

# **LAPORAN TUGAS KECIL 3**

**IF2211 Strategi Algoritma**

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



Dipersiapkan oleh:

Morgen Sudyanto

13518093

**Sekolah Teknik Elektro dan Informatika – Institut Teknologi  
Bandung**

**Jl. Ganesha 10, Bandung 40132**

# BAB I

## CARA KERJA PROGRAM

- I. Deklarasi variabel global
  - puzzle: Sebuah matrix 4x4 yang berisi puzzle awal.
  - pq: Priority queue yang berisi tuple <cost, distanceFromRoot, node, moves, puzzle>. Elemen top dari pq ini adalah elemen dengan cost terkecil.
  - visited: Sebuah list yang berisi state puzzle yang sudah pernah dikunjungi.
  - nodeNumber: Sebuah angka yang menunjukkan banyaknya node yang sudah dibangkitkan.
  - di: Sebuah list yang berisi movement up/down
  - dj: Sebuah list yang berisi movement left/right
  - move: Sebuah list yang berisi sebuah char movement (urdl)
- II. Menerima masukan filename dari user.
- III. Membaca puzzle yang ada di file
- IV. Mencetak puzzle awal ke layar
- V. Menghitung nilai dari  $\sum_{i=1}^{16} KURANG(i) + X$ 
  - Kurang(i) adalah banyaknya kotak tidak kosong yang berisi angka j dimana  $j < i$  dan  $posisi(j) > posisi(i)$ .
  - X bernilai 1 jika kotak kosong (bernilai 16) terletak pada kotak dengan posisi ganjil, dan bernilai 0 jika sebaliknya.
- VI. Menentukan apakah puzzle awal dapat diselesaikan. Jika tidak dapat diselesaikan, maka program mengeluarkan pesan “Tidak ada solusi”, lalu keluar.
  - Puzzle dapat diselesaikan bila nilai dari fungsi kurang + x adalah genap.
- VII. Menjalankan algoritma branch and bound.
 

Algoritma branch and bound:

  1. Masukkan state awal ke dalam pq. State awal terdiri dari nilai G dari puzzle, jarak ke root berupa 0, nomor node yaitu 1, moves berupa string kosong dan puzzle awal.
  2. Cek apakah puzzle awal merupakan solusi. Jika ya, maka solusi sudah ditemukan. Kembalikan string moves (string kosong).
  3. Selama pq tidak kosong, lakukan:
    - a. Ambil dan pop state teratas dari pq.
    - b. Tambahkan puzzle ke dalam list visited.
    - c. Cek apakah puzzle sekarang merupakan solusi. Jika ya, maka solusi sudah ditemukan. Kembalikan string moves. Jika tidak, bangkitkan semua anak dari puzzle yang belum ada di list visited. Hitung costnya dengan

menjumlahkan nilai G dari anak dan jarak anak ke root. Semua anak yang dibangkitkan kemudian dimasukkan ke dalam pq.

- VIII. Cetak langkah yang diperlukan ke layar
- IX. Perlihatkan langkah beserta puzzle setiap langkahnya
- X. Cetak waktu eksekusi. Waktu eksekusi ini hanya waktu yang diperlukan untuk menjalankan algoritma branch and boundnya saja.
- XI. Cetak jumlah simpul yang dibangkitkan. Angka ini diambil dari nodeNumber.

## BAB II

### SCREENSHOT PROGRAM

- I. Test case 1: Ada solusi (yes1.in)

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

```
D:\kuliah\Tucil-Stima\15Puzzle (master -> origin)
λ python solve.py
Masukkan file input: yes1.in
=====
1  3  12 4
5  2  16 7
9  15 6  8
13 10 14 11
=====
KURANG(1) = 0
KURANG(2) = 0
KURANG(3) = 1
KURANG(4) = 1
KURANG(5) = 1
KURANG(6) = 0
KURANG(7) = 1
KURANG(8) = 0
KURANG(9) = 2
KURANG(10) = 0
KURANG(11) = 0
KURANG(12) = 9
KURANG(13) = 2
KURANG(14) = 1
KURANG(15) = 6
KURANG(16) = 9
SUM(KURANG) + X = 34
Moves:  uldrdlurrdlldrruldr
Steps:
=====
1  3  12 4
5  2  16 7
9  15 6  8
13 10 14 11
=====
Step 1 => u
=====
1  3  16 4
5  2  12 7
9  15 6  8
13 10 14 11
=====
```

Step 2 => l

=====

1 16 3 4

5 2 12 7

9 15 6 8

13 10 14 11

=====

Step 3 => d

=====

1 2 3 4

5 16 12 7

9 15 6 8

13 10 14 11

=====

Step 4 => r

=====

1 2 3 4

5 12 16 7

9 15 6 8

13 10 14 11

=====

Step 5 => d

=====

1 2 3 4

5 12 6 7

9 15 16 8

13 10 14 11

=====

Step 6 => l

=====

1 2 3 4

5 12 6 7

9 16 15 8

13 10 14 11

=====

Step 7 => u

=====

1 2 3 4

5 16 6 7

9 12 15 8

13 10 14 11

Step 8 => r

=====

1 2 3 4

5 6 16 7

9 12 15 8

13 10 14 11

=====

Step 9 => r

=====

1 2 3 4

5 6 7 16

9 12 15 8

13 10 14 11

=====

Step 10 => d

=====

1 2 3 4

5 6 7 8

9 12 15 16

13 10 14 11

=====

Step 11 => l

=====

1 2 3 4

5 6 7 8

9 12 16 15

13 10 14 11

=====

Step 12 => l

=====

1 2 3 4

5 6 7 8

9 16 12 15

13 10 14 11

=====

Step 13 => d

=====

1 2 3 4

5 6 7 8

9 10 12 15

13 16 14 11

```

Step 14 => r
=====
1  2  3  4
5  6  7  8
9 10 12 15
13 14 16 11
=====
Step 15 => r
=====
1  2  3  4
5  6  7  8
9 10 12 15
13 14 11 16
=====
Step 16 => u
=====
1  2  3  4
5  6  7  8
9 10 12 16
13 14 11 15
=====
Step 17 => l
=====
1  2  3  4
5  6  7  8
9 10 16 12
13 14 11 15
=====
Step 18 => d
=====
1  2  3  4
5  6  7  8
9 10 11 12
13 14 16 15
=====
Step 19 => r
=====
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16

```

```

Execution time: 2.4019316999999996 sec
Jumlah simpul yang dibangkitkan: 9691

```

II. Test case 2: Ada solusi (yes2.in)

```

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

```

D:\kuliah\Tucil-Stima\15Puzzle (master -> origin)

λ python solve.py

Masukkan file input: yes2.in

=====

16 2 3 7

1 5 4 6

10 11 12 8

9 13 14 15

=====

KURANG(1) = 0

KURANG(2) = 1

KURANG(3) = 1

KURANG(4) = 0

KURANG(5) = 1

KURANG(6) = 0

KURANG(7) = 4

KURANG(8) = 0

KURANG(9) = 0

KURANG(10) = 2

KURANG(11) = 2

KURANG(12) = 2

KURANG(13) = 0

KURANG(14) = 0

KURANG(15) = 0

KURANG(16) = 15

SUM(KURANG) + X = 28

Moves: drurdrulldrd11ldrrr

Steps:

=====

16 2 3 7

1 5 4 6

10 11 12 8

9 13 14 15

=====

Step 1 => d

=====

1 2 3 7

16 5 4 6

10 11 12 8

9 13 14 15

=====

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

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

Test case 2: Aila

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



Step 2 => r

=====

1 2 3 7

5 16 4 6

10 11 12 8

9 13 14 15

=====

Step 3 => u

=====

1 16 3 7

5 2 4 6

10 11 12 8

9 13 14 15

=====

Step 4 => r

=====

1 3 16 7

5 2 4 6

10 11 12 8

9 13 14 15

=====

Step 5 => d

=====

1 3 4 7

5 2 16 6

10 11 12 8

9 13 14 15

=====

Step 6 => r

=====

1 3 4 7

5 2 6 16

10 11 12 8

9 13 14 15

=====

Step 7 => u

=====

1 3 4 16

5 2 6 7

10 11 12 8

9 13 14 15

Step 8 => l

=====

1 3 16 4

5 2 6 7

10 11 12 8

9 13 14 15

=====

Step 9 => l

=====

1 16 3 4

5 2 6 7

10 11 12 8

9 13 14 15

=====

Step 10 => d

=====

1 2 3 4

5 16 6 7

10 11 12 8

9 13 14 15

=====

Step 11 => r

=====

1 2 3 4

5 6 16 7

10 11 12 8

9 13 14 15

=====

Step 12 => r

=====

1 2 3 4

5 6 7 16

10 11 12 8

9 13 14 15

=====

Step 13 => d

=====

1 2 3 4

5 6 7 8

10 11 12 16

9 13 14 15

Step 14 => l

=====

1 2 3 4

5 6 7 8

10 11 16 12

9 13 14 15

=====

Step 15 => l

=====

1 2 3 4

5 6 7 8

10 16 11 12

9 13 14 15

=====

Step 16 => l

=====

1 2 3 4

5 6 7 8

16 10 11 12

9 13 14 15

=====

Step 17 => d

=====

1 2 3 4

5 6 7 8

9 10 11 12

16 13 14 15

=====

Step 18 => r

=====

1 2 3 4

5 6 7 8

9 10 11 12

13 16 14 15

=====

Step 19 => r

=====

1 2 3 4

5 6 7 8

9 10 11 12

13 14 16 15

III. Test case 3: Ada solusi (yes3.in)

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

```
D:\kuliah\Tucil-Stima\15Puzzle (master -> origin)
λ python solve.py
Masukkan file input: yes3.in
=====
1 2 4 8
5 6 3 15
10 11 7 14
9 13 12 16
=====
KURANG(1) = 0
KURANG(2) = 0
KURANG(3) = 0
KURANG(4) = 1
KURANG(5) = 1
KURANG(6) = 1
KURANG(7) = 0
KURANG(8) = 4
KURANG(9) = 0
KURANG(10) = 2
KURANG(11) = 2
KURANG(12) = 0
KURANG(13) = 1
KURANG(14) = 3
KURANG(15) = 7
KURANG(16) = 0
SUM(KURANG) + X = 22
Moves: uuuldddrullldrrr
Steps:
=====
1 2 4 8
5 6 3 15
10 11 7 14
9 13 12 16
=====
Step 1 => u
=====
1 2 4 8
5 6 3 15
10 11 7 16
9 13 12 14
```

```
Step 2 => u
=====
1  2  4  8
5  6  3 16
10 11 7 15
9  13 12 14
=====
Step 3 => u
=====
1  2  4 16
5  6  3  8
10 11 7 15
9  13 12 14
=====
Step 4 => l
=====
1  2 16 4
5  6  3  8
10 11 7 15
9  13 12 14
=====
Step 5 => d
=====
1  2  3  4
5  6 16 8
10 11 7 15
9  13 12 14
=====
Step 6 => d
=====
1  2  3  4
5  6  7  8
10 11 16 15
9  13 12 14
=====
Step 7 => d
=====
1  2  3  4
5  6  7  8
10 11 12 15
9  13 16 14
```

Step 8 => r

=====

1 2 3 4

5 6 7 8

10 11 12 15

9 13 14 16

=====

Step 9 => u

=====

1 2 3 4

5 6 7 8

10 11 12 16

9 13 14 15

=====

Step 10 => l

=====

1 2 3 4

5 6 7 8

10 11 16 12

9 13 14 15

=====

Step 11 => l

=====

1 2 3 4

5 6 7 8

10 16 11 12

9 13 14 15

=====

Step 12 => l

=====

1 2 3 4

5 6 7 8

16 10 11 12

9 13 14 15

=====

Step 13 => d

=====

1 2 3 4

5 6 7 8

9 10 11 12

16 13 14 15

Google

SaladVM Altroid TC nCode Problems 13 1300-3400

Slime3

File View Insert Format Tools Add-ons Help

1 7 100% 0 Normal text 0 Total Rows: 0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

1 2 3 4

5 6 7 8

10 11 12 15

9 13 14 16

=====

Step 9 => u

=====

1 2 3 4

5 6 7 8

10 11 12 16

9 13 14 15

=====

Step 10 => l

=====

1 2 3 4

5 6 7 8

10 11 16 12

9 13 14 15

=====

Step 11 => l

=====

1 2 3 4

5 6 7 8

10 16 11 12

9 13 14 15

=====

Step 12 => l

=====

1 2 3 4

5 6 7 8

16 10 11 12

9 13 14 15

IV) Test case 4: Ticks

=====

```

=====
Step 14 => r
=====
1  2  3  4
5  6  7  8
9 10 11 12
13 16 14 15
=====
Step 15 => r
=====
1  2  3  4
5  6  7  8
9 10 11 12
13 14 16 15
=====
Step 16 => r
=====
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
=====
Execution time: 0.044831300000000024 sec
Jumlah simpul yang dibangkitkan: 377

```

IV. Test case 4: Tidak ada solusi (no1.in)

```

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

```



```
D:\kuliah\Tucil-Stima\15Puzzle (master -> origin)
λ python solve.py
Masukkan file input: no1.in
=====
2 3 12 4
5 1 16 7
9 15 6 8
13 10 14 11
=====
KURANG(1) = 0
KURANG(2) = 1
KURANG(3) = 1
KURANG(4) = 1
KURANG(5) = 1
KURANG(6) = 0
KURANG(7) = 1
KURANG(8) = 0
KURANG(9) = 2
KURANG(10) = 0
KURANG(11) = 0
KURANG(12) = 9
KURANG(13) = 2
KURANG(14) = 1
KURANG(15) = 6
KURANG(16) = 9
SUM(KURANG) + X = 35
Tidak ada solusi.
```

V. Test case 5: Tidak ada solusi (no2.in)

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

D:\kuliah\Tucil-Stima\15Puzzle (master -> origin)

λ python solve.py

Masukkan file input: no2.in

=====

16 1 3 7

2 5 4 6

10 11 12 8

9 13 14 15

=====

KURANG(1) = 0

KURANG(2) = 0

KURANG(3) = 1

KURANG(4) = 0

KURANG(5) = 1

KURANG(6) = 0

KURANG(7) = 4

KURANG(8) = 0

KURANG(9) = 0

KURANG(10) = 2

KURANG(11) = 2

KURANG(12) = 2

KURANG(13) = 0

KURANG(14) = 0

KURANG(15) = 0

KURANG(16) = 15

SUM(KURANG) + X = 27

Tidak ada solusi.

no2.in  
16 1 3 7  
2 5 4 6  
10 11 12 8  
9 13 14 15

V Test case 3

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

**BAB III**  
**CHECKLIST KEBERHASILAN PROGRAM**

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