

BAB III

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 di antara simpul-simpul hidup lainnya. Sedangkan simpul lainnya dimatikan. Nilai ongkos pada simpul i dinyatakan dengan :

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

3.1 Sejarah Algoritma Branch and Bound

Metode Branch And Bound diusulkan pertama kali oleh A.H.Land dan A.G.Doig pada tahun 1960. Sebenarnya metode ini dibuat untuk pemrograman linier (*linier programming*), Namun kenyataanya metode ini mampu menyelesaikan permasalahan seperti TSP dan beberapa masalah lain. Metode ini menggunakan pohon pencarian (*Search Tree*), setiap simpul di pohon merupakan representasi dari sejumlah kemungkinan solusi dari TSP. Metode ini hanya dapat digunakan untuk masalah optimasi saja (*optimazion problem*).

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.

Metode Branch And Bound sebenarnya bukan merupakan metode yang mutlak untuk penyelesaian TSP, metode ini merupakan kumpulan dari berbagai cara penyelesaian masalah (*class of solving problem*), hanya saja persamaan karakteristik cara – cara tersebut yang membuat mereka disebut *Branch And Bound*.

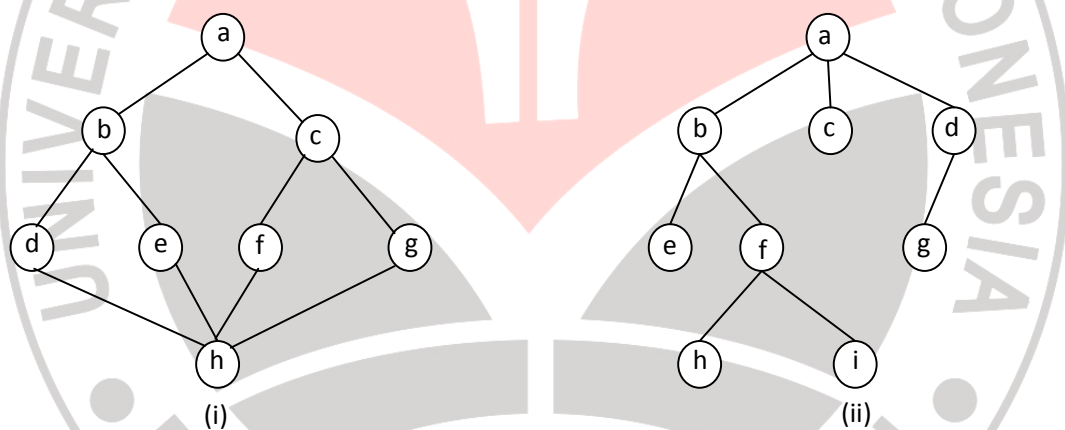
3.2. Algoritma Pencarian Melebar (BFS)

Algoritma pencarian melebar atau Breadth First Search merupakan algoritma yang diterapkan pada graf. Misalkan graf G yang mempunyai n buah simpul, akan dilakukan 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 level d dikunjungi terlebih dahulu sebelum simpul-simpul pada level $d+1$.

Algoritma BFS menggunakan antrian untuk menyimpan simpul-simpul yang baru dibangkitkan. Simpul-simpul yang baru dibangkitkan ditempatkan di belakang antrian. Prinsip antrian yang digunakan adalah FIFO (*First In First Out*). Dengan skema ini, simpul hidup dimasukkan ke dalam antrian, simpul berikutnya yang akan menjadi simpul ekspansi adalah simpul yang pertama masuk ke dalam antrian.

Sebagai contoh dapat dilihat pada graf di bawah ini :



Gambar 3.1. 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 $a - b - c - d - e - f - g - h - i$.

3.3 Solusi

Definisi 3.1: Solusi dari suatu permasalahan adalah adanya sebuah lintasan dari simpul awal ke simpul tujuan. Kualitas dari sebuah solusi diukur dari sebuah fungsi lintasan berbobot, dan sebuah solusi optimal adalah lintasan dengan bobot paling terkecil di antara lintasan lainnya. (Russel and Norvig)

Mengukur kemampuan penyelesaian masalah :

Keluaran dari suatu penyelesaian masalah Algoritma adalah gagal atau berhasil mencari suatu solusi. (Beberapa algoritma mungkin terhenti pada *an infinity loop* sehingga solusinya tidak ada). Ada 2 cara mengukur kemampuan dari sebuah algoritma, yaitu :

1. *Completeness* : Apakah algoritma tersebut dijamin dapat menemukan sebuah solusi manakala hanya ada 1 solusi?
2. *Optimality* : Apakah algoritma tersebut dapat menemukan solusi optimal, sesuai dengan Definisi 3.1?

3.4 Pencarian Solusi Algoritma Branch and Bound

Pada algoritma B&B ruang solusi diorganisasikan ke dalam pohon ruang status yang dibangun dengan skema BFS. Untuk mempercepat pencarian ke simpul solusi, setiap simpul diberi nilai ongkos $c(i)$ yang merupakan taksiran lintasan

termurah dari simpul akar simpul status tujuan. Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan prinsip FIFO pada antrian, melainkan dipilih simpul dengan nilai ongkos terkecil.

Fungsi heuristik untuk menghitung taksiran nilai ongkos dinyatakan secara umum sebagai :

$$c(i) = f(i) + g(i)$$

yang dalam hal ini:

$c(i)$ = ongkos untuk simpul i

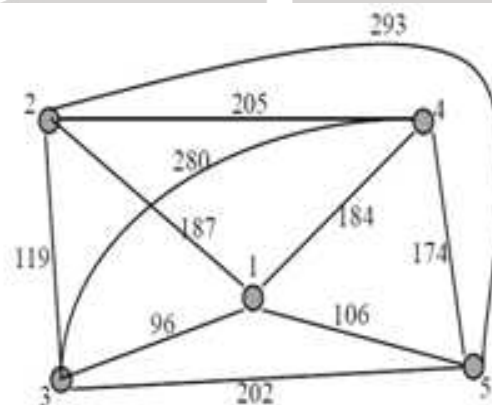
$f(i)$ = ongkos untuk mencapai simpul i

$g(i)$ = ongkos untuk mencapai simpul tujuan dari simpul i

3.5 Penerapan Algoritma Branch And Bound Pada Persoalan Graf

Pada pembahasan ini, penulis hanya akan mengambil 5 simpul dan 10 sisi.

Secara kasar, kelima simpul digambarkan dengan graf sebagai berikut :



Gambar 3.2 Graf yang dijadikan sebuah persoalan.

Simpul : 1, 2, 3, 4 dan 5.

Nilai pada sisi adalah cost yang menyatakan jarak antar simpul. Misalkan menentukan rute terpendek untuk mengunjungi setiap simpul pada graf, dimulai dari simpul 1 hingga kembali lagi ke simpul tersebut.

Secara logika, 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 harus dicoba banyak kemungkinan. Misalnya saja untuk n simpul = 5 (seperti gambar di atas), maka akan terdapat 120 kemungkinan yang harus dicari. Oleh karena itu, algoritma Branch and Bound menawarkan solusi yang lebih praktis untuk menemukan rute terpendek.

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

$$M_{5 \times 5} = \begin{bmatrix} \infty & 187 & 96 & 184 & 106 \\ 187 & \infty & 119 & 205 & 293 \\ 96 & 119 & \infty & 280 & 202 \\ 184 & 205 & 280 & \infty & 174 \\ 106 & 184 & 202 & 174 & \infty \end{bmatrix} \dots (1)$$

Matriks (1) merupakan matriks simetris karena jarak i ke j sama dengan jarak dari j ke i .

Selanjutnya matriks (1) akan direduksi agar lebih sederhana. Reduksi dilakukan dengan cara mengurangkan seluruh elemen pada baris atau kolom tertentu sehingga terdapat nilai 0 pada baris atau kolom tersebut.

Langkah pertama lakukan reduksi baris pada matriks (1) sehingga diperoleh matriks baru :

$$\begin{bmatrix} \infty & 91 & 0 & 88 & 10 \\ 68 & \infty & 0 & 86 & 174 \\ 0 & 23 & \infty & 184 & 106 \\ 10 & 31 & 106 & \infty & 0 \\ 0 & 187 & 96 & 68 & \infty \end{bmatrix} \dots (2)$$

Matriks (2) 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. Setelah mendapatkan nilai 0 pada tiap baris, kemudian periksa tiap kolom pada matriks (2). Jika belum terdapat nilai 0 maka, lakukan reduksi kolom :

$$\begin{bmatrix} \infty & 68 & 0 & 20 & 10 \\ 68 & \infty & 0 & 18 & 174 \\ 0 & 0 & \infty & 116 & 106 \\ 10 & 8 & 106 & \infty & 0 \\ 0 & 164 & 96 & 0 & \infty \end{bmatrix} \dots (3)$$

Matriks (3) dihasilkan dari pengurangan seluruh elemen pada kolom-2 dengan 23 dan kolom ke-4 dengan 68. 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.

$$\begin{matrix} \textcircled{1} \\ 682 \end{matrix}$$

Gambar 3.2 Pohon status yang terbentuk

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 :

- a) Ubah semua nilai pada baris i dan kolom j menjadi ∞ .
- b) Ubah $A(j, i)$ menjadi ∞
- c) 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 berkoresponden 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

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 18 & 174 \\ 0 & \infty & \infty & 116 & 106 \\ 10 & \infty & 106 & \infty & 0 \\ 0 & \infty & 96 & 0 & \infty \end{bmatrix} = \mathbf{B} \quad \dots (4)$$

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

2. Simpul 3; lintasan 1, 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & \infty & 18 & 174 \\ 0 & 0 & \infty & 116 & 106 \\ 10 & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & 156 \\ 0 & 0 & \infty & 116 & 106 \\ 10 & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & 0 & \infty \end{bmatrix} \quad \dots (5)$$

Matriks (5) diperoleh dari pengurangan baris ke-2 oleh 18.

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

3. Simpul 4; lintasan 1, 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & \infty & 174 \\ 0 & 0 & \infty & \infty & 106 \\ 10 & 8 & 106 & \infty & 0 \\ 0 & 164 & 96 & \infty & \infty \end{bmatrix} \quad \dots (6)$$

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

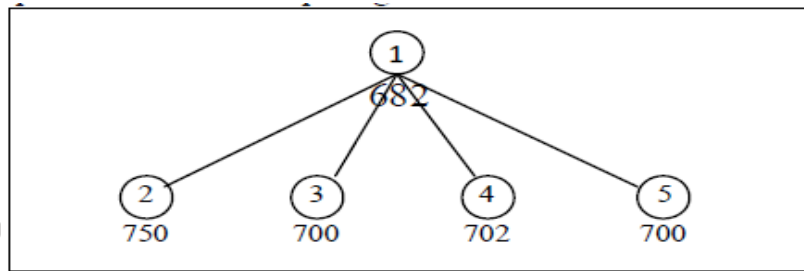
4. Simpul 5; lintasan 1, 5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 18 & \infty \\ 0 & 0 & \infty & 116 & \infty \\ 10 & 8 & 106 & \infty & \infty \\ 0 & 164 & 96 & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 18 & \infty \\ 0 & 0 & \infty & 116 & \infty \\ 2 & 0 & 98 & \infty & \infty \\ 0 & 164 & 96 & 0 & \infty \end{bmatrix} \quad \dots (7)$$

Matriks (7) 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 3.3 Pohon status yang terbentuk.

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

5. Simpul 6; lintasan 1, 3, 2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & 156 \\ \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & 0 & \infty \end{bmatrix} \dots (8)$$

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

6. Simpul 7; lintasan 1, 3, 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & \infty & 156 \\ \infty & \infty & \infty & \infty & \infty \\ 10 & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & 106 \\ \infty & \infty & \infty & \infty & \infty \\ 10 & 8 & \infty & \infty & 0 \\ 0 & 164 & \infty & \infty & \infty \end{bmatrix} \dots (9)$$

Matriks 9 diperoleh dari pengurangan baris ke-2 oleh 50.

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

7. Simpul 8; lintasan 1, 3, 5

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

Matriks 10 diperoleh dari pengurangan baris ke-4 oleh 8.

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

Selanjutnya ekspansi simpul 5 :

8. Simpul 9; lintasan 1, 5, 2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 18 & \infty \\ 0 & \infty & \infty & 116 & \infty \\ 2 & \infty & 98 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & 0 & \infty \\ 0 & \infty & \infty & 98 & \infty \\ 2 & \infty & 98 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} \dots (11)$$

Matriks di atas diperoleh dari pengurangan kolom ke-4 dengan 18.

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

9. Simpul 10; lintasan 1, 5, 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & \infty & 18 & \infty \\ 0 & 0 & \infty & 116 & \infty \\ 2 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 50 & \infty & \infty & 0 & \infty \\ 0 & 0 & \infty & 116 & \infty \\ 2 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} \dots (12)$$

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

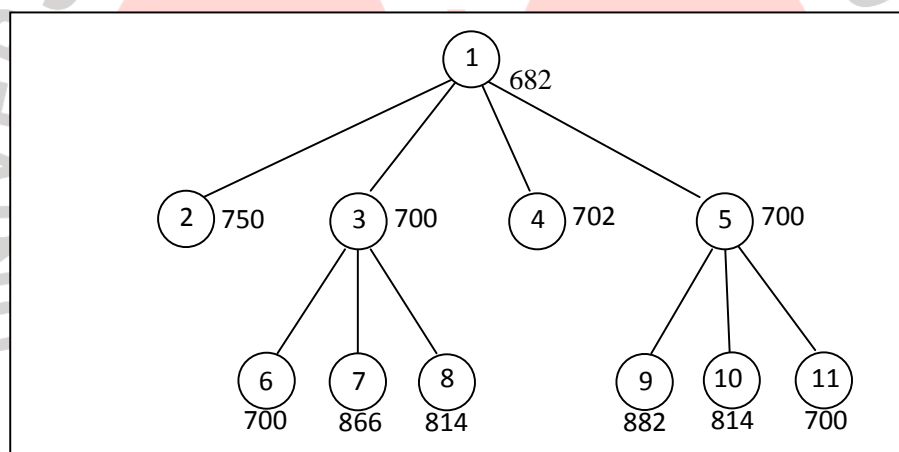
$$\hat{c}(10) = 700 + 96 + 18 = 814$$

10. Simpul 11; lintasan 1, 5, 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ 2 & 0 & 98 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} \dots (13)$$

$$\hat{c}(11) = 700 + 0 + 0 = 700$$

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



Gambar 3.4 Pohon status yang terbentuk.

Selanjutnya, akan diekspansi kembali simpul dengan bobot minimum, dalam hal ini adalah simpul 6 dan simpul 11. Pertama-tama ekspansi simpul 6 terlebih dahulu.

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

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & \infty & \infty \end{bmatrix} \dots (14)$$

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

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

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 0 & \infty \end{bmatrix} \dots (15)$$

Matriks (15) diperoleh dari pengurangan baris ke-4 oleh 10.

$$\hat{c}(13) = 700 + 156 + 10 = 866$$

Selanjutnya ekspansi simpul 11 :

13. Simpul 14; lintasan 1, 5, 4, 2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & 0 & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} \dots (16)$$

$$\hat{c}(14) = 700 + 0 + 0 = 700$$

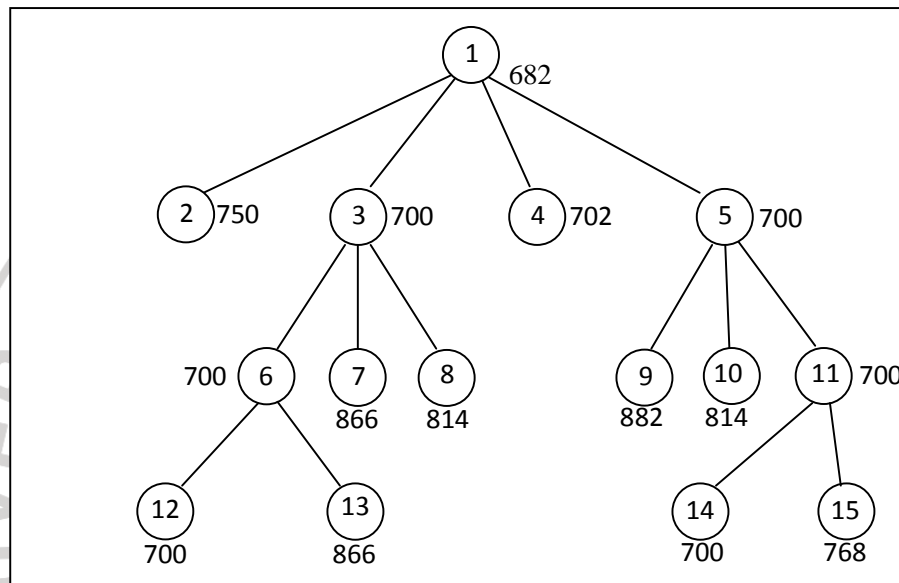
14. Simpul 15; lintasan 1, 5, 4, 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 68 & \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} \dots (17)$$

Matriks (17) diperoleh karena pengurangan baris k-2 dengan 68.

$$\hat{c}(15) = 700 + 0 + 68 = 768$$

Sekarang diperoleh pohon status sebagai berikut :



Gambar 3.5 Pohon status yang terbentuk.

Dari gambar 3.5, dan matriks (18) dapat dilihat bahwa daun yang masih hidup dengan bobot minimum adalah simpul 12 dan 14. Sehingga dapat mengekspansi simpul tersebut. Didapatkan simpul dengan bobot minimum yang sama. Langsung ambil simpul 16 dan simpul 17 dengan bobot : $\hat{c}(S) = 700 + 0 + 0 = 700$.

15. simpul 16; lintasan 1, 3, 2, 4, 5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \end{bmatrix} \dots (18)$$

$$\hat{c}(16) = 700 + 0 + 0 = 700$$

selanjutnya, ekspansi simpul 14 :

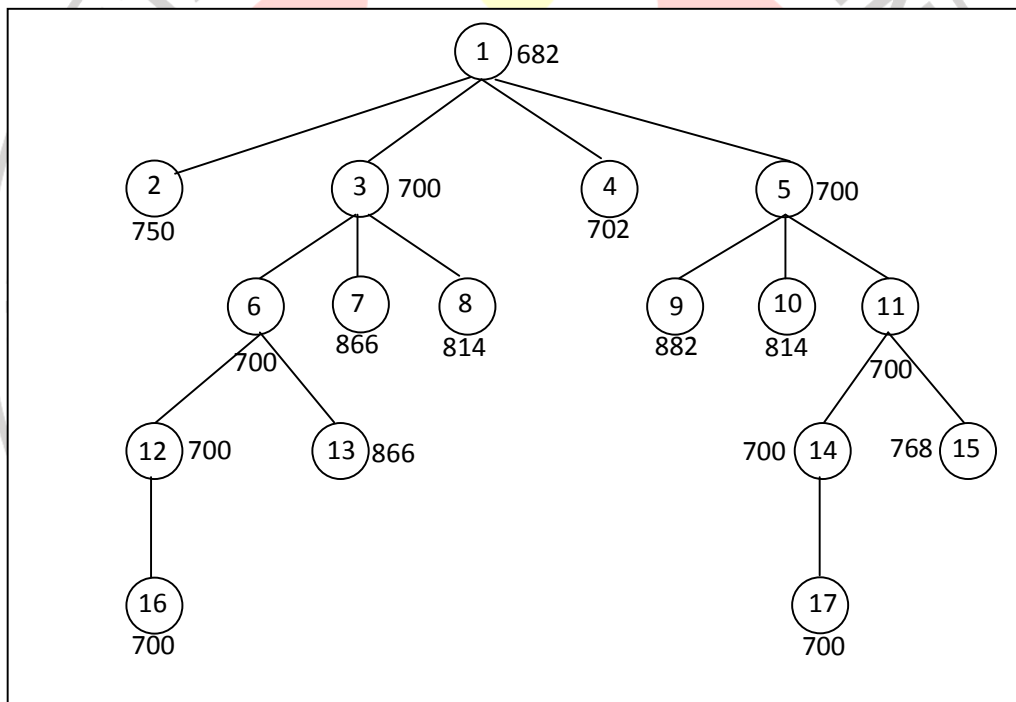
16.

Simpul 17; lintasan 1, 5, 4, 2, 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} \dots (19)$$

$$\hat{c}(17) = 700 + 0 + 0 = 700$$

Maka, gambar akhir pohon status yang diperoleh adalah sebagai berikut :



Gambar 3.6 Pohon status yang terbentuk.

Gambar 3.6 menyimpulkan bahwa rute dengan *cost* minimum, dalam hal ini berarti rute terpendek, adalah melalui simpul 1-3-6-12-16 dan 1-5-11-14-17 (simpul

pada ruang status) yang berarti melalui simpul 1-3-2-4-5-1 dan 1-5-4-2-3-1 (simpul pada graf).

Untuk simpul-simpul yang lain, dapat dicari rute terpendek dengan cara yang sama. Sehingga dengan demikian pencarian rute terpendek dapat lebih efektif dan efisien mengingat rute yang didapat adalah minimum.

