

BAB II

LANDASAN TEORI

2.1 Transportasi dan Distribusi

2.1.1 Pengertian Transportasi

Transportasi merupakan proses pemindahan fisik manusia maupun barang dari satu tempat ke tempat lainnya dalam kurun waktu tertentu. Kegiatan ini dapat dilakukan dengan bantuan tenaga manusia, hewan, atau mesin. Secara umum, transportasi diklasifikasikan ke dalam tiga moda utama, yaitu transportasi darat, laut, dan udara (Said, Maitimu, & Talakua, 2022).

Transportasi adalah layanan yang disediakan untuk memfasilitasi perpindahan orang maupun barang dari satu lokasi ke lokasi lainnya. Dengan kata lain, transportasi mencakup berbagai aktivitas atau upaya dalam mengangkut penumpang atau barang dari suatu tempat ke tempat lain secara efisien dan terencana. Peran transportasi sangat penting dalam mendukung mobilitas, memperlancar arus distribusi, serta mendorong pertumbuhan ekonomi dan konektivitas antarwilayah (Wakari, Rogi, & Makarau, 2019).

2.1.2 Pengertian Distribusi Barang

Distribusi adalah rangkaian kegiatan dalam logistik yang memiliki tujuan untuk mengirimkan produk dari tempat asal ke tempat tujuan yang dikehendaki atau dari pabrik ke konsumen. Sistem ini mencakup seluruh proses mulai dari penyimpanan, pengiriman, penerimaan hingga pemilihan moda transportasi dan rute distribusi yang tepat agar produk sampai pada tempat, waktu, jumlah, dan kondisi yang sesuai (Suryanto, 2016).

Distribusi merupakan salah satu elemen dalam kegiatan pemasaran yang bertujuan untuk menyebarluaskan informasi, membujuk, dan mengingatkan pasar sasaran mengenai keberadaan perusahaan beserta produk yang ditawarkannya. Tujuan utama dari kegiatan ini adalah untuk mendorong minat pasar agar bersedia menerima, membeli, serta tetap loyal terhadap produk yang ditawarkan. Distribusi juga berperan dalam memastikan produk dapat tersedia

di tempat dan waktu yang tepat guna memenuhi kebutuhan konsumen secara efektif (Ardiansyah & Soegoto, 2019).

Adapun Komponen utama sistem distribusi antara lain:

- a. Produsen: Merupakan pihak yang memproduksi suatu barang atau produk.
- b. Distributor: Merupakan pihak perantara antara produsen dengan konsumen, tapi seringkali pihak ini membeli produk dalam jumlah yang besar kepada produsen lalu di salurkan lagi ke pengecer sebelum akhirnya berada di konsumen akhir.
- c. Pengecer: Merupakan pihak yang menyalurkan atau menjual produk kepada konsumen akhir.
- d. Konsumen: Pihak yang menggunakan atau membeli produk atau jasa.
- e. Sarana Transportasi: Merupakan alat yang digunakan untuk memindahkan produk dari tempat A ke tempat B
- f. Pergudangan: Tempat penyimpanan sementara produk.

2.2 Vehicle Routing Problem

2.2.1 Definisi *Vehicle Routing Problem*

Vehicle Routing Problem (VRP) merupakan permasalahan optimasi kombinatorial yang berkaitan dengan penentuan rute terbaik bagi sejumlah kendaraan yang harus mengantarkan layanan atau barang dari satu atau lebih depot ke sejumlah *customer* yang tersebar di berbagai lokasi geografis. Tujuan utamanya adalah meminimalkan total biaya operasional, jarak tempuh, atau jumlah kendaraan yang diperlukan, dengan tetap mempertimbangkan berbagai kendala operasional seperti kapasitas kendaraan dan waktu pelayanan di lokasi *customer* (Taptajani, 2021).

Metode *Vehicle Routing Problem* (VRP) merupakan pendekatan yang digunakan untuk mengoptimalkan penentuan jalur terbaik dalam pengiriman barang dari beberapa titik distribusi. Tujuan utama dari metode ini adalah untuk mempercepat proses distribusi produk sehingga waktu pengantaran dapat diminimalkan. Selain itu, VRP juga berfokus pada efisiensi penggunaan

sumber daya, khususnya dalam mengurangi konsumsi bahan bakar dan biaya operasional kendaraan (Chandra & Setiawan, 2018).

2.2.2 Jenis – Jenis VRP

Vehicle routing problem adalah sebuah masalah optimasi yang memiliki tujuan untuk mencari rute paling optimal untuk kendaraan dalam sebuah pengiriman ke beberapa tempat tujuan, karena kondisi dilapangan sering bervariasi maka vrp sendiri juga memiliki beberapa variasi atau jenisnya diantaranya sebagai berikut:

1. *Capacitated Vehicle Routing Problem (CVRP)*

CVRP adalah sebuah jenis VRP yang paling dasar. Kendaraan memiliki kapasitas yang terbatas lalu total *customer* di tiap rutenya tidak boleh lebih dari kapasitas kendaraan tersebut dan tiap *customer*-nya hanya dikunjungi satu kali dalam satu siklus pengiriman (Rafelda, 2020).

2. *Vehicle Routing Problem with Time Windows (VRPTW)*

VRPTW memiliki batasan waktu untuk setiap *customernya*, kendaraan harus tiba di lokasi tujuan dan melayani *customer* dalam jendela waktu yang sudah ditentukan (Wahyuningsih, Satyananda, & Octoviana, 2019). VRPTW dibagi lagi dalam beberapa jenis yaitu:

- *Hard time windows*

Hard time windows yaitu kendaraan harus tiba di lokasi tujuan dalam jendela waktu yang telah ditentukan. Jika kendaraan tiba di lokasi tujuan di luar waktu yang telah ditentukan maka dianggap tidak valid atau akan mendapatkan penalti.

- *Soft time windows*

Soft time windows hampir sama dengan *hard time windows* bedanya yaitu di *soft time windows* dia masih dianggap valid tetapi akan dikenakan penalti jika datang di luar jendela waktu.

- *Disjoint time windows*

Disjoint time windows yaitu dimana *customer* memiliki beberapa jendela waktu yang terpisah, jenis ini lebih fleksible dalam pendistribusiannya.

- *Multiple time windows*

Multiple time windows mirip dengan *disjoint* bedanya cuman di *multiple customer* dapat memiliki lebih dari satu jendela waktu.

3. *Vehicle Routing Problem with Backhaul*

VRP *Backhaul* merupakan pengembangan dari CVRP yang di mana *customer* dibagi menjadi dua jenis yaitu *Linehaul* dan *Backhaul*. *Linehaul* yaitu *customer* menerima pengiriman barang dari depot sedangkan *Backhaul customer* yang meminta pengembalian barang misalnya kendaraan mengirim barang terlebih dahulu (*linehaul*) setelah mengirim barang sekalian mengambil barang (*backhaul*).

4. *Multi Depot Vehicle Routing Problem*

Pada jenis ini perusahaan mempunyai beberapa depot yang bisa digunakan sebagai titik awal dan akhir dari sebuah siklus rute. Kendaraan dapat berangkat dari satu depot dan kembali ke depot yang sama atau ke depot lain.

5. *Heterogeneous Fleet Vehicle Routing Problem*

Heterogeneous berkebalikan dengan VRP dasar yang mengasumsikan kendaraan memiliki kapasitas dan karakteristik yang sama atau homogen, HVRP mempertimbangkan armada kendaraan yang heterogen yaitu memiliki kapasitas, karakteristik dan biaya operasional yang berbeda-beda.

6. *Vehicle Routing Problem with Pick-up and Delivery*

Jenis VRP ini hampir sama dengan VRP *Backhaul* yang dimana dalam siklusnya dia bisa mengantar barang dan sekaligus mengambil barang tetapi di jenis ini lebih umum. Terkadang jenis ini disebut dengan dial – a – ride problem atau *delivery and collection problem*.

7. *Periodic Vehicle Routing Problem*

Dalam jenis ini *customer* harus dikunjungi secara teratur dan berkala dalam satu periode waktu misalnya tiap minggu atau beberapa bulan sekali.

8. *Stochastic Vehicle Routing Problem*

Berbeda dengan jenis VRP sebelumnya yang mengasumsikan semua informasi (permintaan *customer*, waktu perjalanan) diketahui secara pasti, SVRP mempertimbangkan ketidakpastian. Contohnya adalah permintaan *customer* yang tidak pasti, waktu perjalanan yang bervariasi karena kemacetan, atau kemungkinan kendaraan rusak.

9. *Dynamic Vehicle Routing Problem*

DVRP melibatkan informasi yang berubah secara dinamis selama operasi berlangsung. Misalnya, munculnya permintaan *customer* baru yang mendesak, pembatalan pesanan, atau perubahan kondisi lalu lintas. Solusi harus diperbarui secara real-time.

10. *Green Vehicle Routing Problem*

GVRP berfokus pada keberlanjutan dan lingkungan. Tujuannya adalah untuk meminimalkan emisi karbon, konsumsi bahan bakar, atau dampak lingkungan lainnya, di samping tujuan tradisional seperti meminimalkan jarak atau biaya.

2.3 Vehicle Routing Problem with Time Windows

Vehicle Routing Problem with Time Windows (VRPTW) adalah ekstensi VRP klasik yang menambahkan jendela waktu $[a_i, b_i]$ untuk setiap pelanggan, di mana kendaraan harus tiba dalam interval tersebut agar layanan efisien, dengan tujuan meminimalkan total jarak atau biaya operasional (Ahmad & Garside, 2022).

2.3.1 Formula Matematis *Vehicle Routing Problem with Time Windows*

El-Sherbeny (2020) mengusulkan formula matematis *vehicle routing problem with time windows* sebagai berikut:

Himpunan dan Indeks

- V : himpunan kendaraan (armada)

- **C**: himpunan *customer*
- **N**: himpunan seluruh simpul (depot dan *customer*), yaitu ($N = \{0, 1, 2, \dots, n, n+1\}$)
- **i, j**: indeks simpul (node)
- **h**: indeks simpul *customer* (node antara) yang digunakan pada kendala keseimbangan aliran (*flow conservation*)
- **k**: indeks kendaraan

Parameter

- **c_{ij}**: biaya perjalanan dari simpul i ke j
- **t_{ij}**: waktu tempuh dari simpul i ke j
- **d_i**: permintaan (*demand*) *customer* i
- **q**: kapasitas kendaraan
- **a_i**: waktu awal jendela pelayanan (*earliest time*) di simpul i
- **b_i**: waktu akhir jendela pelayanan (*latest time*) di simpul i
- **S_i**: waktu pelayanan (*service time*) di simpul i
- **K**: konstanta besar (Big M)

Variabel Keputusan

- **x_{ijk}**: 1 jika kendaraan k melewati sisi dari i ke j
0 jika tidak
- **s_{ik}**: waktu mulai pelayanan oleh kendaraan k di simpul i

Notasi Depot

- **0**: depot keberangkatan
- **n+1**: depot tujuan (kembali)

Asumsi Model

- Semua parameter q, a_i, b_i, d_i, c_{ij} adalah bilangan bulat non-negatif
- t_{ij} adalah bilangan bulat positif
- a₀ = 0 sehingga s_{0k} = 0 untuk semua kendaraan k

$$\min = \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} C_{ij} X_{ijk} \quad (1)$$

$$\sum_{k \in V} \sum_{j \in N} X_{ijk} = 1 \quad \forall i \in C \quad (2)$$

Persamaan (2) Menyatakan batasan *customer* dilayani satu kali

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ijk} \leq q \quad \forall k \in V \quad (3)$$

Persamaan (3) Menyatakan batasan kapasitas kendaraan

$$\sum_{j \in N} X_{0jk} = 1, \quad \forall k \in V \quad (4)$$

Persamaan (4) Menyatakan setiap *customer* berangkat dari depot

$$\sum_{i \in N} X_{ihk} - \sum_{j \in N} X_{hjk} = 0, \quad \forall h \in C, \quad \forall k \in V \quad (5)$$

Persamaan (5) Menyatakan bahwa Kontinuitas rute (setiap kendaraan yang tiba di *customer* harus meninggalkan *customer* tersebut)

$$\sum_{i \in N} X_{i,0,k} = 1, \quad \forall k \in V \quad (6)$$

Persamaan (6) Menyatakan bahwa setiap kendaraan kembali ke depot

$$S_{ik} + S_i + t_{ij} - K(1 - X_{ijk}) \leq S_{jk}, \quad \forall i, j \in N, \quad \forall k \in V \quad (7)$$

Persamaan (7) S_{ik} adalah waktu mulai pelayanan di lokasi i , S_i adalah waktu pelayanan di lokasi i , t_{ij} adalah waktu tempuh dari i ke j , dan K adalah konstanta Big M. Total waktu dari ruas kiri kurang dari S_{jk} (waktu pelayanan di lokasi j).

$$a_i \leq S_{ik} \leq b_i, \quad \forall i \in N, \quad \forall k \in V \quad (8)$$

Persamaan (8) Menyatakan bahwa pemenuhan time window. a_i adalah batas awal jendela waktu, S_{ik} adalah waktu mulai pelayanan *customer i*, b_i adalah batas akhir jendela waktu.

$$X_{ijk} \in \{0,1\}, \forall i, j \in N, \quad \forall k \in V \quad (9)$$

Persamaan (9) Menyatakan bahwa variabel X_{ijk} bersifat biner (0,1)

2.4 Metode Penyelesaian *Vehicle Routing Problem*

2.4.1 Klasifikasi Metode Penyelesaian

a. Metode *Exact*

Metode *exact* menggunakan pendekatan matematis yang dapat menjamin ditemukannya solusi optimal, seperti *integer linear programming*, *branch and bound*, *branch and cut*, serta *dynamic generation column*. Metode ini efektif untuk ukuran masalah kecil hingga sedang kurang dari 100 customer, misalnya pada *capacitated vehicle routing problem* dan *vehicle routing problem with time windows* (Pak & Mun, 2024). Namun memiliki keterbatasan dalam menangani masalah berskala besar karena kompleksitas komputasi yang sangat tinggi seiring bertambahnya jumlah *customer* dan kendala yang harus dipenuhi

b. Metode *Heuristic*

Metode *heuristic* adalah pendekatan yang lebih sederhana dan cepat, namun tidak selalu memberikan solusi optimal, melainkan solusi yang “cukup baik” dalam waktu singkat. Contoh metode *heuristic* pada VRP meliputi *Nearest Neighbor*, *Savings*, *Insert*, dan *Sweep*. Heuristik banyak digunakan pada aplikasi nyata yang menuntut kecepatan solusi dan dapat dijalankan untuk mendekati solusi optimal, meskipun performanya menurun pada permasalahan yang sangat kompleks (Liu et al., 2023).

c. Metode *Metaheuristic*

Metode metaheuristik merupakan pengembangan lebih lanjut dari heuristic, dengan merangkai beberapa langkah pencarian solusi dan mekanisme eksplorasi yang lebih kompleks seperti *Simulated Annealing*, *Tabu Search*, *Genetic Algorithm*, dan *Large Neighborhood Search*. Metaheuristik mampu mengeksplorasi ruang solusi secara luas sehingga lebih handal dalam memberikan solusi berkualitas untuk skala besar dan kasus kompleks, serta dapat mengatasi keterbatasan heuristic yang sering terjebak pada local optimum (Liang, Minanda, Gunawan, & Chen, 2024).

2.5 Algoritma Branch and Bound

2.5.1 Pengertian Branch and Bound

Algoritma *Branch and Bound* merupakan salah satu metode yang sering digunakan untuk mencari solusi optimal dalam permasalahan optimisasi, terutama pada masalah diskrit dan kombinatorial. Prosesnya terdiri dari dua tahap utama, yaitu *Branch*, yang berfungsi untuk membentuk seluruh cabang pohon keputusan yang memungkinkan menuju solusi, dan *Bound*, yang digunakan untuk mengevaluasi setiap simpul dengan menentukan apakah simpul tersebut masih layak untuk dikembangkan (*active node*) atau harus dieliminasi (*dead node*) berdasarkan batasan (*constraint*) yang ditentukan (Putri & Widyastiti, 2025).

2.5.2 Konsep Batas Atas dan Batas Bawah

Penghentian pencabangan (*fathoming*) dalam metode ini sangat bergantung pada kedua batas tersebut. Untuk masalah maksimisasi, pencabangan pada suatu submasalah akan dihentikan jika batas atas dari submasalah tersebut tidak lebih besar atau sama dengan batas bawahnya. Sebaliknya, untuk masalah minimisasi, pencabangan akan dihentikan jika batas bawah tidak lebih kecil atau sama dengan batas atasnya. Lebih lanjut, kondisi optimal dalam *Branch and Bound* juga didefinisikan berdasarkan batas ini. Untuk masalah maksimisasi, solusi optimal merupakan solusi submasalah yang

saat ini berfungsi sebagai batas bawah (*lower bound*). Sedangkan untuk masalah minimisasi, solusi optimal adalah solusi submasalah yang saat ini menjadi batas atas (*upper bound*) (Putri & Widyastiti, 2025).

2.5.4 Penyelesaian Algoritma Branch & Bound

Secara umum untuk menyelesaikan algoritma *branch and bound* dibutuhkan langkah-langkah sebagai berikut (Nur & Rahadjeng, 2021):

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 ∞ (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 ∞ (tak hingga) untuk semua elemen pada baris i dan kolom j .
 - Ubah menjadi ∞ untuk elemen $A(j,1)$.
 - Lakukan kembali reduksi baris dan kolom pada matriks A , kecuali elemen ∞ .
 - 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.

2.6 Penelitian Terdahulu Mengenai Vehicle Routing Problem with Time Windows

Beberapa penelitian terdahulu yang telah digunakan untuk menyelesaikan permasalahan *vehicle routing problem with time windows* seperti pada tabel 2.1

Tabel 2. 1 Penelitian Terdahulu

Nama dan tahun	Metode yang Digunakan	Objective Function	Hasil Penelitian
(Herdhiansyah, 2023)	<i>Branch and Bound</i>	Jarak tempuh	Jarak tempuh menjadi efisien sebesar 17,7%
(Yuliza, Suprihatin, Bangun, Puspita, & Octarina, 2023)	<i>Branch and Bound</i>	Jarak tempuh	Pengurangan jarak tempuh tertinggi mencapai 46,14 km.
(Kristina, Sianturi, & Wijaya, 2020)	<i>Ant Colony System</i>	Jarak tempuh dan biaya	Terdapat selisih jarak 0,45% dan biaya 1,19% dengan metode sebelumnya.
(Arrosid et al., 2018)	<i>Branch and Bound</i>	Jarak tempuh	Terjadi pengurangan jarak tempuh kendaraan sebesar 9% dan biaya transportasi sebesar 2%.
(Wisudawati, Valentine, & Patradhiani, 2022)	<i>Branch and Bound + Nearest Neighbour</i>	Jarak tempuh dan biaya	Keduanya menghasilkan jarak tempuh serta biaya transportasi yang sama, namun Nearest Neighbour memiliki waktu tempuh lebih cepat.

(Hidayat, 2024)	<i>Sequential Insertion</i>	Jarak tempuh dan biaya	Menghasilkan penghematan jarak 18,13%, biaya Rp58.758, bahan bakar 7,55 liter, dan waktu tempuh 9,01%.
(Nabhana & Rosyada, 2023)	<i>Nearest Insertion</i>	Jarak tempuh	Terdapat penghematan total jarak tempuh sebesar 14,42 km dibandingkan pemilihan rute awal.
(Garside, Utama, & Fitria, 2021)	<i>Artificial Bee Colony</i>	Biaya	Mampu meminimalkan total biaya distribusi (biaya bahan bakar dan keterlambatan) serta memberikan hasil yang kompetitif dibandingkan metode nearest neighbor.
(Rahma, Wahyuningsih, & Oktoviana, 2022)	<i>Tabu Search</i>	Waktu tempuh	Terdapat sisa waktu untuk setiap kendaraan: kendaraan 1 sebesar 2,41 jam, kendaraan 2 sebesar 3,27 jam, dan kendaraan 3 sebesar 1,52 jam
(IRAWAN, 2023)	<i>Integer Linear Programming</i>	Biaya	Mendapatkan 4 rute terbaik untuk 25 customer, dengan total biaya Rp233.000, waktu distribusi 17 jam 57 menit, dan kapasitas 6.150 kg.