

LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA
SEMESTER II 2022-2023

Penyelesaian Permainan Kartu 24
dengan Algoritma Brute Force

Disusun oleh:

Bintang Dwi Marthen

13521144



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

1. ALGORITMA

1.1. Algoritma Permutasi Kartu

1. Tukar kartu pada indeks ke-m dengan kartu pada indeks ke-m hingga indeks ke-n (n adalah banyaknya kartu)
2. Setiap kali kartu telah ditukar, lakukan algoritma kembali dengan kedalaman satu lebih dalam (m menjadi $m + 1$) hingga m bernilai $n - 1$
3. Setelah pemanggilan fungsi kembali, tukar kembali kartu yang telah ditukar sebelum pemanggilan fungsi sebagai tindakan *backtracking*

1	1 2 3 4	1 2 3 4	1 2 3 4
		1 3 2 4	1 2 4 3
		1 4 3 2	1 3 2 4
2	2 1 3 4	2 1 3 4	1 3 4 2
		2 3 1 4	1 4 3 2
		2 4 3 1	1 4 2 3
3	3 2 1 4	3 2 1 4	2 1 3 4
		3 1 2 4	2 1 4 3
		3 4 1 2	2 3 1 4
4	4 2 3 1	4 2 3 1	2 3 4 1
		4 3 2 1	2 4 3 1
		4 1 3 2	2 4 1 3

Gambar 1.1 Visualisasi Proses Permutasi Empat Kartu

1.2. Algoritma Pencarian Solusi

1. Lakukan permutasi untuk kartu yang dimainkan
2. Lakukan algoritma *bruteforce* (tahap 3-5) untuk setiap permutasi kartu
3. Lakukan iterasi untuk ketiga operator
4. Lakukan kalkulasi untuk $((a_b)_c)_d$, $((a_b)_c)_d$, $(a_((b_c)_d))$, $(a_((b_c)_d))$, dan $((a_b)_c)_d$ dan periksa apakah menghasilkan 24 (_ akan diganti dengan operator yang diiterasi pada tahap 3)
5. Tambahkan ekspresi ke himpunan solusi bila ekspresi pada tahap 4 menghasilkan angka 24

1.3. Penanganan Solusi Duplikat

Dua ekspresi dikatakan duplikat jika dan hanya jika kedua ekspresi persis penulisannya. Dengan definisi tersebut, maka $(1+2)+(3+4)$, $(2+((1+2)+4))$ dan $(2+1)+(3+4)$ adalah tiga ekspresi berbeda. Melalui definisi solusi duplikat tersebut,

digunakanlah STL set untuk menyimpan solusi sehingga apabila terdapat ekspresi yang persis dengan ekspresi yang telah tercatat pada himpunan tidak akan dicatat lagi.

2. SOURCE CODE DALAM BAHASA C++

2.1. Berkas ioHandler.hpp

```
#ifndef IOHANDLER_HPP
#define IOHANDLER_HPP

#include <iostream>
#include <vector>
#include <set>

extern std::set<std::string> solutions;

int cardToValue(std::string card);

std::string valueToCard(int value);

bool inputValidation(std::string input);

bool cardValid(std::string card);

std::vector<int> userInput();

std::vector<int> randomInput();

void writeToFile();

#endif
```

2.2. Berkas ioHandler.cpp

```
#include "ioHandler.hpp"

#include <sstream>
#include <fstream>

std::set<std::string> solutions;

const std::string validCard[] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};

int cardToValue(std::string card){
    if(card == "A"){
        return 1;
    } else if(card == "J"){
        return 11;
    } else if(card == "Q"){
        return 12;
    } else if(card == "K"){
        return 13;
    } else {
        return std::stoi(card);
    }
}
```

```
std::string valueToCard(int value){
    if(value == 1){
        return "A";
    } else if(value == 11){
        return "J";
    } else if(value == 12){
        return "Q";
    } else if(value == 13){
        return "K";
    } else {
        return std::to_string(value);
    }
}

bool inputValidation(std::string input){
    std::stringstream ss(input);
    std::string card;
    int count = 0;
    while(ss >> card){
        if(!cardValid(card)){
            return false;
        }
        count++;
    }
    if(count != 4){
        return false;
    }
    return true;
}

bool cardValid(std::string card){
    for(int i = 0; i < 13; i++){
        if(card == validCard[i]){
            return true;
        }
    }
    return false;
}

std::vector<int> userInput(){
    std::cout << "Masukkan 4 kartu: ";
    std::string input;
    std::getline(std::cin, input);
    while(!inputValidation(input)){
        std::cout << "Input tidak valid\nMasukkan 4 kartu: ";
        std::getline(std::cin, input);
    }
    std::stringstream ss(input);
    std::string card;
    std::vector<int> cards;
    while(ss >> card){
        cards.push_back(cardToValue(card));
    }
    return cards;
}
```

```
std::vector<int> randomInput(){
    std::vector<int> cards;
    for(int i = 0; i < 4; i++){
        cards.push_back(rand() % 13);
    }
    return cards;
}

void writeToFile(){
    std::ofstream file;
    std::string fileName;
    std::cout << "Masukkan nama file: ";
    std::getline(std::cin, fileName);
    file.open(fileName);
    if(!file.is_open()){
        std::cout << "File tidak dapat diakses" << std::endl;
        return;
    }
    if(solutions.empty()){
        std::cout << "Tidak ada solusi" << std::endl;
        return;
    }
    for (auto i = solutions.begin(); i != solutions.end(); i++){
        file << *i << std::endl;
    }
    file.close();
    std::cout << "Solusi berhasil disimpan ke file " << fileName <<
std::endl;
}
```

2.3. Berkas solver.hpp

```
#ifndef SOLVER_HPP
#define SOLVER_HPP

#include <vector>
#include <iostream>

void permute(std::vector<int> vec, int current);

void solve(std::vector<int> cards);

void brute(std::vector<int> cards);

#endif
```

2.4. Berkas solver.cpp

```
#include "ioHandler.hpp"
#include "solver.hpp"

extern std::set<std::string> solutions;

const char ops[] = {'+', '-', '*', '/'};

bool compare(float a, float b){
    return (a-b < 0.0001 && a-b > -0.0001);
}
```

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

void permute(std::vector<int> vec, int current){
    if(current == vec.size()){
        std::string solution = "";
        for(int i = 0; i < vec.size(); i++){
            solution += valueToCard(vec[i]);
        }
        brute(vec);
    }

    for(int i = current; i < vec.size(); i++){
        swap(&vec[current], &vec[i]);
        permute(vec, current + 1);
        swap(&vec[current], &vec[i]);
    }
}

float calculate(float a, float b, char op){
    if(op == '+'){
        return a+b;
    }
    if(op == '-'){
        return a-b;
    }
    if(op == '*'){
        return a*b;
    }
    if(op == '/'){
        return a/b;
    }
    return 0;
}

void solve(std::vector<int> cards){
    permute(cards, 0);
}

void brute(std::vector<int> cards){
    for(auto op1: ops){
        for(auto op2: ops){
            for(auto op3: ops){
                // (((a b) c) d)
                if(compare(calculate(calculate(calculate(cards[0],
cards[1], op1), cards[2], op2), cards[3], op3), 24.0f)){
                    solutions.insert("(" + valueToCard(cards[0]) + op1 +
valueToCard(cards[1]) + ")" + op2 + valueToCard(cards[2]) + ")" + op3 +
valueToCard(cards[3]) + ")");
                }
            }
        }
    }
}
```

```

// ((a (b c)) d)
    if(compare(calculate(calculate(cards[0],
calculate(cards[1], cards[2], op2), op1), cards[3], op3), 24.0f)){
        solutions.insert("(" + valueToCard(cards[0]) + op1 +
 "(" + valueToCard(cards[1]) + op2 + valueToCard(cards[2]) + ")") + op3 +
 valueToCard(cards[3]) + "));");
    }

    // (a ((b c) d)
    if(compare(calculate(cards[0],
calculate(calculate(cards[1], cards[2], op2), cards[3], op3), op1),
24.0f)){
        solutions.insert("(" + valueToCard(cards[0]) + op1 +
 "(" + valueToCard(cards[1]) + op2 + valueToCard(cards[2]) + ")") + op3 +
 valueToCard(cards[3]) + "));");
    }

    // (a (b (cd)))
    if(compare(calculate(cards[0], calculate(cards[1],
calculate(cards[2], cards[3], op3), op2), op1), 24.0f)){
        solutions.insert("(" + valueToCard(cards[0]) + op1 +
 "(" + valueToCard(cards[1]) + op2 + "(" + valueToCard(cards[2]) + op3 +
 valueToCard(cards[3]) + "))))");
    }

    // ((ab) (cd))
    if(compare(calculate(calculate(cards[0], cards[1], op1),
calculate(cards[2], cards[3], op3), op2), 24.0f)){
        solutions.insert("(" + valueToCard(cards[0]) + op1 +
 valueToCard(cards[1]) + ")" + op2 + "(" + valueToCard(cards[2]) + op3 +
 valueToCard(cards[3]) + "));");
    }
}
}
}

```

2.5. Berkas main.cpp

```
#include "ioHandler.hpp"
#include "solver.hpp"

#include <chrono>

int main(){
    std::cout << "Input dari user atau random? (user/random): ";
    std::string input;
    std::cin >> input;
    while(input != "user" && input != "random"){
        std::cout << "Input tidak valid\nInput dari user atau random?
(user/random): ";
        std::cin >> input;
    }
    std::vector<int> cards;
    std::cin.ignore(256, '\n');
    if(input == "user"){
```

```
        cards = userInput();
    } else {
        cards = randomInput();
        std::cout << "Kartu yang didapat: ";
        for(auto x:cards){
            std::cout << valueToCard(x) << " ";
        }
        std::cout << std::endl;
        std::cout << "Tekan enter untuk melanjutkan...";
        std::cin.get();
    }

    auto start = std::chrono::high_resolution_clock::now();
    solve(cards);
    auto stop = std::chrono::high_resolution_clock::now();
    std::chrono::duration<double, std::milli> duration = stop - start;

    if(solutions.size() == 0){
        std::cout << "Tidak ada solusi" << std::endl;
    } else {
        std::cout << solutions.size() << " solusi ditemukan" << std::endl;
        for(auto x:solutions){
            std::cout << x << std::endl;
        }
    }

    std::cout << "Waktu eksekusi: " << duration.count() << " ms" <<
std::endl;

    std::cout << "Apakah ingin menyimpan hasil ke file? (y/n) ";
    std::cin >> input;
    while(input != "y" && input != "n" && input != "Y" && input != "N"){
        std::cout << "Input tidak valid\nApakah ingin menyimpan hasil ke
file? (y/n) ";
        std::cin >> input;
    }
    std::cin.ignore(256, '\n');
    if(input == "y" || input == "Y"){
        writeToFile();
    }
    std::cout << "Tekan enter untuk melanjutkan...";
    std::cin.get();
    return 0;
}
```


3. HASIL EKSEKUSI PROGRAM

3.1. Kartu Acak

Kartu	Tangkapan Layar
<div>2 7 3 6</div>	<pre> Input dari user atau random? (user/random): random Kartu yang didapat: 2 7 3 6 Tekan enter untuk melanjutkan... 24 solusi ditemukan (((2*7)-6)*3) (((6*7)/2)+3) (((6/2)*7)+3) (((7*2)-6)*3) (((7*6)/2)+3) (((7/2)*6)+3) ((3*7)+(6/2)) ((6*(7/2))+3) ((6/(2/7))+3) ((6/2)+(3*7)) ((6/2)+(7*3)) ((7*(6/2))+3) ((7*3)+(6/2)) ((7/(2/6))+3) (3*((2*7)-6)) (3*((7*2)-6)) (3+((6*7)/2)) (3+((6/2)*7)) (3+((7*6)/2)) (3+((7/2)*6)) (3+(6*(7/2))) (3+(6/(2/7))) (3+(7*(6/2))) (3+(7/(2/6))) Waktu eksekusi: 0.2716 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: random1.txt Solusi berhasil disimpan ke file random1.txt Tekan enter untuk melanjutkan... </pre>

8 9 2 A

```

Input dari user atau random? (user/random): random
Kartu yang didapat: 8 9 2 A
Tekan enter untuk melanjutkan...
38 solusi ditemukan
((2*8)+9)-A
((2*8)-A)+9
((8*2)+9)-A
((8*2)-A)+9
((9-A)*2)+8
(2*(9-A))+8
(2*8)+(9-A)
(2*8)-(A-9)
(8*2)+(9-A)
(8*2)-(A-9)
(8*9)/(2+A)
(8*9)/(A+2)
(8/(2+A))*9
(8/(A+2))*9
(9*8)/(2+A)
(9*8)/(A+2)
(9+(2*8))-A
(9+(8*2))-A
(9-A)+(2*8)
(9-A)+(8*2)
(9/(2+A))*8
(9/(A+2))*8
(8*(9/(2+A)))
(8*(9/(A+2)))
(8+((9-A)*2))
(8+(2*(9-A)))
(8-((A-9)*2))
(8-(2*(A-9)))
(8/((2+A)/9))
(8/((A+2)/9))
(9*(8/(2+A)))
(9*(8/(A+2)))
(9+((2*8)-A))
(9+((8*2)-A))
(9-(A-(2*8)))
(9-(A-(8*2)))
(9/((2+A)/8))
(9/((A+2)/8))
Waktu eksekusi: 0.2392 ms
Apakah ingin menyimpan hasil ke file? (y/n) y
Masukkan nama file: random2.txt
Solusi berhasil disimpan ke file random2.txt
Tekan enter untuk melanjutkan...
  
```

<p>Q 3 6 J</p>	<pre> Input dari user atau random? (user/random): random Kartu yang didapat: Q 3 6 J Tekan enter untuk melanjutkan... 24 solusi ditemukan (((3+Q)-J)*6) (((3-J)+Q)*6) (((J-3)-6)*Q) (((J-6)-3)*Q) (((Q+3)-J)*6) (((Q-J)+3)*6) ((3+(Q-J))*6) ((3-(J-Q))*6) ((J-(3+6))*Q) ((J-(6+3))*Q) ((Q+(3-J))*6) ((Q-(J-3))*6) (6*((3+Q)-J)) (6*((3-J)+Q)) (6*((Q+3)-J)) (6*((Q-J)+3)) (6*(3+(Q-J))) (6*(3-(J-Q))) (6*(Q+(3-J))) (6*(Q-(J-3))) (Q*((J-3)-6)) (Q*((J-6)-3)) (Q*(J-(3+6))) (Q*(J-(6+3))) Waktu eksekusi: 0.2385 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: random3.txt Solusi berhasil disimpan ke file random3.txt Tekan enter untuk melanjutkan... </pre>
<p>6 A 8 K</p>	<pre> Input dari user atau random? (user/random): random Kartu yang didapat: 6 A 8 K Tekan enter untuk melanjutkan... 12 solusi ditemukan (((K-8)-A)*6) (((K-A)-8)*6) ((6-8)*(A-K)) ((8-6)*(K-A)) ((A-K)*(6-8)) ((K-(8+A))*6) ((K-(A+8))*6) ((K-A)*(8-6)) (6*((K-8)-A)) (6*((K-A)-8)) (6*(K-(8+A))) (6*(K-(A+8))) Waktu eksekusi: 0.2238 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: random4.txt Solusi berhasil disimpan ke file random4.txt </pre>

4 10 J Q	Input dari user atau random? (user/random): random Kartu yang didapat: 4 10 J Q Tekan enter untuk melanjutkan... 12 solusi ditemukan $((Q/4)+10)+J$ $((Q/4)+J)+10$ $((10+(Q/4))+J)$ $((10+J)+(Q/4))$ $((J+(Q/4))+10)$ $((J+10)+(Q/4))$ $((Q/4)+(10+J))$ $((Q/4)+(J+10))$ $(10+((Q/4)+J))$ $(10+(J+(Q/4)))$ $(J+((Q/4)+10))$ $(J+(10+(Q/4)))$ Waktu eksekusi: 0.9782 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: random5.txt Solusi berhasil disimpan ke file random5.txt Tekan enter untuk melanjutkan...
----------	--

3.2. Kartu Sesuai Masukkan Pengguna

Kartu	Tangkapan Layar
A 9 9 9	Input dari user atau random? (user/random): user Masukkan 4 kartu: A 9 9 9 Tidak ada solusi Waktu eksekusi: 0.2493 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: user1.txt Tekan enter untuk melanjutkan...
A 3 4 6	Input dari user atau random? (user/random): user Masukkan 4 kartu: A 3 4 6 1 solusi ditemukan $(6/(A-(3/4)))$ Waktu eksekusi: 0.307 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: user2.txt Solusi berhasil disimpan ke file user2.txt Tekan enter untuk melanjutkan...

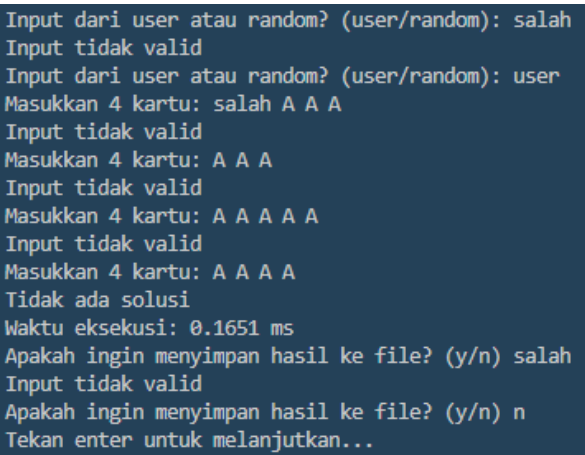
<p>4 9 J J</p>	<pre> Input dari user atau random? (user/random): user Masukkan 4 kartu: 4 9 J J 8 solusi ditemukan (((4*J)-9)-J) (((4*J)-J)-9) (((J*4)-9)-J) (((J*4)-J)-9) ((4*J)-(9+J)) ((4*J)-(J+9)) ((J*4)-(9+J)) ((J*4)-(J+9)) Waktu eksekusi: 0.2818 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: user3.txt Solusi berhasil disimpan ke file user3.txt Tekan enter untuk melanjutkan... </pre>
<p>3 7 9 K</p>	<pre> Input dari user atau random? (user/random): user Masukkan 4 kartu: 3 7 9 K 6 solusi ditemukan ((7*9)-(3*K)) ((7*9)-(K*3)) ((7-(K/3))*9) ((9*7)-(3*K)) ((9*7)-(K*3)) (9*(7-(K/3))) Waktu eksekusi: 0.1967 ms Apakah ingin menyimpan hasil ke file? (y/n) y Masukkan nama file: user4.txt Solusi berhasil disimpan ke file user4.txt Tekan enter untuk melanjutkan... </pre>

2 3 4 5

```

Input dari user atau random? (user/random): user
Masukkan 4 kartu: 2 3 4 5
40 solusi ditemukan
(((3+4)+5)*2)
(((3+5)+4)*2)
(((3+5)-2)*4)
(((3-2)+5)*4)
(((4+3)+5)*2)
(((4+5)+3)*2)
(((5+3)+4)*2)
(((5+3)-2)*4)
(((5+4)+3)*2)
(((5-2)+3)*4)
((3+(4+5))*2)
((3+(5+4))*2)
((3+(5-2))*4)
((3-(2-5))*4)
((4+(3+5))*2)
((4+(5+3))*2)
((5+(3+4))*2)
((5+(3-2))*4)
((5+(4+3))*2)
((5-(2-3))*4)
(2*((3+4)+5))
(2*((3+5)+4))
(2*((4+3)+5))
(2*((4+5)+3))
(2*((5+3)+4))
(2*((5+4)+3))
(2*(3+(4+5)))
(2*(3+(5+4)))
(2*(4+(3+5)))
(2*(4+(5+3)))
(2*(5+(3+4)))
(2*(5+(4+3)))
(4*((3+5)-2))
(4*((3-2)+5))
(4*((5+3)-2))
(4*((5-2)+3))
(4*(3+(5-2)))
(4*(3-(2-5)))
(4*(5+(3-2)))
(4*(5-(2-3)))
Waktu eksekusi: 0.2184 ms
Apakah ingin menyimpan hasil ke file? (y/n) y
Masukkan nama file: user5.txt
Solusi berhasil disimpan ke file user5.txt
Tekan enter untuk melanjutkan...
  
```

3.3. Masukan Tidak Sesuai

Kasus	Tangkapan Layar
Masukan tidak sesuai ketika ditanyakan apakah ingin memasukkan kartu	 <pre> Input dari user atau random? (user/random): salah Input tidak valid Input dari user atau random? (user/random): user Masukkan 4 kartu: salah A A A Input tidak valid Masukkan 4 kartu: A A A Input tidak valid Masukkan 4 kartu: A A A A A Input tidak valid Masukkan 4 kartu: A A A A Tidak ada solusi Waktu eksekusi: 0.1651 ms Apakah ingin menyimpan hasil ke file? (y/n) salah Input tidak valid Apakah ingin menyimpan hasil ke file? (y/n) n Tekan enter untuk melanjutkan... </pre>
Masukan tidak sesuai ketika memasukkan kartu (masukan bukan kartu)	
Masukan tidak sesuai ketika memasukkan kartu (kartu terlalu sedikit)	
Masukan tidak sesuai ketika memasukkan kartu (kartu terlalu banyak)	
Masukan tidak sesuai ketika ditanyakan apakah ingin menyimpan hasil ke file	

4. LAMPIRAN

4.1. Pranala Repository GitHub

https://github.com/Marthenn/Tucil1_13521144

4.2. Tabel Ketercapaian Program

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	✓	