LAPORAN TUGAS KECIL 3

IF2211 Strategi Algoritma

Penyelesaian Persoalan 15-Puzzle dengan Algoritma *Branch and Bound*

Disusun Oleh :

Nama : Johannes Winson Sukiatmodjo

NIM : 13520123

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2022

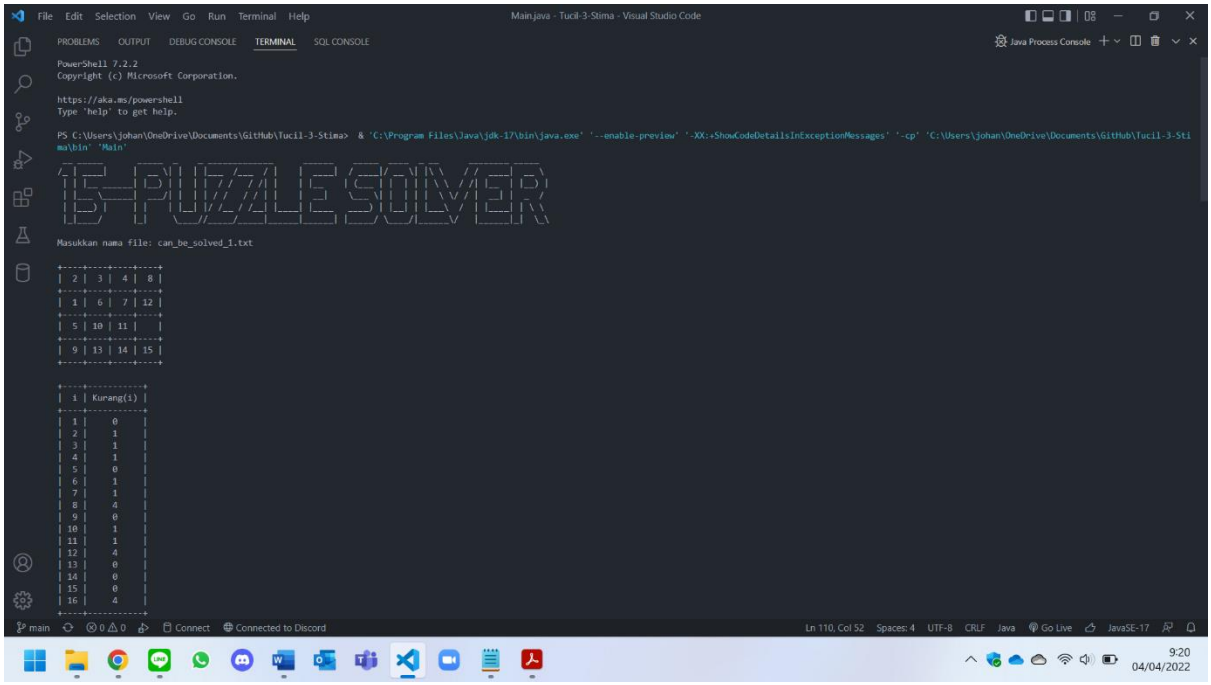
Cara Kerja Program

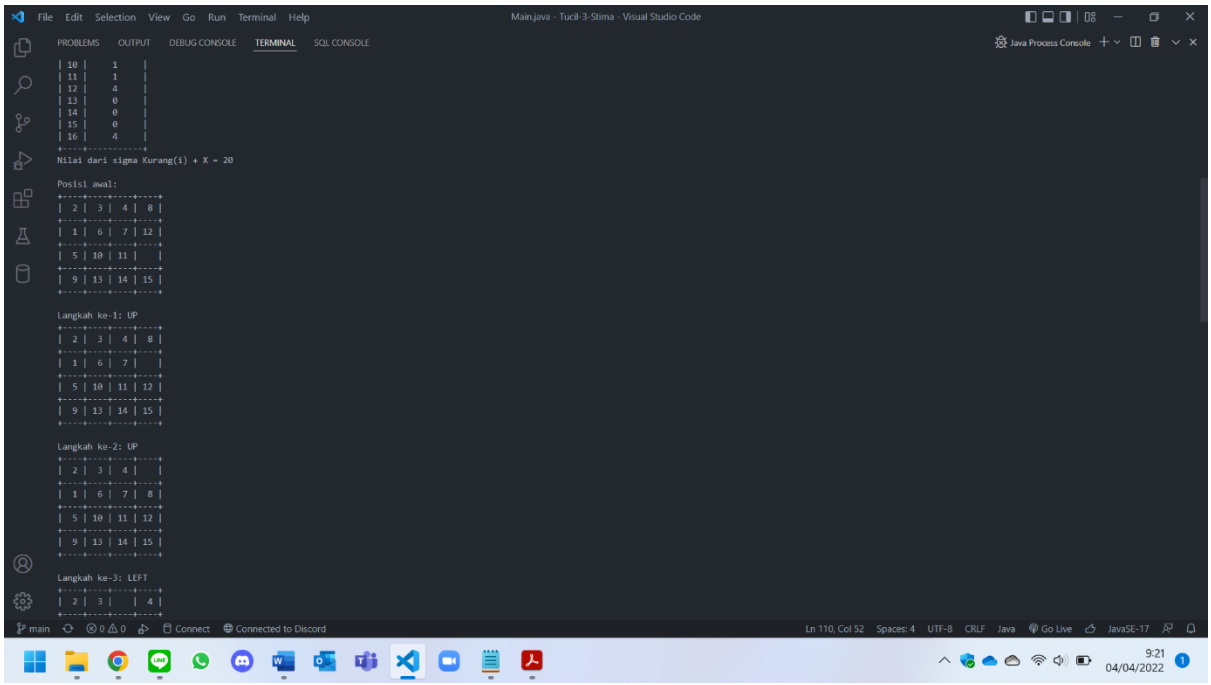
Berikut merupakan cara kerja program Branch and Bound ini dalam menyelesaikan persoalan 15-Puzzle.

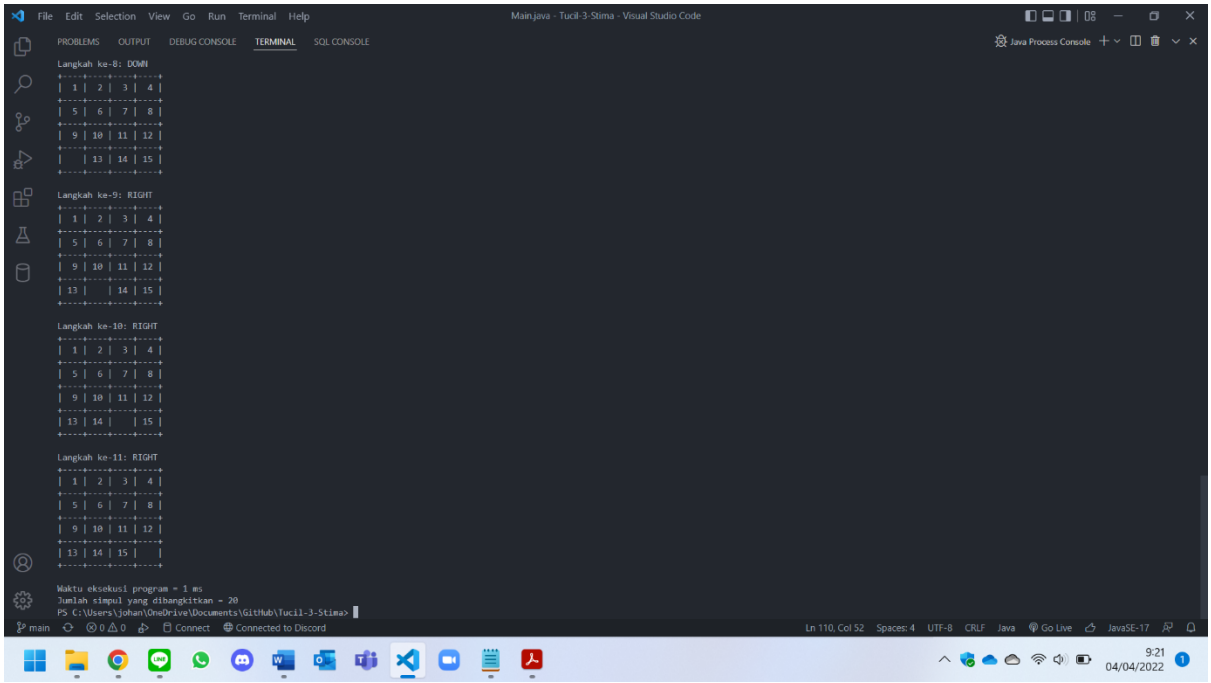
1. Memasukkan matriks posisi awal ke dalam priority queue
2. Melakukan perulangan terus-menerus selama priority queue tidak kosong
3. Mengambil node dengan prioritas paling tinggi untuk dianalisis kemudian, prioritas didasarkan pada nilai cost paling kecil
4. Menambahkan path untuk menuju node tersebut
5. Jika node tersebut sesuai dengan matriks solusi, perulangan akan berhenti
6. Jika tidak, akan dibangkitkan simpul-simpul lain dengan syarat tidak kembali ke posisi awal, lalu memasukkan node tersebut ke dalam priority queue
7. Perulangan berakhir pada saat node tersebut merupakan sebuah matriks solusi atau priority queue telah kosong

Screenshot Input-Output Program

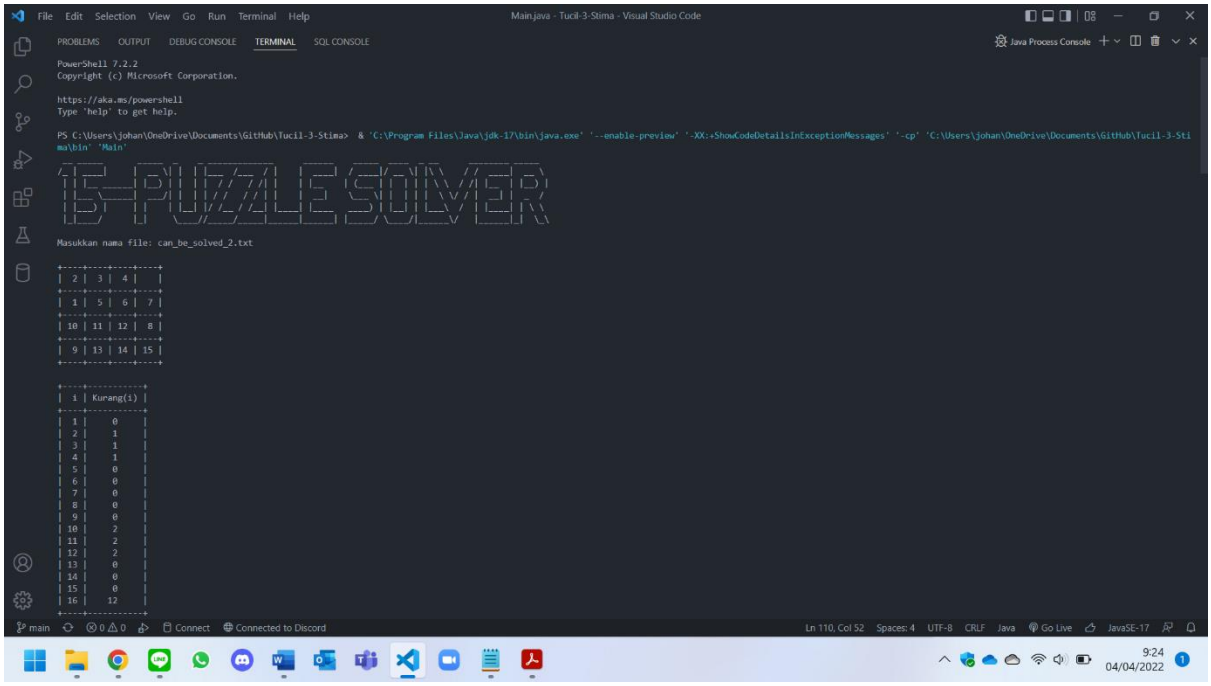
1. can_be_solved_1.txt

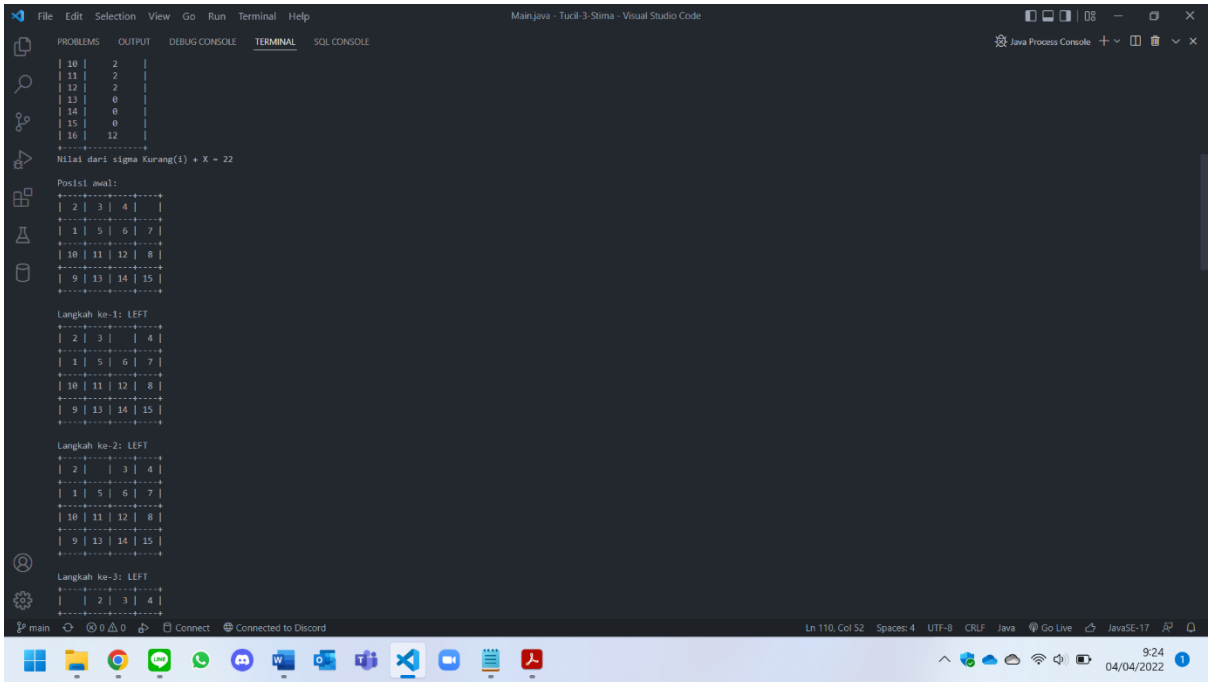


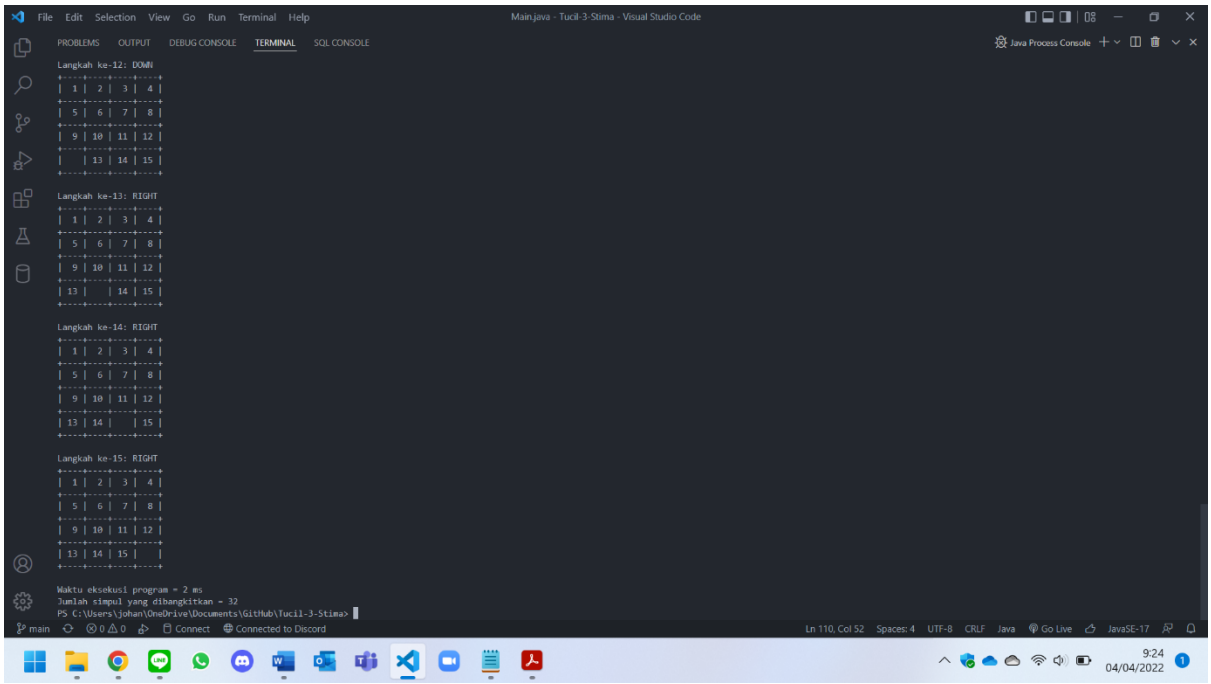




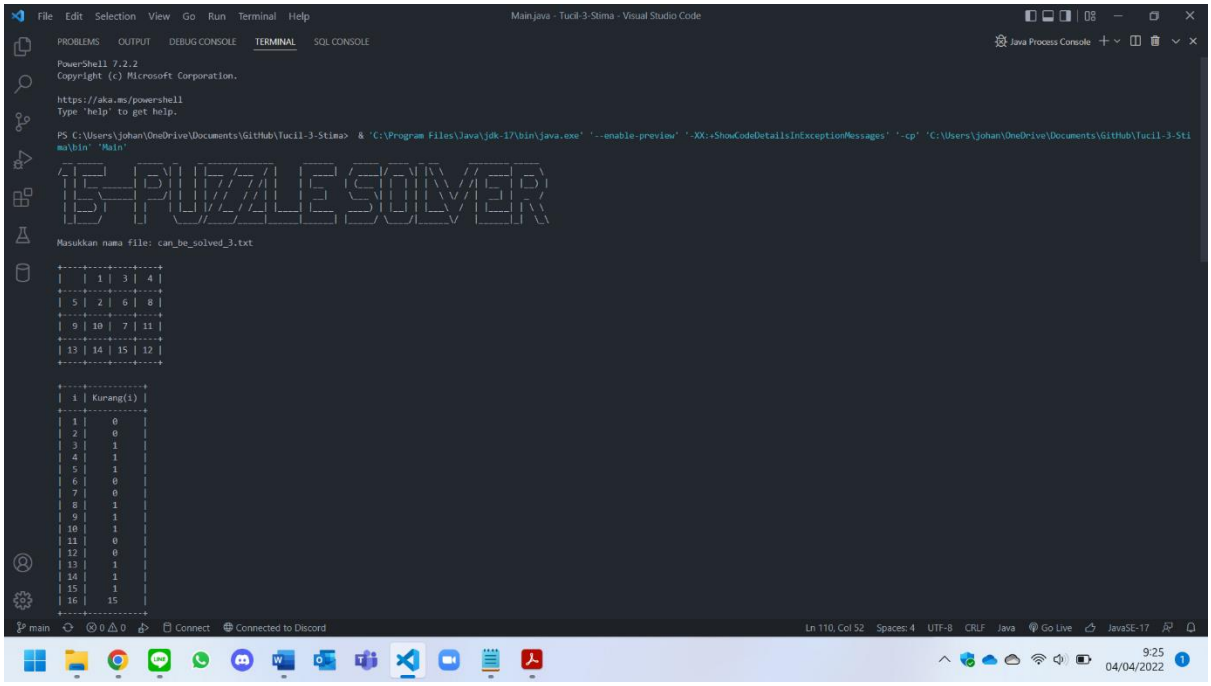
2. can_be_solved_2.txt

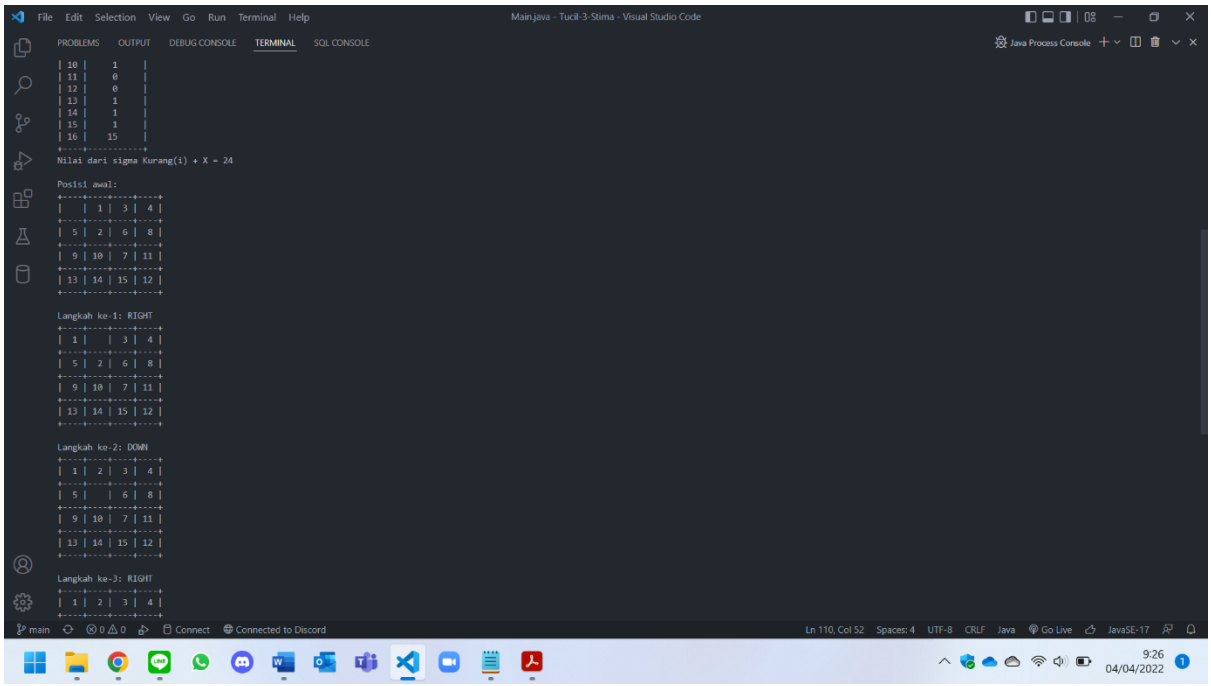


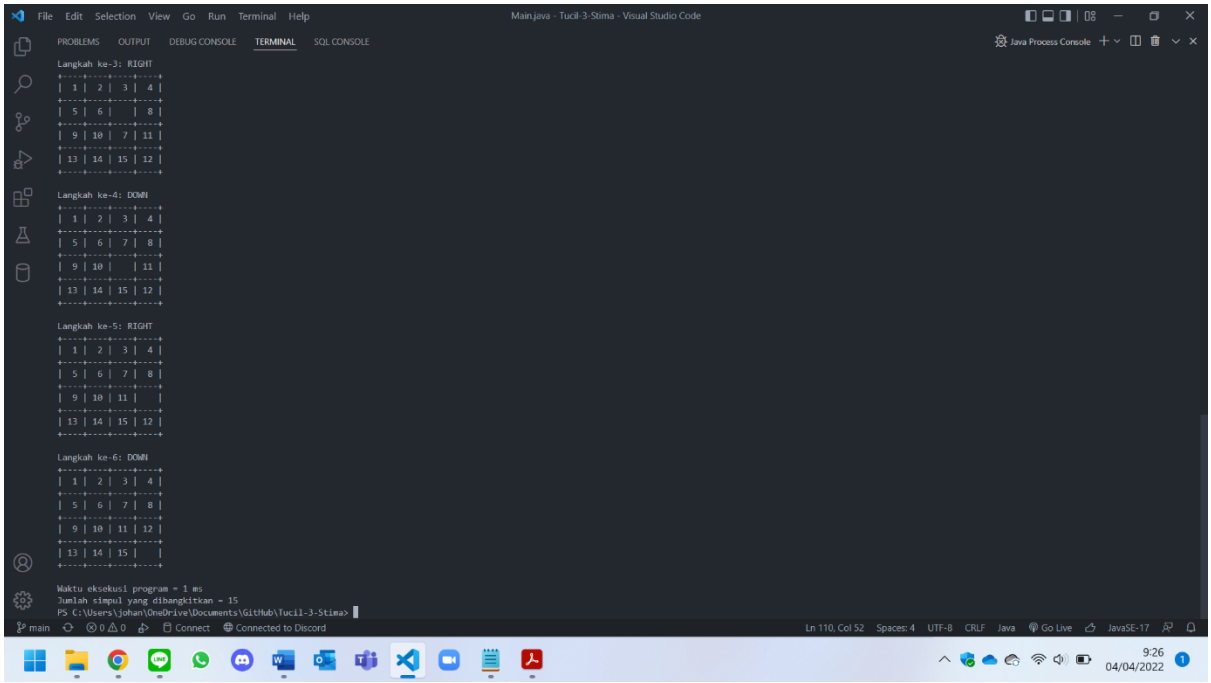

File Edit Selection View Go Run Terminal Help
Main.java - Tucil-3-Stima - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
Java Process Console
10	2
11	2
12	2
13	0
14	0
15	0
16	12
Nilai dari sigma Kurang(i) + X = 22
Posisi awal:
2	3	4	
1	5	6	7
10	11	12	8
9	13	14	15
Langkah ke-1: LEFT
2	3		4
1	5	6	7
10	11	12	8
9	13	14	15
Langkah ke-2: LEFT
2		3	4
1	5	6	7
10	11	12	8
9	13	14	15
Langkah ke-3: LEFT
	2	3	4
main Connect Connected to Discord
Ln 110, Col 52 Spaces: 4 UTF-8 CRLF Java Go Live JavaSE-17
9:24
04/04/2022


File Edit Selection View Go Run Terminal Help
Main.java - Tucil-3-Stima - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
Java Process Console
Langkah ke-12: DOWN
1	2	3	4
5	6	7	8
9	10	11	12
	13	14	15
Langkah ke-13: RIGHT
1	2	3	4
5	6	7	8
9	10	11	12
13		14	15
Langkah ke-14: RIGHT
1	2	3	4
5	6	7	8
9	10	11	12
13	14		15
Langkah ke-15: RIGHT
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	
Waktu eksekusi program = 2 ms
Jumlah simpul yang dibangkitkan = 32
PS C:\Users\johan\OneDrive\Documents\GitHub\Tucil-3-Stima>
main Connect Connected to Discord
Ln 110, Col 52 Spaces: 4 UTF-8 CRLF Java Go Live JavaSE-17
9:24
04/04/2022

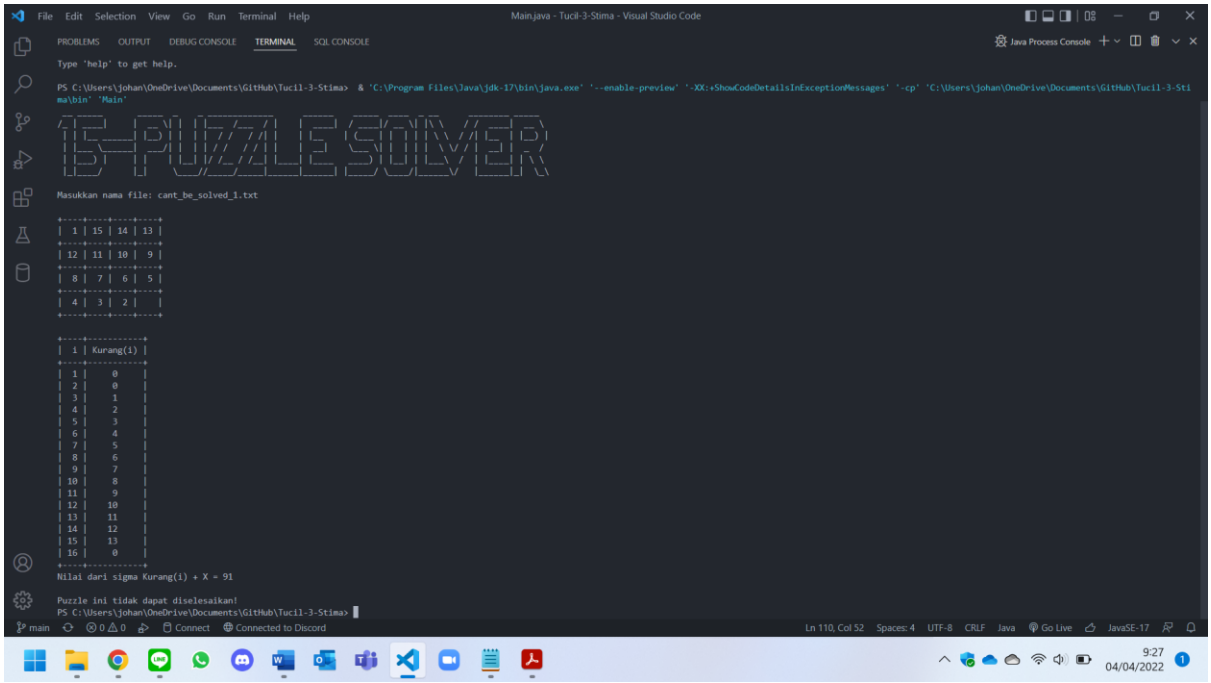
3. can_be_solved_3.txt



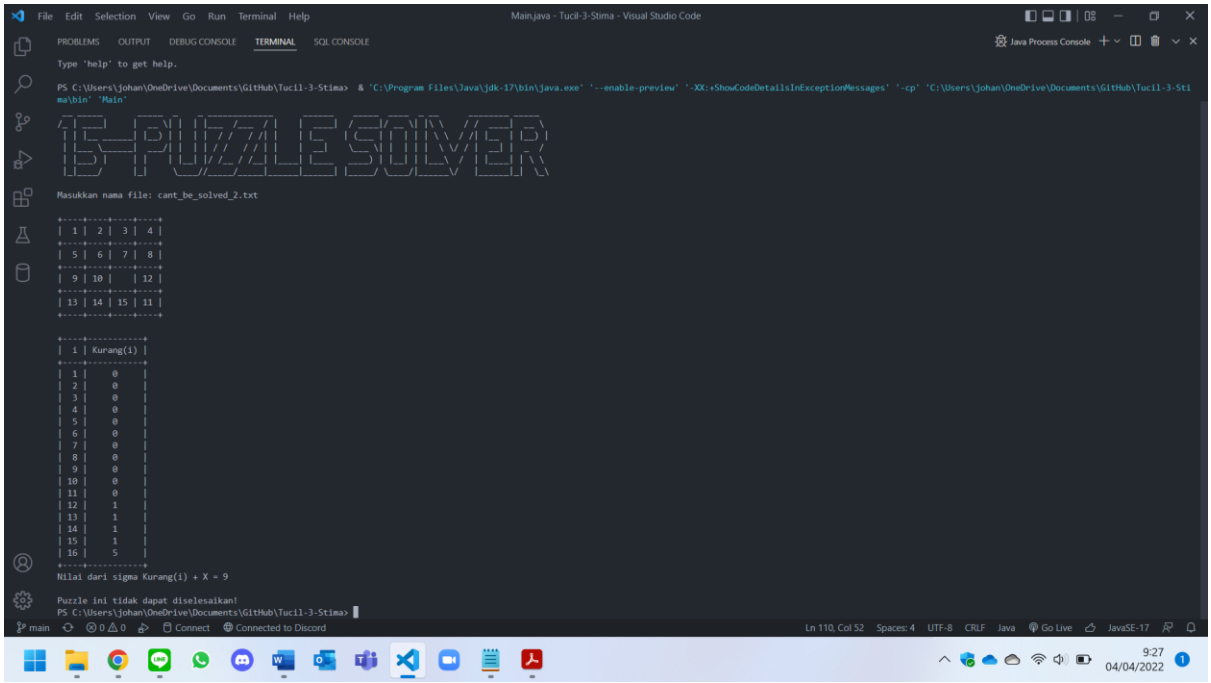




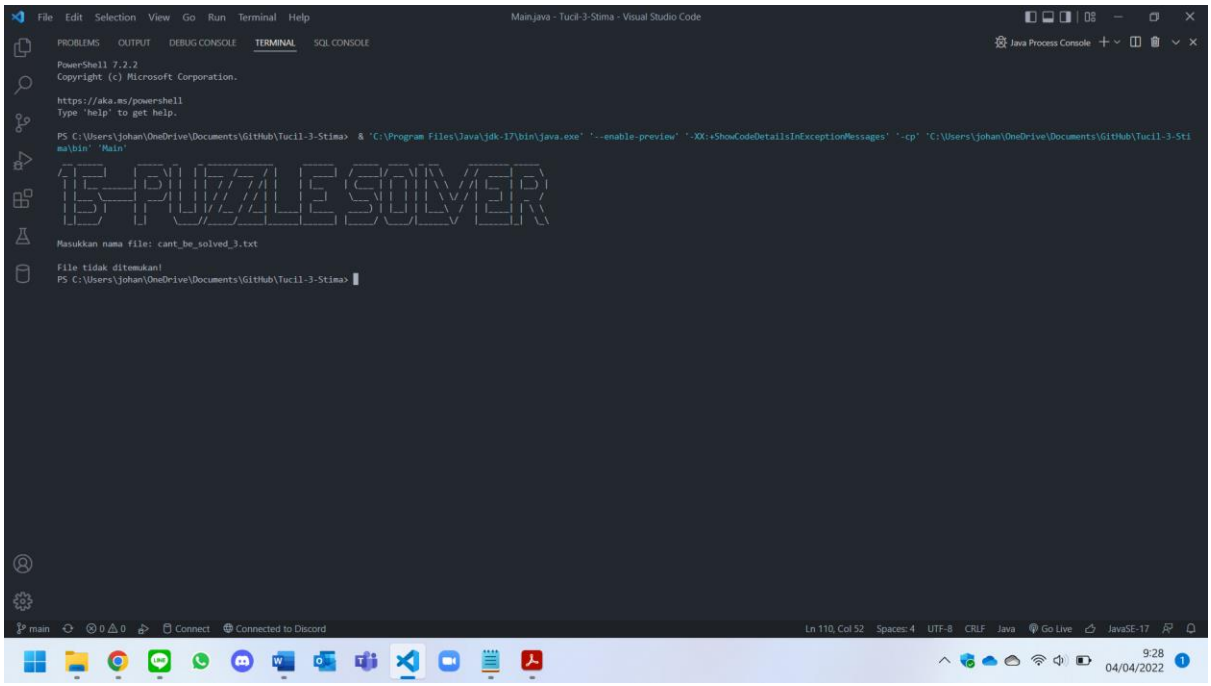
4. cant_be_solved_1.txt



5. cant_be_solved_2.txt



6. Kasus untuk file tidak ditemukan



Kode Program

1. FifteenPuzzle.java

```java
import java.util.ArrayList;
import java.util.List;

public class FifteenPuzzle implements Comparable<FifteenPuzzle> {
    private int row;
    private int col;
    private int[][] matrix;
    private List<String> path;

    // default constructor
    public FifteenPuzzle() {
        this.row = 4;
        this.col = 4;
        this.matrix = new int[this.row][this.col];
        this.path = new ArrayList<>();
        this.path.add("START");
    }

    // solution constructor
    public FifteenPuzzle(String solution) {
        int number = 0;
        this.row = 4;
        this.col = 4;
        this.matrix = new int[this.row][this.col];
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                number++;
                this.matrix[i][j] = number;
            }
        }
        this.path = new ArrayList<>();
    }

    // copy constructor
    public FifteenPuzzle(FifteenPuzzle fifteen_puzzle) {
        this.row = 4;
        this.col = 4;
        this.matrix = new int[this.row][this.col];
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                this.matrix[i][j] = fifteen_puzzle.matrix[i][j];
            }
        }
```

```
        this.path = new ArrayList<>();
        for (String string : fifteen_puzzle.path) {
            this.path.add(string);
        }
    }

    // menentukan PriorityQueue berdasarkan nilai cost suatu matriks
    public int compareTo(FifteenPuzzle fifteen_puzzle) {
        if (this.cost() > fifteen_puzzle.cost()) {
            return 1;
        } else if (this.cost() == fifteen_puzzle.cost()) {
            return 0;
        } else {
            return -1;
        }
    }

    // mendapatkan path untuk menuju suatu matriks
    public List<String> getPath() {
        return this.path;
    }

    // mengubah nilai elemen suatu matriks
    public void setElement(int row, int col, int number) {
        this.matrix[row][col] = number;
    }

    // mengecek kesamaan antara dua buah matriks
    public boolean isSame(FifteenPuzzle fifteen_puzzle) {
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                if (this.matrix[i][j] != fifteen_puzzle.matrix[i][j]) {
                    return false;
                }
            }
        }
        return true;
    }

    // menentukan posisi dua buah angka, true jika posisi(big) < posisi(small)
    public boolean posisi(int big, int small) {
        int decision = 0;
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                if (this.matrix[i][j] == big) {
                    decision = big;
                } else if (this.matrix[i][j] == small) {
                    decision = small;
```

```
                }
            }
        }
        if (decision == small) {
            return true;
        } else {
            return false;
        }
    }

    // menghitung banyaknya number > i dan posisi(number) < posisi(i)
    public int kurang(int number) {
        int count = 0;
        for (int i = 1; i < number; i++) {
            if (this.posisi(number, i)) {
                count++;
            }
        }
        return count;
    }

    // menentukan apakah posisi kosong berada pada arsiran
    public boolean isKosong() {
        for (int i = 0; i < this.row; i++) {
            if (i % 2 == 0) {
                if (this.matrix[i][1] == 16 || this.matrix[i][3] == 16) {
                    return true;
                }
            } else {
                if (this.matrix[i][0] == 16 || this.matrix[i][2] == 16) {
                    return true;
                }
            }
        }
        return false;
    }

    // menghitung nilai sigma Kurang(i) + X
    public int sigmaKurang() {
        int sigma = 0;
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                sigma += this.kurang(this.matrix[i][j]);
            }
        }
        if (this.isKosong()) {
            sigma += 1;
        }
```

```
        return sigma;
    }

    // menghitung nilai cost suatu matriks
    public int cost() {
        int number = 0;
        int f = 1;
        int g = 0;
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                number++;
                if (number == 16) {
                    break;
                }
                if (this.matrix[i][j] != number) {
                    g += 1;
                }
            }
        }
        int c = f + g;
        return c;
    }

    // mengecek apakah bisa melakukan UP
    public boolean checkUP() {
        for (int i = 0; i < this.col; i++) {
            if (this.matrix[0][i] == 16) {
                return false;
            }
        }
        return true;
    }

    // melakukan UP pada this.matrix
    public void UP() {
        if (this.checkUP()) {
            boolean flag = false;
            for (int i = 0; i < this.row; i++) {
                for (int j = 0; j < this.col; j++) {
                    if (this.matrix[i][j] == 16) {
                        this.matrix[i][j] = this.matrix[i-1][j];
                        this.matrix[i-1][j] = 16;
                        flag = true;
                    }
                    if (flag) {
                        break;
                    }
                }
```

```
                if (flag) {
                    break;
                }
            }
        }
    }

    // mengecek apakah this.matrix merupakan UP dari fifteen_puzzle.matrix
    public boolean isUP(FifteenPuzzle fifteen_puzzle) {
        FifteenPuzzle check = new FifteenPuzzle(fifteen_puzzle);
        check.UP();
        return this.isSame(check);
    }

    // mengecek apakah bisa melakukan RIGHT
    public boolean checkRIGHT() {
        for (int i = 0; i < this.row; i++) {
            if (this.matrix[i][3] == 16) {
                return false;
            }
        }
        return true;
    }

    // melakukan RIGHT pada this.matrix
    public void RIGHT() {
        if (this.checkRIGHT()) {
            boolean flag = false;
            for (int i = 0; i < this.row; i++) {
                for (int j = 0; j < this.col; j++) {
                    if (this.matrix[i][j] == 16) {
                        this.matrix[i][j] = this.matrix[i][j+1];
                        this.matrix[i][j+1] = 16;
                        flag = true;
                    }
                    if (flag) {
                        break;
                    }
                }
                if (flag) {
                    break;
                }
            }
        }
    }

    // mengecek apakah this.matrix merupakan RIGHT dari fifteen_puzzle.matrix
    public boolean isRIGHT(FifteenPuzzle fifteen_puzzle) {
```

```
        FifteenPuzzle check = new FifteenPuzzle(fifteen_puzzle);
        check.RIGHT();
        return this.isSame(check);
    }

    // mengecek apakah bisa melakukan DOWN
    public boolean checkDOWN() {
        for (int i = 0; i < this.col; i++) {
            if (this.matrix[3][i] == 16) {
                return false;
            }
        }
        return true;
    }

    // melakukan DOWN pada this.matrix
    public void DOWN() {
        if (this.checkDOWN()) {
            boolean flag = false;
            for (int i = 0; i < this.row; i++) {
                for (int j = 0; j < this.col; j++) {
                    if (this.matrix[i][j] == 16) {
                        this.matrix[i][j] = this.matrix[i+1][j];
                        this.matrix[i+1][j] = 16;
                        flag = true;
                    }
                    if (flag) {
                        break;
                    }
                }
                if (flag) {
                    break;
                }
            }
        }
    }

    // mengecek apakah this.matrix merupakan DOWN dari fifteen_puzzle.matrix
    public boolean isDOWN(FifteenPuzzle fifteen_puzzle) {
        FifteenPuzzle check = new FifteenPuzzle(fifteen_puzzle);
        check.DOWN();
        return this.isSame(check);
    }

    // mengecek apakah bisa melakukan LEFT
    public boolean checkLEFT() {
        for (int i = 0; i < this.row; i++) {
            if (this.matrix[i][0] == 16) {
```

```
                return false;
            }
        }
        return true;
    }

    // melakukan LEFT pada this.matrix
    public void LEFT() {
        if (this.checkLEFT()) {
            boolean flag = false;
            for (int i = 0; i < this.row; i++) {
                for (int j = 0; j < this.col; j++) {
                    if (this.matrix[i][j] == 16) {
                        this.matrix[i][j] = this.matrix[i][j-1];
                        this.matrix[i][j-1] = 16;
                        flag = true;
                    }
                    if (flag) {
                        break;
                    }
                }
                if (flag) {
                    break;
                }
            }
        }
    }

    // mengecek apakah this.matrix merupakan LEFT dari fifteen_puzzle.matrix
    public boolean isLEFT(FifteenPuzzle fifteen_puzzle) {
        FifteenPuzzle check = new FifteenPuzzle(fifteen_puzzle);
        check.LEFT();
        return this.isSame(check);
    }

    // menambahkan path untuk menuju suatu matriks
    public void addToPath(List<String> list, FifteenPuzzle fifteen_puzzle) {
        if (this.isUP(fifteen_puzzle)) {
            list.add("UP");
        } else if (this.isDOWN(fifteen_puzzle)) {
            list.add("DOWN");
        } else if (this.isLEFT(fifteen_puzzle)) {
            list.add("LEFT");
        } else if (this.isRIGHT(fifteen_puzzle)) {
            list.add("RIGHT");
        }
    }
```

```
    // menampilkan suatu matriks ke layar
    public void showMatrix() {
        System.out.println("+----+----+----+----+");
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                if (this.matrix[i][j] >= 1 && this.matrix[i][j] <= 9) {
                    System.out.printf("|  %d ", this.matrix[i][j]);
                } else if (this.matrix[i][j] >= 10 && this.matrix[i][j] <= 15)
{
                    System.out.printf("| %d ", this.matrix[i][j]);
                } else {
                    System.out.print("|    ");
                }
            }
            System.out.println("|");
            System.out.println("+----+----+----+----+");
        }
    }

    // menampilkan nilai dari fungsi Kurang(i)
    public void showKurang() {
        System.out.println("+----+-----------+");
        System.out.println("|  i | Kurang(i) |");
        System.out.println("+----+-----------+");
        int number = 0;
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.col; j++) {
                number++;
                if (number >= 1 && number <= 9 && this.kurang(number) >= 0 &&
this.kurang(number) <= 9) {
                    System.out.printf("|  %d |     %d     |%n", number,
this.kurang(number));
                } else if (number >= 1 && number <= 9 && this.kurang(number)
>= 10) {
                    System.out.printf("|  %d |    %d     |%n", number,
this.kurang(number));
                } else if (number >= 10 && this.kurang(number) >= 0 &&
this.kurang(number) <= 9) {
                    System.out.printf("| %d |     %d     |%n", number,
this.kurang(number));
                } else {
                    System.out.printf("| %d |    %d     |%n", number,
this.kurang(number));
                }
            }
        }
        System.out.println("+----+-----------+");
    }
```

```
    // melaksanakan perintah sesuai command
    public void todo(String command) {
        if (command.equals("UP")) {
            this.UP();
        } else if (command.equals("RIGHT")) {
            this.RIGHT();
        } else if (command.equals("DOWN")) {
            this.DOWN();
        } else if (command.equals("LEFT")) {
            this.LEFT();
        }
    }
}
```

2. Main.java

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.List;
import java.util.PriorityQueue;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        System.out.println(" __ _____       _____  _    _
_____________      ______    _____  ____  _ __      ________ _____       ");
        System.out.println("/_ | ____|     |  __ \\| |  | |__  /__  /
|    |  ____|  / ____|/ __ \\| |\\ \\    / /  ____|  __ \\ ");
        System.out.println(" | | |__ ______| |__) | |  | |  / /   / /| |    |
|__    | (___ | |  | | | \\ \\  / /| |__  | |__) |   ");
        System.out.println(" | |___ \\______|  ___/| |  | | / /   / / |
|    |  __|    \\___ \\| |  | | |  \\ \\/ / |  __| |  _  / ");
        System.out.println(" | |___) |     | |    | |__| |/ /__ / /__| |____|
|____   ____) | |__| | |___\\  /  | |____| | \\ \\   ");
        System.out.println("
|_|____/      |_|     \\____//_____/_____|______|______| |_____/
\\____/|______\\/   |______|_|  \\_\\");

        FifteenPuzzle matrix = new FifteenPuzzle();
        int row = 0;
        int col = 0;

        try {
            // input file
            System.out.print("\nMasukkan nama file: ");
            Scanner scanFile = new Scanner(System.in);
```

```
            String file = scanFile.nextLine();
            scanFile.close();

            // baca file
            File myFile = new File("test/" + file);
            Scanner myReader = new Scanner(myFile);
            while (myReader.hasNextInt()) {
                int data = myReader.nextInt();
                matrix.setElement(row, col, data);
                col++;
                if (col == 4) {
                    row += 1;
                    col = 0;
                }
            }
            myReader.close();

            // menampilkan matriks posisi awal
            System.out.println();
            matrix.showMatrix();

            // menampilkan nilai dari fungsi Kurang(i)
            System.out.println();
            matrix.showKurang();

            // menampilkan nilai dari sigma Kurang(i) + X
            System.out.printf("Nilai dari sigma Kurang(i) + X = %d%n%n",
matrix.sigmaKurang());

            if (matrix.sigmaKurang() % 2 == 0) {
                // waktu dimulai
                long startTime = System.currentTimeMillis();

                FifteenPuzzle solution = new FifteenPuzzle("solution");
                List<String> solutionPath = new ArrayList<>();
                int totalNode = 0;

                // memasukkan matriks awal ke pq
                PriorityQueue<FifteenPuzzle> pq = new PriorityQueue<>();
                pq.add(matrix);

                FifteenPuzzle previous = new FifteenPuzzle();
                FifteenPuzzle now = new FifteenPuzzle();
                while (!pq.isEmpty()) {
                    previous = new FifteenPuzzle(now);
                    now = new FifteenPuzzle(pq.poll());

                    // menambahkan path untuk menuju matriks tersebut
```

```
                    now.addToPath(solutionPath, previous);

                    // jika sudah ketemu, looping akan berhenti
                    if (now.isSame(solution)) {
                        break;
                    }

                    // pembangkitan simpul untuk command UP
                    if (now.checkUP() &&
now.getPath().get(now.getPath().size()-1) != "DOWN") {
                        FifteenPuzzle up = new FifteenPuzzle(now);
                        up.UP();
                        up.getPath().add("UP");
                        pq.add(up);
                        totalNode += 1;
                    }

                    // pembangkitan simpul untuk command RIGHT
                    if (now.checkRIGHT() &&
now.getPath().get(now.getPath().size()-1) != "LEFT") {
                        FifteenPuzzle right = new FifteenPuzzle(now);
                        right.RIGHT();
                        right.getPath().add("RIGHT");
                        pq.add(right);
                        totalNode += 1;
                    }

                    // pembangkitan simpul untuk command DOWN
                    if (now.checkDOWN() &&
now.getPath().get(now.getPath().size()-1) != "UP") {
                        FifteenPuzzle down = new FifteenPuzzle(now);
                        down.DOWN();
                        down.getPath().add("DOWN");
                        pq.add(down);
                        totalNode += 1;
                    }

                    // pembangkitan simpul untuk command LEFT
                    if (now.checkLEFT() &&
now.getPath().get(now.getPath().size()-1) != "RIGHT") {
                        FifteenPuzzle left = new FifteenPuzzle(now);
                        left.LEFT();
                        left.getPath().add("LEFT");
                        pq.add(left);
                        totalNode += 1;
                    }
                }
```

```
                // waktu berhenti
                long stopTime = System.currentTimeMillis();

                // menampilkan urutan matriks dari posisi awal ke posisi akhir
                int count = 0;
                System.out.println("Posisi awal:");
                matrix.showMatrix();
                System.out.println();
                for (String command : solutionPath) {
                    count++;
                    System.out.printf("Langkah ke-%d: %s%n", count, command);
                    matrix.todo(command);
                    matrix.showMatrix();
                    System.out.println();
                }

                // menghitung dan menampilkan waktu eksekusi program
                long elapsedTime = stopTime - startTime;
                System.out.printf("Waktu eksekusi program = %d ms%n",
elapsedTime);

                // menampilkan jumlah simpul yang dibangkitkan
                System.out.printf("Jumlah simpul yang dibangkitkan = %d%n",
totalNode);

            } else {
                System.out.println("Puzzle ini tidak dapat diselesaikan!");
            }

        } catch (FileNotFoundException FileNotFound) {
            System.out.println("\nFile tidak ditemukan!");

        } catch (OutOfMemoryError OutOfMemory) {
            System.out.println("Nampaknya anda belum beruntung, silakan coba
lagi :)");
        }
    }
}
```

Berkas Teks

1. can_be_solved_1.txt

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

2. can_be_solved_2.txt

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

3. can_be_solved_3.txt

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

4. cant_be_solved_1.txt

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

5. cant_be_solved_2.txt

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

No.	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		✓

Alamat GitHub → https://github.com/johannes-ws/Tucil3_13520123