

# PENERAPAN ALGORITMA *BRANCH AND BOUND* DALAM MENENTUKAN RUTE TERPENDEK UNTUK PERJALANAN ANTARKOTA DI JAWA BARAT

M. Pasca Nugraha

Sekolah Teknik Elektro dan Informatika  
Program Studi Teknik Informatika  
Institut Teknologi Bandung  
Jalan Ganesha 10 Bandung  
e-mail: [if17033@students.if.itb.ac.id](mailto:if17033@students.if.itb.ac.id), [mpascanug@yahoo.co.id](mailto:mpascanug@yahoo.co.id)

## ABSTRAK

Pada saat ini, dimana kebutuhan manusia semakin banyak dan beragam, sangat dibutuhkan efektivitas dan efisiensi dalam berbagai kegiatan manusia, termasuk dalam hal memilih jalur atau rute perjalanan. Rute yang lebih pendek pada umumnya akan menghasilkan waktu yang lebih singkat. Tidak jarang seseorang harus mengunjungi beberapa kota untuk keperluan tertentu, baik itu urusan bisnis, pekerjaan, atau sekedar wisata. Oleh karena itu diperlukan suatu cara untuk menentukan rute terpendek agar perjalanan menjadi lebih efektif dan efisien. Masalah ini dapat dikategorikan sebagai masalah TSP (*Travelling Salesperson Problem*).

Algoritma *Branch and Bound* merupakan metode pencarian di dalam ruang solusi secara sistematis. Pada algoritma ini dibangun pohon dengan skema BFS. Algoritma *Branch and Bound* ini cukup baik untuk memberikan solusi optimal pada masalah TSP, termasuk pada pemilihan rute perjalanan.

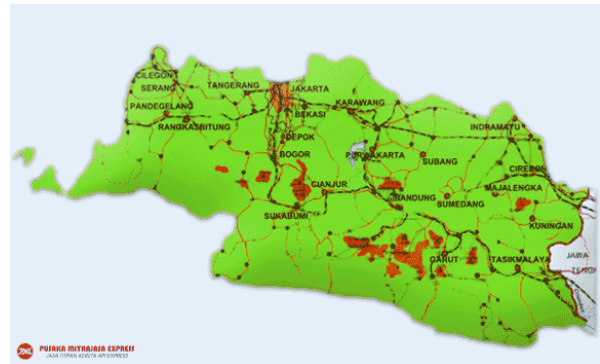
Dalam makalah ini akan dibahas sedikit mengenai pemilihan rute perjalanan antar-beberapa-kota di Propinsi Jawa Barat. Karena terlalu banyaknya kota di Jawa Barat, penulis hanya membatasi pada beberapa kota tertentu saja yang berada di tiap penjuru Propinsi Jawa Barat.

**Kata kunci:** algoritma *Branch and Bound*, rute terpendek, Propinsi Jawa Barat, TSP.

## 1. PENDAHULUAN

Propinsi Jawa Barat merupakan propinsi dengan jumlah penduduk terbanyak di Indonesia. Luas daerahnya cukup besar, yaitu sekitar 35 ribu kilometer persegi. Wilayah ini terdiri atas 17 kabupaten dan 9 kota yang cukup tersebar. Oleh karena itu, jarak antarkota di propinsi ini cukup

beragam. Di bawah ini adalah peta sederhana dari Propinsi Jawa Barat



Gambar 1. Propinsi Jawa Barat

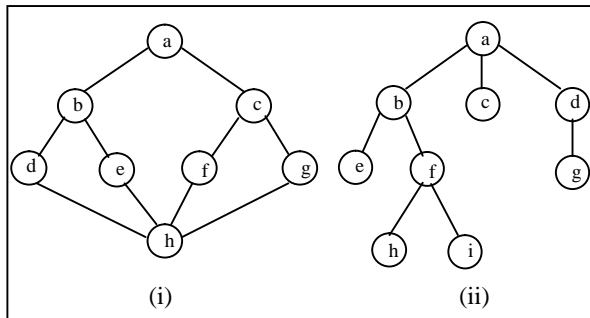
Permasalahan yang akan penulis angkat di sini adalah bagaimana seseorang dapat mengunjungi kota-kota di propinsi ini secara efektif dan efisien, dalam arti rute yang dipilih merupakan rute minimum. Tidak jarang seseorang harus berkunjung ke beberapa kota untuk berbagai keperluan, baik itu keperluan kerja, bisnis, atau sekedar hiburan. Oleh karena itu pemilihan rute yang tepat penulis rasa merupakan sesuatu yang penting, terutama pada masa dimana kebutuhan manusia semakin banyak seperti sekarang ini.

## 2. METODE

Dalam mencari solusi sebuah permasalahan, alangkah baiknya jika digunakan metode tertentu yang sistematis agar didapatkan solusi yang optimum. Pada permasalahan pencarian rute terpendek ini, penulis menggunakan algoritma *Branch and Bound*. Algoritma ini menggunakan skema BFS.

### 2.1 Algoritma Pencarian Melebar (BFS)

Algoritma pencarian melebar atau *Breadth First Search* merupakan algoritma yang diterapkan pada graf. Misalkan kita mempunyai graf G yang mempunyai n buah simpul. Kita akan melakukan traversal di dalam graf, dan misalkan traversal dimulai dari simpul v. Algoritma BFS adalah sebagai berikut : Kunjungi simpul v, kemudian semua simpul yang bertetangga dengan simpul v dikunjungi terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada aras d dikunjungi terlebih dahulu sebelum simpul-simpul pada aras d+1. Sebagai contoh dapat dilihat pada graf di bawah ini :



Gambar 2. Dua buah graf yang dikunjungi

Pada gambar graf (i), jika graf dikunjungi mulai dari simpul a, maka urutan simpul yang dikunjungi adalah a – b – c – d – e – f – g – h.

Pada graf atau pohon berakar (ii), dan graf dikunjungi dari simpul a, maka urutan simpul yang dikunjungi adalah a – b – c – d – e – f – g – h.

### 2.2 Algoritma Branch and Bound

Algoritma *Branch and Bound* merupakan metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi diorganisasikan ke dalam pohon ruang status. Pohon ruang status tersebut dibangun dengan skema BFS. 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 diantara simpul-simpul hidup lainnya. Sedangkan simpul lainnya dimatikan. Nilai ongkos pada simpul i dinyatakan dengan :

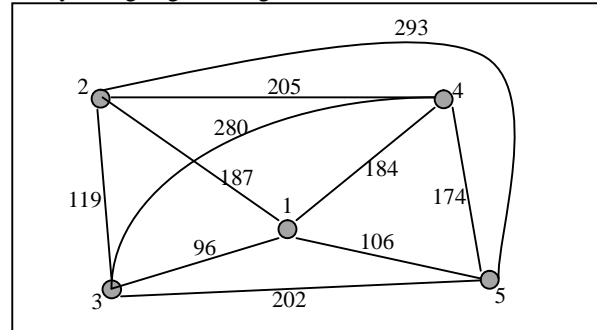
$$\hat{c}(i) = \text{nilai taksiran lintasan termurah dari } i \text{ ke tujuan}$$

### 3. PENERAPAN ALGORITMA BRANCH AND BOUND PADA PERSOALAN

Pada pembahasan ini, penulis hanya akan mengambil sampel 5 buah kota besar, 4 diantaranya yang ada di

Propinsi Jawa Barat, sedangkan 1 lagi adalah Kota Jakarta yang penulis sengaja masukkan karena merupakan pusat kota yang sering dikunjungi. Keempat kota tersebut yaitu Bandung, Tasikmalaya, Sukabumi, dan Indramayu. Bandung dipilih karena merupakan ibukota sekaligus pusat kota, disamping lokasinya yang strategis di tengah-tengah Propinsi Jawa Barat. Sedangkan Tasikmalaya, Sukabumi, dan Indramayu dipilih karena lokasinya yang berada di tiga penjuru propinsi ini.

Secara kasar, kelima kota di atas dapat digambarkan posisinya dengan graf sebagai berikut :



Gambar 3. Graf yang menggambarkan posisi kota

dengan simpul 1 adalah Kota Bandung, simpul 2 adalah Kota Jakarta, simpul 3 adalah Kota Sukabumi, simpul 4 adalah Kabupaten Indramayu, dan simpul 5 adalah Kota Tasikmalaya. Nilai pada sisi adalah *cost* yang menyatakan jarak antar kota.

Misalkan kita akan menentukan rute terpendek untuk mengunjungi setiap simpul pada graf, dimulai dari simpul 1 (Bandung) hingga kembali lagi ke simpul tersebut.

Secara logika, kita dapat menemukan rute terpendek untuk mengunjungi setiap simpul dengan cara mendaftarkan semua kemungkinan yang bisa digunakan (cara *exhaustive search*). Namun, hal ini sangat tidak efisien karena itu berarti kita harus mencoba banyak kemungkinan. Misalnya saja untuk  $\sum$  simpul = 5 (seperti gambar di atas), maka akan terdapat 120 kemungkinan yang harus kita cari. Oleh karena itu, algoritma *Branch and Bound* menawarkan solusi yang lebih praktis untuk menemukan rute terpendek.

Pertama, kita akan menggambarkan graf di atas menjadi sebuah matriks M berukuran 5x5 dimana elemen  $M_{ij}$  adalah jarak dari i ke j, sedangkan i dan j adalah simpul pada graf.

$$M = \begin{bmatrix} \infty & 187 & 96 & 184 & 106 \\ 187 & \infty & 119 & 205 & 293 \\ 96 & 119 & \infty & 280 & 202 \\ 184 & 205 & 280 & \infty & 174 \\ 106 & 184 & 202 & 174 & \infty \end{bmatrix}$$

Matriks di atas merupakan matriks simetris karena jarak I ke j sama dengan jarak dari j ke i.

Selanjutnya kita akan mereduksi matriks di atas agar lebih sederhana. Reduksi dilakukan dengan cara mengurangi seluruh elemen pada baris atau kolom tertentu sehingga terdapat nilai 0 pada baris atau kolom tersebut.

Lakukan reduksi baris :

$$\begin{bmatrix} \infty & 91 & 0 & 88 & 10 \\ 68 & \infty & 0 & 86 & 174 \\ 0 & 23 & \infty & 184 & 106 \\ 10 & 31 & 106 & \infty & 0 \\ 0 & 187 & 96 & 0 & \infty \end{bmatrix}$$

Matriks di atas dihasilkan dari pengurangan tiap baris dengan nilai terkecil pada elemen baris tersebut. Baris-1 dikurangi 96, baris-2 dikurangi 119, baris-3 dikurangi 96, baris-4 dikurangi 174, dan baris-5 dikurangi 106.

Lakukan reduksi kolom :

$$\begin{bmatrix} \infty & 68 & 0 & 20 & 10 \\ 68 & \infty & 0 & 18 & 174 \\ 0 & 0 & \infty & 116 & 106 \\ 10 & 8 & 106 & \infty & 0 \\ 0 & 164 & 96 & 0 & \infty \end{bmatrix}$$

Matriks di atas dihasilkan dari pengurangan seluruh elemen pada kolom-2 dengan 23.

Selanjutnya, proses reduksi ini akan menghasilkan nilai batas simpul akar atau  $\hat{c}(root)$ , yang didapat dari penjumlahan semua elemen pengurang tadi. Jadi,  $\hat{c}(root) = 96 + 119 + 96 + 174 + 106 + 23 + 68 = 682$ . Disini berarti telah dibangkitkan pohon ruang status yang baru berisi satu buah simpul dengan bobot 682.

$$\textcircled{1} \\ 682$$

Selanjutnya, misal A adalah matriks tereduksi untuk simpul R. Misalkan S adalah anak dari simpul R sehingga sisi (R,S) pada pohon ruang status berkorespondensi dengan sisi (i,j). Lakukan langkah-langkah berikut :

- Ubah semua nilai pada baris I dan kolom j menjadi  $\infty$ .
- Ubah A (j,1) menjadi  $\infty$
- Reduksi kembali matriks A

Reduksi matriks A akan menghasilkan matriks lain (misal, B) dan fungsi pembatas. Secara umum, persamaan fungsi pembatas adalah:

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

yang dalam hal ini,

$\hat{c}(S)$  = 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 orangtua 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 pada proses memperoleh matriks tereduksi untuk simpul S.

Selanjutnya dihitung simpul-simpul lain pada pohon ruang status sebagai berikut :

1. Simpul 2; lintasan 1,2 (Bandung-Jakarta)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 18 & 174 \\ 0 & \infty & \infty & 116 & 106 \\ 10 & \infty & 106 & \infty & 0 \\ 0 & \infty & 96 & 0 & \infty \end{bmatrix} = B$$

$$\hat{c}(2) = 682 + 68 + 0 = 750$$

2. Simpul 3; lintasan 1,3 (Bandung-Sukabumi)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & \infty & 18 & 174 \\ \infty & 0 & \infty & 116 & 106 \\ 10 & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & 156 \\ \infty & 0 & \infty & 116 & 106 \\ 10 & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & 0 & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan baris ke 2 oleh 18.

$$\hat{c}(3) = 682 + 0 + 18 = 700$$

3. Simpul 4; lintasan 1,4 (Bandung-Indramayu)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & \infty & 174 \\ 0 & 0 & \infty & \infty & 106 \\ \infty & 8 & 106 & \infty & 0 \\ 0 & 164 & 96 & \infty & \infty \end{bmatrix}$$

$$\hat{c}(4) = 682 + 20 + 0 = 702$$

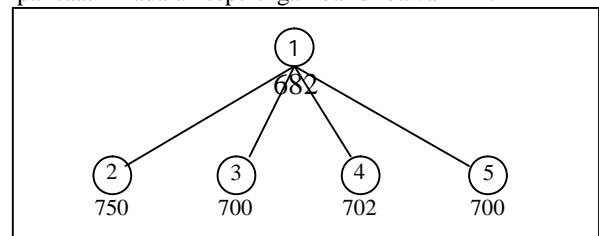
4. Simpul 5; lintasan 1,5 (Bandung-Tasikmalaya)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 18 & \infty \\ 0 & 0 & \infty & 116 & \infty \\ 10 & 8 & 106 & \infty & \infty \\ \infty & 164 & 96 & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 18 & \infty \\ 0 & 0 & \infty & 116 & \infty \\ 2 & 2 & 98 & \infty & \infty \\ \infty & 164 & 96 & 0 & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan baris ke 4 oleh 8.

$$\hat{c}(5) = 682 + 10 + 8 = 700$$

Pohon status yang terbentuk dari proses reduksi di atas sampai saat ini adalah seperti gambar di bawah ini :



Gambar 4. Pohon status yang terbentuk

Dengan penomoran simpul sama seperti sebelumnya, yaitu : (1)Bandung, (2)Jakarta, (3)Sukabumi, (4)Indramayu, dan (5)Tasikmalaya.

Selanjutnya, pilih simpul yang memiliki nilai batas terkecil, dalam hal ini terdapat 2 simpul yaitu simpul 3 dan simpul 5. Pertama-tama, kita ekspansi simpul 3 terlebih dahulu :

5. Simpul 6; lintasan 1,3,2 (Bandung-Sukabumi-Jakarta)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & 156 \\ \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & 0 & \infty \end{bmatrix}$$

$$\hat{c}(6) = 700 + 0 + 0 = 700$$

6. Simpul 7; lintasan 1,3,4 (Bandung-Sukabui-Indramayu)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & \infty & 156 \\ \infty & \infty & \infty & \infty & \infty \\ \infty & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & 106 \\ \infty & \infty & \infty & \infty & \infty \\ \infty & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & \infty & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan baris ke 2 oleh 50.

$$\hat{c}(7) = 700 + 116 + 50 = 866$$

7. Simpul 8; lintasan 1,3,5 (Bandung-Sukabumi-Tasik)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 10 & 8 & \infty & \infty & \infty \\ \infty & 164 & \infty & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 2 & 0 & \infty & \infty & \infty \\ \infty & 164 & \infty & 0 & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan baris ke 4 oleh 8.

$$\hat{c}(8) = 700 + 106 + 8 = 814$$

Selanjutnya kita ekspansi simpul 5 :

8. Simpul 9; lintasan 1,5,2 (Bandung-Tasik-Jakarta)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 18 & \infty \\ 0 & \infty & \infty & 116 & \infty \\ 2 & \infty & 98 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 0 & \infty \\ 0 & \infty & \infty & 98 & \infty \\ 2 & \infty & 98 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan kolom ke 4 oleh 18.

$$\hat{c}(9) = 700 + 164 + 18 = 882$$

9. Simpul 10; lintasan 1,5,3 (Bandung-Tasik-Sukabumi)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & \infty & 18 & \infty \\ \infty & 0 & \infty & 116 & \infty \\ 2 & 2 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & \infty \\ \infty & 0 & \infty & 116 & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan baris ke 2 oleh 18 dan baris ke 4 oleh 2.

$$\hat{c}(10) = 700 + 96 + 20 = 816$$

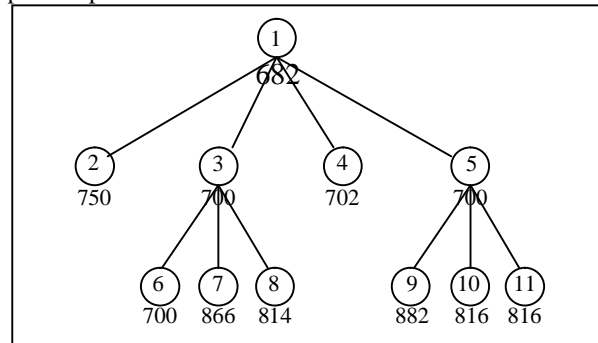
10. Simpul 11; lintasan 1,5,4 (Bandung-Tasik-Indramayu)

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & 2 & 98 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & 0 & 96 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

Matriks di atas diperoleh dari pengurangan baris ke 2 oleh 18 dan baris ke 4 oleh 2.

$$\hat{c}(11) = 700 + 96 + 20 = 816$$

Dari proses-proses reduksi di atas, sampai saat ini diperoleh pohon status berikut :



Gambar 5. Pohon status yang terbentuk

Selanjutnya, kita akan mengekspansi kembali simpul dengan bobot minimum, dalam hal ini adalah simpul 6.

11. Simpul 12; lintasan 1,3,2,4

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

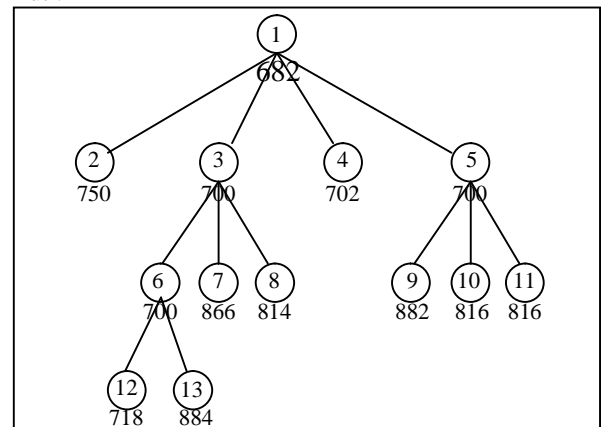
$$\hat{c}(12) = 700 + 18 + 0 = 718$$

12. Simpul 13; lintasan 1,3,2,5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \end{bmatrix}$$

$$\hat{c}(13) = 700 + 174 + 10 = 884$$

Sekarang kita telah memiliki pohon status sebagai berikut :

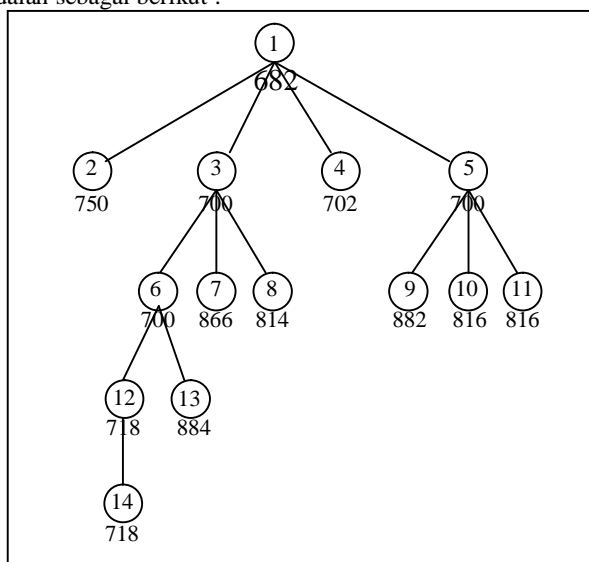


Gambar 6. Pohon status yang terbentuk

Dari gambar di atas, dapat kita lihat bahwa daun yang masih hidup dengan bobot minimum adalah simpul 12, sehingga kita akan mengekspansi simpul tersebut. Karena tinggal tersisa satu simpul, maka dapat kita langsung ambil simpul 14 dengan bobot :

$$\hat{c}(10) = 718 + 0 + 0 = 718$$

Maka, gambar akhir pohon status yang kita peroleh adalah sebagai berikut :



Gambar 7. Pohon status yang terbentuk

Dari gambar di atas, maka kita dapat menyimpulkan bahwa rute dengan *cost* minimum, dalam hal ini berarti rute terpendek, adalah melalui simpul 1-3-6-12-14-1 yang berarti melalui Bandung-Sukabumi-Jakarta-Indramayu-Tasikmalaya-Bandung.

Untuk kota yang lain, kita dapat mencari rute terpendek dengan cara yang sama, sehingga dengan demikian perjalanan yang dilakukan dapat lebih efektif dan efisien mengingat jarak yang ditempuh adalah minimum.

#### 4. KESIMPULAN DAN SARAN

Algoritma *Branch and Bound* cukup efektif digunakan dalam menentukan solusi optimum pada masalah TSP, dalam hal ini mencari rute terpendek untuk perjalanan antarkota di Jawa Barat. Dengan algoritma ini, kita dapat mencari kemungkinan solusi yang masih mungkin dan membuang kemungkinan lain yang tidak akan menghasilkan solusi optimum, hingga pada akhirnya ditemukan solusi tersebut.

Akan tetapi, algoritma ini dalam pengaplikasiannya di lapangan tidak akan selalu tepat. Dunia nyata menyajikan banyak hal yang tidak terduga, seperti kemacetan dan hambatan lain, yang tidak akan bisa diselesaikan hanya dengan algoritma dan komputerisasi. Oleh karena itu, pengembangan algoritma pada umumnya dan makalah ini pada khususnya akan selalu diperlukan untuk mendapatkan solusi yang lebih baik.

#### REFERENSI

- [1] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika STEI ITB, 2009
- [2] <http://kardady.wordpress.com/gambaran-transportasi-jawa-barat/jarak-antar-kota-jabar/>
- [3] <http://maps.google.com/>