

# Penerapan Algoritma Branch & Bound dalam Menentukan Rute Terpendek untuk Menuju ke Sebuah Lokasi di Kampus Institut Teknologi Bandung

M. Azwar Adli (13514601)  
Program Studi Sarjana Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
[azwar.adli@gmail.com](mailto:azwar.adli@gmail.com)

**Abstract**—Makalah ini membahas tentang algoritma Branch and Bound serta perannya dalam mencari rute terpendek untuk menuju ke sebuah lokasi di kampus Institut Teknologi Bandung. Algoritma Branch and Bound dapat digunakan untuk menyelesaikan permasalahan optimasi seperti rute terpendek. Implementasi algoritma Branch and Bound dalam makalah ini menerapkan fungsi batasan berupa reduced cost matrix.

**Keywords**—Branch and Bound, rute terpendek, reduced cost matrix

## I. PENDAHULUAN

### A. Latar Belakang

Institut Teknologi Bandung adalah salah satu dari sekian banyak Institusi terbaik di Indonesia yang terletak di kota Bandung, provinsi Jawa Barat, Indonesia. Institut Teknologi Bandung mempunyai luas 286,83 m<sup>2</sup> dengan jumlah mahasiswa lebih kurang 19.000 mahasiswa di tahun ajaran 2014/2015. Institut Teknologi Bandung mempunyai banyak gedung dan fasilitas yang lokasinya tersebar di seluruh lokasi.

Dengan luas 286,83 m<sup>2</sup> dan bangunan yang tersebar di seluruh kampus, tentu saja banyak sekali alternatif jalan yang bisa dilalui untuk menuju suatu lokasi di kampus Institut Teknologi Bandung. Dengan adanya aturan tidak boleh menginjak rerumputan, maka dari itu jalan yang bisa dilalui hanyalah jalan yang sudah disediakan baik berupa jalan setapak maupun jalan beraspal.

Dengan banyaknya alternatif jalan yang bisa dilalui, dibutuhkan rute terpendek untuk mencapai suatu lokasi apabila kita ingin cepat sampai ke suatu lokasi dan tidak membuang – buang waktu.

Latar Belakang penulis mengangkat judul makalah ini yaitu karena sebagai mahasiswa kita harus memanfaatkan waktu sebaik mungkin selama berada di kampus. Waktu yang terbatas memaksa kita supaya efektif selama melakukan mobilisasi antara satu lokasi ke lokasi lain di Institut Teknologi Bandung sehingga kita bisa melanjutkan

perkuliahan di lokasi satu dengan lokasi yang lain dengan tepat waktu.

Sebagai mahasiswa Teknik Informatika, penulis tertarik mengangkat masalah rute terpendek ini sebagai makalah dengan berbagai pendekatan ilmu informatika berupa algoritma – algoritma yang dapat digunakan untuk mengatasi isu ini.

### B. Tujuan

Tujuan penulis membuat makalah ini yaitu:

- Menambah pemahaman mengenai teori – teori pemecahan masalah terutama pemecahan masalah dengan menggunakan algoritma Branch & Bound.
- Menerapkan implementasi algoritma Branch & Bound.
- Memecahkan isu rute terpendek untuk mencapai sebuah lokasi di Institut Teknologi Bandung
- Memenuhi tugas mata kuliah IF2211 Strategi Algoritma.

## II. DASAR TEORI

### A. Rute Terpendek (Shortest Path)

Pencarian rute terpendek dalam graf merupakan salah satu contoh dari sekian banyak persoalan optimasi. Graf yang digunakan dalam persoalan optimasi adalah graf berbobot, yaitu graf yang memiliki bobot atau nilai di setiap sisinya. Bobot dalam sisi graf menyatakan jarak antar entitas yang direpresentasikan dengan jarak antara simpul satu dengan lainnya.

Nilai dari bobot bisa berupa jarak, waktu yang dibutuhkan, ongkos, dll.

Ada beberapa macam persoalan rute terpendek, antara lain:

1. Lintasan terpendek antara dua buah simpul.
2. Lintasan terpendek antara semua pasangan simpul.
3. Lintasan terpendek dari simpul tertentu ke simpul lain.

- Lintasan terpendek antara beberapa buah simpul yang melalui simpul tertentu terlebih dahulu.

## B. Algoritma Branch & Bound

Algoritma Branch & Bound adalah salah satu algoritma yang digunakan untuk persoalan optimasi, yaitu dengan meminimalkan atau memaksimalkan suatu fungsi objektif, yang tidak melanggar batasan (constraints) persoalan.

Algoritma Branch & Bound menggambarkan suatu persoalan sebagai simpul – simpul, dan memecahkan permasalahan dengan memproses simpul – simpul tersebut.

Algoritma Branch & Bound merupakan gabungan dari algoritma Breadth First Search (BFS) dan Least Cost Search. Karakteristik algoritma Branch & Bound adalah sebagai berikut:

- Setiap simpul diberi sebuah nilai cost:  
 $\hat{c}(i)$  = nilai taksiran termurah ke simpul status tujuan yang melalui simpul status  $i$ .  
 $\hat{c}(i)$  menyatakan batas bawah (*lower bound*) dari ongkos pencarian solusi dari status  $i$ .
- Simpul berikutnya yang akan di-ekspan tidak lagi berdasarkan urutan pembangkitannya, tetapi simpul yang memiliki *cost* yang paling kecil (Least Cost Search) – pada kasus minimalisasi.

Sesuai dengan namanya yaitu algoritma Branch & Bound, algoritma ini mengekskan simpul – simpul solusi sehingga didapatkan seluruh percabangan dari simpul solusi (branch), lalu matikan/batasi simpul yang sekiranya tidak menuju ke simpul solusi (bound).

Terdapat dua informasi tambahan yang diperlukan dalam penyelesaian persoalan dengan Branch & Bound:

- Untuk setiap simpul pada pohon ruang status, diperlukan suatu cara penentuan batas (bound) nilai terbaik fungsi objektif pada setiap solusi yang mungkin, dengan menambahkan komponen pada solusi sementara yang direpresentasikan simpul.
- Nilai dari solusi terbaik sejauh proses terakhir.

Berikut adalah algoritma global Branch & Bound:

- Masukkan simpul akar ke dalam antrian Q. Jika simpul akar adalah simpul solusi (*goal node*), maka solusi telah ditemukan. Berhenti.
- Jika Q kosong, tidak ada solusi. Berhenti.
- Jika Q tidak kosong, pilih dari antrian Q simpul  $i$  yang mempunyai  $\hat{c}(i)$  paling kecil. Jika terdapat beberapa simpul  $i$  yang memenuhi, pilih satu secara sembarang.
- Jika simpul  $i$  adalah simpul solusi, berarti solusi sudah ditemukan, berhenti. Jika simpul  $i$  bukan simpul solusi, maka bangkitkan semua anak – anaknya. Jika  $i$  tidak mempunyai anak, kembali ke langkah 2.
- Untuk setiap anak  $j$  dari simpul  $i$ , hitung  $\hat{c}(j)$ , dan masukkan semua anak – anak tersebut ke dalam Q.

- Kembali ke langkah 2.

Algoritma Branch & Bound memiliki fungsi pembatas yang bertujuan untuk memangkas jalur yang dianggap tidak lagi mengarah ke solusi.

Kriteria pemangkasan secara umum:

- Nilai simpul tidak lebih baik dari nilai terbaik sejauh ini.
- Simpul tidak merepresentasikan solusi yang ‘feasible’ karena ada batasan yang dilanggar.
- Solusi yang ‘feasible’ pada simpul tersebut hanya terdiri atas satu titik → tidak ada pilihan lain.

## C. Graf

Graf digunakan untuk merepresentasikan objek – objek diskrit dan hubungan antara objek – objek tersebut. Graf terdiri dari himpunan tidak-kosong dari simpul – simpul dan himpunan sisi yang menghubungkan sepasang simpul.

Jenis – jenis graf:

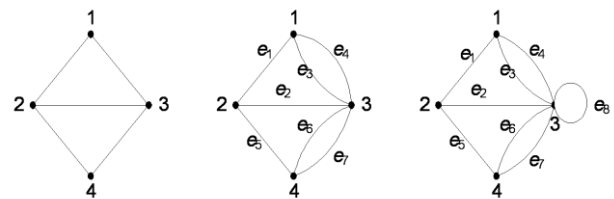
- Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf

Graf sederhana (*simple graph*)

→ Graf yang tidak mengandung gelang maupun sisi-ganda.

Graf tak-sederhana (*unsimple graph*)

→ Graf yang mengandung sisi ganda atau gelang



Gambar 1 - Graf Sederhana.

Sumber: <http://sha-essa.blogspot.com/2011/12/teori-graph-21.html>

Diakses pada: 03 Mei 2015 pukul 10:48 WIB

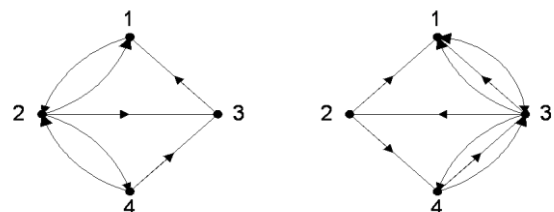
- Berdasarkan orientasi arah pada sisi

Graf tak-berarah (*undirected graph*)

→ Graf yang sisinya tidak mempunyai orientasi arah

Grad berarah (*directed graph*)

→ Graf yang setiap sisinya diberikan orientasi arah

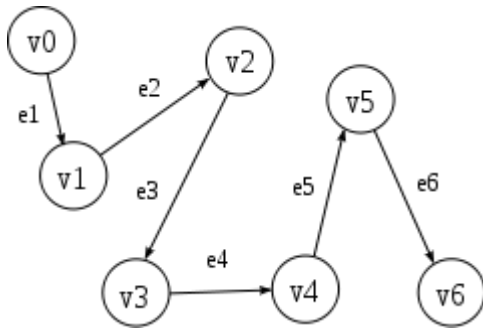


Gambar 2 - Graf Berarah

Sumber: <http://sha-essa.blogspot.com/2011/12/teori-graph-21.html>

Diakses pada: 03 Mei 2015 pukul 12:00 WIB

Aplikasi graf dalam kehidupan sehari – hari sangat banyak. Sebagai contoh salah satunya dapat digunakan untuk menggambarkan alur sebuah proses, dimana simpul – simpul yang terhubung menggambarkan satu proses dengan proses lain yang saling terhubung.

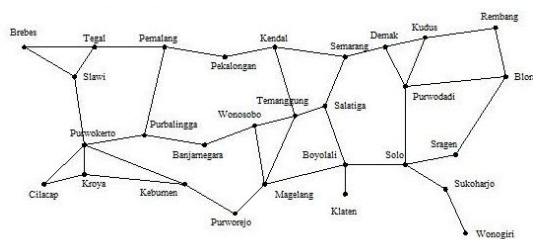


Gambar 3 - Graf Alur

Sumber: [http://id.wikipedia.org/wiki/Teori\\_graf](http://id.wikipedia.org/wiki/Teori_graf)

Diakses pada: 03 Mei 2015 pukul 16:01 WIB

Contoh lain penerapan graf yaitu pemetaan sebuah wilayah dimana graf merepresentasikan keterhubungan antar wilayah. Di dalam pemetaan tersebut, dapat dilihat jarak antar masing – masing wilayah dan wilayah mana yang saling terhubung secara langsung.



Gambar 4 - Graf Wilayah

Sumber:

<https://syarifkapita.wordpress.com/2014/05/02/contoh-teori-graf/>

Diakses pada: 03 Mei 2015 pukul 12:09 WIB

#### D. Reduced Cost Matrix

Reduced Cost Matrix dibutuhkan untuk menyelesaikan persoalan ini, yaitu mendapatkan jarak terpendek dari suatu tempat ke tempat yang lain. Reduce cost matrix adalah matrix yang memiliki minimal 1 buah angka 0 pada setiap baris dan kolomnya.

Cara mendapatkan angka 0 pada setiap baris dan kolomnya yaitu dengan mengurangi tiap angka pada baris dengan angka yang paling kecil per barisnya, lalu lakukan pengecekan pada tiap kolom, apabila di tiap kolom sudah terdapat angka 0 maka selesai. Namun apabila belum, maka lakukan hal yang sama seperti yang dilakukan pada

tiap barisnya, yaitu dengan mengurangi tiap angka pada kolom dengan angka paling kecil per kolomnya.

Setelah itu jumlahkan angka pengurang dari tiap baris dan kolomnya sebagai cost awal matriks tersebut. Dari matriks akar tersebut, bangkitkan anak – anak simpulnya.

Dapatkan matriks tereduksi dari setiap anak simpul yang dibangkitkan dengan cara sebagai berikut, misalkan i adalah simpul asal dan j adalah simpul tujuan,

1. Semua elemen matriks pada baris i dan kolom j diganti dengan  $\infty$

2. Elemen matriks pada (j,1) juga diganti dengan  $\infty$

3. Lakukan kembali reduce cost matrix

Hitung kembali cost masing – masing simpul dengan rumus sebagai berikut:

$$\hat{c}(S) = \hat{c}(R) + A(i,j) + r$$

keterangan:

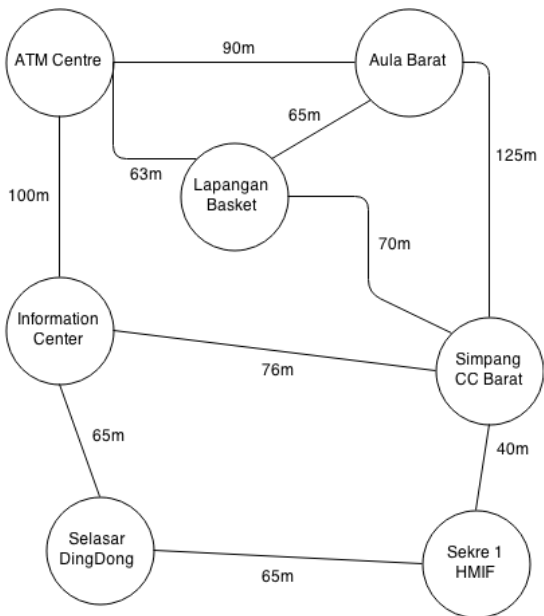
- $\hat{c}(S)$  = bobot perjalanan minimum yang melalui simpul S (simpul di pohon ruang status)
- $\hat{c}(R)$  = bobot perjalanan minimum yang melalui simpul R, yang dalam hal ini R adalah orang tua dari S
- $A(i,j)$  = bobot sisi (i,j) pada graf G yang berkorespondensi dengan sisi (R,S) pada pohon ruang status
- $r$  = jumlah semua pengurang dari tiap baris dan kolom pada simpul S

Setelah didapatkan cost dari seluruh simpul anaknya, maka pilih satu simpul dengan cost terkecil untuk kemudian dibangkitkan kembali semua simpul anak dari simpul terpilih tersebut. Apabila simpul terpilih tersebut adalah simpul solusi maka anak – anak dari simpul tersebut tidak perlu dibangkitkan. Namun jika simpul terpilih tersebut bukan merupakan simpul solusi, maka bangkitkan seluruh anak dari simpul tersebut.

### III. PENERAPAN ALGORITMA BRANCH & BOUND UNTUK MENCARI RUTE TERPENDEK

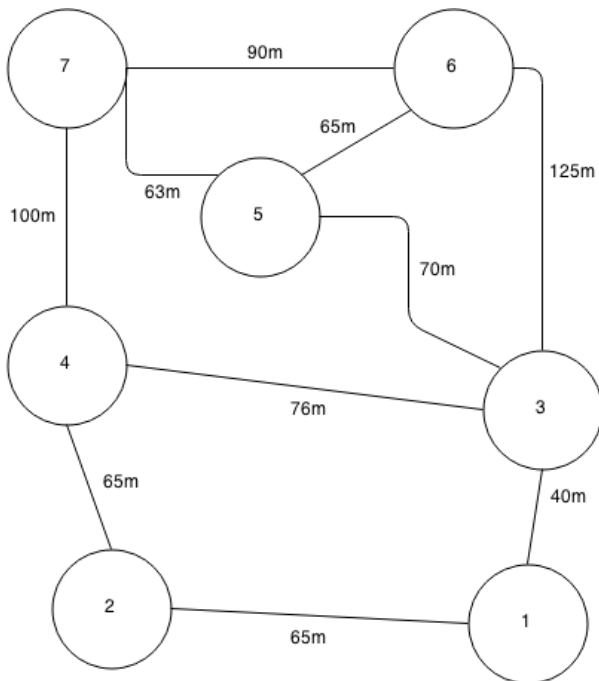
Dalam pencarian rute terpendek dengan Algoritma Branch and Bound, algoritma ini memanfaatkan graf sebagai media pengolahan algoritmanya.

Sebagai contoh implementasi, penulis berikan gambar ilustrasi graf dari labtek V sampai ATM Centre sebagai berikut :



Gambar 5 - Graf Ilustrasi

Untuk memudahkan perhitungan algoritma Branch & Bound maka nama dalam graf diubah menjadi nomor sebagai berikut:



Gambar 6 - Graf Ilustrasi

Simpul tujuan dari graf tersebut adalah simpul 7 dengan simpul awal simpul 1.

$$R = \begin{bmatrix} \infty & 65 & 40 & \infty & \infty & \infty & \infty \\ 65 & \infty & \infty & 65 & \infty & \infty & \infty \\ 40 & \infty & \infty & 76 & 70 & 125 & \infty \\ \infty & 65 & 76 & \infty & \infty & \infty & 100 \\ \infty & \infty & 70 & \infty & \infty & 65 & 63 \\ \infty & \infty & 125 & \infty & 65 & \infty & 90 \\ \infty & \infty & \infty & 100 & 63 & 90 & \infty \end{bmatrix}$$

Di atas adalah matriks ketetanggaan dari graf ilustrasi. Dari graf di atas, dicari matriks reduksinya, yaitu:

$$A = \begin{bmatrix} \infty & 25 & 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty & \infty & \infty \\ 0 & \infty & \infty & 36 & 30 & 83 & \infty \\ \infty & 0 & 11 & \infty & \infty & \infty & 35 \\ \infty & \infty & 7 & \infty & \infty & 0 & 0 \\ \infty & \infty & 60 & \infty & 0 & \infty & 25 \\ \infty & \infty & \infty & 37 & 0 & 25 & \infty \end{bmatrix}$$

$r = 40 + 65 + 40 + 65 + 63 + 65 + 63 + 2 = 403$ . Jadi cost untuk matriks akar adalah 403.

Kemudian, anak – anak simpul dibangkitkan

1. Simpul 2, lintasan 1,2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty & \infty & \infty \\ 0 & \infty & \infty & 36 & 30 & 83 & \infty \\ \infty & \infty & 0 & \infty & \infty & \infty & 24 \\ \infty & \infty & 7 & \infty & \infty & 0 & 0 \\ \infty & \infty & 60 & \infty & 0 & \infty & 25 \\ \infty & \infty & \infty & 37 & 0 & 25 & \infty \end{bmatrix}$$

$r = 11$

$$\hat{c}(2) = 403 + 25 + 11 = 439$$

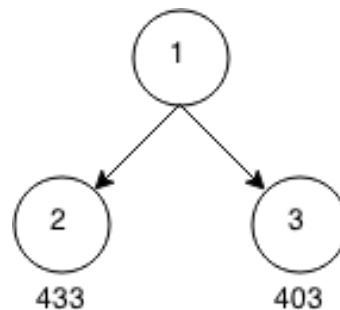
2. Simpul 3, lintasan 1,3

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & 36 & 30 & 83 & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty & 35 \\ \infty & \infty & \infty & \infty & \infty & 0 & 0 \\ \infty & \infty & \infty & \infty & 0 & \infty & 25 \\ \infty & \infty & \infty & 37 & 0 & 25 & \infty \end{bmatrix}$$

$r = 0$

$$\hat{c}(2) = 403 + 0 + 0 = 403$$

dari matriks – matriks di atas jika dibentuk menjadi pohon maka akan menghasilkan pohon sebagai berikut



Dari semua simpul anak yang telah dibangkitkan, yang memiliki cost terkecil adalah simpul 3. Karena simpul 3 merupakan simpul dengan cost terkecil namun bukan merupakan simpul solusi, maka semua anak dari simpul 3 dibangkitkan, dan karena simpul 2 tidak mengarah ke solusi, simpul 2 dibound.

1. Simpul 4, lintasan 3,4

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty & 35 \\ \infty & \infty & \infty & \infty & \infty & 0 & 0 \\ \infty & \infty & \infty & \infty & 0 & \infty & 25 \\ \infty & \infty & \infty & \infty & 0 & 25 & \infty \end{bmatrix}$$

$r = 0$

$\hat{c}(4) = 403 + 36 + 0 = 439$

2. Simpul 5, lintasan 3,5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty & 35 \\ \infty & \infty & \infty & \infty & \infty & 0 & 0 \\ \infty & \infty & \infty & \infty & \infty & 0 & 0 \\ \infty & \infty & \infty & 12 & \infty & 0 & \infty \end{bmatrix}$$

$r = 25 + 25 = 50$

$\hat{c}(5) = 403 + 30 + 50 = 483$

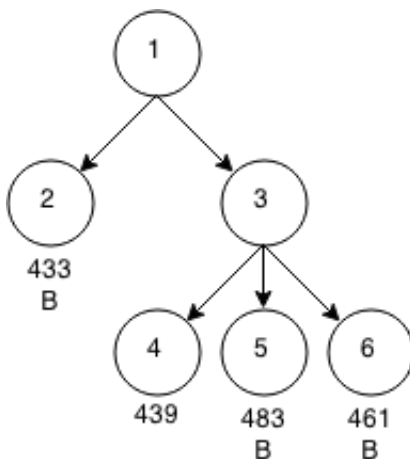
3. Simpul 6, lintasan 3,6

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty & \infty & 35 \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & 0 & \infty & 25 \\ \infty & \infty & \infty & 37 & 0 & \infty & \infty \end{bmatrix}$$

$r = 0$

$\hat{c}(6) = 403 + 58 + 0 = 461$

Dari matriks – matriks di atas jika dibentuk menjadi pohon maka akan menghasilkan pohon sebagai berikut



Dari semua simpul anak yang telah dibangkitkan, yang memiliki cost terkecil adalah simpul 4. Karena simpul 4 merupakan simpul dengan cost terkecil namun bukan merupakan simpul solusi, maka semua anak dari simpul 4 dibangkitkan, dan karena simpul 5 dan 6 tidak mengarah ke solusi, simpul 5 dan 6 dibound.

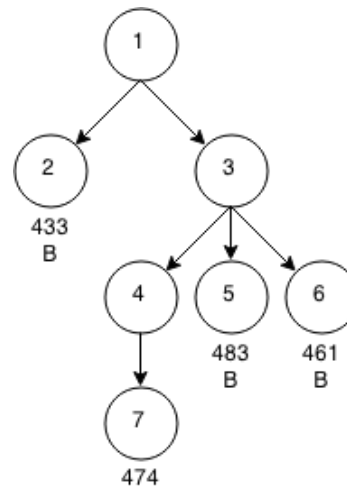
1. Simpul 7, lintasan 4,7

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 & 25 & \infty \end{bmatrix}$$

$r = 0$

$\hat{c}(7) = 439 + 35 + 0 = 474$

dari matriks di atas jika dibentuk menjadi pohon maka akan menghasilkan pohon sebagai berikut



Dari semua simpul anak yang telah dibangkitkan yang memiliki cost terkecil adalah simpul 7. Karena simpul 7 merupakan simpul solusi, maka rute lintasan terpendek yang didapatkan adalah 1 – 3 – 4 – 7, yaitu : Sekre 1 HMIF – Simpang CC Barat – Information Center – ATM Center dengan jarak tempuh 216m.

#### IV. KESIMPULAN

Algoritma Branch and Bound bisa diterapkan untuk mendapatkan rute terpendek dari suatu graf. Namun hasil yang didapatkan bukan hasil terbaik karena apabila dilihat secara kasat mata dari graf, maka rute minimal yang seharusnya didapatkan adalah rute 1 – 3 – 5 – 7, yaitu : Sekre 1 HMIF – Simpang CC Barat – Lapangan Basket – ATM Center dengan jarak tempuh 173m.

Walaupun penerapan algoritma Branch and Bound tidak mendapatkan solusi terbaik, namun dengan algoritma Branch and Bound persoalan optimasi dalam mencari rute terpendek mendapatkan nilai hantaran yang cenderung lebih baik.

## V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga makalah ini dapat selesai dengan lancar. Terima kasih juga penulis ucapkan kepada kedua orang tua penulis yang doa dan kasih sayangnya selalu tercurahkan hingga saat ini.

Tak lupa penulis juga mengucapkan terima kasih kepada dosen pengampu mata kuliah IF2211 Strategi Algoritma yaitu Bapak Rinaldi Munir dan Ibu Nur Ulfa Maulidevi yang telah mencurahkan ilmunya tanpa pamrih. Juga kepada pihak – pihak lain yang tidak bisa penulis sebutkan satu persatu, terima kasih atas segala dukungan yang telah diberikan kepada penulis.

## REFERENSI

- [1] Munir, Rinaldi. Strategi Algoritma. Bandung. Informatika Bandung, 2009.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 4 Mei 2015



M. Azwar Adli (13514601)