IF 2210 STRATEGI ALGORITMA
ALGORITMA BRANCH AND BOUND
PENYELESAIAN 15-PUZZLE PROBLEM

Disusun Oleh

Muhammad Ayyub Abdurrahman 13518076

PRODI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020

BAB 1
CARA KERJA PENYELESAIAN MASALAH

1. Algoritma Branch and Bound

Algoritma Branch and Bound adalah suatu metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diilustrasikan ke dalam pohon ruang status. Pohon ruang status tersebut dibangun dengan skema BFS, namun dengan skema yang berbeda. Untuk mempercepat pencarian ke simpul solusi, maka setiap simpul diberi sebuah nilai ongkos (cost). Simpul berikutnya yang akan diekspansi adalah simpul yang memiliki ongkos paling kecil di antara simpul-simpul hidup lainnya. Sedangkan simpul lainnya dimatikan. Nilai ongkos pada simpul i dinyatakan dengan :

ĉ(i)= nilai taksiran lintasan termurah dari I ke tujuan.

Algoritma dimulai dengan pengisian sebuah nilai ke akar dari pohon pencarian tersebut. Pencabangan dilakukan dengan memasang sebuah pending node ke pending node lain yang lebih rendah levelnya. Bobot juga dihitung pada setiap proses dan ditulis di simpul pohon. Jika sebuah simpul diketahui merupakan solusi yang tidak mungkin bagi persoalan yang dihadapi, simpul tersebut diisi dengan nilai tak terbatas (infinity). Algoritma berhenti ketika sudah tidak mungkin lagi untuk membentuk simpul baru di pohon atau hasil terakhir yang ditemukan merupakan hasil yang lebih rendah (minimum) dari isi simpul yang telah ada pada level yang lebih rendah.

2. 15-Puzzle

15-Puzzle adalah sebuah permainan klasik berupa puzzle geser yang terdiri dari bingkai ubin persegi bernomor yang disusun secara acak dengan satu ubin hilang. Objektif dari puzzle ini adalah menempatkan ubin secara berurutan dengan membuat gerakan geser yang menggunakan ruang kosong.

Permasalahan ini diciptakan dan diperkenalkan oleh Noyes Palmer Chapman pada awal 1874. Lalu kemudia pada tahun 1879 Johnson & Story membuktikan bagaimana permasalahn N-puzzle mungkin dan tidak mungkin untuk diselesaikan. Gambar 2.1 contoh 15-puzzle

Gambar 1.1. 15-Puzzle
Sumber : http://migo.sixbit.org/puzzles/fifteen/ diakses 25-03-2020 pukul 12.46

3. Penyelesaian Permasalahan 15-Puzzle dengan Algoritma Branch and Bound

Sebelum mencari solusi dengan menggunakan algoritma branch and bound, kita harus menentukan apakah puzzle tujuan tersebut dapat dicapai atau tidak dari suatu puzzle awal. Untuk menentukannya, kita harus menghitung nilai dari fungsi kurang dai matriks tersebut. Pertama-tama kita harus menomori posisi semua kotak penempatan ubin (bukan ubinnya) dengan nomor dari 1-16, Lalu kita akan menghitung posisinya dengan fungsi posisi(i).

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

Fungsi Posisi(i) adalah posisi dari ubin yang bernomor i berdasarkan matriks diatas. Jika melihat pada gambar 1.1 (pada halaman sebelumnya), nilai dari fungsi posisi (1) adalah 13.

Syarat suatu kondisi awal yang dapat mencapai kondisi simpul goal adalah

$$\sum \square\square\square\square\square\square(\square) + \square\text{bernilai genap.}$$

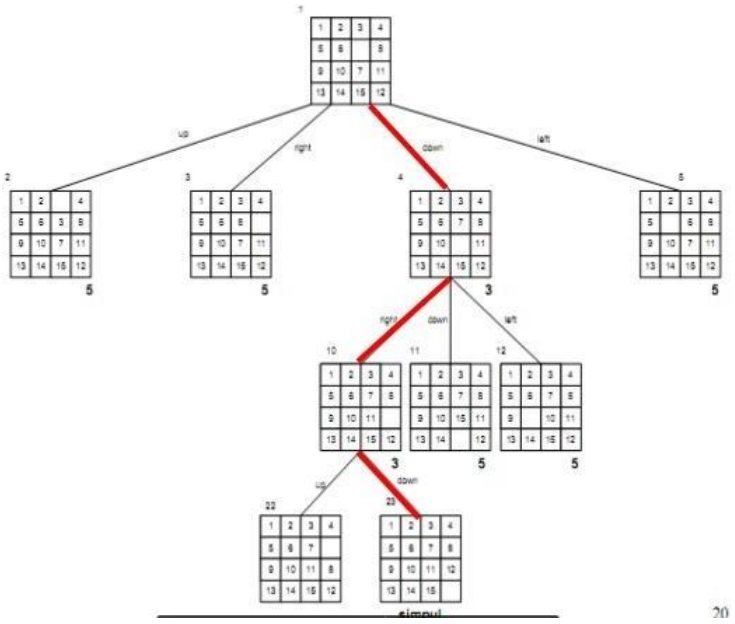
KURANG(i) didefinisikan sebagai jumlah ubin "j" demikian sehingga j < i dan POSISI(j) > POSISI(i). sebagai contoh, pada gambar 1.1 nilai dari fungsi KURANG(1) = 0. Untuk ubin kosong yang kosong,nilai I diset sebagai 16. X adalah posisi ubin kosong di awal (POSISI(16) berada di daerah tak diarsir pada matriks maka X = 0 dan sebaliknya, X = 1 jika posisi 16 berada di daerah yang diarsir). Dengan adanya batasan tersebut maka simpul yang tidak dapat menuju ke simpul tujuan tidak perlu dibangkitkan anak-anaknya. Ide yang diterapkan di BFS dan DFS diterapkan juga disini dimana ubin kosong menjadi acuan dalam menentukan arah atau menuju suatu simpul.

Lalu, langkah selanjutnya adalah menentukan bagaimana menghitung nilai batas / Bounding. Penghitungan nilai ini ditentukan dengan taksiran

$$c'(P) = f(p) + g'(p)$$

dengan
f(p) adalah panjang lintasan dari simpul akar ke P (simpul yang akan diexpand)
g'(p) ubin tidak kosong yang tidak terdapat di simpul tujuan.

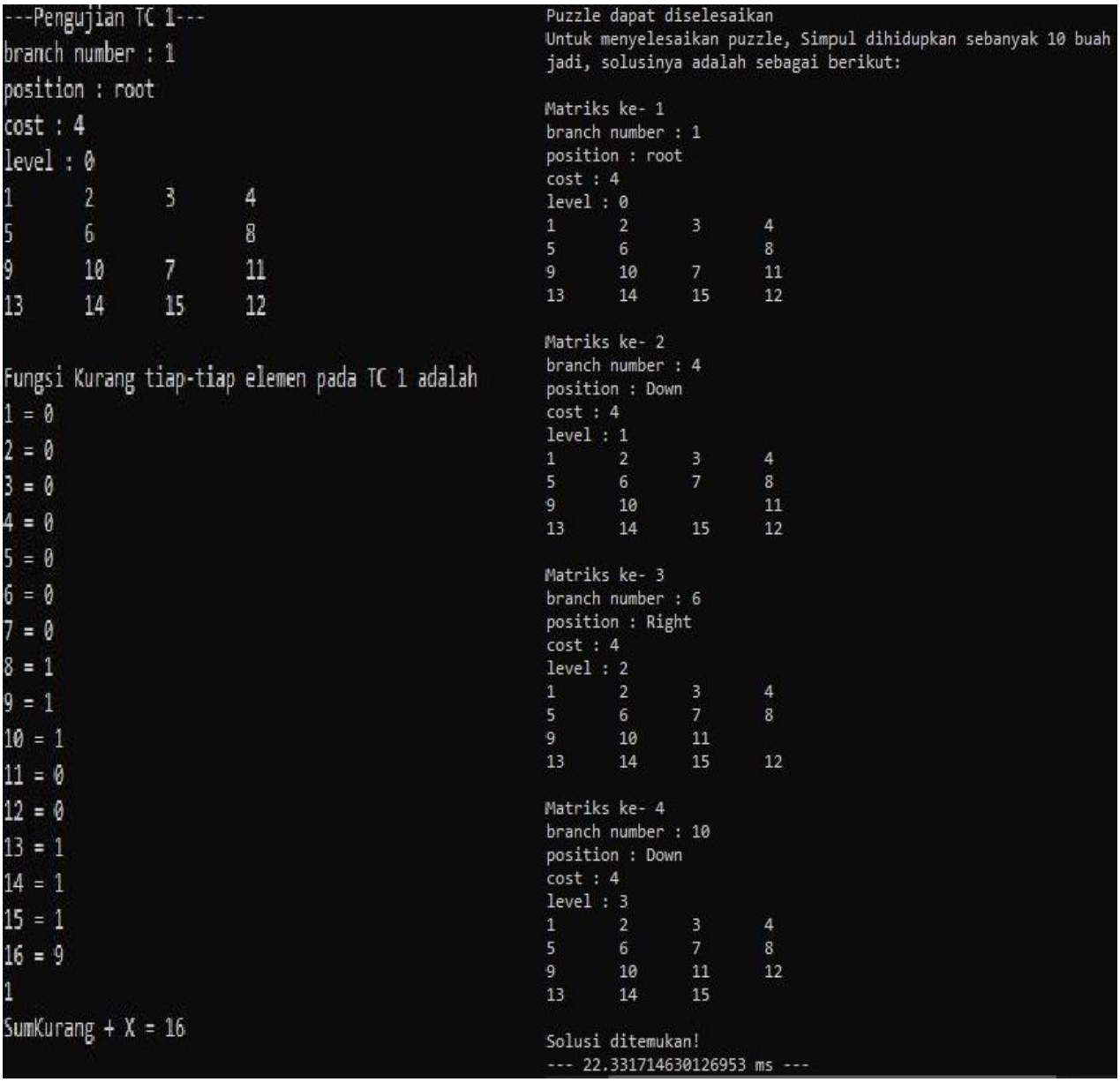
Pada persoalan 15-puzzle ini digunakan juga algoritma backtrack dimana hal ini dilakukan jika kita mengexpand anak-anak dari suatu simpul ternyata nilai bound yang didapat dari mereka lebih besar dari suatu simpul yang pernah diciptakan sebelumnya, maka simpul tersebut akan dihiudpkan lagi(backtracking) dan mengexpand anak-anaknya. Metode ini mirip dengan konsep priority queue dengan komparasi berdasarkan nilai c'(P) masing-masing simpul. Dengan struktur data ini maka otomatis simpul yang memiliki nilai bound yang paling minimal akan berada di head. Penggambaran proses B&B dapat dilihat di gambar 1.2.



Gambar 1.2 Penyelesaian 15-Puzzle
Sumber :Dokumen Spesifikasi Tugas Kecil 3 diakses pukul 14.27

BAB 2
KODE PROGRAM

1. Hasil Output TC1 (file txt : tclaporan1.txt)



Gambar 2.1. Hasil pengujian pada Test Case 1

2. **Hasil Output TC2 (file txt : tclaporan2.txt)**

```
---Pengujian TC 2---
branch number : 1
position : root
cost : 4
level : 0
1     2     3     4
5     6     7     8
9     10    11    12
      13    14    15

Fungsi Kurang tiap-tiap elemen pada TC 2 adalah
1 = 0
2 = 0
3 = 0
4 = 0
5 = 0
6 = 0
7 = 0
8 = 0
9 = 0
10 = 0
11 = 0
12 = 0
13 = 0
14 = 0
15 = 0
16 = 3
1
SumKurang + X = 4


Puzzle dapat diselesaikan
Untuk menyelesaikan puzzle, Simpul dihidupkan sebanyak 7 buah
jadi, solusinya adalah sebagai berikut:

Matriks ke- 1
branch number : 1
position : root
cost : 4
level : 0
1       2       3       4
5       6       7       8
9       10      11      12
        13      14      15

Matriks ke- 2
branch number : 3
position : Right
cost : 4
level : 1
1       2       3       4
5       6       7       8
9       10      11      12
13              14      15

Matriks ke- 3
branch number : 5
position : Right
cost : 4
level : 2
1       2       3       4
5       6       7       8
9       10      11      12
13      14              15

Matriks ke- 4
branch number : 7
position : Right
cost : 4
level : 3
1       2       3       4
5       6       7       8
9       10      11      12
13      14      15

Solusi ditemukan!
--- 19.971370697021484 ms ---
```

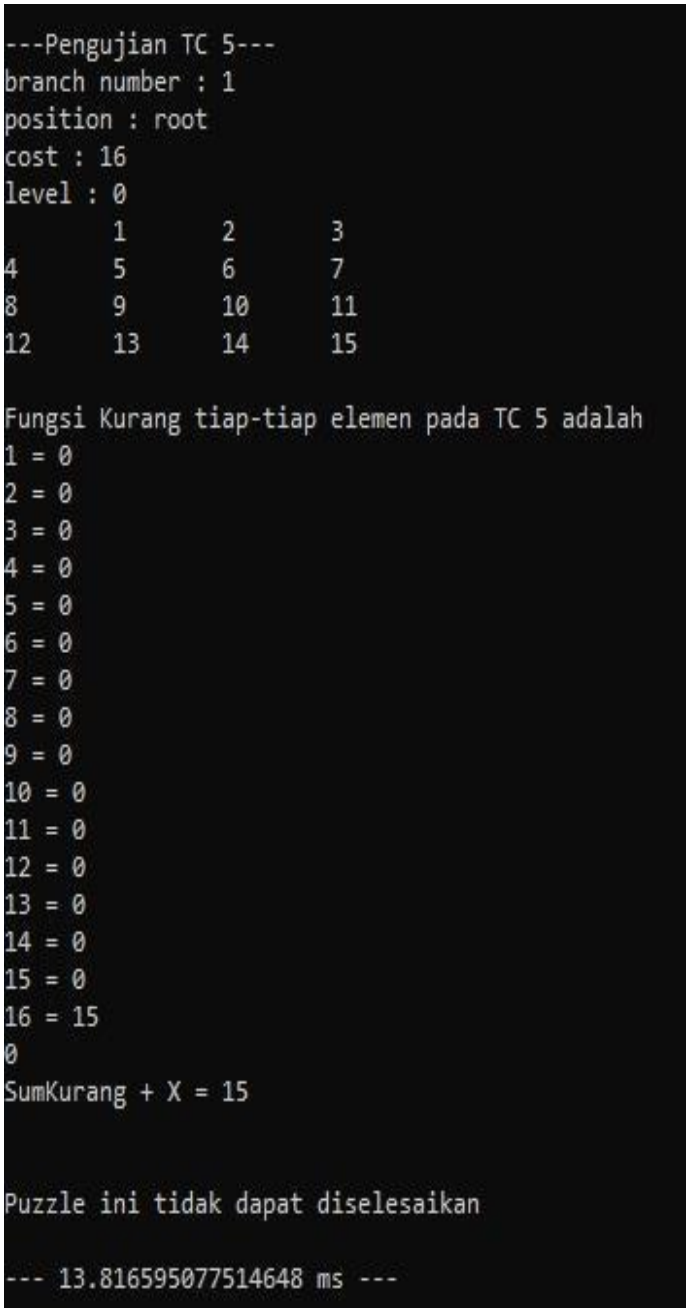
Gambar 2.2 Hasil pengujian pada Test Case 2

3. Hasil Output TC3 (file txt : tclaporan3.txt)

```
---Pengujian TC 3---
branch number : 1
position : root
cost : 4
level : 0
1       2       3       4
5       6       7       8
9       10      15      11
13      14              12

Fungsi Kurang tiap-tiap elemen pada TC 3 adalah
1 = 0
2 = 0
3 = 0
4 = 0
5 = 0
6 = 0
7 = 0
8 = 0
9 = 0
10 = 0
11 = 0
12 = 0
13 = 1
14 = 1
15 = 4
16 = 1
1
SumKurang + X = 8
Puzzle dapat diselesaikan
Untuk menyelesaikan puzzle, Simpul dihidupkan sebanyak 9 buah
jadi, solusinya adalah sebagai berikut:

Matriks ke- 1
branch number : 1
position : root
cost : 4
level : 0
1       2       3       4
5       6       7       8
9       10      15      11
13      14              12

Matriks ke- 2
branch number : 2
position : Up
cost : 4
level : 1
1       2       3       4
5       6       7       8
9       10              11
13      14      15      12

Matriks ke- 3
branch number : 6
position : Right
cost : 4
level : 2
1       2       3       4
5       6       7       8
9       10      11
13      14      15      12

Matriks ke- 4
branch number : 9
position : Down
cost : 4
level : 3
1       2       3       4
5       6       7       8
9       10      11      12
13      14      15

Solusi ditemukan!
--- 27.422666549682617 ms ---
```

Gambar 2.3. Hasil Output untuk Testcase 3

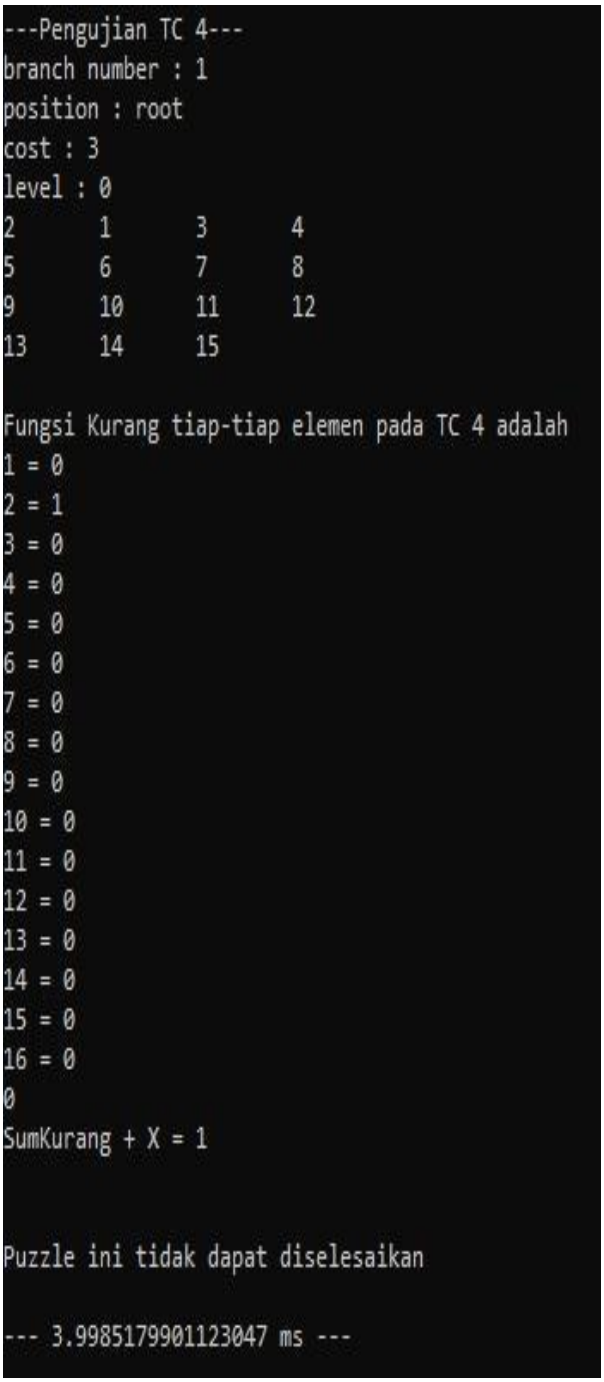
4. **Hasil Output TC4 (file txt : tclaporan4.txt)**



```
---Pengujian TC 5---
branch number : 1
position : root
cost : 16
level : 0
        1       2       3
4       5       6       7
8       9       10      11
12      13      14      15

Fungsi Kurang tiap-tiap elemen pada TC 5 adalah
1 = 0
2 = 0
3 = 0
4 = 0
5 = 0
6 = 0
7 = 0
8 = 0
9 = 0
10 = 0
11 = 0
12 = 0
13 = 0
14 = 0
15 = 0
16 = 15
0
SumKurang + X = 15


Puzzle ini tidak dapat diselesaikan

--- 13.816595077514648 ms ---
```

Gambar 2.4. Hasil Output untuk Test case 4

5. **Kode untuk mengalikan polinom secara brute force**



Gambar 2.5. Hasil Output untuk Test case 5

BAB 3
DATA PENDUKUNG

1. Spesifikasi Komputer

Spesifikasi yang digunakan adalah sebagai berikut

Perangkat Digunakan : Asus A455L
Operating System : Microsoft Windows 10 Pro
Processor : Intel® Core™ i3-5005U CPU @2.00 Ghz (4CPU) ~2Ghz
Memory : 4096 MB RAM

2. Tabel Penilaian

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 *n*	✓	