LAPORAN TUGAS KECIL I

IF2211 STRATEGI ALGORITMA

PENYELESAIAN PERMAINAN KARTU 24 DENGAN ALGORITMA *BRUTE FORCE*

Disusun oleh :

Bill Clinton / 13521064

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022/2023

DAFTAR ISI

1. Algoritma *Brute Force* untuk Penyelesaian Permainan Kartu 24

Secara umum, algoritma *brute force* adalah algoritma yang didasarkan oleh pendekatan yang *straightforward* untuk menyelesaikan suatu permasalahan. Dengan menggunakan algoritma ini, seseorang dapat menyelesaikan permasalahan dengan cara yang langsung dan sangat sederhana.

Permainan kartu 24 sendiri merupakan permainan kartu dengan tujuan untuk mencari berbagai cara untuk mengoperasikan 4 buah angka acak yang nilainya diambil dari kartu-kartu dalam permainan kartu remi, dengan menggunakan operasi-operasi matematika (“+”, “-”, “*”, “/”, “(“, “)”) untuk menghasilkan nilai 24. Kartu-kartu dalam permainan kartu remi sendiri terdiri dari empat jenis, yakni hati, sekop, wajik, dan keriting. Namun, yang diperhatikan dalam permainan ini hanya nilai kartunya saja dan bukan jenisnya. Nilai kartu remi tersebut terdiri A, J, Q, K, 2, 3, 4, 5, 6, 7, 8, 9, dan 10. A bernilai 1, J bernilai 11, Q bernilai 12, dan K bernilai 13. Permainan kartu 24 ini sangat cocok untuk mengasah kemampuan otak kita.

Permainan kartu ini bisa diselesaikan dengan algoritma *brute force* yang telah dijelaskan sebelumnya. Ada beberapa langkah untuk bisa merealisasikannya. Pertama, tentu kita harus mendapatkan nilai yang ingin kita olah untuk dijadikan 24. Dalam program yang penulis buat, pengguna bisa memasukkan empat angka tersebut secara manual atau meminta empat angka acak. Program akan menghasilkan seluruh permutasi baik permutasi untuk angka yang akan diolah dan permutasi operator matematika yang terdiri dari pertambahan, pengurangan, perkalian, dan pembagian. Selanjutnya akan dicek semua kemungkinan penempatan kurung yang mungkin dalam loop permutasi angka dan permutasi operator matematika. Jika hasil dari operasi sama dengan tujuan permainan ini yaitu 24, jumlah solusi akan bertambah dan solusi tersebut disimpan untuk kemudian bisa ditulis ke *file* yang diinginkan. Dalam program yang penulis buat, setelah program berhasil mendapatkan seluruh solusi, program akan menawarkan pengguna untuk menyimpan solusi dalam *file* (pengguna bebas untuk menentukan nama *file*). Setelah itu, program akan menampilkan waktu eksekusi algoritma *brute force* ini. Terakhir, program akan memberikan pilihan kepada pengguna untuk memulai ulang program atau menghentikan program.

2. *Source Code*

Untuk tugas kecil I IF2211 Strategi Algoritma ini, penulis memilih untuk menggunakan bahasa pemrograman C++. Berikut *source code* program untuk penyelesaian permainan kartu 24 ini.

```cpp
// 24-Game Solver
#include <iostream>
#include <vector>
#include <fstream>
#include <ctime>

using namespace std;

string intToStr (int a) {
    return (to_string(a));
}

float compute (float a, float b, char op) {
    if (op == '+') {
        return (a + b);
    } else if (op == '*') {
        return (a * b);
    } else if (op == '-') {
        return (a - b);
    } else if (op == '/') {
        if (b != 0) {
            return (a / b);
        } else {
            return 99999999;
        }
    }
    return 0;
}
```

```cpp
vector<vector<int>> numPermutations(vector<int> num) {
    vector<vector<int>> permutations;
    for(int i = 0; i <= 3; i++) {
        for(int j = 0; j <= 3; j++) {
            if(i != j) {
                for(int k = 0; k <= 3; k++) {
                    if(j != k and k != i) {
                        for(int h = 0; h <= 3; h++) {
                            if(h != i and h != j and h != k) {
                                vector<int> temp;
                                temp.push_back(num[i]);
                                temp.push_back(num[j]);
                                temp.push_back(num[k]);
                                temp.push_back(num[h]);
                                permutations.push_back(temp);
                            }
                        }
                    }
                }
            }
        }
    }
    return permutations;
}
```

```cpp
vector<vector<char>> opPermutations(vector<char> op) {
    vector<vector<char>> permutations;
    for(int i = 0; i <= 3; i++) {
        for(int j = 0; j <= 3; j++) {
            for(int k = 0; k <= 3; k++) {
                vector<char> temp;
                temp.push_back(op[i]);
                temp.push_back(op[j]);
                temp.push_back(op[k]);
                permutations.push_back(temp);
            }
        }
    }
    return permutations;
}
```

```cpp
void startSolver() {
    int inputCount = 0;
    int answerCount = 0;
    int goal = 24;
    bool valid;

    string userInput;
    vector<int> numInput;
    vector<char> op = {'+', '-', '*', '/'};
    vector<string> result;
    bool validInputChoice = false;
    int inputChoice;
    string validInput[13] = {"A", "J", "Q", "K", "2", "3", "4", "5", "6", "7", "8", "9", "10"};

    cout << "========================o======================== \n";
    cout << "Selamat datang di 24-Game Solver! \n";
    cout << "Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10! \n";
    cout << "Keterangan: \n" << "A bernilai 1 \n" << "J bernilai 11 \n" << "Q bernilai 12 \n" << "K bernilai 13 \n\n";
```

```cpp
    // Choose Whether User Wants to Input Number Manually or Generate Random Numbers
    while (!validInputChoice) {
        cout << "Pilih cara pemilihan angka-angka! \n";
        cout << "1. Manual Input \n";
        cout << "2. Random \n";
        cout << "Ketik (1/2) : ";
        cin >> inputChoice;
```

```
if (inputChoice == 1) {
    cout << "Masukkan 4 Kartu (pisahkan dengan spasi atau enter) : ";
    while (inputCount < 4) {
        valid = true;
        cin >> userInput;
        if (userInput == "A") {
            numInput.push_back(1);
        } else if (userInput == "J") {
            numInput.push_back(11);
        } else if (userInput == "Q") {
            numInput.push_back(12);
        } else if (userInput == "K") {
            numInput.push_back(13);
        } else if (userInput == "2") {
            numInput.push_back(2);
        } else if (userInput == "3") {
            numInput.push_back(3);
        } else if (userInput == "4") {
            numInput.push_back(4);
        } else if (userInput == "5") {
            numInput.push_back(5);
        } else if (userInput == "6") {
            numInput.push_back(6);
        } else if (userInput == "7") {
            numInput.push_back(7);
        } else if (userInput == "8") {
            numInput.push_back(8);
        } else if (userInput == "9") {
            numInput.push_back(9);
        } else if (userInput == "10") {
            numInput.push_back(10);
        } else {
            valid = false;
        }
```

```
            if (valid) {
                inputCount++;
            } else {
                cout << "Input tidak valid, silakan input ulang kartu yang tidak valid! \n";
                cout << "Kartu yang valid adalah A, J, Q, K, dan 2 hingga 10! \n";
            }
        }
        validInputChoice = true;
```

```
} else if (inputChoice == 2) {
    cout << "Hasil generate kartu random : ";
    for(int i = 0; i <= 3; i++) {
        int randomIndex = rand() % 13;
        cout << validInput[randomIndex] << " ";
        if (validInput[randomIndex] == "A") {
            numInput.push_back(1);
        } else if (validInput[randomIndex] == "J") {
            numInput.push_back(11);
        } else if (validInput[randomIndex] == "Q") {
            numInput.push_back(12);
        } else if (validInput[randomIndex] == "K") {
            numInput.push_back(13);
        } else if (validInput[randomIndex] == "2") {
            numInput.push_back(2);
        } else if (validInput[randomIndex] == "3") {
            numInput.push_back(3);
        } else if (validInput[randomIndex] == "4") {
            numInput.push_back(4);
        } else if (validInput[randomIndex] == "5") {
            numInput.push_back(5);
        } else if (validInput[randomIndex] == "6") {
            numInput.push_back(6);
        } else if (validInput[randomIndex] == "7") {
            numInput.push_back(7);
        } else if (validInput[randomIndex] == "8") {
            numInput.push_back(8);
        } else if (validInput[randomIndex] == "9") {
            numInput.push_back(9);
        } else if (validInput[randomIndex] == "10") {
            numInput.push_back(10);
        }
    }
    cout << "\n";
```

```
            validInputChoice = true;
        } else {
            cout << "Input pilihan tidak valid, silakan ulangi! \n\n";
        }
    }
```

```
vector<vector<int>> allNumPossibilities = numPermutations(numInput);
vector<vector<char>> allOpPossibilities = opPermutations(op);
clock_t start, end;

// Algorithm to Find All Solutions
start = clock();

for(int i = 0; i <= 23; i++) {
    for(int j = 0; j <= 63; j++) {
        float possibility_1 = compute(allNumPossibilities[i][0], compute(allNumPossibilities[i][1], compute(allNumPossibilities[i][2],
        allNumPossibilities[i][3], allOpPossibilities[j][2]), allOpPossibilities[j][1]), allOpPossibilities[j][0]);
        float possibility_2 = compute(allNumPossibilities[i][0], compute(compute(allNumPossibilities[i][1], allNumPossibilities[i][2],
        allOpPossibilities[j][1]), allNumPossibilities[i][3], allOpPossibilities[j][2]), allOpPossibilities[j][0]);
        float possibility_3 = compute(compute(allNumPossibilities[i][0], compute(allNumPossibilities[i][1], allNumPossibilities[i][2],
        allOpPossibilities[j][1]), allOpPossibilities[j][0]), allNumPossibilities[i][3], allOpPossibilities[j][2]);
        float possibility_4 = compute(compute(allNumPossibilities[i][0], allNumPossibilities[i][1], allOpPossibilities[j][0]), compute
        (allNumPossibilities[i][2], allNumPossibilities[i][3], allOpPossibilities[j][2]), allOpPossibilities[j][1]);
        float possibility_5 = compute(compute(allNumPossibilities[i][0], allNumPossibilities[i][1], allOpPossibilities[j][0]),
        allNumPossibilities[i][2], allOpPossibilities[j][1]), allNumPossibilities[i][3], allOpPossibilities[j][2]);

        // Check whether the result is the same as the goal, which is 24
        if (possibility_1 == 24) {
            string eq = intToStr(allNumPossibilities[i][0]) + allOpPossibilities[j][0] + "(" + intToStr(allNumPossibilities[i][1]) +
            allOpPossibilities[j][1] + "(" + intToStr(allNumPossibilities[i][2]) + allOpPossibilities[j][2] + intToStr(allNumPossibilities[i]
            [3]) + "))";
            (result).push_back(eq);
            answerCount++;
            // cout << eq << "\n";
        }
```

```
        if (possibility_2 == 24) {
            string eq = intToStr(allNumPossibilities[i][0]) + allOpPossibilities[j][0] + "((" + intToStr(allNumPossibilities[i][1]) +
            allOpPossibilities[j][1] + intToStr(allNumPossibilities[i][2]) + ")" + allOpPossibilities[j][2] + intToStr(allNumPossibilities[i]
            [3]) + ")";
            (result).push_back(eq);
            answerCount++;
            // cout << eq << "\n";
        }
        if (possibility_3 == 24) {
            string eq = "(" + intToStr(allNumPossibilities[i][0]) + allOpPossibilities[j][0] + "(" + intToStr(allNumPossibilities[i][1]) +
            allOpPossibilities[j][1] + intToStr(allNumPossibilities[i][2]) + "))" + allOpPossibilities[j][2] + intToStr(allNumPossibilities[i]
            [3]);
            (result).push_back(eq);
            answerCount++;
            // cout << eq << "\n";
        }
        if (possibility_4 == 24) {
            string eq = "(" + intToStr(allNumPossibilities[i][0]) + allOpPossibilities[j][0] + intToStr(allNumPossibilities[i][1]) + ")" +
            allOpPossibilities[j][1] + "(" + intToStr(allNumPossibilities[i][2]) + allOpPossibilities[j][2] + intToStr(allNumPossibilities[i]
            [3]) +")";
            (result).push_back(eq);
            answerCount++;
            // cout << eq << "\n";
        }
        if (possibility_5 == 24) {
            string eq = "((" + intToStr(allNumPossibilities[i][0]) + allOpPossibilities[j][0] + intToStr(allNumPossibilities[i][1]) + ")" +
            allOpPossibilities[j][1] + intToStr(allNumPossibilities[i][2]) + ")" + allOpPossibilities[j][2] + intToStr(allNumPossibilities[i]
            [3]);
            (result).push_back(eq);
            answerCount++;
            // cout << eq << "\n";
        }
    }
}
```

```
// Number of Solutions
if (answerCount == 0) {
    cout << "Tidak ada solusi yang ditemukan! \n";
} else if (answerCount == 1) {
    cout << "Hanya ada 1 solusi yang ditemukan! \n";
} else {
    cout << "Ada " << answerCount << " solusi yang ditemukan! \n";
}

end = clock();
```

```
// Choice to Save the Solution(s) in a File
char choice;
bool validChoice;

validChoice = false;
while (!validChoice) {
    cout << "Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: ";
    cin >> choice;
    if (choice == 'y') {
        string fileName;
        cout << "Masukkan nama file : ";
        cin >> fileName;
        ofstream outputFile;
        string path = "../test/" + fileName + ".txt";
        ofstream outFile(path);

        outputFile.open(path);
        if (answerCount == 0) {
            outputFile << "There is no solution! \n";
        } else if (answerCount == 1) {
            outputFile << "There is only " << answerCount << " solution found! \n";
            for(int i = 0; i < result.size(); i++) {
                outputFile << result.at(i) << "\n";
            }
        } else {
            outputFile << "There are " << answerCount << " solutions found! \n";
            for(int i = 0; i < result.size(); i++) {
                outputFile << result.at(i) << "\n";
            }
        }
        validChoice = true;

        outputFile.close();
```

```
        // Time Taken to Find All Solutions
        double takenTime = (double(end) - double(start)) / double(CLOCKS_PER_SEC);

        cout << "Waktu yang dibutuhkan untuk menemukan seluruh solusi: " << takenTime << " seconds \n";

        // Check If the User Want To Start Over
        char continueChoice;
        bool validContinueChoice;

        validContinueChoice = false;

        while (!validContinueChoice) {
            cout << "Apakah Anda ingin mengulang 24-Game Solver dari awal? \n";
            cout << "Ketik y untuk memulai ulang atau n untuk berhenti: ";
            cin >> continueChoice;
            if (continueChoice == 'y') {
                validContinueChoice = true;
                cout << "========================o======================== \n";
                cout << "\n";
                startSolver();
            } else if (continueChoice == 'n') {
                cout << "Terima kasih telah mencoba 24-Game Solver :) \n";
                cout << "========================o======================== \n";
                cout << "\n";
                validContinueChoice = true;
            } else {
                cout << "Input tidak valid, silakan ulangi! \n";
            }
        }
```

```
    } else if (choice == 'n') {
        validChoice = true;

        // Time Taken to Find All Solutions
        double takenTime = (double(end) - double(start)) / double(CLOCKS_PER_SEC);

        cout << "Waktu yang dibutuhkan untuk menemukan seluruh solusi: " << takenTime << " seconds \n";
        char continueChoice;
        bool validContinueChoice;
```

```
            validContinueChoice = false;

            while (!validContinueChoice) {
                cout << "Apakah Anda ingin mengulang 24-Game Solver dari awal? \n";
                cout << "Ketik y untuk memulai ulang atau n untuk berhenti: ";
                cin >> continueChoice;
                if (continueChoice == 'y') {
                    validContinueChoice = true;
                    cout << "========================o======================== \n";
                    cout << "\n";
                    startSolver();
                } else if (continueChoice == 'n') {
                    cout << "Terima kasih telah mencoba 24-Game Solver :) \n";
                    cout << "========================o======================== \n";
                    cout << "\n";
                    validContinueChoice = true;
                } else {
                    cout << "Input tidak valid, silakan ulangi! \n";
                }
            }
        } else {
            cout << "Pilihan tidak valid, silakan ulangi! \n";
        }
    }
}

int main() {
    startSolver();
}
```

3. Testing Program

Berikut adalah beberapa contoh *test case* dari program penyelesaian permainan kartu 24 yang penulis buat.

Test Case* 1*:

```
========================o========================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 1
Masukkan 4 Kartu (pisahkan dengan spasi atau enter) : 3 5 7 2
Ada 46 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: y
Masukkan nama file : testcase1
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.002 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================o========================
```

test > testcase1.txt

```
There are 46 solutions found!
(3*5)+(7+2)
((3*5)+7)+2
3+((5-2)*7)
(3*5)+(2+7)
((3*5)+2)+7
3+(7*(5-2))
(3*7)+(5-2)
((3*7)+5)-2
3-(7*(2-5))
((3*7)-2)+5
(3*7)-(2-5)
3-((2-5)*7)
5+((3*7)-2)
(5+(3*7))-2
(5*3)+(7+2)
((5*3)+7)+2
(5*3)+(2+7)
((5*3)+2)+7
5+((7*3)-2)
(5+(7*3))-2
(5-2)+(3*7)
5-(2-(3*7))
(5-2)+(7*3)
5-(2-(7*3))
((5-2)*7)+3
7+((3*5)+2)
(7+(3*5))+2
(7*3)+(5-2)
((7*3)+5)-2
((7*3)-2)+5
(7*3)-(2-5)
7+((5*3)+2)
(7+(5*3))+2
(7*(5-2))+3
7+(2+(3*5))
(7+2)+(3*5)
7+(2+(5*3))
(7+2)+(5*3)
2+((3*5)+7)
(2+(3*5))+7
2+((5*3)+7)
(2+(5*3))+7
2+(7+(3*5))
(2+7)+(3*5)
2+(7+(5*3))
(2+7)+(5*3)

```

Test Case 2:

```
========================o=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 1
Masukkan 4 Kartu (pisahkan dengan spasi atau enter) : J 8 2 3
Ada 150 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: n
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.002 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================o=======================
```

Test Case* 3*:

```
========================o=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 1
Masukkan 4 Kartu (pisahkan dengan spasi atau enter) : K Q 2 5
Ada 16 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: y
Masukkan nama file : testcase3
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.001 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================o=======================
```

test > testcase3.txt

```
1   There are 16 solutions found!
2   13+((12/2)+5)
3   (13+(12/2))+5
4   13+(5+(12/2))
5   (13+5)+(12/2)
6   ((13+5)*2)-12
7   (12/2)+(13+5)
8   ((12/2)+13)+5
9   (12/2)+(5+13)
10  ((12/2)+5)+13
11  (2*(13+5))-12
12  (2*(5+13))-12
13  5+(13+(12/2))
14  (5+13)+(12/2)
15  ((5+13)*2)-12
16  5+((12/2)+13)
17  (5+(12/2))+13
18
```

Test Case* 4*:

```
========================o=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 1
Masukkan 4 Kartu (pisahkan dengan spasi atau enter) : K J 2 10
Ada 20 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: n
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.002 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================o=======================
```

Test Case* 5*:

```
========================o=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 2
Hasil generate kartu random : A 9 9 A
Tidak ada solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: y
Masukkan nama file : testcase5
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.002 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================o=======================
```

test > testcase5.txt

```
1  There is no solution!
2
```

Test Case* 6*:

```
========================O=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 2
Hasil generate kartu random : 8 3 A K
Ada 180 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: n
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.003 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================O=======================
```

Test Case 7:

```
========================O=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 2
Hasil generate kartu random : 2 3 3 10
Ada 16 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: y
Masukkan nama file : testcase7
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.003 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================O=======================
```

test > testcase7.txt

```
1   There are 16 solutions found!
2   3*(3+(10/2))
3   (3+(10/2))*3
4   (3*10)-(2*3)
5   3*((10/2)+3)
6   (3*10)-(3*2)
7   3*(3+(10/2))
8   (3+(10/2))*3
9   (3*10)-(2*3)
10  3*((10/2)+3)
11  (3*10)-(3*2)
12  ((10/2)+3)*3
13  ((10/2)+3)*3
14  (10*3)-(2*3)
15  (10*3)-(3*2)
16  (10*3)-(2*3)
17  (10*3)-(3*2)
18
```

Test Case 8:

```
========================O=======================
Selamat datang di 24-Game Solver!
Silakan input 4 kartu valid (nilainya saja), yaitu A, J, Q, K, dan 2 hingga 10!
Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 2
Hasil generate kartu random : A 2 6 9
Ada 72 solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: n
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0.002 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: y
========================O=======================
```

Test Case* 9*:

```
========================O=======================

Keterangan:
A bernilai 1
J bernilai 11
Q bernilai 12
K bernilai 13

Pilih cara pemilihan angka-angka!
1. Manual Input
2. Random
Ketik (1/2) : 2
Hasil generate kartu random : 10 A 6 10
Tidak ada solusi yang ditemukan!
Apakah Anda ingin menyimpan solusi di dalam file? Ketik y untuk menyimpan atau n untuk tidak menyimpan: y
Masukkan nama file : testcase9
Waktu yang dibutuhkan untuk menemukan seluruh solusi: 0 seconds
Apakah Anda ingin mengulang 24-Game Solver dari awal?
Ketik y untuk memulai ulang atau n untuk berhenti: n
Terima kasih telah mencoba 24-Game Solver :)
========================O=======================
```

```
test > testcase9.txt
1   There is no solution!
2
```

4. *Link Repository*

Link Repository untuk tugas kecil I IF2211 Strategi Algoritma ini yaitu https://github.com/billc27/Tucil1_13521064.

5. Pustaka

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf (Diakses 24 Januari 2023, 18.11)

6. Lampiran

Checklist:

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	✓	