

LAPORAN TUGAS KECIL 3
IF2211/STRATEGI ALGORITMA
Semester II tahun 2019/2020

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



Oleh:

13518004 Qurrata A'yuni

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 4

BAB I

Algoritma *Branch and Bound*

Algoritma *Branch and Bound* digunakan untuk persoalan optimasi yang meminimalkan atau memaksimalkan suatu fungsi objektif namun tidak melanggar batasan permasalahan. Algoritma ini memanfaatkan konsep *Breadth First Search* serta *least cost search*. Simpul-simpul akan dibangkitkan dengan urutan pembangkitan menggunakan konsep antrian. Setiap simpul diberi nilai *cost*, sehingga urutan pembangkitan simpul tidak lagi berdasarkan urutannya namun berdasarkan nilai *cost* terkecil.

Dalam persoalan *15-Puzzle* ini, ditentukan apakah posisi awal *puzzle* dapat diselesaikan sampai tujuan. Hal ini dapat diimplementasikan dengan fungsi Kurang(i) dan nilai posisi ubin kosong di kondisi awal (X). *Puzzle* dari suatu posisi awal *solvable* bila nilai fungsi Kurang(i) + X bernilai genap. Kurang(i) diperoleh dengan menghitung nilai yang kurang(i) yang posisinya lebih dari kurang(i). X bernilai satu atau nol, didapatkan melalui percobaan. Posisi ubin kosong (i,j) dan i dijumlahkan j kemudian dimod 2, inilah nilai X.

Bila *solvable*, maka urutan penemuan solusi akan dibangkitkan. Setiap simpul akan dibangkitkan berdasarkan prioritasnya pembangkitan. Simpul akar merupakan susunan awal *Puzzle*, kemudian simpul berikutnya akan dibangkitkan dengan *operator* yang *available* dari posisi ubin kosong pada simpul akar. Prioritas operator adalah *up, right, down, left*. Aturan pembangkitan simpul dengan BFS tetapi dioptimasi dengan *least cost search*. *Cost* setiap simpul i adalah jumlah ongkos mencapai simpul i dari akar ditambah dengan ongkos mencapai simpul tujuan dari simpul i. Sehingga simpul solusi yang akan dibangkitkan berdasarkan nilai terkecil dari simpul yang dihasilkan dari satu simpul.

Inisia

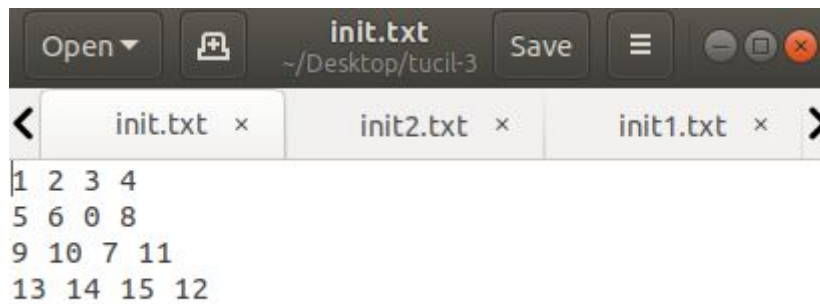
Secara umum berikut algoritma B&B:

1. Masukkan simpul akar ke dalam antrian Q, jika simpul akar adalah simpul solusi, maka solusi ditemukan. Berhenti.
2. Jika antrian Q kosong, tidak ada solusi. Berhenti.
3. Jika antrian Q tidak kosong, pilih antrian Q simpul i yang mempunyai nilai *cost* i paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih satu secara acak.
4. Jika simpul i adalah solusi, berhenti. Bila simpul i bukan simpul solusi, maka bangkitkan semua anak-anaknya. Jika ia tidak mempunyai anak, kembali ke langkah 2.
5. Untuk setiap anak j dari simpul i, hitung *cost* j, masukkan semua anak ke dalam antrian Q.
6. Kembali ke langkah 2.

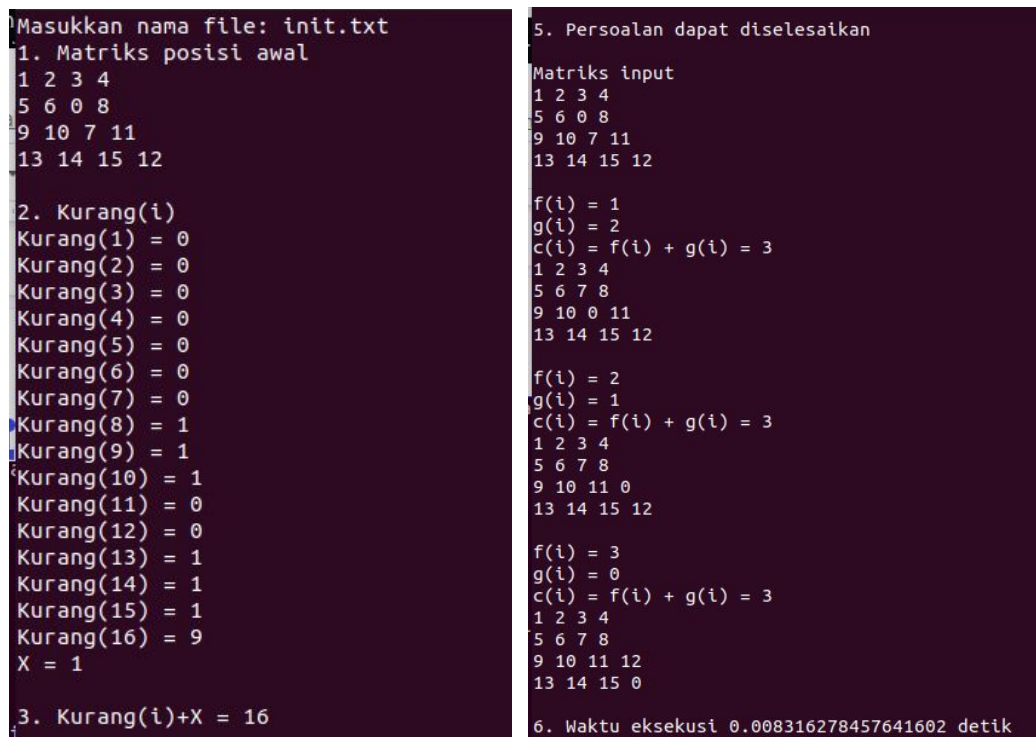
BAB II

Screenshot Input-Output Program

1. Input Pertama

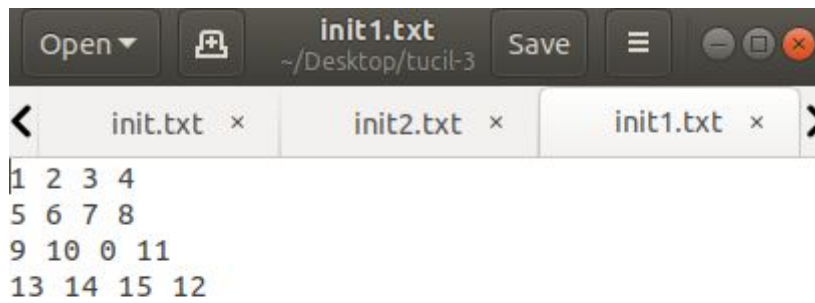


Gambar 1 init.txt

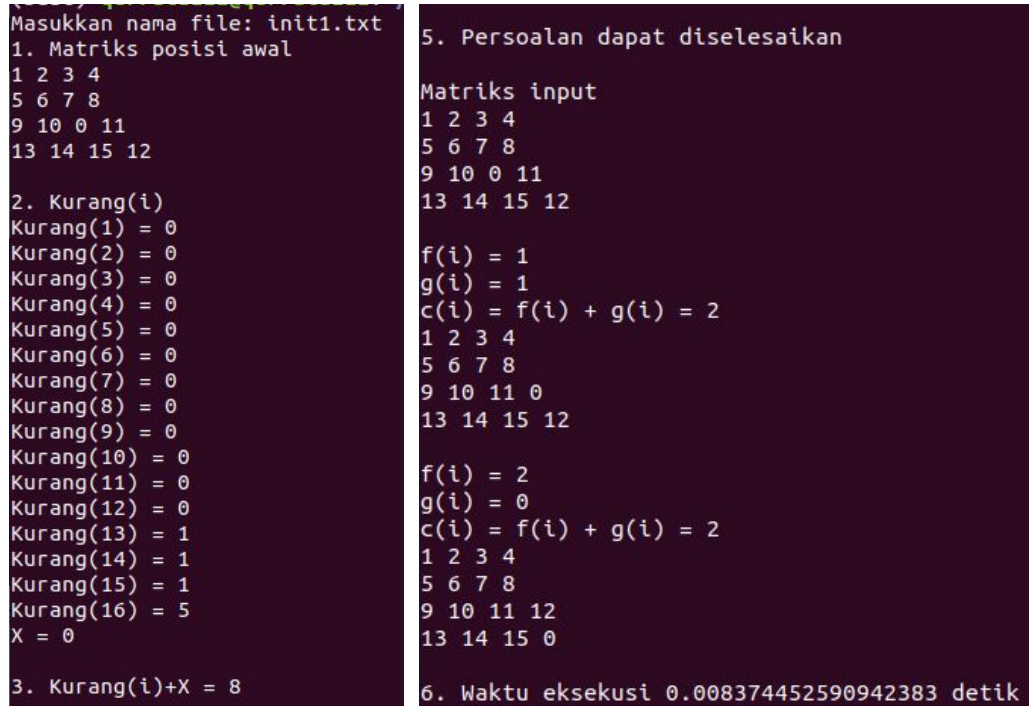


Gambar 2 Output Program

2. Input Kedua

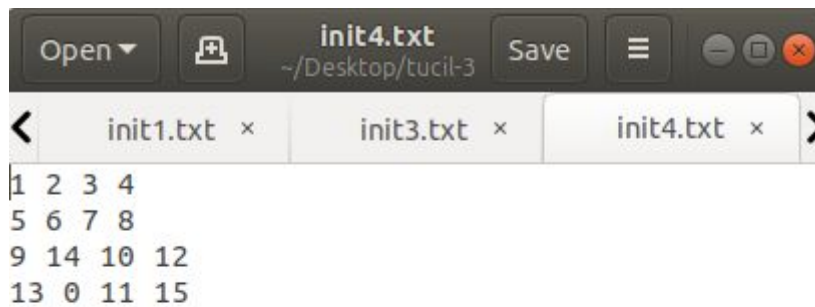


Gambar 3 init1.txt

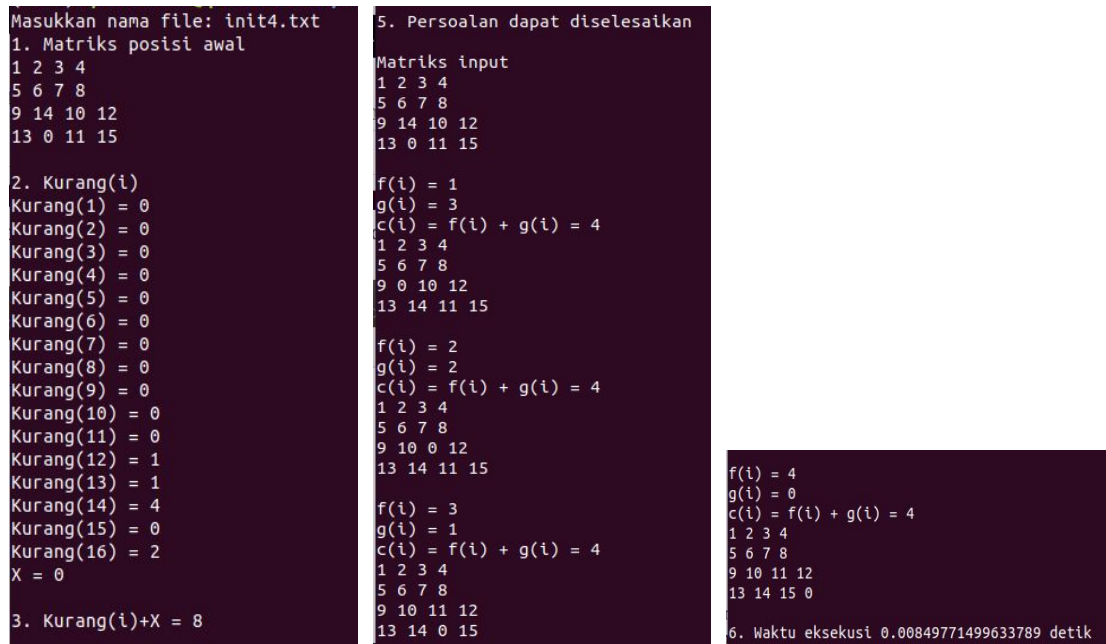


Gambar 4 Output Program

3. Input Ketiga

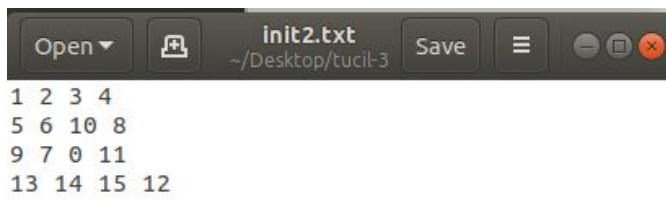


Gambar 5 init4.txt



Gambar 6 Output Program

4. Input Keempat

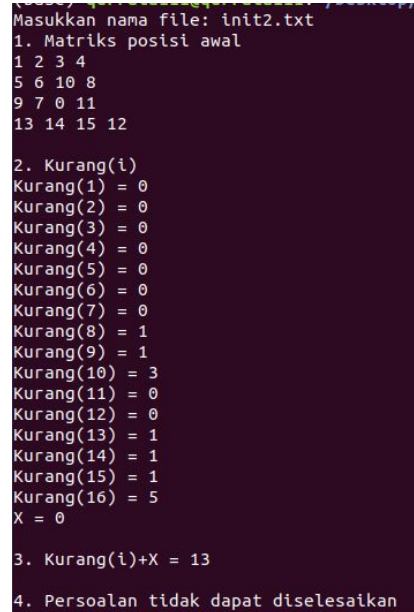


```

1 2 3 4
5 6 10 8
9 7 0 11
13 14 15 12

```

Gambar 7 init2.txt



```

Masukkan nama file: init2.txt
1. Matriks posisi awal
1 2 3 4
5 6 10 8
9 7 0 11
13 14 15 12

2. Kurang(i)
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) = 3
Kurang(11) = 0
Kurang(12) = 0
Kurang(13) = 1
Kurang(14) = 1
Kurang(15) = 1
Kurang(16) = 5
X = 0

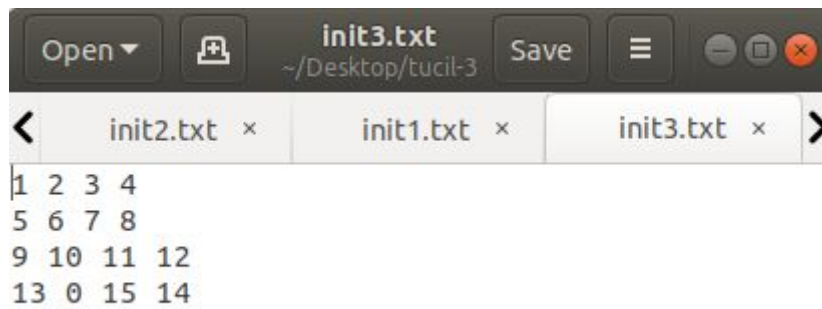
3. Kurang(i)+X = 13

4. Persoalan tidak dapat diselesaikan

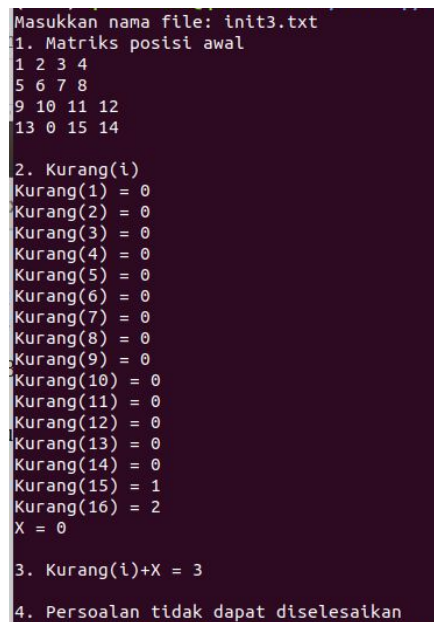
```

Gambar 8 Output Program

5. Input Kelima



Gambar 9 init3.txt



Gambar 10 Output Program

Spesifikasi Personal Computer yang digunakan

CPU : Intel core i5-8250U
 RAM : 4 Gb
 Core : i5
 OS : Linux (Ubuntu 18.04.3 LTS)

Tabel 1 Ceklist Uji Coba Program

Poin	Ya	Tidak
Program berhasil dikompilasi	✓	
Program berhasil running	✓	
Program dapat menerima input dan menuliskan output	✓	
Luaran sudah benar untuk semua n	✓	

REFERENSI

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Branch-&-Bound-\(2018\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Branch-&-Bound-(2018).pdf)

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Latihan-stima-2018.pdf>

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Jawaban-Latihan-2018.pdf>