



Optimasi Rute Pendistribusian Barang Menggunakan Kombinasi *Algoritma Branch and Bound* dan *Cheapest Insertion Heuristic*

Whennie Youngger Oeitama¹, Whannie Youngger Oeitama²,
Flora Frisilia Sitandi³, Syamsuddin Mas'ud^{4*}

^{1,2,3,4} Universitas Negeri Makassar

*syamsuddinm@unm.ac.id

ABSTRAK

Permasalahan yang sering terjadi dalam proses pendistribusian barang adalah belum optimalnya rute pendistribusian sehingga biaya yang dikeluarkan meningkat dan durasi perjalanan menjadi lebih lama. Hal ini dapat diselesaikan dengan mencari jalur terpendek yang dapat dilalui atau yang dikenal dengan istilah *Traveling Salesman Problem* (TSP). Penelitian ini bertujuan untuk menentukan rute pendistribusian barang yang optimal menggunakan kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*. Data yang digunakan berupa nama lokasi dan jarak antar lokasi yang sebelumnya telah dihimpun oleh Putra BJ Bangun dalam penelitiannya yang berjudul *Penyelesaian Traveling Salesman Problem (TSP) dengan metode Branch and Bound (Aplikasi Permasalahan Pengangkutan Barang Kantor Pos Palembang)*. Hasil penelitian menunjukkan bahwa rute optimal untuk pendistribusian barang di kantor pos Kota Palembang, menggunakan kombinasi kedua algoritma, adalah: KPRK Palembang → KPC Kapt A. Rivai → KPC Pakjo → KPC Talang Ratu → KPC Sukarami → KPC Alang Lebar → KPC Sekip → KPC Cinde → KPRK Palembang, dengan total jarak 24,3 km. Rute ini dapat menjadi alternatif bagi *salesman* untuk mengunjungi beberapa KPC dan kembali ke KPRK, dengan biaya dan waktu yang lebih hemat karena merupakan rute terpendek. Selain itu, kombinasi algoritma ini lebih efisien dan sederhana dalam hal langkah pengerjaan dan waktu komputasi dibandingkan penggunaan algoritma *Branch and Bound* saja.

Kata Kunci: *Traveling Salesman Problem* (TSP), *Branch and Bound*, *Cheapest Insertion Heuristic*, Kombinasi Algoritma, Optimal.

ABSTRACT

The problem that often occurs in the process of distributing goods is that the distribution route is not optimal which results in higher costs and longer travel times. This can be solved by finding the shortest path that can be passed or widely recognized as the *Traveling Salesman Problem* (TSP). This study aims to determine the optimal distribution route for goods using a combination of the *Branch and Bound* and *Cheapest Insertion Heuristic* algorithms. The data used are in the form of location names and distances between locations that have been collected by Putra BJ Bangun in his research entitled *Solving the Traveling Salesman Problem (TSP) with the Branch and Bound Method (Application of the Palembang Post Office Goods Transportation Problem)*. The results of the research indicate that the optimal route for the distribution of goods at the Palembang City Post Office, using a combination of both algorithms, is: KPRK Palembang → KPC Kapt A. Rivai → KPC Pakjo → KPC Talang Ratu → KPC Sukarami → KPC Alang Lebar → KPC Sekip → KPC Cinde → KPRK Palembang, spanning a total of 24.3 km. This route can be an alternative for *salesmen* to visit several KPCs and return to KPRK, with more efficient costs and time because it is the shortest route. In addition, this combination of algorithms is more efficient and simpler in terms of processing steps and computing time compared to using the *Branch and Bound* algorithm.

Keywords: *Traveling Salesman Problem* (TSP), *Branch and Bound*, *Cheapest Insertion Heuristic*, *Algorithm Combination*, *Optimal*.

1. PENDAHULUAN

Distribusi secara umum merupakan proses yang mencakup pengangkutan dan pengiriman barang sehingga dapat sampai pada konsumen (Sulistiyorini & Mahmudy, 2015). Salah satu permasalahan yang kerap terjadi pada suatu proses pendistribusian barang ataupun produk yaitu rute pendistribusian yang belum optimal sehingga biaya yang harus dikeluarkan semakin besar dan waktu tempuh akan semakin lama (Auliasari et al., 2018). Permasalahan tersebut dapat diselesaikan dengan optimasi seperti mencari jalur terpendek dari jalur pendistribusian (Sihombing & Ahyaningsih, 2023). Hal ini juga dikenal dengan istilah *Traveling Salesman Problem* (TSP). TSP adalah suatu permasalahan untuk menentukan rute dengan jarak tempuh minimum dari titik awal menuju semua titik yang akan dituju dan kembali ke titik awal dimana setiap dair titik tujuan hanya didatangi sebanyak satu kali untuk mendapatkan rute distribusi yang optimal (Paillin & Sosebeko, 2017). Solusi dari TSP ini dapat ditemukan dengan pemodelan melalui graf (Nur & Rahadjeng, 2021).

Terdapat berbagai algoritma yang bisa digunakan untuk menyelesaikan masalah yang melibatkan TSP, diantaranya algoritma Genetika dalam penelitian oleh Rohman et al. (2020) yang menghasilkan nilai yang optimum, tetapi jika nilai n lokasi semakin besar maka waktu proses algoritma semakin lama (Saputra, 2022). Lakutu et al. (2023) menemukan bahwa algoritma Dijkstra memberikan hasil optimal dalam penentuan rute pengiriman barang di Kantor Pos Gorontalo dengan mempertimbangkan jarak terpendek. Selain itu, Usman & Oktiarso (2018) menggunakan algoritma Greedy untuk pengoptimalan biaya pengiriman dan durasi waktu tempuh pendistribusian produk obat dan barang. Adapun, algoritma Prim juga dapat digunakan untuk menentukan rute terpendek seperti pada masalah pendistribusian naskah soal USBN di daerah Sleman (Rahmadi & Sandariria, 2023).

Selain beberapa algoritma di atas, terdapat juga algoritma *Cheapest Insertion Heuristic* dan *Branch and Bound* yang banyak diterapkan untuk menyelesaikan persoalan TSP. Penelitian oleh Saleh & Helmi (2015) memanfaatkan algoritma *Cheapest Insertion Heuristic* dalam menyelesaikan permasalahan TSP pada kasus pendistribusian barang dan diperoleh penurunan total jarak tempuh. Saputra (2022) dalam penelitiannya terkait optimasi rute distribusi kurir menggunakan algoritma *Branch and Bound*, menemukan bahwa dengan menggunakan algoritma tersebut maka diperoleh rute distribusi terpendek sehingga menghasilkan waktu tempuh minimum. Hal itu selaras dengan hasil kajian yang dilakukan oleh Bangun et al. (2015), yang menunjukkan bahwa penggunaan algoritma *Branch and Bound* pada persoalan pengangkutan barang Kantor Pos Palembang mampu menemukan rute perjalanan terpendek.

Penelitian oleh Pailin & Tupan (2018) membandingkan hasil algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*, dan menemukan bahwa algoritma *Branch and Bound* memberikan hasil rute yang lebih optimal dengan selisih jarak yang tidak terlalu signifikan bila dikomparasikan dengan algoritma *Cheapest Insertion Heuristic*. Hal serupa juga diperoleh dari hasil penelitian oleh Lattan et al. (2021) yang memperlihatkan bahwa algoritma *Branch and Bound* menghasilkan rute yang lebih baik (optimal) dibandingkan dengan menggunakan algoritma *Cheapest Insertion Heuristic*. Berdasarkan hal tersebut, penulisan ini menggunakan data yang didapatkan oleh Bangun et al. (2015) yang menggunakan algoritma *Branch and Bound*. Namun, penulis mencoba mencari rute yang

lebih baik dengan menggunakan kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*, seperti yang dilakukan oleh Nur & Rahadjeng (2021) dalam penelitian tentang penentuan rute perjalanan terpendek layanan jemput donasi LAZIS, dan Safitri et al. (2024) dalam penelitian tentang penentuan rute terpendek distribusi kurir paket JNT di Kecamatan Batang Cenaku. Hasil penelitian menunjukkan bahwa diperoleh rute distribusi yang optimal dengan jarak tempuh minimum menggunakan kombinasi kedua algoritma tersebut (Nur & Rahadjeng, 2021; Safitri et al., 2024). Oleh karena itu dalam tulisan ini akan ditentukan rute pendistribusian barang yang optimal (termasuk dari sisi pencarian rute) menggunakan kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*.

2. METODE PENELITIAN

Penelitian ini merupakan penelitian kepustakaan (*library research*), yaitu studi yang mengkaji berbagai referensi dari hasil kajian (penelitian) terdahulu yang berkenaan dengan pembahasan pada penelitian ini, kemudian dijadikan sebagai landasan teori (Sari & Asmendri, 2020). Data yang digunakan pada kajian ini merupakan data sekunder, yaitu data yang sebelumnya telah dikumpulkan peneliti terdahulu terkait penentuan rute pengiriman barang di kantor pos dengan memanfaatkan algoritma *Branch and Bound* (Bangun et al., 2015). Data yang dimaksud diantaranya adalah data nama lokasi dan jarak antara setiap lokasi distribusi barang. Pada kajian ini, metode yang digunakan oleh peneliti terdahulu dalam menentukan rute distribusi diganti dengan menggunakan algoritma lain, yaitu kombinasi dari algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*.

Proses pengumpulan data dilakukan dengan mengumpulkan data berupa lokasi dan jarak antar lokasi berdasarkan penelitian sebelumnya. Selanjutnya, data tersebut dianalisis menggunakan metode kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* untuk menentukan rute yang optimal. Adapun langkah-langkah yang ditempuh dalam kajian ini meliputi: (1) menguraikan konsep dari algoritma *Branch and Bound*, algoritma *Cheapest Insertion Heuristic*, dan penggabungan (kombinasi) kedua algoritma tersebut; (2) mengimplementasikan penggabungan algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* dalam suatu kasus; (3) melakukan analisis perbandingan rute dengan algoritma *Branch and Bound* dan kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* sesuai dari hasil penyesuaian kasus yang ada; dan (4) membuat kesimpulan terkait rute pendistribusian optimal menggunakan kombinasi kedua algoritma.

3. HASIL DAN PEMBAHASAN

3.1. Distribusi

Distribusi adalah aktivitas pengiriman barang yang diproduksi perusahaan tertentu kepada konsumen (Sihombing & Ahyaningsih, 2023). Tujuan utama dari aktivitas ini adalah untuk menjamin bahwa barang tiba di tempat yang dituju secara efisien, tepat waktu, dan menggunakan biaya seminimal mungkin (Anassuka et al., 2022). Distribusi juga sangat penting agar perusahaan dapat memperoleh profit atau keuntungan yang maksimal dari barang hasil produksi (Sihombing & Ahyaningsih, 2023). Salah satu aspek penting dari kegiatan distribusi adalah rute pendistribusian, yaitu jalur yang ditempuh atau dilalui oleh kendaraan distribusi untuk mengantarkan barang ke berbagai tujuan (Paillin & Tupan, 2018).

3.2. Optimasi Rute

Optimasi diambil dari kata optimal dan memiliki arti terbaik, tertinggi, dan paling menguntungkan. Optimasi didefinisikan sebagai suatu upaya atau cara untuk mendapatkan penyelesaian (solusi) yang paling baik dari berbagai alternatif penyelesaian yang layak (Sihombing & Ahyaningsih, 2023). Arora (2015) memaparkan bahwa penyelesaian yang layak adalah penyelesaian yang mencocoki setiap kendala pada permasalahan optimasi sedangkan penyelesaian yang paling baik adalah solusi yang dapat membuat biaya pemrosesan menjadi minimal atau efisiensi dari sistem menjadi maksimal (Sihombing & Ahyaningsih, 2023). Adapun optimasi rute adalah proses untuk menentukan rute terbaik dari suatu perjalanan atau pengiriman barang dengan tujuan dapat meminimalkan jarak tempuh, waktu perjalanan, dan biaya (Anassuka et al., 2022). Optimasi rute biasanya menggunakan beberapa algoritma agar dapat memberikan solusi yang optimal. Salah satu bentuk permasalahan terkait optimasi yang kerap kali ditemui di keseharian seseorang maupun perusahaan yaitu masalah *Traveling Salesman Problem* (TSP).

3.3. Graf

Budayasa (2007) menjelaskan bahwa secara matematis, graf didefinisikan sebagai pasangan terurut yang memuat himpunan $V(G)$ dan himpunan $E(G)$ atau dapat ditulis dengan notasi $G = \{V, E\}$ (Nur & Rahadjeng, 2021). Himpunan $V(G) = \{v_1, v_2, v_3, \dots\}$ adalah himpunan yang berhingga dan tak kosong, dengan anggota berupa objek-objek yang disebut simpul atau titik (*vertex*) dan disimbolkan dengan noktah (Rozi & Multahadah, 2021). Himpunan $E(G) = \{e_1, e_2, e_3, \dots\}$ adalah himpunan hingga yang mungkin saja kosong dari objek-objek yang disebut sisi atau busur (*edge*) dan disimbolkan dalam bentuk kurva sederhana (Nur & Rahadjeng, 2021). Jika setiap titik terhubung ke semua titik lainnya oleh satu sisi maka disebut graf lengkap (Safitri et al., 2024). Di sisi lain, graf yang setiap sisinya mempunyai bobot satuan, dapat berupa jarak, panjang, atau jumlah suatu barang, disebut sebagai graf berbobot (*weighted graph*) (Ningrum et al., 2023). Graf berbobot biasa digunakan untuk mencari lintasan terpendek (Yuliantari & Musabbikhah, 2022). Panjang dari lintasan suatu graf berbobot adalah total bobot dari semua sisi yang ada di lintasan itu (Nur & Rahadjeng, 2021).

3.4. *Traveling Salesman Problem* (TSP)

TSP adalah masalah dalam teori graf dengan tujuan untuk menentukan rute optimal (paling pendek) bagi *salesman* yang harus berkunjung ke setiap titik satu kali, kemudian kembali ke titik awal (Sihombing & Ahyaningsih, 2023; Lattan et al., 2021). Pada konteks ini, titik (*vertex*) dalam graf mewakili kota atau lokasi yang dikunjungi, sedangkan sisi (*edge*) dalam graf melambangkan jalan yang menjadi penghubung antar lokasi kota. Bobot pada sisi mewakili jarak antara dua kota atau lokasi yang berbeda. Ada dua jenis TSP, yaitu *Symmetric TSP* (STSP) dan *Asymmetric TSP* (ATSP) (Nur & Rahadjeng, 2021; Hignasari, 2019). Pada STSP, jarak dari kota A ke kota B adalah sama dengan jarak dari kota B ke kota A. Sementara itu, pada ATSP, jarak dari kota A ke kota B tidak selalu sama dengan jarak dari kota B ke kota A.

Adapun, formulasi model TSP untuk penentuan rute dirumuskan pada (Sihombing & Ahyaningsih, 2023; Paillin & Tupan, 2018). Misalkan C_{ij} merepresentasikan jarak perjalanan

dari titik i ke titik j , dan variabel keputusan X_{ij} merepresentasikan ada atau tidaknya perjalanan dari titik i ke titik j dalam rute, yaitu:

$$X_{ij} = \begin{cases} 1, & \text{jika terdapat perjalanan salesman dari } i \text{ ke } j \\ 0, & \text{jika tidak terdapat perjalanan salesman dari } i \text{ ke } j \end{cases} \quad 3.1$$

Jika Z merupakan fungsi tujuan dari TSP, maka fungsi Z dirumuskan sebagai upaya untuk meminimalkan:

$$Z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij} \quad 3.2$$

dengan batasan,

$$\sum_{i=1}^n X_{ij} = 1 \quad (i = 1, 2, 3, \dots, n), \quad 3.3$$

$$\sum_{j=1}^n X_{ij} = 1 \quad (j = 1, 2, 3, \dots, n), \quad 3.4$$

$i \neq j$.

Persamaan 3.3 dan 3.4 mendeskripsikan bahwa rute perjalanan *salesman* adalah dengan mengunjungi setiap lokasi satu kali dan meninggalkan lokasi tersebut satu kali juga. Untuk menyelesaikan masalah TSP dalam konteks distribusi barang, dua algoritma yang dapat diterapkan, yakni algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* (Nur & Rahadjeng, 2021; Safitri et al., 2024).

3.5. Algoritma *Branch and Bound*

Metode *Branch and Bound* adalah algoritma yang digunakan untuk menemukan solusi optimal atau mendekati optimal dalam masalah optimasi. Prinsip utamanya meliputi “*branching*”, yaitu membagi masalah besar menjadi sub-masalah yang lebih mudah dikelola, serta “*bounding*”, yaitu menetapkan batas bawah dan batas atas untuk mempersempit pencarian solusi optimal (Azzahrha et al., 2021). Langkah-langkah untuk menjalankan algoritma *Branch and Bound* dijelaskan sebagai berikut (Nur & Rahadjeng, 2021; Safitri et al., 2024):

1. Menentukan matriks (C_{ij}) berukuran $n \times n$, dengan n adalah banyaknya titik yang akan dikunjungi oleh *salesman*. Matriks C_{ij} memuat elemen jarak dari titik i ke titik j . Inisiasikan jarak dari titik i ke titik i dengan nilai \sim (tak hingga).
2. Menentukan nilai terkecil pada setiap baris matriks, lalu mengurangi setiap baris dengan nilai terkecil yang telah diidentifikasi.
3. Menentukan nilai terkecil pada setiap kolom matriks, lalu mengurangi setiap kolom dengan nilai terkecil yang telah diidentifikasi.
4. Jumlahkan semua nilai terkecil baris dan nilai terkecil kolom pada langkah 2 dan 3 sehingga diperoleh nilai batas simpul akar $C(R)$.
5. Setelah dilakukan reduksi matriks untuk simpul R , diperoleh matriks A . Anak dari simpul R (dimisalkan S dan S bukan simpul daun), sedemikian sehingga sisi (i, j) pada perjalanan berkoresponden dengan sisi (R, S) pada pohon status, maka lakukan langkah-langkah pada matriks A sebagai berikut:
 - a. Semua elemen pada baris i dan kolom j diubah menjadi \sim (tak hingga)
 - b. Elemen $A(j, 1)$ diubah menjadi \sim (tak hingga)

- c. Lakukan kembali reduksi baris dan kolom pada matriks A , kecuali elemen \sim (tak hingga)
- d. Menghitung nilai batas dengan menggunakan persamaan:

$$C(S) = C(R) + A(i, j) + r \tag{3.5}$$

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 reduksi matriks secara iteratif hingga terbentuk pohon status dengan nilai batas minimum.
- 7. Menyusun rute dengan mengacu pada simpul yang telah diperoleh.

3.6. Algoritma *Cheapest Insertion Heuristic*

Algoritma *Cheapest Insertion Heuristic* adalah metode optimasi rute yang menyusun jalur terpendek dengan menambahkan (menyisipkan) titik-titik baru ke dalam rute yang ada berdasarkan bobot penambahan terkecil. Proses ini dimulai dari siklus kecil dan berlanjut hingga semua titik telah dikunjungi (Aristi, 2014). Langkah-langkah untuk menjalankan algoritma *Cheapest Insertion Heuristic* yaitu sebagai berikut (Nur & Rahadjeng, 2021; Safitri et al., 2024) adalah sebagai berikut.

- 1. Rute dimulai dari titik awal yang terhubung ke titik akhir.
- 2. Membuat *subtour* antara dua titik. *Subtour* adalah rute perjalanan yang dimulai dan berakhir di titik asal dengan mengunjungi titik-titik yang ingin dituju.
- 3. Mengganti sebuah sisi (rute) pada *subtour* tersebut, misalnya sisi (i, j) , dengan kombinasi dua sisi yaitu (i, k) dan (k, j) , di mana titik k belum termasuk dalam *subtour* sehingga dihasilkan kombinasi yang merupakan pilihan rute terkecil. *Subtour* terbaru menjadi

$$C(i, k) + C(k, j) - C(i, j) \tag{3.6}$$

Keterangan:

$C(i, k)$: 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 seluruh titik masuk dalam *subtour*.

3.7. Kombinasi Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*

Algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* dapat digabungkan untuk menghasilkan rute perjalanan yang optimal. Berikut adalah langkah-langkah kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* (Nur & Rahadjeng, 2021; Mas' ud, 2024) adalah sebagai berikut.

- 1. Menentukan matriks (C_{ij}) berukuran $n \times n$, dengan n adalah banyaknya titik yang akan dikunjungi oleh *salesman*. Matriks C_{ij} memuat elemen jarak dari titik i ke titik j . Inisialisasi jarak dari titik i ke titik i dengan nilai \sim (tak hingga).

2. Menentukan nilai terkecil pada setiap baris matriks, lalu mengurangkan setiap baris dengan nilai terkecil yang telah diidentifikasi.
3. Menentukan nilai terkecil pada setiap kolom matriks, lalu mengurangkan setiap kolom dengan nilai terkecil yang telah diidentifikasi.
4. Jumlahkan semua nilai terkecil baris dan nilai terkecil kolom pada langkah 2 dan 3 sehingga diperoleh nilai batas simpul akar $C(R)$.
5. Rute dimulai dari titik awal yang terhubung ke titik akhir.
6. Membuat *subtour* antara dua titik. *Subtour* adalah rute perjalanan yang dimulai dan berakhir di titik asal dengan mengunjungi titik-titik yang ingin dituju.
7. Mengganti sebuah sisi (rute) pada *subtour* tersebut, misalnya sisi (i, j) , dengan kombinasi dua sisi yaitu (i, k) dan (k, j) , di mana titik k belum termasuk dalam *subtour* sehingga dihasilkan kombinasi yang merupakan pilihan rute terkecil. *Subtour* terbaru menjadi:

$$C(R) + C(i, k) + C(k, j) - C(i, j) \quad 3.7$$

Keterangan:

$C(R)$: nilai simpul akar

$C(i, k)$: 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

8. Ulangi langkah ke-7 hingga seluruh titik masuk dalam *subtour*, dan STOP.

3.8. Data Lokasi dan Jarak Tempuh

Berikut disajikan data lokasi dan jarak tempuh antar lokasi berdasarkan data yang diambil dari penelitian Bangun et al. (2015), pada Tabel 3.1.

Tabel 3.1. Daftar Lokasi dan Jarak Antar Lokasi (km)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	2,5	1,5	5	4,5	8	12	6
x_2	2,5	0	2	2,8	3,5	7	10,5	5,5
x_3	1,5	2	0	4	3,5	6,5	10,5	5,5
x_4	5	2,8	4	0	2	5,5	9	4,5
x_5	4,5	3,5	3,5	2	0	3,5	7	2,5
x_6	8	7	6,5	5,5	3,5	0	4	6
x_7	12	10,5	10,5	9	7	4	0	2,5
x_8	6	5,5	5,5	4,5	2,5	6	2,5	0

Keterangan:

x_1 : Kantor Pos Pemeriksa (KPRK) Palembang

x_2 : Kantor Pos Cabang (KPC) Kapt A. Rivai

x_3 : KPC Cinde

x_4 : KPC Pakjo

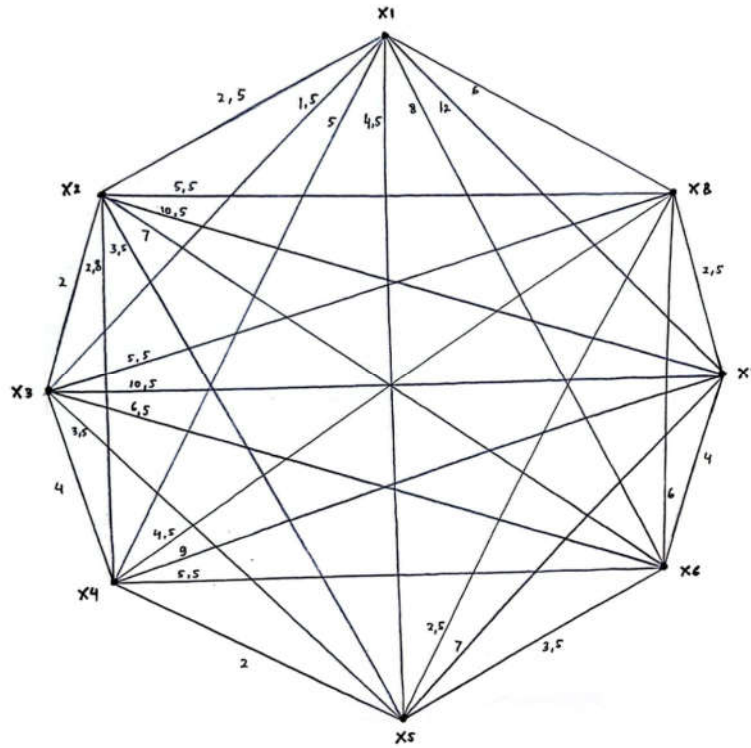
x_5 : KPC Talang Ratu

x_6 : KPC Sukarami

x_7 : KPC Alang Lebar

x_8 : KPC Sekip

Penentuan rute pendistribusian barang dilakukan dengan menjadikan x_1 (Kantor Pos Pemeriksa Palembang) sebagai titik awal sekaligus titik akhir pendistribusian dengan bentuk graf disajikan seperti pada Gambar 3.1.



Gambar 3.1. Graf Pendistribusian Barang di Kantor Pos Kota Palembang

3.9. Penentuan Rute

Penelitian ini memanfaatkan data sekunder dari penelitian Bangun et al. (2015) untuk menentukan rute optimal dengan menggunakan kombinasi metode algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*. Distribusi barang di kantor pos Kota Palembang melibatkan satu Kantor Pos Pemeriksa (KPRK) dan beberapa Kantor Pos Cabang (KPC). Berdasarkan data jarak antar lokasi yang ditampilkan pada Tabel 3.1 dan Gambar 3.1, langkah pertama adalah membentuk matriks (C_{ij}) berukuran 8×8 sebagai berikut.

Tabel 3.2. Matriks Jarak Antar Titik

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	~	2,5	1,5	5	4,5	8	12	6
x_2	2,5	~	2	2,8	3,5	7	10,5	5,5
x_3	1,5	2	~	4	3,5	6,5	10,5	5,5
x_4	5	2,8	4	~	2	5,5	9	4,5
x_5	4,5	3,5	3,5	2	~	3,5	7	2,5
x_6	8	7	6,5	5,5	3,5	~	4	6
x_7	12	10,5	10,5	9	7	4	~	2,5
x_8	6	5,5	5,5	4,5	2,5	6	2,5	~

Langkah kedua adalah mengidentifikasi nilai terkecil dalam setiap baris matriks berdasarkan data yang tertera di Tabel 3.2, kemudian mengurangkan setiap baris dengan nilai terkecil yang telah diidentifikasi. Hasil dari proses ini ditampilkan pada Tabel 3.3 berikut.

Tabel 3.3: Hasil Reduksi Matriks Baris

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	~	1	0	3,5	3	6,5	10,5	4,5
x_2	0,5	~	0	0,8	1,5	5	8,5	3,5
x_3	0	0,5	~	2,5	2	5	9	4
x_4	3	0,8	2	~	0	3,5	7	2,5
x_5	2,5	1,5	1,5	0	~	1,5	5	0,5
x_6	4,5	3,5	3	2	0	~	0,5	2,5
x_7	9,5	8	8	6,5	4,5	1,5	~	0
x_8	3,5	3	3	2	0	3,5	0	~

Langkah ketiga melibatkan penentuan nilai terkecil untuk setiap kolom matriks berdasarkan data dari Tabel 3.3. Selanjutnya, setiap kolom dikurangi dengan nilai terkecil yang telah diidentifikasi. Hasil dari proses ini ditampilkan dalam Tabel 3.4.

Tabel 3.4: Hasil Reduksi Matriks Kolom

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	~	0,5	0	3,5	3	5	10,5	4,5
x_2	0,5	~	0	0,8	1,5	3,5	8,5	3,5
x_3	0	0	~	2,5	2	3,5	9	4
x_4	3	0,3	2	~	0	2	7	2,5
x_5	2,5	1	1,5	0	~	0	5	0,5
x_6	4,5	3	3	2	0	~	0,5	2,5
x_7	9,5	7,5	8	6,5	4,5	0	~	0
x_8	3,5	2,5	3	2	0	2	0	~

Langkah keempat adalah mencari nilai batas simpul akar $C(R)$ dengan menjumlahkan nilai terkecil pada setiap baris dan kolom dari langkah sebelumnya. Proses reduksi matriks menghasilkan $C(R) = 19,5$. Langkah kelima adalah menetapkan rute perjalanan yang dimulai dan diakhiri di titik x_1 , yaitu KPRK Palembang. Setelah menentukan titik awal dan titik akhir, langkah keenam adalah membentuk *subtour* antara dua titik, yaitu $(x_1, x_3) - (x_3, x_1)$ sebagai *subtour* awal. Selanjutnya, dibuat tabel yang berisikan hasil penyisipan titik-titik lain ke dalam *subtour* awal beserta tambahan bobotnya.

Tabel 3.5. Penyisipan pada *Subtour* Pertama

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot <i>Subtour</i>
(x_1, x_3)	$(x_1, x_2) - (x_2, x_3)$	$C(R) + C(x_1, x_2) + C(x_2, x_3) - C(x_1, x_3)$	22,5
(x_1, x_3)	$(x_1, x_4) - (x_4, x_3)$	$C(R) + C(x_1, x_4) + C(x_4, x_3) - C(x_1, x_3)$	27
(x_1, x_3)	$(x_1, x_5) - (x_5, x_3)$	$C(R) + C(x_1, x_5) + C(x_5, x_3) - C(x_1, x_3)$	26
(x_1, x_3)	$(x_1, x_6) - (x_6, x_3)$	$C(R) + C(x_1, x_6) + C(x_6, x_3) - C(x_1, x_3)$	32,5
(x_1, x_3)	$(x_1, x_7) - (x_7, x_3)$	$C(R) + C(x_1, x_7) + C(x_7, x_3) - C(x_1, x_3)$	40,5
(x_1, x_3)	$(x_1, x_8) - (x_8, x_3)$	$C(R) + C(x_1, x_8) + C(x_8, x_3) - C(x_1, x_3)$	29,5

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot <i>Subtour</i>
(x_3, x_1)	$(x_3, x_2) - (x_2, x_1)$	$C(R) + C(x_3, x_2) + C(x_2, x_1) - C(x_3, x_1)$	22,5
(x_3, x_1)	$(x_3, x_4) - (x_4, x_1)$	$C(R) + C(x_3, x_4) + C(x_4, x_1) - C(x_3, x_1)$	27
(x_3, x_1)	$(x_3, x_5) - (x_5, x_1)$	$C(R) + C(x_3, x_5) + C(x_5, x_1) - C(x_3, x_1)$	26
(x_3, x_1)	$(x_3, x_6) - (x_6, x_1)$	$C(R) + C(x_3, x_6) + C(x_6, x_1) - C(x_3, x_1)$	32,5
(x_3, x_1)	$(x_3, x_7) - (x_7, x_1)$	$C(R) + C(x_3, x_7) + C(x_7, x_1) - C(x_3, x_1)$	40,5
(x_3, x_1)	$(x_3, x_8) - (x_8, x_1)$	$C(R) + C(x_3, x_8) + C(x_8, x_1) - C(x_3, x_1)$	29,5

Berdasarkan Tabel 3.5, bobot minimum yang diperoleh adalah 22,5. Hal ini terjadi jika sisi (x_1, x_3) diganti dengan $(x_1, x_2) - (x_2, x_3)$, atau sisi (x_3, x_1) diganti dengan $(x_3, x_2) - (x_2, x_1)$. Karena ada dua kemungkinan dengan jarak terkecil, salah satu dapat dipilih. Misalnya, jika memilih kemungkinan pertama, *subtour* yang baru menjadi $(x_1, x_2) - (x_2, x_3) - (x_3, x_1)$. Langkah berikutnya adalah melakukan kembali penyisipan titik yang belum termasuk dalam *subtour* dengan menggunakan nilai $C(R) = 22,5$.

Tabel 3.6: Penyisipan pada *Subtour* Kedua

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot <i>Subtour</i>
(x_1, x_2)	$(x_1, x_4) - (x_4, x_2)$	$C(R) + C(x_1, x_4) + C(x_4, x_2) - C(x_1, x_2)$	27,8
(x_1, x_2)	$(x_1, x_5) - (x_5, x_2)$	$C(R) + C(x_1, x_5) + C(x_5, x_2) - C(x_1, x_2)$	28
(x_1, x_2)	$(x_1, x_6) - (x_6, x_2)$	$C(R) + C(x_1, x_6) + C(x_6, x_2) - C(x_1, x_2)$	35
(x_1, x_2)	$(x_1, x_7) - (x_7, x_2)$	$C(R) + C(x_1, x_7) + C(x_7, x_2) - C(x_1, x_2)$	42,5
(x_1, x_2)	$(x_1, x_8) - (x_8, x_2)$	$C(R) + C(x_1, x_8) + C(x_8, x_2) - C(x_1, x_2)$	31,5
(x_2, x_3)	$(x_2, x_4) - (x_4, x_3)$	$C(R) + C(x_2, x_4) + C(x_4, x_3) - C(x_2, x_3)$	27,3
(x_2, x_3)	$(x_2, x_5) - (x_5, x_3)$	$C(R) + C(x_2, x_5) + C(x_5, x_3) - C(x_2, x_3)$	27,5
(x_2, x_3)	$(x_2, x_6) - (x_6, x_3)$	$C(R) + C(x_2, x_6) + C(x_6, x_3) - C(x_2, x_3)$	34
(x_2, x_3)	$(x_2, x_7) - (x_7, x_3)$	$C(R) + C(x_2, x_7) + C(x_7, x_3) - C(x_2, x_3)$	41,5
(x_2, x_3)	$(x_2, x_8) - (x_8, x_3)$	$C(R) + C(x_2, x_8) + C(x_8, x_3) - C(x_2, x_3)$	31,5
(x_3, x_1)	$(x_3, x_4) - (x_4, x_1)$	$C(R) + C(x_3, x_4) + C(x_4, x_1) - C(x_3, x_1)$	30
(x_3, x_1)	$(x_3, x_5) - (x_5, x_1)$	$C(R) + C(x_3, x_5) + C(x_5, x_1) - C(x_3, x_1)$	29
(x_3, x_1)	$(x_3, x_6) - (x_6, x_1)$	$C(R) + C(x_3, x_6) + C(x_6, x_1) - C(x_3, x_1)$	35,5
(x_3, x_1)	$(x_3, x_7) - (x_7, x_1)$	$C(R) + C(x_3, x_7) + C(x_7, x_1) - C(x_3, x_1)$	43,5
(x_3, x_1)	$(x_3, x_8) - (x_8, x_1)$	$C(R) + C(x_3, x_8) + C(x_8, x_1) - C(x_3, x_1)$	32,5

Berdasarkan Tabel 3.6, bobot minimum yang diperoleh adalah 27,3 ketika sisi (x_2, x_3) digantikan oleh sisi $(x_2, x_4) - (x_4, x_3)$, sehingga terbentuk *subtour* baru $(x_1, x_2) - (x_2, x_4) - (x_4, x_3) - (x_3, x_1)$. Karena titik x_5, x_6, x_7 , dan x_8 belum termasuk dalam *subtour*, dilakukan penyisipan titik dengan memilih rute (sisi) yang menyebabkan penambahan bobot terkecil. Proses ini menggunakan nilai $C(R)$ sebesar 27,3. Hasil penyisipan titik pada *subtour* $(x_1, x_2) - (x_2, x_4) - (x_4, x_3) - (x_3, x_1)$ disajikan pada Tabel 3.7.

Tabel 3.7: Penyisipan pada *Subtour* Ketiga

Sisi awal	Sisi hasil sisipan	Penambahan bobot sisipan	Total bobot <i>subtour</i>
(x_1, x_2)	$(x_1, x_5) - (x_5, x_2)$	$C(R) + C(x_1, x_5) + C(x_5, x_2) - C(x_1, x_2)$	32,8
(x_1, x_2)	$(x_1, x_6) - (x_6, x_2)$	$C(R) + C(x_1, x_6) + C(x_6, x_2) - C(x_1, x_2)$	39,8
(x_1, x_2)	$(x_1, x_7) - (x_7, x_2)$	$C(R) + C(x_1, x_7) + C(x_7, x_2) - C(x_1, x_2)$	47,3
(x_1, x_2)	$(x_1, x_8) - (x_8, x_2)$	$C(R) + C(x_1, x_8) + C(x_8, x_2) - C(x_1, x_2)$	36,3
(x_2, x_4)	$(x_2, x_5) - (x_5, x_4)$	$C(R) + C(x_2, x_5) + C(x_5, x_4) - C(x_2, x_4)$	30
(x_2, x_4)	$(x_2, x_6) - (x_6, x_4)$	$C(R) + C(x_2, x_6) + C(x_6, x_4) - C(x_2, x_4)$	37
(x_2, x_4)	$(x_2, x_7) - (x_7, x_4)$	$C(R) + C(x_2, x_7) + C(x_7, x_4) - C(x_2, x_4)$	44
(x_2, x_4)	$(x_2, x_8) - (x_8, x_4)$	$C(R) + C(x_2, x_8) + C(x_8, x_4) - C(x_2, x_4)$	34,5
(x_4, x_3)	$(x_4, x_5) - (x_5, x_3)$	$C(R) + C(x_4, x_5) + C(x_5, x_3) - C(x_4, x_3)$	28,8
(x_4, x_3)	$(x_4, x_6) - (x_6, x_3)$	$C(R) + C(x_4, x_6) + C(x_6, x_3) - C(x_4, x_3)$	35,3
(x_4, x_3)	$(x_4, x_7) - (x_7, x_3)$	$C(R) + C(x_4, x_7) + C(x_7, x_3) - C(x_4, x_3)$	42,8
(x_4, x_3)	$(x_4, x_8) - (x_8, x_3)$	$C(R) + C(x_4, x_8) + C(x_8, x_3) - C(x_4, x_3)$	33,3
(x_3, x_1)	$(x_3, x_5) - (x_5, x_1)$	$C(R) + C(x_3, x_5) + C(x_5, x_1) - C(x_3, x_1)$	33,8
(x_3, x_1)	$(x_3, x_6) - (x_6, x_1)$	$C(R) + C(x_3, x_6) + C(x_6, x_1) - C(x_3, x_1)$	40,3
(x_3, x_1)	$(x_3, x_7) - (x_7, x_1)$	$C(R) + C(x_3, x_7) + C(x_7, x_1) - C(x_3, x_1)$	48,3
(x_3, x_1)	$(x_3, x_8) - (x_8, x_1)$	$C(R) + C(x_3, x_8) + C(x_8, x_1) - C(x_3, x_1)$	37,3

Berdasarkan Tabel 3.7, bobot minimum yang diperoleh adalah 28,8, apabila sisi (x_4, x_3) diganti dengan $(x_4, x_5) - (x_5, x_3)$ sehingga *subtour* yang baru menjadi $(x_1, x_2) - (x_2, x_4) - (x_4, x_5) - (x_5, x_3) - (x_3, x_1)$. Langkah untuk proses penyisipan titik-titik lain diulangi hingga semua titik telah masuk dalam *subtour*. Hasil penyisipan titik-titik selanjutnya disajikan pada Tabel 3.8, 3.9, dan 3.10 dengan nilai $C(R)$ masing-masing sebesar 28,8; 33,3; dan 40,3.

Tabel 3.8: Penyisipan pada *Subtour* Keempat

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot <i>Subtour</i>
(x_1, x_2)	$(x_1, x_6) - (x_6, x_2)$	$C(R) + C(x_1, x_6) + C(x_6, x_2) - C(x_1, x_2)$	41,3
(x_1, x_2)	$(x_1, x_7) - (x_7, x_2)$	$C(R) + C(x_1, x_7) + C(x_7, x_2) - C(x_1, x_2)$	48,8
(x_1, x_2)	$(x_1, x_8) - (x_8, x_2)$	$C(R) + C(x_1, x_8) + C(x_8, x_2) - C(x_1, x_2)$	37,8
(x_2, x_4)	$(x_2, x_6) - (x_6, x_4)$	$C(R) + C(x_2, x_6) + C(x_6, x_4) - C(x_2, x_4)$	38,5
(x_2, x_4)	$(x_2, x_7) - (x_7, x_4)$	$C(R) + C(x_2, x_7) + C(x_7, x_4) - C(x_2, x_4)$	45,5
(x_2, x_4)	$(x_2, x_8) - (x_8, x_4)$	$C(R) + C(x_2, x_8) + C(x_8, x_4) - C(x_2, x_4)$	36
(x_4, x_5)	$(x_4, x_6) - (x_6, x_5)$	$C(R) + C(x_4, x_6) + C(x_6, x_5) - C(x_4, x_5)$	35,8
(x_4, x_5)	$(x_4, x_7) - (x_7, x_5)$	$C(R) + C(x_4, x_7) + C(x_7, x_5) - C(x_4, x_5)$	42,8
(x_4, x_5)	$(x_4, x_8) - (x_8, x_5)$	$C(R) + C(x_4, x_8) + C(x_8, x_5) - C(x_4, x_5)$	33,8

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot Subtour
(x_5, x_3)	$(x_5, x_6) - (x_6, x_3)$	$C(R) + C(x_5, x_6) + C(x_6, x_3) - C(x_5, x_3)$	35,3
(x_5, x_3)	$(x_5, x_7) - (x_7, x_3)$	$C(R) + C(x_5, x_7) + C(x_7, x_3) - C(x_5, x_3)$	42,8
(x_5, x_3)	$(x_5, x_8) - (x_8, x_3)$	$C(R) + C(x_5, x_8) + C(x_8, x_3) - C(x_5, x_3)$	33,3
(x_3, x_1)	$(x_3, x_6) - (x_6, x_1)$	$C(R) + C(x_3, x_6) + C(x_6, x_1) - C(x_3, x_1)$	41,8
(x_3, x_1)	$(x_3, x_7) - (x_7, x_1)$	$C(R) + C(x_3, x_7) + C(x_7, x_1) - C(x_3, x_1)$	49,8
(x_3, x_1)	$(x_3, x_8) - (x_8, x_1)$	$C(R) + C(x_3, x_8) + C(x_8, x_1) - C(x_3, x_1)$	38,8

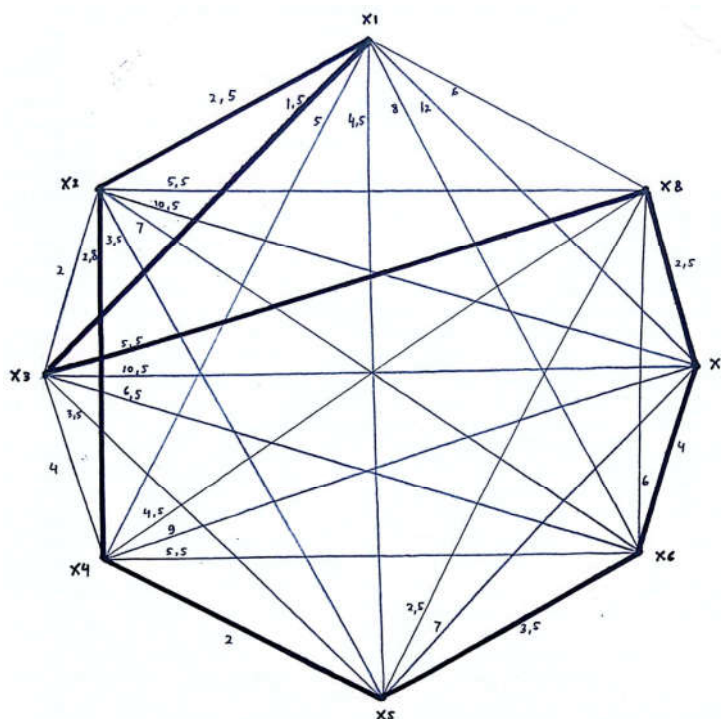
Tabel 3.9: Penyisipan pada Subtour Kelima

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot Subtour
(x_1, x_2)	$(x_1, x_6) - (x_6, x_2)$	$C(R) + C(x_1, x_6) + C(x_6, x_2) - C(x_1, x_2)$	45,8
(x_1, x_2)	$(x_1, x_7) - (x_7, x_2)$	$C(R) + C(x_1, x_7) + C(x_7, x_2) - C(x_1, x_2)$	53,3
(x_2, x_4)	$(x_2, x_6) - (x_6, x_4)$	$C(R) + C(x_2, x_6) + C(x_6, x_4) - C(x_2, x_4)$	43
(x_2, x_4)	$(x_2, x_7) - (x_7, x_4)$	$C(R) + C(x_2, x_7) + C(x_7, x_4) - C(x_2, x_4)$	50
(x_4, x_5)	$(x_4, x_6) - (x_6, x_5)$	$C(R) + C(x_4, x_6) + C(x_6, x_5) - C(x_4, x_5)$	40,3
(x_4, x_5)	$(x_4, x_7) - (x_7, x_5)$	$C(R) + C(x_4, x_7) + C(x_7, x_5) - C(x_4, x_5)$	47,3
(x_5, x_8)	$(x_5, x_6) - (x_6, x_8)$	$C(R) + C(x_5, x_6) + C(x_6, x_8) - C(x_5, x_8)$	40,3
(x_5, x_8)	$(x_5, x_7) - (x_7, x_8)$	$C(R) + C(x_5, x_7) + C(x_7, x_8) - C(x_5, x_8)$	40,3
(x_8, x_3)	$(x_8, x_6) - (x_6, x_3)$	$C(R) + C(x_8, x_6) + C(x_6, x_3) - C(x_8, x_3)$	40,3
(x_8, x_3)	$(x_8, x_7) - (x_7, x_3)$	$C(R) + C(x_8, x_7) + C(x_7, x_3) - C(x_8, x_3)$	40,8
(x_3, x_1)	$(x_3, x_6) - (x_6, x_1)$	$C(R) + C(x_3, x_6) + C(x_6, x_1) - C(x_3, x_1)$	46,3
(x_3, x_1)	$(x_3, x_7) - (x_7, x_1)$	$C(R) + C(x_3, x_7) + C(x_7, x_1) - C(x_3, x_1)$	54,3

Tabel 3.10: Penyisipan pada Subtour Keenam

Sisi Awal	Sisi Hasil Sisipan	Penambahan Bobot Sisipan	Total Bobot Subtour
(x_1, x_2)	$(x_1, x_7) - (x_7, x_2)$	$C(R) + C(x_1, x_7) + C(x_7, x_2) - C(x_1, x_2)$	60,3
(x_2, x_4)	$(x_2, x_7) - (x_7, x_4)$	$C(R) + C(x_2, x_7) + C(x_7, x_4) - C(x_2, x_4)$	57
(x_4, x_5)	$(x_4, x_7) - (x_7, x_5)$	$C(R) + C(x_4, x_7) + C(x_7, x_5) - C(x_4, x_5)$	54,3
(x_5, x_6)	$(x_5, x_7) - (x_7, x_6)$	$C(R) + C(x_5, x_7) + C(x_7, x_6) - C(x_5, x_6)$	47,8
(x_6, x_8)	$(x_6, x_7) - (x_7, x_8)$	$C(R) + C(x_6, x_7) + C(x_7, x_8) - C(x_6, x_8)$	40,8
(x_8, x_3)	$(x_8, x_7) - (x_7, x_3)$	$C(R) + C(x_8, x_7) + C(x_7, x_3) - C(x_8, x_3)$	47,8
(x_3, x_1)	$(x_3, x_7) - (x_7, x_1)$	$C(R) + C(x_3, x_7) + C(x_7, x_1) - C(x_3, x_1)$	61,3

Setelah melakukan tujuh kali penyisipan, rute perjalanan terpendek untuk mengunjungi kedelapan titik ini adalah sebagai berikut: $(x_1, x_2) - (x_2, x_4) - (x_4, x_5) - (x_5, x_6) - (x_6, x_7) - (x_7, x_8) - (x_8, x_3) - (x_3, x_1)$ atau KPRK Palembang \rightarrow KPC Kapt A. Rivai \rightarrow KPC Pakjo \rightarrow KPC Talang Ratu \rightarrow KPC Sukarami \rightarrow KPC Alang Lebar \rightarrow KPC Sekip \rightarrow KPC Cinde \rightarrow KPRK Palembang (Gambar 3.2), dengan total jarak tempuh 24,3 km. Rute ini merupakan rute optimal karena meminimalkan total jarak tempuh, yang dapat menghemat biaya dan waktu.



Gambar 3.2. Rute Pendistribusian Optimal

3.10. Analisis Perbandingan Rute Branch and Bound dan Rute Kombinasi

Setelah dilakukan perhitungan, kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* menghasilkan rute terpendek dengan jumlah jarak keseluruhan yaitu 24,3 km. Hasil yang sama dicapai ketika hanya memanfaatkan algoritma *Branch and Bound*, meskipun dengan rute yang berbeda (Bangun et al., 2015). Namun, dalam kasus ini kombinasi algoritma lebih praktis karena dalam algoritma *Branch and Bound*, setiap kali menghitung nilai $C(R)$, diperlukan langkah pengerjaan menggunakan matriks, yang meningkatkan risiko kesalahan dan memerlukan waktu lebih banyak untuk menemukan nilai terkecil di setiap baris dan kolom matriks. Proses reduksi matriks ini dilakukan secara berulang. Sebaliknya, pada kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic*, pengerjaan hanya memerlukan operasi pada tabel, yang lebih mudah dibandingkan dengan operasi matriks seperti dalam penelitian Bangun et al. (2015). Selain itu, dalam algoritma kombinasi, penentuan nilai $C(R)$ menggunakan matriks hanya dilakukan di awal, kemudian nilai $C(R)$ dapat langsung ditemukan dengan mengidentifikasi penambahan bobot terkecil pada setiap penyisipan titik baru dalam *subtour*. Hal ini membuat waktu dan komputasi lebih efisien dan sederhana.

Menggabungkan algoritma *Branch and Bound* dengan metode lain, seperti *Cheapest Insertion Heuristic*, memungkinkan untuk mengeliminasi cabang-cabang yang tidak relevan

dan mengurangi jumlah sub-masalah yang perlu dievaluasi, sehingga mempercepat pencarian solusi. Selain itu, algoritma *Branch and Bound* kurang cocok digunakan untuk jumlah titik atau lokasi yang besar. Namun, ketika dikombinasikan dengan *Cheapest Insertion Heuristic*, metode ini tetap stabil digunakan dalam menyelesaikan TSP dengan jumlah titik atau lokasi yang besar. Hal ini sejalan dengan penelitian Nur & Rahadjeng (2021) dan Safitri et al. (2024) yang menunjukkan bahwa kombinasi metode *Branch and Bound* dan *Cheapest Insertion Heuristic* dalam kasus tertentu lebih baik dalam menghasilkan rute optimal.

4. KESIMPULAN

Menurut hasil dan pembahasan, maka disimpulkan bahwa rute optimal untuk pendistribusian barang di kantor pos Kota Palembang dengan menggunakan algoritma *Branch and Bound* yang dikombinasikan dengan *Cheapest Insertion Heuristic* adalah: KPRK Palembang → KPC Kapt A. Rivai → KPC Pakjo → KPC Talang Ratu → KPC Sukarami → KPC Alang Lebar → KPC Sekip → KPC Cinde → KPRK Palembang, dengan total jarak tempuh 24,3 km. Penelitian oleh Bangun et al. (2015) yang hanya memanfaatkan algoritma *Branch and Bound* untuk mendapatkan total jarak yang sama, meskipun dengan rute yang berbeda. Pada kasus ini, kombinasi algoritma *Branch and Bound* dan *Cheapest Insertion Heuristic* lebih efisien dan sederhana dalam hal langkah pengerjaan dan waktu komputasi. Selain itu, metode ini masih cocok untuk diterapkan pada masalah TSP dengan jumlah titik atau lokasi yang lebih besar.

5. REFERENSI

- Anassuka, B., Cholissodin, I., & Rahayudi, B. (2022). Optimasi Rute Multiple Travelling Salesman Problem Distribusi Produk PT Indomarco Adi Prima (Stock Point Nganjuk) menggunakan Algoritme Ant Colony Optimization dan Algoritme Genetika. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(3), 1290–1297.
- Aristi, G. (2014). Perbandingan Algoritma Greedy, Algoritma Cheapest Insertion Heuristics dan Dynamic Programming dalam Penyelesaian Travelling Salesman Problem. *Paradigma*, 16(2), 52–58.
- Auliasari, K., Kertaningtyas, M., & Basuki, D. W. L. (2018). Optimalisasi Rute Distribusi Produk Menggunakan Metode Traveling Salesman Problem. *Optimalisasi Rute Distribusi Produk Menggunakan Metode Traveling Salesman Problem*, 16(1), 15–23.
- Azzahrha, F. K., Sari, R. P., & Fauzi, M. D. R. (2021). Optimalisasi Produksi Tahu Menggunakan Metode Branch and Bound dan Cutting Plane. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 6(2), 175–184.
- Bangun, P. B. J., Octarina, S., & Purba, B. V. (2015). Penyelesaian Travelling Salesman Problem (TSP) dengan Metode Branch and Bound (Aplikasi Permasalahan Pengangkutan Barang Kantor Pos Palembang). *SEMIRATA 2015*, 1(1), 399–408.
- Hignasari, L. V. (2019). Komparasi Algoritma Cheapest Insertion Heuristic (CIH) dan Greedy dalam Optimasi Rute Pendistribusian Barang. *Jurnal Ilmiah Vastuwidya*, 2(2), 31–39.
- Lakutu, N. F., Mahmud, S. L., Katili, M. R., & Yahya, N. I. (2023). Algoritma Dijkstra dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos

- Gorontalo. *Euler: Jurnal Ilmiah Matematika, Sains Dan Teknologi*, 11(1), 55–65.
- Lattan, B. W., Tupan, J. M., & Paillin, D. B. (2021). Pemecahan Traveling Salesmen Problem Menggunakan Teknik Branch and Bound dan Cheapest Insertion Heuristic. *I Tabaos*, 1(1), 13–22.
- Mas'ud, S. (2024). Route Determination for Distribution by Using a Combination of Branch and Bound Algorithm and Cheapest Insertion Heuristic. *ARRUS Journal of Mathematics and Applied Science*, 4(1), 20–27.
- Ningrum, E. R., Sanwidi, A., Akbarita, R., & Qomaruddin, M. N. H. (2023). Optimasi Rute Pendistribusian Gas Elpiji Menggunakan Algoritma Floyd Warshall dan Algoritma Greedy. *Jurnal Ilmiah Matematika Dan Terapan*, 20(1), 1–14.
- Nur, M. A. S., & Rahadjeng, B. (2021). Kombinasi Algoritma Branch and Bound dan Cheapest Insertion Heuristic dalam Menyelesaikan Asymmetric Travelling Salesman Problem. *MATHunesa: Jurnal Ilmiah Matematika*, 9(2), 351–358.
- Paillin, D. B., & Sosebeko, F. (2017). Penentuan Rute Optimal Distribusi Produk Nestle Dengan Metode Traveling Salesman Problem (TSP)(Studi Kasus: PT. Paris Jaya Mandiri). *Arika*, 11(1), 35–44.
- 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 2018 Surakarta 7-8 Mei*.
- Rahmadi, D., & Sandariria, H. (2023). Penerapan Minimum Spanning Tree dalam Menentukan Rute Terpendek Distribusi Naskah Soal USBN di SMA Negeri se-Sleman. *Basis: Jurnal Ilmiah Matematika*, 2(1), 66–71.
- Rozi, S., & Multahadah, C. (2021). Rute Terpendek untuk Pengangkutan Sampah dengan Pendekatan Lintasan Hamilton. *E-Jurnal Matematika*, 10(2), 115–121.
- Safitri, E., Basriati, S., Widiarti, W., & Sukmawati, S. (2024). Kombinasi Algoritma Branch and Bound dan Cheapest Insertion Heuristic dalam Mengoptimalkan Rute Distribusi Kurir Paket JNT di Kecamatan Batang Cenaku. *Jurnal Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika Dan Statistika*, 5(1), 561–572.
- Saleh, K., & Helmi, B. P. (2015). Penentuan Rute Terpendek Dengan Menggunakan Algoritma Cheapest Insertion Heuristic (Studi Kasus: Pt. Wicaksana Overseas International Tbk. Cabang Pontianak). *Bimaster: Buletin Ilmiah Matematika, Statistika Dan Terapannya*, 4(03), 295–304.
- Saputra, D. W. (2022). Optimalisasi Rute Distribusi Kurir Menggunakan Metode Traveling Salesman Problem (Studi Kasus: Jne Balige). *G-Tech: Jurnal Teknologi Terapan*, 6(2), 159–165.
- Sari, M., & Asmendri, A. (2020). Penelitian Kepustakaan (Library Research) dalam Penelitian Pendidikan IPA. *Natural Science*, 6(1), 41–53.
- Sihombing, D. E., & Ahyaningsih, F. (2023). Optimalisasi Rute Distribusi Air Minum Dalam Kemasan Menggunakan Algoritma Genetika Pada PT. Mual Natio Maju Bersama. *JURNAL Riset RUMPUN ILMU PENDIDIKAN*, 2(1), 70–83.
- Sulistiyorini, R., & Mahmudy, W. F. (2015). Penerapan Algoritma genetika untuk Permasalahan Optimasi Distribusi Barang Dua Tahap. *DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya*, 5(12), 1–12.

- Usman, M. Z., & Oktiarso, T. (2018). Implementasi Algoritma Greedy Untuk Menyelesaikan Travelling Salesman Problem di Distributor PT. *Z. Journal of Integrated System*, *1*(2), 216–229.
- Yuliantari, R., & Musabbikhah, L. (2022). Review Artikel: Analisis Penggunaan Algoritma Dijkstra untuk Mencari Rute Terpendek di Rumah Sakit. *Edu ElektriKa Journal*, *11*(1), 1–5.