Penggunaan Algoritma Brute Force dalam Penyelesaian Permainan 24 Kartu

Tugas Kecil 1 IF2211

Semester II tahun 2022/2023

Disusun Oleh :

Angger Ilham Amanullah

NIM : 13521001

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG 2022

I. Penjelasan Masalah

Algoritma brute force adalah metode pencarian atau pemecahan masalah dengan cara mencoba semua kemungkinan secara berurutan, tanpa memperhatikan kondisi atau optimasi tertentu. Hal ini membuat algoritma ini memiliki kompleksitas waktu yang cukup tinggi, tetapi sering digunakan karena mudah dipahami dan diterapkan. Algoritma ini juga akan selalu memberikan kita penyelesaian dari suatu masalah tadi dikarenakan dengan penggunaan brute force maka akan mencoba seluruh kemungkinan yang ada sehingga selalu ditemukannya penyelesaian dari permasalahan yang dihadapi.

Permasalahan yang diangkat pada tugas kecil kali ini yaitu penggunaan brute force dalam menemukan semua solusi dari dari permainan kartu 24. Permainan Kartu 24 adalah permainan matematika yang menggunakan 4 kartu dengan angka yang berbeda-beda dan tujuan dari permainan ini adalah untuk menemukan kombinasi dari 4 kartu tersebut yang dapat dioperasikan dengan 5 operator matematika yaitu tambah (+), kurang(-), kali (*), bagi (/), dan kurung (()) untuk menghasilkan nilai 24. Contohnya, jika kartu yang didapat adalah 1, 2, 3, 4, maka kombinasi yang dapat digunakan untuk mencapai 24 adalah 1 * 2 * 3 * 4 = 24. Permainan ini dapat digunakan untuk meningkatkan kemampuan matematika serta logika seseorang.

Pada Tugas Kecil kali ini kita diminta untuk membuat program dalam salah satu dari 3 bahasa yaitu C, C++, atau Java. Kartu yang disediakan yaitu pada kartu remi sehingga dibatasi angka yang bisa dipakai dari angka 1 hingga angka 13. Untuk kartu As mewakili angka 1, Jack mewakili angka 11, Queen mewakili angka 12, King mewakili angka 13, dan sisanya mewakili bilangan pada kartu itu sendiri.

Pada kali ini penulis menggunakan Bahasa C++ dalam pembuatan program untuk penyelesaian game kartu 24 ini. Penulis menggunakan permutasi dari angka – angka pada kartu dan juga permutasi dari operator yang akan digunakan. Jadi setelah program mendapatkan 4 kartu yang bisa didapat dari inputan user maupun auto generate dari program, lalu langkah selanjutnya dari program tersebut yaitu akan mencoba seluruh kemungkinan operasi dari 4 kartu tersebut. Jika terdapat operasi yang bisa menghasilkan angka 24, maka operasi tadi disimpan didalam array of string yang merupakan hasil dari perhitungan 4 angka tersebut. Jika semua kemungkinan operasi sudah dijalankan, maka selanjutnya kartu akan diacak kembali secara permutasi dan proses sebelumnya akan dijalankan hingga semua kemungkinan acakan kartu telah diproses. Setelah semua kemungkinan acakan kartu telah diproses, maka program akan menampilkan semua hasil kemungkinan operasi yang dapat menghasilkan angka 24. Jika tidak ada operasi yang menghasilkan angka 24, program akan menampilkan kalimat ‘ Tidak ada solusi ’.

Lalu setelah itu, program akan menanyakan apakah ingin menyimpan hasil solusi. Jika ya, maka akan diminta nama file yang akan digunakan untuk menyimpan solusi tersebut.

II. Source Code

src/main.cpp

```cpp
#include<bits/stdc++.h>
using namespace std;



double operation(double x, double y, int k){

    if(k == 0) return x + y;
    else if(k == 1) return x - y;
    else if(k == 2) return x * y;
    else return x / y;
}

int op(int k){
    if(k == 0) return '+';
    else if(k == 1) return '-';
    else if(k == 2) return '*';
    else return '/';
}

void intToString(int k, string *s){
    if(k == 0) *s = "+";
    else if(k == 1) *s = "-";
    else if(k == 2) *s = "*";
    else *s = "/";
}

bool isValid(string s){
    if(s.length() == 1){
        if(s[0] >= '2' and s[0] <= '9'){
            return true;
        }
        else if(s[0] == 'A' or s[0] == 'J' or s[0] == 'Q' or s[0] == 'K'){
            return true;
        }
        else{
            return false;
        }
    }
    else if(s.length() == 2){
        if(s[0] == '1' and s[1] == '0'){
            return true;
        }
        else{
            return false;
```

```
        }
    }
    else{
        return false;
    }
}

int stringToInt(string s){
    if(s == "A"){
        return 1;
    }
    else if(s[0] == '1' and s[1] == '0'){
        return 10;
    }
    else if(s[0] == 'J'){
        return 11;
    }
    else if(s[0] == 'Q'){
        return 12;
    }
    else if(s[0] == 'K'){
        return 13;
    }
    else{
        return s[0] - '0';
    }
}

int main(){
    while(true){
        cout << "Selamat datang di 24 Game\n";
        cout << "Pilih menu :\n";
        cout << "1. Input 4 kartu\n";
        cout << "2. Generate 4 kartu secara random\n";
        cout << "3. Keluar\n";
        int n;
        cin>>n;
        while(n != 1 and n != 2 and n != 3){
            cout << "\n\nInput tidak valid\n\n";
            cout << "Pilih menu :\n";
            cout << "1. Input 4 kartu\n";
            cout << "2. Generate 4 kartu secara random\n";
            cout << "3. Keluar\n";
            cout << "Masukkan pilihan : ";
            cin >> n;
```

```
        }

        string s1,s2,s3,s4;
        int card[4];
        if(n==1 or n==2){
            if(n==1){
                cout << "Masukkan 4 kartu : ";
                cin >> s1 >> s2 >> s3 >> s4;
                while(!isValid(s1) or !isValid(s2) or !isValid(s3) or
!isValid(s4)){
                    cout << "Input tidak valid\n";
                    cout << "Masukkan 4 kartu: ";
                    cin >> s1 >> s2 >> s3 >> s4;
                }
                card[0] = stringToInt(s1);
                card[1] = stringToInt(s2);
                card[2] = stringToInt(s3);
                card[3] = stringToInt(s4);
            }
            else if(n == 2){
                for(int i=0; i<4; i++){
                    card[i] = rand() % 13 + 1;
                }
                cout << "Kartu yang terpilih :";
                for(int i=0; i<4; i++){
                    if(card[i] == 1){
                        cout << " A";
                    }
                    else if(card[i] == 11){
                        cout << " J";
                    }
                    else if(card[i] == 12){
                        cout << " Q";
                    }
                    else if(card[i] == 13){
                        cout << " K";
                    }
                    else{
                        cout << " " << card[i];
                    }
                }
                cout << endl;
            }
                int total = 0;
                string operationRes[6969] = {};
```

```
                sort(card, card+4);
                auto start = chrono::steady_clock::now();
                do{
                    for(int i=0; i<4; i++){
                        for(int j=0; j<4; j++){
                            for(int k=0; k<4; k++){
                                string op1,op2,op3;
                                intToString(i, &op1);
                                intToString(j, &op2);
                                intToString(k, &op3);
                                // cout << op3 << endl;

                                // ((card op card) op card) op card
                                if(operation(operation(operation(card[0],
card[1], i), card[2], j), card[3], k) == 24){
                                    total++;
                                    operationRes[total] = "((" +
to_string(card[0]) + " " + op1 + " " + to_string(card[1]) + ") " + op2 + " " +
to_string(card[2]) + ") " + op3 + " " + to_string(card[3]);
                                }
                                // (card op (card op card)) op card
                                if(operation(operation(card[0],
operation(card[1], card[2], j), i), card[3], k) == 24){
                                    total++;
                                    operationRes[total] = "(" +
to_string(card[0]) + " " + op1 + " (" + to_string(card[1]) + " " + op2 + " " +
to_string(card[2]) + ")) " + op3 + " " + to_string(card[3]);
                                }
                                // (card op card) op (card op card)
                                if(operation(operation(card[0], card[1], i),
operation(card[2], card[3], k), j) == 24){
                                    total++;
                                    operationRes[total] = "(" +
to_string(card[0]) + " " + op1 + " " + to_string(card[1]) + ") " + op2 + " (" +
to_string(card[2]) + " " + op3 + " " + to_string(card[3]) + ")";
                                }
                                // card op ((card op card) op card)
                                if(operation(card[0],
operation(operation(card[1], card[2], j), card[3], k), i) == 24){
                                    total++;
                                    operationRes[total] = to_string(card[0]) + "
" + op1 + " ((" + to_string(card[1]) + " " + op2 + " " + to_string(card[2]) + ")
" + op3 + " " + to_string(card[3]) + ")";
                                }
                            }
```

```
                    }
                }
            }while(next_permutation(card, card+4));

            auto end = chrono::steady_clock::now();
            auto duration = chrono::duration_cast<chrono::microseconds>(end -
start);

            if(total == 0){
                cout << "Tidak ada solusi\n";
            }
            else{
                cout<< "Jumlah solusi: " << total << "\n";
                cout << "Solusi: \n";
                for(int i = 1; i <= total; i++){
                    cout << operationRes[i] << "\n";
                }
                cout<< "\nWaktu eksekusi program : " << duration.count() << "
mikrosekon\n";
            }
        cout << "Apakah anda ingin menyimpan hasil pencarian ke dalam file?
(y/n) ";
        char c;
        cin >> c;
        while(c != 'y' and c != 'n'){
            cout << "Input tidak valid\n";
            cout << "Apakah anda ingin menyimpan hasil pencarian ke dalam
file? (y/n) ";
            cin >> c;
        }
        if(c == 'y'){
            ofstream file;
            string namafile;
            cout << "Masukkan nama file: ";
            cin >> namafile;
            namafile += ".txt";
            file.open(namafile);
            file << "Kartu yang terpilih :";
            for(int i=0; i<4; i++){
                if(card[i] == 1){
                    file << " A";
                }
                else if(card[i] == 11){
                    file << " J";
                }
```

```
                    else if(card[i] == 12){
                        file << " Q";
                    }
                    else if(card[i] == 13){
                        file << " K";
                    }
                    else{
                        file << " " << card[i];
                    }
                }
                file << "\n";
                if(total == 0){
                    file << "Tidak ada solusi\n";
                }
                else{
                    file << "Jumlah solusi: " << total << "\n";
                    file << "Solusi: \n";
                    for(int i = 1; i <= total; i++){
                        file << operationRes[i] << "\n";
                    }
                }
                file << "\nWaktu eksekusi program : " << duration.count() << "
mikrosekon";
                file.close();
            }
            else{
                cout << "Hasil pencarian tidak disimpan\n";
            }
            cout << "Apakah anda ingin melanjutkan permainan? (y/n) ";
            char c2;
            cin >> c2;
            while(c2 != 'y' and c2 != 'n'){
                cout << "Input tidak valid\n";
                cout << "Apakah anda ingin melanjutkan permainan? (y/n) ";
                cin >> c2;
            }
            if(c2 == 'n'){
                cout << "Terima kasih telah bermain\n";
                return 0;
            }
        }
        else if(n == 3){
            cout << "Terima kasih telah bermain\n";
            return 0;
        }
```

```
    }
}
```

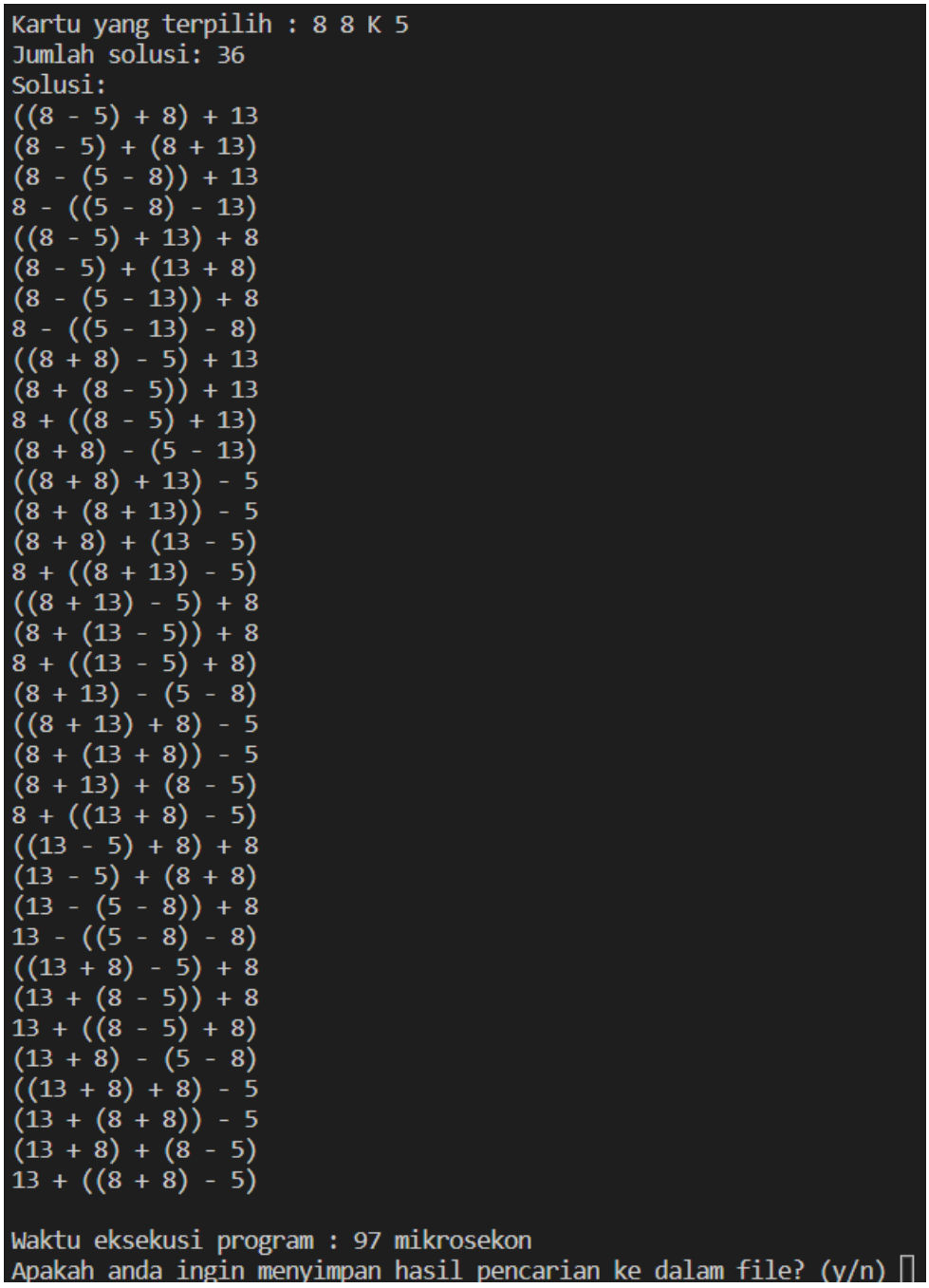
III. Screenshot Testcase

A. Testcase 1

```
Selamat datang di 24 Game
Pilih menu :
1. Input 4 kartu
2. Generate 4 kartu secara random
3. Keluar
2
Kartu yang terpilih : 3 8 4 7
Jumlah solusi: 4
Solusi:
(4 * (7 - 3)) + 8
((7 - 3) * 4) + 8
8 - ((3 - 7) * 4)
8 + ((7 - 3) * 4)

Waktu eksekusi program : 149 mikrosekon
Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n)
```

B. Testcase 2


Kartu yang terpilih : 8 8 K 5
Jumlah solusi: 36
Solusi:
((8 - 5) + 8) + 13
(8 - 5) + (8 + 13)
(8 - (5 - 8)) + 13
8 - ((5 - 8) - 13)
((8 - 5) + 13) + 8
(8 - 5) + (13 + 8)
(8 - (5 - 13)) + 8
8 - ((5 - 13) - 8)
((8 + 8) - 5) + 13
(8 + (8 - 5)) + 13
8 + ((8 - 5) + 13)
(8 + 8) - (5 - 13)
((8 + 8) + 13) - 5
(8 + (8 + 13)) - 5
(8 + 8) + (13 - 5)
8 + ((8 + 13) - 5)
((8 + 13) - 5) + 8
(8 + (13 - 5)) + 8
8 + ((13 - 5) + 8)
(8 + 13) - (5 - 8)
((8 + 13) + 8) - 5
(8 + (13 + 8)) - 5
(8 + 13) + (8 - 5)
8 + ((13 + 8) - 5)
((13 - 5) + 8) + 8
(13 - 5) + (8 + 8)
(13 - (5 - 8)) + 8
13 - ((5 - 8) - 8)
((13 + 8) - 5) + 8
(13 + (8 - 5)) + 8
13 + ((8 - 5) + 8)
(13 + 8) - (5 - 8)
((13 + 8) + 8) - 5
(13 + (8 + 8)) - 5
(13 + 8) + (8 - 5)
13 + ((8 + 8) - 5)
Waktu eksekusi program : 97 mikrosekon
Apakah anda ingin menyimpan hasil pencarian ke dalam file? (y/n)

C. Testcase 3

```
Kartu yang terpilih : Q 6 4 J
Jumlah solusi: 16
Solusi:
(4 * 6) * (12 - 11)
(4 * 6) / (12 - 11)
(4 * (12 - 11)) * 6
4 * ((12 - 11) * 6)
(4 / (12 - 11)) * 6
4 / ((12 - 11) / 6)
(6 * 4) * (12 - 11)
(6 * 4) / (12 - 11)
(6 * (12 - 11)) * 4
6 * ((12 - 11) * 4)
(6 / (12 - 11)) * 4
6 / ((12 - 11) / 4)
((12 - 11) * 4) * 6
(12 - 11) * (4 * 6)
((12 - 11) * 6) * 4
(12 - 11) * (6 * 4)

Waktu eksekusi program : 137 mikrosekon
```

D. Testcase 4

```
2
Kartu yang terpilih : 9 6 3 J
Jumlah solusi: 22
Solusi:
3 * ((6 - 9) + 11)
3 * ((6 + 11) - 9)
(3 - 11) * (6 - 9)
3 * ((11 + 6) - 9)
(3 * (11 - 6)) + 9
3 * ((11 - 9) + 6)
((6 - 3) * 11) - 9
(6 - 9) * (3 - 11)
((6 - 9) + 11) * 3
(6 - (9 - 11)) * 3
((6 + 11) - 9) * 3
(6 + (11 - 9)) * 3
9 - ((6 - 11) * 3)
(9 - 6) * (11 - 3)
9 + ((11 - 6) * 3)
(11 - 3) * (9 - 6)
((11 - 6) * 3) + 9
(11 * (6 - 3)) - 9
((11 + 6) - 9) * 3
(11 + (6 - 9)) * 3
((11 - 9) + 6) * 3
(11 - (9 - 6)) * 3

Waktu eksekusi program : 144 mikrosekon
```

E. Testcase 5

```
Kartu yang terpilih : 7 7 2 10
Jumlah solusi: 3
Solusi:
(2 + (10 / 7)) * 7
7 * ((10 / 7) + 2)
((10 / 7) + 2) * 7

Waktu eksekusi program : 81 mikrosekon
```

F. Testcase 6

```
Kartu yang terpilih : J 6 A 4
Jumlah solusi: 16
Solusi:
((1 - 6) + 11) * 4
(1 - (6 - 11)) * 4
((1 + 11) - 6) * 4
(1 + (11 - 6)) * 4
(1 + 11) * (6 - 4)
4 * ((1 - 6) + 11)
4 * ((1 + 11) - 6)
4 * ((11 + 1) - 6)
4 * ((11 - 6) + 1)
(6 - 4) * (1 + 11)
(6 - 4) * (11 + 1)
((11 + 1) - 6) * 4
(11 + (1 - 6)) * 4
(11 + 1) * (6 - 4)
((11 - 6) + 1) * 4
(11 - (6 - 1)) * 4

Waktu eksekusi program : 173 mikrosekon
```

IV. Link Github

https://github.com/ilhamanullah/Tucil1_13521001

V. Checklist

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	