendekatan algoritma *Brute Force* rata-rata dilakukan dengan mengenumerasi segala kemungkinan solusi, hal ini membuat algoritma ini menjadi tidak efisien, dengan kompleksitas waktu yang sangat tinggi. Meskipun begitu, terdapat beberapa persoalan yang hanya dapat diselesaikan dengan algoritma *Brute Force*, salah satunya adalah persoalan mencari jumlah nilai dari satu array.

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 ini menarik cukup banyak peminat dikarenakan dapat meningkatkan kemampuan berhitung serta mengasah otak agar dapat berpikir dengan cepat dan akurat. Permainan Kartu 24 biasa dimainkan dengan menggunakan kartu remi. Kartu remi terdiri dari 52 kartu yang terbagi menjadi empat suit (sekop, hati, keriting, dan wajik) yang masing-masing terdiri dari 13 kartu (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). Yang perlu diperhatikan hanyalah nilai kartu yang didapat (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). As bernilai 1, Jack bernilai 11, Queen bernilai 12, King bernilai 13, sedangkan kartu bilangan memiliki nilai dari bilangan itu sendiri. Pada awal permainan moderator atau salah satu pemain mengambil 4 kartu dari dek yang sudah dikocok secara random. Permainan berakhir ketika pemain berhasil menemukan solusi untuk membuat kumpulan nilainya menjadi 24. Pengubahan nilai tersebut dapat dilakukan menggunakan operasi dasar matematika penjumlahan (+), pengurangan (-), perkalian (\times), divisi (/) dan tanda kurung (). Tiap kartu harus digunakan tepat sekali dan urutan penggunaannya bebas.

1.3 Algoritma 24 Game Solver dengan Algoritma Brute Force

Dalam menyelesaikan persoalan 24 Game Solver, penulis menggunakan algoritma *Brute Force*. Berikut langkah-langkah penyelesaian persoalan tersebut dijelaskan secara deskriptif :

1. Menerima input dari user berupa pilihan apakah user ingin angka di *generate* secara random atau langsung dari user sendiri
2. Tampung angka-angka yang didapat dalam sebuah array

3. Buat dua array baru untuk menampung segala susunan angka dan segala susunan tiga operator
4. Setelah terdapat dua array yang menampung segala susunan angka dan segala susunan operator, lakukan looping untuk mencari semua kemungkinan dengan segala variasi peletakkan kurung
5. Buat array baru untuk menampung semua solusi yang ada
6. Setelah mendapatkan segala variasi peletakkan kurung, susunan angka, dan susunan operator, lakukan evaluasi untuk segala kemungkinan tersebut. Apabila menghasilkan angka 24 maka masukkan variasi tersebut kedalam array solusi yang telah ada
7. Keluarkan semua solusi tersebut ke terminal

BAB 2

IMPLEMENTASI ALGORITMA DALAM BAHASA C++

Dalam pembuatan program ini, penulis menggunakan bahasa pemrograman C++. Program ini hanya dibuat dengan menggunakan 1 file saja yang berisi beberapa fungsi-fungsi pembantu algoritma dan algoritma utama yaitu 24Solver.cpp. Berikut fungsi-fungsi pembantu yang terdapat pada file 24Solver.cpp.

Nama	Parameter	Output	Deskripsi
cekInputValidKartu	String input	Integer 1 atau 0	Fungsi ini digunakan untuk mengecek apakah input dari user valid atau tidak, jika valid akan mengembalikan 1 dan jika tidak akan mengembalikan 0
Operasi	Double a, b Char ops	Double hasil	Fungsi ini digunakan untuk mencari hasil operasi antara dua angka dan operator yang ada pada parameter
ubahNilai	Char input	Int nilai	Fungsi ini digunakan untuk mengubah input-input dari user yang masih bertipe char menjadi integer agar dapat diolah
ubahKartu	Char input	String hasil	Fungsi ini digunakan untuk mengubah input user yang masih berupa huruf seperti A J Q K menjadi angka 1 11 12 13
ubahNilaitoString	String input	String hasil	Fungsi ini kebalikan dari fungsi ubahKartu dimana fungsi ini akan mengubah 1 11 12 13 menjadi A J Q K
isInVector	Vector <string> solusi String kalimat	Bool ada	Fungsi ini digunakan untuk mengecek apakah solusi telah ada di dalam array solusi atau belum

			jika ada akan meng return true dan jika tidak akan meng return false
lengthstring	String kalimat	Int Panjang	Fungsi ini digunakan untuk menghitung panjang sebuah string
lengthvector	Vector <string> dummy	Int panjang	Fungsi ini digunakan untuk menghitung panjang sebuah array vector
makeEmpty	Vector <string> dummy	Tidak ada karena merupakan void	Fungsi ini digunakan untuk membuat vector kosong

BAB 3

SOURCE CODE PROGRAM

3.1 Repository Program

Repository program dapat diakses melalui :

- Github : https://github.com/MarcelRyan/Tucil1_13521127

3.2 Source Code Program

```
#include<iostream>
#include<vector>
#include<string>
#include<fstream>
#include<time.h>

using namespace std;

int cekInputValidKartu(string input) {
    // Function untuk mengecek apakah input kartu valid atau tidak
    if (input == "A" || input == "Q" || input == "J" || input == "K" || input == "2" || input == "3" || input == "4" || input == "5" || input == "6" || input == "7" || input == "8" || input == "9" || input == "10") {
        return 1;
    }
    else {
        return 0;
    }
}

double Operasi(double a, double b, char ops) {
    // Function untuk melakukan operasi penjumlahan, pengurangan, perkalian, dan pembagian
    if (ops == '+') {
        return a + b;
    }
    else if (ops == '-') {
        return a-b;
    }
    else if (ops == '*') {
        return a*b;
    }
    else if (ops == '/') {
        return a/b;
    }
}

int ubahNilai(char input) {
    // Function untuk mengubah nilai input menjadi angka apabila input berupa A J Q K
```

```

    if (input == 'A') {
        return 1;
    }
    else if (input == 'J') {
        return 11;
    }
    else if (input == 'Q') {
        return 12;
    }
    else if (input == 'K') {
        return 13;
    }
    else if (input == 'L') {
        return 10;
    }
    return input-'0';
}

string ubahKartu(char input) {
    // Function untuk mengubah kartu A J Q K menjadi string angka
    if (input == 'A') {
        return "1";
    }
    else if (input == 'J') {
        return "11";
    }
    else if (input == 'Q') {
        return "12";
    }
    else if (input == 'K') {
        return "13";
    }
    else if (input == 'L') {
        return "10";
    }
    else {
        string temp(1, input);
        return temp;
    }
}

string ubahNilaitoString(string input) {
    // Function untuk mengubah angka 11 12 13 10 dan 1 menjadi huruf J Q K L dan A
    if (input == "1") {
        return "A";
    }
    else if (input == "10") {

```



```

        return "L";
    }
    else if (input == "11") {
        return "J";
    }
    else if (input == "12") {
        return "Q";
    }
    else if (input == "13") {
        return "K";
    }
    else {
        return input;
    }
}

bool isInVector(vector<string> solusi, string kalimat) {
    // Function untuk mengecek apakah solusi sudah masuk ke dalam array
    for (string i : solusi) {
        if (i == kalimat) {
            return true;
        }
    }
    return false;
}

int lengthstring(string hehe) {
    // Function untuk menghitung panjang string
    int panjang = 0;
    for (char c : hehe) {
        panjang++;
    }
    return panjang;
}

int lengthvector(vector<string> dummy) {
    // Function untuk mencari panjang dari sebuah array vector
    int panjang = 0;
    for (string i : dummy) {
        panjang++;
    }
    return panjang;
}

void makeEmpty (vector<string> * solusi) {
    // Function untuk membuat array angka kembali kosong apabila input tidak
    // valid dan user masih ingin memakai program
    while (lengthvector(*solusi) != 0) {

```

```

        solusi->pop_back();
    }
}

int main() {
    // Container
    vector <string> number;
    string ops[4] = {"+", "-", "*", "/"};

    // Menu awal
    string lanjut = "y";
    while (lanjut == "y") {
        // ASCII ART
        cout << R"(
                                ,--
,
,      ,----,      ,--.'|      .----.      ,--
,
.'    .' \      ,--, | :      / /      '      ,--
.'|
,----,'      |,---.'| : '      | : /` /      ,---. | |
:
|      : . ;; : | | ;      ; | |--`      ' ,'\ : : '      .---
.      ,', ,'/ /|
;      |.' / | | : _' |      | : ;_      / / || ' |      / . ./| ,--
--.' | | | |
`-----'/ ; : : |.' |      \ \      ` . . ; ,. : ' | | .-' . ' |
/      \ | | ,',
/ ; / | ' ' ; :      `-----. \ ' | | : | | : /___/ \:
|/      / |' : /
; / /-, \ \ .' . |
.      ' / || | '
/ / /.' | `---` : | '      / /`--' /| : || |
.' | \ \ ' ' ; /| ; : |
./___;      :      ' ; |      '---'.      / \ \ / ; : ;
\ \ ' | / || , ;      `---'---' `---' | , / \ \
| :      .'      | : ;      `---'---' `---' | , / \ \
| :      | ---'
; | .'      ' ,/      ---`-'      '---"
\ \ /
`---'      '---'
--'
)" << endl;
        cout << "Selamat datang di program 24 Solver" << endl;

```

```

        cout << "Berikut beberapa pilihan yang terdapat pada program ini :" << endl;

        cout << "1. Nilai-Nilai Kartu akan digenerate secara random" << endl;
        cout << "2. Nilai-nilai kartu di input sendiri" << endl;
        cout << "Silahkan input pilihan Anda : ";
        string pilihan;
        cin >> pilihan;
        while (pilihan != "1" && pilihan != "2") {
            cout << "Pilihan tidak sesuai, silahkan masukan pilihan berupa angka 1 atau 2" << endl;
            cout << "Silahkan input pilihan Anda : ";
            cin >> pilihan;
        }
        if (pilihan == "1") {
            srand(time(NULL));
            int number1 = (rand() % 13) + 1;
            int number2 = (rand() % 13) + 1;
            int number3 = (rand() % 13) + 1;
            int number4 = (rand() % 13) + 1;

            cout << "Berikut bilangan-bilangan yang didapat :" << endl;
            cout << number1 << " " << number2 << " " << number3 << " " << number4 << endl;

            string numberbaru1 = to_string(number1);
            string numberbaru2 = to_string(number2);
            string numberbaru3 = to_string(number3);
            string numberbaru4 = to_string(number4);
            numberbaru1 = ubahNilaitoString(numberbaru1);
            numberbaru2 = ubahNilaitoString(numberbaru2);
            numberbaru3 = ubahNilaitoString(numberbaru3);
            numberbaru4 = ubahNilaitoString(numberbaru4);
            number.push_back(numberbaru1);
            number.push_back(numberbaru2);
            number.push_back(numberbaru3);
            number.push_back(numberbaru4);
        }
        else if (pilihan == "2") {
            // Input nilai-nilai kartu
            /* Note penting pada program ini diasumsikan input selalu berupa value kartu yang valid sehingga input 1 11 12 dan 13 bukan input yang valid dikarenakan pada kartu remi tidak ada kartu dengan angka 1 11 12 dan 13 melainkan hanya ada A J Q K */
            bool valid = false;
            cout << "Silahkan Masukan Nilai Kartu : ";
            cin.ignore();
            while (!valid) {
                string input = "";
                getline(cin, input);
            }
        }
    }
}

```

```

        int n = 0;
        while (n < lengthstring(input)) {
            if (input[n] == '1' && input[n+1] == '0' && n <
lengthstring(input)-1) {
                number.push_back("10");
                n++;
            }
            else if (input[n] != ' ') {
                string temp(1, input[n]);
                number.push_back(temp);
            }
            n++;
        }
        if (lengthvector(number) != 4 && lengthvector(number) > 4) {
            makeEmpty(&number);
            cout << "Input tidak valid, Silahkan masukan input
sebanyak 4 string saja dan berupa angka 1-10 atau huruf A J Q K saja" << endl;
            cout << "Silahkan Masukan Nilai Kartu : ";
        }
        else if (lengthvector(number) == 4) {
            if ((cekInputValidKartu(number[0]) == 0 ||
cekInputValidKartu(number[1]) == 0 || cekInputValidKartu(number[3]) == 0 ||
cekInputValidKartu(number[2]) == 0)) {
                cout << "Input tidak valid, Silahkan masukan input
sebanyak 4 string saja dan berupa angka 1-10 atau huruf A J Q K saja" << endl;
                cout << "Silahkan Masukan Nilai Kartu : ";
                makeEmpty(&number);
            }
            else {
                valid = true;
            }
        }
    }
}

clock_t start = clock();
// Semua kemungkinan susunan angka dan operator
vector <string> susunanAngka;
// Apabila tidak ada angka yang sama
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) {
                    continue;
                }
            }
        }
    }
}

```

```

        string kombinasi = number[i] + number[j] + number[k] +
number[1];
        susunanAngka.push_back(kombinasi);
    }
}

// Kasus khusus apabila semua angka sama
if (number[0] == number[1] && number[0] == number[2] && number[0] ==
number[3]) {
    string kombinasi = number[0] + number[1] + number[2] + number[3];
    susunanAngka.push_back(kombinasi);
}

int index = 0;
for (string i : susunanAngka) {
    string newi = "";
    for (int j = 0; j < lengthstring(i)-1; j++) {
        if (i[j] == '1' && i[j+1] == '0') {
            i[j] = 'L';
            i[j+1] = '\\0';
        }
    }
    for (char c : i) {
        if (c != '\\0') {
            newi += c;
        }
    }
    susunanAngka[index] = newi;
    index++;
}

vector <string> susunanOperator;
for (string i : ops) {
    for (string j : ops) {
        for (string k : ops) {
            string kombinasiops = i + j + k;
            susunanOperator.push_back(kombinasiops);
        }
    }
}

/* Looping untuk kombinasi semua tanda kurung
Semua variasi dari tanda kurung adalah 1. (a op b) op (c op d)
2. a op ((b op c) op d)
3. a op (b op (c op d))
4. ((a op b) op c) op d

```



```

        int angka3 = ubahNilai(susunan[2]);
        int angka4 = ubahNilai(susunan[3]);
        double hasil = Operasi(1.0*angka2, 1.0*angka3,
susunanop[1]);

        double hasil2 = Operasi(hasil, 1.0*angka4,
susunanop[2]);

        if (hasil2 == 0 && susunanop[0] == '/') {
            continue;
        }
        double hasilakhir = Operasi(1.0*angka1, hasil2,
susunanop[0]);

        if (hasilakhir == 24.0 && !isInVector(seluruhSolusi,
solusi)) {
            seluruhSolusi.push_back(solusi);
        }
    }
    if (i == 2) {
        for (string susunanop : susunanOperator) {
            // Mengubah terlebih dahulu tipe data char menjadi
string agar bisa di konkat
            string temp1 = ubahKartu(susunan[0]); string temp2 =
ubahKartu(susunan[1]); string temp3 = ubahKartu(susunan[2]); string temp4 =
ubahKartu(susunan[3]); string temp5(1, susunanop[0]); string temp6(1,
susunanop[1]); string temp7(1, susunanop[2]);
            string solusi = temp1 + " " + temp5 + " ( " + temp2 +
" " + temp6 + " ( " + temp3 + " " + temp7 + " " + temp4 + " ) )";
            int angka1 = ubahNilai(susunan[0]);
            int angka2 = ubahNilai(susunan[1]);
            int angka3 = ubahNilai(susunan[2]);
            int angka4 = ubahNilai(susunan[3]);
            double hasil = Operasi(1.0*angka3, 1.0*angka4,
susunanop[2]);

            if (hasil == 0 && susunanop[1] == '/') {
                continue;
            }
            double hasil2 = Operasi(1.0*angka2, hasil,
susunanop[1]);

            if (hasil2 == 0 && susunanop[0] == '/') {
                continue;
            }
            double hasilakhir = Operasi(1.0*angka1, hasil2,
susunanop[0]);

            if (hasilakhir == 24.0 && !isInVector(seluruhSolusi,
solusi)) {
                seluruhSolusi.push_back(solusi);
            }
        }
    }
}

```

```

    }
    if (i == 3) {
        for (string susunanop : susunanOperator) {
            // Mengubah terlebih dahulu tipe data char menjadi
            string agar bisa di konkat
            string temp1 = ubahKartu(susunan[0]); string temp2 =
            ubahKartu(susunan[1]); string temp3 = ubahKartu(susunan[2]); string temp4 =
            ubahKartu(susunan[3]); string temp5(1, susunanop[0]); string temp6(1,
            susunanop[1]); string temp7(1, susunanop[2]);
            string solusi = "(" + temp1 + " " + temp5 + " " +
            temp2 + " ) " + temp6 + " " + temp3 + " ) " + temp7 + " " + temp4;
            int angka1 = ubahNilai(susunan[0]);
            int angka2 = ubahNilai(susunan[1]);
            int angka3 = ubahNilai(susunan[2]);
            int angka4 = ubahNilai(susunan[3]);
            double hasil = Operasi(1.0*angka1, 1.0*angka2,
            susunanop[0]);
            double hasil2 = Operasi(hasil, 1.0*angka3,
            susunanop[1]);
            double hasilakhir = Operasi(hasil2, 1.0*angka4,
            susunanop[2]);
            if (hasilakhir == 24.0 && !isInVector(seluruhSolusi,
            solusi)) {
                seluruhSolusi.push_back(solusi);
            }
        }
    }
    if (i == 4) {
        for (string susunanop : susunanOperator) {
            // Mengubah terlebih dahulu tipe data char menjadi
            string agar bisa di konkat
            string temp1 = ubahKartu(susunan[0]); string temp2 =
            ubahKartu(susunan[1]); string temp3 = ubahKartu(susunan[2]); string temp4 =
            ubahKartu(susunan[3]); string temp5(1, susunanop[0]); string temp6(1,
            susunanop[1]); string temp7(1, susunanop[2]);
            string solusi = "(" + temp1 + " " + temp5 + " ( " +
            temp2 + " " + temp6 + " " + temp3 + " )) " + temp7 + " " + temp4;
            int angka1 = ubahNilai(susunan[0]);
            int angka2 = ubahNilai(susunan[1]);
            int angka3 = ubahNilai(susunan[2]);
            int angka4 = ubahNilai(susunan[3]);
            double hasil = Operasi(1.0*angka2, 1.0*angka3,
            susunanop[1]);
            if (hasil == 0 && susunanop[0] == '/') {
                continue;
            }
            double hasil2 = Operasi(1.0*angka1, hasil,
            susunanop[0]);

```



```

        double hasilakhir = Operasi(hasil2, 1.0*angka4,
susunanop[2]);
        if (hasilakhir == 24.0 && !isInVector(seluruhSolusi,
solusi)) {
            seluruhSolusi.push_back(solusi);
        }
    }
}

// Print semua solusi
if (lengthvector(seluruhSolusi) == 0) {
    cout << "Tidak ada solusi untuk kombinasi kartu ini" << endl;
    clock_t end = clock();
    cout << "Waktu eksekusi program adalah " << end-start << "
milliseconds" << endl;
}
else {
    cout << "Berikut adalah seluruh solusi untuk kombinasi kartu ini"
<< endl;
    for (string i : seluruhSolusi) {
        cout << i << endl;
    }
    cout << "Ada " << lengthvector(seluruhSolusi) << " untuk kombinasi
kartu ini" << endl;
    clock_t end = clock();
    cout << "Waktu eksekusi program adalah " << end-start << "
milliseconds" << endl;

    // Bagian pemrosesan save file
    string inginsave = "";
    cout << "Apakah ingin menyimpan solusi ke dalam sebuah file txt?
(ketik y atau Y jika ya dan n atau N jika tidak)" << endl;
    cin >> inginsave;
    while (inginsave != "y" && inginsave != "Y" && inginsave != "N" &&
inginsave != "n") {
        cout << "Input tidak valid silahkan input y atau Y apabila
ingin menyimpan solusi ke dalam sebuah file txt dan n atau N jika tidak" <<
endl;
        cin >> inginsave;
    }
    if (inginsave == "y" || inginsave == "Y") {
        cout << "Input nama file tanpa menggunakan .txt di nama file"
<< endl;

        string namafile;
        cin >> namafile;
        namafile += ".txt";
    }
}

```

```

        ofstream filesave;
        filesave.open(namafile.c_str());
        for (string i : seluruhSolusi) {
            filesave << i << "\n";
        }
        filesave.close();
    }
}

    cout << "Apakah masih ingin menggunakan program ini jika ya ketik y
dan jika tidak ketik n" << endl;
    cin >> lanjut;
    if (lanjut == "n" || lanjut == "N") {
        cout << "Terima kasih telah menggunakan program 24 Solver ini" <<
endl;
    }
    else if (lanjut == "y" || lanjut == "Y") {
        makeEmpty(&number);
    }
}
}

```

BAB 4

EKSPERIMEN

- Apabila input tidak valid



Gambar 4.1 Apabila Input lebih dari 4 angka



Gambar 4.2 Apabila input bukan kartu yang sesuai

- Apabila input valid dan memiliki solusi

24 Solver

Selamat datang di program 24 Solver

Berikut beberapa pilihan yang terdapat pada program ini :

1. Nilai-Nilai Kartu akan digenerate secara random

2. Nilai-nilai kartu di input sendiri

Silahkan input pilihan Anda : 2

Silahkan Masukkan Nilai Kartu : A 8 9 Q

Berikut adalah seluruh solusi untuk kombinasi kartu ini

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

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

Ada 48 untuk kombinasi kartu ini

Waktu eksekusi program adalah 116 milliseconds

Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)

s

Gambar 4.3 Apabila input valid dan memiliki solusi

- Apabila input valid dan memiliki solusi, serta user ingin menyimpan solusi di file

```

Ada 48 untuk kombinasi kartu ini
Waktu eksekusi program adalah 116 milliseconds
Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)
y
Input nama file tanpa menggunakan .txt di nama file
test1
Apakah masih ingin menggunakan program ini jika ya ketik y dan jika tidak ketik n

```

Gambar 4.4 Program ketika solusi ada menanyakan user apakah ingin melakukan save file

```

test > test1.txt
1  (( 1 - 8 ) + 9 ) * 12
2  ( 1 - ( 8 - 9 ) ) * 12
3  ( 1 * 8 ) * ( 12 - 9 )
4  1 * ( 8 * ( 12 - 9 ) )
5  (( 1 + 9 ) - 8 ) * 12
6  ( 1 + ( 9 - 8 ) ) * 12
7  1 * (( 12 - 9 ) * 8 )
8  (( 1 * 12 ) - 9 ) * 8
9  ( 1 * ( 12 - 9 ) ) * 8
10 ( 8 * 1 ) * ( 12 - 9 )
11 ( 8 / 1 ) * ( 12 - 9 )
12 8 * (( 1 * 12 ) - 9 )
13 8 * ( 1 * ( 12 - 9 ) )
14 8 / ( 1 / ( 12 - 9 ) )
15 8 * (( 12 * 1 ) - 9 )
16 8 * (( 12 / 1 ) - 9 )
17 8 * ( 12 - ( 1 * 9 ) )
18 8 * (( 12 - 9 ) * 1 )
19 8 * (( 12 - 9 ) / 1 )
20 8 * ( 12 - ( 9 * 1 ) )
21 8 * ( 12 - ( 9 / 1 ) )
22 ( 8 * ( 12 - 9 ) ) * 1
23 ( 8 * ( 12 - 9 ) ) / 1
24 (( 9 + 1 ) - 8 ) * 12
25 ( 9 + ( 1 - 8 ) ) * 12
26 (( 9 - 8 ) + 1 ) * 12
27 ( 9 - ( 8 - 1 ) ) * 12
28 12 * (( 1 - 8 ) + 9 )
29 12 * ( 1 - ( 8 - 9 ) )
30 12 * (( 1 + 9 ) - 8 )
31 12 * ( 1 + ( 9 - 8 ) )
32 (( 12 * 1 ) - 9 ) * 8
33 (( 12 / 1 ) - 9 ) * 8
34 ( 12 - ( 1 * 9 ) ) * 8
35 ( 12 - 9 ) * ( 1 * 8 )
36 ( 12 - 9 ) / ( 1 / 8 )
37 12 * (( 9 + 1 ) - 8 )
38 12 * ( 9 + ( 1 - 8 ) )
39 (( 12 - 9 ) * 1 ) * 8
40 (( 12 - 9 ) / 1 ) * 8
41 ( 12 - ( 9 * 1 ) ) * 8
42 ( 12 - ( 9 / 1 ) ) * 8
43 ( 12 - 9 ) * ( 8 * 1 )
44 ( 12 - 9 ) * ( 8 / 1 )
45 12 * (( 9 - 8 ) + 1 )
46 12 * ( 9 - ( 8 - 1 ) )
47 (( 12 - 9 ) * 8 ) * 1
48 (( 12 - 9 ) * 8 ) / 1

```

Gambar 4.5 File test1 ketika user melakukan save ke file test1

- Apabila input tidak memiliki solusi

```

Selamat datang di program 24 Solver
Berikut beberapa pilihan yang terdapat pada program ini :
1. Nilai-Nilai Kartu akan digenerate secara random
2. Nilai-nilai kartu di input sendiri
Silahkan input pilihan Anda : 2
Silahkan Masukkan Nilai Kartu : K K K K
Tidak ada solusi untuk kombinasi kartu ini
Waktu eksekusi program adalah 64 milliseconds
Apakah masih ingin menggunakan program ini jika ya ketik y dan jika tidak ketik n

```

Gambar 4.6 Apabila input kartu tidak memiliki solusi

- Apabila user ingin meng-generate angka secara random

```

Selamat datang di program 24 Solver
Berikut beberapa pilihan yang terdapat pada program ini :
1. Nilai-Nilai Kartu akan digenerate secara random
2. Nilai-nilai kartu di input sendiri
Silahkan input pilihan Anda : 1
Berikut bilangan-bilangan yang didapat :
9 1 13 5
Berikut adalah seluruh solusi untuk kombinasi kartu ini
( 1 + 5 ) * ( 13 - 9 )
( 13 - 9 ) * ( 1 + 5 )
( 13 - 9 ) * ( 5 + 1 )
( 5 + 1 ) * ( 13 - 9 )
Ada 4 untuk kombinasi kartu ini
Waktu eksekusi program adalah 88 milliseconds
Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)

```

Gambar 4.7 Tampilan apabila user ingin meng-generate angka secara random

- Test2.txt

```
1. Nilai-Nilai Kartu akan digenerate secara random
2. Nilai-nilai kartu di input sendiri
Silahkan input pilihan Anda : 2
Silahkan Masukkan Nilai Kartu : 7 5 8 3
Berikut adalah seluruh solusi untuk kombinasi kartu ini
( 7 * 5 ) - ( 8 + 3 )
(( 7 * 5 ) - 8 ) - 3
( 7 * 5 ) - ( 3 + 8 )
(( 7 * 5 ) - 3 ) - 8
( 7 * ( 8 - 5 ) ) + 3
( 7 * 3 ) - ( 5 - 8 )
(( 7 * 3 ) - 5 ) + 8
( 7 * 3 ) + ( 8 - 5 )
(( 7 * 3 ) + 8 ) - 5
( 5 * 7 ) - ( 8 + 3 )
(( 5 * 7 ) - 8 ) - 3
( 5 * 7 ) - ( 3 + 8 )
(( 5 * 7 ) - 3 ) - 8
8 + (( 7 * 3 ) - 5 )
( 8 + ( 7 * 3 ) ) - 5
( 8 - 5 ) + ( 7 * 3 )
8 - ( 5 - ( 7 * 3 ) )
(( 8 - 5 ) * 7 ) + 3
( 8 - 5 ) + ( 3 * 7 )
8 - ( 5 - ( 3 * 7 ) )
8 + (( 3 * 7 ) - 5 )
( 8 + ( 3 * 7 ) ) - 5
( 3 * 7 ) - ( 5 - 8 )
3 - ( 7 * ( 5 - 8 ) )
(( 3 * 7 ) - 5 ) + 8
( 3 * 7 ) + ( 8 - 5 )
3 + ( 7 * ( 8 - 5 ) )
(( 3 * 7 ) + 8 ) - 5
3 - (( 5 - 8 ) * 7 )
3 + (( 8 - 5 ) * 7 )
Ada 30 untuk kombinasi kartu ini
Waktu eksekusi program adalah 88 milliseconds
Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)
y
Input nama file tanpa menggunakan .txt di nama file
test2
Apakah masih ingin menggunakan program ini jika ya ketik y dan jika tidak ketik n
```

```
src > test2.txt
1  ( 7 * 5 ) - ( 8 + 3 )
2  (( 7 * 5 ) - 8 ) - 3
3  ( 7 * 5 ) - ( 3 + 8 )
4  (( 7 * 5 ) - 3 ) - 8
5  ( 7 * ( 8 - 5 ) ) + 3
6  ( 7 * 3 ) - ( 5 - 8 )
7  (( 7 * 3 ) - 5 ) + 8
8  ( 7 * 3 ) + ( 8 - 5 )
9  (( 7 * 3 ) + 8 ) - 5
10 ( 5 * 7 ) - ( 8 + 3 )
11 (( 5 * 7 ) - 8 ) - 3
12 ( 5 * 7 ) - ( 3 + 8 )
13 (( 5 * 7 ) - 3 ) - 8
14 8 + (( 7 * 3 ) - 5 )
15 ( 8 + ( 7 * 3 ) ) - 5
16 ( 8 - 5 ) + ( 7 * 3 )
17 8 - ( 5 - ( 7 * 3 ) )
18 (( 8 - 5 ) * 7 ) + 3
19 ( 8 - 5 ) + ( 3 * 7 )
20 8 - ( 5 - ( 3 * 7 ) )
21 8 + (( 3 * 7 ) - 5 )
22 ( 8 + ( 3 * 7 ) ) - 5
23 ( 3 * 7 ) - ( 5 - 8 )
24 3 - ( 7 * ( 5 - 8 ) )
25 (( 3 * 7 ) - 5 ) + 8
26 ( 3 * 7 ) + ( 8 - 5 )
27 3 + ( 7 * ( 8 - 5 ) )
28 (( 3 * 7 ) + 8 ) - 5
29 3 - (( 5 - 8 ) * 7 )
30 3 + (( 8 - 5 ) * 7 )
31
```

Gambar 4.8 Input dan Output pada terminal serta isi file test2.txt


```

24 Solver

Selamat datang di program 24 Solver
Berikut beberapa pilihan yang terdapat pada program ini :
1. Nilai-Nilai Kartu akan digenerate secara random
2. Nilai-nilai kartu di input sendiri
Silahkan input pilihan Anda : 2
Silahkan Masukkan Nilai Kartu : 6 6 6 6
Berikut adalah seluruh solusi untuk kombinasi kartu ini
( 6 + 6 ) + ( 6 + 6 )
( 6 * 6 ) - ( 6 + 6 )
6 + (( 6 + 6 ) + 6 )
6 + ( 6 + ( 6 + 6 ))
(( 6 + 6 ) + 6 ) + 6
(( 6 * 6 ) - 6 ) - 6
( 6 + ( 6 + 6 )) + 6
Ada 7 untuk kombinasi kartu ini
Waktu eksekusi program adalah 152 milliseconds
Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)
y
Input nama file tanpa menggunakan .txt di nama file
test3
Apakah masih ingin menggunakan program ini jika ya ketik y dan jika tidak ketik n

```

```

1  ( 6 + 6 ) + ( 6 + 6 )
2  ( 6 * 6 ) - ( 6 + 6 )
3  6 + (( 6 + 6 ) + 6 )
4  6 + ( 6 + ( 6 + 6 ))
5  (( 6 + 6 ) + 6 ) + 6
6  (( 6 * 6 ) - 6 ) - 6
7  ( 6 + ( 6 + 6 )) + 6
8

```

Gambar 4.9 Tampilan input dan output pada terminal dan isi file test3.txt

```

Selamat datang di program 24 Solver
Berikut beberapa pilihan yang terdapat pada program ini :
1. Nilai-Nilai Kartu akan digenerate secara random
2. Nilai-nilai kartu di input sendiri
Silahkan input pilihan Anda : 2
Silahkan Masukkan Nilai Kartu : 5 3 5 2
Berikut adalah seluruh solusi untuk kombinasi kartu ini
( 5 + 3 ) * ( 5 - 2 )
( 5 * 5 ) - ( 3 - 2 )
(( 5 * 5 ) - 3 ) + 2
( 5 * 5 ) + ( 2 - 3 )
(( 5 + 5 ) - 2 ) * 3
(( 5 * 5 ) + 2 ) - 3
( 5 + ( 5 - 2 ) ) * 3
( 5 - 2 ) * ( 3 + 5 )
( 5 - 2 ) * ( 5 + 3 )
(( 5 - 2 ) + 5 ) * 3
( 5 - ( 2 - 5 ) ) * 3
( 3 + 5 ) * ( 5 - 2 )
3 * (( 5 + 5 ) - 2 )
3 * ( 5 + ( 5 - 2 ) )
3 * (( 5 - 2 ) + 5 )
3 * ( 5 - ( 2 - 5 ) )
2 + (( 5 * 5 ) - 3 )
( 2 + ( 5 * 5 ) ) - 3
( 2 - 3 ) + ( 5 * 5 )
2 - ( 3 - ( 5 * 5 ) )
Ada 20 untuk kombinasi kartu ini
Waktu eksekusi program adalah 72 milliseconds
Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)
y
Input nama file tanpa menggunakan .txt di nama file
test4
Apakah masih ingin menggunakan program ini jika ya ketik y dan jika tidak ketik n

```

```

( 5 + 3 ) * ( 5 - 2 )
( 5 * 5 ) - ( 3 - 2 )
(( 5 * 5 ) - 3 ) + 2
( 5 * 5 ) + ( 2 - 3 )
(( 5 + 5 ) - 2 ) * 3
(( 5 * 5 ) + 2 ) - 3
( 5 + ( 5 - 2 ) ) * 3
( 5 - 2 ) * ( 3 + 5 )
( 5 - 2 ) * ( 5 + 3 )
(( 5 - 2 ) + 5 ) * 3
( 5 - ( 2 - 5 ) ) * 3
( 3 + 5 ) * ( 5 - 2 )
3 * (( 5 + 5 ) - 2 )
3 * ( 5 + ( 5 - 2 ) )
3 * (( 5 - 2 ) + 5 )
3 * ( 5 - ( 2 - 5 ) )
2 + (( 5 * 5 ) - 3 )
( 2 + ( 5 * 5 ) ) - 3
( 2 - 3 ) + ( 5 * 5 )
2 - ( 3 - ( 5 * 5 ) )

```

Gambar 4.10 Tampilan input dan output pada terminal dan isi file test4.txt

```

Selamat datang di program 24 Solver
Berikut beberapa pilihan yang terdapat pada program ini :
1. Nilai-Nilai Kartu akan digenerate secara random
2. Nilai-nilai kartu di input sendiri
Silahkan input pilihan Anda : 2
Silahkan Masukkan Nilai Kartu : 7 3 8 1
Berikut adalah seluruh solusi untuk kombinasi kartu ini
(( 7 - 3 ) - 1 ) * 8
( 7 - ( 3 + 1 ) ) * 8
(( 7 - 1 ) - 3 ) * 8
( 7 - ( 1 + 3 ) ) * 8
3 / ( 1 - ( 7 / 8 ) )
8 * (( 7 - 3 ) - 1 )
8 * ( 7 - ( 3 + 1 ) )
8 * (( 7 - 1 ) - 3 )
8 * ( 7 - ( 1 + 3 ) )
Ada 9 untuk kombinasi kartu ini
Waktu eksekusi program adalah 106 milliseconds
Apakah ingin menyimpan solusi ke dalam sebuah file txt? (ketik y atau Y jika ya dan n atau N jika tidak)
y
Input nama file tanpa menggunakan .txt di nama file
test5
Apakah masih ingin menggunakan program ini jika ya ketik y dan jika tidak ketik n

```

```

src > test5.txt
1 (( 7 - 3 ) - 1 ) * 8
2 ( 7 - ( 3 + 1 ) ) * 8
3 (( 7 - 1 ) - 3 ) * 8
4 ( 7 - ( 1 + 3 ) ) * 8
5 3 / ( 1 - ( 7 / 8 ) )
6 8 * (( 7 - 3 ) - 1 )
7 8 * ( 7 - ( 3 + 1 ) )
8 8 * (( 7 - 1 ) - 3 )
9 8 * ( 7 - ( 1 + 3 ) )
10

```

Gambar 4.11 Tampilan input dan output pada terminal serta isi file test5.txt

REFERENSI

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tucil1-Stima-2023.pdf>

LAMPIRAN

Checklist Fitur

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