

# LAPORAN TUGAS KECIL 3

## IF2211 STRATEGI ALGORITMA

Atabik Muhammad Azfa Shofi – 13520159

[Link Repository](#)

### 1. Algoritma Branch and Bound pada penyelesaian 15-Puzzle

Algoritma Branch and Bound adalah salah satu algoritma yang mengaplikasikan algoritma BFS dengan elemen prioritas menggunakan least cost Search. Inti dari algoritma ini adalah dengan memberikan suatu value atau cost tertentu yang nantinya akan menjadi penentuan simpul berikutnya yang akan dikembangkan.

Persoalan 15-puzzle adalah salah satu persoalan yang dapat diselesaikan dengan menggunakan algoritma Branch and Bound ini. Persoalan ini berupa 16 kotak pada matriks berukuran 4x4 berisikan angka dari 1-15 dan satu kotak kosong yang nantinya dapat diisi oleh kotak sekitarnya dengan cara menggeser.



### 2. Source-code Program

Source-code utama program dibagi menjadi dua bagian yaitu kelas PuzzleMatrix dan PuzzleTree. Kelas PuzzleMatrix lebih berfokus pada satu keadaan Matrix puzzle dan method-method untuk membantu operasi pada matrix. Sedangkan, kelas PuzzleTree berfokus pada Pembuatan Tree dan method-method yang berkaitan dengan pembuatan tree. Isi source-code sendiri dapat dilihat pada repository karena akan memakan banyak ruang pada laporan. Namun, secara garis besar fungsi-fungsinya dapat dijelaskan sebagai berikut :

#### 1. Kelas PuzzleMatrix

Berikut disediakan tabel berisikan atribut dan method dari kelas PuzzleMatrix :

Atribut	Keterangan
Int [ ] [ ] matrices	Array of array of integer, merupakan matrix untuk menampung puzzle, berukuran 4x4
Int depth	Bertipe integer sebagai penyimpan depth (kedalaman) dari matrix pada pohon
Int cost	Bertipe integer sebagai penyimpan total cost dari simpul
Method	Keterangan
Konstruktor	Konstruktor kelas, default atau user-defined
Getter/setter	Getter/setter dari atribut depth dan cost
Void readFile	Membaca matriks dari file
Void printMatrix	Melakukan print elemen matriks
Int kurangI	Mendapatkan nilai kurangI dari input i
Int [ ] findEmptySlot	Mendapatkan indeks dari slot kosong
Boolean canSolved	Menghasilkan true jika puzzle dapat diselesaikan
Void Up/down/left/right	Melakukan pergeseran

2. Kelas PuzzleTree

Atribut	Keterangan
ArrayList<PuzzleMatrix> puzzleTree	Arraylist berisikan objek PuzzleMatrix, digunakan untuk menyimpan semua simpul yang telah dibangkitkan
ArrayList<PuzzleMatrix> solutionList	Arraylist untuk menyimpan puzzleMatrix hasil solusi
HashMap<PuzzleMatrix,PuzzleMatrix> prevRoot	HashMap menyimpan root sebelumnya dari PuzzleMatrix untuk mengetahui alur proses puzzle dari awal hingga akhir
Int simplices	Bertipe integer sebagai penyimpan total simpul yang telah dibangkitkan
Method	Keterangan
Konstruktor	Konstruktor kelas
PuzzleMatrix leastCost	Mencari simpul dengan cost terendah pada list puzzleTree
Void generateUp/Down/Right/Left	Melakukan generate simpul-simpul anak jika kondisi memenuhi
Void generateChilds	Menggabungkan seluruh method generate dengan kondisi tertentu
Void PuzzleSolving	Melakukan solving puzzle hingga dicapai hasil target lalu menyimpan seluruh solusi pada solutionList

3. Main program dibuat untuk melakukan program utama mulai dari input hingga output program. Berikut adalah tangkapan layar dari main program:

```

public static void main(String[] args) {
    PuzzleMatrix init = new PuzzleMatrix();
    init.readFile();
    init.printMatrix();
    if(init.canSolved()){
        long startTime = System.currentTimeMillis();
        PuzzleTree tree = new PuzzleTree();
        tree.puzzleTree.add(init);
        tree.generateChilds(init);
        tree.PuzzleSolving();
        long endTime = System.currentTimeMillis();
        for(PuzzleMatrix matrix : tree.solutionList){
            System.out.println("Proses: ");
            matrix.printMatrix();
        }
        System.out.println("Simpul yang dibangkitkan : " + tree.simplices);
        long totalTime = endTime - startTime;
        System.out.println("Total time : " + totalTime + " ms");
    }
}

```

### 3. Hasil Uji

Input	Output	Proses
<pre>1 2 3 4 5 6 E 8 9 10 7 11 13 14 15 12</pre> <p>test.txt</p>	<pre>KURANG[1] = 0 KURANG[2] = 0 KURANG[3] = 0 KURANG[4] = 0 KURANG[5] = 0 KURANG[6] = 0 KURANG[7] = 0 KURANG[8] = 1 KURANG[9] = 1 KURANG[10] = 1 KURANG[11] = 0 KURANG[12] = 0 KURANG[13] = 1 KURANG[14] = 1 KURANG[15] = 1 KURANG[16] = 9  Total + X = 16</pre> <p>Simpul yang dibangkitkan : 11 Total time : 0 ms</p>	<pre>Proses: 1 2 3 4 5 6 7 8 9 10 E 11 13 14 15 12  Proses: 1 2 3 4 5 6 7 8 9 10 11 E 13 14 15 12  Proses: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 E</pre>
<pre>1 2 3 E 5 6 7 4 9 10 12 8 13 14 11 15</pre> <p>test2.txt</p>	<pre>KURANG[1] = 0 KURANG[2] = 0 KURANG[3] = 0 KURANG[4] = 0 KURANG[5] = 1 KURANG[6] = 1 KURANG[7] = 1 KURANG[8] = 0 KURANG[9] = 1 KURANG[10] = 1 KURANG[11] = 0 KURANG[12] = 2 KURANG[13] = 1 KURANG[14] = 1 KURANG[15] = 0 KURANG[16] = 12  Total + X = 22</pre> <p>Simpul yang dibangkitkan : 15 Total time : 15 ms</p>	<pre>Proses: 1 2 3 4 5 6 7 8 9 10 E 12 13 14 11 15  Proses: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 E 15  Proses: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 E</pre>

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

test3.txt

```
KURANG[1] = 0
KURANG[2] = 1
KURANG[3] = 0
KURANG[4] = 2
KURANG[5] = 1
KURANG[6] = 0
KURANG[7] = 1
KURANG[8] = 5
KURANG[9] = 2
KURANG[10] = 9
KURANG[11] = 0
KURANG[12] = 1
KURANG[13] = 1
KURANG[14] = 1
KURANG[15] = 0
KURANG[16] = 8
```

Total + X = 32

```
Simpul yang dibangkitkan : 31896
Total time : 219 ms
```

```
Proses:
10 2 4 8
1 5 3 E
9 7 6 12
13 14 11 15
```

```
Proses:
10 2 4 E
1 5 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
10 2 E 4
1 5 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
10 E 2 4
1 5 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
E 10 2 4
1 5 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
1 10 2 4
E 5 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
1 10 2 4
5 E 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
1 E 2 4
5 10 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
1 2 E 4
5 10 3 8
9 7 6 12
13 14 11 15
```

```
Proses:
1 2 3 4
5 10 E 8
9 7 6 12
13 14 11 15
```

		<p>Proses:</p> <p>1 2 3 4</p> <p>5 10 6 8</p> <p>9 7 E 12</p> <p>13 14 11 15</p>	
		<p>Proses:</p> <p>1 2 3 4</p> <p>5 10 6 8</p> <p>9 E 7 12</p> <p>13 14 11 15</p>	
		<p>Proses:</p> <p>1 2 3 4</p> <p>5 E 6 8</p> <p>9 10 7 12</p> <p>13 14 11 15</p>	
		<p>Proses:</p> <p>1 2 3 4</p> <p>5 6 E 8</p> <p>9 10 7 12</p> <p>13 14 11 15</p>	
		<p>Proses:</p> <p>1 2 3 4</p> <p>5 6 7 8</p> <p>9 10 E 12</p> <p>13 14 11 15</p>	
		<p>Proses:</p> <p>1 2 3 4</p> <p>5 6 7 8</p> <p>9 10 11 12</p> <p>13 14 E 15</p>	
		<p>Proses:</p> <p>1 2 3 4</p> <p>5 6 7 8</p> <p>9 10 11 12</p> <p>13 14 15 E</p>	

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

test4.txt

```
KURANG[1] = 0
KURANG[2] = 1
KURANG[3] = 1
KURANG[4] = 1
KURANG[5] = 3
KURANG[6] = 0
KURANG[7] = 0
KURANG[8] = 1
KURANG[9] = 2
KURANG[10] = 2
KURANG[11] = 0
KURANG[12] = 1
KURANG[13] = 2
KURANG[14] = 2
KURANG[15] = 8
KURANG[16] = 8
```

Total + X = 32

Simpul yang dibangkitkan : 82536  
Total time : 3579 ms

Proses:

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

Proses:

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

Proses:

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

Proses:

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

Proses:

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

Proses:

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

Proses:

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

Proses:

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

Proses:

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

Proses:

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

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

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

test5.txt

```
KURANG[1] = 0
KURANG[2] = 0
KURANG[3] = 1
KURANG[4] = 1
KURANG[5] = 0
KURANG[6] = 0
KURANG[7] = 1
KURANG[8] = 0
KURANG[9] = 0
KURANG[10] = 0
KURANG[11] = 3
KURANG[12] = 6
KURANG[13] = 0
KURANG[14] = 4
KURANG[15] = 11
KURANG[16] = 10
```

Total + X = 37

Puzzle tidak dapat diselesaikan!

-



Poin	Ya	Tidak	Keterangan
Program berhasil dikompilasi	v	-	Tidak ada compile error dari seluruh file .java
Program berhasil running	v	-	Program dapat berjalan untuk beberapa kasus yang tidak terlalu besar
Program dapat menerima input dan menuliskan output	v	-	Program berhasil membaca input file dan menuliskan output
Luaran sudah benar untuk semua data uji	v	-	Program menuliskan gagal jika puzzle tidak dapat diselesaikan, dan menuliskan proses hingga puzzle selesai jika bisa
Bonus dibuat	-	v	-

## REFERENSI

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian1.pdf>