Penerapan Algoritma *Branch and Bound* untuk Optimasi Rute Penempelan Poster di Papan Mading ITB

Zain Fathoni
13508079
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18079@students.if.itb.ac.id

***Abstrak*—Banyaknya kegiatan yang ada di Institut Teknologi Bandung (ITB) menimbulkan konsekuensi terhadap banyaknya publikasi yang diperlukan. Salah satu media publikasi yang cukup efektif di ITB adalah publikasi melalui poster di mading-mading yang tersebar di seluruh penjuru ITB. Banyaknya tempat penempelan poster yang tersebar merata di ITB ini terkadang membuat para penempel poster terlalu lelah karena rute perjalanan yang diambil kurang optimal. Permasalahan semacam ini sering dikenal dengan nama *Travelling Salesman Problem (TSP)*. Dalam makalah ini penulis mencoba menerapkan Algoritma *Branch and Bound* untuk pencarian solusi dalam permalasahan ini. Algoritma *Branch and Bound* mengadaptasi skema traversal pada Algoritma *Breadth First Search*. Penulis mengambil sampel lima titik papan mading umum yang ada di ITB, kemudian mencoba menerapkan Algoritma *Branch and Bound* ini untuk mencari rute perjalanan yang paling optimal untuk penempelan poster di kelima titik tersebut. Dengan menggunakan algoritma ini, solusi optimal dapat ditemukan tanpa harus menelusuri seluruh kemungkinan solusi permasalahan yang ada.**

***Kata Kunci*—*Branch and Bound*, optimasi, papan mading ITB, *Travelling Salesman Problem*.**

I. PENDAHULUAN

Sebagai bagian dari sivitas akademika Institut Teknologi Bandung (ITB), tentunya kita mengetahui bahwa kegiatan yang ada di ITB ini sangatlah beragam. Baik yang diselenggarakan oleh internal ITB sendiri, termasuk di dalamnya rektorat ITB dan berbagai organisasi kemahasiswaan ITB, maupun oleh entitas eksternal ITB. Adanya kegiatan yang diselenggarakan tentu saja identik dengan publikasi yang dilancarkan kepada seluruh massa kampus, dan salah satu media publikasi yang cukup strategis di ITB adalah papan mading.

Sangat banyak papan mading yang ada di ITB, baik yang berstatus independen maupun milik lembaga seperti himpunan mahasiswa, unit, tata usaha program studi, dan lain-lain. Banyaknya papan mading di ITB ini cukup membuat berbagai pihak yang ingin memasang poster untuk publikasi cukup kewalahan untuk menelusuri semuanya, terlebih apabila rute yang diambil tidak optimal, ini tentunya akan sangat melelahkan bagi para penempel poster tersebut.

Oleh karena itu, dalam makalah ini penulis akan mencoba membahas bagaimana pemilihan rute yang dapat diambil untuk menelusuri mading-mading di ITB sedemikian sehingga perjalanan menjadi optimal dan tidak terlalu melelahkan bagi para penempel poster. Masalah pemilihan rute ini termasuk salah satu contoh dari permasalahan *Travelling Salesman Problem*. Yang akan dibahas dalam makalah ini adalah pencarian solusi optimal dari permasalahan tersebut dengan menggunakan Algoritma *Branch and Bound*.

II. PRINSIP DASAR

A. Pengenalan Algoritma Branch and Bound

Algoritma *Branch and Bound* yang juga sering disingkat menjadi algoritma B&B ini merupakan salah satu cara atau metode untuk pencarian di dalam ruang solusi secara sistematis. Dalam hal ini, ruang solusi akan diorganisasikan ke dalam pohon ruang status. Pembentukan pohon ruang status pada algoritma ini dibangun dengan skema pencarian melebar yang biasa disebut juga dengan Algoritma *Breadth First Search*, seringkali disingkat menjadi BFS [1].

Apabila kita memiliki sebuah graf G yang memiliki n buah simpul, kemudian kita ingin melakukan proses traversal dalam graf tersebut dimulai dari salah satu simpul yang ada dalam G, misalnya simpul v. Dalam skema BFS, penelusuran akan dimulai dari simpul v, kemudian dilanjutkan dengan mengunjungi terlebih dahulu simpul-simpul yang berdekatan dengan simpul v. Selanjutnya, mengunjungi simpul-simpul yang belum dikunjungi dan bertetangga dengan simpul yang sedang dikunjungi tersebut. Demikian seterusnya hingga proses traversal dianggap selesai, yakni seluruh simpul telah dikunjungi.

Apabila kita menggunakan skema BFS biasa dalam penelusuran graf seperti pada kasus di atas, akan membutuhkan waktu yang relatif lama. Algoritma *Branch and Bound* (B&B) ini mengadaptasi skema BFS dan menambahkan beberapa aspek dalam pembangunan ruang solusinya. Untuk mempercepat pencarian ke simpul solusi, maka setiap simpul akan diberikan sebuah nilai ongkos (*cost*). Dari nilai ongkos yang diperoleh, simpul-simpul yang telah dibangkitkan di langkah sebelumnya tidak seluruhnya akan diekspansi lagi, tetapi yang akan diekspansi adalah simpul yang memiliki nilai ongkos yang paling kecil. Pencarian berdasarkan ongkos terkecil ini juga biasa disebut dengan *least cost search*.

Nilai ongkos pada tiap simpul diberikan berdasarkan perkiraan atau taksiran ongkos paling kecil dari simpul tersebut ke simpul yang dituju (*goal node*). Dengan kata lain, nilai ongkos simpul i (dilambangkan dengan ĉ(i)) merupakan batas bawah (*lower bound*) dari ongkos pencarian solusi dari status i. Untuk menghitung ongkos ini digunakan sebuah fungsi pembatas. Fungsi pembatas ini berguna untuk membatasi pembangkitan simpul yang tidak mengarah kepada simpul solusi.

B. Prinsip Pencarian Solusi dengan Algoritma Branch and Bound

Sebagaimana telah dijelaskan sebelumnya bahwa algoritma B&B ini mengadaptasi skema BFS dalam membangun pohon ruang solusinya. Prinsip antrian yang digunakan pada metode BFS sama dengan skema FIFO (*First In First Out*), sehingga simpul berikutnya yang akan menjadi simpul yang dikunjungi adalah simpul yang pertama kali masuk ke dalam antrian. Prinsip pemilihan simpul seperti ini tidak memberi jaminan pencapaian ke simpul solusi dengan cepat, karena kita tidak tahu apakah simpul tersebut mengarah dengan cepat ke simpul solusi atau tidak.

Pada algoritma B&B, pencarian ke simpul solusi dapat dipercepat dengan memilih simpul hidup berdasarkan nilai ongkos yang menyatakan nilai batas, seperti yang telah diutarakan di upa-bab sebelumnya. Secara umum algoritma B&B adalah sebagai sebagai berikut [1]:

1. Masukkan simpul akar ke dalam antrian Q. Jika simpul akar adalah simpul solusi, maka solusi telah ditemukan dan pencarian dihentikan.
2. Jika Q kosong, tidak ada solusi. Pencarian dihentikan.
3. Jika Q tidak kosong, pilih dari antrian Q simpul I yang mempunyai ĉ(i) paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih salah satu secara sembarang.
4. Jika simpul i adalah simpul solusi, artinya solusi telah ditemukan dan pencarian dihentikan. Jika bukan, bangkitkan anak-anaknya. Jika simpul tidak memiliki anak, maka ulang lagi langkah 2.
5. Untuk setiap anak j dari simpul i, hitung ĉ(j) dan masukkan semua anak tersebut ke dalam antria Q.
6. Kembali ke langkah 2 hingga solusi ditemukan.

C. Travelling Salesman Person

Dalam upa-bab ini, akan dibahas penyelesaian masalah *Travelling Salesman Person* (TSP) dengan mneggunakan algoritma *Branch and Bound*.

Sebagai contoh, misalkan diberikan sebuah graf G=(V,E) yang merupakan graf lengkap yang merepresentasikan rute-rute perjalanan yang mungkin akan ditempuh. |V| adalah jumlah simpul dalam G yang masing-masing simpulnya diberi nomor secara berurutan dan identik 1, 2, 3, …, n. Jarak yang haris ditempuh dari sebuah simpul i ke simpul lainnya misalnya j adalah c_{ij}. Perjalanan akan berawal dari sebuah simpul dan berakhir di simpul itu juga. Maka S adalah ruang solusi, yang dalam hal ini S = {(1, π, 1)} di mana π adalah permutasi dadri simpul-simpul yang ada. Dan banyaknya solusi yang bisa terbentuk dinyatakan dengan |S| adalah sebanyak (|V|-1)! buah solusi. Solusi TSP dinyatakan sebagai vektor X = (1, x_1, x_2, …, x_{n-1}, 1). Dengan asumsi bahwa $x_0 = x_n = 1$, karena perjalanan berawal dan berakhir di simpul yang sama.

Kemudian pemberian nilai ongkos akan dilakukan dengan menggunakan matriks tereduksi dari matriks jarak antar simpul yang dibentuk dari graf G. Yang dimaksud dengan matriks tereduksi di sini adalah matriks yang tiap kolom dan tiap barisnya mengandung paling sedikit satu buah angka 0 dan elemen-elemen lainnya bernilai non-negatif. Untuk mendapatkan matriks tereduksi, maka tiap baris atau kolom yang belum mengandung angka 0 dikurangi dengan nilai terkecil pada baris atau kolom tersebut.

Semua angka yang digunakan untuk mengurangi tiap baris atau kolom di atas kemudian dijumlahkan. Hasil dari penjumlahan inilah yang kemudian dijadikan sebagai ĉ(*root*) atau nilai ongkos dari simpul awal/akar. Hal ini juga berarti bahwa soluis pada persoalan TSP tersebut paling tidak memiliki bobot minimum sebesar nilai ĉ(*root*) yang diperoleh tersebut.

Selanjutnya, misalkan A adalah matriks tereduksi untuk simpul R dan misal S adalah anak dari simpul R sedemikian sehingga sisi (R,S) pada pohon ruang status melambangkan sisi (i, j) pada graf G. Jika S bukan merupakan simpul daun, maka untuk mendapatkan matriks jarak yang tereduksi untuk simpul S adalah dengan cara sebagai berikut:

1. Jadikan semua elemen matriks pada baris i dan kolom j menjadi bernilai ∞.
2. Ubah elemen matriks di baris j kolom pertama menjadi bernilai ∞.
3. Reduksi kembali matriks dengan cara seperti yang telah dijelaskan sebelumnya.

Berdasarkan hasil dari perhitungan di atas, maka fungsi pembatas yang juga merupakan nilai ongkos untuk simpul S adalah:

$$\hat{c}(S) = \hat{c}(R) + A(i,j) + r \qquad (1)$$

Keterangan:

- ĉ(S) merupakan ongkos perjalanan minimum yang melalui simpul S.
- ĉ(R) merupakan ongkos perjalanan minimum yang melalui simpul R, di mana R adalah simpul bapak (*parent*) dari S.
- A(i, j) merupakan nilai pada baris i dan kolom j pada matriks jarak graf G yang berkoresponden

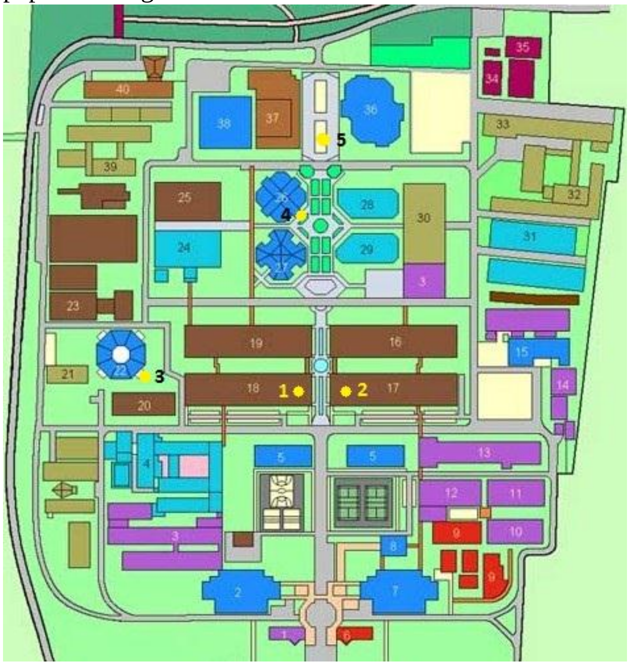
dengan sisi (R, S) pada pohon ruang status.

- r merupakan jumlah total dari angka-angka pengurang pada proses reduksi matriks untuk simpul S.

Hasil dari reduksi matriks di atas kemudian kita sebut sebagai matriks B.

III. PENERAPAN ALGORITMA

Dalam bab ini akan dibahas lebih lanjut mengenai cara optimasi pemilihan rute untuk penempelan poster di papan mading ITB.



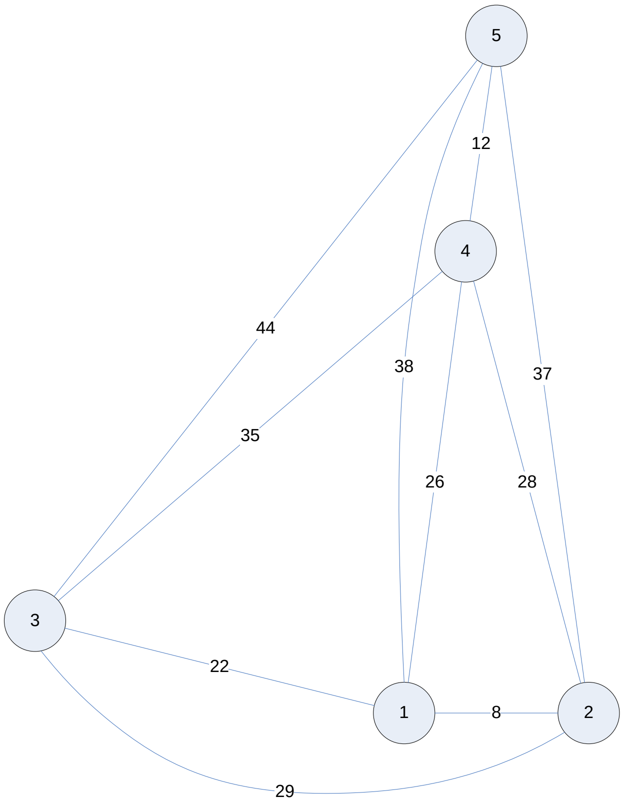
Gambar 1. Peta Persebaran 5 Papan Mading di ITB

A. Batasan Permasalahan

Pada bab sebelumnya telah disebutkan bahwa terdapat banyak sekali papan mading di ITB, baik yang bersifat independen maupun dalam wewenang pengelolaan lembaga tertentu. Apabila seluruh papan mading tersebut hendak dicari solusinya, maka pembahasannya akan terlalu panjang untuk disampaikan pada makalah ini. Karena adanya keterbatasan itulah, penulis membatasi permasalahan yang akan diselesaikan pada makalah ini. Penulis membatasi jumlah papan mading yang diambil hanya pada lima titik, dan kelimanya merupakan papan mading yang bersifat independen, yakni papan mading yang dapat ditempeli secara bebas oleh siapapun tanpa ada prosedur formal yang mengikat. Kelima titik tersebut di antaranya adalah papan mading di:

1. Selasar Labtek V (Benny Subianto)
2. Selasar Labtek VIII (Achmad Bakrie)
3. GKU Barat
4. Oktagon
5. Sunken Court

Dari peta di atas, kita dapat membuat graf yang menghubungkan kelima titik papan mading tersebut beserta jarak antarsimpulnya. Jarak ditentukan berdasarkan perbandingan dengan jarak-jarak antarsimpul lainnya. Graf yang akan terbentuk adalah sebagai berikut:



Gambar 2. Graf Tak-Berarah 5 Papan Mading di ITB Beserta Jaraknya

Dari graf tersebut kita dapat membuat matriks jarak antarpapan mading yang merepresentasikan jarak antara setiap papan mading. Berikut matriks hasil bentukannya:

$$\begin{bmatrix} \infty & 8 & 22 & 26 & 38 \\ 8 & \infty & 29 & 28 & 37 \\ 22 & 29 & \infty & 35 & 44 \\ 26 & 28 & 35 & \infty & 12 \\ 38 & 37 & 44 & 12 & \infty \end{bmatrix}$$

B. Penerapan Algoritma

Langkah awal yang harus dilakukan adalah menentukan $\hat{c}(root)$ untuk kasus ini melalui reduksi matriks. Matriks di atas akan direduksi seperti di bawah ini:

$$\begin{bmatrix} \infty & 8 & 22 & 26 & 38 \\ 8 & \infty & 29 & 28 & 37 \\ 22 & 29 & \infty & 35 & 44 \\ 26 & 28 & 35 & \infty & 12 \\ 38 & 37 & 44 & 12 & \infty \end{bmatrix} \begin{array}{c} R1-8 \\ R2-8 \\ R3-22 \\ R4-12 \\ R5-12 \end{array} \begin{bmatrix} \infty & 0 & 14 & 18 & 30 \\ 0 & \infty & 21 & 20 & 29 \\ 0 & 7 & \infty & 13 & 22 \\ 14 & 16 & 23 & \infty & 0 \\ 26 & 25 & 32 & 0 & \infty \end{bmatrix}$$

$$\begin{bmatrix} \infty & 0 & 14 & 18 & 30 \\ 0 & \infty & 21 & 20 & 29 \\ 0 & 7 & \infty & 13 & 22 \\ 14 & 16 & 23 & \infty & 0 \\ 26 & 25 & 32 & 0 & \infty \end{bmatrix} C3-14 \begin{bmatrix} \infty & 0 & 0 & 18 & 30 \\ 0 & \infty & 7 & 20 & 29 \\ 0 & 7 & \infty & 13 & 22 \\ 14 & 16 & 9 & \infty & 0 \\ 26 & 25 & 18 & 0 & \infty \end{bmatrix}$$

$$\begin{bmatrix} \infty & 0 & 0 & 18 & 30 \\ 0 & \infty & 7 & 20 & 29 \\ 0 & 7 & \infty & 13 & 22 \\ 14 & 16 & 9 & \infty & 0 \\ 26 & 25 & 18 & 0 & \infty \end{bmatrix} = A$$

Matriks tereduksi yang telah kita peroleh kita namanakn dengan matriks A. Jumlah total angka-angka yang digunakan untuk mereduksi matriks tersebut merupakan ĉ($root$), yakni batas dari simpul akar.

$$\hat{c}(root) = 8 + 8 + 22 + 12 + 12 + 14 = 76$$

Selanjutnya kita ekspansi simpul akar untuk mendapatkan simpul anak-anaknya. Penomoran simpul pada pohon ruang status berbeda dengan penomoran pada graf, sehingga penomoran untuk keempat simpul anaknya adalah 2, 3, 4, dan 5. Dari setiap simpul anak tersebut akan dicari nilai ongkosnya.

Jika simpul yang bersangkutan bukan merupakan simpul daun, maka reduksi matriks dikerjakan dengan cara sebagai berikut:

1. Ubah semua nilai pada baris i dan kolom j menjadi ∞. Ini untuk mencegah agar tidak ada lintasan yang keluar dari simpul i atau masuk pada simpul j.
2. Ubah A(j, 1) menjadi ∞. Ini untuk mencegah penggunaan sisi (j, 1), supaya penelusuran papan mading tidak kembali ke titik nomor 1 lagi.
3. Reduksi kembali semua baris dan kolom pada matriks A kecuali untuk elemen ∞.

Perthitungan dilakukan sesuai dengan ketentuan pada upa-bab 2.C. Berikut hasil perhitungannya:

1. Simpul 2; Lintasan: 1, 2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 13 & 22 \\ 0 & \infty & \infty & 13 & 22 \\ 14 & \infty & 9 & \infty & 0 \\ 26 & \infty & 18 & 0 & \infty \end{bmatrix} = B$$
$$r = 7, \hat{c}(2) = 76 + 0 + 7 = 83$$

2. Simpul 3; Lintasan: 1, 3

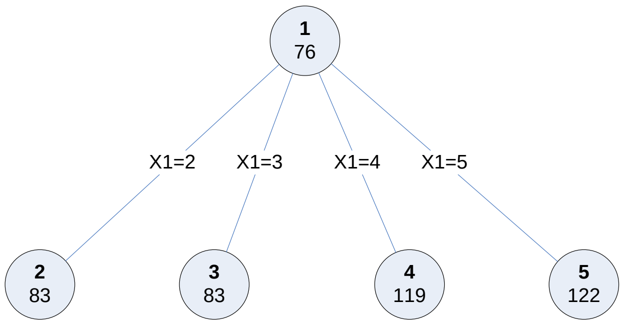
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 20 & 29 \\ \infty & 0 & \infty & 6 & 15 \\ 14 & 16 & \infty & \infty & 0 \\ 26 & 25 & \infty & 0 & \infty \end{bmatrix} = B$$
$$r = 7, \hat{c}(3) = 76 + 0 + 7 = 83$$

3. Simpul 4; Lintasan: 1, 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 7 & \infty & 29 \\ 0 & 0 & \infty & \infty & 22 \\ \infty & 9 & 9 & \infty & 0 \\ 8 & 0 & 0 & \infty & \infty \end{bmatrix} = B$$
$$r = 18 + 7, \hat{c}(4) = 76 + 18 + 25 = 119$$

4. Simpul 5; Lintasan: 1, 5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & 7 & 20 & \infty \\ 0 & 0 & \infty & 13 & \infty \\ 5 & 0 & 0 & \infty & \infty \\ \infty & 18 & 18 & 0 & \infty \end{bmatrix} = B$$
$$r = 9 + 7, \hat{c}(5) = 76 + 30 + 16 = 122$$



Gambar 3. Pohon Ruang Status Hasil Ekspansi Simpul 1 (akar)

Dari pohon ruang status di atas, terdapat empat simpul yang masih hidup. Urutan simpul yang masih hidup tersebut berdasarkan ongkos terkecil adalah 2, 3, 4, dan 5. Selanjutnya kita dapat memilih simpul dengan ongkos terkecil, yakni simpul 2 atau simpul 3. Kita dapat memilihnya secara acak, dalam kasus ini, mari kita coba memilih simpul 2 terlebih dahulu untuk diekspansi. Berikut penghitungan ongkosnya:

1. Simpul 6; Lintasan: 1, 2, 3

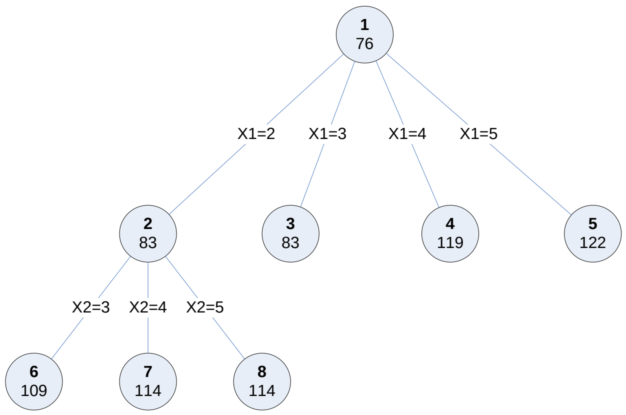
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & 9 \\ \infty & \infty & \infty & 0 & 9 \\ 14 & \infty & \infty & \infty & 0 \\ 26 & \infty & \infty & 0 & \infty \end{bmatrix} = C$$
$$r = 13 + 13, \hat{c}(6) = 83 + 0 + 26 = 109$$

2. Simpul 7; Lintasan: 1, 2, 4

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & 22 \\ 0 & \infty & \infty & \infty & 22 \\ \infty & \infty & 9 & \infty & 0 \\ 8 & \infty & 0 & \infty & \infty \end{bmatrix} = C$$
$$r = 18, \hat{c}(7) = 83 + 13 + 18 = 114$$

3. Simpul 8; Lintasan: 1, 2, 5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & 13 & \infty \\ 0 & \infty & \infty & 13 & \infty \\ 5 & \infty & 0 & \infty & \infty \\ \infty & \infty & 18 & 0 & \infty \end{bmatrix} = C$$
$$r = 9, \hat{c}(8) = 83 + 22 + 9 = 114$$



Gambar 4. Pohon Ruang Status Setelah Ekspansi Simpul 2

Dari pohon ruang status di atas, terdapat enam simpul yang masih hidup. Urutan simpul yang masih hidup tersebut berdasarkan ongkos terkecil adalah 3, 6, 7, 8, 4, dan 5. Selanjutnya kita pilih simpul dengan ongkos terkecil, yakni simpul 3 untuk diekspansi. Berikut penghitungan ongkosnya:

1. Simpul 9; Lintasan: 1, 3, 2

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & 9 \\ \infty & \infty & \infty & 0 & 9 \\ 0 & \infty & \infty & \infty & 0 \\ 12 & \infty & \infty & 0 & \infty \end{bmatrix} = C$$

$$r = 20 + 6 + 14, \hat{c}(9) = 83 + 0 + 40 = 123$$

2. Simpul 10; Lintasan: 1, 3, 4

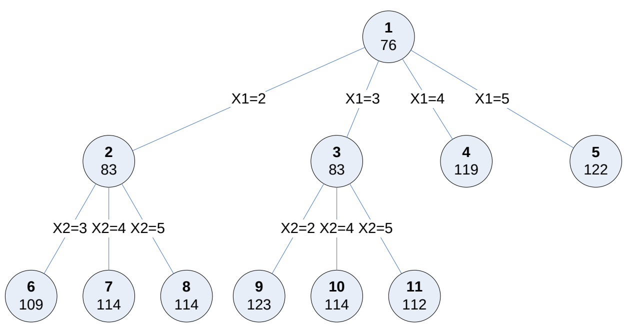
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & 29 \\ \infty & 0 & \infty & \infty & 15 \\ \infty & 16 & \infty & \infty & 0 \\ 1 & 0 & \infty & \infty & \infty \end{bmatrix} = C$$

$$r = 25, \hat{c}(10) = 83 + 6 + 25 = 114$$

3. Simpul 11; Lintasan: 1, 3, 5

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 20 & \infty \\ \infty & 0 & \infty & 6 & \infty \\ 0 & 2 & \infty & \infty & \infty \\ \infty & 25 & \infty & 0 & \infty \end{bmatrix} = C$$

$$r = 14, \hat{c}(11) = 83 + 15 + 14 = 112$$



Gambar 5. Pohon Ruang Status Setelah Ekspansi Simpul 3

Dari pohon ruang status di atas, terdapat delapan simpul yang masih hidup. Urutan simpul yang masih hidup tersebut berdasarkan ongkos terkecil adalah 6, 11, 7, 8, 10, 4, 5, dan 9. Selanjutnya kita pilih simpul dengan ongkos terkecil, yakni simpul 6 untuk diekspansi. Berikut penghitungan ongkosnya:

1. Simpul 12; Lintasan: 1, 2, 3, 4

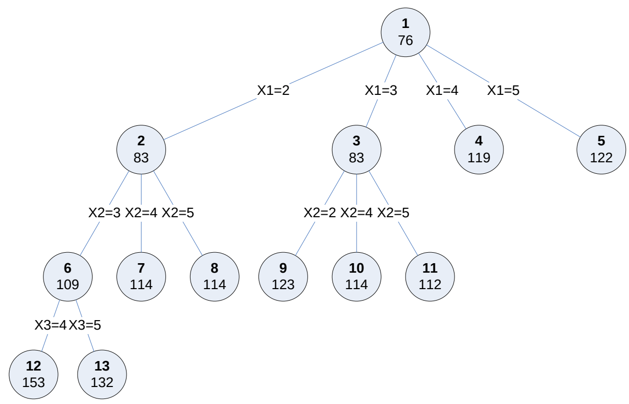
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & 0 \\ 0 & \infty & \infty & \infty & \infty \end{bmatrix} = D$$

$$r = 9 + 9 + 26, \hat{c}(12) = 109 + 0 + 44 = 153$$

2. Simpul 13; Lintasan: 1, 2, 3, 5

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

$$r = 14, \hat{c}(13) = 109 + 9 + 14 = 132$$



Gambar 6. Pohon Ruang Status Setelah Ekspansi Simpul 6

Dari pohon ruang status di atas, terdapat sembilan simpul yang masih hidup. Urutan simpul yang masih hidup tersebut berdasarkan ongkos terkecil adalah 11, 7, 8, 10, 4, 5, 9, 13, dan 12. Selanjutnya kita pilih simpul dengan ongkos terkecil, yakni simpul 11 untuk diekspansi. Berikut penghitungan ongkosnya:

1. Simpul 14; Lintasan: 1, 3, 5, 2

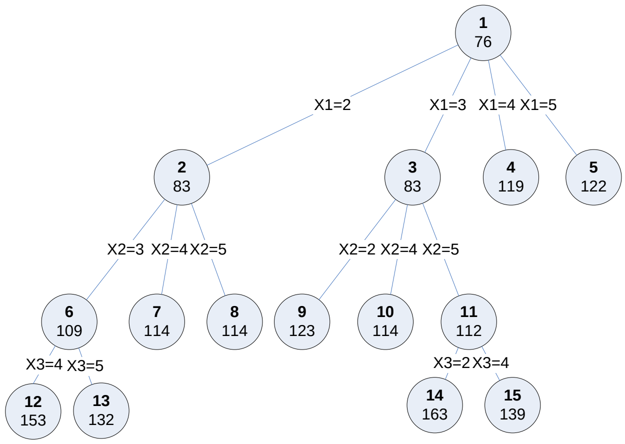
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \end{bmatrix} = D$$

$$r = 20 + 6, \hat{c}(14) = 112 + 25 + 26 = 163$$

2. Simpul 15; Lintasan: 1, 3, 5, 4

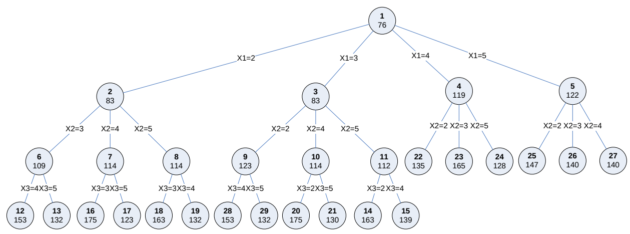
$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty & \infty \end{bmatrix} = D$$

$$r = 2 + 25, \hat{c}(15) = 112 + 0 + 27 = 139$$



Gambar 7. Pohon Ruang Status Setelah Ekspansi Simpul 11

Penelusuran ini cukup panjang, terlalu banyak apabila seluruhnya dimuat di dalam makalah ini. Oleh karena itu, langkah-langkah selanjutnya tidak ditampilkan satu per satu. Hingga ekspansi pada simpul 27 (yang menghasilkan simpul 28 dan 29, pohon ruang status berkembang menjadi seperti berikut:



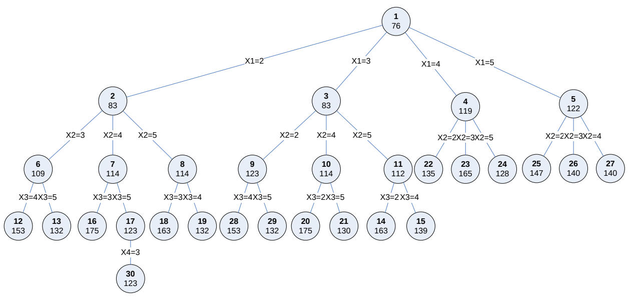
Gambar 8. Pohon Ruang Status Setelah Ekspansi Simpul 27

Berikutnya akan dilakukan ekspansi pada simpul 17, yang merupakan simpul dengan ongkos terkecil di antara semua simpul yang masih hidup. Berikut penghitungannya:

1. Simpul 30; Lintasan: 1, 2, 4, 5, 3

$$\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \end{bmatrix} = E$$

$$r = 0, \hat{c}(14) = 123 + 0 + 0 = 123$$



Gambar 9. Pohon Ruang Status Setelah Ekspansi Simpul 29

Karena simpul 30 adalah simpul daun, maka solusi pertama ditemukan, yaitu perjalanan dengan lintasan 1, 2, 4, 5, dan 3, dengan panjang 123. Solusi ini merupakan solusi terbaik sampai sejauh ini.

Apabila kita lihat simpul-simpul hidup lainnya, ongkos yang dimiliki lebih besar dari 123. Dengan demikian, simpul-simpul tersebut tidak mungkin lagi menghasilkan solusi rute perjalanan dengan ongkos kurang kari 123. Oleh karena itu, semua simpul hidup yang tersisa dibunuh, dan tidak akan diekspansi lagi.

Karena tidak ada lagi simpul hidup di dalam pohon ruang status, maka rute perjalanan 1 → 2 → 4 → 5 → 3 menjadi solusi dari pencarian rute terpendek ini.

C. Solusi TSP untuk Penempelan Poster di Lima Titik Papan Mading di ITB

Dari pencarian nilai ongkos di upa-bab sebelumnya dapat kita simpulkan bahwa rute TSP terbaik untuk penempelan poster di kelima titik papan mading di ITB jika perjalanan berawal dari titik papan mading di Selasar Labtek V (Benny Subianto) dan berakhir di tempat yang sama, adalah rute Labtek V → Labtek VIII → Oktagon → Sunken Court → GKU Barat → Labtek V, dengan panjang lintasan 123 satuan perbandingan panjang yang digunakan.

IV. ANALISIS

Apabila kita perhatikan lebih lanjut, dalam contoh kasus yang diangkat pada makalah ini Algoritma *Branch and Bound* menunjukkan perbedaan apabila dibandingkan dengan Algoritma *Breadth First Search* biasa.

Dengan berbedanya pola ekspansi simpul yang ditentukan oleh fungsi pembatas dalam algoritma B&B, kita dapat mengurangi banyaknya simpul yang harus diekspansi. Sebagai contohnya, apabila menggunakan Algoritma BFS biasa, kita harus mengekspansi simpul-simpul 22, 23, 24, 25, 26, 27, dan simpul-simpul lainnya terlebih dahulu sebelum mencapai simpul 17 yang menuntun kita pada solusi permasalahan. Bahkan kita harus menelusuri seluruh kemungkinan solusi apabila kita ingin mencari solusi optimal di antara seluruh solusi yang mungkin.

Dari fenomena ini, penulis menyimpulkan bahwa Algoritma *Branch and Bound* mampu untuk memberikan solusi optimal untuk *Travelling Salesman Problem* dengan cukup efisien, tanpa harus mengekspansi seluruh simpul hingga mencapai daun. Namun demikian, Algoritma B&B ini belum tentu merupakan algoritma terbaik untuk memecahkan permasalahan ini, karena diperlukan studi yang lebih mendalam dan menyeluruh untuk itu.

IV. KESIMPULAN

1. Algoritma *Branch and Bound* banyak digunakan sebagai metode pencarian solusi. Algoritma ini menggunakan skema *Breadth First Search* sebagai skema traversal dalam grafnya.
2. Salah satu metode pencarian solusi untuk *Travelling Salesman Problem* adalah algoritma *Branch and Bound*.
3. Pencarian solusi TSP pada *Branch and Bound* adalah dengan menggunakan fungsi pembatas yang dirumuskan oleh persamaan (1).
4. Rute perjalanan optimal untuk penempelan poster di kelima titik papan mading di ITB dapat dicari dengan menggunakan Algoritma *Branch and Bound*. Rute optimal yang ditemukan adalah Labtek V → Labtek VIII → Oktagon → Sunken Court → GKU Barat → Labtek V.

REFERENSI

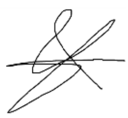
[1] Rinaldi Munir, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika STEI ITB, 2009.

[2] http://bangkamil.wordpress.com/2008/08/19/keajaiban-keunikan-itb/ diakses 08 Desember 2010, Pukul 20.15.

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, 8 Desember 2010



Zain Fathoni
13508079