

**Tugas Kecil 1 IF2211 Strategi Algoritma
Penyelesaian Permainan Kartu 24 dengan
Algoritma *Brute Force***



Disusun oleh:
Eunice Sarah Siregar
13521013

**Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022/2023**

BAB I

PENDAHULUAN

A. Algoritma Brute Force

Brute Force adalah pendekatan yang *straightforward* dalam memecahkan suatu masalah atau persoalan yang bersifat *to the point*, jelas, dan sederhana. Algoritma ini memiliki karakteristik seperti berikut:

- membutuhkan memori yang besar dan waktu yang lama dalam menyelesaikan masalah,
- cocok untuk persoalan yang bersifat sederhana dengan pertimbangan seperti implementasinya yang mudah mengingat cara kerja algoritma ini, dan
- walaupun kurang efisien, tetapi algoritma *brute force* terkenal sebagai solusi dari hampir semua masalah (*wide applicability*).

Exhaustive Search adalah teknik pencarian solusi secara brute force untuk persoalan-persoalan kombinatorial, seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan. Walaupun menggunakan teknik *exhaustive search* memecahkan permasalahan, tetapi waktu dan sumber daya yang dikeluarkan juga cukup besar. Adapun permasalahan-permasalahan yang menggunakan teknik *exhaustive search* seperti *travelling salesperson problem*, *1/0 knapsack*, kriptografi, dan lain-lain.

B. Permainan Kartu 24

Permainan kartu 24 merupakan permainan yang mampu meningkatkan kecerdasan otak karena menguji keterampilan menghitung pemain. Dalam permainan ini, pemain diminta untuk mengubah empat angka yang dimiliki sampai mendapatkan hasil akhir 24 dengan operasi-operasi aritmatika sederhana (\times , \div , $+$, dan $-$). Permainan ini biasa dimainkan menggunakan kartu remi yang berisikan 52 kartu berbeda, dengan rincian 4 simbol (hati, wajik, sekop, dan keriting) dan 13 kartu (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, dan K) dengan A bernilai 1, J bernilai 11, Q bernilai 12, dan K bernilai 13. Pada program ini, simbol-simbol yang dimiliki tiap kartu diabaikan.

C. Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force

Untuk menyelesaikan permainan kartu 24, penulis menggunakan algoritma brute force dengan deskripsi langkah-langkah sebagai berikut.

- Awalnya, pengguna akan diminta memasukkan sebanyak empat angka dengan syarat angka bernilai 2 - 9 atau A, K, Q, J atau memilih untuk program men-generate angka sendiri
- Selanjutnya akan dilakukan empat kali pengulangan dengan tiga diantaranya untuk permutasi operasi aritmatika dan sisanya untuk permutasi jenis kurung yang akan dipakai, seperti berikut:

$$((x_1 \text{ sym } x_2) \text{ sym } x_3) \text{ sym } x_4$$

$(x_1 \text{ sym } (x_2 \text{ sym } x_3)) \text{ sym } x_4$

$x_1 \text{ sym } ((x_2 \text{ sym } x_3) \text{ sym } x_4)$

$x_1 \text{ sym } (x_2 \text{ sym } (x_3 \text{ sym } x_4))$

$(x_1 \text{ sym } x_2) \text{ sym } (x_3 \text{ sym } x_4)$

dengan sym adalah operator aritmatika sederhana (\times , \div , $+$, dan $-$) dan disimpan kedalam sebuah variabel dengan tipe data *string*.

3. Setelah itu, dilakukan perhitungan sesuai dengan tingkatan operator aritmatika. Ketika hasil dari perhitungan tersebut memiliki selisih kurang dari 0.000001 dengan 24, maka variabel tersebut dimasukkan ke dalam *vector* bertipe data *string*. Selain itu, program juga mengecek untuk menghindari operasi yang duplikat. Untuk kombinasi angka, penulis menggunakan penukaran secara manual didalam *looping*.
4. Selanjutnya, program akan mengeluarkan seluruh hasil operasi yang memenuhi perhitungan.
5. Terdapat opsi untuk menyimpan hasil operasi kedalam .txt dan juga opsi untuk mengulang program kembali.

BAB II

SOURCE PROGRAM DALAM C++

A. Algorithm.cpp

```
#include <string>
#include <vector>
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <bits/stdc++.h>
#include <chrono>
#include "splashscreen.cpp"

using namespace std;

int convert(string X){
    if (X == "2" || X == "3" || X == "4" || X == "5" || X == "6" || X ==
"7" || X == "8" || X == "9" || X == "10"){
        return stoi(X);
    }
    else if (X == "A")
    {
        return 1;
    }
    else if (X == "K")
    {
        return 13;
    }
    else if (X == "Q")
    {
        return 12;
    }
    else if (X == "J")
    {
        return 11;
    }
}

bool validate(string X){
    if (X == "A"){
        return true;
    }
    else if (X == "K")
    {
        return true;
    }
    else if (X == "Q")
    {
        return true;
    }
    else if (X == "J")
    {
        return true;
    }
}
```

```

        else if(X == "2" || X == "3" || X == "4" || X == "5" || X == "6" || X
== "7" || X == "8" || X == "9" || X == "10"){
            return true;
        }
        else{
            return false;
        }
    }
float division(int a, int b) {
    if (b == 0) {
        return 0;
    } else {
        return (float) a / b;
    }
}
float calculate(int x1, int x2, int x3, int x4, int op1, int op2, int op3, int
bracket){
    float result = 0;
    float results = 0;
    if (bracket == 0){
        if (op1 == 0){
            result = x1 + x2;
        }
        else if (op1 == 1){
            result = x1 - x2;
        }
        else if (op1 == 2){
            result = division(x1, x2);
        }
        else{
            result = x1 * x2;
        }

        if (op2 == 0){
            result += x3;
        }
        else if (op2 == 1){
            result -= x3;
        }
        else if (op2 == 2){
            result = division(result, x3);
        }
        else{
            result *= x3;
        }

        if (op3 == 0){ result += x4; }
        else if (op3 == 1){
            result -= x4;
        }
        else if (op3 == 2){
            result = division(result, x4);
        }
        else{
            result *= x4;
        }
    }
}

```

```

}

if (bracket == 1){
    if (op2 == 0){
        result = x2 + x3;
    }
    else if (op2 == 1){
        result = x2 - x3;
    }
    else if (op2 == 2){
        result = division(x2, x3);
    }
    else{
        result = x2 * x3;
    }

    if (op1 == 0){
        result += x1;
    }
    else if (op1 == 1){
        result = x1 - result;
    }
    else if (op1 == 2){
        result = division(x1, result);
    }
    else{
        result *= x1;
    }

    if (op3 == 0){
        result += x4;
    }
    else if (op3 == 1){
        result -= x4;
    }
    else if (op3 == 2){
        result = division(result, x4);
    }
    else{
        result *= x4;
    }
}

if (bracket == 2){
    if (op2 == 0){
        result = x2 + x3;
    }
    else if (op2 == 1){
        result = x2 - x3;
    }
    else if (op2 == 2){
        result = division(x2, x3);
    }
    else{
        result = x2 * x3;
    }
}

```

```

    if (op3 == 0){
        result += x4;
    }
    else if (op3 == 1){
        result -= x4;
    }
    else if (op3 == 2){
        result = division(result, x4);
    }
    else{
        result *= x4;
    }

    if (op1 == 0){
        result += x1;
    }
    else if (op1 == 1){
        result = x1 - result;
    }
    else if (op1 == 2){
        result = division(x1, result);
    }
    else{
        result *= x1;
    }
}

if (bracket == 3){
    if (op3 == 0){
        result = x3 + x4;
    }
    else if (op3 == 1){
        result = x3 - x4;
    }
    else if (op3 == 2){
        result = division(x3, x4);
    }
    else{
        result = x3 * x4;
    }

    if (op2 == 0){
        result += x2;
    }
    else if (op2 == 1){
        result = x2 - result;
    }
    else if (op2 == 2){
        result = division(x2, result);
    }
    else{
        result *= x2;
    }

    if (op1 == 0){

```

```

        result += x1;
    }
    else if (op1 == 1){
        result = x1 - result;
    }
    else if (op1 == 2){
        result = division(x1, result);
    }
    else{
        result = result * x1;
    }
}

if (bracket == 4){
    if (op1 == 0){
        result = x1 + x2;
    }
    else if (op1 == 1){
        result = x1 - x2;
    }
    else if (op1 == 2){
        result = division(x1, x2);
    }
    else{
        result = x1 * x2;
    }

    if (op3 == 0){
        results = x3 + x4;
    }
    else if (op3 == 1){
        results = x3 - x4;
    }
    else if (op3 == 2){
        results = division(x3, x4);
    }
    else{
        result = x3 * x4;
    }

    if (op2 == 0){
        result += results;
    }
    else if (op2 == 1){
        result -= results;
    }
    else if (op2 == 2){
        result = division(result, results);
    }
    else{
        result = result * results;
    }
}
return result;
}

```



```

vector <string> addList(vector <string> list, int x1, int x2, int x3, int x4,
int op1, int op2, int op3, int bracket){
    string sym1, sym2, sym3, temp;
    if (op1 == 0){
        sym1 = "+";
    }
    else if (op1 == 1) {
        sym1 = "-";
    }
    else if (op1 == 2) {
        sym1 = "/";
    }
    else {
        sym1 = "*";
    }

    if (op2 == 0){
        sym2 = "+";
    }
    else if (op2 == 1) {
        sym2 = "-";
    }
    else if (op2 == 2) {
        sym2 = "/";
    }
    else {
        sym2 = "*";
    }

    if (op3 == 0){
        sym3 = "+";
    }
    else if (op3 == 1) {
        sym3 = "-";
    }
    else if (op3 == 2) {
        sym3 = "/";
    }
    else {
        sym3 = "*";
    }

    if(bracket == 0) {
        temp = "(" + to_string(x1) + sym1 + to_string(x2) + ")" + sym2 +
to_string(x3) + ")" + sym3 + to_string(x4);
    }
    else if(bracket == 1) {
        temp = "(" + to_string(x1) + sym1 + "(" + to_string(x2) + sym2 +
to_string(x3) + ")") + sym3 + to_string(x4);
    }
    else if(bracket == 2) {
        temp = to_string(x1) + sym1 + "(" + to_string(x2) + sym2 +
to_string(x3) + ")" + sym3 + to_string(x4) + ")";
    }
    else if(bracket == 3) {
        temp = to_string(x1) + sym1 + "(" + to_string(x2) + sym2 + "(" +

```

```

to_string(x3) + sym3 + to_string(x4) + "));";
    }
    else if(bracket == 4) {
        temp = "(" + to_string(x1) + sym1 + to_string(x2) + ")" + sym2 + "(" +
to_string(x3) + sym3 + to_string(x4) + "));";
    }

    list.push_back(temp);
    return list;
}

vector <string> combine(vector <string> list, int x1, int x2, int x3, int x4,
int op1, int op2, int op3, int bracket){
    vector <string> solutions ;
    for (int i = 0; i < 4; i++){
        for (int j = 0; j < 4; j++){
            for (int k = 0; k < 4; k++){
                for (int l = 0; l < 5; l++){
                    if(abs(calculate(x1, x2, x3, x4, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x1, x2, x3, x4, i, j,
k, l);
                    }
                    if(abs(calculate(x1, x3, x4, x2, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x1, x3, x4, x2, i, j,
k, l);
                    }
                    if(abs(calculate(x1, x4, x2, x3, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x1, x4, x2, x3, i, j,
k, l);
                    }
                    if(abs(calculate(x1, x2, x4, x3, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x1, x2, x4, x3, i, j,
k, l);
                    }
                    if(abs(calculate(x1, x3, x2, x4, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x1, x3, x2, x4, i, j,
k, l);
                    }
                    if(abs(calculate(x1, x4, x3, x2, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x1, x4, x3, x2, i, j,
k, l);
                    }
                    if(abs(calculate(x2, x3, x4, x1, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x2, x3, x4, x1, i, j,
k, l);
                    }
                    if(abs(calculate(x2, x4, x1, x3, i, j, k, l) - 24) <
0.000001) {
                        solutions = addList(solutions, x2, x4, x1, x3, i, j,

```

```

k, 1);
    }
    if(abs(calculate(x2, x1, x3, x4, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x2, x1, x3, x4, i, j,
k, 1);
    }
    if(abs(calculate(x2, x3, x1, x4, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x2, x3, x1, x4, i, j,
k, 1);
    }
    if(abs(calculate(x2, x4, x3, x1, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x2, x4, x3, x1, i, j,
k, 1);
    }
    if(abs(calculate(x2, x1, x4, x3, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x2, x1, x4, x3, i, j,
k, 1);
    }
    if(abs(calculate(x3, x4, x1, x2, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x3, x4, x1, x2, i, j,
k, 1);
    }
    if(abs(calculate(x3, x1, x2, x4, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x3, x1, x2, x4, i, j,
k, 1);
    }
    if(abs(calculate(x3, x2, x4, x1, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x3, x2, x4, x1, i, j,
k, 1);
    }
    if(abs(calculate(x3, x4, x2, x1, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x3, x4, x2, x1, i, j,
k, 1);
    }
    if(abs(calculate(x3, x1, x4, x2, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x3, x1, x4, x2, i, j,
k, 1);
    }
    if(abs(calculate(x3, x2, x1, x4, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x3, x2, x1, x4, i, j,
k, 1);
    }
    if(abs(calculate(x4, x1, x2, x3, i, j, k, 1) - 24) <
0.000001) {
        solutions = addList(solutions, x4, x1, x2, x3, i, j,

```

```

k, l);
    }
    if(abs(calculate(x4, x2, x3, x1, i, j, k, l) - 24) <
0.000001) {
        solutions = addList(solutions, x4, x2, x3, x1, i, j,
k, l);
    }
    if(abs(calculate(x4, x3, x1, x2, i, j, k, l) - 24) <
0.000001) {
        solutions = addList(solutions, x4, x3, x1, x2, i, j,
k, l);
    }
    if(abs(calculate(x4, x1, x3, x2, i, j, k, l) - 24) <
0.000001) {
        solutions = addList(solutions, x4, x1, x3, x2, i, j,
k, l);
    }
    if(abs(calculate(x4, x2, x1, x3, i, j, k, l) - 24) <
0.000001) {
        solutions = addList(solutions, x4, x2, x1, x3, i, j,
k, l);
    }
    if(abs(calculate(x4, x3, x2, x1, i, j, k, l) - 24) <
0.000001) {
        solutions = addList(solutions, x4, x3, x2, x1, i, j,
k, l);
    }
    }
    }
    }
    return solutions;
}

void saveFile(vector <string> a, string filename, float x1, float x2, float
x3, float x4, float time) {
    ofstream myfile;
    myfile.open("../test/" + filename + ".txt");
    myfile << "Jumlah solusi yang didapatkan adalah: " << a.size() << endl;
    myfile << "Waktu yang dibutuhkan adalah: " << time << " detik" << endl;
    for (int i = 0; i < a.size(); i++) {
        myfile << a[i] << endl;
    }
    myfile.close();
}

void execute(){
    char pilih;
    string X1, X2, X3, X4;
    int x1, x2, x3, x4;
    vector <string> solutions;
    cout << "Apakah anda ingin memasukkan angka secara manual? (y/n)" << endl;
    cin >> pilih;
    while(pilih != 'y' && pilih != 'n'){
        cout << "Apakah anda ingin memasukkan angka secara manual? (y/n)" <<
endl;

```

```

        cin >> pilih;
    }
    if (pilih == 'y'){
        cout << "Masukkan 4 angka yang akan dihitung (Masukkan berupa angka
2-9 atau A K J Q" << endl;
        cin >> X1 >> X2 >> X3 >> X4;
        while (validate(X1) != true || validate(X2) != true || validate(X3) !=
true || validate(X4) != true)
        {
            cout << "Masukkan 4 angka yang akan dihitung" << endl;
            cin >> X1 >> X2 >> X3 >> X4;
        }
        x1 = convert(X1);
        x2 = convert(X2);
        x3 = convert(X3);
        x4 = convert(X4);
    }
    else if (pilih == 'n')
    {
        srand(time(0));
        x1 = (rand() % 13) + 1;
        x2 = (rand() % 13) + 1;
        x3 = (rand() % 13) + 1;
        x4 = (rand() % 13) + 1;
        cout << x1 << " " << x2 << " " << x3 << " " << x4 << endl;
    }
    auto start = chrono::high_resolution_clock::now();
    vector <string> a = combine(solutions, x1, x2, x3, x4, 0, 0, 0, 0);
    auto end = a.end();
    for (auto it = a.begin(); it != end; ++it) {
        end = remove(it + 1, end, *it);
    }
    a.erase(end, a.end());
    for (int i = 0; i < a.size(); i++) {
        cout << a[i] << endl;
    }
    cout << "Jumlah solusi yang didapatkan adalah: " << a.size() << endl;
    auto finish = chrono::high_resolution_clock::now();
    float extime = chrono::duration_cast<chrono::nanoseconds>(finish -
start).count();
    extime *= 1e-9;
    cout << "Waktu yang dibutuhkan adalah: " << extime << " detik" << endl;

    char save;
    cout << "Apakah anda ingin menyimpan hasil ke file? (y/n)" << endl;
    cin >> save;
    if (save == 'y'){
        string filename;
        cout << "Masukkan nama file: ";
        cin >> filename;
        saveFile(a, filename, x1, x2, x3, x4, extime);
        cout << "File berhasil disimpan" << endl;
    }
    else if (save == 'n'){
        cout << "Terima kasih" << endl;    }
}

```

```

int main(){
    char loop;
    do{
        splashscreen();
        execute();
        cout << "Apakah anda ingin mengulang program? (y/n)" << endl;
        cin >> loop;
        system("CLS");
    } while(loop == 'y');
}

```

B. splashscreen.cpp

```

#include <iostream>

using namespace std;

int splashscreen(){
    cout <<"          " << endl;
    cout <<" |2 . |          " << endl;
    cout <<" | /.\\ ||S ^ |          " << endl;
    cout <<" | (._.) || / \\ ||O _ |          " << endl;
    cout <<" | | || \\ / || ( ) ||L _ |" << endl;
    cout <<" |___4|| . || ( '_ ) || ( v )|" << endl;
    cout <<" |___ V|| _ | || \\ / |" << endl;
    cout <<" |___ E|| . |" << endl;
    cout <<" |___ R|" << endl;
    cout <<" Welcome to 24 Solver" << endl;
}

```

BAB III

TEST CASE

```

      2 .
      /.\
      (..)
      |
      4

      S ^
      / \
      \ /
      .
      v

      0 _
      ( )
      ( _ )
      |
      E

      L _
      ( v )
      \ /
      .
      R

Welcome to 24 Solver
Apakah anda ingin memasukkan angka secara manual? (y/n)
y
Masukkan 4 angka yang akan dihitung
K 10 8 2
(8+(2*13))-10
(8+(13*2))-10
8+((2*13)-10)
8+((13*2)-10)
8-(10-(2*13))
8-(10-(13*2))
((10-8)*13)-2
((13*2)+8)-10
((2*13)+8)-10
(13*2)+(8-10)
(2*13)+(8-10)
((13*2)-10)+8
((2*13)-10)+8
(13*(10-8))-2
(13*2)-(10-8)
(2*13)-(10-8)
Jumlah solusi yang didapatkan adalah: 16

```

Gambar 1. Hasil Test Case I

```

      2 .
      /.\
      (..)
      |
      4

      S ^
      / \
      \ /
      .
      v

      0 _
      ( )
      ( _ )
      |
      E

      L _
      ( v )
      \ /
      .
      R

Welcome to 24 Solver
Apakah anda ingin memasukkan angka secara manual? (y/n)
y
Masukkan 4 angka yang akan dihitung
9 J 10 7
10+((11-9)*7)
10+(7*(11-9))
10-((9-11)*7)
((11-9)*7)+10
((10-7)*11)-9
10-(7*(9-11))
(7*(11-9))+10
(11*(10-7))-9
Jumlah solusi yang didapatkan adalah: 8
Waktu yang dibutuhkan adalah: 0.002476 detik

```

Gambar 2. Hasil Test Case II

```

  2 . | S ^ | 0 | L _ |
  / \ | / \ | ( ) | ( v ) |
  ( _ ) | \ / | ( ' ) | \ / |
  | | . | | | | |
  _ 4 | _ V | _ E | _ R |

Welcome to 24 Solver
Apakah anda ingin memasukkan angka secara manual? (y/n)
y
Masukkan 4 angka yang akan dihitung
6 6 6 6
((6+6)+6)+6
(6+(6+6))+6
6+((6+6)+6)
6+(6+(6+6))
(6+6)+(6+6)
(6*6)-(6+6)
((6*6)-6)-6
Jumlah solusi yang didapatkan adalah: 7
Waktu yang dibutuhkan adalah: 0.009642 detik

```

Gambar 3. Hasil Test Case III

```

  2 . | S ^ | 0 | L _ |
  / \ | / \ | ( ) | ( v ) |
  ( _ ) | \ / | ( ' ) | \ / |
  | | . | | | | |
  _ 4 | _ V | _ E | _ R |

Welcome to 24 Solver
Apakah anda ingin memasukkan angka secara manual? (y/n)
n
9 10 3 9
((9+9)-10)*3
(9+(9-10))*3
((9-10)+9)*3
(9-(10-9))*3
3*((9+9)-10)
3*(9+(9-10))
3*((9-10)+9)
3*(9-(10-9))
Jumlah solusi yang didapatkan adalah: 8
Waktu yang dibutuhkan adalah: 0.003216 detik

```

Gambar 4. Hasil Test Case IV


```

Welcome to 24 Solver
Apakah anda ingin memasukkan angka secara manual? (y/n)
n
13 8 5 6
((13+5)/6)*8
((5+13)/6)*8
((13+5)*8)/6
((5+13)*8)/6
(13+5)*(8/6)
(5+13)*(8/6)
(8/(13/5))*6
(6/(13/5))*8
(8/6)*(13+5)
(8/6)*(5+13)
(8*(13+5))/6
(8*(5+13))/6
8*((13+5)/6)
8*((5+13)/6)
8*(6/(13/5))
6*(8/(13/5))
(8*6)/(13/5)
(6*8)/(13/5)
Jumlah solusi yang didapatkan adalah: 18
Waktu yang dibutuhkan adalah: 0.002976 detik

```

Gambar 5. Hasil Test Case V

```

  2  .  |  S ^  |  O _  |  L _  |
  / . \  |  / \  |  ( )  |  ( v )
 ( _ . _ ) |  \ /  |  ( ' _ ) |  \ /
  |         |  .   |  |         |
  _ _ _ 4   |  _ _ _ v |  _ _ _ E |  _ _ _ R
                                    

Welcome to 24 Solver
Apakah anda ingin memasukkan angka secara manual? (y/n)
n
11 11 7 4
(7-(11/11))*4
4*(7-(11/11))
Jumlah solusi yang didapatkan adalah: 2
Waktu yang dibutuhkan adalah: 0.000511 detik

```

Gambar 6. Hasil Test Case VI

LAMPIRAN

Link Github

https://github.com/eunicesarah/Tucil1_13521013.git

Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	V	
2. Program berhasil running	V	
3. Program dapat membaca input / generate sendiri dan memberikan luaran	V	
4. Solusi yang diberikan program memenuhi (berhasil mencapai 24)	V	
5. Program dapat menyimpan solusi dalam file teks	V	