

Tugas Kecil IF2211 Strategi Algoritma

**Laporan Solusi Permainan 24 dengan
Algoritma *Brute Force***



Disusun oleh:

Muhhamad Syauqi Jannatan

K03 / 13521014

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

2023

1. Algoritma *Brute Force*

Algoritma *brute force* merupakan sebuah metode untuk menemukan solusi dengan mencoba semua kemungkinan secara berurutan. Dalam dunia komputasi, algoritma ini sangat sering digunakan dalam memecahkan suatu masalah karena metodenya yang praktis dan dapat digunakan untuk memecahkan hampir seluruh permasalahan. Dalam beberapa kasus, sebagian perhitungan dapat diabaikan karena bersifat sama persis dengan yang lainnya sehingga tidak diperlukan lagi.

Konsep algoritma bruteforce dapat diterapkan pada program pencarian solusi permainan 24 dengan cara mencoba semua kombinasi operator matematika dan angka yang mungkin, satu per satu, sampai ditemukan solusi yang benar. Algoritma ini akan mengevaluasi setiap ekspresi yang dibentuk dari setiap kombinasi operator dan angka, dan menyimpan ekspresi yang menghasilkan nilai 24. Namun, metode ini dapat sangat memakan waktu dan sumber daya jika jumlah kombinasi yang mungkin sangat besar. Langkah algoritma *Brute Force* yang diterapkan pada program pencari solusi permainan 24 antara lain adalah:

1. Inisialisasi: Program menerima input berupa 4 set angka dari user yang akan digunakan dalam permainan.
2. Program meng-generate semua kombinasi dari angka dan operator yang tervalidasi.
3. Program mengevaluasi setiap ekspresi yang dihasilkan dari kombinasi.
4. Apabila ekspresi menghasilkan 24, maka seluruh proses operasi yang telah dilakukan akan dicatat sebagai salah satu solusi. Kemudian program akan mengecek apakah solusi tersebut ganda atau tidak. Jika solusi tersebut sudah ada sebelumnya, maka tidak akan dimasukkan ke dalam solusi.
5. Program akan melakukan iterasi sehingga program akan meninjau kemungkinan dari seluruh permutasi yang terjadi pada angka dan pada operasi.
6. Program akan berhenti saat seluruh permutasi telah dilakukan. Kemudian program akan menampilkan hasil output berupa solusi yang dihasilkan.

2. Source Program

A. Operators

```
#include "operators.hpp"

void splashScreen()
{
    cout << "          Welcome to 24 SOLVER!          " << endl;
    cout << "===== " << endl;
    cout << "                                     " << endl;
    cout << "  |2_ _ | |A /\ | |Q . | |T _ | " << endl;
    cout << "  |( \ \ / ) | | / \ \ | | // \ \ | | ( ) | " << endl;
    cout << "  | \ \ / | | \ \ / | |( _ , _ ) | |( _ X _ ) | " << endl;
    cout << "  | \ \ / 2 | | \ \ / A | | I Q | | Y T | " << endl;
    cout << "  `-----` `-----` `-----` `-----` " << endl;
    cout << "      created by: Muhhamad Syauqi Jannatan " << endl;
    cout << "===== " << endl;
    cout << endl;
}

int convertString(string s)
{
    if(s=="A"){
        return 1;
    }
    else if(s=="J"){
        return 11;
    }
    else if(s=="Q"){
        return 12;
    }
    else if(s=="K"){
        return 13;
    }
    else{
        return stoi(s);
    }
}

string convertOp(int op)
{
    if (op == 0){
        return "+";
    }
    else if (op == 1){
```

```

        return "-";
    }
    else if (op == 2){
        return "*";
    }
    else if (op == 3){
        return "/";
    }
    else{
        return "error";
    }
}

float operation(float a, float b, int op)
{
    if(op==0){
        return a+b;
    }
    else if(op==1){
        return a-b;
    }
    else if(op==2){
        return a*b;
    }
    else if(op==3){
        if (b==0){
            return 0;
        }
        else{
            return a/b;
        }
    }
    else{
        return 999;
    }
}

void printSolution(vector<string> v)
{
    for(int i=0; i<v.size(); i++){
        cout << v[i] << endl;
    }
    cout << endl;
}

bool isDuplicate(vector<string> v, string s)
{
    for(int i=0; i<v.size(); i++){

```

```

        if(v[i]==s){
            return true;
        }
    }
    return false;
}

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

vector<vector<int>> permutation(vector<int> &nums)
{
    if (nums.size() <= 1)
    {
        return {nums};
    }

    vector<vector<int>> result;
    for (int i = 0; i < nums.size(); i++)
    {
        vector<int> sol(nums.begin(), nums.end());
        sol.erase(sol.begin() + i);
        auto res = permutation(sol);

        for (int j = 0; j < res.size(); j++)
        {
            vector<int> sol_ = res[j];
            sol_.insert(sol_.begin(), nums[i]);
            result.push_back(sol_);
        }
    }
    return result;
}

void saveFile(vector<string> solution, string a, string b, string c, string d)
{
    ofstream myfile;
    string fileName, filePath;
    cout << "Enter filename: ";
    cin >> fileName;
    filePath = "../test/" + fileName + ".txt";

```

```

myfile.open (filePath);
myfile <<"Your Number : " << a << " " << b << " " << c << " " << d <<
endl;
myfile << "Total Solutions: " << solution.size() << endl;
myfile << "======" << endl;
for(int i = 0; i < solution.size(); i++){
    myfile << solution[i] << endl;
}
myfile.close();
}

```

B. Solve24 Algorithm

```

#include "solve24.hpp"

void solve24(int a, int b, int c, int d, vector<int> numbers, vector<string>
*solution, int counter)
{
    vector<vector<int>> nums_res = permutation(numbers);
    string temp = "";
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            for (int k = 0; k < 4; k++)
            {
                for (int x = 0; x < 24; x++)
                {
                    int a = nums_res[x][0];
                    int b = nums_res[x][1];
                    int c = nums_res[x][2];
                    int d = nums_res[x][3];

                    // a _ (b _ (c _ d))
                    if (operation(a,operation(b,operation(c,d,k),j),i) ==
24.0){
                        temp = to_string(a) + " " + convertOp(i) + " (" +
to_string(b) + " " + convertOp(j) + " (" + to_string(c) + " " + convertOp(k) +
" " + to_string(d) + "))";
                        if(!isDuplicate(*solution, temp)){
                            (*solution).push_back(temp);
                            counter++;
                        }
                    }
                }
            }
        }
    }

    // a _ ((b _ c) _ d)

```

```

        if (operation(a,operation(operation(b,c,j),d,k),i) ==
24.0){
            temp = to_string(a) + " " + convertOp(i) + " (" +
to_string(b) + " " + convertOp(j) + " " + to_string(c) + ") " + convertOp(k) +
" " + to_string(d) + ")";
            if(!isDuplicate(*solution, temp)){
                (*solution).push_back(temp);
                counter++;
            }
        }

        // (a _ (b _ c)) _ d
        if (operation(operation(a,operation(b,c,j),i),d,k) ==
24.0){
            temp = "(" + to_string(a) + " " + convertOp(i) + " (" +
+ to_string(b) + " " + convertOp(j) + " " + to_string(c) + ")) " +
convertOp(k) + " " + to_string(d);
            if(!isDuplicate(*solution, temp)){
                (*solution).push_back(temp);
                counter++;
            }
        }

        // ((a _ b) _ c) _ d
        if (operation(operation(operation(a,b,i),c,j),d,k) ==
24.0){
            temp = "(" + to_string(a) + " " + convertOp(i) + " " +
+ to_string(b) + ") " + convertOp(j) + " " + to_string(c) + ") " +
convertOp(k) + " " + to_string(d);
            if(!isDuplicate(*solution, temp)){
                (*solution).push_back(temp);
                counter++;
            }
        }

        // (a _ b) _ (c _ d)
        if (operation(operation(a,b,i),operation(c,d,k),j) ==
24.0){
            temp = "(" + to_string(a) + " " + convertOp(i) + " " +
to_string(b) + ") " + convertOp(j) + " (" + to_string(c) + " " + convertOp(k)
+ " " + to_string(d) + ")";
            if(!isDuplicate(*solution, temp)){
                (*solution).push_back(temp);
                counter++;
            }
        }
    }
}

```

```

    }
}
printSolution(*solution);
cout << counter << " solutions found" << endl;
}

```

C. Main

```

#include <iostream>
#include <string>
#include <vector>
#include <chrono>
#include <fstream>
#include "operators.cpp"
#include "solve24.cpp"

using namespace std;

int main()
{
    int counter=0;
    splashScreen();
    vector<int> numbers;
    vector<string> solution;
    using Clock = chrono::high_resolution_clock;
    cout << "1.Input Manual" << endl;
    cout << "2.Input Random" << endl;
    cout << "Choose: ";
    int choose;
    cin >> choose;
    if (choose == 1)
    {
        string input[4];
        cout << "Input 4 numbers: ";
        cin >> input[0] >> input[1] >> input[2] >> input[3];
        while
(!checkInput(input[0])&&!checkInput(input[1])&&!checkInput(input[2])&&!checkInput(input[3]))
        {
            cout << "Invalid Input" << endl;
            cout << "Input 4 numbers: ";
            cin >> input[0] >> input[1] >> input[2] >> input[3];
        }
        cout << "======" << endl;
        cout << "Your Numbers: ";
        for (int i = 0; i < 4; i++)
        {

```



```

        cout << input[i] << " ";
        numbers.push_back(convertString(input[i]));
    }
    cout << endl;
    cout << "The solution is: " << endl;

}
else if (choose == 2)
{
    srand(time(NULL));
    cout << "random numbers: ";
    for (int i = 0; i < 4; i++)
    {
        numbers.push_back(rand() % 13 + 1);
        cout << numbers[i] << " ";
    }
    cout << endl;
}
else
{
    cout << "Invalid Input" << endl;
    return 0;
}
auto start = Clock::now();
solve24(numbers[0], numbers[1], numbers[2], numbers[3], numbers,
&solution, counter);
auto end = Clock::now();
auto duration = chrono::duration_cast<chrono::milliseconds>(end - start);
cout << "Execution Time: " << duration.count() << " ms" << endl;
cout << "Do you want to save the solution to a file? (y/n): ";
char save;
cin >> save;
if(save == 'y'){
    saveFile(solution, to_string(numbers[0]), to_string(numbers[1]),
to_string(numbers[2]), to_string(numbers[3]));
}
return 0;
}

```

3. Screenshots Hasil Input dan Output Program

```

Welcome to 24 SOLVER!
=====

  2  A  Q  T
  (V) (/ \ ( // \ ( )
  \ / \ \ / \ ( , ) (X)
  V 2 V A I Q Y T

created by: Muhhamad Syauqi Jannatan
=====

1.Input Manual
2.Input Random
Choose: 1
Input 4 numbers:
3
3
3
3
=====
Your Numbers: 3 3 3 3
The solution is:
(3 * (3 * 3)) - 3
((3 * 3) * 3) - 3

2 solutions found
Execution Time: 1 ms
Do you want to save the solution to a file? (y/n): █

```

Gambar 3.1 Testcase 1

```

1.Input Manual
2.Input Random
Choose: 1
Input 4 numbers:
A
K
2
3
=====
Your Numbers: A K 2 3
The solution is:
1 + ((13 * 2) - 3)
(1 + (13 * 2)) - 3
1 + ((2 * 13) - 3)
(1 + (2 * 13)) - 3
(1 - 3) + (13 * 2)
(1 - 3) + (2 * 13)
1 - (3 - (13 * 2))
1 - (3 - (2 * 13))
((3 - 1) * 13) - 2
((13 * 2) + 1) - 3
(13 * 2) + (1 - 3)
((2 * 13) + 1) - 3
(2 * 13) + (1 - 3)
((13 * 2) - 3) + 1
((2 * 13) - 3) + 1
(13 * 2) - (3 - 1)
(13 * (3 - 1)) - 2
(2 * 13) - (3 - 1)

18 solutions found
Execution Time: 8 ms
Do you want to save the solution to a file? (y/n): █

```

Gambar 3.2 Testcase 2

```

1.Input Manual
2.Input Random
Choose: 1
Input 4 numbers:
10
10
10
3
=====
Your Numbers: 10 10 10 3
The solution is:

0 solutions found
Execution Time: 1 ms
Do you want to save the solution to a file? (y/n): █

```

Gambar 3.3 Testcase 3

```

1.Input Manual
2.Input Random
Choose: 2
random numbers: 8 4 2 9
(9 - (4 + 2)) * 8
(9 - (2 + 4)) * 8
((9 - 4) - 2) * 8
((9 - 2) - 4) * 8
8 * (9 - (4 + 2))
8 * (9 - (2 + 4))
8 * ((9 - 4) - 2)
8 * ((9 - 2) - 4)

8 solutions found
Execution Time: 4 ms
Do you want to save the solution to a file? (y/n): █

```

Gambar 3.4 Testcase 4

```

1.Input Manual
2.Input Random
Choose: 2
random numbers: 11 10 4 12
11 + (10 + (12 / 4))
(11 + 10) + (12 / 4)
10 + (11 + (12 / 4))
(10 + 11) + (12 / 4)
11 + ((12 / 4) + 10)
(11 + (12 / 4)) + 10
10 + ((12 / 4) + 11)
(10 + (12 / 4)) + 11
((12 / 4) + 11) + 10
(12 / 4) + (11 + 10)
((12 / 4) + 10) + 11
(12 / 4) + (10 + 11)

12 solutions found
Execution Time: 4 ms
Do you want to save the solution to a file? (y/n): █

```

Gambar 3.5 Testcase 5

```

Choose: 1
Input 4 numbers:
5
9
13
11
=====
Your Numbers: 5 9 13 11
The solution is:
(5 - 11) * (9 - 13)
(9 - 13) * (5 - 11)
(13 - 9) * (11 - 5)
(11 - 5) * (13 - 9)

4 solutions found
Execution Time: 3 ms
Do you want to save the solution to a file? (y/n): y
Enter filename: test6

```

Gambar 3.6 Testcase 6

```

tucil 1 > test > test6.txt
1 Your Number : 5 9 13 11
2 Total Solutions: 4
3 =====
4 (5 - 11) * (9 - 13)
5 (9 - 13) * (5 - 11)
6 (13 - 9) * (11 - 5)
7 (11 - 5) * (13 - 9)
8

```

Gambar 3.7 File .txt dari Testcase 6

4. *Link Repository*

https://github.com/syauqijan/Tucil1_13521014

5. To Do List

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	V	
2. Program berhasil running	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 file teks	V	