

## BAB III

### METODOLOGI PENELITIAN

Bab ini membahas metodologi yang digunakan untuk menyelesaikan masalah mTSP-TW yang diawali dengan mendeskripsikan masalah, menjelaskan tahapan penelitian, membangun model matematika, dan penyelesaiannya dengan IACO.

#### 3.1 Deskripsi Masalah

Penelitian ini membahas mTSP-TW yang merupakan salah satu varian dari TSP. Misal terdapat  $m$  salesman dan  $n$  lokasi. Setiap lokasi harus dikunjungi tepat satu kali oleh seorang salesman. Setiap lokasi hanya dapat dikunjungi dalam rentang waktu yang telah ditentukan yang disebut *time windows*. Penyelesaian mTSP-TW bertujuan menentukan rute perjalanan bagi setiap salesman agar total jarak yang ditempuh seminimum mungkin. Setiap rute harus berawal dan berakhir di depot. Dalam penelitian ini, mTSP-TW akan diselesaikan dengan menggunakan algoritma IACO.

#### 3.2 Tahap Penelitian

Penelitian ini dilakukan melalui tahapan berikut:

1. Studi Pustaka

Pada tahapan ini, dilakukan studi pustaka dengan cara mempelajari teori-teori yang berkaitan dengan penelitian ini dari buku, jurnal, maupun sumber lainnya. Beberapa konsep yang dipelajari meliputi teori dasar graf, mTSP-TW, dan IACO. Kemudian, dilakukan pengumpulan informasi yang relevan dan sumber kajian ilmiah untuk pemahaman mengenai mTSP-TW dan IACO.

2. Pengumpulan Data

Data-data yang dibutuhkan dalam penelitian ini meliputi jarak antar pelanggan, waktu tempuh, waktu pelayanan, waktu tunggu, batasan waktu, dan jumlah salesman.

3. Representasi Graf

Data-data yang diperoleh pada tahapan pengumpulan data akan direpresentasikan ke dalam bentuk graf yang terdiri dari simpul dan sisi. Simpul dari graf mewakili lokasi pelanggan dan sisi dari graf mewakili jalur yang dilalui oleh salesman.

#### 4. Formulasi Model Optimisasi

Pada tahap ini, model matematika akan diformulasikan untuk menyelesaikan masalah mTSP-TW dengan mendefinisikan terlebih dahulu asumsi-asumsi yang digunakan, menentukan parameter, serta menetapkan variabel keputusan yang akan digunakan dalam penyelesaian masalah.

#### 5. Penyelesaian Model

Pada tahapan ini, akan diuraikan teknik penyelesaian yang akan digunakan untuk menyelesaikan masalah mTSP-TW. Selanjutnya, algoritma IACO akan diimplementasikan untuk mencari solusi terbaik.

#### 6. Validasi

Model matematika yang sudah dibangun akan divalidasi dengan cara menyelesaikan suatu kasus mTSP-TW dengan skala kecil menggunakan metode IACO. Validasi dilakukan dengan cara membandingkan solusi dari hasil perhitungan manual dengan solusi hasil implementasi IACO pada pemrograman *python* 3.12.2. Jika solusi yang dihasilkan sama, maka tahapan akan dilanjutkan dengan implementasi. Jika solusi yang dihasilkan berbeda, maka tahapan akan diulang dari pembangunan model optimisasi.

#### 7. Implementasi

Pada tahapan ini, model optimisasi dan teknik penyelesaiannya akan diimplementasikan pada data sekunder dengan dataset yang tersedia di internet.

#### 8. Penarikan Kesimpulan

Pada tahapan ini akan dilakukan penarikan kesimpulan berdasarkan analisis hasil yang didapatkan.

### 3.3 Model Optimisasi

Model optimisasi pada penelitian ini diformulasikan dengan mengambil asumsi berikut:

1. Jarak dari lokasi  $i$  ke lokasi  $j$  sama dengan jarak dari lokasi  $j$  ke lokasi  $i$  (simetris).
2. Seluruh jalan dianggap normal dengan mengabaikan arus lalu lintas serta kecepatan kendaraan *salesman* konstan.

3. Pembagian lokasi yang dikunjungi oleh setiap salesman dilakukan secara merata atau sebisa mungkin seimbang.
4. Hanya terdapat satu depot.
5. *Salesman* tersedia dalam jumlah yang cukup untuk mengunjungi seluruh lokasi.

Formulasi model optimisasi mTSP-TW merujuk dari (Bektas, 2006). Berikut adalah pendefinisian himpunan, parameter dan variabel keputusan pada model optimisasi mTSP-TW:

1. Himpunan

$V$  : Himpunan lokasi dan depot

$N$  : Himpunan lokasi

$K$  : Himpunan kendaraan

2. Parameter

$c_{ij}$ : Jarak dari titik  $i$  ke titik  $j$ ;

$t_{ij}$  : Waktu tempuh dari titik  $i$  ke titik  $j$

$s_i$  : Waktu pelayanan di titik  $i$

$e_i$  : Waktu awal pelayanan titik  $i$

$l_i$  : Waktu akhir pelayanan titik  $i$

$A_i$  : Waktu kedatangan *salesman* ke titik  $i$

$T_i$  : Waktu *salesman* mulai melayani titik  $i$

$W_i$ : Waktu tunggu untuk melayani  $i$

3. Variabel Keputusan

$$x_{ijk} = \begin{cases} 1, & \text{jika terdapat perjalanan } \textit{salesman} \textit{ } k \textit{ } \text{ dari lokasi } i \textit{ } \text{ ke } j; \\ 0, & \text{yang lainnya;} \end{cases}$$

Fungsi tujuan dari penelitian ini adalah untuk meminimumkan total jarak tempuh bagi seluruh *salesman*. Fungsi ini dituliskan sebagai berikut:

**Meminimumkan:**

$$Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk}$$

Adapun kendala-kendala dalam model mTSP-TW adalah sebagai berikut:

1. Setiap titik harus dikunjungi tepat satu kali. Kendala ini dituliskan sebagai berikut:

$$\sum_{k \in K} \sum_{i \in V} x_{ijk} = 1, \forall j \in N$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \forall i \in N$$

2. Setiap *salesman* memulai dan mengakhiri perjalanannya di depot. Kendala ini dituliskan sebagai berikut:

$$\sum_{i \in N} x_{i1k} = 1, \forall k \in K$$

$$\sum_{j \in N} x_{1jk} = 1, \forall k \in K$$

di mana 1 adalah indeks depot.

3. Setiap *salesman* harus meninggalkan lokasi yang dikunjungi. Kendala ini dituliskan sebagai berikut:

$$\sum_{i \in V} x_{ilk} = \sum_{j \in V} x_{ljk}, \forall l \in N, k \in K$$

4. Setiap *salesman* hanya dapat mengunjungi lokasi sesuai batas waktu yang ditentukan. Kendala ini dituliskan sebagai berikut:

$$e_i \leq T_i \leq l_i, \forall i \in N, k \in K$$

5. Kendala untuk menjelaskan hubungan antara waktu mulai pelayanan di suatu lokasi dengan waktu mulai pelayanan lokasi berikutnya. Kendala ini dituliskan sebagai berikut:

$$\text{Jika } x_{ijk} = 1, \text{ maka } T_i + s_i + t_{ij} + W_j = T_j$$

6. Setiap *salesman* tidak bisa melayani sebelum waktu yang diperbolehkan dan harus menunggu jika datang lebih awal. Kendala ini dituliskan sebagai berikut:

$$W_i = T_i - A_i$$

7. Kendala untuk mencegah adanya subroute yang tidak valid di dalam rute *salesman*. Kendala ini dituliskan sebagai berikut:

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1; \forall S \subset N, k \in K$$

Batasan variabel model menyatakan bahwa semua variabel Keputusan model bernilai biner. Batasan ini dinyatakan sebagai:

$$x_{ijk} \in \{0,1\}, \quad \forall i, j \in V, k \in K$$

Selengkapnya, model mTSP-TW pada penelitian ini dapat dimodelkan sebagai model optimasi sebagai berikut:

**Meminimumkan:**

$$Z = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ijk} \quad (3.1)$$

terhadap:

$$\sum_{k \in K} \sum_{i \in V} x_{ijk} = 1, \forall j \in N \quad (3.2)$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \forall i \in N$$

$$\sum_{i \in N} x_{i1k} = 1, \forall k \in K \quad (3.3)$$

$$\sum_{j \in N} x_{1jk} = 1, \forall k \in K$$

$$\sum_{i \in V} x_{ilk} = \sum_{j \in V} x_{ljk}, \forall l \in N, k \in K \quad (3.4)$$

$$e_i \leq T_i \leq l_i, \forall i \in N, k \in K \quad (3.5)$$

$$x_{ijk} = 1 \rightarrow T_i + s_i + t_{ij} + W_j = T_j \quad (3.6)$$

$$W_i = T_i - A_i \quad (3.7)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1; \forall S \subset N, k \in K \quad (3.8)$$

$$x_{ijk} \in \{0,1\}, \quad \forall i, j \in V, k \in K$$

### 3.4 Teknik Penyelesaian Model menggunakan IACO

Pada bagian ini akan dijelaskan mengenai teknik penyelesaian model mTSP-TW menggunakan algoritma IACO merujuk pada Karjono dkk (2016). Algoritma

IACO dimulai dengan inialisasi parameter seperti *alpha*, *beta*, dan jumlah *pheromone* awal. Semut ditempatkan pada posisi awal, lalu probabilitas pemilihan jalur dihitung berdasarkan *pheromone* dan *visibility*. Setiap rute yang terbentuk dibagi untuk masing-masing *salesman*. Selanjutnya, dilakukan mutasi dengan menukar satu lokasi antar dua rute *salesman*, diikuti oleh *local search* menggunakan metode 2-opt untuk meningkatkan kualitas solusi. Proses ini diterapkan pada setiap semut. Setelah semua semut menyelesaikan rutanya, *pheromone* diperbarui berdasarkan kualitas solusi. Iterasi berlanjut hingga kriteria penghentian terpenuhi. solusi optimal pada IACO diwakili berdasarkan jalur terbaik yang ditemukan semut selama iterasi, yang diperkuat melalui mekanisme pembaruan *pheromone*.

Langkah kerja algoritma IACO adalah sebagai berikut:

#### 1. Inialisasi Parameter

Tahap pertama dalam algoritma IACO adalah inialisasi parameter yang akan digunakan pada tahapan selanjutnya. Parameter yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Parameter IACO

No	Parameter	Notasi
1	Intensitas <i>pheromone</i>	$\tau_{ij}$
2	Tetapan pengendali <i>pheromone</i>	$\alpha$
3	Tetapan pengendali visibilitas	$\beta$
4	<i>Visibility</i>	$\eta_{ij}$
5	Banyak semut	$m$
6	Tetapan penguapan jejak semut	$\rho \in (0 - 1)$
7	Laju <i>pheromone</i> lokal	$\xi$
8	Bilangan pembatas	$q_0$

#### 2. Penempatan Semut

Pada awalnya, Setiap semut ditempatkan di depot. Kemudian setiap semut akan melakukan perjalanannya untuk membuat rute masing-masing.

#### 3. Konstruksi Solusi

Setiap semut dalam proses pembangunan solusi memilih langkah berdasarkan kemungkinan yang dipengaruhi oleh *pheromone* dan *visibility*. Dalam konteks pencarian rute terpendek, setiap semut menentukan tujuan selanjutnya berdasarkan kemungkinan pemilihan tujuan tersebut (Harison dkk., 2023). Model matematika yang digunakan untuk menghitung peluang pemilihan suatu tujuan dapat dinyatakan dengan persamaan sebagai berikut:

$$p_{ij}(m) = \frac{(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\sum_{l \in N_i} (\tau_{il}^{\alpha})(\eta_{il}^{\beta})} \quad (3.8)$$

$p_{ij}$  adalah probabilitas semut mengunjungi kota  $i$  ke kota  $j$  dengan  $i \neq j$ ,  $N_i$  adalah himpunan simpul tetangga dari  $i$  yang belum dikunjungi semut, yang mencakup semua simpul yang mungkin dapat dituju semut saat berada di simpul  $i$ . Rute yang diperoleh dari setiap semut akan dibagi menjadi beberapa rute yang masing-masing akan dikunjungi oleh seorang *salesman*. Pembagian rute dilakukan berdasarkan batasan waktu yang dapat dipenuhi oleh tiap *salesman*.

Dalam proses pemilihan kota, dibangkitkan bilangan  $q \in [0,1]$  dan bilangan pembatas  $q_0 \in [0,1]$ . Proses pemilihan akan dilakukan berdasarkan aturan berikut:

a. Eksploitasi

Jika  $q < q_0$ , maka semut akan memilih kota berdasarkan probabilitas terbesar.

b. Eksplorasi

Jika  $q > q_0$ , maka semut akan memilih kota berdasarkan probabilitas acak.

Berikut adalah langkah kerja semut jika  $q > q_0$ .

- 1) Hitung probabilitas pemilihan untuk setiap kota
- 2) Bangkitkan bilangan acak  $r \in [0,1]$
- 3) Tentukan kota yang akan dikunjungi berdasarkan kumulatif probabilitas, yaitu jika  $r$  berada dalam rentang kumulatif antara Kota 1 dan Kota 2, maka kota tersebut yang dipilih.

Proses dimulai dengan *salesman* pertama, yang mengunjungi lokasi hingga mencapai batas waktu yang ditentukan. Jika kunjungan ke lokasi berikutnya

melebihi batas waktu yang ditentukan, lokasi tersebut diberikan ke *salesman* berikutnya.

#### 4. Mutasi

Tujuan mutasi dalam IACO adalah untuk memperkenalkan variasi pada solusi yang dihasilkan oleh semut, mencegah konvergensi terlalu cepat pada solusi suboptimal, dan meningkatkan eksplorasi solusi ruang yang lebih luas.

Tahapan ini tidak ada dalam algoritma ACO. Sehingga tahapan mutasi pada algoritma IACO merupakan salah satu tahapan yang membedakan dengan algoritma ACO. Mutasi yang digunakan adalah mutasi *reciprocal exchange*. Mutasi *reciprocal exchange* adalah memilih dua posisi gen secara acak kemudian menukar gen yang berada di kedua posisi tersebut. Berikut adalah langkah-langkah proses mutasi *reciprocal exchange*:

- 1) Pilih dua titik secara acak dari solusi awal
- 2) Tukar posisi kedua titik yang dipilih pada langkah pertama.

#### 5. *Local Search 2-opt*

*Local Search 2-opt* adalah *Local search* yang mengeliminasi jalur yang bersilangan pada suatu rute tunggal dengan cara mengambil 2 jalur menghubungkan kembali keempat *vertex* (lokasi) pelanggan yang berdekatan. *Local Search* pada algoritma IACO adalah tahapan untuk menentukan nilai jarak terbaik. Nilai ini akan diperoleh dari jarak rute yang telah melalui proses mutasi. Sehingga, tahapan ini dilakukan setelah tahapan mutasi dilakukan. Setiap rute perjalanan yang terbentuk dari proses mutasi akan masuk pada proses *local search 1*, kemudian dipilih perjalanan dengan nilai jarak yang lebih baik. Jika nilai jarak yang terbentuk tidak lebih baik, maka rute perjalanan baru akan diabaikan. Sedangkan setiap rute yang tidak mengalami proses mutasi, akan masuk proses *local search 2*, yaitu langsung digunakan nilai total jaraknya.

Cara kerjanya adalah:

- 1) Inisialisasi solusi yang diperoleh dari tahapan mutasi

- 2) Pilih dua tepi yang berbeda pada rute, baik secara acak maupun secara sistematis
- 3) Balik urutan titik-titik antara kedua tepi yang dipilih
- 4) Hitung kembali total jarak rute baru
- 5) Perbarui rute jika jarak rute baru lebih pendek dari rata jarak rute sebelumnya dan cek apakah kendala waktunya terpenuhi
- 6) Ulangi langkah 2 sampai tidak ada perbaikan yang dapat dilakukan.
6. Pembaruan *pheromone* lokal

Tujuan dari pembaruan *pheromone* dalam algoritma IACO adalah untuk memperbarui informasi yang digunakan oleh semut untuk memilih rute terbaik. *Pheromone* yang lebih banyak di jalur yang lebih baik akan meningkatkan probabilitas jalur tersebut dipilih oleh semut pada iterasi berikutnya.

Tahapan pembaruan *pheromone* pada IACO sama dengan tahapan pembaruan *pheromone* pada algoritma ACO. Pembaruan *pheromone* dihitung dengan persamaan berikut:

$$\tau_{ij} = (1 - \xi) \times \tau_{ij} + \xi \times \tau_0 \quad (3.9)$$

dimana:

$\tau_{ij}$  = Intensitas *pheromone*;

$\xi \in (0,1)$  = Tingkat laju *pheromone*;

$\tau_0$  = *Pheromone* awal

#### 7. Evaluasi Solusi

Solusi yang dihasilkan dievaluasi berdasarkan fungsi tujuan yang diberikan yaitu memperoleh jarak terpendek dan memenuhi semua fungsi kendalanya (*feasible*). Kemudian simpan solusi terbaik yang ditemukan pada iterasi tersebut, sehingga jika ditemukan solusi yang lebih baik dan *feasible* pada iterasi selanjutnya, maka perbarui solusi terbaik.

#### 8. Pembaruan *Pheromone* Global

Tingkat *pheromone* secara global diperbarui berdasarkan solusi terbaik yang ditemukan. Ketika seekor semut menemukan jalur terpendek, maka semut tersebut akan memperbarui kepadatan jejak *pheromone* yang selanjutnya

disebut sebagai global *pheromone* (Yang & Zhuang, 2010). Pembaruan *pheromone* global dihitung dengan persamaan berikut:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho \sum_{m \in M} \Delta\tau_{ij}(m) \quad 3.10)$$

di mana:

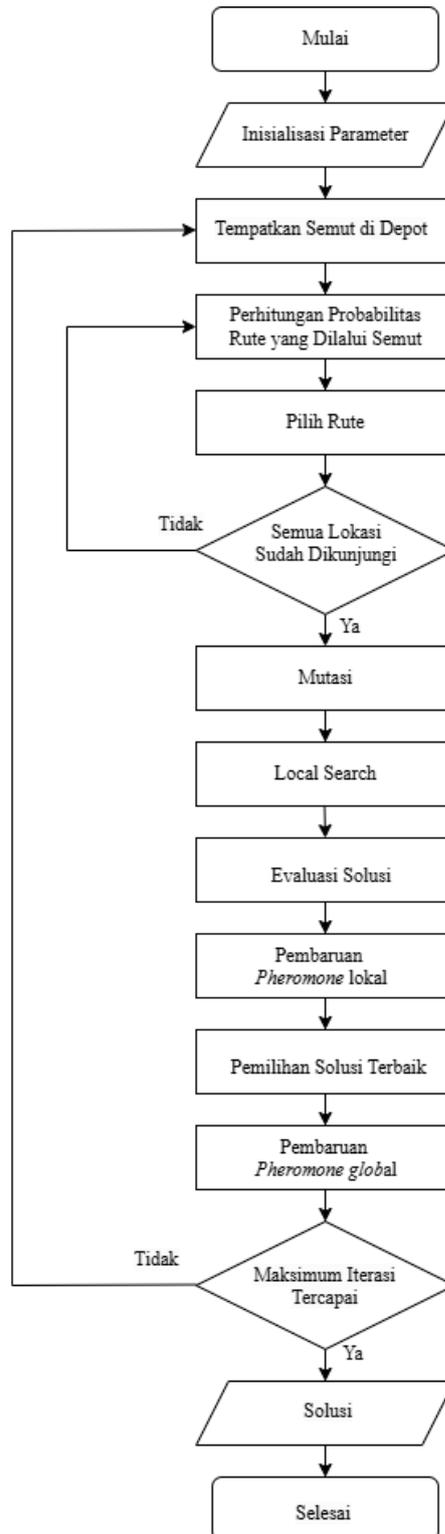
$\Delta\tau_{ij}(m) = \frac{Q}{L_{best}}$  = jumlah *pheromone* yang dihasilkan dari titik  $i$  ke titik  $j$

ketika semut  $m$  berjalan dengan jarak terpendek secara global  $L_{best}$

#### 9. Iterasi

Langkah 2 hingga 8 diulangi sebanyak iterasi yang ditentukan atau telah memenuhi kriteria pemberhentian, yaitu ketika solusi tidak dapat lebih baik lagi setelah beberapa iterasi.

Berikut adalah *flowchart* yang akan menggambarkan penyelesaian mTSP-TW dengan menggunakan Algoritma IACO terdapat pada Gambar 3.1.



Gambar 3.1 Flowchart tahapan penyelesaian mTSP-TW dengan menggunakan Algoritma IACO