

LAPORAN TUGAS KECIL
IF2211 STRATEGI ALGORITMA
Penyelesaian Permainan Kartu 24
dengan Algoritma *Brute Force*




Disusun oleh

Akhmad Setiawan - 13521164 - K2

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2211-TK1-13521164-K2		26
		Revisi	1	25-01-2023

Daftar Isi

1 Deskripsi Algoritma <i>Brute Force</i> yang Diimplementasikan	3
2 <i>Source Program</i>	4
3 Tampilan Program dan Contoh Input/Output	18
4 Link Repository	26
5 Ceklis	26

1 Deskripsi Algoritma *Brute Force* yang Diimplementasikan

Permainan kartu 24 adalah permainan di mana pemain menghitung 4 buah angka yang direpresentasikan dengan kartu remi. Kartu remi yang digunakan ialah dari As, yang direpresentasikan sebagai angka 1 hingga kartu King yang direpresentasikan dengan angka 13. Pemain diharuskan mencari operasi aritmatika yang menghubungkan keempat angka tersebut sehingga terbentuk sebuah angka 24. Operasi aritmatika yang dapat digunakan dalam permainan tersebut adalah tambah (+), kurang (-), kali (x), dan bagi (/). Angka pada setiap kartu harus digunakan sekali dan hanya boleh digunakan sekali dalam proses operasi perhitungan. Penyelesaian permainan ini dapat menggunakan beberapa pendekatan, salah satunya dengan algoritma *brute force*.

Algoritma *brute force* yang ialah metode penyelesaian persoalan dengan secara lempang (straightforward). Artinya, jika menyelesaikan permainan kartu 24 dengan metode ini, kita akan mencari semua kemungkinan jawaban dari semua kombinasi operasi dan angka yang ada, dan mengumpulkan semua kemungkinan yang hasil nilainya adalah 24. Berikut adalah kemungkinan susunan angka-angka dalam setiap operasi pada 4 kartu tadi.

- Dalam permainan 24, jumlah angka yang dapat digunakan dalam operasi haruslah 4. Maka dari itu, dengan menggunakan permutasi, terdapat maksimal $4! = 24$ kemungkinan susunan angka yang mungkin. Namun, ada kemungkinan terdapat duplikat antara keempat angka tersebut, sehingga perlu diperhatikan terkait permutasi dengan unsur yang sama.
- Terdapat 4 buah operator yang dapat digunakan dalam permainan 24, yaitu tambah, kurang, kali, dan bagi. Karena terdapat 4 buah angka dalam permainan ini, maka terdapat 3 tempat di mana operator harus dimasukkan untuk menghasilkan operasi aritmatika, yaitu 1 di antara 2 angka pertama, 1 di antara 2 angka tengah, dan 1 di antara 2 angka terakhir. Karena terdapat 4 operator dan 3 buah tempat yang tersedia, dengan aturan perkalian, maka kita mendapatkan maksimal $4 \times 4 \times 4 = 64$ buah kemungkinan susunan operator yang mungkin.
- Penempatan tanda kurung dalam operasi aritmatika berfungsi untuk menentukan urutan operasi mana yang dikerjakan terlebih dahulu. Dari penempatan tanda kurung tersebut, terdapat 5 buah jenis solusi yang ada.
 1. $(a + b) + (c + d)$
 2. $((a + b) + c) + d$
 3. $(a + (b + c)) + d$
 4. $a + ((b + c) + d)$
 5. $a + (b + (c + d))$

Dalam implementasinya ke dalam kode program, pertama-tama yang saya lakukan ialah dengan menelusuri semua kemungkinan angka yang didapat. Terdapat beberapa variasi, yakni:

1. Jika semua kartu memiliki angka yang berbeda
2. Jika ada sepasang kartu dengan angka yang sama
3. Jika ada 2 pasang kartu dengan angka yang sama
4. Jika ada 3 kartu dengan angka yang sama, dan
5. Jika semua kartu angkanya sama.

Setiap variasi ini dicari dengan metode *brute force* secara manual satu-per-satu. Masing-masing dari variasi tersebut kemudian dicari lagi secara menyeluruh terkait variasi tanda kurung yang telah disebutkan di atas. Dalam setiap variasi tanda kurung pula, dicari semua kemungkinan

dengan mengakses semua variasi operator yang tersedia. Dengan metode ini lah, diperoleh hasil semua perhitungan dari semua variasi angka kartu, variasi tanda kurung, dan variasi operator yang tersedia. Namun, yang hanya kita pedulikan ialah variasi yang menghasilkan angka 24 saja. Untuk fungsi, prosedur, dan variabel yang digunakan dapat dilihat pada sumber kode program yang akan ditampilkan di bawah.

2 Source Program

Program ini ditulis dengan bahasa C++ dengan bantuan beberapa *library*:

1. *iostream*, untuk *handling input dan output*
2. *cstdlib*, untuk bantuan *men-generate random number*
3. *string*, untuk *input dalam bentuk string*
4. *vector*, untuk menyimpan solusi yang banyaknya tidak tentu (dinamis)
5. *time.h*, untuk menghitung waktu eksekusi dan *generate random number*
6. *fstream*, untuk *handling save output to file*

```
#include <iostream>
#include <cstdlib>
#include <string>
#include <vector>
#include <time.h>
#include <fstream>

using namespace std;
```

Gambar 2.1 Daftar *Library* yang digunakan

Adapun beberapa deklarasi variabel, *container*, void, dan function dapat dilihat pada gambar berikut.

```
// Variables
float a, b, c, d;
string e, f, g, h;
int count = 0;
float exec;

// Container
string cardSign[13] = {"A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K"};
float cardValue[13] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
vector<string> solutions;
```

Gambar 2.2 Deklarasi Variabel Global dan Container

```

// Declaration of Void and Function
void manualInput();
// input manual oleh user

void randomInput();
// input dengan random generator

bool checkInput(string input);
// pengecekan validitas input

float strToFloat(string x);
// typecasting string ke float

void twoSameNumbers(float p, float q, float r, float s);
// variasi jika ada sepasang kartu dengan angka yang sama

void doubleTwoSameNumbers(float p, float q, float r, float s);
// variasi jika terdapat 2 pasang kartu dengan angka yang sama

void threeSameNumbers(float p, float q, float r, float s);
// variasi jika terdapat 3 kartu dengan angka yang sama

void noSameNumbers(float p, float q, float r, float s);
// variasi jika tidak ada angka yang sama

void countIf24(float a, float b, float c, float d);
// menghitung jika jumlah hasil operasi adalah 24

void is24(float a, float b, float c, float d);
// mengecek apakah hasil suatu operasi adalah 24

float combinationOfCalculation(float a, float b, int sign);
// hasil kombinasi dari semua operasi terhadap keempat angka

string flagToOperator(int flag);
// mengubah integer flag sebagai penanda menjadi operator

void counter(float a, float b, float c, float d, int i, int j, int k);
// menghitung solusi

void printResult(float a, float b, float c, float d, int i, int j, int k);
// print-out solusi ke layar

void saveToFile(string filename);
// menyimpan solusi ke dalam masukan file

```

Gambar 2.3 Deklarasi Void dan Function

Setelah deklarasi void dan function, selanjutnya terdapat implementasi program utama yang dijalankan untuk mendapatkan solusi. Implementasinya dapat dilihat seperti gambar berikut.

```
// Main Program
int main(){
    string userChoice;
    string saveOrNo;
    string filename;

    cout << "Hello, welcome to make it 24 Card Game!" << endl;

    // memilih input
    bool loop = true;
    while (loop == true){
        cout << "How would you like to input your card value?" << endl;
        cout << "1. Manual" << endl;
        cout << "2. Random" << endl;
        cout << "Your choice is: ";
        cin >> userChoice;
        cout << endl;
        if (userChoice == "1"){
            manualInput();
            loop = false;
        } else if (userChoice == "2"){
            randomInput();
            loop = false;
        } else{
            cout << "Invalid input! Please enter again" << endl;
        }
    }
    // konversi kartu ke valuenya
    a = strToFloat(e);
    b = strToFloat(f);
    c = strToFloat(g);
    d = strToFloat(h);

    clock_t start = clock(); // untuk menghitung waktu eksekusi
```

```

// variasi angka input
if ((a == b) && (a != c) && (a != d) && (c != d)){ // aacd
    twoSameNumbers(a, b, c, d);
}
if ((a == c) && (a != b) && (a != d) && (b != d)){ // abad
    twoSameNumbers(a, c, b, d);
}
if ((a == d) && (a != b) && (a != c) && (b != c)){ // abca
    twoSameNumbers(a, d, b, c);
}
if ((b == c) && (b != a) && (b != d) && (a != d)){ // abbd
    twoSameNumbers(b, c, a, d);
}
if ((b == d) && (b != a) && (b != c) && (a != c)){ // abcb
    twoSameNumbers(b, d, a, c);
}
if ((c == d) && (c != a) && (c != b) && (a != b)){ // abcc
    twoSameNumbers(c, d, a, b);
}
if ((a == b) && (c == d) && (a != c)){ // aacc
    doubleTwoSameNumbers(a, b, c, d);
}
if ((a == c) && (b == d) && (a != b)){ // abab
    doubleTwoSameNumbers(a, c, b, d);
}
if ((a == d) && (b == c) && (a != b)){ // abba
    doubleTwoSameNumbers(a, d, b, c);
}

```

```

    if ((a == b) && (a == c) && (a != d)){ // aaad
        threeSameNumbers(a, b, c, d);
    }
    if ((a == b) && (a == d) && (a != c)){ // aaca
        threeSameNumbers(a, b, d, c);
    }
    if ((a == c) && (a == d) && (a != b)){ // abaa
        threeSameNumbers(a, d, c, b);
    }
    if ((b == c) && (b == d) && (a != b)){ // abbb
        threeSameNumbers(d, b, c, a);
    }
    if ((a == b) && (c == d) && (a == c)){ // aaaa
        countIf24(a, b, c, d);
        cout << count << " solution(s) found" << endl;
        cout << endl;
        is24(a, b, c, d);
    }
    if ((a != b) && (a != c) && (a != d) && (b != c) && (b != d) && (c != d)){ // abcd
        noSameNumbers(a, b, c, d);
    }

    exec = (float)(clock() - start)/CLOCKS_PER_SEC; // akhir waktu eksekusi
    cout << endl << "Execution time is " << exec << endl;

    // save to file
    loop = true;
    while (loop == true){
        cout << "Would you like to save solutions into file?" << endl;
        cout << "1. Yes" << endl;
        cout << "2. No" << endl;
        cout << "Your choice is: ";
        cin >> saveOrNo;
        cout << endl;
        if (saveOrNo == "1" || saveOrNo == "yes" || saveOrNo == "Yes" || saveOrNo == "YES" || saveOrNo == "y" || saveOrNo == "Y"){
            cout << "input your txt filename: ";
            cin >> filename;
            filename = filename + ".txt";
            saveToFile(filename);
            loop = false;
        } else if (saveOrNo == "2" || saveOrNo == "no" || saveOrNo == "No" || saveOrNo == "NO" || saveOrNo == "n" || saveOrNo == "N"){
            loop = false;
        } else{
            cout << "Invalid input! Please enter again" << endl;
        }
    }
    return 0;
}

```

Gambar 2.4 Implementasi *Main Program*

Setelah implementasi *main program*, kode dilanjutkan dengan implementasi void dan function yang sudah dideklarasikan di awal tadi, seperti gambar berikut.

```
// Implementation of Void and Function
void manualInput(){
    cout << "Input your 4 cards value: " << endl;
    cin >> e >> f >> g >> h;
    cout << endl;
}

void randomInput(){
    int idx1, idx2, idx3, idx4;

    srand(time(NULL));
    idx1 = (rand() % 13);
    idx2 = (rand() % 13);
    idx3 = (rand() % 13);
    idx4 = (rand() % 13);
    e = cardSign[idx1];
    f = cardSign[idx2];
    g = cardSign[idx3];
    h = cardSign[idx4];
    cout << "Your random cards: " << e << f << g << h << endl;
}

bool checkInput(string input){
    bool found = false;
    int idx = 0;

    while (!found && idx < 14){
        if (input == cardSign[idx]){
            found = true;
            return found;
        }
        idx++;
    }
    return found;
}
```

```

float strToFloat(string x){
    bool found = false;
    float convertResult;
    int idx = 0;

    while (!found && idx < 14){
        if (x == cardSign[idx]){
            convertResult = cardValue[idx];
            found = true;
            return convertResult;
        }
        idx++;
    }
    return stof(x);
}

```

```

void noSameNumbers(float p, float q, float r, float s){
    countIf24(p, q, r, s);
    countIf24(p, q, s, r);
    countIf24(p, r, q, s);
    countIf24(p, r, s, q);
    countIf24(p, s, q, r);
    countIf24(p, s, r, q);
    countIf24(q, p, r, s);
    countIf24(q, p, s, r);
    countIf24(q, r, p, s);
    countIf24(q, r, s, p);
    countIf24(q, s, p, r);
    countIf24(q, s, r, p);
    countIf24(r, p, q, s);
    countIf24(r, p, s, q);
    countIf24(r, q, p, s);
    countIf24(r, q, s, p);
    countIf24(r, s, p, q);
    countIf24(r, s, q, p);
    countIf24(s, p, q, r);
    countIf24(s, p, r, q);
    countIf24(s, q, p, r);
    countIf24(s, q, r, p);
    countIf24(s, r, p, q);
    countIf24(s, r, q, p);
    cout << count << " solution(s) found" << endl;
    cout << endl;
}

```

```
is24(p, q, r, s);  
is24(p, q, s, r);  
is24(p, r, q, s);  
is24(p, r, s, q);  
is24(p, s, q, r);  
is24(p, s, r, q);  
is24(q, p, r, s);  
is24(q, p, s, r);  
is24(q, r, p, s);  
is24(q, r, s, p);  
is24(q, s, p, r);  
is24(q, s, r, p);  
is24(r, p, q, s);  
is24(r, p, s, q);  
is24(r, q, p, s);  
is24(r, q, s, p);  
is24(r, s, p, q);  
is24(r, s, q, p);  
is24(s, p, q, r);  
is24(s, p, r, q);  
is24(s, q, p, r);  
is24(s, q, r, p);  
is24(s, r, p, q);  
is24(s, r, q, p);
```

```
}
```

```

void twoSameNumbers(float p, float q, float r, float s){
    countIf24(p, p, r, s);
    countIf24(r, p, p, s);
    countIf24(p, r, p, s);
    countIf24(r, p, s, p);
    countIf24(p, r, s, p);
    countIf24(s, r, p, p);
    countIf24(r, s, p, p);
    countIf24(p, s, r, p);
    countIf24(s, p, r, p);
    countIf24(p, p, s, r);
    countIf24(s, p, p, r);
    countIf24(p, s, p, r);
    cout << count << " solution(s) found" << endl;
    cout << endl;
    is24(p, p, r, s);
    is24(r, p, p, s);
    is24(p, r, p, s);
    is24(r, p, s, p);
    is24(p, r, s, p);
    is24(s, r, p, p);
    is24(r, s, p, p);
    is24(p, s, r, p);
    is24(s, p, r, p);
    is24(p, p, s, r);
    is24(s, p, p, r);
    is24(p, s, p, r);
}

```

```

void doubleTwoSameNumbers(float p, float q, float r, float s){
    countIf24(p, p, r, r);
    countIf24(r, p, p, r);
    countIf24(p, r, p, r);
    countIf24(r, p, r, p);
    countIf24(p, r, r, p);
    countIf24(r, r, p, p);
    cout << count << " solution(s) found" << endl;
    cout << endl;
    is24(p, p, r, r);
    is24(r, p, p, r);
    is24(p, r, p, r);
    is24(r, p, r, p);
    is24(p, r, r, p);
    is24(r, r, p, p);
}

void threeSameNumbers(float p, float q, float r, float s){
    countIf24(p, p, p, s);
    countIf24(p, p, s, p);
    countIf24(p, s, p, p);
    countIf24(s, p, p, p);
    cout << count << " solution(s) found" << endl;
    cout << endl;
    is24(p, p, p, s);
    is24(p, p, s, p);
    is24(p, s, p, p);
    is24(s, p, p, p);
}

```

```

void countIf24(float a, float b, float c, float d){
    int i, j, k;

    for (i = 1; i <= 4; i++){
        for (j = 1; j <= 4; j++){
            for (k = 1; k <= 4; k++){
                counter(a, b, c, d, i, j, k);
            }
        }
    }
}

void is24(float a, float b, float c, float d){
    int i, j, k;

    for (i = 1; i <= 4; i++){
        for (j = 1; j <= 4; j++){
            for (k = 1; k <= 4; k++){
                printResult(a, b, c, d, i, j, k);
            }
        }
    }
}

```

```

float combinationOfCalculation(float x, float y, int flag){
    switch(flag){
        case 1:
            return (x+y);
            break;
        case 2:
            return (x*y);
            break;
        case 3:
            return (x-y);
            break;
        case 4:
            return (x/y);
            break;
    }
}

string flagToOperator(int flag){
    switch(flag){
        case 1:
            return "+";
            break;
        case 2:
            return "*";
            break;
        case 3:
            return "-";
            break;
        case 4:
            return "/";
            break;
    }
    return 0;
}

```

```

void counter(float a, float b, float c, float d, int i, int j, int k){
    float result;

    result = combinationOfCalculation(combinationOfCalculation(a, b, i), combinationOfCalculation(c, d, k), j);
    // operasi kurung: (a + b) + (c + d)
    if (result > 23.99999 && result < 24.00001){
        count++;
    }

    result = combinationOfCalculation(combinationOfCalculation(combinationOfCalculation(a, b, i), c, j), d, k);
    // operasi kurung: ((a + b) + c) + d
    if (result > 23.99999 && result < 24.00001){
        count++;
    }

    result = combinationOfCalculation(combinationOfCalculation(a, combinationOfCalculation(b, c, j), i), d, k);
    // operasi kurung: (a + (b + c)) + d
    if (result > 23.99999 && result < 24.00001){
        count++;
    }

    result = combinationOfCalculation(a, combinationOfCalculation(combinationOfCalculation(b, c, j), d, k), i);
    // operasi kurung: a + ((b + c) + d)
    if (result > 23.99999 && result < 24.00001){
        count++;
    }

    result = combinationOfCalculation(a, combinationOfCalculation(b, combinationOfCalculation(c, d, k), j), i);
    // operasi kurung: a + (b + (c + d))
    if (result > 23.99999 && result < 24.00001){
        count++;
    }
}

void printResult(float a, float b, float c, float d, int i, int j, int k){
    float result;

    result = combinationOfCalculation(combinationOfCalculation(a, b, i), combinationOfCalculation(c, d, k), j);
    // operasi kurung: (a + b) + (c + d)
    if (result > 23.99999 && result < 24.00001){
        cout << "(" << a << " " << flagToOperator(i) << " " << b << " " <<
            flagToOperator(j) << " (" << c << " " << flagToOperator(k) << " " << d << endl;
        solutions.push_back("(" + to_string((int)a) + " " + flagToOperator(i) + " " + to_string((int)b) + " " +
            flagToOperator(j) + " (" + to_string((int)c) + " " + flagToOperator(k) + " " + to_string((int)d) + ")");
    }

    result = combinationOfCalculation(combinationOfCalculation(combinationOfCalculation(a, b, i), c, j), d, k);
    // operasi kurung: ((a + b) + c) + d
    if (result > 23.99999 && result < 24.00001){
        cout << "(" << a << " " << flagToOperator(i) << " " << b << " " <<
            flagToOperator(j) << " " << c << " " << flagToOperator(k) << " " << d << endl;
        solutions.push_back("(" + to_string((int)a) + " " + flagToOperator(i) + " " + to_string((int)b) + " " +
            flagToOperator(j) + " " + to_string((int)c) + " " + flagToOperator(k) + " " + to_string((int)d));
    }

    result = combinationOfCalculation(combinationOfCalculation(a, combinationOfCalculation(b, c, j), i), d, k);
    // operasi kurung: (a + (b + c)) + d
    if (result > 23.99999 && result < 24.00001){
        cout << "(" << a << " " << flagToOperator(i) << " (" << b << " " <<
            flagToOperator(j) << " " << c << " " << flagToOperator(k) << " " << d << endl;
        solutions.push_back("(" + to_string((int)a) + " " + flagToOperator(i) + " (" + to_string((int)b) + " " +
            flagToOperator(j) + " " + to_string((int)c) + ") " + flagToOperator(k) + " " + to_string((int)d));
    }
}

```



```

result = combinationOfCalculation(a, combinationOfCalculation(combinationOfCalculation(b, c, j), d, k), i);
// operasi kurung: a + ((b + c) + d)
if (result > 23.99999 && result < 24.00001){
    cout << a << " " << flagToOperator(i) << " (" << b << " " << flagToOperator(j) <<
        " " << c << ") " << flagToOperator(k) << " " << d << ")" << endl;
    solutions.push_back(to_string((int)a) + " " + flagToOperator(i) + " (" + to_string((int)b) + " " + flagToOperator(j) +
        " " + to_string((int)c) + ") " + flagToOperator(k) + " " + to_string((int)d) + ")");
}

result = combinationOfCalculation(a, combinationOfCalculation(b, combinationOfCalculation(c, d, k), j), i);
// operasi kurung: a + (b + (c + d))
if (result > 23.99999 && result < 24.00001){
    cout << a << " " << flagToOperator(i) << " (" << b << " " << flagToOperator(j) <<
        " (" << c << " " << flagToOperator(k) << " " << d << ")") << endl;
    solutions.push_back(to_string((int)a) + " " + flagToOperator(i) + " (" + to_string((int)b) + " " + flagToOperator(j) +
        " (" + to_string((int)c) + " " + flagToOperator(k) + " " + to_string((int)d) + ")")");
}
}

void saveToFile(string filename){
    ofstream output("../test/" + filename);

    if (output.is_open()){
        output << count << " solution(s) found" << "\n";
        for (int i=0; i<solutions.size(); i++){
            output << solutions.at(i) << "\n";
        }
        output << "Execution time is " << exec;
    }
    output.close();
    cout << "File berhasil disimpan" << endl;
}

```

Gambar 2.5 Implementasi Void dan Function

3 Tampilan Program dan Contoh Input/Output

Program dibuat dengan CLI (*Command Line Interface*). Pengguna dapat menjalankan program run.bat setelah berada di folder directory Tucil1_13521164, untuk proses compiling dan menjalankan program. File hasil compile akan tersimpan di folder bin. Program dijalankan dengan memasukkan input sesuai dengan instruksi yang ada pada terminal.

```
C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>run.bat

C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>g++ src/main.cpp -o bin/main.exe

C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>bin/main.exe
Hello, welcome to make it 24 Card Game!
How would you like to input your card value?
1. Manual
2. Random
Your choice is: |
```

Gambar 3.1 Tampilan awal ketika program dijalankan

```
Input your 4 cards value:
A 3 5 7

8 solution(s) found

(1 + 5) * (7 - 3)
(3 - 1) * (5 + 7)
(3 - 1) * (7 + 5)
(5 + 1) * (7 - 3)
(5 + 7) * (3 - 1)
(7 - 3) * (1 + 5)
(7 - 3) * (5 + 1)
(7 + 5) * (3 - 1)

Execution time is 0.007 s
Would you like to save solutions into file?
1. Yes
2. No
```

Gambar 3.2 Pengguna Menginput Manual Nilai Kartu

```
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: test1
File berhasil disimpan di folder test
```

Gambar 3.3 Pengguna Menyimpan Output ke Dalam File test1.txt

```

Hello, welcome to make it 24 Card Game!
How would you like to input your card value?
1. Manual
2. Random
Your choice is: 2

Your random cards: 9 J A A
0 solution(s) found

Execution time is 0.001 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: |

```

Gambar 3.4 Pengguna Menggunakan Generate Random Card

```

Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 2

Okay then, see you later!

C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>|

```

Gambar 3.5 Pengguna Tidak Ingin Menyimpan Output ke Dalam File

Berikut adalah 3 contoh input secara manual dan 3 contoh input secara random beserta outputnya.

```
Hello, welcome to make it 24 Card Game!
How would you like to input your card value?
1. Manual
2. Random
Your choice is: 1

Input your 4 cards value:
K Q 2 4

0 solution(s) found

Execution time is 0.001 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: testManual1
File berhasil disimpan di folder test
PS C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>
```

Gambar 3.6 Input Manual Kartu K Q 2 4

```

Hello, welcome to make it 24 Card Game!
How would you like to input your card value?
1. Manual
2. Random
Your choice is: 1

Input your 4 cards value:
6 4 J A

20 solution(s) found

(6 - 4) * (11 + 1)
(6 - 4) * (1 + 11)
4 * ((11 - 6) + 1)
4 * (11 - (6 - 1))
4 * ((11 + 1) - 6)
4 * (11 + (1 - 6))
4 * ((1 - 6) + 11)
4 * (1 - (6 - 11))
4 * ((1 + 11) - 6)
4 * (1 + (11 - 6))
((11 - 6) + 1) * 4
(11 - (6 - 1)) * 4
(11 + 1) * (6 - 4)
((11 + 1) - 6) * 4
(11 + (1 - 6)) * 4
((1 - 6) + 11) * 4
(1 - (6 - 11)) * 4
(1 + 11) * (6 - 4)
((1 + 11) - 6) * 4
(1 + (11 - 6)) * 4

Execution time is 0.018 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: testManual2
File berhasil disimpan di folder test
PS C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>

```

Gambar 3.7 Input Manual Kartu 6 4 J A

```

Input your 4 cards value:
3 Q 7 9

30 solution(s) found

(3 * (12 - 7)) + 9
3 + ((12 - 9) * 7)
3 + (7 * (12 - 9))
(3 * 7) + (12 - 9)
((3 * 7) + 12) - 9
((3 * 7) - 9) + 12
(3 * 7) - (9 - 12)
3 - (7 * (9 - 12))
3 - ((9 - 12) * 7)
(12 + (3 * 7)) - 9
12 + ((3 * 7) - 9)
(12 + (7 * 3)) - 9
12 + ((7 * 3) - 9)
((12 - 7) * 3) + 9
(12 - 9) + (3 * 7)
12 - (9 - (3 * 7))
(12 - 9) + (7 * 3)
((12 - 9) * 7) + 3
12 - (9 - (7 * 3))
(7 * 3) + (12 - 9)
((7 * 3) + 12) - 9
((7 * 3) - 9) + 12
(7 * 3) - (9 - 12)
((7 - 3) * 9) - 12
(7 * (12 - 9)) + 3
9 + (3 * (12 - 7))
9 - (3 * (7 - 12))
9 + ((12 - 7) * 3)
(9 * (7 - 3)) - 12
9 - ((7 - 12) * 3)

Execution time is 0.027 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: testManual3
File berhasil disimpan di folder test
PS C:\Users\Lenovo\Documents\Stima\Tucil1\Tucil1_13521164>

```

Gambar 3.8 Input Manual 3 Q 7 9

```

Hello, welcome to make it 24 Card Game!
How would you like to input your card value?
1. Manual
2. Random
Your choice is: 2

Your random cards: K 4 7 Q
24 solution(s) found

(13 - (4 + 7)) * 12
((13 - 4) - 7) * 12
(13 - (7 + 4)) * 12
((13 - 7) - 4) * 12
4 * ((7 - 13) + 12)
4 * (7 - (13 - 12))
4 * ((7 + 12) - 13)
4 * (7 + (12 - 13))
4 * ((12 - 13) + 7)
4 * (12 - (13 - 7))
4 * ((12 + 7) - 13)
4 * (12 + (7 - 13))
((7 - 13) + 12) * 4
(7 - (13 - 12)) * 4
((7 + 12) - 13) * 4
(7 + (12 - 13)) * 4
12 * (13 - (4 + 7))
12 * ((13 - 4) - 7)
12 * (13 - (7 + 4))
12 * ((13 - 7) - 4)
((12 - 13) + 7) * 4
(12 - (13 - 7)) * 4
((12 + 7) - 13) * 4
(12 + (7 - 13)) * 4

Execution time is 0.021 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: testRandom1
File berhasil disimpan di folder test
PS C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>

```

Gambar 3.9 Input Random dan Memperoleh Kartu K 4 7 Q

```

Hello, welcome to make it 24 Card Game!
How would you like to input your card value?
1. Manual
2. Random
Your choice is: 2

Your random cards: J 9 K 7
0 solution(s) found

Execution time is 0 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: testRandom2
File berhasil disimpan di folder test
PS C:\Users\Lenovo\Documents\Stima\Tucil1\Tucil1_13521164>

```

Gambar 3.10 Input Random dan Memperoleh Kartu J 9 K 7


```

How would you like to input your card value?
1. Manual
2. Random
Your choice is: 2

Your random cards: 6 9 5 J
32 solution(s) found

(6 * (9 + 11)) / 5
6 * ((9 + 11) / 5)
(6 / 5) * (9 + 11)
6 / (5 / (9 + 11))
(6 / 5) * (11 + 9)
6 / (5 / (11 + 9))
(6 * (11 + 9)) / 5
6 * ((11 + 9) / 5)
(9 / 6) * (5 + 11)
9 / (6 / (5 + 11))
(9 / 6) * (11 + 5)
9 / (6 / (11 + 5))
(9 * (5 + 11)) / 6
9 * ((5 + 11) / 6)
(9 + 11) * (6 / 5)
((9 + 11) * 6) / 5
((9 + 11) / 5) * 6
(9 + 11) / (5 / 6)
(9 * (11 + 5)) / 6
9 * ((11 + 5) / 6)
((5 + 11) / 6) * 9
(5 + 11) / (6 / 9)
(5 + 11) * (9 / 6)
((5 + 11) * 9) / 6
(11 + 9) * (6 / 5)
((11 + 9) * 6) / 5
((11 + 9) / 5) * 6
(11 + 9) / (5 / 6)
((11 + 5) / 6) * 9
(11 + 5) / (6 / 9)
(11 + 5) * (9 / 6)
((11 + 5) * 9) / 6

Execution time is 0.035 s
Would you like to save solutions into file?
1. Yes
2. No
Your choice is: 1

input your txt filename: testRandom3
File berhasil disimpan di folder test
PS C:\Users\Lenovo\Documents\Stima\Tucil\Tucil1_13521164>





```

Gambar 3.11 Input Random dan Memperoleh Kartu 6 9 5 J

4 Link Repository

Repository Github dapat diakses pada: https://github.com/blixa-rd/Tucil1_13521164

5 Ceklis

No	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 (mencapai 24)		
5	Program dapat menyimpan solusi dalam file teks	