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

Penyelesaian Permainan Kartu 24
dengan Algoritma Brute Force

Disusun oleh:

Tobias Natalio Sianipar 13521090

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

1. ALGORITMA

1.1. Algoritma Solver Game 24

1. Lakukan loop untuk setiap *starting point* dengan cara mengambil 2 kartu acak yang berbeda dari list kartu.
2. Lakukan algoritma *bruteforce* dengan melakukan permutasi 2 kartu dan 1 operator dari semesta 4 kartu dan 4 operator dengan cara *looping* tiap 2 kartu berbeda dan tiap operator.
3. Lakukan rekursi hasil dari kalkulasi 2 kartu tersebut dengan 2 kartu sisanya dan seterusnya sampai kartu habis yang merupakan basis fungsi.
4. Tambahkan ekspresi ke himpunan solusi bila ekspresi pada tahap 3 menghasilkan angka 24

2. SOURCE CODE DALAM BAHASA C++

2.1. Berkas 24solver.cpp

```
#include <stdio.h>
#include <iostream>
#include <deque>
#include <string>
#include <fstream>
#include <ctime>
#include <chrono>




//+ - * /
double operation(double a, double b, int op) {
	double ans = 0;
	switch (op) {
	case 0:
		ans = a + b;
		break;
	case 1:
		ans = a - b;
		break;
	case 2:
		ans = a * b;
		break;
	case 3:
		ans = a / b;
		break;
	}
	return ans;
}

std::string translator(std::string input, std::deque<int> cards) {
	int bracount = 0;
	int var = 0;
	bool lastop = false;
	bool justclosed = false;
```

```
	std::string output = "";
	for (int i = 1; i < input.size(); i++) {

		output += input[i];
		if (bracount > 0) {
			if (lastop) {

				if (input[i] == 'a' || input[i] == 'b' ||
input[i] == 'c' || input[i] == 'd') {
					lastop = false;
					for (int j = 0; j < var; j++) {
						if (bracount > 0) {
							output += ')';
							bracount--;
						}
					}
					var = 0;
					justclosed = true;
				}
			}
			if (input[i] == '(') {
				bracount++;
				lastop = false;
			}
			if (input[i] == '0' || input[i] == '1' || input[i] ==
'2' || input[i] == '3') {
				lastop = true;
			}
			if (input[i] == 'a' || input[i] == 'b' || input[i] ==
'c' || input[i] == 'd') {
				var++;
			}
		} else {
			if (input[i] == '(') {
				bracount++;
			}
		}
	}
	for (int i = 0; i < bracount; i++) {
		output += ")";
	}

	std::string trueoutput = "";
	for (int i = 0; i < output.size(); i++) {
		switch (output[i]) {
			case 'a' :
				trueoutput += std::to_string(cards[0]);
				break;
			case 'b':
				trueoutput += std::to_string(cards[1]);
				break;
			case 'c':
```

```
                        trueoutput += std::to_string(cards[2]);
                        break;
                case 'd':
                        trueoutput += std::to_string(cards[3]);
                        break;
                case '0':
                        trueoutput += '+';
                        break;
                case '1':
                        trueoutput += '-';
                        break;
                case '2':
                        trueoutput += '*';
                        break;
                case '3':
                        trueoutput += '/';
                        break;
                default:
                        trueoutput += output[i];
                }
        }

        return trueoutput;
}

int solve24(std::deque<int> intcards, std::deque<double> cards,
std::deque<std::string> outputs, int& count, std::deque<std::string>&
answers) {

        if (cards.size() == 0) {
                return 0;
        }
        double ans = 0;
        for (int i = 0; i < cards.size(); i++) {
                for (int j = 0; j < cards.size(); j++) {
                        if (j != i) {
                                for (int op = 0; op < 4; op++) {
                                        ans = operation(cards[i], cards[j], op);
                                        std::string result = "(" + outputs[i] +
std::to_string(op) + outputs[j];
                                        if (ans == 24 && cards.size() == 2) {
                                                std::string trueoutput =
translator(result, intcards);
                                                answers.push_front(trueoutput);
                                                std::cout << trueoutput <<
std::endl;
                                                //std::cout << result <<
std::endl;
                                                count++;
                                        }
                                        else {
                                                std::deque<double> tempcards;
```

```
                                        std::deque<std::string>
tempoutputs;

                                        for (int l = 0; l < cards.size();
l++) {
                                              if (l != i && l != j) {

      tempoutputs.push_front(outputs[l]);

      tempcards.push_front(cards[l]);
                                              }
                                        }

                                        tempoutputs.push_front(result);
                                        tempcards.push_front(ans);
                                        solve24(intcards, tempcards,
tempoutputs, count, answers);
                                  }
                            }
                      }
               }
      }
      return 0;
}

int converttoint(std::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);
      }
}

std::string converttocard(int number) {
      if (number == 1) {
            return "A";
      }
      if (number == 11) {
            return "J";
      }
      if (number == 12) {
            return "Q";
```

```
	}
	if (number == 13) {
		return "K";
	}
	else {
		return std::to_string(number);
	}
}

int main() {


	/*std::deque<double> a = { 6,6,3,1 };
	std::deque<std::string> b = { "a","b","c","d" };
	std::deque<std::string> ans;
	std::deque<int> intcards = {6,6,3,1};
	int count = 0;
	solve24(intcards,a,b, count, ans);
	std::cout << count << " " << ans.size();*/
	char option;
	do {
		std::cout << "selamat datang di solver24, ketik '4' untuk
exit. ketik '1' untuk mulai.\n\n" << std::endl;
		std::cin >> option;
		if (option == '1') {
			int A, B, C, D;
			do {

				std::cout << "apakah kartu di random?(y/n)" <<
std::endl;


				std::cin >> option;
				if (option == 'y') {
					std::srand(static_cast<unsigned
int>(std::time(nullptr)));

					A = std::rand() % 12 + 1;
					B = std::rand() % 12 + 1;
					C = std::rand() % 12 + 1;
					D = std::rand() % 12 + 1;

					std::cout << "kartu anda adalah " <<
converttocard(A) << " " << converttocard(B) << " " << converttocard(C) <<
" " << converttocard(D) << std::endl;

				}
				else if (option == 'n') {
					do {
						std::cout << "input 4 kartu" <<
std::endl;
						std::string a, b, c, d;
```

```
                                    std::cin >> a >> b >> c >> d;

                                    A = converttoint(a);
                                    B = converttoint(b);
                                    C = converttoint(c);
                                    D = converttoint(d);

                                    if ((A > 13 || A < 1) || (B > 13
|| B < 1) || (C > 13 || C < 1) || (D > 13 || D < 1)) {
                                          std::cout << "terjadi
kesalahan input." << std::endl << std::endl;
                                    }
                              } while ((A > 13 || A < 1) || (B > 13 ||
B < 1) || (C > 13 || C < 1) || (D > 13 || D < 1));
                        }
                        else if (option == '4') { return 0; }
                        else {
                              std::cout << "terjadi kesalahan input."
<< std::endl << std::endl;
                        }
                  } while (!((option == 'y') || (option == 'n')));
                  std::deque<int> intcards = { A,B,C,D };
                  std::deque<double> doublecards =
{ static_cast<double>(A),static_cast<double>(B),static_cast<double>(C),sta
tic_cast<double>(D) };
                  std::deque<std::string> ans;
                  std::deque<std::string> mark = { "a","b","c","d" };
                  int count = 0;
                  auto started =
std::chrono::high_resolution_clock::now();


                  solve24(intcards, doublecards, mark, count, ans);

                  std::cout << "jumlah solusi yang mungkin sebanyak " <<
count << std::endl;

                  auto done = std::chrono::high_resolution_clock::now();

                  std::cout << "waktu kalkulasi: " <<
std::chrono::duration_cast<std::chrono::milliseconds>(done -
started).count() << "ms\n" << std::endl;

                  do {
                        std::cout << "apakah ingin save jawaban?(y/n) "
<< std::endl;
                        std::cin >> option;

                        if (option == 'y') {
                              std::string nama;
                              std::cout << "tuliskan nama file
penyimpanan(tidak perlu .txt)" << std::endl;
```

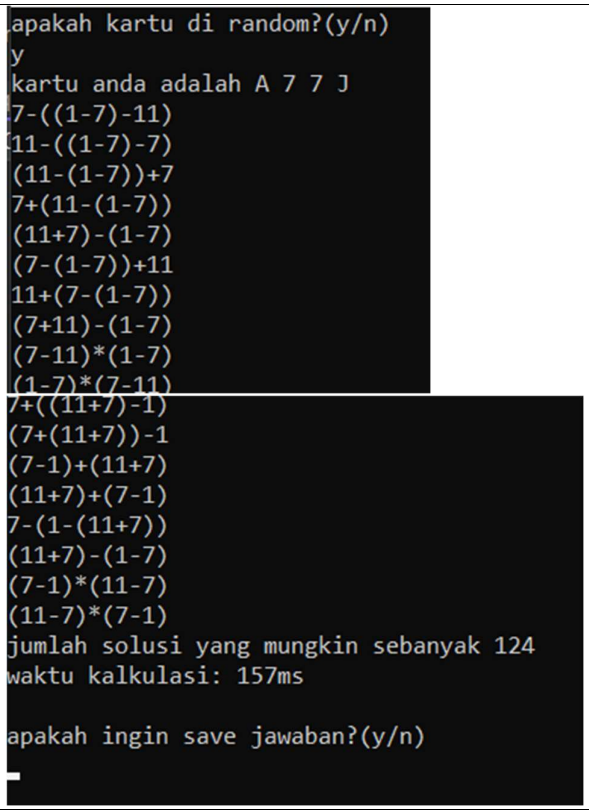
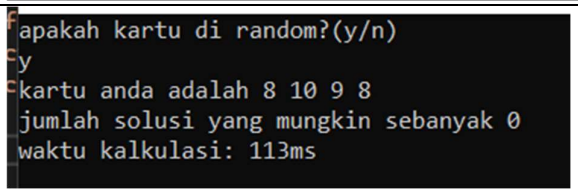
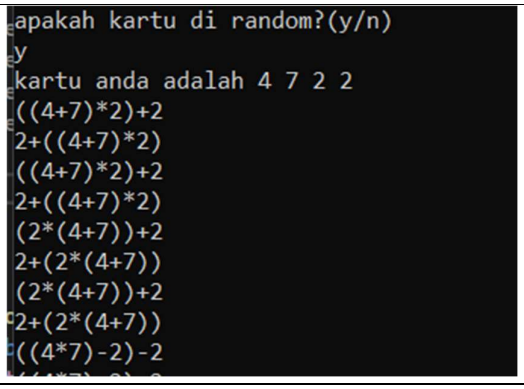
```
                                    std::cin >> nama;
                                    std::fstream mfile;
                                    mfile.open(nama + ".txt", std::ios::out);
                                    if (!mfile) {
                                          std::cout << "File not created!";
                                    }
                                    else {
                                          std::cout << "File created
successfully!\n\n";
                                          mfile << "kartu anda adalah " <<
converttocard(A) << " " << converttocard(B) << " " << converttocard(C) <<
" " << converttocard(D) << std::endl;
                                          for (int i = 0; i < ans.size();
i++) {
                                                mfile << ans[i] <<
std::endl;
                                          }
                                          mfile << "jumlah solusi yang
mungkin sebanyak " << count << std::endl;
                                          mfile.close();
                                    }

                              }
                              else if (option == 'n') {
                                    std::cout << "ok,\n\n";
                              }
                              else {
                                    std::cout << "terjadi kesalahan input."
<< std::endl << std::endl;
                              }
                        } while (!((option == 'y') || (option == 'n')));
                  }
            } while (option != '4');
}
```

3. HASIL EKSEKUSI PROGRAM

3.1. Tes Kartu

1 2 3 4	apakah kartu di random?(y/n) n input 4 kartu 1 2 3 4 ((1+2)+3)*4 4*((1+2)+3) (3+(1+2))*4 4*(3+(1+2)) ((1*2)*4)*3 3*((1*2)*4) ((1*2)*3)*4 (1*(4*3))*2 2*(1*(4*3)) 2/(1/(4*3)) (1*2)*(4*3) (4*3)*(1*2) (4*3)/(1/2) jumlah solusi yang mungkin sebanyak 292 waktu kalkulasi: 209ms apakah ingin save jawaban?(y/n) n program selesai.
9 9 1 1	apakah kartu di random?(y/n) n input 4 kartu 9 9 1 1 jumlah solusi yang mungkin sebanyak 0 waktu kalkulasi: 113ms apakah ingin save jawaban?(y/n)

3 3 6 2	apakah kartu di random?(y/n) y kartu anda adalah 3 3 6 2 ((3+3)-2)*6 6*((3+3)-2) ((3+3)+6)*2 2*((3+3)+6) (6+(3+3))*2 2*(6+(3+3)) (6-2)*(3+3) (3+3)*(6-2) ((3*3)*2)+6 6+((3*3)*2) (2*(3*3))+6 6+(2*(3*3)) ((3+6)+3)*2 2*((3+6)+3) ((2*3)*3)+6 6+((2*3)*3) (6*3)+(2*3) (2*3)+(6*3) (3*(2*3))+6 6+(3*(2*3)) (3*6)+(2*3) (2*3)+(3*6) jumlah solusi yang mungkin sebanyak 104 waktu kalkulasi: 150ms apakah ingin save jawaban?(y/n)

A 7 7 J	
8 10 9 8	
4 7 2 2	

	(7-(2/2))*4 4*(7-(2/2)) jumlah solusi yang mungkin sebanyak 56 waktu kalkulasi: 125ms apakah ingin save jawaban?(y/n)

4. LAMPIRAN

4.1. Pranala *Repository GitHub* https://github.com/tobisns/tucilstima1

4.2. Tabel Ketercapaian Program

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil *running*	✓	
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	✓	