

LAPORAN TUGAS KECIL 1
IF-2211 STRATEGI ALGORITMA

PENYELESAIAN PERMAINAN KARTU 24 DENGAN
ALGORITMA BRUTE FORCE



Disusun oleh:

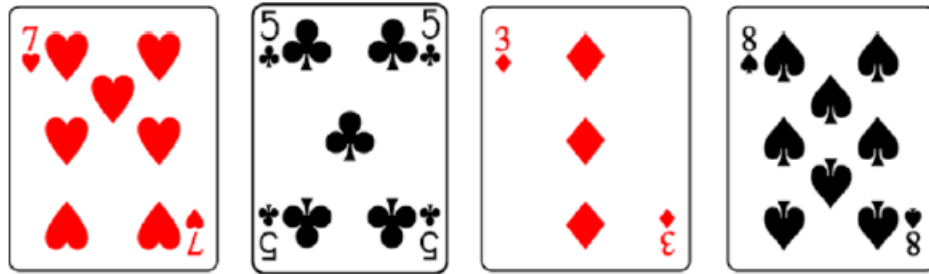
Christian Albert Hasiholan

13521078

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

BAB I

DESKRIPSI MASALAH



MAKE IT 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.

BAB II

ALGORITMA

1. Input

Pertama-tama user akan diberikan opsi apakah ingin input kartu yang akan digunakan atau degenerate secara random. Untuk opsi random, digunakan function `rand()` dan `srand()` untuk menghasilkan angka yang random. Function `srand()` digunakan bersamaan dengan function `time()` agar angka yang dihasilkan oleh `rand()` tidak selalu bernilai sama. Sedangkan untuk opsi input, user dapat menginput kartu-kartu dengan contoh seperti A 2 3 J. Apabila input user tidak sesuai dengan kartu yang ada, maka akan dilakukan validasi.

2. Validasi input

Validasi dilakukan dengan memanfaatkan array yang berisi semua kartu yang bisa diinput oleh user. Pada validasi ini digunakan loop pada input dan juga loop pada deck, yang dibantu juga dengan variable boolean. Input pertama akan dicek dengan deck, bila ditemukan string yang sama maka loop berlanjut ke input kedua. Namun bila input pertama tidak ditemukan pada deck, maka input sudah pasti tidak valid.

3. Permutasi angka

Permutasi angka didapatkan dengan menggunakan loop sebanyak jumlah digit, yaitu 4. Pertama, loop akan menentukan angka pada digit pertama, lalu loop selanjutnya menentukan digit kedua, namun dengan angka pada digit pertama tadi tidak boleh digunakan lagi pada digit kedua dan seterusnya. Hal ini berlaku juga pada digit ketiga dan keempat, dimana pada digit kedua, angka yang telah digunakan pada digit pertama dan kedua tidak boleh digunakan lagi, sedangkan untuk digit keempat hanya 1 angka tersisa saja yang dapat digunakan. Namun menentukan permutasi dengan metode ini masih belum sempurna, karena bila terdapat angka yang sama maka tetap dianggap sebagai angka yang berbeda, jadi untuk tiap 4 kartu dapat diperoleh $4 * 3 * 2 * 1 = 24$ permutasi.

4. Menentukan solusi

Untuk menentukan solusi yang dapat menghasilkan nilai 24, digunakanlah loop serta set. Loop digunakan untuk membuat operator yang berbeda, yaitu +, -, *, dan /. Setelah didapat operator

yang akan digunakan, maka penentuan solusi dibagi lagi menjadi 5 kasus berdasarkan tanda kurung pada operasi. Setelah itu dilakukan perhitungan dan juga penanganan masalah rounding yang mungkin dapat terjadi pada operasi pembagian. Lalu hasil tersebut dikomparasi dengan nilai 24, jika sama atau mendekati maka solusi tersebut akan dimasukkan pada set. Set digunakan karena sifatnya yang tidak menyimpan duplikat dari elemen yang sudah ada, oleh karena itu meskipun permutasi sebelumnya bisa menghasilkan urutan angka yang sama, namun dengan set maka solusi yang dihasilkan tidak ada yang duplikat.

BAB III

SOURCE CODE

Source code dari file **main.cpp**

```
#include <iostream>
#include <cstdlib>
#include <string>
#include <set>
#include <time.h>
#include <chrono>
#include <fstream>
using namespace std;

//Memudahkan operasi perhitungan sekaligus pengganti eval dalam python
double operation(double num1, double num2, char oprt) {
    if (oprt == '+') {
        return num1 + num2;
    }
    if (oprt == '-') {
        return num1 - num2;
    }
    if (oprt == '*') {
        return num1*num2;
    }
    if (oprt == '/') {
        return num1/num2;
    }
    else return 0;
}

//Memvalidasi input user sesuai dengan kartu yang ada
bool validateInput(string a, string b, string c, string d) {
    string card[4] = {a, b, c, d};
    string possibleCard[13] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10",
    "J", "Q", "K"};
    bool inpvalid;

    for(int i = 0; i < 4; i++) {
        inpvalid = false;
        for(int j = 0; j < 13; j++) {
            if (card[i] == possibleCard[j]) {
```

```

        inval = true;
        break;
    }
}
if (!inval) {
    break;
}
}

return inval;
}

//Mengubah kartu menjadi angka, diperlukan untuk operasi perhitungan
int convertCardToInt(string card) {
    if (card == "A") {
        return 1;
    }
    if (card == "J") {
        return 11;
    }
    if (card == "Q") {
        return 12;
    }
    if (card == "K") {
        return 13;
    }
    else {
        return stoi(card);
    }
}

//Mengubah angka menjadi kartu, digunakan dalam menunjukkan kartu yang dihasilkan
secara random
string convertIntToCard(int number) {
    if (number == 1) {
        return "A";
    }
    if (number == 11) {
        return "J";
    }
    if (number == 12) {
        return "Q";
    }
    if (number == 13) {
        return "K";
    }
}

```

```

    }
    else {
        return to_string(number);
    }
}

//Handle rounding error, khususnya dalam operasi pembagian
bool compareResult(double result) {
    return (result >= (24 - 0.0001) && result <= (24 + 0.0001));
}

int main() {
    string rndm, card1, card2, card3, card4;
    int intCard[4];
    double doubleCard[4];

    //Memastikan jenis input
    cout << "\n\nSelamat Datang di 24 Game Solver\n\n\n";
    cout << "Apakah kartu dirandom atau tidak?(y/n): ";
    cin >> rndm;
    while (rndm != "y" && rndm != "n") { //Error handling untuk input di luar opsi
        cout << "\nApakah kartu dirandom atau tidak?(y/n): ";
        cin >> rndm;
    }
    if (rndm == "y") { //Generate kartu secara random
        srand(time(0));
        for (int rn = 0; rn < 4; rn++) {
            intCard[rn] = (rand() % 13) + 1;
        }
        card1 = convertIntToCard(intCard[0]);
        card2 = convertIntToCard(intCard[1]);
        card3 = convertIntToCard(intCard[2]);
        card4 = convertIntToCard(intCard[3]);
    }
    else { //Generate kartu dari input user
        bool found;
        cout << "Input 4 kartu: ";
        cin >> card1 >> card2 >> card3 >> card4;
        while (validateInput(card1, card2, card3, card4) != 1) { //Error handling
            cout << "\nInput tidak sesuai. Ulangi masukan kembali: ";
            cin >> card1 >> card2 >> card3 >> card4;
        }
        intCard[0] = convertCardToInt(card1);
        intCard[1] = convertCardToInt(card2);
    }
}

```

```

        intCard[2] = convertCardToInt(card3);
        intCard[3] = convertCardToInt(card4);
    }

    cout << "Kartu anda adalah " << card1 << " " << card2 << " " << card3 << " " <<
card4 << "\n\n";

    auto begin = chrono::high_resolution_clock::now();

    doubleCard[0] = intCard[0];
    doubleCard[1] = intCard[1];
    doubleCard[2] = intCard[2];
    doubleCard[3] = intCard[3];

    set<string> solution;    //Menggunakan set agar tidak ada solusi duplikat

    char oprt[4] = {'+', '-', '*', '/'};
    for (int i = 0; i < 4; i++) {    //Loop untuk mencari semua permutasi yang
    mungkin
        for (int j = 0; j < 4; j++) {
            if (i != j) {
                for (int k = 0; k < 4; k++) {
                    if (i != k && j != k) {
                        for (int l = 0; l < 4; l++) {
                            if (i != l && j != l && k != l) {

                                string a = to_string(intCard[i]);
                                string b = to_string(intCard[j]);
                                string c = to_string(intCard[k]);
                                string d = to_string(intCard[l]);

                                double v = doubleCard[i];
                                double w = doubleCard[j];
                                double x = doubleCard[k];
                                double y = doubleCard[l];

                                for (int p = 0; p < 4; p++) {    //Loop untuk
    menentukan operator yang digunakan
                                    for (int q = 0; q < 4; q++) {
                                        for (int r = 0; r < 4; r++) {

                                            //(a b) (c d)
                                            if (compareResult(operation(operation(v,
w, oprt[p]),operation(x, y, oprt[r]), oprt[q]))) {

```



```

        solution.insert("(" + a + oprt[p] + b
+ ")" + oprt[q] + "(" + c + oprt[r] + d + ")");
    }
    //(a b) c) d
    if
(compareResult(operation(operation(operation(v, w, oprt[p]), x, oprt[q]), y,
oprt[r]))) {
        solution.insert("(" + a + oprt[p] +
b + ")" + oprt[q] + c + ")" + oprt[r] + d);
    }
    //(a (b c)) d
    if (compareResult(operation(operation(v,
operation(w, x, oprt[q]), oprt[p]), y, oprt[r]))) {
        solution.insert("(" + a + oprt[p] +
 "(" + b + oprt[q] + c + ")") + oprt[r] + d);
    }
    //a ((b c) d)
    if (compareResult(operation(v,
operation(operation(w, x, oprt[q]), y, oprt[r]), oprt[p]))) {
        solution.insert(a + oprt[p] + "(" +
b + oprt[q] + c + ")" + oprt[r] + d + ")");
    }
    //a (b (c d))
    if (compareResult(operation(v,
operation(w, operation(x, y, oprt[r]), oprt[q]), oprt[p]))) {
        solution.insert(a + oprt[p] + "(" +
b + oprt[q] + "(" + c + oprt[r] + d + ")")");
    }
}
}
}
}
}
}
}
}
}

auto end = chrono::high_resolution_clock::now();
auto duration = chrono::duration_cast<chrono::milliseconds>(end-begin);

//Menampilkan banyak solusi dan solusi yang mungkin
int nSolution = solution.size();
if (nSolution == 0) {

```

```

        cout << "Tidak ada solusi yang mungkin";
    }
    else {
        cout << "Terdapat " << nSolution << " solusi";
    }
    for (auto it = solution.begin(); it != solution.end(); ++it)
        cout << '\n' << *it;

    //Membuat file baru untuk menyimpan solusi
    string save;
    cout << "\n\nApakah ingin menyimpan solusi pada file baru?(y/n): ";
    cin >> save;
    while (save != "y" && save != "n") {
        cout << "\nApakah ingin menyimpan solusi pada file baru?(y/n): ";
        cin >> save;
    }

    if (save == "y") {
        string filename;
        cout << "Masukkan nama file yang ingin dibuat(tanpa .txt): ";
        cin >> filename;

        fstream newfile;
        newfile.open(filename + ".txt", ios::out);
        if (!newfile) {
            cout << "File baru gagal dibuat";
        }
        else {
            cout << "File baru berhasil dibuat";
            if (nSolution == 0) {
                newfile << "Tidak ada solusi yang mungkin";
            }
            else {
                newfile << "Terdapat " << nSolution << " solusi";
            }
            for (auto it = solution.begin(); it != solution.end(); ++it)
                newfile << '\n' << *it;
        }
        newfile.close();
    }

    cout << "\n\nWaktu eksekusi: " << duration.count() << "ms\n\n";

    cout << "\nTerima kasih telah memakai program ini. Sampai jumpa\n\n\n";

```

```
return 0;  
}
```

BAB IV

TEST PROGRAM

1. Testcase 1

```
Selamat Datang di 24 Game Solver

Apakah kartu dirandom atau tidak?(y/n): y
Kartu anda adalah A J 8 3

Terdapat 8 solusi
((11*3)-1)-8
((11*3)-8)-1
((3*11)-1)-8
((3*11)-8)-1
(11*3)-(1+8)
(11*3)-(8+1)
(3*11)-(1+8)
(3*11)-(8+1)

Apakah ingin menyimpan solusi pada file baru?(y/n): y
Masukkan nama file yang ingin dibuat(tanpa .txt): testcase1
File baru berhasil dibuat

Waktu eksekusi: 0ms

Terima kasih telah memakai program ini. Sampai jumpa
```

2. Testcase 2

```
Selamat Datang di 24 Game Solver

Apakah kartu dirandom atau tidak?(y/n): y
Kartu anda adalah 5 8 Q 10

Terdapat 16 solusi
((10+5)-12)*8
((10-12)+5)*8
((5+10)-12)*8
((5-12)+10)*8
(10+(5-12))*8
(10-(12-5))*8
(5+(10-12))*8
(5-(12-10))*8
8*((10+5)-12)
8*((10-12)+5)
8*((5+10)-12)
8*((5-12)+10)
8*(10+(5-12))
8*(10-(12-5))
8*(5+(10-12))
8*(5-(12-10))

Apakah ingin menyimpan solusi pada file baru?(y/n): y
Masukkan nama file yang ingin dibuat(tanpa .txt): testcase2
File baru berhasil dibuat

Waktu eksekusi: 0ms

Terima kasih telah memakai program ini. Sampai jumpa
```

3. Testcase 3

```
Selamat Datang di 24 Game Solver

Apakah kartu dirandom atau tidak?(y/n): y
Kartu anda adalah Q Q 10 3

Tidak ada solusi yang mungkin

Apakah ingin menyimpan solusi pada file baru?(y/n): y
Masukkan nama file yang ingin dibuat(tanpa .txt): testcase3
File baru berhasil dibuat

Waktu eksekusi: 0ms

Terima kasih telah memakai program ini. Sampai jumpa
```

4. Testcase 4

```
Selamat Datang di 24 Game Solver

Apakah kartu dirandom atau tidak?(y/n): n
Input 4 kartu: A 4 6 4
Kartu anda adalah A 4 6 4

Terdapat 10 solusi
 $((1+6)*4)-4$ 
 $((6+1)*4)-4$ 
 $((6-1)*4)+4$ 
 $(4*(1+6))-4$ 
 $(4*(6+1))-4$ 
 $(4*(6-1))+4$ 
 $4+((6-1)*4)$ 
 $4+(4*(6-1))$ 
 $4-((1-6)*4)$ 
 $4-(4*(1-6))$ 

Apakah ingin menyimpan solusi pada file baru?(y/n): y
Masukkan nama file yang ingin dibuat(tanpa .txt): testcase4
File baru berhasil dibuat

Waktu eksekusi: 0ms

Terima kasih telah memakai program ini. Sampai jumpa
```

5. Testcase 5

```
Selamat Datang di 24 Game Solver

Apakah kartu dirandom atau tidak?(y/n): n
Input 4 kartu: K J 3 8
Kartu anda adalah K J 3 8

Tidak ada solusi yang mungkin

Apakah ingin menyimpan solusi pada file baru?(y/n): y
Masukkan nama file yang ingin dibuat(tanpa .txt): testcase5
File baru berhasil dibuat

Waktu eksekusi: 0ms

Terima kasih telah memakai program ini. Sampai jumpa
```

6. Testcase 6

```
Selamat Datang di 24 Game Solver

Apakah kartu dirandom atau tidak?(y/n): n
Input 4 kartu: 4 4 2 2
Kartu anda adalah 4 4 2 2

Terdapat 12 solusi
((2*4)+4)*2
((2*4)-2)*4
((4*2)+4)*2
((4*2)-2)*4
(4+(2*4))*2
(4+(4*2))*2
2*((2*4)+4)
2*((4*2)+4)
2*(4+(2*4))
2*(4+(4*2))
4*((2*4)-2)
4*((4*2)-2)

Apakah ingin menyimpan solusi pada file baru?(y/n): y
Masukkan nama file yang ingin dibuat(tanpa .txt): testcase6
File baru berhasil dibuat

Waktu eksekusi: 0ms

Terima kasih telah memakai program ini. Sampai jumpa
```

LAMPIRAN

Link Github : https://github.com/ChrisAlberth/Tucil1_13521078.git

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil <i>running</i>	✓	
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	✓	