

LAPORAN TUGAS KECIL 3 IF2211 STRATEGI ALGORITMA SEMESTER II TAHUN 2019/2020

Penyelesaian Persoalan 15-Puzzle dengan Algoritma Branch and Bound



Dipersiapkan oleh:

Muchammad Ibnu Sidqi - 13518072

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 4013

Daftar Isi

1. Algoritma Branch and Bound.....	3
2. Dokumentasi Program	4
4. Lampiran.....	11

1. Algoritma Branch and Bound

1.1. Implementasi algoritma Branch and Bound pada Persoalan 15-Puzzle

Misalkan terdapat puzzle berukuran 4×4 dengan masing-masing memiliki susunan angka acak dengan angka berkisar antara 1 hingga 15 yang unik sehingga terdapat 1 puzzle yang tidak memiliki angka. Bagaimanakah cara agar susunan angka tersebut teratur?

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

(a) Susunan awal

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

(b) Susunan akhir

gambar 1. Contoh Persoalan 15-Puzzle

1. Lakukan analisis terhadap status yang memungkinkan terjadi pada setiap langkah dalam mencari solusi dari persoalan. Dalam Persoalan 15-Puzzle terdapat 4 buah status yang mungkin terjadi pada setiap langkah yaitu terjadi pergeseran puzzle yang tidak memiliki angka dengan arah ke atas, bawah, kiri, dan kanan.
2. Periksa masukkan 15-Puzzle apakah dapat diselesaikan atau tidak dengan menggunakan fungsi $kurang(i)$ dan faktor X . $kurang(i)$ = banyaknya puzzle bernomor j sedemikian sehingga $j < i$ dan $POSISI(j) > POSISI(i)$. $POSISI(i)$ = posisi puzzle bernomor i pada susunan yang diperiksa. serta faktor X dengan X bernilai 1 bila puzzle kosong terdapat pada puzzle yang diarsir. Lalu dengan menggunakan persamaan $total = \sum_{i=1}^{16} kurang(i) + X$ bila didapatkan total bernilai ganjil maka masukkan 15-Puzzle tidak dapat diselesaikan, tetapi bila didapatkan total bernilai genap maka masukkan 15-Puzzle dapat diselesaikan.

gambar 2. Arsir Puzzle pada Faktor X

3. Bila masukkan 15-Puzzle dapat diselesaikan, Masukkan simpul akar ke dalam antrian Queue. Jika simpul akar adalah simpul solusi (goal node), maka solusi telah ditemukan. Pengecekan simpul solusi dilaksanakan dengan menggunakan persamaan $\widehat{c}(i) = \widehat{f}(i) + \widehat{g}(i)$ dengan $\widehat{c}(i)$ adalah total biaya(cost) pada simpul, $\widehat{f}(i)$ adalah panjang lintasan dari simpul akar serta $\widehat{g}(i)$ adalah jumlah puzzle tidak kosong yang susunannya tidak terdapat pada susunan akhir. Dalam hal ini, simpul solusi adalah simpul yang memiliki nilai $\widehat{g}(i) = 0$. Bila simpul akar merupakan solusi akhir hentikan pencarian solusi.

4. Pilih dari antrian Queue simpul i yang memiliki nilai total biaya($cost$) $\widehat{c(i)}$ paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih satu secara sembarang.
5. Cek apakah simpul yang dipilih merupakan simpul solusi akhir seperti pada langkah 3, bila simpul merupakan simpul solusi akhir maka hentikan pencarian. Bila tidak, Bangkitkan seluruh simpul dengan status yang mungkin terjadi lalu masukkan dalam antrian Queue. Bila seluruh simpul anak sudah dibangkitkan maka lakukan pengecekan pada antrian Queue, bila antrian Queue kosong hentikan pencarian solusi.
6. Lakukan kembali langkah 4 hingga ditemukan simpul solusi akhir atau bila antrian Queue kosong.

2.Dokumentasi Program

2.1. Solve 15-Puzzle



Masukkan persoalan 15-Puzzle dengan nilai 16 menandakan puzzle(ubin) kosong:

```

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

```

Output program:

```

Masukkan nama file: puzzle1.txt
PUZZLE AWAL:
-----
 1  2  3  4
 5  6      8
 9 10  7 11
13 14 15 12
-----

PERHITUNGAN NILAI KURANG:
-----
Indeks ke- 1
kurang( 1 ) = 0
-----
Indeks ke- 2
kurang( 2 ) = 0
-----
Indeks ke- 3
kurang( 3 ) = 0
-----
Indeks ke- 4
kurang( 4 ) = 0
-----
Indeks ke- 5
kurang( 5 ) = 0
-----

```

```
-----  
Indeks ke- 6  
kurang( 6 ) = 0  
-----
```

```
-----  
Indeks ke- 7  
kurang( 7 ) = 0  
-----
```

```
-----  
Indeks ke- 8  
kurang( 8 ) = 1  
-----
```

```
-----  
Indeks ke- 9  
kurang( 9 ) = 1  
-----
```

```
-----  
Indeks ke- 10  
kurang( 10 ) = 1  
-----
```

```
-----  
Indeks ke- 11  
kurang( 11 ) = 0  
-----
```

```
-----  
Indeks ke- 12  
kurang( 12 ) = 0  
-----
```

```
-----  
Indeks ke- 13  
kurang( 13 ) = 1  
-----
```

```
-----  
Indeks ke- 14  
kurang( 14 ) = 1  
-----
```

```
-----  
Indeks ke- 15  
kurang( 15 ) = 1  
-----
```

```
-----  
Indeks ke- 16  
kurang( 16 ) = 9  
-----
```

```
-----  
Nilai kurang total = 15  
Nilai X = 1  
Nilai Total = 16  
Persoalan DAPAT diselesaikan  
-----
```

PATH MENUJU SOLUSI:

```
-----  
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  
-----
```

Jumlah simpul yang dibangkitkan adalah 9
Waktu program untuk menjalankan proses: 10547700 nanoseconds

Masukkan persoalan 15-Puzzle:

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

output program:

Masukkan nama file: puzzle3.txt
PUZZLE AWAL:

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

PERHITUNGAN NILAI KURANG:

Indeks ke- 1
kurang(1) = 0

Indeks ke- 2
kurang(2) = 0

Indeks ke- 3
kurang(3) = 0

Indeks ke- 4
kurang(4) = 0

Indeks ke- 5
kurang(5) = 0

Indeks ke- 6
kurang(6) = 0

Indeks ke- 7
kurang(7) = 0

Indeks ke- 8
kurang(8) = 1

Indeks ke- 9
kurang(9) = 1

Indeks ke- 10
kurang(10) = 1

Indeks ke- 11
kurang(11) = 0

```

-----
Indeks ke- 12
kurang( 12 ) = 1
-----
Indeks ke- 13
kurang( 13 ) = 1
-----
Indeks ke- 14
kurang( 14 ) = 1
-----
Indeks ke- 15
kurang( 15 ) = 0
-----
Indeks ke- 16
kurang( 16 ) = 10
-----
Nilai kurang total = 16
Nilai X = 0
Nilai Total = 16
Persoalan DAPAT diselesaikan

```

```

PATH MENUJU SOLUSI:
-----
1  2  3  4
5  6      8
9 10  7 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
-----
Jumlah simpul yang dibangkitkan adalah 12
Waktu program untuk menjalankan proses: 2974600 _nanoseconds

```

2.2. *Unsolve 15-Puzzle*



Masukkan persoalan 15-Puzzle:

```

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

```

Output program:

Masukkan nama file: puzzle2.txt
PUZZLE AWAL:

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

PERHITUNGAN NILAI KURANG:

```
-----  
Indeks ke- 1  
kurang( 1 ) = 0  
-----
```

```
-----  
Indeks ke- 2  
kurang( 2 ) = 0  
-----
```

```
-----  
Indeks ke- 3  
kurang( 3 ) = 1  
-----
```

```
-----  
Indeks ke- 4  
kurang( 4 ) = 1  
-----
```

```
-----  
Indeks ke- 5  
kurang( 5 ) = 0  
-----
```

```
-----  
Indeks ke- 6  
kurang( 6 ) = 0  
-----
```

```
-----  
Indeks ke- 7  
kurang( 7 ) = 1  
-----
```

```
-----  
Indeks ke- 8  
kurang( 8 ) = 0  
-----
```

```
-----  
Indeks ke- 9  
kurang( 9 ) = 0  
-----
```



```

-----
Indeks ke- 9
kurang( 9 ) = 0
-----
Indeks ke- 10
kurang( 10 ) = 0
-----
Indeks ke- 11
kurang( 11 ) = 3
-----
Indeks ke- 12
kurang( 12 ) = 6
-----
Indeks ke- 13
kurang( 13 ) = 0
-----
Indeks ke- 14
kurang( 14 ) = 4
-----
Indeks ke- 15
kurang( 15 ) = 11
-----
Indeks ke- 16
kurang( 16 ) = 10
-----
Nilai kurang total = 37
Nilai X = 0
Nilai Total = 37
Persoalan TIDAK DAPAT diselesaikan

```



Masukkan persoalan 15-Puzzle:

```

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

```

Output program:

```

Masukkan nama file: puzzle4.txt
PUZZLE AWAL:
-----
    1  2  3
  4  5  6  7
  8  9 10 11
12 13 14 15
-----

PERHITUNGAN NILAI KURANG:
-----
Indeks ke- 1
kurang( 1 ) = 0
-----
Indeks ke- 2
kurang( 2 ) = 0
-----

```

```
-----  
Indeks ke- 3  
kurang( 3 ) = 0  
-----  
Indeks ke- 4  
kurang( 4 ) = 0  
-----  
Indeks ke- 5  
kurang( 5 ) = 0  
-----  
Indeks ke- 6  
kurang( 6 ) = 0  
-----  
Indeks ke- 7  
kurang( 7 ) = 0  
-----  
Indeks ke- 8  
kurang( 8 ) = 0  
-----  
Indeks ke- 9  
kurang( 9 ) = 0  
-----  
Indeks ke- 10  
kurang( 10 ) = 0  
-----  
Indeks ke- 11  
kurang( 11 ) = 0  
-----
```

```
-----  
Indeks ke- 12  
kurang( 12 ) = 0  
-----  
Indeks ke- 13  
kurang( 13 ) = 0  
-----  
Indeks ke- 14  
kurang( 14 ) = 0  
-----  
Indeks ke- 15  
kurang( 15 ) = 0  
-----  
Indeks ke- 16  
kurang( 16 ) = 15  
-----  
Nilai kurang total = 15  
Nilai X = 0  
Nilai Total = 15  
Persoalan TIDAK DAPAT diselesaikan
```

Lampiran

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat menerima input dan menuliskan <i>output</i>	✓	
4. Luaran sudah benar untuk semua n	✓	

Spesifikasi Komputer

Operating System: Windows 10 Home Single Language 64-bit (10.0, Build 17763)
Language: English (Regional Setting: English)
System Manufacturer: ASUSTeK COMPUTER INC.
System Model: VivoBook 14_ASUS Laptop X442UF
BIOS: X442UF.308
Processor: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (8 CPUs), ~2.0GHz
Memory: 8192MB RAM
Page file: 2976MB used, 6377MB available
DirectX Version: DirectX 12