

LAPORAN TUGAS KECIL 1
IF 2211 STRATEGI ALGORITMA
SEMESTER II 2022-2023

PENYELESAIAN PERMAINAN KARTU 24
DENGAN ALGORITMA BRUTE FORCE

Disusun oleh :

Fakih Anugerah Pratama

13521091



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

I. Algoritma

1.1. Algoritma Pengurutan Kartu dan Pemilihan Operator

Program akan melewati seluruh kemungkinan pengurutan kartu yang mungkin yaitu 24 susunan, yang berasal dari pemilihan $4 \times 3 \times 2 \times 1$ susunan. Untuk sementara, program mengabaikan kemungkinan adanya permutasi bersyarat yang diakibatkan oleh beberapa kartu memiliki nilai yang sama.

Langkah yang dilakukan, yaitu:

- Pilih kartu beserta urutannya, mengabaikan kemungkinan duplikasi nilai kartu
- Pilih operator yang digunakan dan urutannya
- Cek 5 pola notasi *postfix* yang menjamin seluruh operasi antara 4 kartu dapat diperiksa
- Cek apakah Expression yang terbentuk memiliki hasil 24
- Cek apakah Expression tersebut sudah pernah ditemukan sebelumnya
- Masukkan Expression ke dalam set solusi akhir

1.2. Algoritma Pengecekan Nilai Solusi

Setiap ekspresi yang mewakili kombinasi operator dan angka yang dapat menghasilkan nilai 24 akan dibentuk dalam sebuah objek Expression yang dapat dengan mudah ditentukan nilai yang dihasilkannya. Untuk mempermudah, konsep *postfix expression* atau *Reverse Polish Notation* digunakan sehingga ekspresi Matematika untuk tinggi derajat penghitungan, yang disebabkan baik oleh tanda kurung maupun operator dengan derajat yang lebih tinggi, dapat ditentukan dengan jelas.

Langkah yang dilakukan, yaitu :

- Baca notasi postfix satu per satu dari depan
- Lakukan *push* pada setiap angka yang dijumpai ke dalam stack angka hingga menemukan sebuah operator
- Jika yang dijumpai adalah operator, lakukan *pop* terhadap 2 angka terakhir yang berada dalam stack angka dan lakukan

- operasi sesuai dengan nilai operator yang sedang dijumpai, lalu *push* kembali hasil operasi ke dalam stack angka.
- d. Lakukan Kembali langkah b dan c hingga notasi telah terbaca semuanya dan tersisa satu angka dalam stack.

1.3. Algoritma Pengecekan Duplikat Solusi

Solusi yang memiliki notasi PRN yang sama dianggap sebagai sebuah ekspresi yang sama, sehingga harus terlebih dulu dicek keberadaannya sebelum di-push ke dalam vector/stack solusi akhir.

II. Source Code dalam Bahasa C++

2.1 Berkas main.cpp

```
#include "card.h"
#include "deck.h"
#include "solver.h"
#include "expression.h"

#include <limits>
#include <iostream>
#include <vector>
#include <time.h>

using namespace std;

int main() {
    Deck currentCards;
    cardSolver csolver;

    srand(time(0));

    bool start = true;
    string choice = "-1";
    while (start)
    {
        cout << "##### 24-Card Game #####" << endl;
        cout << endl << "Kartu saat ini :" << endl;
        cout << "\t"; currentCards.DisplayDeck();
        cout << endl << "Menu" << endl;
        cout << " 1. Acak Kartu" << endl;
        cout << " 2. Atur Kartu" << endl;
        cout << " 3. Simpan Solusi" << endl;
        cout << endl;
        cout << " 0. Cari Solusi" << endl;
        cout << endl << endl;
        cout << "99. Keluar" << endl;

        cout << endl << "Pilihan Anda : "; cin >> choice; cout << endl;

        if (choice == "1"){
            currentCards.RandomizeDeck();
        } else if (choice == "2"){
            vector<int> _cardChoices = {0, 0, 0, 0};
```

```
        cout << "Masukkan nilai kartu baru :" << endl; cin >> _cardChoices[0]
>> _cardChoices[1] >> _cardChoices[2] >> _cardChoices[3]; cout << endl;

        if(!cin)
            cin.clear();
        else {
            bool valid = true;
            for(auto _i:_cardChoices){
                if (_i < 1 || _i > 13){
                    valid = false;
                }
            }

            if(valid)
                currentCards.SetDeck(_cardChoices);
        }

        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    } else if (choice == "3"){
        csolver.SaveSolution();
    } else if (choice == "0"){
        csolver.SetDeck(currentCards);
        csolver.GenerateSolution();
        csolver.ShowInfixSolution();

        csolver.ShowExecutionTime();
    } else if (choice == "99"){
        start = false;
    }

    choice == "-1";
}

return 0;
}
```

2.2Berkas card.h

```
#ifndef CARD_H
#define CARD_H

#include <iostream>
#include <vector>
#include <string>
#include <random>

#include <time.h>

using namespace std;

int randint(int start, int end);

class Card {
    int value = 0; // NOT A CARD

public:
    Card();
    Card(int init);

    int asInt();
    string asCard();
    string asString();

    void SetCard(int val);
    void RandomizeCard();
};

#endif
```

2.3Berkas card.cpp

```
#include "card.h"

int randint(int start, int end){
    return rand() % (end - start + 1) + start;
}

Card::Card(){
    // default constructor overload
}

Card::Card(int init){
    value = init;
}

string Card::asCard(){
    if (value == 0){
        return "X"; // NOT A CARD
    } else 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 to_string(value);
    }
}

int Card::asInt(){
    return (int) value;
}

string Card::asString(){
    return to_string(value);
}

void Card::SetCard(int val){
    value = val;
}
```

```
void Card::RandomizeCard(){  
    value = randint(1, 13);  
}
```

2.4 Berkas deck.h

```
#ifndef DECK_H  
#define DECK_H  
    #include "card.h"  
  
    #include <iostream>  
  
    class Deck {  
    public:  
        vector<Card> cardDeck = {Card(), Card(), Card(), Card()};  
  
        void DisplayDeck();  
  
        void RandomizeDeck();  
        void SetDeck(int a, int b, int c, int d);  
        void SetDeck(vector<int> vec);  
    };  
  
#endif
```


2.5 Berkas deck.cpp

```
#include "deck.h"

void Deck::DisplayDeck(){
    for(auto i:cardDeck){
        cout << i.asCard() << " ";
    } cout << endl;
}

void Deck::RandomizeDeck(){
    for(int i = 0; i < 4; i++){
        cardDeck[i].RandomizeCard();
    }
}

void Deck::SetDeck(int a, int b, int c, int d){
    cardDeck = {Card(a), Card(b), Card(c), Card(d)};
}

void Deck::SetDeck(vector<int> vec){
    cardDeck = {Card(vec[0]), Card(vec[1]), Card(vec[2]), Card(vec[3])};
}
```

2.6 Berkas expression.h

```
#ifndef EXPRESSION_H
#define EXPRESSION_H

#include <vector>
#include <string>
#include <iostream>

#include "deck.h"

using namespace std;

class Expression{
    vector<string> expr;

public:
    Expression(vector<string> init);
}
```

```
        int getResult();

        float stringToFloat(string str);

        void SetExpr(vector<string> vec);
        string GetExpr();
        string GetExprAsInfix();

        void ShowExpr();
        void ShowExprAsInfix();

        bool Equal(Expression eqTarget);

};

#endif // !EXPRESSION_H
```

2.7 Berkas expression.cpp

```
#include "expression.h"

Expression::Expression(vector<string> init){
    expr = init;
}

int Expression::getResult(){
    vector<float> numStack;

    for(auto a: expr){
        if(a != "+" && a != "-" && a != "*" && a != "/"){
            numStack.push_back(stringToFloat(a));
        } else {
            float _temp1 = numStack.back(); numStack.pop_back();
            float _temp2 = numStack.back(); numStack.pop_back();

            if(a == "+"){
                numStack.push_back(_temp2 + _temp1);
            } else
            if (a == "-"){
                numStack.push_back(_temp2 - _temp1);
            } else
            if (a == "*"){
```

```
        numStack.push_back(_temp2 * _temp1);
    } else
    {
        if (a == "/"){
            numStack.push_back(_temp2 / _temp1);
        }
    }
}

if ((numStack[0] - (int)numStack[0]) == 0)
    return (int)numStack[0];
else
    return 0;
}

float Expression::stringToFloat(string str){
    for(int i = 1; i <= 13; i++){
        if(to_string(i) == str){
            return (float) i;
        }
    }
}

void Expression::SetExpr(vector<string> vec){
    expr = vec;
}

string Expression::GetExpr(){
    string result;

    for(auto i:expr){
        result += i + " ";
    } result += "\n";

    return result;
}

string Expression::GetExprAsInfix(){
    vector<float> numStack;
    vector<string> infixStack;

    for(auto a:expr){
        if(a != "+" && a != "-" && a != "*" && a != "/"){
            numStack.push_back(stringToFloat(a));
            infixStack.push_back(a);
        }
    }
}
```

```
        } else {
            float _temp1 = numStack.back(); numStack.pop_back();
            float _temp2 = numStack.back(); numStack.pop_back();

            string _tempStr1 = infixStack.back(); infixStack.pop_back();
            string _tempStr2 = infixStack.back(); infixStack.pop_back();

            if(a == "+"){
                numStack.push_back(_temp2 + _temp1);
                infixStack.push_back("(" + _tempStr2 + " + " + _tempStr1 + "
)");
            } else
            if (a == "-"){
                numStack.push_back(_temp2 - _temp1);
                infixStack.push_back("(" + _tempStr2 + " - " + _tempStr1 + "
)");
            } else
            if (a == "*"){
                numStack.push_back(_temp2 * _temp1);
                infixStack.push_back("(" + _tempStr2 + " * " + _tempStr1 + "
)");
            } else
            if (a == "/"){
                numStack.push_back(_temp2 / _temp1);
                infixStack.push_back("(" + _tempStr2 + " / " + _tempStr1 + "
)");
            }
        }
    }

    return infixStack[0] + "\n";
}

void Expression::ShowExpr(){
    for(auto i:expr){
        cout << i << " ";
    } cout << endl;
}

void Expression::ShowExprAsInfix(){
    vector<float> numStack;
    vector<string> infixStack;

    for(auto a:expr){
```

```
        if(a != "+" && a != "-" && a != "*" && a != "/"){
            numStack.push_back(stringToFloat(a));
            infixStack.push_back(a);
        } else {
            float _temp1 = numStack.back(); numStack.pop_back();
            float _temp2 = numStack.back(); numStack.pop_back();

            string _tempStr1 = infixStack.back(); infixStack.pop_back();
            string _tempStr2 = infixStack.back(); infixStack.pop_back();

            if(a == "+"){
                numStack.push_back(_temp2 + _temp1);
                infixStack.push_back("(" + _tempStr2 + " + " + _tempStr1 + "
            )");
            } else
            if (a == "-"){
                numStack.push_back(_temp2 - _temp1);
                infixStack.push_back("(" + _tempStr2 + " - " + _tempStr1 + "
            )");
            } else
            if (a == "*"){
                numStack.push_back(_temp2 * _temp1);
                infixStack.push_back("(" + _tempStr2 + " * " + _tempStr1 + "
            )");
            } else
            if (a == "/"){
                numStack.push_back(_temp2 / _temp1);
                infixStack.push_back("(" + _tempStr2 + " / " + _tempStr1 + "
            )");
            }
        }
    }

    cout << infixStack[0] << endl;
}

bool Expression::Equal(Expression eqTarget){
    if(expr.size() != eqTarget.expr.size())
        return false;

    for(int i = 0; i < expr.size(); i++){
        if(expr[i] != eqTarget.expr[i]){
            return false;
        }
    }
}
```

```
    }  
  
    return true;  
}
```

2.8 Berkas solver.h

```
#ifndef SOLVER_H  
#define SOLVER_H  
  
#include <vector>  
#include <fstream>  
  
#include <time.h>  
  
#include "expression.h"  
#include "deck.h"  
  
using namespace std;  
  
static vector<string> ops = {"+", "-", "*", "/"};  
  
class cardSolver {  
    vector<Expression> solutionExpr;  
  
    Deck deckToSolve;  
  
public:  
    int lastExecutionTime;  
    void SetDeck(Deck deck);  
    void ShowDeck();  
  
    void GenerateSolution();  
    void ClearSolution();  
    void SaveSolution();  
  
    void ShowExecutionTime();  
  
    void PushSolution(Expression candidateSolution);  
  
    void ShowSolution();  
    void ShowInfixSolution();  
};
```

```
        bool IsSolutionContains(vector<Expression> _solutionExpr, Expression
        _expr);

    };

#endif
```

2.9 Berkas solver.cpp

```
#include "solver.h"

void cardSolver::SetDeck(Deck deck){
    deckToSolve = deck;
}

void cardSolver::ShowDeck(){
    deckToSolve.DisplayDeck();
}

void cardSolver::GenerateSolution(){
    ClearSolution();

    clock_t startTime, endTime;
    startTime = clock();

    for(int di = 0; di < 4; di++){
        for(int dj = 0; dj < 4; dj++){
            if(dj == di)
                continue;

            for(int dk = 0; dk < 4; dk++){
                if(dk == dj || dk == di)
                    continue;

                for(int dl = 0; dl < 4; dl++){
                    if(dl == di || dl == dj || dl == dk)
                        continue;

                    for(auto i:ops) {
                        for(auto j:ops){
                            for(auto k:ops){
```

```

        PushSolution(Expression({deckToSolve.cardDeck[di
].asString(), deckToSolve.cardDeck[dj].asString(), i,
deckToSolve.cardDeck[dk].asString(), j, deckToSolve.cardDeck[dl].asString(),
k}));

        PushSolution(Expression({deckToSolve.cardDeck[di
].asString(), deckToSolve.cardDeck[dj].asString(),
deckToSolve.cardDeck[dk].asString(), i, j, deckToSolve.cardDeck[dl].asString(),
k}));

        PushSolution(Expression({deckToSolve.cardDeck[di
].asString(), deckToSolve.cardDeck[dj].asString(),
deckToSolve.cardDeck[dk].asString(), i, deckToSolve.cardDeck[dl].asString(), j,
k}));

        PushSolution(Expression({deckToSolve.cardDeck[di
].asString(), deckToSolve.cardDeck[dj].asString(),
deckToSolve.cardDeck[dk].asString(), deckToSolve.cardDeck[dl].asString(), i, j,
k}));

        PushSolution(Expression({deckToSolve.cardDeck[di
].asString(), deckToSolve.cardDeck[dj].asString(), i,
deckToSolve.cardDeck[dk].asString(), deckToSolve.cardDeck[dl].asString(), j,
k}));

    }
}
}
}
}
}
}

endTime = clock();

lastExecutionTime = double(endTime - startTime) / double(CLOCKS_PER_SEC) *
1000;
}

void cardSolver::ClearSolution(){
    solutionExpr.clear();
}

void cardSolver::SaveSolution(){
    string path = "./test/Solution_" + to_string(time(0)) + ".txt";

    ofstream SolutionFile(path);

```



```
        for(auto i:solutionExpr){
            SolutionFile << i.GetExprAsInfix();
        }

        SolutionFile.close();

        cout << "Solutions saved as " + path << endl;
    }

void cardSolver::ShowExecutionTime(){
    cout << "Execution time : " << lastExecutionTime << " ms" << endl << endl;
}

void cardSolver::PushSolution(Expression candidateSolution){
    if(candidateSolution.getResult() == 24){
        if(!IsSolutionContains(solutionExpr, candidateSolution))
            solutionExpr.push_back(candidateSolution);
    }
}

void cardSolver::ShowSolution(){
    cout << endl;
    if(solutionExpr.size() == 0)
        cout << "There is no solution for this set of cards." << endl;
    else
        cout << "There are " << solutionExpr.size() << " solutions." << endl;

    for(auto i:solutionExpr){
        i.ShowExpr();
    }
    cout << endl << endl;
}

void cardSolver::ShowInfixSolution(){
    cout << endl;
    if(solutionExpr.size() == 0)
        cout << "There is no solution for this set of cards." << endl;
    else
        cout << "There are " << solutionExpr.size() << " solutions." << endl;

    for(auto i:solutionExpr){
        i.ShowExprAsInfix();
    }
    cout << endl << endl;
}
```

```
}  
  
bool cardSolver::IsSolutionContains(vector<Expression> _solutionExpr, Expression  
_expr){  
    for(auto i:_solutionExpr){  
        if(i.Equal(_expr))  
            return true;  
    }  
  
    return false;  
}
```

III. Hasil Eksekusi Program

3.1. Tampilan Utama

```
##### 24-Card Game #####  
  
Kartu saat ini :  
    X X X X  
  
Menu  
  1. Acak Kartu  
  2. Atur Kartu  
  3. Simpan Solusi  
  
  0. Cari Solusi  
  
99. Keluar  
  
Pilihan Anda : █
```

3.2. Kartu Acak

2 10 9 Q	<pre>Pilihan Anda : 1 ##### 24-Card Game ##### Kartu saat ini : 2 10 9 Q Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 0</pre>
----------	---

```

There are 24 solutions.
( ( 2 - 10 ) * ( 9 - 12 ) )
( ( 2 * ( 10 - 9 ) ) * 12 )
( 2 * ( ( 10 - 9 ) * 12 ) )
( ( 2 / ( 10 - 9 ) ) * 12 )
( 2 / ( ( 10 - 9 ) / 12 ) )
( 2 * ( 12 * ( 10 - 9 ) ) )
( 2 * ( 12 / ( 10 - 9 ) ) )
( ( 2 * 12 ) * ( 10 - 9 ) )
( ( 2 * 12 ) / ( 10 - 9 ) )
( ( 10 - 2 ) * ( 12 - 9 ) )
( ( ( 10 - 9 ) * 2 ) * 12 )
( ( 10 - 9 ) * ( 2 * 12 ) )
( ( ( 10 - 9 ) * 12 ) * 2 )
( ( 10 - 9 ) * ( 12 * 2 ) )
( ( 9 - 12 ) * ( 2 - 10 ) )
( 12 * ( 2 * ( 10 - 9 ) ) )
( 12 * ( 2 / ( 10 - 9 ) ) )
( ( 12 * 2 ) * ( 10 - 9 ) )
( ( 12 * 2 ) / ( 10 - 9 ) )
( ( 12 * ( 10 - 9 ) ) * 2 )
( 12 * ( ( 10 - 9 ) * 2 ) )
( ( 12 / ( 10 - 9 ) ) * 2 )
( 12 / ( ( 10 - 9 ) / 2 ) )
( ( 12 - 9 ) * ( 10 - 2 ) )
    
```

9 9 A Q		<pre>There are 8 solutions. (((9 / 9) + 1) * 12) ((9 - 1) * (12 - 9)) ((9 - 12) * (1 - 9)) ((1 - 9) * (9 - 12)) ((1 + (9 / 9)) * 12) ((12 - 9) * (9 - 1)) (12 * ((9 / 9) + 1)) (12 * (1 + (9 / 9))) Execution time : 593 ms</pre>	
----------------	--	--	--

Pilihan Anda : 1

24-Card Game

Kartu saat ini :
7 K 4 5

Menu

1. Acak Kartu
2. Atur Kartu
3. Simpan Solusi

0. Cari Solusi

99. Keluar

Pilihan Anda : 0

There are 8 solutions.

$((7 * 13) + 5) / 4$
 $(7 - 4) * (13 - 5)$
 $((13 * 7) + 5) / 4$
 $(13 - 5) * (7 - 4)$
 $(4 - 7) * (5 - 13)$
 $(5 + (7 * 13)) / 4$
 $(5 + (13 * 7)) / 4$
 $(5 - 13) * (4 - 7)$

Execution time : 503 ms

7 K 4 5

Pilihan Anda : 1

24-Card Game

Kartu saat ini :
7 K 4 5

Menu

1. Acak Kartu
2. Atur Kartu
3. Simpan Solusi

0. Cari Solusi

99. Keluar

Pilihan Anda : 0

There are 8 solutions.

```
(( ( 7 * 13 ) + 5 ) / 4 )
(( 7 - 4 ) * ( 13 - 5 ) )
(( ( 13 * 7 ) + 5 ) / 4 )
(( 13 - 5 ) * ( 7 - 4 ) )
(( 4 - 7 ) * ( 5 - 13 ) )
(( 5 + ( 7 * 13 ) ) / 4 )
(( 5 + ( 13 * 7 ) ) / 4 )
(( 5 - 13 ) * ( 4 - 7 ) )
```

Execution time : 503 ms

3.3. Kartu Masukan Pengguna

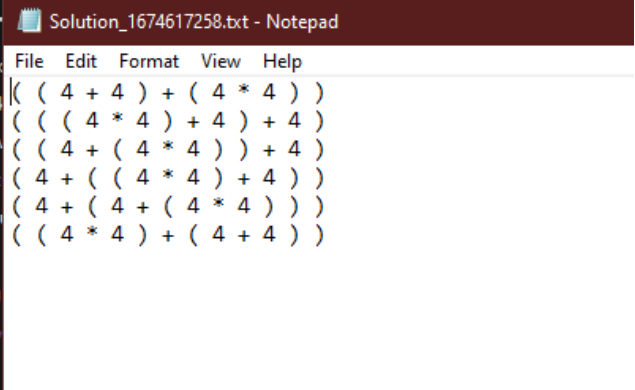
9 8 A 2	<pre>Pilihan Anda : 2 Masukkan nilai kartu baru : 9 8 1 2 ##### 24-Card Game ##### Kartu saat ini : 9 8 A 2 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 0</pre>
---------	--

		<pre> There are 38 solutions. (9 * (8 / (1 + 2))) ((9 * 8) / (1 + 2)) (9 * (8 / (2 + 1))) ((9 + (8 * 2)) - 1) ((9 * 8) / (2 + 1)) (9 + ((8 * 2) - 1)) ((9 - 1) + (8 * 2)) (9 - (1 - (8 * 2))) ((9 / (1 + 2)) * 8) (9 / ((1 + 2) / 8)) (((9 - 1) * 2) + 8) ((9 - 1) + (2 * 8)) (9 - (1 - (2 * 8))) ((9 + (2 * 8)) - 1) (9 + ((2 * 8) - 1)) ((9 / (2 + 1)) * 8) (9 / ((2 + 1) / 8)) (8 * (9 / (1 + 2))) (8 + ((9 - 1) * 2)) ((8 * 9) / (1 + 2)) (8 * (9 / (2 + 1))) ((8 * 9) / (2 + 1)) (8 - ((1 - 9) * 2)) ((8 / (1 + 2)) * 9) (8 / ((1 + 2) / 9)) (8 + (2 * (9 - 1))) (((8 * 2) + 9) - 1) ((8 * 2) + (9 - 1)) ((8 / (2 + 1)) * 9) (8 / ((2 + 1) / 9)) (8 - (2 * (1 - 9))) (((8 * 2) - 1) + 9) ((8 * 2) - (1 - 9)) ((2 * (9 - 1)) + 8) (((2 * 8) + 9) - 1) ((2 * 8) + (9 - 1)) (((2 * 8) - 1) + 9) ((2 * 8) - (1 - 9)) Execution time : 366 ms </pre>	
--	--	---	--

4 4 4 4	<pre>Pilihan Anda : 2 Masukkan nilai kartu baru : 4 4 4 4 ##### 24-Card Game ##### Kartu saat ini : 4 4 4 4 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 0 There are 6 solutions. ((4 + 4) + (4 * 4)) (((4 * 4) + 4) + 4) ((4 + (4 * 4)) + 4) (4 + ((4 * 4) + 4)) (4 + (4 + (4 * 4))) ((4 * 4) + (4 + 4)) Execution time : 310 ms</pre>
----------------	---

<p>9 A Q A</p>	<pre>Pilihan Anda : 2 Masukkan nilai kartu baru : 9 1 12 1 ##### 24-Card Game ##### Kartu saat ini : 9 A Q A Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 0 There is no solution for this set of cards. Execution time : 426 ms</pre>
----------------	---

3.4. Menyimpan Hasil

<p>Proses</p>	<pre> There are 6 solutions. ((4 + 4) + (4 * 4)) (((4 * 4) + 4) + 4) ((4 + (4 * 4)) + 4) (4 + ((4 * 4) + 4)) (4 + (4 + (4 * 4))) ((4 * 4) + (4 + 4)) Execution time : 555 ms ##### 24-Card Game ##### Kartu saat ini : 4 4 4 4 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 3 Solutions saved as ../test/Solution_1674617258.txt </pre>
<p>Hasil</p>	 <pre> Solution_1674617258.txt - Notepad File Edit Format View Help ((4 + 4) + (4 * 4)) (((4 * 4) + 4) + 4) ((4 + (4 * 4)) + 4) (4 + ((4 * 4) + 4)) (4 + (4 + (4 * 4))) ((4 * 4) + (4 + 4)) </pre>

3.5. Penanganan Masukan yang Tidak Sesuai

<p>Kasus pilihan salah (tidak tersedia atau tidak sesuai)</p>	<pre>##### 24-Card Game ##### Kartu saat ini : 9 A Q A Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 1391abc ##### 24-Card Game ##### Kartu saat ini : 9 A Q A Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : █</pre>
---	---

<p>Kasus percobaan <i>abuse</i> pada <i>buffer</i> cin yang menerima masukan yang salah</p>	<pre>Kartu saat ini : A 2 3 4 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 2 Masukkan nilai kartu baru : 1 2 a 9 ##### 24-Card Game ##### Kartu saat ini : A 2 3 4 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : █</pre>	
---	---	--

<p>Kasus masukan melebihi banyak kartu (nilai yang tak digunakan akan dipotong)</p>	<pre> Kartu saat ini : 8 3 10 9 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : 2 Masukkan nilai kartu baru : 1 2 3 4 99 ##### 24-Card Game ##### Kartu saat ini : A 2 3 4 Menu 1. Acak Kartu 2. Atur Kartu 3. Simpan Solusi 0. Cari Solusi 99. Keluar Pilihan Anda : █ </pre>	
---	---	--

IV. Lampiran

4.1. Pranala *Repository* Github

https://github.com/fakihap/Tucil1_13521091

4.2. Tabel Ketercapaian

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	✓	