

# **LAPORAN TUGAS KECIL 1**

## **IF2211 STRATEGI ALGORITMA**

### **Penyelesaian Permainan Kartu 24 dengan *Algoritma Brute Force***

Dosen Pengampu : Dr. Nur Ulfa Maulidevi, S.T, M.Sc.



Oleh :

Satria Octavianus Nababan / 13521168

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2022/2023**

## DAFTAR ISI

<b>BAB 1 .....</b>	<b>3</b>
<b>DESKRIPSI PROGRAM .....</b>	<b>3</b>
1.1 Deskripsi Umum .....	3
1.2 Algoritma <i>Brute Force</i> .....	3
1.3 Logika Pengerjaan Program .....	4
1.4 Alur Pengerjaan Program .....	5
<b>BAB II .....</b>	<b>6</b>
<b><i>SOURCE CODE PROGRAM</i> .....</b>	<b>6</b>
2.1 Asumsi Pada Program .....	6
2.2 Souce Code Program .....	6
2.3 Tabel Kompetensi .....	17
<b>BAB III .....</b>	<b>17</b>
<b>EKSPERIMEN .....</b>	<b>17</b>
<b>Link repository .....</b>	<b>22</b>
<b>DAFTAR PUSTAKA .....</b>	<b>23</b>

# BAB 1

## DESKRIPSI PROGRAM

### 1.1 Deskripsi Umum

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.

### 1.2 Algoritma *Brute Force*

*Brute Force* merupakan pendekatan yang lempang (*straight-forward*) dalam hal pemecahan suatu masalah atau persoalan dengan sangat sederhana (simple), langsung, dan jelas (*obvious way*). Algoritma *Brute Force* sering kali disebut juga sebagai algoritma naif (*naive algorithm*).

Karakteristik algoritma brute force umumnya tidak “pintar” dan tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya, sehingga terkadang algoritma brute force disebut juga algoritma naif (*naïve algorithm*). Algoritma brute force seringkali merupakan pilihan yang kurang disukai karena ketidakmangkusannya itu, tetapi dengan mencari polapola yang mendasar, keteraturan, atau trik-trik khusus, biasanya akan membantu kita menemukan algoritma yang lebih cerdas dan lebih mangkus. Untuk masalah yang

ukurannya kecil, kesederhanaan brute force biasanya lebih diperhitungkan daripada ketidakmangkusannya. Algoritma brute force sering digunakan sebagai basis bila membandingkan beberapa alternatif algoritma yang mangkus. Algoritma brute force seringkali lebih mudah diimplementasikan daripada algoritma yang lebih canggih, dan karena kesederhanaannya, kadang-kadang algoritma brute force dapat lebih mangkus (ditinjau dari segi implementasi).

### 1.3 Logika Pengerjaan Program

Terdapat banyak kemungkinan cara untuk menentukan operasi mendapatkan hasil 24. Saya akan membagi kemungkinan tersebut kedalam 3 bagian :

#### 1. Susunan angka-angka dalam operasi

Dalam permainan 24, jumlah angka yang dapat digunakan dalam operasi haruslah 4. Maka dari itu, dengan menggunakan permutasi, terdapat  $4! = 24$  kemungkinan susunan angka yang mungkin menghasilkan angka 24. Namun, ada kemungkinan terdapat duplikat.

#### 2. Susunan operator-operator dalam operasi

Terdapat 4 buah operator yang dapat digunakan dalam permainan 24, yaitu tambah, kurang, kali, dan bagi. Karena terdapat 4 buah angka dalam permainan ini, maka terdapat 3 tempat di mana operator harus dimasukkan untuk menghasilkan operasi aritmatika, yaitu 1 di antara 2 angka.

Karena terdapat 4 operator dan 3 buah tempat yang tersedia, dengan aturan perkalian, maka kita mendapatkan  $4 \times 4 \times 4 = 64$  buah kemungkinan susunan operator yang mungkin menghasilkan angka 24.

Namun, jika terdapat angka yang sama dari keempat angka tersebut, maka akan terdapat solusi yang sama dari solusi-solusi yang dihasilkan.

#### 3. Susunan penempatan tanda kurung dalam operasi

Penempatan tanda kurung dalam operasi aritmatika berfungsi untuk menentukan urutan operasi mana yang dikerjakan terlebih dahulu. Dari penempatan tanda kurung tersebut, terdapat 5 buah jenis solusi yang ada.

Jika dimisalkan salah satu susunan bilangan tersebut adalah a, b, c, dan d serta operasi yang digunakan adalah 3 operasi penjumlahan, maka susunan tanda kurung tersebut adalah sebagai berikut.

1.  $a + (b + (c + d))$
2.  $a + ((b + c) + d)$
3.  $(a + b) + (c + d)$
4.  $(a + (b + c)) + d$
5.  $((a + b) + c) + d$

Dari ketiga poin di atas, maka dapat disimpulkan bahwa jumlah seluruh kemungkinan solusi dari program yang dibuat adalah  $24 \times 64 \times 5 = 7.680$  buah kemungkinan solusi. Namun, hasilnya akan kurang dari 7.680 karena beberapa solusi tidak menghasilkan angka 24.

#### 1.4 Alur Pengerjaan Program

1. Pertama-tama, program akan meminta user untuk memilih teknis permainan, apakah ingin menginput sendiri 4 kartunya atau bermain dengan kartu yang dirandom oleh program.
2. Jika user bermain dengan menginput sendiri kartunya, maka keempat angka tersebut dibaca dengan menggunakan prosedur `readCards`, yang hanya berfungsi menerima input. Input yang dimasukkan haruslah masukan yang valid yaitu (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, dan K) yang dipisahkan oleh spasi.
3. Setelah mendapatkan 4 kartu dari user, maka dibuatlah kemungkinan susunan kartu sebanyak  $4! = 24$  kemungkinan yang dimasukkan ke dalam array. Hal ini dilakukan dengan prosedur `generateArray`, namun sebelum dimasukkan ke array, prosedur `generateArray` mengecek terlebih dahulu jika susunan angka tersebut telah ada sebelumnya atau belum untuk menghindari duplikat.
4. Setelah mendapatkan kemungkinan susunan yang telah dimasukkan ke array kemudian beralih ke prosedur `check24`. Pada prosedur `check24` ini membuat setiap kemungkinan operator-operator aritmatika yang mungkin ditempatkan di antara susunan 4 angka tersebut. Jumlah kemungkinan yang mungkin untuk setiap susunan angka tersebut adalah  $4^3$  kemungkinan.
5. Kemungkinan yang didapatkan kemudian dihitung menggunakan prosedur `countResult`, pada prosedur tersebut setiap angka dan operator dioperasikan dengan 5 kemungkinan susunan tanda kurung. Tanda kurung merepresentasikan operasi yang dikerjakan lebih dulu yang didalamnya

terdapat 2 angka dan 1 operator. Selanjutnya, fungsi calculate menghitung operasi yang terdapat didalam tanda kurung.

6. Selanjutnya, semua kemungkinan hasil yang didapat dicek dan jika memberikan hasil 24, maka akan dimasukkan ke variabel result dan ditampilkan sebagai output. Program juga akan menampilkan jumlah seluruh hasil yang sesuai beserta dengan lamanya waktu eksekusi program.
7. Setelahnya, user akan diberikan pilihan, apakah akan menyimpan hasil yang didapat kedalam file .txt atau tidak.

## BAB II

### *SOURCE CODE PROGRAM*

#### 2.1 Asumsi Pada Program

Terdapat beberapa asumsi yang ada pada program sesuai dengan spesifikasi yang diberikan :

1. User memasukkan input yang dipisahkan oleh spasi.
2. Input yang diberikan terdiri atas 4 masukan dan harus memberikan input yang valid yaitu (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, dan K).
3. Karena terdapat bilangan pecahan yang hasilnya harus dibulatkan oleh tipe data float, hasil perhitungan terkadang tidak pas 24, sehingga dibuat galat sebesar 0,00001 dari bilangan 24.

#### 2.2 Souce Code Program

Bahasa yang digunakan dalam membuat program ini adalah bahasa C++ yang source codenya dapat dilihat, sebagai berikut :

```
/* Nama      : Satria Octavianus Nababan
   NIM       : 13521168
   Mata Kuliah : STRATEGI ALGORITMA
   Kelas     : 02 */

#include <iostream> // Input dan output
#include <time.h>   // Menghitung lama waktu eksekusi
#include <fstream>  // save File
#include <vector>
```

```

using namespace std;

// KAMUS GLOBAL
string a, b, c, d;
int count = 0, pilKartu;
bool saveFile = false, flag = false;
string inputPil;
string namaFile;
vector<string> resultlist;
float waktu;

// Float ini berfungsi menyimpan hasil kemungkinan perhitungan
float result;

int e[22], f[25], g[25], h[25];
int Neff = 0;

// DEKLARASI FUNGSI DAN PROSEDUR
void generateArray(int a, int b, int c, int d);
// Mencari tahu apakah susunan angka a, b, c, dan d terdapat dalam array

void readCards();
// Membaca input 4 kartu

void check24 (int w, int x, int y, int z);
// Mengecek apakah hasilnya mendapatkan angka 24 atau tidak

void countResult (int w, int x, int y, int z, int i, int j, int k);
// Menghitung hasil dari kombinasi input dan operator yang direpresentasikan oleh
angka
// w, x, y, z adalah keempat masukan yang diberikan user
// i, j, k adalah operator-operator yang dihasilkan

float calculate (float a, float b, int ops);
// Menghitung operasi dengan 2 buah float dan 1 operator yang diwakili angka

char sign (int ops);
// Mengembalikan char operator yang direpresentasikan oleh angka

// MAIN PROGRAM
int main()
{
    // KAMUS LOKAL
    int i;

```

```

// ALGORITMA
cout << "

      " << endl;
cout << " \\_____ \\ / | | / _____/_____ _____ / _____/
_____ | |_____ " << endl;
cout << " / _____// | |_ / \\ _____\\_ \\ / \\_/_ /
\\ \\_____ \\ / _ \\| |\\ \\ \\ / \\_ \\_ \\ " << endl;
cout << "/ \\_ \\ ^ / \\ \\_\\ \\ \\_ \\_ \\| Y
Y \\ \\_ / / ( <_> ) |_\\_ /\\_ _/ | \\_/" << endl;
cout << "\\_____ \\_____ | \\_____ (____ /_ |_ /\\_ >
/_____ /\\_/_/|_/_/\\_/_/ \\_ >_ | " << endl;
cout <<
"      \\_ \\ |_ | \\_ \\ \\_ \\ \\_ \\ \\_ \\
      \\_ \\ " << endl;

cout << "By Satria Octavianus Nababan - 13521168" << endl;

cout << "\nPilih sistem permainan:\n      (1) Input nilai kartu sendiri\n      (2)
Pilih kartu secara acak\nInput answer : ";
while (!flag)
{ // Validasi masukkan user
    cin >> pilKartu;
    if (pilKartu == 1 or pilKartu == 2)
        flag = true;
    else
        cout << "\nMasukkan tidak sesuai! Ulangi (1/2)\nInput answer : ";
}

if (pilKartu == 2)
{ // Mengacak 4 kartu
    string number[13] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"J", "Q", "K"};
    cout << "Kartu :";
    srand(time(0));
    a = to_string(rand()%13+1);
    b = to_string(rand()%13+1);
    c = to_string(rand()%13+1);
    d = to_string(rand()%13+1);
    string e,f,g,h;
    e = a;
    f = b;
    g = c;
    h = d;
    if (a=="1"){

```



```
e = "A";
}
if (a=="11"){
    e = "J";
}
if (a=="12"){
    e = "Q";
}
if (a=="13"){
    e = "K";
}
if (b=="1"){
    f = "A";
}
if (b=="11"){
    f = "J";
}
if (b=="12"){
    f = "Q";
}
if (b=="13"){
    f = "K";
}
if (c=="1"){
    g = "A";
}
if (c=="11"){
    g = "J";
}
if (c=="12"){
    g = "Q";
}
if (c=="13"){
    g = "K";
}
if (d=="1"){
    h = "A";
}
if (d=="11"){
    h = "J";
}
if (d=="12"){
    h = "Q";
}
if (d=="13"){
```

```

        h = "K";
    }
    cout << " " << e << " " << f << " " << g << " " << h;

    cout << "\n";
}

if (pilKartu == 1)
{
    // input 4 kartu
    readCards();
}

// Mencari solusi
cout << "Mencari solusi..." << endl;
clock_t start = clock();

// Men-generate semua kemungkinan sususnan angka, terdapat 4! = 24
kemungkinan
generateArray(std::stof(a), std::stof(b), std::stof(c), std::stof(d));
generateArray(std::stof(a), std::stof(b), std::stof(d), std::stof(c));
generateArray(std::stof(a), std::stof(c), std::stof(b), std::stof(d));
generateArray(std::stof(a), std::stof(c), std::stof(d), std::stof(b));
generateArray(std::stof(a), std::stof(d), std::stof(b), std::stof(c));
generateArray(std::stof(a), std::stof(d), std::stof(c), std::stof(b));
generateArray(std::stof(b), std::stof(a), std::stof(c), std::stof(d));
generateArray(std::stof(b), std::stof(a), std::stof(d), std::stof(c));
generateArray(std::stof(b), std::stof(c), std::stof(a), std::stof(d));
generateArray(std::stof(b), std::stof(c), std::stof(d), std::stof(a));
generateArray(std::stof(b), std::stof(d), std::stof(a), std::stof(c));
generateArray(std::stof(b), std::stof(d), std::stof(c), std::stof(a));
generateArray(std::stof(c), std::stof(b), std::stof(b), std::stof(d));
generateArray(std::stof(c), std::stof(a), std::stof(d), std::stof(b));
generateArray(std::stof(c), std::stof(b), std::stof(a), std::stof(d));
generateArray(std::stof(c), std::stof(b), std::stof(d), std::stof(a));
generateArray(std::stof(c), std::stof(d), std::stof(a), std::stof(b));
generateArray(std::stof(c), std::stof(d), std::stof(b), std::stof(a));
generateArray(std::stof(d), std::stof(a), std::stof(b), std::stof(c));
generateArray(std::stof(d), std::stof(a), std::stof(c), std::stof(b));
generateArray(std::stof(d), std::stof(b), std::stof(a), std::stof(c));
generateArray(std::stof(d), std::stof(b), std::stof(c), std::stof(a));
generateArray(std::stof(d), std::stof(c), std::stof(a), std::stof(b));
generateArray(std::stof(d), std::stof(c), std::stof(b), std::stof(a));

for (i = 1; i <= Neff; i++)
{

```

```

        check24(e[i], f[i], g[i], h[i]);
    }
    cout << endl << "Terdapat " << count << " buah solusi" << endl;

    // Print lama waktu eksekusi
    waktu = (float)(clock() - start)/CLOCKS_PER_SEC;
    printf("Lama Waktu eksekusi: %.5fs\n", (float)(clock() -
start)/CLOCKS_PER_SEC);

    // simpan solusi ke file .txt
    saveFile = false;
    cout << "\nApakah Anda ingin menyimpan hasil ? (Y/N)\nInput answer : ";
    while (!saveFile )
    {
        cin >> inputPil;
        if (inputPil == "Y" or inputPil == "y")
        {
            saveFile = true;
            cout << "Masukkan nama file : ";
            cin >> namaFile;
            ofstream file;
            file.open("../test/" + namaFile + ".txt");
            file << "Kartu : " << a << " " << b << " " << c << " " << d << "\n";
            if (count != 0)
            {
                file << count << " Solution Found!\n";
                for (auto i : resultlist)
                {
                    file << i << "\n";
                }
            }
            else
            {
                file << "No solution found\n";
            }
            file << "Lama Waktu eksekusi: " << waktu << "s";
            file.close();
            cout << "Berhasil menyimpan solusi di '../test/' + namaFile +
".txt'\n";
        }
        else if (inputPil == "N" or inputPil == "n")
            saveFile = true;
        else
        {

```

```

        cout << "\nMasukkan tidak sesuai! Ulangi (Y/N)\n Masukkan Pilihan :
";
    }
}
return 0;
}

void readCards()
// Membaca keempat angka untuk permainan 24
{
    bool valid = true;
    bool loop = true;
    while (loop){
        if (!valid) cout << "input tidak valid" << endl << "input yang valid yaitu (A,
2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)" << endl;
        cout << "Masukkan keempat kartu yang mungkin" << endl;
        cin >> a >> b >> c >> d;
        if (!(a=="A" || a=="K" || a=="Q" || a=="J" || (std::stof(a)>=2 &&
std::stof(a)<=10))){
            valid=false;
            continue;
        }
        if (!(b=="A" || b=="K" || b=="Q" || b=="J" || (std::stof(b)>=2 &&
std::stof(b)<=10))){
            valid=false;
            continue;
        }
        if (!(c=="A" || c=="K" || c=="Q" || c=="J" || (std::stof(c)>=2 &&
std::stof(c)<=10))){
            valid=false;
            continue;
        }
        if (!(d=="A" || d=="K" || d=="Q" || d=="J" || (std::stof(d)>=2 &&
std::stof(d)<=10))){
            valid=false;
            continue;
        }
        if (a=="A") a="1";
        if (a=="K") a="13";
        if (a=="Q") a="12";
        if (a=="J") a="11";

        if (b=="A") b="1";
        if (b=="K") b="13";
        if (b=="Q") b="12";

```

```

    if (b=="J") b ="11";

    if (c=="A") c="1";
    if (c=="K") c ="13";
    if (c == "Q") c="12";
    if (c=="J") c ="11";

    if (d=="A") d="1";
    if (d=="K") d ="13";
    if (d == "Q") d="12";
    if (d=="J") d ="11";

    cout << "Angka yang diproses adalah..." << endl;
    cout << a << ", " << b << ", " << c << ", dan " << d << endl << endl;
    loop = false;
}
}

void generateArray(int a, int b, int c, int d)
// Mencari tahu apakah susunan angka a, b, c, dan d terdapat dalam array
// Jika ada, maka angka tersebut tidak dimasukkan ke dalam array
// Jika tidak, maka angka tersebut dimasukkan ke dalam array dan Neff ditambah 1
{
    // Kamus lokal
    int i;
    bool found;

    // Algoritma
    found = false;
    i = 1;

    while (i <= Neff && not found)
    {
        if (a == e[i] && b == f[i] && c == g[i] && d == h[i])
        {
            found = true;
        }
        else
        {
            i++;
        }
    }
    if (not found)
    {
        Neff++;
    }
}

```

```

    e[Neff] = a;
    f[Neff] = b;
    g[Neff] = c;
    h[Neff] = d;
}
}

void check24 (int w, int x, int y, int z)
// Mengecek apakah hasilnya dapat mendapatkan 24 atau tidak
{
    // Operator
    int i, j, k;

    // Terdapat 3 buah operator yang membatasi keempat angka karena
    // di antara 2 angka, terdapat 1 operator
    for (i = 1; i <= 4; i++)
    {
        for (j = 1; j <= 4; j++)
        {
            for (k = 1; k <= 4; k++)
            {
                countResult (w, x, y, z, i, j, k);
            }
        }
    }
}

void countResult (int w, int x, int y, int z, int i, int j, int k)
// Menghitung hasil dari kombinasi angka dan operator yang direpresentasikan oleh
// angka
// w, x, y, z adalah keempat masukan dari user
// i, j, k adalah operator-operator yang ada
{
    // ((w + x) + y) + z
    result = calculate(calculate(calculate((float)w, (float)x, i), (float) y, j),
(float)z, k);
    if (result > 23.99999 && result < 24.00001)
    {
        count++;
        // string haha = "aa" "aaa"
        resultlist.push_back("(" + to_string(w) + " " + sign(i) + " " + to_string(x) +
") " + sign(j) + " " + to_string(y) + ") " + sign(k) + " " + to_string(z));
        cout << "(" << w << " " << sign(i) << " " << x << " " <<
            sign(j) << " " << y << " " << sign(k) << " " << z << endl;
    }
}

```

```

}
// (w + x) + (y + z)
result = calculate(calculate((float)w, (float)x, i), calculate((float) y,
(float) z, k), j);
if (result > 23.99999 && result < 24.00001)
{
    count++;
    resultlist.push_back("(" + to_string(w)+ " " + sign(i) + " " + to_string(x) +
") " + sign(j) + "(" + to_string(y)+ " " + sign(k) + " " + to_string(z) + ")");
    cout << "(" << w << " " << sign(i) << " " << x << ")" <<
        sign(j) << " (" << y << " " << sign(k) << " " << z << ")" << endl;
}
// (w + (x + y)) + z
result = calculate(calculate((float)w, calculate((float)x, (float)y, j), i),
(float) z, k);
if (result > 23.99999 && result < 24.00001)
{
    count++;
    resultlist.push_back("(" + to_string(w)+ " " + sign(i) + "(" + to_string(x) + " "
+ sign(j) + " " + to_string(y)+"))" + sign(k) + " " + to_string(z));
    cout << "(" << w << " " << sign(i) << " (" << x << " " << sign(j) <<
        " " << y << "))" << sign(k) << " " << z << endl;
}
// w + ((x + y) + z)
result = calculate((float)w, calculate(calculate((float)x, (float)y, j),
(float)z, k), i);
if (result > 23.99999 && result < 24.00001)
{
    count++;
    resultlist.push_back(to_string(w)+ " " + sign(i) + "(" + to_string(x) + " " +
sign(j) + " " + ")" + sign(k) + " " + to_string(z) + ")");
    cout << w << " " << sign(i) << " (" << x << " " << sign(j) << " " <<
        y << ")" << sign(k) << " " << z << ")" << endl;
}
// w + (x + (y + z))
result = calculate((float)w, calculate((float)x, calculate((float)y, (float)z,
k), j), i);
if (result > 23.99999 && result < 24.00001)
{
    count++;
    resultlist.push_back(to_string(w)+ " " + sign(i) + "(" + to_string(x) + " " +
sign(j) + "(" + to_string(y)+ " " + sign(k) + " " + to_string(z) + "))");
    cout << w << " " << sign(i) << " (" << x << " " << sign(j) << " (" <<
        y << " " << sign(k) << " " << z << "))" << endl;
}
}

```

```

}

float calculate (float a, float b, int ops)
// Menghitung operasi dengan 2 buah float dan 1 operator yang diwakili angka
{
    if (ops == 1)
    {
        return (a + b);
    }
    else if (ops == 2)
    {
        return (a - b);
    }
    else if (ops == 3)
    {
        return (a * b);
    }
    else if (ops == 4)
    {
        return ((float)a / (float)b);
    }
}

char sign (int ops)
// Mengembalikan char operator yang direpresentasikan oleh angka
{
    if (ops == 1)
    {
        return ('+');
    }
    if (ops == 2)
    {
        return ('-');
    }
    if (ops == 3)
    {
        return ('*');
    }
    if (ops == 4)
    {
        return ('/');
    }
}

```



## 2.3 Tabel Kompetensi

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	✓	

## BAB III

## EKSPERIMEN

NO

1

TEST CASE

```

PS D:\Semester 4\STIPA\Tucil1\Tucil1_13521168\src> cd "d:\Semester 4\STIPA\Tucil1\Tucil1_13521168\src"
PS D:\Semester 4\STIPA\Tucil1\Tucil1_13521168\src> & .\"main.exe"

      24
    24 24
  24 24 24
24 24 24 24

By Satria Octavianus Nababan - 13521168

Pilih sistem permainan:
(1) Input nilai kartu sendiri
(2) Pilih kartu secara acak

Input answer : 1
Masukkan keempat kartu yang mungkin
8 3 8 3

Angka yang diproses adalah...
8, 3, 8, dan 3

Mencari solusi...
8 / (3 - (8 / 3))
((8 + 3) - 3) * 3
(8 + (3 - 3)) * 3
((8 - 3) + 3) * 3
(8 - (3 - 3)) * 3
((8 * 3) + 3) - 3
(8 * 3) + (3 - 3)
8 * ((3 + 3) - 3)
8 * (3 + (3 - 3))
((8 * 3) - 3) + 3
8 * ((3 - 3) + 3)
(8 * 3) - (3 - 3)
8 * (3 - (3 - 3))
((8 * 3) * 3) / 3
(8 * 3) * (3 / 3)
(8 * (3 * 3)) / 3
8 * ((3 * 3) / 3)
8 * (3 * (3 / 3))

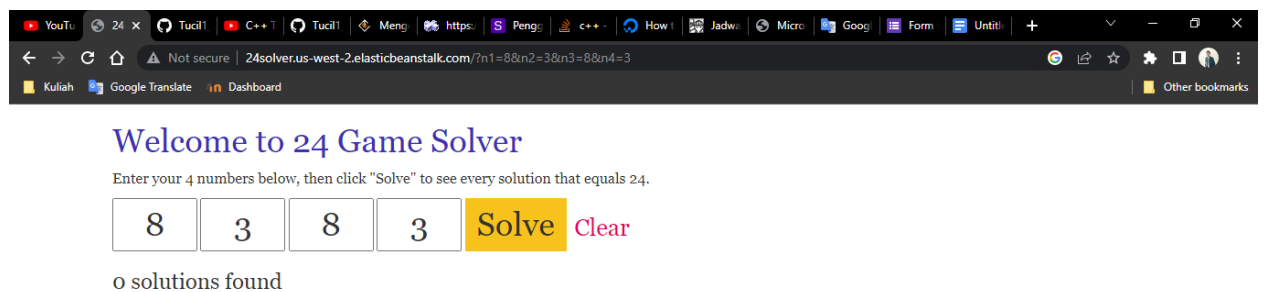
```

```
((8 * 3) / 3) * 3
(8 * (3 / 3)) * 3
8 * ((3 / 3) * 3)
(8 * 3) / (3 / 3)
8 * (3 / (3 / 3))
((8 / 3) * 3) * 3
(8 / 3) * (3 * 3)
(8 / (3 / 3)) * 3
8 / (3 / (3 * 3))
8 / ((3 / 3) / 3)

Terdapat 28 buah solusi
Lama waktu eksekusi: 0.14500s

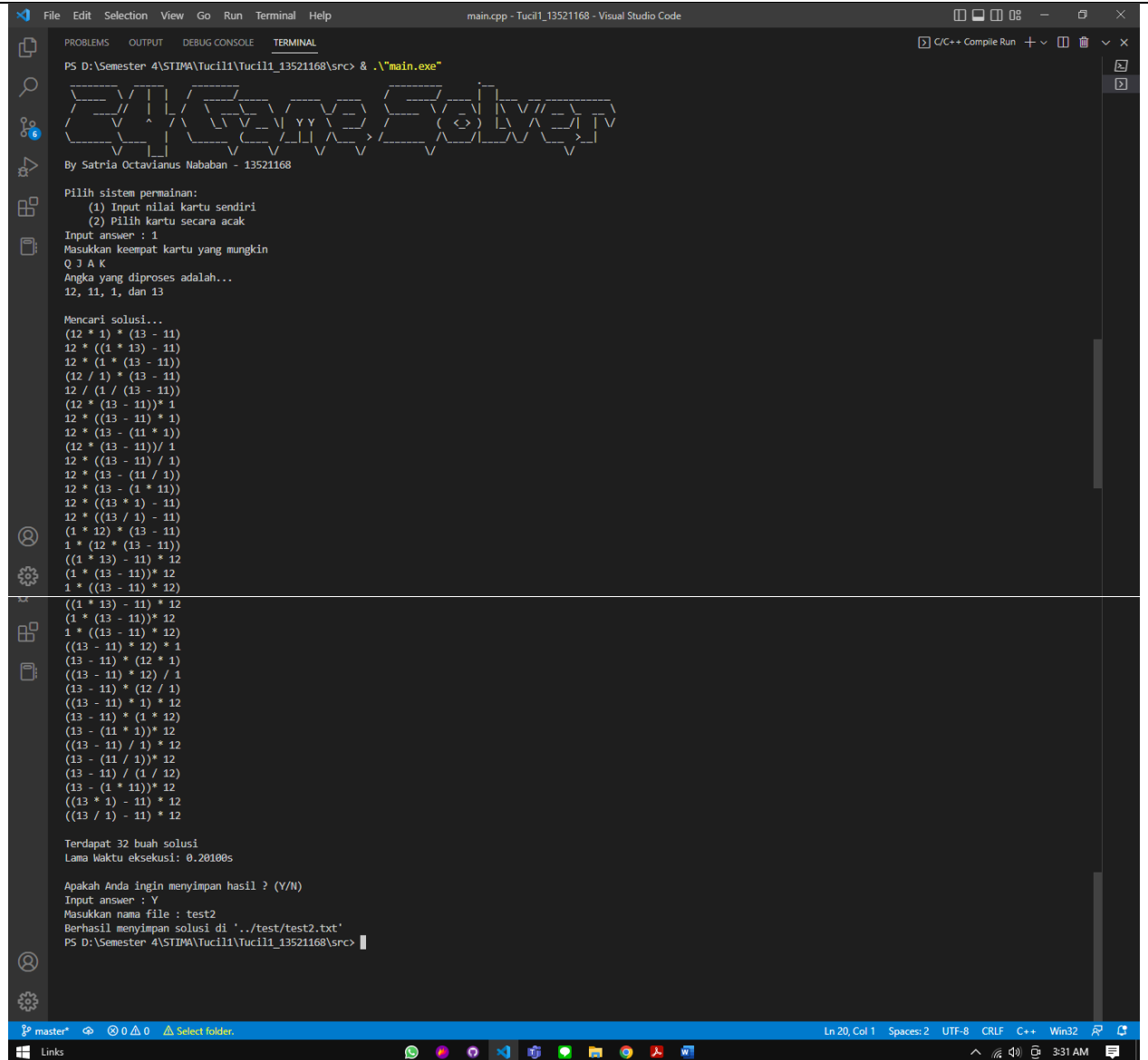
Apakah Anda ingin menyimpan hasil ? (Y/N)
Input answer : Y
Masukkan nama file : test1
Berhasil menyimpan solusi di '..\test\test1.txt'
PS D:\Semester 4\STIPA\Tucil11\Tucil11_13521168\src>
```

Gambar 1 memberikan 28 buah solusi untuk masukan 8,3,8,3 dengan lama waktu eksekusi program yaitu 0.14500s.



Disini saya mengambil salah satu contoh perbandingan solusi yang diambil dari <http://24solver.us-west-2.elasticbeanstalk.com/>. Pada gambar rujukan dapat dilihat bahwa untuk masukan 8,3,8,3 memberikan 0 buah solusi. Perbandingan ini dapat menjadi salah satu bukti bahwa di dalam beberapa ekspresi aritmatika yang dibentuk, terdapat pecahan, namun dalam perhitungan dengan komputer diubah menjadi decimal sehingga terdapat kehilangan keakuratan. Oleh karena itu, hasil yang seharusnya dicek dengan 24, dicek dengan range antara 23.9999999999 dan 24.0000000001.

2



```
File Edit Selection View Go Run Terminal Help
main.cpp - Tuc11_13521168 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\Semester 4\STIMA\Tuc11\Tuc11_13521168\src> & .\"main.exe"

By Satria Octavianus Nababan - 13521168

Pilih sistem permainan:
(1) Input nilai kartu sendiri
(2) Pilih kartu secara acak
Input answer : 1
Masukkan keempat kartu yang mungkin
Q J A K
Angka yang diproses adalah...
12, 11, 1, dan 13

Mencari solusi...
(12 * 1) * (13 - 11)
12 * ((1 * 13) - 11)
12 * (1 * (13 - 11))
(12 / 1) * (13 - 11)
12 / (1 / (13 - 11))
(12 * (13 - 11)) * 1
12 * ((13 - 11) * 1)
12 * (13 - (11 * 1))
(12 * (13 - 11)) / 1
12 * ((13 - 11) / 1)
12 * (13 - (11 / 1))
12 * (13 - (1 * 11))
12 * ((13 * 1) - 11)
12 * ((13 / 1) - 11)
(1 * 12) * (13 - 11)
1 * (12 * (13 - 11))
((1 * 13) - 11) * 12
(1 * (13 - 11)) * 12
1 * ((13 - 11) * 12)
((1 * 13) - 11) * 12
(1 * (13 - 11)) * 12
1 * ((13 - 11) * 12)
((13 - 11) * 12) * 1
(13 - 11) * (12 * 1)
((13 - 11) * 12) / 1
(13 - 11) * (12 / 1)
((13 - 11) * 1) * 12
(13 - 11) * (1 * 12)
(13 - (11 * 1)) * 12
((13 - 11) / 1) * 12
(13 - (11 / 1)) * 12
(13 - 11) / (1 / 12)
(13 - (1 * 11)) * 12
((13 * 1) - 11) * 12
((13 / 1) - 11) * 12

Terdapat 32 buah solusi
Lama Waktu eksekusi: 0.20100s

Apakah Anda ingin menyimpan hasil ? (Y/N)
Input answer : Y
Masukkan nama file : test2
Berhasil menyimpan solusi di '...\test\test2.txt'
PS D:\Semester 4\STIMA\Tuc11\Tuc11_13521168\src>
```

Pada test case 2, diberikan input Q J A K yang merupakan representasi dari angka 12, 11, 1, 13 yang memberikan 32 buah solusi dengan lama waktu 0.20100s.

3

```

File Edit Selection View Go Run Terminal Help
test3.txt - Tuci11_13521168 - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\Semester 4\STIMA\Tuci11_13521168\src> & .\main.exe"

By Satria Octavianus Nababan - 13521168

Pilih sistem permainan:
(1) Input nilai kartu sendiri
(2) Pilih kartu secara acak
Input answer : 1
Masukkan keempat kartu yang mungkin
2 4 5 13
input tidak valid
input yang valid yaitu (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
Masukkan keempat kartu yang mungkin
1 3 5 7
input tidak valid
input yang valid yaitu (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
Masukkan keempat kartu yang mungkin
2 4 6 7 8
Angka yang diproses adalah...
2, 4, 6, dan 7

Mencari solusi...
(2 + (4 * 7)) - 6
2 + ((4 * 7) - 6)
(2 - 6) + (4 * 7)
2 - (6 - (4 * 7))
(2 + 6) * (7 - 4)
(2 - 6) + (7 * 4)
2 - (6 - (7 * 4))
(2 + (7 * 4)) - 6
2 + ((7 * 4) - 6)
((2 * 7) + 4) + 6
(2 * 7) + (4 + 6)
4 + ((7 * 2) + 6)
((4 * 7) + 2) - 6
(4 * 7) + (2 - 6)
((4 * 7) - 6) + 2
(4 * 7) - (6 - 2)
(6 + (2 * 7)) + 4
6 + ((2 * 7) + 4)
(6 + 2) * (7 - 4)
((6 - 2) * 7) - 4
6 / (2 - (7 / 4))
(6 + 4) + (2 * 7)
6 + (4 + (2 * 7))
(6 + 4) + (7 * 2)
6 + (4 + (7 * 2))
(6 + (7 * 2)) + 4
6 + ((7 * 2) + 4)
(7 * 2) + 4 + 6
(7 * 2) + (4 + 6)
(7 * 2) + 6 + 4
(7 * 2) + (6 + 4)
(7 - 4) * (2 + 6)
(7 * 4) + 2 - 6
(7 * 4) + (2 - 6)
(7 - 4) * (6 + 2)
((7 * 4) - 6) + 2
(7 * 4) - (6 - 2)
(7 * (6 - 2)) - 4

Terdapat 47 buah solusi
Lama Waktu eksekusi: 0.27700s

Apakah Anda ingin menyimpan hasil ? (Y/N)
Input answer :
Masukkan tidak sesuai! Ulangi (Y/N)
Masukkan Pilihan : Y
Masukkan nama file : test3
Berhasil menyimpan solusi di '../test/test3.txt'
PS D:\Semester 4\STIMA\Tuci11_13521168\src>

```

Pada test case 3 ini, dapat dilihat bahwa input yang tidak valid tidak akan diproses oleh program dan akan kembali meminta input yang valid dari user. Sedangkan jika user memberikan input yang tidak sesuai dengan asumsi yaitu input melebihi 4 kartu, maka yang dihitung hanya 4 kartu awal.

4

```

File Edit Selection View Go Run Terminal Help
main.cpp - Tucil1_13521168 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\Semester 4\STIMA\Tucil1\Tucil1_13521168\src> & .\"main.exe"

████████████████████████████████████████████████████████████████████████████████
By Satria Octavianus Nababan - 13521168

Pilih sistem permainan:
(1) Input nilai kartu sendiri
(2) Pilih kartu secara acak
Input answer : 1
Masukkan keempat kartu yang mungkin
A A A A
Angka yang diproses adalah...
1, 1, 1, dan 1

Mencari solusi...

Terdapat 0 buah solusi
Lama Waktu eksekusi: 0.00200s

Apakah Anda ingin menyimpan hasil ? (Y/N)
Input answer : Y
Masukkan nama file : test4
Berhasil menyimpan solusi di '../test/test4.txt'
PS D:\Semester 4\STIMA\Tucil1\Tucil1_13521168\src>

```

Test case 4 merupakan contoh input yang tidak memiliki solusi.

5

```

File Edit Selection View Go Run Terminal Help
main.cpp - Tucil1_13521168 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\Semester 4\STIMA\Tucil1\Tucil1_13521168\src> & .\"main.exe"

████████████████████████████████████████████████████████████████████████████████
By Satria Octavianus Nababan - 13521168

Pilih sistem permainan:
(1) Input nilai kartu sendiri
(2) Pilih kartu secara acak
Input answer : 2
Kartu : 5 5 2 7
Mencari solusi...
(5 + 5) + (2 * 7)
5 + (5 + (2 * 7))
(5 + 5) + (7 * 2)
5 + (5 + (7 * 2))
(5 + (2 * 7)) + 5
5 + ((2 * 7) + 5)
(5 + (7 * 2)) + 5
5 + ((7 * 2) + 5)
((2 * 7) + 5) + 5
(2 * 7) + (5 + 5)
((7 * 2) + 5) + 5
(7 * 2) + (5 + 5)

Terdapat 12 buah solusi
Lama Waktu eksekusi: 0.09700s

Apakah Anda ingin menyimpan hasil ? (Y/N)
Input answer : Y
Masukkan nama file : test5
Berhasil menyimpan solusi di '../test/test5.txt'
PS D:\Semester 4\STIMA\Tucil1\Tucil1_13521168\src>

```

Pada test case 5, user bermain dengan kartunya dipilih secara random oleh program. Program memberikan kartu yang telah diacak yaitu 5,5,2,7 dan memberikan 12 buah solusi dalam waktu 0.09700s.

6

```

File Edit Selection View Go Run Terminal Help
main.cpp - Tucil1_13521168 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\Semester 4\STIPA\Tucil1\Tucil1_13521168\src> & .\main.exe

By Satria Octavianus Nababan - 13521168

Pilih sistem permainan:
(1) Input nilai kartu sendiri
(2) Pilih kartu secara acak

Input answer : 2
Kartu : A 4 4 3
Mencari solusi...
((1 * 4) + 4) * 3
(1 * (4 + 4)) * 3
1 * ((4 + 4) * 3)
(1 * 3) * (4 + 4)
1 * (3 * (4 + 4))
(4 + (1 * 4)) * 3
((4 * 1) + 4) * 3
((4 / 1) + 4) * 3
((4 - 1) + 3) * 4
(4 - (1 - 3)) * 4
((4 + 4) * 1) * 3
(4 + 4) * (1 * 3)
(4 + (4 * 1)) * 3
((4 + 4) / 1) * 3
(4 + (4 / 1)) * 3
(4 + 4) / (1 / 3)
4 * ((4 - 1) + 3)
4 * (4 - (1 - 3))
((4 + 4) * 3) * 1
(4 + 4) * (3 * 1)
((4 + 4) * 3) / 1
(4 + 4) * (3 / 1)
4 * ((4 + 3) - 1)
((4 + 3) - 1) * 4
(4 + (3 - 1)) * 4
4 * ((3 - 1) + 4)
4 * (3 - (1 - 4))
4 * ((3 + 4) - 1)
4 * (3 + (4 - 1))
((3 - 1) + 4) * 4
(3 - (1 - 4)) * 4
(3 * 1) * (4 + 4)
3 * ((1 * 4) + 4)
3 * (1 * (4 + 4))
(3 / 1) * (4 + 4)
3 / (1 / (4 + 4))
((3 + 4) - 1) * 4
(3 + (4 - 1)) * 4
3 * (4 + (1 * 4))
3 * ((4 * 1) + 4)
3 * ((4 / 1) + 4)
(3 * (4 + 4)) * 1
3 * ((4 + 4) * 1)
3 * (4 + (4 * 1))
(3 * (4 + 4)) / 1
3 * ((4 + 4) / 1)
3 * (4 + (4 / 1))

Terdapat 48 buah solusi
Lama waktu eksekusi: 0.34500s

Apakah Anda ingin menyimpan hasil ? (Y/N)
Input answer : y
Masukkan nama file : test6
Berhasil menyimpan solusi di '...\test\test6.txt'

Ln 20, Col 1 Spaces: 2 UTF-8 CRLF C++ Win32

```

Contoh hasil lainnya, bermain dengan kartu yang dipilih acak oleh program.

## Link repository

[https://github.com/satrianababan/Tucil1\\_13521168](https://github.com/satrianababan/Tucil1_13521168)

## DAFTAR PUSTAKA

Munir, Rinaldi (2022). *Bahan Kuliah IF2211 Strategi Algoritma – Algoritma Brute Force (Bagian 1)*. Diakses pada 23 Januari 2023 pukul 23.10 dari sumber [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)

Munir, Rinaldi (2022). *Bahan Kuliah IF2211 Strategi Algoritma – Algoritma Brute Force (Bagian 2)*. Diakses pada 23 Januari 2023 pukul 23.15 dari sumber [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag2.pdf)

Munir, Rinaldi (2016). *Penerapan Algoritma Brute Force pada Permainan Kartu 24 (24 game)*. Diakses pada 24 Januari 2023 pukul 21.12 dari sumber <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/MakalahStima-2016-038.pdf>

Admin. (n.d.). *Permutation and Combination - Definition, Formulas, Derivation, Examples*. Cuemath. Diakses pada 24 Januari 2023 pukul 23.34 dari sumber <https://www.cuemath.com/data/permutations-and-combinations/>