

**Laporan Tugas Kecil 3**  
**IF 2211 Strategi Algoritma**  
**Penyelesaian Persoalan 15-Puzzle dengan Algoritma Branch**  
**and Bound**



**Disusun Oleh :**  
**Daffa Romyz Aufa 13520162**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**

## A. Algoritma Branch And Bound

- Memilih apakah posisi awal 15-puzzle acak atau diambil dari sebuah file teks. Jika file teks dipilih maka akan diminta untuk memasukan nama file.
- Program akan menentukan apakah posisi awal 15-puzzle dapat diselesaikan dengan menghitung  $\sum_{i=1}^{15} KURANG(i) + X$ . Jika ganjil maka puzzle tidak dapat diselesaikan dan akan mengeluarkan pesan. Jika genap maka program berlanjut.
- Status awal puzzle akan menjadi simpul R yang akan dibangkitkan anaknya. Simpul R akan ditambahkan dalam suatu List simpul hidup.
- Keluarkan Simpul R dari list simpul hidup.
- Bangkitkan simpul-simpul valid dari simpul R yang ubin kosongnya di geser ke atas, kanan, bawah, dan kiri. Simpul-simpul yang valid untuk dibangkitkan akan dimasukan ke dalam list simpul hidup.
- Tentukan simpul X dalam list simpul hidup yang memiliki cost paling kecil. Cost didapat dari jumlah ubin tidak kosong yang tidak terdapat pada susunan akhir dan jarak simpul awal ke simpul X.
- Tentukan apakah simpul X telah mencapai posisi akhir. Jika tidak, maka simpul X akan menjadi simpul R yang baru. Lakukan lagi Langkah 4-7 hingga simpul X telah mencapai posisi akhir. Jika iya, maka program akan menampilkan urutan matriks dari posisi awal ke posisi akhir, waktu eksekusi program, dan jumlah simpul yang dibangkitkan.

## B. Screen-shot input-output program

- Puzzle awal dari file teks rejected1.txt

```

C:\Windows\System32\cmd.exe
D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>java Main
Pilih Masukan :
[1] Acak
[2] File
>>> 2
Pilih File
Masukan Nama File:
>>> rejected1.txt

-----
|1 |3 |4 |15 |
|2 | |5 |12 |
|7 |6 |11 |14 |
|8 |9 |10 |13 |
-----
Kurang(1) = 0
Kurang(3) = 1
Kurang(4) = 1
Kurang(15) = 11
Kurang(2) = 0
Kurang() ubin kosong = 10
Kurang(5) = 0
Kurang(12) = 6
Kurang(7) = 1
Kurang(6) = 0
Kurang(11) = 3
Kurang(14) = 4
Kurang(8) = 0
Kurang(9) = 0
Kurang(10) = 0
Kurang(13) = 0
Total Fungsi Kurang adalah 37
Persoalan Tidak Dapat Diselesaikan

D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>
D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>

```

## 2. Puzzle awal dari file teks rejected2.txt

```

C:\Windows\System32\cmd.exe
D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>java Main
Pilih Masukan :
[1] Acak
[2] File
>>> 2
Pilih File
Masukan Nama File:
>>> rejected2.txt

-----
|3 |9 |1 |15 |
|14|11|4 |6 |
|13 | |10|12 |
|2 |7 |8 |5 |
-----
Kurang(3) = 2
Kurang(9) = 7
Kurang(1) = 0
Kurang(15) = 11
Kurang(14) = 10
Kurang(11) = 7
Kurang(4) = 1
Kurang(6) = 2
Kurang(13) = 6
Kurang() ubin kosong = 6
Kurang(10) = 4
Kurang(12) = 4
Kurang(2) = 0
Kurang(7) = 1
Kurang(8) = 1
Kurang(5) = 0
Total Fungsi Kurang adalah 63
Persoalan Tidak Dapat Diselesaikan

D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>

```

## 3. Puzzle awal dari file teks accepted1.txt

```

C:\Windows\System32\cmd.exe
D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>java Main
Pilih Masukan :
[1] Acak
[2] File
>>> 2
Pilih File
Masukan Nama File:
>>> accepted1.txt

-----
|1 |2 |3 |4 |
|5 |6 | |8 |
|9 |10|7 |11|
|13|14|15|12|
-----
Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 0
Kurang(4) = 0
Kurang(5) = 0
Kurang(6) = 0
Kurang() ubin kosong = 9
Kurang(8) = 1
Kurang(9) = 1
Kurang(10) = 1
Kurang(7) = 0
Kurang(11) = 0
Kurang(13) = 1
Kurang(14) = 1
Kurang(15) = 1
Kurang(12) = 0
Total Fungsi Kurang adalah 16
Solusi Ditemukan
Urutan Matriks :
-----
|1 |2 |3 |4 |

```

```
C:\Windows\System32\cmd.exe
Solusi Ditemukan
Urutan Matriks :

-----
|1 |2 |3 |4 |
-----
|5 |6 | |8 |
-----
|9 |10 |7 |11 |
-----
|13 |14 |15 |12 |
-----

-----
|1 |2 |3 |4 |
-----
|5 |6 |7 |8 |
-----
|9 |10 | |11 |
-----
|13 |14 |15 |12 |
-----

-----
|1 |2 |3 |4 |
-----
|5 |6 |7 |8 |
-----
|9 |10 |11 | |
-----
|13 |14 |15 |12 |
-----

-----
|1 |2 |3 |4 |
-----
|5 |6 |7 |8 |
-----
|9 |10 |11 |12 |
-----
|13 |14 |15 | |
-----

C:\Windows\System32\cmd.exe
|5 |6 | |8 |
-----
|9 |10 |7 |11 |
-----
|13 |14 |15 |12 |
-----

-----
|1 |2 |3 |4 |
-----
|5 |6 |7 |8 |
-----
|9 |10 | |11 |
-----
|13 |14 |15 |12 |
-----

-----
|1 |2 |3 |4 |
-----
|5 |6 |7 |8 |
-----
|9 |10 |11 | |
-----
|13 |14 |15 |12 |
-----

-----
|1 |2 |3 |4 |
-----
|5 |6 |7 |8 |
-----
|9 |10 |11 |12 |
-----
|13 |14 |15 | |
-----

Execution Time : 39 ms

Jumlah Matriks yang Dibangkitkan : 12

D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>
```

#### 4. Puzzle awal dari file teks accepted2.txt

```

C:\Windows\System32\cmd.exe
D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>java Main
Pilih Masukan :
[1] Acak
[2] File
>>> 2
Pilih File
Masukan Nama File:
>>> accepted2.txt

-----
|5 |1 |7 |3 |
|9 |2 |11|4 |
|13|6 |15|8 |
|  |10|14|12|
-----
Kurang(5) = 4
Kurang(1) = 0
Kurang(7) = 4
Kurang(3) = 1
Kurang(9) = 4
Kurang(2) = 0
Kurang(11) = 4
Kurang(4) = 0
Kurang(13) = 4
Kurang(6) = 0
Kurang(15) = 4
Kurang(8) = 0
Kurang() ubin kosong = 3
Kurang(10) = 0
Kurang(14) = 1
Kurang(12) = 0
Total Fungsi Kurang adalah 30
Solusi Ditemukan
Urutan Matriks :

-----
|5 |1 |7 |3 |
|9 |2 |11|4 |
|13|6 |15|8 |
|  |10|14|12|
-----

-----
|5 |1 |7 |3 |
|9 |2 |11|4 |
|  |6 |15|8 |
|13|10|14|12|
-----

-----
|5 |1 |7 |3 |
|  |2 |11|4 |
|9 |6 |15|8 |
|13|10|14|12|
-----

-----
|  |1 |7 |3 |
|5 |2 |11|4 |
|9 |6 |15|8 |
|13|10|14|12|
-----

```



```
Select C:\Windows\System32\cmd.exe

|1 |2 |7 |3 |
|5 |6 |11 |4 |
|9 |10 |8 |
|13 |14 |15 |12 |
-----

|1 |2 |7 |3 |
|5 |6 |4 |
|9 |10 |11 |8 |
|13 |14 |15 |12 |
-----

|1 |2 |3 | |
|5 |6 |7 |4 |
|9 |10 |11 |8 |
|13 |14 |15 |12 |
-----

|1 |2 |3 | |
|5 |6 |7 |4 |
|9 |10 |11 |8 |
|13 |14 |15 |12 |
-----

|5 |6 |7 |4 |
|9 |10 |11 |8 |
|13 |14 |15 |12 |
-----

|1 |2 |3 |4 |
|5 |6 |7 |
|9 |10 |11 |
|13 |14 |15 |12 |
-----

|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10 |11 |
|13 |14 |15 |12 |
-----

|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10 |11 |12 |
|13 |14 |15 |
-----

Execution Time : 109 ms

Dumlah Matriks yang Dibangkitkan : 47

D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>
```

## 5. Puzzle awal dari file teks accepted3.txt

```

C:\Windows\System32\cmd.exe
D:\ITB\Kuliah\Sem4\Strategi Algoritma\Tugas\Tucil3\Tucil3_13520162\bin>java Main
Pilih Masukan :
[1] Acak
[2] File
>>> 2
Pilih File
Masukan Nama File:
>>> accepted3.txt

-----
|2 |3 |7 |4 |
-----
|1 |5 | |10|
-----
|9 |6 |12|8 |
-----
|13|14|11|15|
-----
Kurang(2) = 1
Kurang(3) = 1
Kurang(7) = 4
Kurang(4) = 1
Kurang(1) = 0
Kurang(5) = 0
Kurang() ubin kosong = 9
Kurang(10) = 3
Kurang(9) = 2
Kurang(6) = 0
Kurang(12) = 2
Kurang(8) = 0
Kurang(13) = 1
Kurang(14) = 1
Kurang(11) = 0
Kurang(15) = 0
Total Fungsi Kurang adalah 26
Solusi Ditemukan
Urutan Matriks :

-----
|2 |3 |7 |4 |
-----
|1 |5 | |10|
-----
|9 |6 |12|8 |
-----
|13|14|11|15|
-----

-----
|2 |3 |7 |4 |
-----
|1 |5 |10|8 |
-----
|9 |6 |12| |
-----
|13|14|11|15|
-----

-----
|2 |3 |7 |4 |
-----
|1 |5 |10|8 |
-----
|9 |6 | |12|
-----
|13|14|11|15|
-----

```



The image shows two screenshots of a Windows command prompt window. The title bar reads "Select C:\Windows\System32\cmd.exe". The window contains a 4x4 grid of numbers, with the last cell in each row being empty. The numbers are arranged as follows:

2	3	7	4
1	5		8
9	6	10	12
13	14	11	15

The grid is repeated three times in the first screenshot. In the second screenshot, the grid is repeated four times, with the second row of the second grid containing the number 5 in the first column and an empty cell in the second column.

```

Select C:\Windows\System32\cmd.exe
|5 |6 |7 |8 |
|9 | |10 |12 |
|13 |14 |11 |15 |
-----
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10 | |12 |
|13 |14 |11 |15 |
-----
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10 |11 |12 |
|13 |14 | |15 |
-----
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10 |11 |12 |
|13 |14 |15 | |
-----
Execution Time : 93 ms
Jumlah Matriks yang Dibangkitkan : 408

```

### C. Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	
2. Program berhasil running	✓	
3. Program dapat menerima input dan menuliskan output.	✓	
4. Luaran sudah benar untuk semua data uji	✓	
5. Bonus dibuat		✓

### D. Kode program dalam Java

#### 1. Matriks.java

```

import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Objects;

```

```

class Matriks {
    private int m[][]; // status puzzle
    private String prev; // aksi sebelumnya
    private int c; // cost
    private Matriks parent; // simpul parent

    // constructor untuk puzzle acak
    public Matriks() {
        this.m = new int[4][4];
        this.prev = "none";
        this.parent = null;
        ArrayList<Integer> a = new ArrayList<Integer>();
        for (int i = 0; i < 16 ; i++) {
            a.add(i+1);
        }
        int b = new Random().nextInt(16);
        int k = 0;
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                this.m[i][j] = a.get(b);
                a.remove(b);
                k++;
                if (b > 1) {
                    b = new Random().nextInt(16 - k);
                } else {
                    b = 0;
                }
            }
        }
        this.c = this.g() + this.f();
    }

    // constructor untuk puzzle dari file teks
    public Matriks(File file) {
        try {
            this.m = new int[4][4];
            this.prev = "none";
            this.parent = null;
            Scanner scan = new Scanner(file);
            int i = 0;
            int j = 0;
            while (scan.hasNextLine()) {
                String line = scan.nextLine();
                String[] numbers = line.split(" ");
                for (String number : numbers) {
                    if (j > 3) {
                        j = 0;
                        i++;
                    }
                    this.m[i][j] = Integer.parseInt(number);
                    j++;
                }
            }
            scan.close();
            this.c = this.g() + this.f();
        } catch (FileNotFoundException e) {

```

```

        System.out.println("File Not Found");
    }
}

// copy constructor
public Matriks(Matriks M) {
    this.m = new int[4][4];
    this.prev = "none";
    this.parent = M.parent;
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            this.m[i][j] = M.m[i][j];
        }
    }
    this.c = this.g() + this.f();
}

// membangkitkan simpul baru dengan aksi atas, kanan, bawah, kiri
public Matriks Move(String dir) {
    Matriks Mat = new Matriks(this);
    Mat.parent = this;
    int[] point = Mat.findBlank();
    if (dir.equals("up") && point[0] != 0 && Mat.prev != "up") {
        Mat.m[point[0]][point[1]] = Mat.m[point[0] - 1][point[1]];
        Mat.m[point[0] - 1][point[1]] = 16;
        Mat.prev = "up";
        Mat.c = Mat.g() + Mat.f();
        return Mat;
    }
    else if (dir.equals("right") && point[1] != 3 && Mat.prev != "right")
    {
        Mat.m[point[0]][point[1]] = Mat.m[point[0]][point[1] + 1];
        Mat.m[point[0]][point[1] + 1] = 16;
        Mat.prev = "right";
        Mat.c = Mat.g() + Mat.f();
        return Mat;
    }
    else if (dir.equals("down") && point[0] != 3 && Mat.prev != "down") {
        Mat.m[point[0]][point[1]] = Mat.m[point[0] + 1][point[1]];
        Mat.m[point[0] + 1][point[1]] = 16;
        Mat.prev = "down";
        Mat.c = Mat.g() + Mat.f();
        return Mat;
    }
    else if (dir.equals("left") && point[1] != 0 && Mat.prev != "left") {
        Mat.m[point[0]][point[1]] = Mat.m[point[0]][point[1] - 1];
        Mat.m[point[0]][point[1] - 1] = 16;
        Mat.prev = "left";
        Mat.c = Mat.g() + Mat.f();
        return Mat;
    }
    else {
        return null;
    }
}

// mencari posisi ubin kosong

```

```

public int[] findBlank() {
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            if (this.m[i][j] == 16) {
                int point[] = new int[2];
                point[0] = i;
                point[1] = j;
                return point;
            }
        }
    }
    return null;
}

// menampilkan matriks puzzle
public void printMatriks() {
    for (int i = 0; i < 4; i++) {
        System.out.printf("%n-----%n");
        System.out.printf("|");
        for (int j = 0; j < 4; j++) {
            if (this.m[i][j] == 16) {
                System.out.printf("  ");
            }
            else if (this.m[i][j] < 10) {
                System.out.printf("%d ", this.m[i][j]);
            } else {
                System.out.printf("%d ", this.m[i][j]);
            }
        }
        System.out.printf("|");
    }
    System.out.printf("%n-----%n");
}

// menentukan apakah puzzle dapt diselesaikan dengan fungsi sigma(i=1,16)
kurang(i) + X
// fungsi kurang() untuk ubin kosong juga ikut dihitung
public Boolean reachableGoal() {
    int kurang = 0;
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            kurang += this.kurang(i, j);
        }
    }
    int[] point = this.findBlank();
    if ((point[0] + point[1]) % 2 == 1) {
        kurang++;
    }
    System.out.printf("Total Fungsi Kurang adalah %d\n", kurang);
    return kurang % 2 == 0;
}

// fungsi kurang()

```

```

public int kurang(int x, int y) {
    int kurang = 0;
    for (int i = y; i < 4; i++) {
        if (this.m[x][y] > this.m[x][i] && this.m[x][i] != 16) {
            kurang++;
        }
    }
    for (int k = x+1; k < 4; k++) {
        for (int l = 0; l < 4; l++) {
            if (this.m[x][y] > this.m[k][l] && this.m[k][l] != 16) {
                kurang++;
            }
        }
    }
    if (this.m[x][y] != 16) {
        System.out.printf("Kurang(%d) = %d\n", this.m[x][y], kurang);
    } else {
        System.out.printf("Kurang() ubin kosong = %d\n", kurang);
    }
    return kurang;
}

// jumlah ubin tidak kosong yang tidak terdapat pada susunan akhir
public int g() {
    int g = 0;
    int e = 1;
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            if (this.m[i][j] != e && this.m[i][j] != 16) {
                g++;
            }
            e++;
        }
    }
    return g;
}

// jarak simpul ini ke simpul awal
public int f() {
    int f = 0;
    Matriks N = this.getParent();
    while (!Objects.isNull(N)) {
        f++;
        N = N.getParent();
    }
    return f;
}

// menentukan apakah puzzle telah mencapai status akhir
public Boolean goalState() {
    int e = 1;
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            if (this.m[i][j] != e) {
                return false;
            }
        }
    }
}

```

```

        e++;
    }
    return true;
}

// getter simpul parent
public Matriks getParent() {
    return this.parent;
}

// getter cost
public int getC() {
    return this.c;
}
}

```

## 2. Main.java

```

import java.util.ArrayList;
import java.util.Objects;
import java.util.*;
import java.io.File;

public class Main {
    public static void main(String[] args) {
        // memilih puzzle awal
        System.out.println("Pilih Masukan :");
        System.out.println("[1] Acak");
        System.out.println("[2] File");
        Scanner scaninput = new Scanner(System.in);
        System.out.print(">>> ");
        int in = scaninput.nextInt();
        Matriks R;

        // inisiasi simpul awal R
        Boolean input = false;
        do {
            if (in == 1) {
                System.out.println("Pilih Acak");
                R = new Matriks();
                input = true;
            }
            else if (in == 2) {
                System.out.println("Pilih File");
                System.out.println("Masukan Nama File:");
                System.out.print(">>> ");
                String filename = scaninput.next();
                String F = ("..\test\\" + filename);
                File file = new File(F);
                while (!file.exists()) {
                    System.out.println("File tidak ada. Ulangi !");
                    System.out.println("Masukan Nama File:");
                    System.out.print(">>> ");
                }
            }
        } while (!input);
    }
}

```

```

        filename = scaninput.next();
        F = ("..\test\\" + filename);
        file = new File(F);
    }
    R = new Matriks(file);
    input = true;
}
else {
    R = new Matriks();
    System.out.println("Masukan Salah, Ulangi !");
    System.out.println("Pilih Masukan :");
    System.out.println("[1] Acak");
    System.out.println("[2] File");
    System.out.print(">>> ");
    in = scaninput.nextInt();
}
} while (!input);

scaninput.close();

// memulai waktu
long start = System.currentTimeMillis();
R.printMatriks();

// menentukan apakah puzzle dapat diselesaikan
if (!R.reachableGoal()) {
    System.out.println("Persoalan Tidak Dapat Diselesaikan");
    System.exit(0);
}

// menentukan apakah puzzle awal sudah mencapai status akhir
if (R.goalState()) {
    System.out.println("Solusi Ditemukan");
    System.out.println("Urutan Matriks :");
    R.printMatriks();
}

// list untuk simpul hidup
ArrayList<Matriks> Hidup = new ArrayList<Matriks>();

// list untuk simpul solusi
ArrayList<Matriks> Solusi = new ArrayList<Matriks>();

// memasukan puzzle awal ke list simpul hidup
Hidup.add(R);

Boolean found = false;

// inisiasi jumlah d=simpul yang telah dibangkitkan
int jumlah = 1;

// membangkitkan simpul-simpul valid dari simpul R
while (!found) {
    // mengeluarkan simpul R dari list simpul hidup
    Hidup.remove(R);

```



```

        // membangkitkan simpul untuk aksi atas
        Matriks MU = R.Move("up");
        if (!Objects.isNull(MU)) {
            jumlah++;
            Hidup.add(MU);
        }

        // membangkitkan simpul untuk aksi kanan
        Matriks MR = R.Move("right");
        if (!Objects.isNull(MR)) {
            jumlah++;
            Hidup.add(MR);
        }

        // membangkitkan simpul untuk aksi bawah
        Matriks MD = R.Move("down");
        if (!Objects.isNull(MD)) {
            jumlah++;
            Hidup.add(MD);
        }

        // membangkitkan simpul untuk aksi kiri
        Matriks ML = R.Move("left");
        if (!Objects.isNull(ML)) {
            jumlah++;
            Hidup.add(ML);
        }

        // mencari simpul yang memiliki cost terkecil dan menjadikanya
        simpul R baru
        R = Hidup.get(0);
        for (int i = 1; i < Hidup.size(); i++) {
            if (Hidup.get(i).getC() < R.getC()) {
                R = Hidup.get(i);
            }
        }

        // menentukan jika simpul R sudah mencapai status akhir
        if (R.goalState()) {
            found = true;
        }
    }

    // memasukan simpul awal sampai simpul akhir kedalam list simpul
    solusi
    Solusi.add(R);
    R = R.getParent();
    while (!Objects.isNull(R)) {
        Solusi.add(R);
        R = R.getParent();
    }

    // menampilkan urutan matriks dari posisi awal ke posisi akhir
    Collections.reverse(Solusi);
    System.out.println("Solusi Ditemukan");

```

```

        System.out.println("Urutan Matriks :");
        for (Matriks m : Solusi) {
            m.printMatriks();
        }

        // mengakhiri waktu
        long stop = System.currentTimeMillis();

        // menampilkan waktu eksekusi
        System.out.println("Execution Time : " + (stop - start) + " ms\n");

        // menampilkan jumlah simpul yang dibangkitkan
        System.out.printf("Jumlah Matriks yang Dibangkitkan : %d\n", jumlah);
    }
}

```

### E. Contoh Instansiasi 5 Buah Persoalan 15-puzzle

Program dapat menerima file teks sebagai posisi awal 15-puzzle. File teks terdiri dari 4 buah baris. Setiap baris terdiri dari 4 buah angka yang dipisahkan oleh spasi. Enam belas angka ini akan menjadi posisi awal puzzle. Angka 16 merepresentasikan ubin kosong.

#### 1. rejected1.txt

1	3	4	15
2	16	5	12
7	6	11	14
8	9	10	13

#### 2. rejected2.txt

3	9	1	15
14	11	4	6
13	16	10	12
2	7	8	5

#### 3. accepted1.txt

1	2	3	4
5	6	16	8
9	10	7	11
13	14	15	12

4. accepted2.txt

5 1 7 3
9 2 11 4
13 6 15 8
16 10 14 12

5. accepted3.txt

2 3 7 4
1 5 16 10
9 6 12 8
13 14 11 15

F. Alamat Github

[https://github.com/DaffaRomyz/Tucil3\\_13520162](https://github.com/DaffaRomyz/Tucil3_13520162)