

Optimasi Rute Pengangkutan Sampah di ITB dengan Algoritma *Branch And Bound*

Pranoto Budi Sasongko

NIM: 13506098

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha No.10, Bandung
pranoto_budi_s@students.itb.ac.id

ABSTRAK

Zona pengangkutan sampah di ITB terbagi menjadi 2 bagian yaitu barat dan timur. Makalah ini akan membahas optimasi sistem pengangkutan sampah di ITB khususnya pada zona timur karena memiliki rute yang lebih kompleks. Masalah pengangkutan sampah ini dimodelkan sebagai persoalan TSP (*Traveling SalesPerson Problem*) karena kita akan mencari jalur terpendek untuk mengunjungi semua tempat sampah yang ada. Selama ini proses pengambilan sampah dilakukan sesuai intuisi dari supir saja. Dalam makalah ini akan dibahas secara singkat pencarian rute angkut optimum dengan menggunakan algoritma *Branch And Bound*.

Kata kunci: Rute Angkut Sampah ITB, Pengangkutan Sampah di ITB dengan *Branch And Bound*, Algoritma *Branch And Bound*

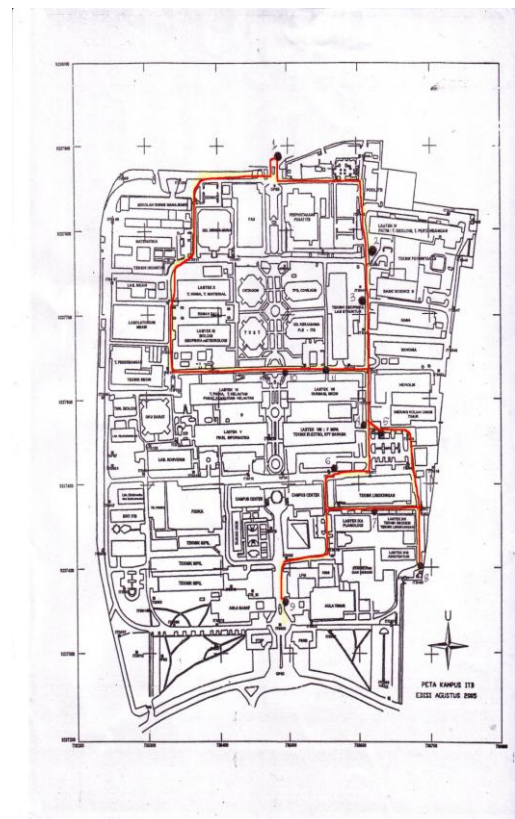
I. PENDAHULUAN

TSP (*Traveling SalesPerson Problem*) didefinisikan sebagai persoalan dimana diberikan beberapa tempat dan kita harus mengunjungi masing-masing tempat dan diharapkan kita akan mendapatkan jarak yang minimum. Proses menentukan rute pengambilan sampah di ITB dapat dikategorikan dalam TSP.

Di ITB sendiri ada beberapa *spot* tempat sampah, untuk zona timur terdapat 7 *spot* tempat sampah. Proses pengambilan sampah akan dimulai dari Gerbang Depan ITB yang disimbolkan dengan node 6 pada tabel di samping dan rute akan berakhir kembali ke Gerbang Depan ITB.

Table 1 Lokasi Tempat Sampah

No.	Lokasi
1	Labtek I (Kimia)
2	Labtek VIII bagian kanan
3	Labtek VIII bagian depan
4	Teknik Lingkungan
5	Arsitektur
6	Gerbang Depan
7	TVST



Gambar 1. Rute Angkutan Sampah zona timur di ITB

Dan untuk jarak antara masing-masing tempat sampah di tiap tempat tadi digambarkan dalam tabel di bawah ini:

Table 2 Jarak antar Tempat Sampah

	1	2	3	4	5	6	7
1	∞	82	∞	∞	∞	∞	109
2	82	∞	124	139	147	∞	114
3	∞	124	∞	112	142	142	∞
4	∞	139	112	∞	82	135	∞
5	∞	147	142	82	∞	165	∞
6	∞	∞	142	135	165	∞	∞
7	109	114	∞	∞	∞	∞	∞

II. Algoritma Branch And Bound

Algoritma Branch & Bound didasarkan pada algoritma BFS (Breadth First Search), namun dengan sedikit modifikasi. Pada algoritma BFS semua anak yang dibangkitkan dari simpul asal akan disimpan di Queue. kemudian setiap simpul anak tadi akan dibangkitkan lagi satu per satu sesuai antrian di Queue. Hal ini akan terus dilakukan hingga menemukan solusi pertama. namun pada algoritma B&B pencarian ke simpul solusi dapat dipercepat dengan memilih simpul hidup berdasarkan nilai ongkos (cost). Setiap simpul yang hidup diasosiasikan dengan sebuah ongkos yang menyatakan nilai batas (bound). Untuk setiap simpul X, nilai batas ini dapat berupa :

- jumlah simpul dalam upapohon X yang perlu dibangkitkan sebelum simpul solusi ditemukan, atau
- panjang lintasan dari simpul X ke simpul solusi terdekat (di dalam upapohon X yang bersangkutan)

Pemberian nilai batas pada persoalan yang akan kita selesaikan adalah dengan menggunakan fungsi heuristik untuk menghitung taksiran nilai tersebut.

$$c(S) = c(R) + A(i,j) + r$$

$c(S)$ = ongkos perjalanan minimum yang melalui simpul S

$c(R)$ = Bobot perjalanan minimum yang melalui simpul R, yang dalam hal ini adalah orangtua dari S.

r = jumlah semua pengurang pada proses memperoleh matriks tereduksi untuk simpul S.

nilai c digunakan untuk mengurutkan pencarian. Simpul berikutnya yang dipilih untuk diekspansi adalah simpul yang memiliki nilai c minimum. Jadi algoritma B&B secara umum adalah sebagai berikut:

- Masukkan simpul akar ke dalam antrian Q. Jika simpul akar adalah simpul solusi (*goal node*), maka solusi telah ditemukan. Stop.
- Jika Q kosong, tidak ada solusi. Stop.
- Jika Q tidak kosong, pilih dari antrian Q simpul I yang mempunyai $c(i)$ paling kecil. Jika terdapat beberapa simpul I yang memenuhi, pilih satu secara sembarang.
- Jika simpul I adalah simpul solusi, berarti solusi sudah ditemukan, stop. 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 $c(j)$, dan masukkan semua anak-anaknya ke dalam antrian Q.
- Kembali ke langkah 2.

III. METODE PENYELESAIAN

Untuk menyelesaikan persoalan ini, kita akan mereduksi matriks sebelumnya yaitu matriks jarak antar tempat sampah di ITB sehingga setiap baris dan kolom memiliki elemen bernilai 0. Dari sana kita akan mendapatkan Cost minimum dari akar (*root*).

Table 3 Matriks tereduksi untuk Simpul Akar

	1	2	3	4	5	6	7
1	∞	0	∞	∞	∞	∞	0
2	0	∞	35	57	65	∞	5
3	∞	12	∞	0	30	0	∞
4	∞	57	23	∞	0	13	∞
5	∞	65	53	0	∞	53	∞
6	∞	∞	0	0	30	∞	∞
7	0	5	∞	∞	∞	∞	∞

$$C(\text{root}) = 82 + 82 + 112 + 82 + 135 + 109 + 7 + 30 + 27$$

$$C(\text{root}) = 748$$

Perhitungan selanjutnya untuk simpul-simpul lain adalah sebagai berikut:

1. Simpul 3; lintasan 6,3

Table 4 Matriks tereduksi untuk Simpul 3

	1	2	3	4	5	6	7
1	∞	0	∞	∞	∞	∞	0
2	0	∞	∞	57	65	∞	5
3	∞	12	∞	0	30	∞	∞
4	∞	57	∞	∞	0	0	∞
5	∞	65	∞	0	∞	40	∞
6	∞	∞	∞	∞	∞	∞	∞
7	0	5	∞	∞	∞	∞	∞

$$\begin{aligned} C(3) &= C(6) + A(6,3) + r \\ &= 748 + 0 + 13 = 761 \end{aligned}$$

2. Simpul 4; lintasan 6,4

Table 5 Matriks tereduksi untuk Simpul 4

	1	2	3	4	5	6	7
1	∞	0	∞	∞	∞	∞	0
2	0	∞	35	∞	65	∞	5
3	∞	12	∞	∞	30	0	∞
4	∞	57	23	∞	0	∞	∞
5	∞	12	0	∞	∞	0	∞
6	∞	∞	∞	∞	∞	∞	∞
7	0	5	∞	∞	∞	∞	∞

$$\begin{aligned} C(4) &= C(6) + A(6,4) + r \\ &= 748 + 0 + 53 = 801 \end{aligned}$$

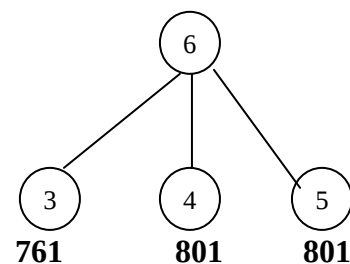
3. Simpul 5; lintasan 6,5

Table 6 Matriks tereduksi untuk Simpul 4

	1	2	3	4	5	6	7
1	∞	0	∞	∞	∞	∞	0
2	0	∞	∞	57	65	∞	5
3	∞	12	∞	0	30	∞	∞
4	∞	57	∞	∞	0	0	∞
5	∞	65	∞	0	∞	40	∞
6	∞	∞	∞	∞	∞	∞	∞
7	0	5	∞	∞	∞	∞	∞

$$\begin{aligned} C(5) &= C(6) + A(6,5) + r \\ &= 748 + 0 + 13 = 801 \end{aligned}$$

Pohon ruang status yang terbentuk hingga sekarang adalah



karena simpul 3 memiliki cost yang lebih kecil maka kita akan memilih simpul 3. Selanjutnya kita akan memilih antara simpul 2, 3, 5

4. Simpul 2; lintasan 6,3,2

Table 7 Matriks tereduksi untuk simpul 2

	1	2	3	4	5	6	7
1	∞	∞	∞	∞	∞	∞	0
2	∞	∞	∞	57	65	∞	5
3	∞	∞	∞	∞	∞	∞	∞
4	∞	∞	∞	∞	0	0	∞
5	∞	∞	∞	0	∞	40	∞
6	∞	∞	∞	∞	∞	∞	∞
7	0	∞	∞	∞	∞	∞	∞

$$\begin{aligned} C(2) &= C(3) + A(3,2) + r \\ &= 761 + 12 + 0 = 773 \end{aligned}$$

5. Simpul 4; lintasan 6,3, 4

Table 8 Matriks tereduksi untuk simpul 4

	1	2	3	4	5	6	7
1	∞	0	∞	∞	∞	∞	0
2	0	∞	∞	∞	65	∞	5
3	∞	∞	∞	∞	∞	∞	∞
4	∞	52	∞	∞	0	0	∞
5	∞	60	∞	∞	∞	40	∞
6	∞	∞	∞	∞	∞	∞	∞
7	0	0	∞	∞	∞	∞	∞

$$\begin{aligned} C(4) &= C(3) + A(3, 4) + 5 \\ &= 761 + 0 + 5 = 766 \end{aligned}$$

6. Simpul 5; lintasan 6, 3, 5

Table 9 Matriks tereduksi simpul 5

	1	2	3	4	5	6	7
1	∞	0	∞	∞	∞	∞	0
2	0	∞	∞	57	∞	∞	5
3	∞	∞	∞	∞	∞	∞	∞
4	∞	44	∞	∞	∞	0	∞
5	∞	65	∞	0	∞	40	∞
6	∞	∞	∞	∞	∞	∞	∞
7	0	5	∞	∞	∞	∞	∞

$$\begin{aligned} C(5) &= C(3) + A(3,5) + 5 \\ &= 761 + 30 + 0 = 791 \end{aligned}$$

Proses ini akan terus dilakukan hingga ditemukan solusi paling optimal. pada akhir proses kita mendapatkan rute sebagai berikut:

6- 3 – 2 – 1 – 7 – 4 – 5 – 6

Jika dijabarkan rute yang dipilih adalah:

Gerbang Depan → Labtek VIII bagian depan → Labtek VIII bagian kanan → Labtek I (kimia) → TVST → Teknik Lingkungan → Arsitektur → Gerbang Depan

IV. KESIMPULAN

Algoritma *Branch And Bound* telah berhasil diterapkan untuk menyelesaikan masalah optimasi pengambilan sampah di lingkungan ITB. Algoritma ini belum dapat dipastikan apakah memberikan hasil rute yang paling optimum, karena belum dibandingkan dengan algoritma yang lain dan belum ada pembuktian secara matematis. Namun dari solusi yang didapat secara sekilas algoritma ini dirasa cukup efisien dalam pencarian rute minimum pengambilan sampah di lingkungan ITB.

REFERENSI

[1] Rinaldi Munir, *Diktat Kuliah Strategi Algoritmik*, Departemen Teknik Informatika Institut Teknologi Bandung 2007.