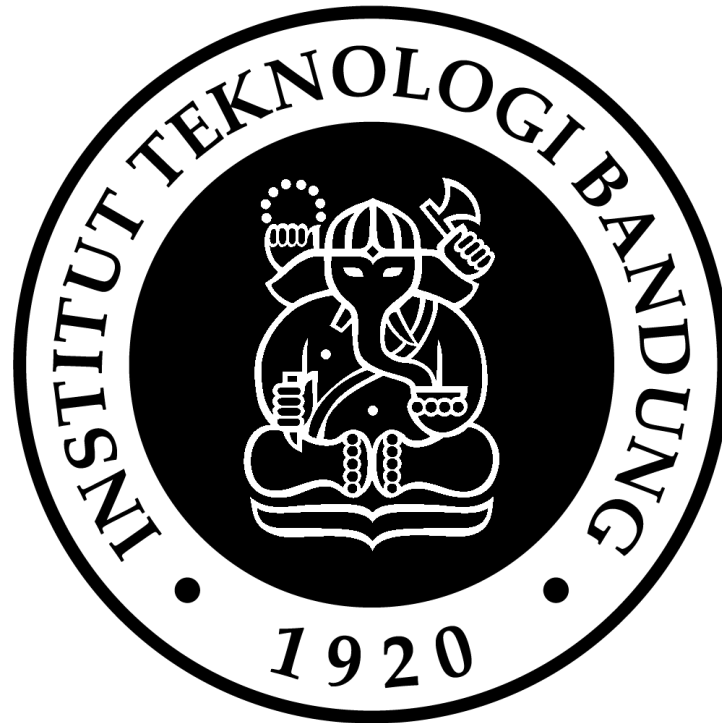


TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA



Disusun Oleh:

| | |
|----------|---------------------------|
| 13521131 | Jeremya Dharmawan Raharjo |
|----------|---------------------------|

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

2023

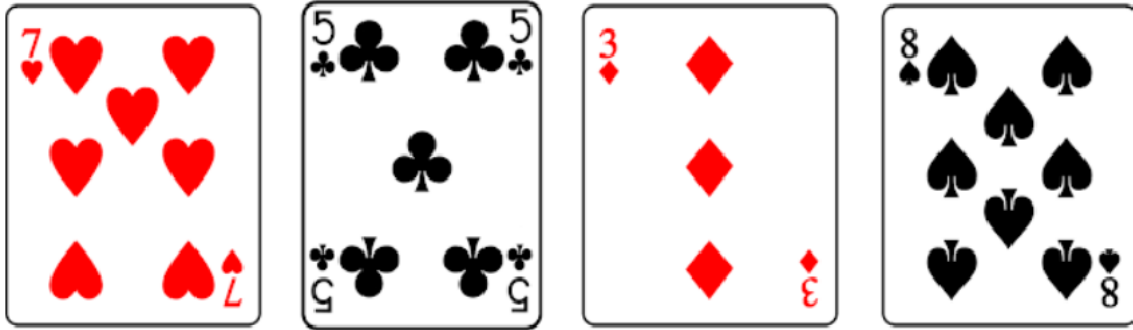
DAFTAR ISI

| | |
|--|-----------|
| DAFTAR ISI | 1 |
| BAB I | 2 |
| 1.1 Deskripsi Masalah | 2 |
| BAB II | 3 |
| 1.1 Definisi General Algoritma Brute Force | 3 |
| 1.2 Penerapan Algoritma Brute Force Dalam Permainan 24 | 3 |
| 1.3 Permutasi Bilangan dan Operator | 4 |
| BAB III | 6 |
| 3.1 Kasus Input User dan Invalid | 6 |
| 3.2 Kasus Input User dan Save File | 6 |
| 3.3 Kasus Tidak Ada Solusi | 7 |
| 3.4 Kasus Input Random Tanpa Menyimpan Hasil | 7 |
| 3.5 Kasus Input User Tanpa Menyimpan Hasil | 8 |
| 3.6 Kasus Input User dan Save File(2) | 8 |
| 3.7 Kasus Tidak Ada Solusi(2) | 9 |
| BAB IV | 10 |
| DAFTAR REFERENSI | 11 |
| LAMPIRAN | 11 |
| i. Source Code (C++) | 12 |
| ii. Tautan Repository | 23 |

BAB I

DESKRIPSI MASALAH

1.1 Deskripsi Masalah



Gambar 1 Permainan Kartu 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

METODE PENYELESAIAN

1.1 Definisi General Algoritma Brute Force

Algoritma *brute force* merupakan pendekatan yang lempang (*straightforward*) untuk memecahkan suatu persoalan. Biasanya algoritma brute force didasarkan pada pernyataan pada persoalan (*problem statement*) dan definisi/konsep yang dilibatkan. Algoritma *brute force* sendiri memecahkan persoalan dengan sangat sederhana, langsung, dan jelas caranya (*obvious way*).

1.2 Penerapan Algoritma Brute Force Dalam Permainan 24

Terdapat beberapa tahapan untuk menerapkan teknik *brute force* dalam permainan 24 ini. tahap. Penyelesaian permasalahan permainan 24 ini melibatkan problema kombinatorial antara angka-angka dan operator-operator sehingga dapat membentuk 24. Berikut ini adalah penjabaran secara dasar mengenai langkah-langkah penyelesaiannya:

- Setiap baris yang akan dievaluasi memiliki tepat 3 buah operasi (terdapat 4 operasi, tambah, kurang, kali, bagi). Dengan kaidah permutasi, jumlah kombinasi yang ada sebesar $4 \times 4 \times 4$ atau 64 kombinasi.
- Setiap baris yang akan dievaluasi memiliki tepat 4 buah angka. Jadi, total banyaknya permutasi angka tanpa berulang pada 4 angka adalah sebanyak $4!$ atau 24.
- Terdapat 10 kemungkinan untuk penempatan kurung dalam mencari solusi 24 seperti yang dijabarkan pada poin-poin berikut (dimana a, b, c, d adalah angka dan op adalah operasi):
 - $(a \text{ op } b) \text{ op } c \text{ op } d$
 - $a \text{ op } (b \text{ op } c) \text{ op } d$
 - $a \text{ op } b \text{ op } (c \text{ op } d)$
 - $(a \text{ op } b \text{ op } c) \text{ op } d$
 - $((a \text{ op } b) \text{ op } c) \text{ op } d$
 - $(a \text{ op } (b \text{ op } c)) \text{ op } d$
 - $a \text{ op } (b \text{ op } c \text{ op } d)$
 - $a \text{ op } ((b \text{ op } c) \text{ op } d)$
 - $a \text{ op } (b \text{ op } (c \text{ op } d))$

- $(a \text{ op } b) \text{ op } (c \text{ op } d)$

Sehingga, kalkulasi dilakukan sebanyak $64 \times 24 \times 10 = 15.360$ (sebagai kemungkinan terburuk pencarian juga).

Pencarian dari solusi-solusi dalam permainan kartu 24 perlu mempersiapkan beberapa hal pada implementasi program. Berikut merupakan penjabaran hal-hal yang perlu diimplementasikan:

1. Siapkan larik dengan tipe data String sebagai masukkan dengan elemen sebanyak 4.
2. Siapkan juga larik dengan tipe data Real(Float) dengan elemen sebanyak 4 setelah string masukkan dikonversi.
3. Siapkan juga prosedur untuk mencari permutasi dari larik yang menampung tipe data real tersebut (implementasi pencarian permutasi dengan prosedur secara rekursi; yang akan menghasilkan 24 kombinasi berbeda)
4. Mempersiapkan prosedur yang akan digunakan untuk melakukan operasi, terdapat 64 kombinasi operasi dengan membagi dua jenis: `single_calculation` sebagai prosedur yang melakukan sebuah operasi dengan 2 buah bilangan (4 kombinasi) dan `double_calculation` sebagai prosedur yang melakukan 2 buah operasi dengan 3 buah bilangan (16 kombinasi).
5. Mengevaluasi tiap-tiap kemungkinan penempatan kurung dengan membuat prosedur yang akan menghasilkan 10 buah kombinasi penempatan kurung yang berbeda.

1.3 Permutasi Bilangan dan Operator

1. Untuk operasi semua permutasi operasi: $[+++]$, $[++-]$, $[++*]$, $[++/]$, $[+-+]$, $[+ -]$, $[+ -*]$, $[+ -/]$, $[+ *+]$, $[+ * -]$, $[+ **]$, $[+ */]$, $[+/+]$, $[+/-]$, $[+/*]$, $[+//]$, $[-+++]$, $[-+-]$, $[-+*]$, $[-+/-]$, $[-+]$, $[- -]$, $[-*]$, $[-/]$, $[-*+]$, $[-*-]$, $[-**]$, $[-*/]$, $[-/+]$, $[-/-]$, $[-/*]$, $[-//]$, $[*++]$, $[*+-]$, $[*+*]$, $[*+/-]$, $[*-+]$, $[*-]$, $[*-*]$, $[*-/]$, $[**+]$, $[**-]$, $[***]$, $[**/]$, $[*/+]$, $[*/-]$, $[*/ *]$, $[*/ /]$, $[/+ +]$, $[/+ -]$, $[/+ *]$, $[/+ /]$, $[- +]$, $[- *]$, $[- /]$, $[/* +]$, $[/* -]$, $[/* *]$, $[/* /]$, $[// +]$, $[// -]$, $[// *]$, $[// /]$
2. Untuk kombinasi angka a, b, c, dan d: $[a, b, c, d]$, $[a, b, d, c]$, $[a, c, b, d]$, $[a, c, d, b]$, $[a, d, b, c]$, $[a, d, c, b]$, $[b, a, c, d]$, $[b, a, d, c]$, $[b, c, a, d]$, $[b, c, d, a]$, $[b, d, a, c]$, $[b, d, c, a]$, $[c, a, b, d]$, $[c, a, d, b]$, $[c, b, a, d]$, $[c, b, d, a]$, $[c, d, a, b]$, $[c, d, b, a]$

a, b, d], [c, a, d, b], [c, b, a, d], [c, b, d, a], [c, d, a, b], [c, d, b, a], [d, a, b, c], [d, a, c, b], [d, b, a, c], [d, b, c, a], [d, c, a, b], [d, c, b, a]

BAB III

EKSPERIMEN

3.1 Kasus Input User dan Invalid

```
=====
====Selamat datang di 24 Game Solver! :)====
=====

Masukkan pilihan anda:
1. Random
2. Input User
3. Exit Program

2

Masukkan 4 angka yang diperlukan:
A J Q 0
Input tidak valid!
```

3.2 Kasus Input User dan Save File

```
Masukkan 4 angka yang diperlukan:
A 2 3 4
Waktu eksekusi: 1 milisekon

Terdapat 464 solusi!
(1*2)*3*4
1*(2*3)*4
1*2*(3*4)
(1+2+3)*4
(1*2*3)*4
((1+2)+3)*4
((1*2)*3)*4
(1+(2+3))*4
```

```

4*(3*(2*1))
4*(3*(2/1))
(4*3)*(2*1)
(4*3)*(2/1)
Ingin menyimpan file? (Y/N)
Y
Masukkan nama file yang ingin disimpan:
stima1.txt
File berhasil disimpan!

```

3.3 Kasus Tidak Ada Solusi

```

Masukkan pilihan anda:
1. Random
2. Input User
3. Exit Program

2

Masukkan 4 angka yang diperlukan:
A A A A
Waktu eksekusi: 1 milisekon
Tidak ada solusi!

```

3.4 Kasus Input Random Tanpa Menyimpan Hasil

```

Masukkan pilihan anda:
1. Random
2. Input User
3. Exit Program

1

Meng-generate random numbers...
Berikut adalah bilangan random anda: 3 8 4 7
Waktu eksekusi: 1 milisekon

Terdapat 12 solusi!

8-(3-7)*4
8-((3-7)*4)

```


3.5 Kasus Input User Tanpa Menyimpan Hasil

```
Masukkan 4 angka yang diperlukan:  
4 10 J K  
Waktu eksekusi: 2 milisekon
```

```
Terdapat 12 solusi!
```

```
4-10*(11-13)  
4-(10*(11-13))  
4+10*(13-11)  
4+(10*(13-11))  
4-(11-13)*10  
4-((11-13)*10)  
4+(13-11)*10  
4+((13-11)*10)  
10*(13-11)+4  
(10*(13-11))+4  
(13-11)*10+4  
((13-11)*10)+4
```

```
Ingin menyimpan file? (Y/N)  
N
```

3.6 Kasus Input User dan Save File(2)

```
Masukkan 4 angka yang diperlukan:  
J Q K A  
Waktu eksekusi: 1 milisekon
```

```
Terdapat 66 solusi!
```

```
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*((13-11)*1)  
12*((13-11)/1)  
12*(13-(11*1))
```

3.7 Kasus Tidak Ada Solusi(2)

Masukkan pilihan anda:

1. Random
2. Input User
3. Exit Program

2

Masukkan 4 angka yang diperlukan:

7 J Q K

Waktu eksekusi: 1 milisekon

Tidak ada solusi!

BAB IV

CHECKLIST

| Poin | Ya | Tidak |
|---|----|-------|
| 1. Program berhasil dikompilasi tanpa kesalahan | ✓ | |
| 2. Program berhasil <i>running</i> | ✓ | |
| 3. Program dapat membaca <i>input</i> / <i>generate</i> sendiri dan memberikan luaran | ✓ | |
| 4. Solusi yang diberikan program memenuhi (berhasil mencapai 24) | ✓ | |
| 5. Program dapat menyimpan solusi dalam file teks | ✓ | |

DAFTAR REFERENSI

Munir, Rinaldi. *Homepage Rinaldi Munir*. Diakses dari
<https://informatika.stei.itb.ac.id/~rinaldi.munir/>

Chandra, Evita. Penerapan Algoritma Brute Force pada Permainan Kartu 24 (24 game) . 2016.
[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/Makalah
Stima-2016-038.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/Makalah%20Stima-2016-038.pdf)

LAMPIRAN

i. Source Code (C++)

```
#include <iostream>
#include <stdlib.h>
#include <string>
#include <cstdlib>
#include <fstream>
#include <ctime>
#include <chrono>
#include <iomanip>

using namespace std;

#define TWENTY_FOUR 24
#define EPSILON 0.001

string buffer[4];
fstream my_file;
float numbers[4];
char opp[4] = {'+', '-', '*', '/'};
int counter24 = 0;
float result_1, result_2, result_3;
char opp_1, opp_2, opp_3;

#define validInput(x) if (abs(x-TWENTY_FOUR) < EPSILON)

uint64_t theTime() {
    using namespace std::chrono;
    return
duration_cast<milliseconds>(system_clock::now().time_since_epo
ch()).count();
}

int rangeInput(int x){
    return (x >= 1 && x <=13);
}

void swap(float *a, float*b){
    float temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

int strToInt(string x){
    if (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 == "10") return 10;
        else if (x == "J") return 11;
        else if (x == "Q") return 12;
        else if (x == "K") return 13;
        else return -1;
    }

    string f_i_str(float y){ //Convert float menjadi string
    integer
        return to_string((int) y);
    }

    void displayOption(){
        cout << "\nMasukkan pilihan anda:\n";
        cout << "1. Random\n";
        cout << "2. Input User\n";
        cout << "3. Exit Program\n\n";
    }

    void single_operator(float x, float y, float *result, int
    num_op, char *op){
        switch (num_op)
        {
            case 0:
                *result = x+y;
                *op = opp[0];
                break;
            case 1:
                *result = x-y;
                *op = opp[1];
                break;
            case 2:
                *result = x*y;
                *op = opp[2];
                break;
            case 3:
                *result = x/y;
                *op = opp[3];
                break;
            default:
                break;
        }
    }

    void double_operator(float x, float y, float z, float *result,
    int num_op, char *op1, char*op2){
        switch (num_op)
        {

```

```

case 0:
*result = x+y+z; // ++
*op1 = opp[0];
*op2 = opp[0];
break;
case 1:
*result = x+y-z; // +-
*op1 = opp[0];
*op2 = opp[1];
break;
case 2:
*result = x+y*z; // +*
*op1 = opp[0];
*op2 = opp[2];
break;
case 3:
*result = x+y/z; // +/
*op1 = opp[0];
*op2 = opp[3];
break;
case 4:
*result = x-y+z; // -+
*op1 = opp[1];
*op2 = opp[0];
break;
case 5:
*result = x-y-z; // --
*op1 = opp[1];
*op2 = opp[1];
break;
case 6:
*result = x-y*z; // -*
*op1 = opp[1];
*op2 = opp[2];
break;
case 7:
*result = x-y/z; // -/
*op1 = opp[1];
*op2 = opp[3];
break;
case 8:
*result = x*y+z; // *+
*op1 = opp[2];
*op2 = opp[0];
break;
case 9:
*result = x*y-z; // *-
*op1 = opp[2];
*op2 = opp[1];
break;
case 10:

```

```

        *result = x*y*z; // **
        *op1 = opp[2];
        *op2 = opp[2];
        break;
    case 11:
        *result = x*y/z; // */
        *op1 = opp[2];
        *op2 = opp[3];
        break;
    case 12:
        *result = x/y+z; // /+
        *op1 = opp[3];
        *op2 = opp[0];
        break;
    case 13:
        *result = x/y-z; // /-
        *op1 = opp[3];
        *op2 = opp[1];
        break;
    case 14:
        *result = x/y*z; // /*
        *op1 = opp[3];
        *op2 = opp[2];
        break;
    case 15:
        *result = x/y/z; // //
        *op1 = opp[3];
        *op2 = opp[3];
        break;
    default:
        break;
    }
}

void theSolution(float arr[],int *state,string *res){
    for(int h = 0; h < 10; h++){
        switch (h)
        {
            /*1 Group of 2*/
            case 0: // (a op b) op c op d
                for(int i = 0; i < 4; i++){

single_operator(arr[0],arr[1],&result_1,i,&opp_1);
                    for(int j = 0; j < 16; j++){

double_operator(result_1,arr[2],arr[3],&result_2,j,&opp_2,&opp_3);

                        validInput(result_2){
                            counter24++;
                            // cout << "(" << arr[0] << opp_1 <<
arr[1] << ")" << opp_2 << arr[2] << opp_3 << arr[3] << "\n";

```



```

        *res = *res + '(' + f_i_str(arr[0]) +
opp_1 + f_i_str(arr[1]) + ')' + opp_2 + f_i_str(arr[2]) +
opp_3 + f_i_str(arr[3]) + '\n';
    }
    }
    break;
case 1: // a op (b op c) op d
    for(int i = 0; i < 4; i++){
single_operator(arr[1],arr[2],&result_1,i,&opp_2);
        for(int j = 0; j < 16; j++){
double_operator(arr[0],result_1,arr[3],&result_2,j,&opp_1,&opp
_3);
            validInput(result_2){
                counter24++;
                // cout << arr[0] << opp_1 << "(" <<
arr[1] << opp_2 << arr[2] << ")" << opp_3 << arr[3] << "\n";
                *res = *res + f_i_str(arr[0]) + opp_1 +
'(' + f_i_str(arr[1]) + opp_2 + f_i_str(arr[2]) + ')' + opp_3
+ f_i_str(arr[3]) + '\n';
            }
        }
    }
    break;
case 2: // a op b op (c op d)
    for(int i = 0; i < 4; i++){
single_operator(arr[2],arr[3],&result_1,i,&opp_3);
        for(int j = 0; j < 16; j++){
double_operator(arr[0],arr[1],result_1,&result_2,j,&opp_1,&opp
_2);
            validInput(result_2){
                counter24++;
                // cout << arr[0] << opp_1 << arr[1] <<
opp_2 << "(" << arr[2] << opp_3 << arr[3] << ")" << "\n";
                *res = *res + f_i_str(arr[0]) + opp_1 +
f_i_str(arr[1]) + opp_2 + '(' + f_i_str(arr[2]) + opp_3 +
f_i_str(arr[3]) + ")" << "\n";
            }
        }
    }
    break;

/*1 Group of 3*/
case 3: // (a op b op c) op d
    for(int i = 0; i < 16; i++){
double_operator(arr[0],arr[1],arr[2],&result_1,i,&opp_1,&opp_2

```

```

);
        for(int j = 0; j < 4; j++){
single_operator(result_1,arr[3],&result_2,j,&opp_3);
        validInput(result_2){
            counter24++;
            // cout << "(" << arr[0] << opp_1 <<
arr[1] << opp_2 << arr[2] << ")" << opp_3 << arr[3] << "\n";
            *res = *res + '(' + f_i_str(arr[0]) +
opp_1 + f_i_str(arr[1]) + opp_2 + f_i_str(arr[2]) + ')' +
opp_3 + f_i_str(arr[3]) + '\n';
        }
    }
    }
    break;
    case 4: //((a op b) op c) op d
        for(int i = 0; i < 4; i++){
single_operator(arr[0],arr[1],&result_1,i,&opp_1);
        for(int j = 0; j < 4; j++){
single_operator(result_1,arr[2],&result_2,j,&opp_2);
        for(int k = 0; k < 4; k++){
single_operator(result_2,arr[3],&result_3,k,&opp_3);
        validInput(result_3){
            counter24++;
            // cout << "(" << arr[0] << opp_1
<< arr[1] << ")" << opp_2 << arr[2] << ")" << opp_3 << arr[3]
<< "\n";
            *res = *res + "(" +
f_i_str(arr[0]) + opp_1 + f_i_str(arr[1]) + ')' + opp_2 +
f_i_str(arr[2]) + ')' + opp_3 + f_i_str(arr[3]) + '\n';
        }
    }
    }
    }
    break;
    case 5: //(a op (b op c)) op d
        for(int i = 0; i < 4; i++){
single_operator(arr[1],arr[2],&result_1,i,&opp_2);
        for(int j = 0; j < 4; j++){
single_operator(arr[0],result_1,&result_2,j,&opp_1);
        for(int k = 0; k < 4; k++){
single_operator(result_2,arr[3],&result_3,k,&opp_3);
        validInput(result_3){
            counter24++;
            // cout << "(" << arr[0] << opp_1

```

```

<< "(" << arr[1] << opp_2 << arr[2] << ")" << opp_3 << arr[3]
<< "\n";
                                *res = *res + '(' + f_i_str(arr[0])
+ opp_1 + '(' + f_i_str(arr[1]) + opp_2 + f_i_str(arr[2]) +
")" + opp_3 + f_i_str(arr[3]) + '\n';
                                }
                                }
                        }
                }
                break;
        case 6: // a op (b op c op d)
                for(int i = 0; i < 16; i++){

double_operator(arr[1],arr[2],arr[3],&result_1,i,&opp_2,&opp_3
);
                                for(int j = 0; j < 4; j++){

single_operator(arr[0],result_1,&result_2,j,&opp_1);
                                validInput(result_2){
                                counter24++;
                                // cout << arr[0] << opp_1 << "(" <<
arr[1] << opp_2 << arr[2] << opp_3 << arr[3] << ")"<< "\n";
                                *res = *res + f_i_str(arr[0]) + opp_1 +
'(' + f_i_str(arr[1]) + opp_2 + f_i_str(arr[2]) + opp_3 +
f_i_str(arr[3]) + ")"<< "\n";
                                }
                                }
                        }
                }
                break;
        case 7: // a op ((b op c) op d)
                for(int i = 0; i < 4; i++){

single_operator(arr[1],arr[2],&result_1,i,&opp_2);
                                for(int j = 0; j < 4; j++){

single_operator(result_1,arr[3],&result_2,j,&opp_3);
                                for(int k = 0; k < 4; k++){

single_operator(arr[0],result_2,&result_3,k,&opp_1);
                                validInput(result_3){
                                counter24++;
                                // cout<< arr[0] << opp_1 << "("
<< arr[1] << opp_2 << arr[2] << ")" << opp_3 << arr[3] <<
")<< "\n";
                                *res = *res + f_i_str(arr[0]) +
opp_1 + "(" + f_i_str(arr[1]) + opp_2 + f_i_str(arr[2]) +
')' + opp_3 + f_i_str(arr[3]) + ")"<< "\n";
                                }
                                }
                        }
                }
}

```

```

        break;
    case 8:// a op (b op (c op d))
        for(int i = 0; i < 4; i++){
            single_operator(arr[2],arr[3],&result_1,i,&opp_3);
            for(int j = 0; j < 4; j++){
                single_operator(arr[1],result_1,&result_2,j,&opp_2);
                for(int k = 0; k < 4; k++){
                    single_operator(arr[0],result_2,&result_3,k,&opp_1);
                    validInput(result_3){
                        counter24++;
                        // cout<< arr[0] << opp_1 << "(" <<
arr[1] << opp_2 << "("<< arr[2] << opp_3 << arr[3] << ")\n";
                        *res = *res + f_i_str(arr[0]) +
opp_1 + "(" + f_i_str(arr[1]) + opp_2 + '(' +
f_i_str(arr[2]) + opp_3 + f_i_str(arr[3]) + ")\n";
                    }
                }
            }
        }
        break;

    /*2 Group of 2*/
    case 9: //(a op b) op (c op d)
        for(int i = 0; i < 4; i++){
            single_operator(arr[0],arr[1],&result_1,i,&opp_1);
            for(int j = 0; j < 4; j++){
                single_operator(arr[2],arr[3],&result_2,j,&opp_3);
                for(int k = 0; k < 4; k++){
                    single_operator(result_1,result_2,&result_3,k,&opp_2);
                    validInput(result_3){
                        counter24++;
                        // cout<< "(" << arr[0] << opp_1
<< arr[1] << ")" << opp_2 << "(" << arr[2]<< opp_3 << arr[3]
<< ")\n";
                        *res = *res + '(' +
f_i_str(arr[0]) + opp_1 + f_i_str(arr[1]) + ')' + opp_2 +
'(' + f_i_str(arr[2]) + opp_3 + f_i_str(arr[3]) + ")\n";
                    }
                }
            }
        }
        break;
    default:
        break;
}

```

```

    }
    *state = counter24;
}

void thePermutation(float arr[], int size, int n, int
*state, string *res){
    if(n == 1){ //Basis
        theSolution(arr, state, res);
    }
    else{ // Rekurens
        float *bufferArr = (float*)malloc((size) *
sizeof(float));
        //Copy Array
        for(int k = 0; k < size; k++){
            bufferArr[k] = arr[k];
        }
        for(int w = size-n; w < size; w++){
            swap(&bufferArr[size-n], &bufferArr[w]);
            thePermutation(bufferArr, size, n-1, state, res);
        }
    }
}

void manual_input(){
    cout << "Masukkan 4 angka yang diperlukan:\n";
    int state = 1;
    for(int i = 0; i<4; i++){
        cin >> buffer[i];
        state &= rangeInput(strToInt(buffer[i]));
        numbers[i] = (float) strToInt(buffer[i]);
    }

    if(state){
        int state_2 = 0;
        string result = "";
        uint64_t tic = theTime();
        thePermutation(numbers, 4, 4, &state_2, &result);
        cout << "Waktu eksekusi: " << theTime()-tic << "
milisekon\n";
        if(state_2){
            cout << "\nTerdapat " << counter24 << " solusi!\n";
            cout << result;
            cout << "Ingin menyimpan file? (Y/N)\n";
            char validate;
            cin >> validate;
            while(1){
                if(validate == 'y' || validate == 'Y'){
                    string w;
                    cout << "Masukkan nama file yang ingin
disimpan: \n";

```

```

        cin >> w;
        my_file.open(w, ios::out);
        if(!my_file) cout<< "File gagal
disimpan!\n";

        else{
            cout << "File berhasil disimpan!\n";
            result = f_i_str(numbers[0]) + " " +
f_i_str(numbers[1]) + " " + f_i_str(numbers[2]) + " " +
f_i_str(numbers[3]) + "\nTerdapat " + to_string(counter24) + "
solusi!\n\n" + result;
            my_file << result;
            my_file.close();
        }
        break;
    }
    else if(validate == 'n' || validate == 'N')
break;
    else{
        cout << "Input tidak valid! Masukkan Y/N
untuk menyimpan file.\n";
        cin >> validate;
    }

    }

    }
    else cout << "Tidak ada solusi!\n";
    }
    else cout << "Input tidak valid!\n";

}

void random_input(){
    cout << "Meng-generate random numbers...\n";
    for(int i = 0; i<4; i++){
        numbers[i] = (float) (1 + rand()%13);
    }
    cout << "Berikut adalah bilangan random anda: ";
    for(int i = 0; i<4; i++) cout << numbers[i] << " ";
    cout << "\n";

    int state = 0;
    string result = "";
    uint64_t tic = theTime();
    thePermutation(numbers,4,4,&state,&result);
    cout << "Waktu eksekusi: " << theTime()-tic << "
milisekon\n";
    if(state) {
        cout << "\nTerdapat " << counter24 << " solusi!\n\n";
        cout << result;
        cout << "Ingin menyimpan file? (Y/N)\n";
        char validate;

```

```

        cin >> validate;
        while(1){
            if(validate == 'y' || validate == 'Y'){
                string w;
                cout << "Masukkan nama file yang ingin
disimpan: \n";
                cin >> w;
                my_file.open(w, ios::out);
                if(!my_file) cout<< "File gagal disimpan!\n";
                else{
                    cout << "File berhasil disimpan!\n";
                    result = f_i_str(numbers[0]) + " " +
f_i_str(numbers[1]) + " " + f_i_str(numbers[2]) + " " +
f_i_str(numbers[3]) + "\nTerdapat " + to_string(counter24) +
"solusi!\n\n" + result;
                    my_file << result;
                    my_file.close();
                }
                break;
            }
            else if(validate == 'n' || validate == 'N') break;
            else{
                cout << "Input tidak valid! Masukkan Y/N untuk
menyimpan file.\n";
                cin >> validate;
            }
        }
    }
    else cout << "Tidak ada solusi!\n";
}

// MAIN PROGRAM
int main(){
    cout << "=====\n";
    cout << "===Selamat datang di 24 Game Solver! :)===\n";
    cout << "=====\n";
    while(1){
        int option;
        displayOption();
        cin >> option;
        cout << '\n';
        switch(option){
            case 1:
                random_input();
                break;
            case 2:
                manual_input();
                break;

```

```
        case 3:
            cout << "Terima kasih telah memakai program
kami!\n";
            break;
        default:
            cout << "Input tidak valid!\n";
            break;
    }

    if(option == 3) break;
}

return 0;
}
```

ii. Tautan Repository

Tautan repository tugas kecil: https://github.com/jejejery/Tucil1_13521131