

Laporan Tugas Kecil IF2211 Strategi Algoritma

Penyelesaian Permainan Make It 24 Menggunakan Algoritma Brute-Force

Ditujukan untuk memenuhi salah satu tugas kecil mata kuliah IF2211 Strategi Algoritma pada Semester II Tahun Akademik 2022/2023



Disusun oleh :

Ulung Adi Putra

K2/13521122

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

A. Algoritma Brute-Force

Algoritma Brute Force adalah algoritma dengan pendekatan yang lempang (straightforward) untuk memecahkan persoalan. Algoritma ini biasanya didasarkan pada pernyataan pada persoalan dan definisi konsep yang digunakan. Algoritma Brute Force memecahkan persoalan dengan sederhana, langsung, dan jelas. Contoh persoalan yang dapat diselesaikan oleh algoritma brute force : pencarian elemen pada suatu barisan, menghitung matriks, pengurutan elemen, dan sebagainya. Pada tugas ini, mahasiswa diminta untuk membuat solusi dari permainan “Make It 24” dengan menggunakan algoritma brute force.

Alur algoritma brute force yang digunakan untuk menyelesaikan permainan Make It 24 :

- i) Program akan meminta input string yang berisi 4 angka maupun melalui random generator.
- ii) Jika program meminta input dari user, maka program akan memanggil prosedur inputNumber, sedangkan random maka program akan memanggil prosedur randomInput.
- iii) Program akan mencoba semua kemungkinan permutasi angka dan operasi yang menghasilkan 24 dengan memanggil prosedur result.
- iv) Setelah hasil didapatkan, program akan memanggil fungsi clearResult untuk mendapatkan hasil dengan tiap elemennya unik. Kemudian program akan memanggil prosedur displayResult untuk menampilkan hasil ke layar.
- v) Program juga akan menanyakan apakah hasil ingin disimpan atau tidak. Jika pengguna memasukkan Y, maka hasil tersebut akan disimpan kedalam file txt.
- vi) Program juga akan menanyakan apakah ingin melanjutkan atau selesai. Jika ingin melanjutkan, maka program akan kembali meminta input seperti di awal.

Fungsi dan prosedur yang digunakan untuk melakukan komputasi :

- i) Char2Int
Fungsi ini menerima parameter X yang bertipe char, kemudian fungsi ini akan mengembalikan nilai integer dari X. Karakter yang tidak sesuai dengan ketentuan akan mengembalikan nilai 0.
- ii) Int2Char
Fungsi ini menerima parameter X bertipe integer, kemudian fungsi akan mengembalikan nilai string yang merepresentasikan X.
- iii) Split
Fungsi ini akan menerima parameter str yang bertipe string, kemudian mengembalikan vector<int> temp. fungsi ini akan mengkonversikan tiap char selain spasi yang ada di str menjadi integer dan kemudian

disimpan di temp.

iv) isValid

Fungsi ini akan menerima `vector<int>` input yang merupakan vector hasil konversi fungsi split. Fungsi ini akan mengembalikan true jika semua elemen vector valid. Tidak valid jika terdapat minimal satu elemen yang bernilai 0 atau jumlah elemen vector input tidak sama dengan 4.

v) inputNumber

Prosedur ini berfungsi meminta input string dari user kemudian dilakukan split. Setelah split dilakukan, hasil dari split akan dicek menggunakan fungsi isValid. Jika isValid bernilai false, maka inputNumber akan meminta input user hingga valid.

vi) randomInput

Prosedur ini akan menghasilkan 4 angka input yang valid.

vii) PermutationNumber

Fungsi ini menerima parameter `vector<int>` Number yang merupakan input yang sudah diubah menjadi vector integer. Kemudian fungsi akan mengembalikan vector of `vector<int>` yang berisi semua permutasi dari keempat input. Permutasi didapatkan menggunakan looping sebanyak 4 kali.

viii) PermutationOpr

Fungsi ini akan mengembalikan `vector<vector<char>>` yang berisi semua permutasi dari operator. Permutasi didapatkan menggunakan looping sebanyak 3 kali.

ix) getComputation

Fungsi ini menerima 3 parameter yaitu X dan Y yang bertipe integer dan Operator bertipe char. Kemudian akan mengembalikan nilai float dari perhitungan X operator Y.

x) result

Prosedur ini menerima 2 parameter yaitu hasil yang bertipe pointer of `vector<string>` dan Angka yang bertipe `vector<int>`. Angka merupakan input user yang sudah di split dan valid, sedangkan hasil akan digunakan untuk menyimpan semua kemungkinan yang menghasilkan nilai 24. Pada prosedur ini akan dilakukan 2 buah looping yang berfungsi untuk menguji semua permutasi angka dan operator yang didapat dari permutationNumber dan permutationOpr. Kemudian tiap kali loop akan dilakukan 5 perhitungan sebagai berikut :

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)
5. (a op (b op c)) op d

Tiap perhitungan yang menghasilkan nilai 24 akan dimasukkan kedalam vector hasil.

xi) DisplayResult

Prosedur yang digunakan untuk menampilkan semua hasil ke layar.

xii) clearResult

Fungsi ini menerima parameter result yang bertipe vector<string>. result adalah return value dari fungsi Result. Fungsi ini akan mengembalikan clearResult yang bertipe vector<string> yang elemennya merupakan elemen dari result tetapi hanya muncul 1 kali.

xiii) Makefile

Prosedur yang digunakan untuk membuat dan menulis hasil komputasi kedalam file.

B. Source Code Program

Untuk menyelesaikan persoalan ini, program ditulis menggunakan Bahasa pemrograman c++, dengan code sebagai berikut :

a) Int2Char

```
int char2Int (char X){  
    if(X == 'A' or X == 'a'){  
        return 1;  
    }else if(X == '2'){  
        return 2;  
    }else if(X == '3'){  
        return 3;  
    }else if(X == '4'){  
        return 4;  
    }else if(X == '5'){  
        return 5;  
    }else if(X == '6'){  
        return 6;  
    }else if(X == '7'){  
        return 7;  
    }else if(X == '8'){  
        return 8;  
    }else if(X == '9'){  
        return 9;  
    }else if(X == 'T'){  
        return 10;  
    }else if(X == 'J' or X == 'j'){  
        return 11;  
    }else if(X == 'Q' or X == 'q'){  
        return 12;  
    }else if(X == 'K' or X == 'k'){  
        return 13;  
    }else{  
        return 0;  
    }  
}
```

b) Int2char

```
string int2Char (int X){
    switch (X)
    {
        case 1:
            return "A";
            break;
        case 2:
            return "2";
            break;
        case 3:
            return "3";
            break;
        case 4:
            return "4";
            break;
        case 5:
            return "5";
            break;
        case 6:
            return "6";
            break;
        case 7:
            return "7";
            break;
        case 8:
            return "8";
            break;
        case 9:
            return "9";
            break;
        case 10:
            return "10";
            break;
        case 11:
            return "J";
            break;
        case 12:
            return "Q";
            break;
        case 13:
            return "K";
            break;
        default:
            break;
    }
}
```

c) Split

```
vector<int> split(string str){
    int i = 0;
    vector<int> temp;
    while(i < str.size()){
        if(str[i] != ' '){
            if(str[i] == '1' && str[i+1] == '0'){
                temp.push_back(char2Int('T'));
                i++;
            }
            else{
                temp.push_back(char2Int(str[i]));
            }
        }
        i++;
    }
    return temp;
}
```

d) isValid

```

bool isValid(vector<int> input){
    if(input.size() != 4){
        return false;
    }
    else{
        bool valid = true;
        int i =0;
        while(i<input.size() && valid){
            if(input[i] == 0){
                valid = false;
            }
            else{
                i++;
            }
        }
        return valid;
    }
}

```

e) inputNumber

```

void inputNumber (vector<int>* Number){
    string input;
    bool valid = false;
    cout << "Masukkan 4 angka! (spasi untuk input angka selanjutnya, enter jika selesai input)"<< endl;
    input = "";
    getline(cin, input);
    // inputStr(&input);
    vector<int> temp;
    temp = split(input);
    while(!isValid(temp)){
        cout<< "Input kamu gak valid gan, ulangi!!!"<< endl;
        cout<< "Input harus 4 angka dan valid (A,2,3,4,5,6,7,8,9,10,J,Q,K)"<<endl;
        input = "";
        getline(cin, input);
        temp = split(input);
    }
    *Number = temp;
}

```

f) randomInput

```

void randomInput (vector<int>* Number){
    srand(time(NULL));
    cout << "Angka yang didapat dari random : "<< endl;
    for(int i = 0; i<4; i++){
        int a = 1 + (rand() % 13);
        cout << int2Char(a)<< " ";
        (*Number).push_back(a);
    }
    cout << endl;
}

```

g) permutationNumber

```

vector<vector<int>> permutationNumber(vector<int> Number){
    vector<vector<int>> permutation;
    for(int i = 0; i<4; i++){
        for(int j = 0; j<4; j++){
            if(j != i){
                for(int k = 0; k<4; k++){
                    if(k != j and k != i){
                        for(int l = 0; l<4; l++){
                            if(l != k and l != j and l != i){
                                vector<int> temp;
                                temp.push_back(Number[i]);
                                temp.push_back(Number[j]);
                                temp.push_back(Number[k]);
                                temp.push_back(Number[l]);
                                permutation.push_back(temp);
                            }
                        }
                    }
                }
            }
        }
    }
    return permutation;
}

```

h) permutationOpr

```
vector<vector<char>> permutationOpr (vector<char> Operator){
    vector<vector<char>> permutation;
    for(int i = 0; i<4; i++){
        for(int j = 0; j<4; j++){
            for(int k = 0; k<4; k++){
                vector<char> temp;
                temp.push_back(Operator[i]);
                temp.push_back(Operator[j]);
                temp.push_back(Operator[k]);
                permutation.push_back(temp);
            }
        }
    }
    return permutation;
}
```

i) getComputation

```
float getComputation (float X, float Y, char Operator){
    if(Operator == '*'){
        return X*Y;
    }
    else if(Operator == '+'){
        return X+Y;
    }
    else if(Operator == '-'){
        return X - Y;
    }
    else if(Operator == '/'){
        if(Y == 0){
            return 9999999;
        }
        else{
            return X/Y;
        }
    }
}
```

j) result

```
void result(vector<string> * hasil, vector<int> AngkaInput){
    vector<vector<int>> permutation;
    vector<char> Operator = {'+', '-', '*', '/'};
    vector<vector<char>> permutationOP;
    permutation = permutationNumber(AngkaInput);
    permutationOP = permutationOpr(Operator);
    for(int i = 0; i<24; i++){
        for(int j = 0; j<64; j++){
            float result1 = getComputation(permutation[i][0], permutation[i][1], permutationOP[j][0]), getComputation(permutation[i][2], permutation[i][3], permutationOP[j][2]), permutationOP[j][1]);
            float result2 = getComputation(permutation[i][0], getComputation(permutation[i][1], getComputation(permutation[i][2], permutation[i][3], permutationOP[j][2]), permutationOP[j][1]), permutationOP[j][0]);
            float result3 = getComputation(getComputation(permutation[i][0], permutation[i][1], permutationOP[j][0]), permutation[i][2], permutationOP[j][1]), permutation[i][3], permutationOP[j][2]);
            float result4 = getComputation(permutation[i][0], getComputation(permutation[i][1], permutationOP[j][1]), permutation[i][2], permutationOP[j][2]), permutationOP[j][0]);
            float result5 = getComputation(permutation[i][0], getComputation(permutation[i][1], permutation[i][2], permutationOP[j][1]), permutationOP[j][0], permutation[i][3], permutationOP[j][2]);
            if(result1 == 24){
                string Equation = "(" + int2Char(permutation[i][0]) + permutationOP[j][0] + int2Char(permutation[i][1]) + ")" + permutationOP[j][1] + "(" + int2Char(permutation[i][2]) + permutationOP[j][2] + int2Char(permutation[i][3]) + ")" + permutationOP[j][0];
                (*hasil).push_back(Equation);
            }
            if(result2 == 24){
                string Equation = int2Char(permutation[i][0]) + permutationOP[j][0] + "(" + int2Char(permutation[i][1]) + permutationOP[j][1] + "(" + int2Char(permutation[i][2]) + permutationOP[j][2] + int2Char(permutation[i][3]) + ")" + permutationOP[j][0];
                (*hasil).push_back(Equation);
            }
            if(result3 == 24){
                string Equation = "(" + int2Char(permutation[i][0]) + permutationOP[j][0] + int2Char(permutation[i][1]) + ")" + permutationOP[j][1] + int2Char(permutation[i][2]) + ")" + permutationOP[j][2] + int2Char(permutation[i][3]);
                (*hasil).push_back(Equation);
            }
            if(result4 == 24){
                string Equation = int2Char(permutation[i][0]) + permutationOP[j][0] + "(" + int2Char(permutation[i][1]) + permutationOP[j][1] + int2Char(permutation[i][2]) + ")" + permutationOP[j][2] + int2Char(permutation[i][3]) + permutationOP[j][0];
                (*hasil).push_back(Equation);
            }
            if(result5 == 24){
                string Equation = "(" + int2Char(permutation[i][0]) + permutationOP[j][0] + "(" + int2Char(permutation[i][1]) + permutationOP[j][1] + int2Char(permutation[i][2]) + ")" + permutationOP[j][2] + int2Char(permutation[i][3]);
                (*hasil).push_back(Equation);
            }
        }
    }
}
```

k) displayResult

```
void displayResult(vector<string> result){
    int length = result.size();
    for(int i = 0; i<length; i++){
        cout << result[i] << " = 24" << endl;
    }
}
```

l) clearResult


```

vector<string> clearResult(vector<string> result){
    vector<string> clearresult;
    clearresult.push_back(result[0]);
    for(int i = 1; i<result.size(); i++){
        bool same = false;
        for(int j = 0; j < i; j++){
            if(result[i] == result[j]){
                same = true;
                break;
            }
        }
        if(same == false){
            clearresult.push_back(result[i]);
        }
    }
    return clearresult;
}

```

m) makeFile

```

void makefile(string fileName, vector<string> clear, vector<int> input){
    string path = "test/" + fileName + ".txt";
    std::ofstream outfile (path);
    outfile << "Input : ";
    for(int i = 0; i<4; i++){
        outfile << int2Char(input[i]) << " ";
    }
    outfile << "ditemukan solusi sebanyak : " << clear.size() << endl;
    outfile << endl;
    for(int i = 0; i< clear.size(); i++){
        outfile << clear[i] << " = 24" << endl;
    }
}

```

n) main

```

int main(){
    bool run = true;
    while (run){
        string inputUser = "";
        vector<int> Input;
        vector<string> Hasil;
        string simpan;
        cout << "Ingin input angka sendiri? (Y/N) "<< endl;
        getline(cin, inputUser);
        while(inputUser != "Y" && inputUser != "N"){
            cout << "Input Y atau N !!!"<< endl;
            inputUser = "";
            getline(cin, inputUser);
        }
        if(inputUser == "Y"){
            inputNumber(&Input);
        }
        else{
            randomInput(&Input);
        }
        auto start = high_resolution_clock::now();
        result(&Hasil, Input);
        vector<string> clear;
        if(Hasil.size() > 0){
            clear = clearResult(Hasil);
        }
        else{
            clear = Hasil;
        }
        auto stop = high_resolution_clock::now();
        cout << "ditemukan solusi sebanyak : " << clear.size() << endl;
        displayResult(clear);
        auto duration = duration_cast<microseconds>(stop - start);
        cout << "execution time : " << duration.count() << " microseconds"<< endl;
        cout << "Apakah hasil ingin disimpan? (Y/N) ";
        cin >> simpan;
        if(simpan == "Y"){
            string Name;
            cout<<"Masukkan nama file! ";
            cin >> Name;
            makefile(Name, clear, Input);
            cout << "hasil berhasil disimpan di folder test"<< endl;
        }
    }
}

```

```
cout << "Ketik 1 jika ingin lanjut, 0 jika ingin udahan ! ";  
string lanjut;  
cin >> lanjut;  
while(lanjut != "1" && lanjut != "0"){  
    cout<< "Inputnya 1 atau 0 bang, ulangi!!! "<< endl;  
    cin >> lanjut;  
}  
if(lanjut == "0"){  
    run = false;  
}  
}
```

C. Pengujian Program

- Input angka menggunakan random generator

```
PS C:\Users\ACER\OneDrive\Dokumen\Kuliah!!!!\STIMA\Tucil 1> bin\main
Ingin input angka sendiri? (Y/N)
N
Angka yang didapat dari random :
4 6 6 3
ditemukan solusi sebanyak : 26
((4+6)*3)-6 = 24
(4*3)+(6+6) = 24
((4*3)+6)+6 = 24
((6-4)+6)*3 = 24
(6-(4-6))*3 = 24
6+((4*3)+6) = 24
(6+(4*3))+6 = 24
((6+4)*3)-6 = 24
(6+6)+(4*3) = 24
6+(6+(4*3)) = 24
((6+6)-4)*3 = 24
(6+(6-4))*3 = 24
(6*6)-(4*3) = 24
(6+6)+(3*4) = 24
6+(6+(3*4)) = 24
(6*6)-(3*4) = 24
6+((3*4)+6) = 24
(6+(3*4))+6 = 24
(3*4)+(6+6) = 24
((3*4)+6)+6 = 24
(3*(4+6))-6 = 24
(3*(6+4))-6 = 24
3*((6-4)+6) = 24
3*(6-(4-6)) = 24
3*(6+(6-4)) = 24
3*((6+6)-4) = 24
execution time : 388 microseconds
Apakah hasil ingin disimpan? (Y/N) N
Ketik 1 jika ingin lanjut, 0 jika ingin udahan ! █
```

- Input angka dari pengguna

```
Ingin input angka sendiri? (Y/N)
Input Y atau N !!!
Y
Masukkan 4 angka! (spasi untuk input angka selanjutnya, enter jika selesai input)
6 6 6 6
ditemukan solusi sebanyak : 7
(6+6)+(6+6) = 24
6+(6+(6+6)) = 24
((6+6)+6)+6 = 24
6+((6+6)+6) = 24
(6+(6+6))+6 = 24
(6*6)-(6+6) = 24
((6*6)-6)-6 = 24
execution time : 1250 microseconds
Apakah hasil ingin disimpan? (Y/N) N
Ketik 1 jika ingin lanjut, 0 jika ingin udahan ! █
```

- Jika input tidak valid

```

Ingin input angka sendiri? (Y/N)
Input Y atau N !!!
Y
Masukkan 4 angka! (spasi untuk input angka selanjutnya, enter jika selesai input)
6 6 6 6
Input kamu gak valid gan, ulangi!!!
Input harus 4 angka dan valid (A,2,3,4,5,6,7,8,9,10,J,Q,K)
6 6 6 Z
Input kamu gak valid gan, ulangi!!!
Input harus 4 angka dan valid (A,2,3,4,5,6,7,8,9,10,J,Q,K)

```

- Input valid setelah beberapa percobaan

```

Ingin input angka sendiri? (Y/N)
Input Y atau N !!!
Y
Masukkan 4 angka! (spasi untuk input angka selanjutnya, enter jika selesai input)
6 6 6 6
Input kamu gak valid gan, ulangi!!!
Input harus 4 angka dan valid (A,2,3,4,5,6,7,8,9,10,J,Q,K)
6 6 6 Z
Input kamu gak valid gan, ulangi!!!
Input harus 4 angka dan valid (A,2,3,4,5,6,7,8,9,10,J,Q,K)
2 2 2 2
ditemukan solusi sebanyak : 0
execution time : 621 microseconds

```

- Input bila tidak ada kemungkinan yang menghasilkan 24

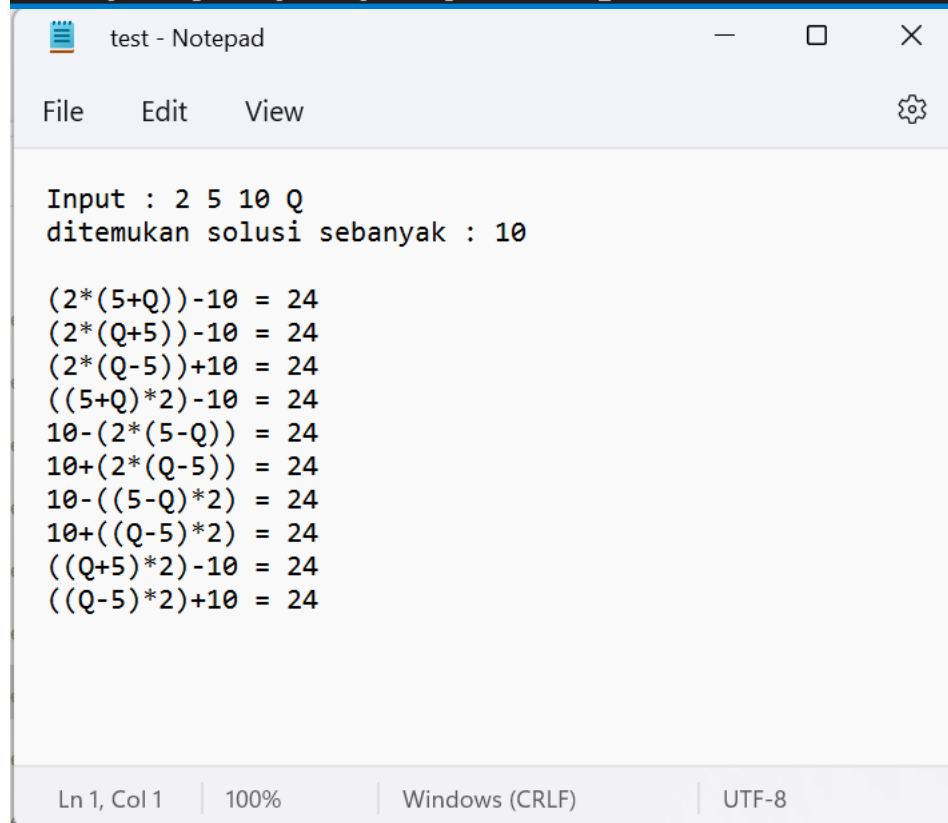
```

Ingin input angka sendiri? (Y/N)
Input Y atau N !!!
Y
Masukkan 4 angka! (spasi untuk input angka selanjutnya, enter jika selesai input)
A A A A
ditemukan solusi sebanyak : 0
execution time : 912 microseconds
Apakah hasil ingin disimpan? (Y/N)

```

- Menyimpan ke dalam file

```
Ingin input angka sendiri? (Y/N)
Input Y atau N !!!
Y
Masukkan 4 angka! (spasi untuk input angka selanjutnya, enter jika selesai input)
2 5 10 Q
ditemukan solusi sebanyak : 10
(2*(5+Q))-10 = 24
(2*(Q+5))-10 = 24
(2*(Q-5))+10 = 24
((5+Q)*2)-10 = 24
10-(2*(5-Q)) = 24
10+(2*(Q-5)) = 24
10-((5-Q)*2) = 24
10+((Q-5)*2) = 24
((Q+5)*2)-10 = 24
((Q-5)*2)+10 = 24
execution time : 537 microseconds
Apakah hasil ingin disimpan? (Y/N) Y
Masukkan nama file! test
hasil berhasil disimpan di folder test
ketik 1 jika ingin lanjut, 0 jika ingin udahan ! █
```



```
test - Notepad

File Edit View

Input : 2 5 10 Q
ditemukan solusi sebanyak : 10

(2*(5+Q))-10 = 24
(2*(Q+5))-10 = 24
(2*(Q-5))+10 = 24
((5+Q)*2)-10 = 24
10-(2*(5-Q)) = 24
10+(2*(Q-5)) = 24
10-((5-Q)*2) = 24
10+((Q-5)*2) = 24
((Q+5)*2)-10 = 24
((Q-5)*2)+10 = 24

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

D. Link Repository

https://github.com/Ulung32/Tucil1_13521122.git

E. Tabel

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	✓	