Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2022/2023

Penyelesaian Permainan Kartu 24 dengan Algoritma *Brute Force*

Disusun oleh:

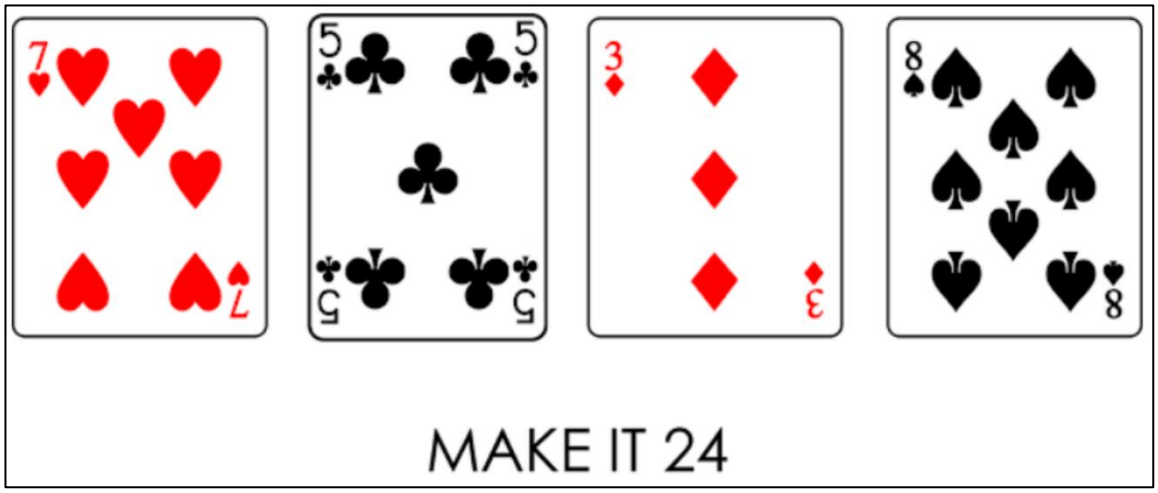
Mohammad Rifqi Farhansyah

13521166

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

Bab 1 Deskripsi Masalah

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. Dengan bermain game 24, kita dapat meningkatkan kecerdasan Anda dan meningkatkan keterampilan menghitung anda. . Sehingga dapat membantu meningkatkan kemampuan matematika anak-anak sekolah dan mengurangi risiko penyakit seperti Alzheimer pada kaum lansia. Permainan ini juga melatih aritmatika mental kita, sehingga bermanfaat untuk semua orang di semua kategori usia. 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 (×), divisi (/) dan tanda kurung (()). Tiap kartu harus digunakan tepat sekali dan urutan penggunaannya bebas. Paragraf tersebut dikutip dari makalah strategi algoritma berjudul “Penerapan Algoritma *Brute Force* pada Permainan Kartu 24 (*24 game*) oleh Evita Chanda (13514034) yang dapat diakses melalui pranala berikut.



Gambar 1.1 Permainan Kartu 24
Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/MakalahStima-2016-038.pdf

Bab 2 Teori Singkat

Landasan teori yang akan digunakan pada penerapan strategi algoritma dalam permainan kartu 24, yaitu: algoritma *Brute Force. Brute Force* merupakan salah satu kajian pada mata kuliah IF2211 – Strategi Algoritma.

A. Definisi Algoritma *Brute Force*

Brute Force merupakan pendekatan yang lemapang (*straight-forward*) dalam hal pemecahan suatu masalah atau persoalan dengan sangat sederhana, langsung, dan jelas (*obvious-way*). Algoritma *Brute Force* seringkali disebut juga sebagai algoritma naif (*naive algorithm*).

B. Karakteristik Algoritma *Brute Force*

Karakteristik algoritma *Brute Force* umumnya tidak mangkus dan sangkil, karena membutuhkan jumlah langkah yang besar dalam penyelesaiannya, sehingga terkadang algoritma *Brute Force* disebut juga sebagai algoritma yang naif. Algoritma *Brute Force* seringkali merupakan pilihan yang kurang disukai karena ketidakmangkusannya itu, tetapi dengan mencari pola-pola yang mendasar, keteraturan, atau trik-trik khusus, biasanya akan membantu kita menemukan algoritma yang lebih cerdas dan lebih mangkus. Berkaitan dengan masalah yang berukuran cukup kecil, kesederhanaan algoritma *Brute Force* biasanya lebih diperhitungkan daripada ketidakmangkusannya. Algoritma *Brute Force* sering digunakan sebagai basis saat membandingkan beberapa alternatif algoritma yang mangkus. Algoritma *Brute Force* seringkali lebih mudah diimplementasikan daripada algoritma-algoritma yang memerlukan penalaran logika lebih mendalam, terkadang algoritma *Brute Force* dapat lebih mangkus (ditinjau dari segi implementasi).

C. Keunggulan dan Kekurangan Algoritma *Brute Force*

Algoritma *Brute Force* memiliki beberapa keunggulan dan kekurangan dibandingkan dengan algoritma-algoritma lainnya. Berikut ini merupakan keunggulan dari algoritma *Brute Force*:

- Metode *Brute Force* dapat digunaan untuk memecahkan hampir sebagian besar masalah.
- Metode *Brute Force* cenderung lebih sederhana dan mudah untuk diimplementasikan.
- Metode *Brute Force* menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, dan perkalian matriks.
- Metode *Brute Force* menghasilkan algoritma baku untuk tugas-tugas komputasi seperti penjumlahan dan perkalian sebuah bilangan, menentukan elemen minimum atau maksimum dalam sebuah senarai, dan lain-lain.

Sementara itu, kelemahan dari algoritma *Brute Force* antara lain:

- Metode *Brute Force* jarang menghasilkan algoritma yang mangkus dan sangkil.
- Beberapa algoritma *Brute Force* lambat sehingga tidak dapat diterima.
- Tidak sekonstruktif/sekreatif teknik pemecahan lain yang serupa.

D. Deskripsi Singkat Permainan Kartu 24

Permainan kartu 24 dimainkan dengan tujuan mencari cara untuk mengubah 4 buah angka random sehingga mendapatkan hasil akhir sejumlah 24 dengan menggunakan beberapa operator bilangan. 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 masingmasing terdiri dari 13 kartu (As, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, dan King). Pada permainan ini jenis suit kartu yang didapat (sekop, hati, keriting, dan wajik) tidak diperhitungkan. 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 salah satu pemain berhasil menebak bagaimana cara mengubah nilai-nilai dari 4 kartu tersebut menjadi 24. Pemain yang paling cepat menebaknya dinyatakan sebagai pemenangnya. Permainan ini sudah dimainkan di Shanghai sejak 1960-an.

Bab 3 Penerapan Algoritma *Brute Force*

A. Algoritma *Brute Force* pada Permainan Kartu 24

Pada permainan kartu 24 ini, solusi dari kombinasi kartu-kartu yang ada dapat diperoleh dengan mengimplementasikan teknik pencarian dengan mencoba semua kemungkinan yang dapat terjadi. Langkah-langkah yang dapat dilakukan dalam memecahkan permainan Kartu 24 dengan metode *Brute Force* adalah sebagai berikut:

- Misalkan terdapat empat buah variabel yang akan mewakili entitas kartu yang didapat (ex : n0, n1, n2, n3).
- Dalam melakukan operasi aritmatika, akan digunakan 3 dari total 4 operator (+,-,*,/) yang nantinya akan digunakan untuk menghubungkan keempat variabel dalam sebuah operasi matematika tertentu.

  n0 op0 n1 op1 n2 op2 n3
- Lalu, tentukan kombinasi dari penggunakan operator tanda kurung atau *bracket* dari keempat variabel tersebut. Total terdapat 5 kemungkinan penyusunan operator tanda kurung pada operasi aritmatika.

  ((n0 op0 n1) op1 n2) op2 n3

  (n0 op0 (n1 op1 n2)) op2 n3

  (n0 op0 n1) op1 (n2 op2 n3)

  n0 op0 ((n1 op1 n2) op2 n3)

  n0 op0 (n1 op1 (n2 op2 n3))
- Kemudian dengan memanfaatkan konsep perulangan, coba semua kombinasi angka dan operator pada operasi aritmatika.

  +++, ++-, ++*, ++/

  +-+, +--, +-*, +-/

  +*+, +*-, +**, +*/

  +/+, +/-, +/*, +//

  -++, -+-, -+*, -+/

  --+, ---, --*, --/

  -*+, -*-, -**, -*/

  -/+, -/-, -/*, -//

  *++, *+-, *+*, *+/

  *-+, *--, *-*, *-/

  +, **-, *, **/

  */+, */-, */*, *//

  /++, /+-, /+*, /+/

  /-+, /--, /-*, /-/

  /*+, /*-, /**, /*/

  //+, //-, //*, ///
- Cek juga seluruh kemungkinan permutasi dari ke-4 angka yang ada, sehingga pada kasus terburuk (*worst case*) total perlu dilakukan 15,360 pencarian untuk menemukan solusi.

B. Implementasi Program pada Bahasa Pemrograman C++

Directory Tree dari source code implementasi program adalah sebagai berikut:

```bash
│  README.md
│
├──.vscode
│      settings.json
│
├──bin
│      main.exe
```

```
│
├───doc
│       Tucil1_K2_13521166_MohammadRifqiFarhansyah.pdf
│
├───image
│       1.png
│       2.png
│       3.png
│       4.png
│       5.png
│
├───src
│       main.cpp
│
└───test
    ├───input
    │       TestInputFile1.txt
    │       TestInputFile2.txt
    │       TestInputFile3.txt
    │       TestInputFile4.txt
    │       TestInputFile5.txt
    │
    └───output
            TestOutputCLI1.txt
            TestOutputCLI2.txt
            TestOutputCLI3.txt
            TestOutputCLI4.txt
            TestOutputCLI5.txt
            TestOutputFile1.txt
            TestOutputFile2.txt
            TestOutputFile3.txt
            TestOutputFile4.txt
            TestOutputFile5.txt
            TestOutputRandom1.txt
            TestOutputRandom2.txt
            TestOutputRandom3.txt
            TestOutputRandom4.txt
            TestOutputRandom5.txt
```

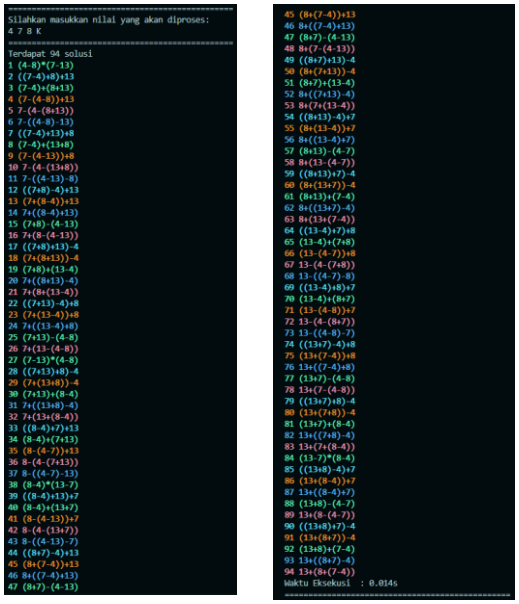
Program terdiri atas satu program utama (main.cpp) yang terdapat pada folder *'src'*. Program utama berisi seluruh alur program yang telah dirancang sedemikian rupa hingga memiliki efektivitas perhitungan yang cukup tinggi. Selain itu, terdapat pula folder *'bin'* yang berisi executable file dari program setelah di-compile dengan MinGW versi 8.1.0. Terdapat pula folder *'doc'* yang berisi laporan Tugas Kecil 1 Strategi Algoritma. Pada folder *'image'* berisi tangkapan layar beberapa fitur utama program. Selain itu, terdapat pula folder *'test'* yang berisi input file txt dalam subfolder *'input'* serta keluaran dari hasil program dalam format file txt di subfolder *'output'*. Untuk lebih lengkapnya, source code dapat diakses melalui pranala berikut ini.

C. Modul Eksternal

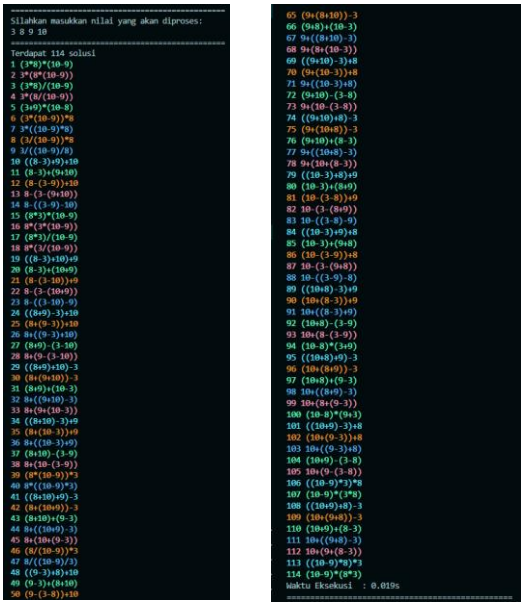
Pada implementasi program digunakan beberapa modul eksternal, seperti *iostream*, *fstream*, *string*, *cstdlib*, dan *ctime*. Setiap modul memiliki fungsi tersendiri yang akan dimanfaatkan dalam program. *Iostream* adalah modul yang digunakan untuk *input/output* standar, seperti memasukkan dan mengeluarkan data dari *keyboard* dan layar. *Fstream* merupakan modul yang digunakan untuk operasi file, seperti membuka, menulis, dan membaca file. String adalah modul yang digunakan untuk manipulasi string, seperti *concatenation*, *substring*, dll. *Cstdlib* adalah modul yang digunakan untuk operasi sistem, seperti *random number generator*, *exit function*, dll. *Ctime* adalah modul yang digunakan untuk operasi waktu, seperti mendapatkan waktu saat ini, mengkonversi waktu ke string, dll.

D. Testing dan Hasil Tangkapan Layar *Input* dan *Output*

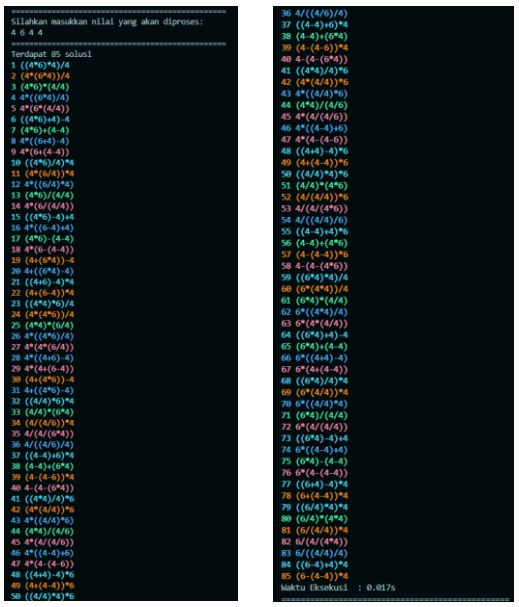
- Ujicoba Pada Input via CLI



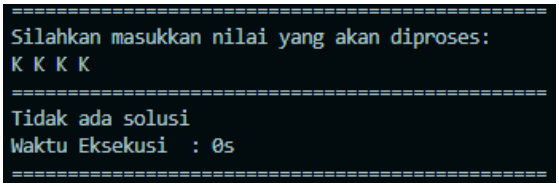
Gambar 3.1.1 Ujicoba CLI (4 7 8 K) : 94 solusi
Sumber: Dokumen Penulis



Gambar 3.1.2 Ujicoba CLI (3 8 9 10) : 114 solusi
Sumber:Dokumen Penulis

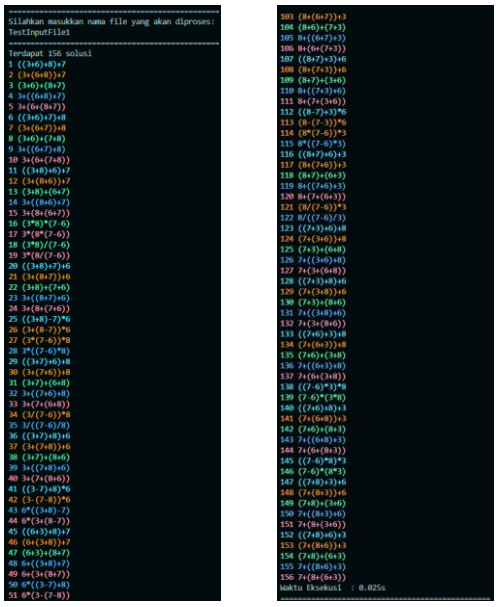


Gambar 3.1.3 Ujicoba CLI (4 6 4 4) : 85 solusi
Sumber: Dokumen Penulis

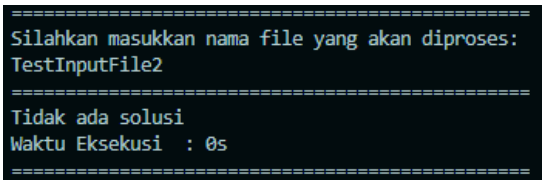


Gambar 3.1.4 Ujicoba CLI (K K K K) : 0 solusi
Sumber:Dokumen Penulis

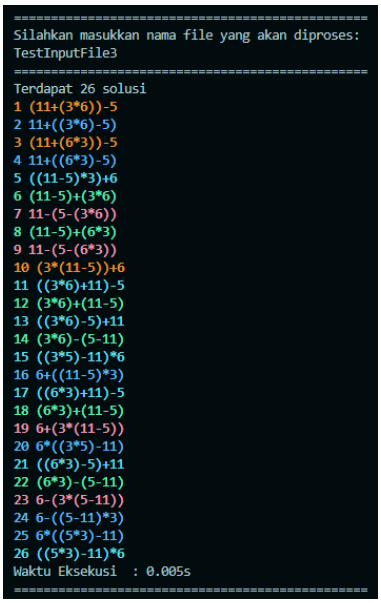
- Ujicoba Pada input via File txt



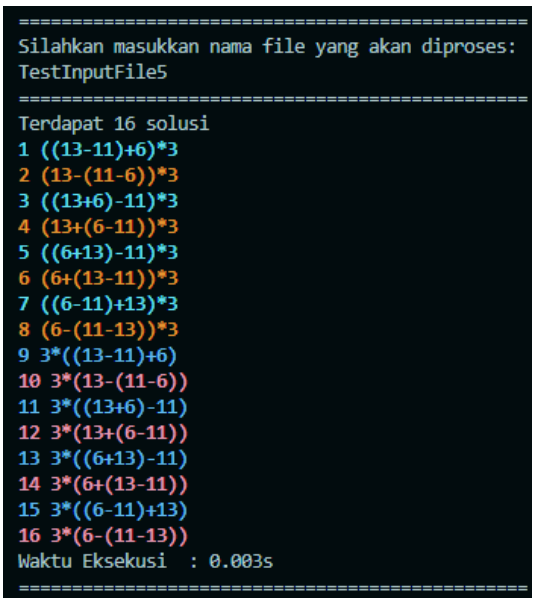
Gambar 3.2.1 Ujicoba File (3 6 8 7) : 156 solusi
Sumber: Dokumen Penulis



Gambar 3.2.2 Ujicoba File (A 8 7 6) : 0 solusi
Sumber:Dokumen Penulis

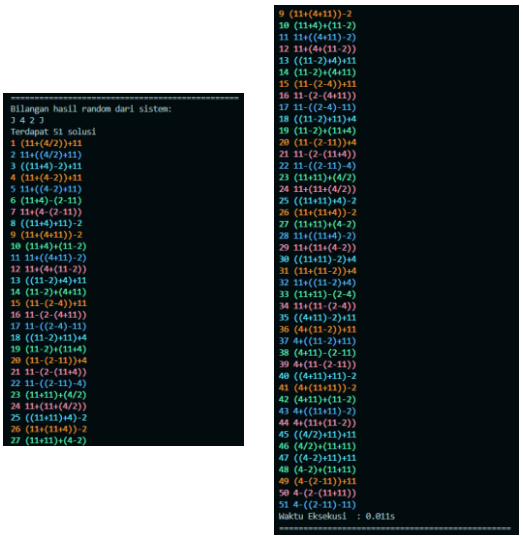


Gambar 3.2.3 Ujicoba File (J 3 6 5) : 26 solusi
Sumber: Dokumen Penulis



Gambar 3.2.4 Ujicoba File (K J 6 3) : 16 solusi
Sumber:Dokumen Penulis

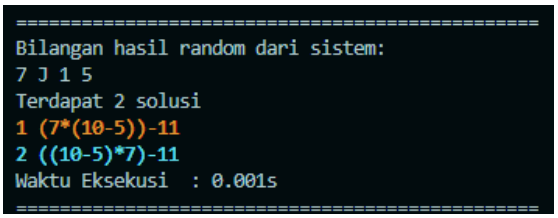
- Ujicoba Pada Input via Random Number oleh Program



Gambar 3.3.1 Ujicoba Random (J 4 2 J) : 51 solusi
Sumber: Dokumen Penulis

Gambar 3.3.2 Ujicoba Random (5 A 6 8) : 16 solusi
Sumber:Dokumen Penulis

Gambar 3.3.3 Ujicoba Random (7 8 8 9) : 6 solusi
Sumber: Dokumen Penulis



Gambar 3.3.4 Ujicoba Random (7 J 1 5) : 2 solusi
Sumber:Dokumen Penulis

Bab 4 Kode Program dalam Bahasa C++

Berikut ini adalah kode program penulis yang dapat menyelesaikan permainan 24 kartu dengan metode Brute Force dan ditulis dalam bahasa pemrograman C++.

```
#include <bits/stdc++.h>
#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
#include <ctime>

using namespace std;

/* INISIALISAI VARIABEL */
char Op[] = {'*','+','/','-'};  string NumString[4]; double Num[4];
double ArrNum[24][4]; string txt; ifstream file; string str; string
txtOut; ofstream fileOutput;

double OpFunc (char op, double a, double b) {
/* Fungsi empat operator aritmatika aritmatika */
  if (op == '+') return (double)a+b;
  else if (op == '-') return (double)a-b;
  else if (op == '/') { if (b != 0) return (double)a/b;}
  else if (op == '*') return (double)a*b;
}

int main () {
  ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0); // Mempercepat
IO
  bool valid = false;
  int no = 0; int n = 0; int solusiCounter = 0; int Hasil = 24; double
EPS = 0.0000000001; bool IsExist; string pilihanMode; string pilihanSave;
int printError=0;
  str = "";

  /* ASCII ART sebagai kode start program */
  cout << "\033[1;31m"
  << R"(
        222222222222222        444444444
      2:::::::::::::::22      4::::::::4
      2::::::222222:::::2    4:::::::::4
      2222222     2:::::2   4::::44::::4
                  2:::::2  4::::4 4::::4
                  2:::::24::::4  4::::4
               2222::::24::::4   4::::4
           22222::::::224::::444444::::444
         22::::::::222  4:::::::::::::::4
      2:::::22222     444444444:::::444
      2:::::2                    4::::4
```

```
      2:::::2                    4::::4
      2:::::2       222222       4::::4
      2::::::2222222:::::2     44::::::44
      2::::::::::::::::::2     4::::::::4
      22222222222222222222     4444444444
  )";
  cout << "\033[0m";
  cout << "\033[1;32m"
              << "G    A   M    E    S    O    L    V    E    R\n"
              << "\033[0m"
              << endl;

  /* Kode untuk memilih jenis masukan */
  while(!valid){
    cout << "================================================" << endl;
    cout<<"Masukkan pilihan masukan:"<<endl;
    cout<<"1. Masukan via CLI"<<endl;
    cout<<"2. Masukan via file (*.txt)"<<endl;
    cout<<"3. Masukan random dari sistem"<<endl;
    valid=true;
    cin >> pilihanMode;
      if(pilihanMode != "1" && pilihanMode != "2" && pilihanMode != "3"){
        valid = false;
        if(printError == 0){
          cout << "Input tidak valid. Silahkan coba kembali."<<endl;
          printError++;
        }
      }
    printError=0;
  }
  valid=false;
  cout << "================================================" << endl;
  switch(pilihanMode[0]){
    case '1':
      while(!valid){
        valid = true;
        cout<<"Silahkan masukkan nilai yang akan diproses: "<<endl;
        for (int i = 0 ; i < 4 ; ++i){
          cin >> NumString[i];
          if(NumString[i] == "2" || NumString[i] == "3" || NumString[i]
== "4" || NumString[i] == "5" || NumString[i] == "6" || NumString[i] ==
"7" || NumString[i] == "8" || NumString[i] == "9" || NumString[i] == "A"
|| NumString[i] == "J" || NumString[i] == "Q" || NumString[i] == "K" ||
NumString[i] == "10"){
            switch(NumString[i][0]){
              case 'A':
                Num[i] = 1;
                break;
              case 'J':
                Num[i] = 11;
```

```
              break;
            case 'Q':
              Num[i] = 12;
              break;
            case 'K':
              Num[i] = 13;
              break;
            default:
              if(NumString[i] == "10"){
                Num[i] = 10;
              }else{
                Num[i] = double(NumString[i][0]) - 48;
              }
              break;
          }
        } else{
          valid = false;
          if(printError == 0){
            cout << "Input tidak valid. Silahkan masukkan karakter yang
valid (A, 2-10, J, Q, K)!"<<endl;
            cout << "================================================="
<< endl;
            printError++;
          }
          for(int i = 0 ; i < 4 ; i ++){
            NumString[i] = "";
            Num[i] = 0;
          }
        }
      }
    }
    cout << "=================================================" << endl;
    printError=0;
    break;
  case '2':
    while(!valid){
      valid = true;
      cout<<"Silahkan masukkan nama file yang akan diproses: "<<endl;
      cin >> txt;
      file.open("../test/input/" + txt + ".txt");
      for(int i = 0 ; i < 4 ; i++){
        file >> NumString[i];
        if(NumString[i] == "2" || NumString[i] == "3" || NumString[i]
== "4" || NumString[i] == "5" || NumString[i] == "6" || NumString[i] ==
"7" || NumString[i] == "8" || NumString[i] == "9" || NumString[i] == "A"
|| NumString[i] == "J" || NumString[i] == "Q" || NumString[i] == "K" ||
NumString[i] == "10"){
          switch(NumString[i][0]){
            case 'A':
              Num[i] = 1;
```

```
              break;
            case 'J':
              Num[i] = 11;
              break;
            case 'Q':
              Num[i] = 12;
              break;
            case 'K':
              Num[i] = 13;
              break;
            default:
              if(NumString[i] == "10"){
                Num[i] = 10;
              }else{
                Num[i] = double(NumString[i][0]) - 48;
              }
              break;
          }
        }
        else{
          valid = false;
          if(printError == 0){
            cout << "Input tidak valid. Silahkan masukkan nama file
yang berisi karakter valid (A, 2-10, J, Q, K)!"<<endl;
            cout << "================================================="
<< endl;
            printError++;
          }
          for(int i=0; i<4; i++)
            NumString[i]="";
            Num[i]=0;
        }
      }
      file.close();
    }
    cout << "=================================================" << endl;
    printError=0;
    break;
  case '3':
    srand(time(0));
    for(int i = 0; i < 4 ; i++){
      int x = rand() % 13 + 1;
      Num[i] = x;
      switch(x){
        case 1:
          NumString[i] = 'A';
          break;
        case 11:
          NumString[i] = 'J';
          break;
```

```
          case 12:
            NumString[i] = 'Q';
            break;
          case 13:
            NumString[i] = 'K';
            break;
          default:
            string conv = to_string(Num[i]);
            NumString[i] = conv[0];
            break;
        }
      }
      cout << "Bilangan hasil random dari sistem: " << endl;
      cout << NumString[0] << " " << NumString[1] << " " << NumString[2]
<< " " << NumString[3] << endl;
  }

  /* Kode untuk mengecek solusi yang ada */
  for (int a = 0 ; a < 4 ; a++){
    for(int b = 0 ; b <4 ; b++) {
      if (b != a){
        for (int c = 0 ; c < 4 ; c++) {
          if ((c != a) && (c != b)) {
            for (int d = 0 ; d < 4 ; d++) {
              if ((d !=a ) && (d != b) && ( d!= c)){
                ArrNum[n][0] = Num[a]; ArrNum[n][1] = Num[b];
                ArrNum[n][2] = Num[c]; ArrNum[n][3] = Num[d]; IsExist =
false;
                for (int f = 0 ; f < n ; f++) {
                  for (int g = 0 ; g < 4 ; g++) {
                    if (g < 3) {
                      if (ArrNum[f][g] != ArrNum[n][g]) break;
                    } else {
                      if (ArrNum[f][g] == ArrNum[n][g]) IsExist = true;
                    }
                  }
                  if (IsExist == true) break;
                }

                if (IsExist == false ) {
                  n++;
                  for (int i = 0 ; i < 4 ; i++) { // Operator pertama
                    for (int j = 0 ; j < 4 ; j++) { // Operator kedua
                      for (int k = 0; k < 4 ; k++) { // Operator ketiga
                      /* Fungsi untuk mengatasi semua kemungkinan bracket
(5 kemungkinan) */
                        if (fabs(OpFunc(Op[k], (OpFunc(Op[j],
(OpFunc(Op[i],Num[a], Num[b])), Num[c])), Num[d]) -Hasil) <= EPS) { //
((n0 op0 n1) op1 n2) op2 d3
                          solusiCounter++;
```

```
                        }
                        if (fabs(OpFunc(Op[k], (OpFunc( Op[i], Num[a],
(OpFunc(Op[j],Num[b], Num[c])) ) ) , Num[d])- Hasil) <= EPS) { // (n0 op0
(n1 op1 n2)) op2 d3
                          solusiCounter++;
                        }
                        if (fabs(OpFunc(Op[j],OpFunc(Op[i], Num[a],
Num[b]), OpFunc(Op[k], Num[c], Num[d])) - Hasil ) <= EPS) { // (n0 op0
n1) op1 (n2 op2 n3)
                          solusiCounter++;
                        }
                        if
(fabs(OpFunc(Op[i],Num[a],OpFunc(Op[k],OpFunc(Op[j],Num[b],Num[c]),
Num[d])) - Hasil) <= EPS) { // n0 op0 ((n1 op1 n2) op2 n3)
                          solusiCounter++;
                        }
                        if
(fabs(OpFunc(Op[i],Num[a],OpFunc(Op[j],Num[b],OpFunc(Op[k],
Num[c],Num[d]))) - Hasil) <= EPS) { // n0 op0 (n1 op1 (n2 op2 n3))
                          solusiCounter++;
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
  if(solusiCounter == 0){
    cout << "Tidak ada solusi" << endl;
    str += "Tidak ada solusi\n";
  } else {
    cout << "Terdapat " << solusiCounter << " solusi" <<endl;
    str += "Terdapat ";
    str += to_string(solusiCounter);
    str += " solusi\n";
  }

  clock_t Mulai = clock(); //Waktu Dimulai
  n=0;
  for (int a = 0 ; a < 4 ; a++){
    for(int b = 0 ; b <4 ; b++) {
      if (b != a){
        for (int c = 0 ; c < 4 ; c++) {
          if ((c != a) && (c != b)) {
            for (int d = 0 ; d < 4 ; d++) {
```

```
            if ((d !=a ) && (d != b) && ( d!= c)){
              ArrNum[n][0] = Num[a]; ArrNum[n][1] = Num[b];
              ArrNum[n][2] = Num[c]; ArrNum[n][3] = Num[d]; IsExist =
false;
              for (int f = 0 ; f < n ; f++) {
                for (int g = 0 ; g < 4 ; g++) {
                  if (g < 3) {
                    if (ArrNum[f][g] != ArrNum[n][g]) break;
                  } else {
                    if (ArrNum[f][g] == ArrNum[n][g]) IsExist = true;
                  }
                }
                if (IsExist == true) break;
              }

              if (IsExist == false ) {
                n++;
                for (int i = 0 ; i < 4 ; i++) { // Operator pertama
                  for (int j = 0 ; j < 4 ; j++) { // Operator kedua
                    for (int k = 0; k < 4 ; k++) { // Operator ketiga
                    /* Fungsi untuk mengatasi semua kemungkinan bracket
(5 kemungkinan) */
                      if (fabs(OpFunc(Op[k], (OpFunc(Op[j],
(OpFunc(Op[i],Num[a], Num[b])), Num[c])), Num[d]) -Hasil) <= EPS) { //
((n0 op0 n1) op1 n2) op2 d3
                        no++; cout<< "\033[1;36m"; cout << no << " "<<
"((" << Num[a] << Op[i] <<  Num[b] << ")" << Op[j]  << Num[c] << ")" <<
Op[k] << Num[d] << endl; cout << "\033[0m";
                        str += to_string(no); str += " "; str += "((";
str += to_string(int(Num[a])); str += Op[i]; str +=
to_string(int(Num[b])); str += ")"; str += Op[j]; str +=
to_string(int(Num[c])); str += ")"; str += Op[k]; str +=
to_string(int(Num[d])); str += "\n";
                      }
                      if (fabs(OpFunc(Op[k], (OpFunc( Op[i], Num[a],
(OpFunc(Op[j],Num[b], Num[c])) ) ) , Num[d])- Hasil) <= EPS) { // (n0 op0
(n1 op1 n2)) op2 d3
                        no++; cout << "\033[1;33m"; cout << no << " "<<
"(" << Num[a] << Op[i] << "("  << Num[b]  << Op[j]  << Num[c] << "))" <<
Op[k] << Num[d] << endl; cout << "\033[0m";
                        str += to_string(no); str += " "; str += "(";
str += to_string(int(Num[a])); str += Op[i]; str += "("; str +=
to_string(int(Num[b])); str += Op[j]; str += to_string(int(Num[c])); str
+= "))"; str += Op[k]; str += to_string(int(Num[d])); str += "\n";
                      }
                      if (fabs(OpFunc(Op[j],OpFunc(Op[i], Num[a],
Num[b]), OpFunc(Op[k], Num[c], Num[d])) - Hasil ) <= EPS) { // (n0 op0
n1) op1 (n2 op2 n3)
```

```
                        no++; cout << "\033[1;32m"; cout << no << " "<<
"(" << Num[a] << Op[i] << Num[b] << ")" << Op[j] << "(" << Num[c] <<
Op[k] << Num[d] << ")" << endl; cout << "\033[0m";
                        str += to_string(no); str += " "; str += "(";
str += to_string(int(Num[a])); str += Op[i]; str +=
to_string(int(Num[b])); str += ")"; str += Op[j]; str += "("; str +=
to_string(int(Num[c])); str += Op[k]; str += to_string(int(Num[d])); str
+= ")"; str += "\n";
                    }
                    if
(fabs(OpFunc(Op[i],Num[a],OpFunc(Op[k],OpFunc(Op[j],Num[b],Num[c]),
Num[d])) - Hasil) <= EPS) { // n0 op0 ((n1 op1 n2) op2 n3)
                        no++; cout << "\033[1;34m"; cout << no <<" " <<
Num[a] << Op[i] << "((" << Num[b] << Op[j] << Num[c] << ")" << Op[k]
<< Num[d] << ")" << endl; cout << "\033[0m";
                        str += to_string(no); str += " "; str +=
to_string(int(Num[a])); str += Op[i]; str += "(("; str +=
to_string(int(Num[b])); str += Op[j]; str += to_string(int(Num[c])); str
+= ")"; str += Op[k]; str += to_string(int(Num[d])); str += ")"; str +=
"\n";
                    }
                    if
(fabs(OpFunc(Op[i],Num[a],OpFunc(Op[j],Num[b],OpFunc(Op[k],
Num[c],Num[d]))) - Hasil) <= EPS) { // n0 op0 (n1 op1 (n2 op2 n3))
                        no++; cout << "\033[1;35m"; cout << no << " "
<< Num[a] << Op[i] << "(" << Num[b] << Op[j] << "(" << Num[c] << Op[k]
<< Num[d] << "))" << endl; cout << "\033[0m";
                        str += to_string(no); str += " "; str +=
to_string(int(Num[a])); str += Op[i]; str += "("; str +=
to_string(int(Num[b])); str += Op[j]; str += "("; str +=
to_string(int(Num[c])); str += Op[k]; str += to_string(int(Num[d])); str
+= "))"; str += "\n";
                    }
                }
            }
        }
    }
  }

  /* Kode untuk menghitung waktu eksekusi */
  cout << "\033[0m";
  cout << "Waktu Eksekusi\t: " << (double)(clock() -
Mulai)/CLOCKS_PER_SEC << "s\n";
  str += "Waktu Eksekusi\t: ";
```

```
  str += to_string(double(clock()-Mulai)/CLOCKS_PER_SEC);
  str += "s";
  str += "\n";

  /* Kode untuk menampilkan kartu masukan */
  str += "Kartu Masukan\t: ";
  for(int i = 0 ; i < 4 ; i++){
    str += NumString[i];
    str += " ";
  }
  str += "\n";

  /* Kode untuk menyimpan hasil */
  valid = false;
  while(!valid){
    cout << "================================================" << endl;
    cout << "Apakah hasil ingin disimpan? (y/n): " << endl;
    cin >> pilihanSave;
    valid = true;
    if(pilihanSave != "y" && pilihanSave != "n"){
      valid = false;
      if(printError==0){
        cout << "Input tidak valid. Silahkan coba kembali." << endl;
        printError++;
      }
    }
    printError = 0;
  }
  cout << "================================================" << endl;
  switch(pilihanSave[0]){
    case 'y':
      cout << "Masukkan nama file untuk keluaran:" << endl;
      cin >> txtOut;
      cout << "================================================" << endl;
      fileOutput.open("../test/output/" + txtOut +".txt",std::ios::out);
      if(fileOutput.is_open()){
        fileOutput << str << endl;
        cout << "File output berhasil disimpan" << endl;
        cout << "================================================" <<
endl;
      }
      else{
        cout << "Gagal untuk melakukan proses penyimpanan" << endl;
        cout << "================================================" <<
endl;
      }
      break;
    case 'n':
      cout << "File tidak disimpan" << endl;
      cout << "================================================" << endl;
```

```
        break;
    }

    /* Kode untuk menutup program */
    cout << "\033[1;32m";
    cout << " T    H    A    N    K    Y    O    U    !    !"<<endl;
    cout << "  For using our app to solve your 24-Game :))  "<<endl;
    cout << "\033[0m";
    return 0;
}
```

Repository program ini dapat diakses melalui pranala berikut ini.

Bab 5 Tabel Penilaian

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpakesalahan	√	
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.	√	

Bab 6 Kesimpulan

Dalam melakukan penyelesaian permainan 24 Kartu dengan komputasi terdapat berbagai macam metode yang dapat diaplikasikan. Salah satunya adalah dengan algoritma *Brute Force*, yaitu: dengan mencoba semua kemungkinan operasi aritmatika yang dibentuk oleh 4 komponen kartu (dengan nilai bervariasi), 3 komponen operator, serta 5 variasi penyusunan tanda kurung (*bracket*).

Pada Tugas Kecil 1 Strategi Algoritma ini, penulis membuat sebuah aplikasi berbasis *Command Line Interface* untuk menyelesaikan permainan 24 Kartu yang ditulis dalam bahasa pemrograman C++. Aplikasi penulis berhasil melakukan penyelesaian permainan dengan cukup baik, walaupun masih memiliki tingkat efektifitas dan efisiensi yang belum begitu baik (waktu eksekusi program berkisar antara 0.001s – 0.020s).

Bab 7 Referensi

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/stima21-22.htm (diakses pada 20 Januari 2023)
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tucil1-Stima-2023.pdf (diakses pada 19 Januari 2023)
https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/MakalahStima-2016-038.pdf (diakses pada 19 Januari 2023)