

## KOMBINASI ALGORITMA BRANCH AND BOUND DAN CHEAPEST INSERTION HEURISTIC DALAM MENYELESAIKAN ASYMMETRIC TRAVELLING SALESMAN PROBLEM

**Muhammad Alifullah Sampurno Nur**

Program Studi Matematika, FMIPA, Universitas Negeri Surabaya  
e-mail : muhammad.17030214030@mhs.unesa.ac.id

**Budi Rahadjeng**

Program Studi Matematika, FMIPA, Universitas Negeri Surabaya  
e-mail : budirahadjeng@unesa.ac.id

### Abstrak

Travelling Salesman Problem (TSP) adalah suatu permasalahan seorang salesman yang mengunjungi setiap kota tepat satu kali dan kembali lagi ke kota asal dengan jarak tempuh minimum. Tujuan dalam artikel ini adalah menentukan rute perjalanan layanan jemput donasi LAZIS dengan menerapkan kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* dalam menyelesaikan *Asymmetric TSP*. Data yang digunakan adalah data sekunder berisi alamat donatur yang didapatkan dari LAZIS. Analisis data dilakukan dengan cara menginterpretasikan permasalahan ke dalam bentuk graf kemudian dilakukan pencarian dan penentuan jarak dengan menggunakan aplikasi *Google Maps*, memberi bobot pada graf dengan jarak yang diperoleh kemudian kombinasi Algoritma *Branch And Bound* dan *Cheapest Insertion Heuristic* digunakan untuk menyelesaikan permasalahan. Hasil yang didapatkan untuk rute terpendeknya adalah Kantor LAZIS → Sri → Reza → Bayu → Tasya → Maisaroh → Sarmo → Khusnul → Lely → Yayuk → Ayniyatur → Istiqomah → Nina → Heny → Ainur → Ratna → Kantor LAZIS dengan total jarak 54,9 km.

**Kata Kunci:** teori graf, travelling salesman problem, branch and bound, cheapest insertion heuristic.

### Abstract

*Travelling Salesman Problem (TSP) is an issue which a salesman visits each city exactly once and returns to the origin city by shortest possible distance. This research aimed to determine the peregrination routes about Donation Pick Up Services of LAZIS within apply combination of Branch-and-Bound and Cheapest Insertion Heuristic Algorithm in order to accomplish Asymmetric TSP. Data of this research is secondary data which contained donors address from LAZIS database. Data analysis method is interpreting this issue to become a Graph form, then finding and determining its possible distance using Google Maps, then adding weight and distance which obtained to graph and the last solving this problem using Branch-and-Bound and Cheapest Insertion Heuristic Algorithm Design. Result of this research is the shortest possible routes of Donation Pick Up Services may be reached was LAZIS Head Office → Sri → Reza → Bayu → Tasya → Maisaroh → Sarmo → Khusnul → Lely → Yayuk → Ayniyatur → Istiqomah → Nina → Heny → Ainur → Ratna → returned to LAZIS head office that has total distance 54,9 km.*

**Keywords:** graph theory, travelling salesman problem, branch and bound algorithm, cheapest insertion heuristic.

## PENDAHULUAN

Sebagai cabang dari matematika, teori graf telah ada sejak dua abad silam. Pada 1736, Euler seorang matematikawan terkenal dari Swiss menulis jurnal pertama tentang teori graf. Teori graf berkembang pesat pada beberapa puluh tahun terakhir. Dikarenakan pengaplikasiannya dalam berbagai bidang ilmu seperti: Ilmu Komputer, Teknik, Sains, Bisnis dan Ilmu Sosial (Budayasa, 2007). Banyak permasalahan dalam kehidupan sehari-hari yang dapat direpresentasikan dalam

bentuk graf, salah satunya adalah persoalan pedagang keliling. Persoalan pedagang keliling atau yang lebih dikenal dengan nama Travelling Salesman Problem (TSP) merupakan persoalan dalam teori graf dan termasuk masalah optimasi yang menarik para peneliti sejak lama.

Kusrini (2007) dalam penelitiannya menerapkan Algoritma *Cheapest Insertion Heuristics* dan Basis Data dalam menyelesaikan TSP, didapatkan bahwa waktu proses dari algoritma ini bergantung dari jumlah kota, semakin banyak kota maka semakin besar pula waktu yang dibutuhkan

(Kusrini & Istiyanto, 2007). Lebih lanjut, Dian (2013) membandingkan beberapa algoritma dalam menyelesaikan TSP diantaranya adalah Algoritma *Greedy*, *Artificial Bee Colony*, *Cheapest Insertion Heuristic* dan *Genetika*. Sebagai hasil dari penelitian, disimpulkan bahwa tidak ada algoritma yang berlaku umum dan dapat digunakan dalam menyelesaikan semua permasalahan (Wiyanti, 2013). Menurut penelitian Pailin dan Tupan dalam membandingkan hasil *Branch and Bound* dan *Cheapest Insertion Heuristic*, disimpulkan bahwa Algoritma *Branch and Bound* memberikan hasil rute yang lebih optimal dibandingkan dengan Algoritma *Cheapest Insertion Heuristic* dengan selisih jarak yang tidak terlalu signifikan (Paillin & Tupan, 2018).

Kasus lain yang berhubungan dengan TSP, salah satunya adalah penentuan rute perjalanan terpendek layanan jemput donasi LAZIS. Berdasarkan beberapa penelitian yang telah dilakukan untuk menyelesaikan TSP, belum ditemukan adanya penyelesaian TSP dengan menggunakan kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*. Sehingga dalam artikel ini dilakukan penyelesaian TSP dengan menggunakan kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* untuk mendapatkan rute perjalanan yang optimal atau paling tidak mendekati optimal.

## KAJIAN TEORI

### GRAF

Pasangan terurut yang memuat himpunan  $V(G)$  dan himpunan  $E(G)$ . Dengan  $V(G)$  adalah himpunan berhingga (tak kosong) dari elemen-elemen yang disebut titik (disimbolkan dengan *noktah*), dan  $E(G)$  adalah himpunan berhingga (mungkin kosong) dari elemen-elemen yang disebut sisi (disimbolkan dengan *kurva sederhana*), sehingga setiap elemen pada  $E(G)$  adalah pasangan tak berurutan dari  $V(G)$  (Budayasa, 2007).

### GRAF RANGKAP

Graf yang terdapat sisi rangkap tetapi tidak terdapat gelung. **Sisi rangkap** adalah apabila pada suatu graf terdapat lebih dari satu sisi yang menghubungkan dua titik  $u$  dan  $v$ , sedangkan **gelung** adalah apabila sebuah titik dihubungkan dengan dirinya sendiri oleh sebuah sisi graf (Budayasa, 2007).

### GRAF HAMILTON

Graf yang memuat siklus Hamilton. Siklus Hamilton adalah sebuah siklus yang memuat semua titik sebuah graf. Siklus adalah sebuah jejak tertutup yang titik awal dan semua titik internalnya berbeda (Budayasa, 2007).

### GRAF BOBOT

Suatu bilangan real yang dikaitkan dengan setiap sisi dari sebuah graf. Bobot graf  $G$ , disimbolkan  $w(G)$ . Panjang lintasan sebuah graf bobot adalah jumlah bobot semua sisi pada lintasan tersebut (Budayasa, 2007).

### GRAF BERARAH

Pasangan terurut yang memuat himpunan  $V(D)$  dan himpunan  $\Gamma(D)$ . Dengan  $V(D)$  adalah himpunan berhingga (tak kosong) dari elemen-elemen yang disebut titik (disimbolkan dengan *noktah*), dan  $\Gamma(D)$  adalah himpunan berhingga (mungkin kosong) dari elemen-elemen yang disebut busur (disimbolkan dengan *kurva sederhana berarah*), sehingga setiap elemen pada  $\Gamma(D)$  adalah pasangan berurutan dari dua titik di  $V(D)$  (Budayasa, 2007).

### TRAVELLING SALESMAN PROBLEM

Permasalahan seorang "sales" yang mengunjungi setiap kota tepat satu kali, kecuali kota awal yang dikunjungi dua kali (Budayasa, 2007).

TSP terdiri dari dua bentuk, yakni *Symmetric TSP* (STSP) dan *Asymmetric TSP* (ATSP). STSP adalah kondisi dimana jarak dari kota A ke kota B sama dengan jarak antara kota B ke kota A, sedangkan untuk ATSP, jarak dari kota A ke kota B tidak sama dengan jarak dari Kota B ke kota A. Dengan  $n$  kota, berturut-turut jumlah kemungkinan rute pada STSP dan ATSP adalah  $\frac{(n-1)!}{2}$  dan  $(n-1)!$  (Greco, 2008).

### ALGORITMA BRANCH AND BOUND

Algoritma yang terdiri dari dua langkah yaitu *branching* dimana titik (kota) yang bertetangga dengan titik awal terlebih dulu dibangkitkan, sedangkan *bounding* adalah menyelesaikan optimasi dengan menentukan nilai batas atas dan batas bawah. Kompleksitas dari algoritma ini adalah  $(n-1)!$ , dengan  $n$  adalah jumlah kota.

Langkah-langkah untuk menjalankan Algoritma *Branch and Bound* sebagai berikut:

1. Menentukan matriks ( $C_{ij}$ ) berukuran  $n \times n$ , dengan  $n$  adalah banyaknya titik yang akan dikunjungi. Matriks  $C_{ij}$  berisikan elemen jarak dari titik  $i$  ke titik  $j$ . Inisiasikan jarak dari titik  $i$  ke titik  $i$  dengan nilai  $\infty$  (tak hingga).
2. Identifikasi nilai terkecil pada tiap baris matriks dan kurangkan tiap baris dengan nilai terkecil yang telah teridentifikasi sebelumnya.
3. Identifikasi nilai terkecil pada tiap kolom matriks dan kurangkan tiap kolom dengan nilai terkecil yang telah teridentifikasi sebelumnya.
4. Jumlahkan semua nilai terkecil baris dan nilai terkecil kolom pada langkah sebelumnya sehingga dihasilkan nilai batas simpul akar  $C(R)$ .
5. Hasil reduksi matriks untuk simpul  $R$  membentuk matriks  $A$ . Anak dari simpul  $R$  (dimisalkan  $S$  dan  $S$  bukan simpul daun), sedemikian hingga sisi  $(i,j)$  pada perjalanan berkoresponden dengan sisi  $(R,S)$  pada pohon status, maka lakukan langkah-langkah pada matriks  $A$  sebagai berikut :
  - Ubah menjadi  $\infty$  (tak hingga) untuk semua elemen pada baris  $i$  dan kolom  $j$ .
  - Ubah menjadi  $\infty$  untuk elemen  $A(j,1)$ .
  - Lakukan kembali reduksi baris dan kolom pada matriks  $A$ , kecuali elemen  $\infty$ .
  - Untuk nilai batas dapat dihitung dengan persamaan :
 
$$C(S) = C(R) + A(i, j) + r$$
 Keterangan :  
 $C(S)$  = nilai bobot minimum simpul  $S$   
 $C(R)$  = nilai bobot minimum simpul  $R$   
 $A(i,j)$  = nilai bobot sisi  $(i,j)$   
 $r$  = jumlah total elemen pengurang
6. melakukan perulangan reduksi matriks hingga membentuk pohon status dengan nilai batas minimum.
7. Dengan mengacu pada simpul yang telah diperoleh, dilakukan penyusunan rute sebagai langkah terakhirnya.

(Auliasari, Kertaningtyas, & Basuki, 2018)

### ALGORITMA CHEAPEST INSERTION HEURISTIC

Algoritma membuat rute jalur terpendek dengan bobot minimal dan secara berturut-turut ditambah dengan tempat baru hingga terbentuk suatu *tour* (perjalanan). Nilai penyisipan minimum dilakukan dengan pemilihan titik baru bersamaan dengan pemilihan sisi.

Langkah-langkah untuk menjalankan Algoritma *Cheapest Insertion Heuristic* sebagai berikut:

1. Rute dimulai dari titik awal dan dihubungkan dengan titik akhir.
2. Membuat *subtour* antara 2 titik. *Subtour* adalah rute perjalanan dari titik asal dan berakhir di titik asal dengan mengunjungi titik-titik yang hendak dituju.
3. Mengganti salah satu arc (rute) dengan kombinasi antar 2 titik, contohnya menyisipkan titik  $k$  yang belum termasuk anggota *subtour* pada arc  $(i,j)$  sehingga menjadi arc  $(i,k)$  dan arc  $(k,j)$ , dengan hasil kombinasi adalah pilihan rute terkecil. *Subtour* terbaru menjadi:
 
$$C(i, k) + C(k, j) - C(i, j)$$
 $C(i, k)$  : jarak dari titik  $i$  ke titik  $k$   
 $C(k, j)$  : jarak dari titik  $k$  ke titik  $j$   
 $C(i, j)$  : jarak dari titik  $i$  ke titik  $j$
4. Ulangi langkah ke 3 hingga semua titik masuk dalam *subtour*.  
(Wilyanto, 2020)

### METODE

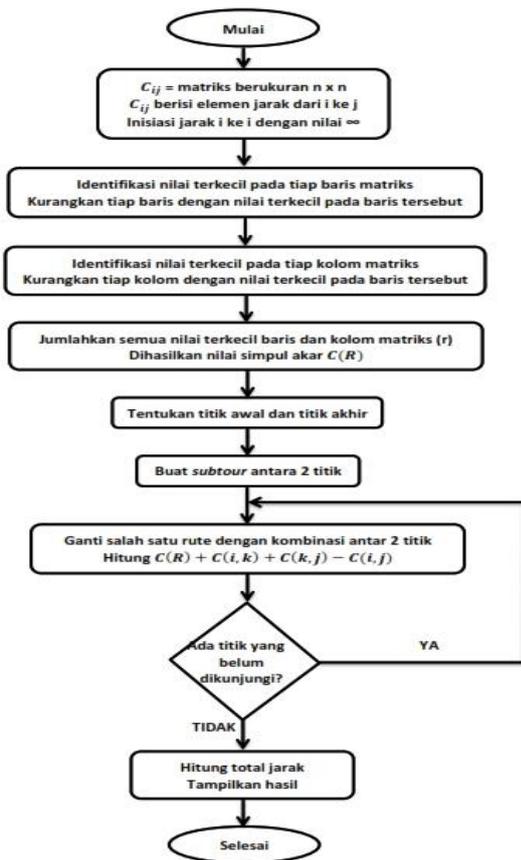
Data yang digunakan dalam artikel ini adalah data sekunder yang didapatkan dari Lembaga Amil Zakat Infaq dan Shadaqoh (LAZIS). Data ini berisikan alamat-alamat donatur dan juga rute reguler layanan jemput donasi LAZIS, yakni Kantor LAZIS → Sri → Reza → Tasya → Maisaroh → Sarmo → Lely → Khusnul → Ayniyatur → Yayuk → Nina → Istiqomah → Ratna → Ainur → Heny → Bayu → Kantor LAZIS dengan total jarak tempuh 65,7 km.

Langkah-langkah dalam artikel ini adalah sebagai berikut: (1) Menganalisis permasalahan dan kemudian menginterpretasikan permasalahan ke dalam bentuk graf. (2) Pencarian dan penentuan jarak dengan menggunakan aplikasi

Google Maps. (3) Memberi bobot pada graf, dengan jarak yang diperoleh dari Google Maps (4) Merepresentasikan bobot graf ke dalam matriks (5) Menyelesaikan permasalahan dengan menggunakan kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*, kemudian menganalisa hasilnya. selanjutnya dilakukan penarikan kesimpulan sebagai hasil akhir dari pemecahan permasalahan.

**KOMBINASI ALGORITMA BRANCH AND BOUND DAN CHEAPEST INSERTION HEURISTIC**

Pada Gambar 1. disajikan *flowchart* kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*.



Gambar 1. Flowchart kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*

Penjelasan rinci mengenai flowchart adalah sebagai berikut :

1. Menentukan matriks  $(C_{ij})$  berukuran  $n \times n$ , dengan  $n$  adalah banyaknya titik yang akan dikunjungi. Matriks  $C_{ij}$  berisikan elemen jarak dari titik  $i$  ke titik  $j$ . Inisiasikan jarak dari titik  $i$  ke titik  $i$  dengan nilai  $\infty$  (tak hingga).

2. Identifikasi nilai terkecil pada tiap baris matriks dan kurangkan tiap baris dengan nilai terkecil yang telah teridentifikasi sebelumnya.
3. Kemudian identifikasikan nilai terkecil pada tiap kolom matriks dan kurangkan tiap kolom dengan nilai terkecil yang telah teridentifikasi sebelumnya.
4. Jumlahkan semua nilai terkecil baris dan nilai terkecil kolom pada langkah 2 dan 3 sehingga dihasilkan nilai batas simpul akar  $C(R)$ .
5. Rute dimulai dari titik awal dan dihubungkan dengan titik akhir.
6. Membuat *subtour* antara 2 titik. *Subtour* adalah rute perjalanan dari titik asal dan berakhir di titik asal dengan mengunjungi titik-titik yang hendak dituju.
7. Mengganti salah satu arc (rute) dengan kombinasi antar 2 titik, contohnya menyisipkan titik  $k$  yang belum termasuk anggota *subtour* pada arc  $(i,j)$  sehingga menjadi arc  $(i,k)$  dan arc  $(k,j)$ , dengan hasil kombinasi adalah pilihan rute terkecil. *Subtour* terbaru menjadi:  

$$C(R) + C(i, k) + C(k, j) - C(i, j)$$

$$C(R) : \text{ nilai simpul akar}$$

$$C(i, k) : \text{ jarak dari titik } i \text{ ke titik } k$$

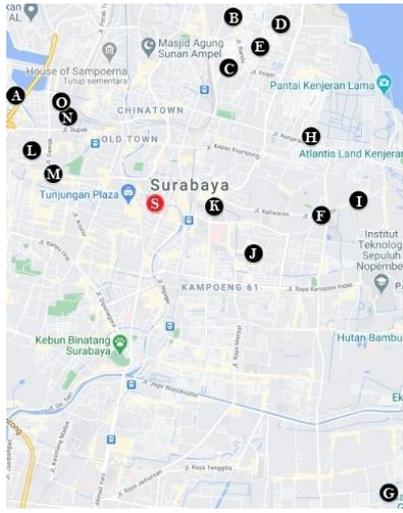
$$C(k, j) : \text{ jarak dari titik } k \text{ ke titik } j$$

$$C(i, j) : \text{ jarak dari titik } i \text{ ke titik } j$$
8. Ulangi langkah ke-7 hingga semua titik masuk dalam *subtour*, dan STOP.

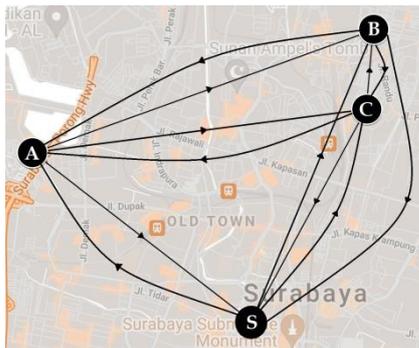
OUTPUT yang dihasilkan dari algoritma ini adalah berupa urutan atau rute yang dimulai dengan titik awal dan melalui setiap titik tujuan dan kembali ke titik semula dengan bobot minimal (mendekati minimal).

**HASIL DAN PEMBAHASAN**

Dalam mencari dan menentukan rute perjalanan layanan jemput donasi LAZIS dengan menggunakan kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* diperlukan data-data sebagai berikut: (1) Plotting titik lokasi (Gambar 2.) dan bentuk graf rangkap berarah seperti ditunjukkan pada Gambar 3.



Gambar 2. Ilustrasi plotting titik lokasi



Gambar 3. Ilustrasi dalam bentuk graf rangkap berarah

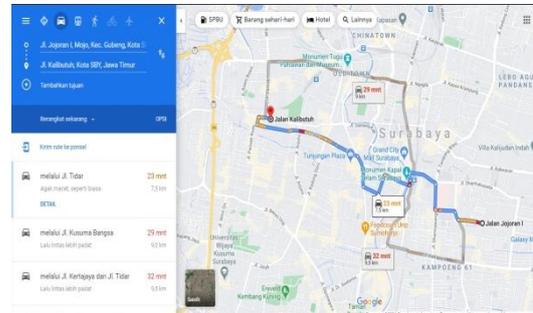
Pada tabel 1. dijabarkan keterangan mengenai titik yang menunjukkan Kantor LAZIS dan lokasi donatur.

Tabel 1A. Kode dan Keterangan lokasi

Kode	Keterangan	
	Nama	Alamat
S	Kantor LAZIS	Jl. Wuni
A	Heny	Jl. Tambak Asri
B	Ayniyatur	Jl. Bulak Banteng Wetan
C	Yayuk	Jl. Sidotopo Wetan Mulia
D	Khusnul	Jl. Kedinding Lor
E	Lely	Jl. Platuk Donomulyo
F	Tasya	Jl. Mulyorejo
G	Bayu	Jl. Medokan Kampung
H	Sarmo	Jl. Lebak Rejo
I	Maisaroh	Jl. Sutorejo
J	Reza	Jl. Jojoran I
K	Sri	Jl. Kedungsroko Tegal
L	Ainur	Jl. Asem Jaya
M	Ratna	Jl. Kalibutih
N	Nina	Jl. Jatisari
O	Istiqomah	Jl. Tuban

Kombinasi Algoritma *Branch And Bound* dan *Cheapest Insertion Heuristic* dalam melakukan pencarian rute terpendek layanan jemput donasi memerlukan suatu data yang berisi jarak antara

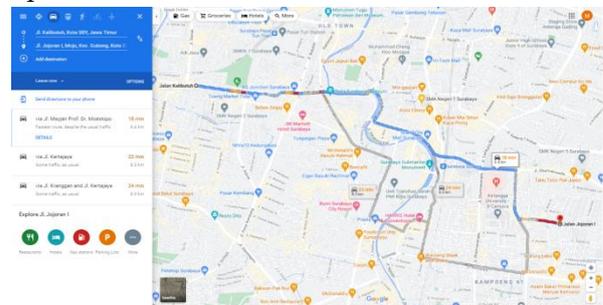
Kantor LAZIS dengan semua donatur. Proses pencarian dan penentuan jarak dilakukan dengan menggunakan bantuan aplikasi *Google Maps*. Berikut ini, pada Gambar 4. disajikan contoh pencarian jarak dengan menggunakan bantuan aplikasi *Google Maps* dari lokasi donatur Reza (Jl. Jojoran I) menuju donatur Ratna (Jl. Kalibutih).



Gambar 4. Rute dari Reza menuju Ratna

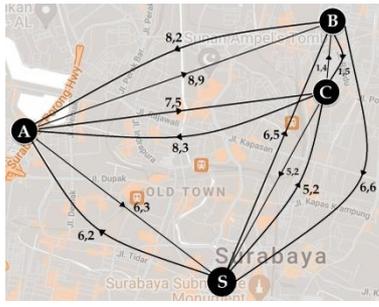
Pada Gambar 4. dihasilkan tiga pilihan rute yang dapat dilalui, yakni melalui Jl. Tidar (7,5 km), melalui Jl. Kusuma Bangsa (9,0 km), dan melalui Jl. Kertajaya dan Jl. Tidar (9,5 km), sehingga akan dipilih rute dengan jarak terpendek yakni melalui Jl. Tidar (7,5 km).

Untuk rute sebaliknya, yakni dari lokasi Ratna (Jl. Kalibutih) menuju Reza (Jl. Jojoran I) (Gambar 5.) dihasilkan juga tiga rute, yakni melalui Jl. Mayjen Prof. Dr. Moestopo (6,4 km), melalui Jl. Kertajaya (8,3 km), dan melalui Jl. Kranggan dan Jl. Kertajaya (8,3 km). Jarak rute sebaliknya yang ditempuh ini tidaklah sama dengan jarak rute sebelumnya dikarenakan tidak semua jalan di Kota Surabaya dapat dilalui dua arah.



Gambar 5. Rute dari Ratna menuju Reza

Gambar 6. Ilustrasi graf rangkap berarah dengan bobot menunjukkan tujuan dari satu titik ke titik lainnya dan bobot menunjukkan jarak antar titik.



Gambar 6. Ilustrasi graf rangkap berarah dengan bobot

Berdasarkan hasil proses pencarian dan penentuan jarak sebelumnya untuk semua titik, selanjutnya dibentuk matriks jarak (Tabel 2).

Tabel 2. Matriks jarak

	S	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
S	0	6,2	6,5	5,2	7,9	6,1	5,5	12,4	6,2	6,7	3,4	1,9	5,4	4,9	6,1	6,3
A	6,3	0	8,9	7,5	10,1	8,5	11	17,2	8,8	11,5	8,9	7,5	2,3	2,5	2,5	2,5
B	6,6	8,2	0	1,5	1,3	1,4	7,8	16,5	5,7	8,4	8	5,9	9,3	8,9	6,7	6,9
C	5,2	8,3	1,4	0	2,1	1,7	6,4	15,1	4,3	7	6,6	4,7	8,5	8,1	6,8	7
D	7,2	9	1,3	2,2	0	1	7,2	15,9	5,2	7,8	8,3	6,2	10	9,6	7,5	7,7
E	6,1	9	1,6	1,7	1	0	6,5	15,8	4,5	7,7	7,5	5,4	9,7	9,3	7,5	7,7
F	5,2	9,6	7,1	6,7	6,7	6,3	0	9,1	2,9	1	3	2,9	9,5	9,1	9,4	9,7
G	13,2	19,3	16,2	15,7	15,8	15,4	9,4	0	12	10	10,2	12,3	17,5	17	18,9	18,9
H	7,5	10,6	6,5	6	6,1	5,7	2,8	11,5	0	3,4	6,5	5,3	10,5	10,1	10,4	10,6
I	6,2	10,6	7,8	7,3	7,3	7	1	9,7	3,3	0	4,4	3,9	10,5	10,1	10,4	10,7
J	3,8	9,5	9	7,6	8,9	8,5	3,1	9,6	6,3	4,4	0	3,1	8	7,5	9,4	9,3
K	2,6	7,2	6,3	5,2	6,3	5,5	4,1	11,2	4,6	4,6	2,1	0	6,7	6,3	7	7,3
L	4,8	2,5	8,5	7,1	10,1	8,1	9,6	15,8	8,4	10,9	7,5	6	0	1,1	2,1	2,1
M	3,7	3,9	8,2	6,8	9,6	7,6	8,5	14,7	7,9	9,8	6,4	4,9	2,2	0	3,5	3,5
N	4,8	2,7	7,5	6,1	8,6	7,1	9,5	16,8	7,4	10	7,5	6	2,4	2,2	0	0,7
O	6	2,7	8,2	7,3	9	8,3	10,6	17	8,6	11,2	8,7	7,1	2,5	2,2	0,7	0

Setelah didapatkan matriks jarak (Tabel 2), maka selanjutnya dilakukan reduksi matriks seperti yang ditunjukkan pada Tabel 3.

Tabel 3. Hasil reduksi matriks jarak

	S	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
S	M	2,9	4,6	3,1	6	4,2	3,6	4	2,4	4,8	1,5	0	3,5	3	4,2	4,4
A	3,5	M	6,6	5	7,8	6,2	8,7	8,4	4,6	9,2	6,6	5,2	0	0,2	0,2	0,2
B	4,8	5,5	M	0	0	0,1	6,5	8,7	2,5	7,1	6,7	4,6	8	7,6	5,4	5,6
C	3,3	5,5	0	M	0,7	0,3	5	7,2	1	5,6	5,2	3,3	7,1	6,7	5,4	5,6
D	5,7	6,6	0,3	1	M	0	6,2	8,4	2,3	6,8	7,3	5,2	9	8,6	6,5	6,7
E	4,6	6,6	0,6	0,5	0	M	5,5	8,3	1,6	6,7	6,5	4,4	8,7	8,3	6,5	6,7
F	3,7	7,2	6,1	5,5	5,7	5,3	M	1,6	0	0	2	1,9	8,5	8,1	8,4	8,7
G	3,3	8,5	6,8	6,1	6,4	6	0	M	0,7	0,6	0,8	2,9	8,1	7,6	9,5	9,5
H	4,2	6,4	3,7	3	3,3	2,9	0	2,2	M	0,6	3,7	2,5	7,7	7,3	7,6	7,8
I	4,7	8,2	6,8	6,1	6,3	6	0	2,2	0,4	M	3,4	2,9	9,5	9,1	9,4	9,7
J	0,2	5	5,9	4,3	5,8	5,4	0	0	1,3	1,3	M	0	4,9	4,4	6,3	6,2
K	0	3,7	4,2	2,9	4,2	3,4	2	2,6	0,6	2,5	0	M	4,6	4,2	4,9	5,2
L	3,2	0	7,4	5,8	9	7	8,5	8,2	5,4	9,8	6,4	4,9	M	0	1	1
M	1	0,3	6	4,4	7,4	5,4	6,3	6	3,8	7,6	4,2	2,7	0	M	1,3	1,3
N	3,6	0,6	6,8	5,2	7,9	6,4	8,8	9,6	4,8	9,3	6,8	5,3	1,7	1,5	M	0
O	4,8	0,6	7,5	6,4	8,3	7,6	9,9	9,8	6	10,5	8	6,4	1,8	1,5	0	M

Dari proses reduksi matriks didapatkan nilai simpul akar  $C(R)$  sebesar 43,5.

Selanjutnya untuk mencari rute terpendek melalui 15 titik tersebut, lakukan langkah-langkah sebagai berikut :

- 1) Ambil titik S sebagai titik asal sekaligus titik akhir.
- 2) Buat *subtour* antara 2 titik. Contoh (S,A)→(A,S)
- 3) Buat tabel yang berisikan titik yang dapat disisipkan ke dalam *subtour* beserta tambahan bobotnya, ditunjukkan dalam Tabel 4.

Tabel 4. *Subtour* pertama

Arc yang akan	Arc yang ditambahkan	Nilai bobot <i>subtour</i>	Total Bobot
(S,A)	(S,B)-(B,A)	$C_R + C_{SB} + C_{BA} - C_{SA}$	50,7
(S,A)	(S,C)-(C,A)	$C_R + C_{SC} + C_{CA} - C_{SA}$	49,2
(S,A)	(S,D)-(D,A)	$C_R + C_{SD} + C_{DA} - C_{SA}$	53,2
(S,A)	(S,E)-(E,A)	$C_R + C_{SE} + C_{EA} - C_{SA}$	51,4
(S,A)	(S,F)-(F,A)	$C_R + C_{SF} + C_{FA} - C_{SA}$	51,4
(S,A)	(S,G)-(G,A)	$C_R + C_{SG} + C_{GA} - C_{SA}$	53,1
(S,A)	(S,H)-(H,A)	$C_R + C_{SH} + C_{HA} - C_{SA}$	49,4
(S,A)	(S,I)-(I,A)	$C_R + C_{SI} + C_{IA} - C_{SA}$	53,6
(S,A)	(S,J)-(J,A)	$C_R + C_{SJ} + C_{JA} - C_{SA}$	47,1
(S,A)	(S,K)-(K,A)	$C_R + C_{SK} + C_{KA} - C_{SA}$	44,3
(S,A)	(S,L)-(L,A)	$C_R + C_{SL} + C_{LA} - C_{SA}$	44,1
(S,A)	(S,M)-(M,A)	$C_R + C_{SM} + C_{MA} - C_{SA}$	43,9
(S,A)	(S,N)-(N,A)	$C_R + C_{SN} + C_{NA} - C_{SA}$	45,4
(S,A)	(S,O)-(O,A)	$C_R + C_{SO} + C_{OA} - C_{SA}$	45,6
(A,S)	(A,B)-(B,S)	$C_R + C_{AB} + C_{BS} - C_{AS}$	51,4
(A,S)	(A,C)-(C,S)	$C_R + C_{AC} + C_{CS} - C_{AS}$	48,3
(A,S)	(A,D)-(D,S)	$C_R + C_{AD} + C_{DS} - C_{AS}$	53,5
(A,S)	(A,E)-(E,S)	$C_R + C_{AE} + C_{ES} - C_{AS}$	50,8
(A,S)	(A,F)-(F,S)	$C_R + C_{AF} + C_{FS} - C_{AS}$	52,4
(A,S)	(A,G)-(G,S)	$C_R + C_{AG} + C_{GS} - C_{AS}$	51,7
(A,S)	(A,H)-(H,S)	$C_R + C_{AH} + C_{HS} - C_{AS}$	48,8
(A,S)	(A,I)-(I,S)	$C_R + C_{AI} + C_{IS} - C_{AS}$	53,9
(A,S)	(A,J)-(J,S)	$C_R + C_{AJ} + C_{JS} - C_{AS}$	46,8
(A,S)	(A,K)-(K,S)	$C_R + C_{AK} + C_{KS} - C_{AS}$	45,2
(A,S)	(A,L)-(L,S)	$C_R + C_{AL} + C_{LS} - C_{AS}$	43,2
(A,S)	(A,M)-(M,S)	$C_R + C_{AM} + C_{MS} - C_{AS}$	41,2
(A,S)	(A,N)-(N,S)	$C_R + C_{AN} + C_{NS} - C_{AS}$	43,8
(A,S)	(A,O)-(O,S)	$C_R + C_{AO} + C_{OS} - C_{AS}$	45

Dari Tabel 4. diperoleh bobot terkecil apabila titik M disisipkan pada *arc* (A,S) sehingga menjadi *arc* (A,M) dan *arc* (M,S), sehingga *subtour* yang baru menjadi :

$$(S,A) \rightarrow (A,M) \rightarrow (M,S)$$

- 4) Lakukan kembali penyisipan titik untuk titik-titik yang belum masuk kedalam *subtour*.

Tabel 5. *Subtour* kedua

Arc yang akan	Arc yang ditambahkan	Nilai bobot <i>subtour</i>	Total Bobot
(S,A)	(S,B)-(B,A)	$C_R + C_{SB} + C_{BA} - C_{SA}$	48,4
(S,A)	(S,C)-(C,A)	$C_R + C_{SC} + C_{CA} - C_{SA}$	46,9
(S,A)	(S,D)-(D,A)	$C_R + C_{SD} + C_{DA} - C_{SA}$	50,9
(S,A)	(S,E)-(E,A)	$C_R + C_{SE} + C_{EA} - C_{SA}$	49,1
(S,A)	(S,F)-(F,A)	$C_R + C_{SF} + C_{FA} - C_{SA}$	49,1
(S,A)	(S,G)-(G,A)	$C_R + C_{SG} + C_{GA} - C_{SA}$	50,8
(S,A)	(S,H)-(H,A)	$C_R + C_{SH} + C_{HA} - C_{SA}$	47,1
(S,A)	(S,I)-(I,A)	$C_R + C_{SI} + C_{IA} - C_{SA}$	51,3
(S,A)	(S,J)-(J,A)	$C_R + C_{SJ} + C_{JA} - C_{SA}$	44,8
(S,A)	(S,K)-(K,A)	$C_R + C_{SK} + C_{KA} - C_{SA}$	42
(S,A)	(S,L)-(L,A)	$C_R + C_{SL} + C_{LA} - C_{SA}$	41,8
(S,A)	(S,N)-(N,A)	$C_R + C_{SN} + C_{NA} - C_{SA}$	43,1
(S,A)	(S,O)-(O,A)	$C_R + C_{SO} + C_{OA} - C_{SA}$	43,3
(A,M)	(A,B)-(B,M)	$C_R + C_{AB} + C_{BM} - C_{AM}$	55,2
(A,M)	(A,C)-(C,M)	$C_R + C_{AC} + C_{CM} - C_{AM}$	52,7
(A,M)	(A,D)-(D,M)	$C_R + C_{AD} + C_{DM} - C_{AM}$	57,4
(A,M)	(A,E)-(E,M)	$C_R + C_{AE} + C_{EM} - C_{AM}$	55,5
(A,M)	(A,F)-(F,M)	$C_R + C_{AF} + C_{FM} - C_{AM}$	57,8
(A,M)	(A,G)-(G,M)	$C_R + C_{AG} + C_{GM} - C_{AM}$	57
(A,M)	(A,H)-(H,M)	$C_R + C_{AH} + C_{HM} - C_{AM}$	52,9
(A,M)	(A,I)-(I,M)	$C_R + C_{AI} + C_{IM} - C_{AM}$	59,3
(A,M)	(A,J)-(J,M)	$C_R + C_{AJ} + C_{JM} - C_{AM}$	52

(A,M)	(A,K)-(K,M)	$C_R + C_{AK} + C_{KM} - C_{AM}$	50,4
(A,M)	(A,L)-(L,M)	$C_R + C_{AL} + C_{LM} - C_{AM}$	41
(A,M)	(A,N)-(N,M)	$C_R + C_{AN} + C_{NM} - C_{AM}$	42,7
(A,M)	(A,O)-(O,M)	$C_R + C_{AO} + C_{OM} - C_{AM}$	42,7
(M,S)	(M,B)-(B,S)	$C_R + C_{MB} + C_{BS} - C_{MS}$	51
(M,S)	(M,C)-(C,S)	$C_R + C_{MC} + C_{CS} - C_{MS}$	47,9
(M,S)	(M,D)-(D,S)	$C_R + C_{MD} + C_{DS} - C_{MS}$	53,3
(M,S)	(M,E)-(E,S)	$C_R + C_{ME} + C_{ES} - C_{MS}$	50,2
(M,S)	(M,F)-(F,S)	$C_R + C_{MF} + C_{FS} - C_{MS}$	50,2
(M,S)	(M,G)-(G,S)	$C_R + C_{MG} + C_{GS} - C_{MS}$	49,5
(M,S)	(M,H)-(H,S)	$C_R + C_{MH} + C_{HS} - C_{MS}$	48,2
(M,S)	(M,I)-(I,S)	$C_R + C_{MI} + C_{IS} - C_{MS}$	52,5
(M,S)	(M,J)-(J,S)	$C_R + C_{MJ} + C_{JS} - C_{MS}$	44,6
(M,S)	(M,K)-(K,S)	$C_R + C_{MK} + C_{KS} - C_{MS}$	42,9
(M,S)	(M,L)-(L,S)	$C_R + C_{ML} + C_{LS} - C_{MS}$	43,4
(M,S)	(M,N)-(N,S)	$C_R + C_{MN} + C_{NS} - C_{MS}$	45,1
(M,S)	(M,O)-(O,S)	$C_R + C_{MO} + C_{OS} - C_{MS}$	46,3

Dari Tabel 5. diperoleh bobot terkecil apabila *arc* (A,M) diganti dengan *arc* (A,L) dan *arc* (L,M), sehingga *subtour* yang baru menjadi :

$$(S,A) \rightarrow (A,L) \rightarrow (L,M) \rightarrow (M,S)$$

- Lakukan kembali penyisipan titik untuk titik-titik yang belum masuk kedalam *subtour*.

Tabel 6. *Subtour* ketiga

Arc yang akan	Arc yang ditambahkan	Nilai bobot <i>subtour</i>	Total Bobot
(S,A)	(S,B)-(B,A)	$C_R + C_{SB} + C_{BA} - C_{SA}$	48,2
(S,A)	(S,C)-(C,A)	$C_R + C_{SC} + C_{CA} - C_{SA}$	46,7
(S,A)	(S,D)-(D,A)	$C_R + C_{SD} + C_{DA} - C_{SA}$	50,7
(S,A)	(S,E)-(E,A)	$C_R + C_{SE} + C_{EA} - C_{SA}$	48,9
(S,A)	(S,F)-(F,A)	$C_R + C_{SF} + C_{FA} - C_{SA}$	48,9
(S,A)	(S,G)-(G,A)	$C_R + C_{SG} + C_{GA} - C_{SA}$	50,6
(S,A)	(S,H)-(H,A)	$C_R + C_{SH} + C_{HA} - C_{SA}$	46,9
(S,A)	(S,I)-(I,A)	$C_R + C_{SI} + C_{IA} - C_{SA}$	51,1
(S,A)	(S,J)-(J,A)	$C_R + C_{SJ} + C_{JA} - C_{SA}$	44,6
(S,A)	(S,K)-(K,A)	$C_R + C_{SK} + C_{KA} - C_{SA}$	41,8
(S,A)	(S,N)-(N,A)	$C_R + C_{SN} + C_{NA} - C_{SA}$	42,9
(S,A)	(S,O)-(O,A)	$C_R + C_{SO} + C_{OA} - C_{SA}$	43,1
(A,L)	(A,B)-(B,L)	$C_R + C_{AB} + C_{BL} - C_{AL}$	55,6
(A,L)	(A,C)-(C,L)	$C_R + C_{AC} + C_{CL} - C_{AL}$	53,1
(A,L)	(A,D)-(D,L)	$C_R + C_{AD} + C_{DL} - C_{AL}$	57,8
(A,L)	(A,E)-(E,L)	$C_R + C_{AE} + C_{EL} - C_{AL}$	55,9
(A,L)	(A,F)-(F,L)	$C_R + C_{AF} + C_{FL} - C_{AL}$	58,2
(A,L)	(A,G)-(G,L)	$C_R + C_{AG} + C_{GL} - C_{AL}$	57,5
(A,L)	(A,H)-(H,L)	$C_R + C_{AH} + C_{HL} - C_{AL}$	53,3
(A,L)	(A,I)-(I,L)	$C_R + C_{AI} + C_{IL} - C_{AL}$	59,7
(A,L)	(A,J)-(J,L)	$C_R + C_{AJ} + C_{JL} - C_{AL}$	52,5
(A,L)	(A,K)-(K,L)	$C_R + C_{AK} + C_{KL} - C_{AL}$	50,8
(A,L)	(A,N)-(N,L)	$C_R + C_{AN} + C_{NL} - C_{AL}$	42,9
(A,L)	(A,O)-(O,L)	$C_R + C_{AO} + C_{OL} - C_{AL}$	43
(L,M)	(L,B)-(B,M)	$C_R + C_{LB} + C_{BM} - C_{LM}$	56
(L,M)	(L,C)-(C,M)	$C_R + C_{LC} + C_{CM} - C_{LM}$	53,5
(L,M)	(L,D)-(D,M)	$C_R + C_{LD} + C_{DM} - C_{LM}$	58,6
(L,M)	(L,E)-(E,M)	$C_R + C_{LE} + C_{EM} - C_{LM}$	56,3
(L,M)	(L,F)-(F,M)	$C_R + C_{LF} + C_{FM} - C_{LM}$	57,6
(L,M)	(L,G)-(G,M)	$C_R + C_{LG} + C_{GM} - C_{LM}$	56,8
(L,M)	(L,H)-(H,M)	$C_R + C_{LH} + C_{HM} - C_{LM}$	53,7
(L,M)	(L,I)-(I,M)	$C_R + C_{LI} + C_{IM} - C_{LM}$	59,9
(L,M)	(L,J)-(J,M)	$C_R + C_{LJ} + C_{JM} - C_{LM}$	51,8
(L,M)	(L,K)-(K,M)	$C_R + C_{LK} + C_{KM} - C_{LM}$	50,1
(L,M)	(L,N)-(N,M)	$C_R + C_{LN} + C_{NM} - C_{LM}$	43,5
(L,M)	(L,O)-(O,M)	$C_R + C_{LO} + C_{OM} - C_{LM}$	43,5
(M,S)	(M,B)-(B,S)	$C_R + C_{MB} + C_{BS} - C_{MS}$	50,8
(M,S)	(M,C)-(C,S)	$C_R + C_{MC} + C_{CS} - C_{MS}$	47,7
(M,S)	(M,D)-(D,S)	$C_R + C_{MD} + C_{DS} - C_{MS}$	53,1
(M,S)	(M,E)-(E,S)	$C_R + C_{ME} + C_{ES} - C_{MS}$	50
(M,S)	(M,F)-(F,S)	$C_R + C_{MF} + C_{FS} - C_{MS}$	50
(M,S)	(M,G)-(G,S)	$C_R + C_{MG} + C_{GS} - C_{MS}$	49,3
(M,S)	(M,H)-(H,S)	$C_R + C_{MH} + C_{HS} - C_{MS}$	48
(M,S)	(M,I)-(I,S)	$C_R + C_{MI} + C_{IS} - C_{MS}$	52,3
(M,S)	(M,J)-(J,S)	$C_R + C_{MJ} + C_{JS} - C_{MS}$	44,4
(M,S)	(M,K)-(K,S)	$C_R + C_{MK} + C_{KS} - C_{MS}$	42,7
(M,S)	(M,N)-(N,S)	$C_R + C_{MN} + C_{NS} - C_{MS}$	44,9
(M,S)	(M,O)-(O,S)	$C_R + C_{MO} + C_{OS} - C_{MS}$	46,1

Dari Tabel 5. diperoleh bobot terkecil apabila *arc* (S,A) diganti dengan *arc* (S,K) dan *arc* (K,A), sehingga *subtour* yang baru menjadi :

$$(S,K) \rightarrow (K,A) \rightarrow (A,L) \rightarrow (L,M) \rightarrow (M,S)$$

- Lakukan kembali penyisipan titik hingga semua titik masuk dalam *subtour*.

Berdasarkan langkah-langkah yang telah dilakukan, diperoleh rute terpendek untuk mengunjungi ke 15 titik yakni :

$$(S,K) \rightarrow (K,J) \rightarrow (J,G) \rightarrow (G,F) \rightarrow (F,I) \rightarrow (I,H) \rightarrow (H,D) \rightarrow (D,E) \rightarrow (E,C) \rightarrow (C,B) \rightarrow (B,O) \rightarrow (O,N) \rightarrow (N,A) \rightarrow (A,L) \rightarrow (L,M) \rightarrow (M,S)$$

dengan total jarak 54,9 km.

## PENUTUP

## SIMPULAN

Penentuan rute terpendek layanan jemput donasi LAZIS dengan menerapkan kombinasi Algoritma Branch and Bound dan Cheapest Insertion Heuristic dilakukan dengan cara merepresentasikan lokasi dan rute layanan jemput donasi ke dalam bentuk graf berarah dengan bobot. Maka dapat disimpulkan untuk rute perjalanannya adalah rute melalui Kantor LAZIS → Sri → Reza → Bayu → Tasya → Maisaroh → Sarmo → Khusnul → Lely → Yayuk → Ayniyatur → Istiqomah → Nina → Heny → Ainur → Ratna → Kantor LAZIS dengan total jarak 54,9 km. Hasil total jarak tempuh yang didapatkan lebih optimal jika dibandingkan dengan rute yang telah dimiliki LAZIS sebelumnya, dimana jarak tempuh rute sebelumnya adalah 65,7 km. Kelebihan dari algoritma ini adalah untuk kasus TSP dengan jumlah kota yang banyak masih stabil digunakan sedangkan kekurangan dari algoritma ini adalah semakin banyak kota tentu semakin berpengaruh pada waktu komputasi.

## SARAN

- Dengan prosedur ini, diasumsikan bahwa model jaringan jalan berupa graph bobot berarah terhubung kuat (jika ini tdk dipenuhi, prosedur TIDAK dapat dijalankan).
- Dari prosedur ini, belum dapat dibuktikan bahwa hasil yang diperoleh merupakan solusi paling optimal, akan tetapi masih merupakan solusi pendekatan.
- Dalam tulisan ini, analisis kompleksitas prosedur tidak dilakukan.

Ketiga poin tersebut di atas, merupakan keterbatasan dalam tulisan ini. Sehingga diharapkan hal-hal tersebut di atas dapat dilibatkan dalam pengembangan artikel selanjutnya.

#### DAFTAR PUSTAKA

- Auliasari, K., Kertaningtyas, M., & Basuki, D. W. (2018). Optimalisasi Rute Distribusi Produk Menggunakan Metode Traveling Salesman Problem. *Jurnal Sains, Teknologi dan Industri*, 1-9.
- Budayasa, I. K. (2007). *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press.
- Greco, F. (2008). *Travelling Salesman Problem*. Perugia: In-Teh.
- Kusrini, & Istiyanto, J. E. (2007). Penyelesaian Travelling Salesman Problem dengan Algoritma Cheapest Insertion Heuristic dan Basis Data. *Jurnal Informatika*, 109-114.
- Paillin, D. B., & Tupan, J. M. (2018). Pemecahan Travelling Salesman Problem Menggunakan Teknik Branch and Bound dan Cheapest Insertion Heuristic (Studi Kasus : PT. Paris Jaya Mandiri - Ambon). *Seminar dan Konferensi Nasional IDEC*.
- Wilyanto. (2020). *Penentuan Rute Perjalanan untuk Perencanaan Wisata di Kota Medan dengan Algoritma Cheapest Insertion Heuristics*. Medan: Universitas Sumatera Utara.
- Wiyanti, D. T. (2013). Algoritma Optimasi untuk Penyelesaian Travelling Salesman Problem. *Jurnal Transformatika*, 1-6.