

## **BAB III**

### **METODOLOGI PENELITIAN**

Bab ini akan membahas deskripsi masalah, metodologi penelitian, model optimasi *Multi-Criteria Travelling Salesman Problem*, penyelesaian model metode algoritma *Artificial Bee Colony*, serta ilustrasi penyelesaiannya.

#### **3.1 Deskripsi Masalah**

Penelitian ini akan membahas penentuan rute pendistribusian paket kiriman barang dari Gateway J&T Express Sukabumi ke seluruh agen J&T Express yang ada di kota Sukabumi dengan mengaplikasikan penyelesaian *Multi-Criteria Travelling Salesman Problem* (MCTSP) menggunakan algoritma *Artificial Bee Colony* (ABC). MCTSP merupakan perluasan dari *Travelling Salesman Problem* (TSP) berupa penentuan rute dengan mengoptimalkan beberapa kriteri secara bersamaan. Dalam permasalahan yang akan diangkat oleh penulis, lokasi dari Gateway J&T Express merepresentasikan kota awal dan akhir perjalanan.

Penelitian ini mengaplikasikan penyelesaian MCTSP untuk mendapatkan rute yang meminimumkan jarak dan waktu tempuh secara bersamaan dalam pendistribusikan paket kiriman dari gudang utama ke seluruh gudang agen di jaringan J&T Express dan tidak ada gudang agen yang terlewatkan. Tahapan pertama dalam penelitian ini adalah menganalisis masalah. Tahapan ini bertujuan untuk mengetahui sejumlah sumber daya yang dimiliki serta fungsi tujuan yang harus dipenuhi. Dari hasil analisis akan diperoleh variabel input yang akan digunakan dalam penyelesaian Algoritma ABC untuk penentuan rute pendistribusian paket.

Pada penelitian ini, Gateway J&T Express Sukabumi berperan sebagai gudang utama yang akan menjadi titik awal dan akhir dari rute distribusi. Paket-paket yang disimpan di sana selanjutnya akan didistribusikan ke 13 gudang agen J&T Express yang tersebar di Kota dan Kabupaten Sukabumi.

#### **3.2 Model Optimasi untuk *Multi-Criteria Travelling Salesman Problem* (MCTSP)**

Pada bagian ini, model matematika dikonstruksi untuk mengoptimalkan rute pendistribusian paket kiriman berdasarkan permasalahan dan data yang diperoleh.

Model matematika dibangun berdasarkan referensi model yang telah ada dengan memperhatikan beberapa asumsi, yaitu:

1. Jumlah *gateway* dan agen dalam penelitian ini masing-masing adalah satu gudang utama dan 13 gudang agen J&T Express yang tersebar di Kota dan Kabupaten Sukabumi, sedangkan agen dan *gateway* lainnya diabaikan.
2. Hanya terdapat satu depot sebagai titik awal dan akhir rute perjalanan, yaitu Gateway J&T Express.
3. Hanya terdapat satu kendaraan yang beroperasi dalam satu waktu.
4. Setiap agen dikunjungi tepat satu kali.
5. Kapasitas kendaraan yang digunakan dalam pendistribusian bernilai sangat besar atau kendaraan yang digunakan mampu membawa seluruh paket yang akan didistribusikan ke setiap gudang agen.
6. Setiap kendaraan tidak diperkenankan untuk kembali ke depot sebelum semua agen dikunjungi.

Dalam membangun model MCTSP, terlebih dahulu didefinisikan himpunan, parameter, dan variabel yang terlibat. Pada penelitian ini, himpunan yang digunakan, yaitu:

- $X$  : himpunan solusi dalam bentuk rute perjalanan,  
 $V$  : himpunan gudang utama dan gudang agen, dan  
 $A$  : himpunan jalur penghubung antar agen.

Adapun parameter yang digunakan adalah:

- $F_Z$  : fungsi tujuan dari MCTSP  
 $f_d$  : fungsi tujuan untuk jarak perjalanan,  
 $f_t$  : fungsi tujuan untuk waktu perjalanan,  
 $i, j$  : indeks untuk gudang utama dan gudang agen,  
 $d_{ij}$  : jarak tempuh dari gudang  $i$  ke gudang  $j$ ,  
 $t_{ij}$  : waktu tempuh dari gudang  $i$  ke gudang  $j$ ,  
 $n$  : jumlah gudang utama dan agen, dan  
 $\omega_d$  : bobot kriteria jarak

$\omega_t$  : bobot kriteria waktu

Selanjutnya variabel keputusan dari model MCTSP adalah menentukan ada atau tidaknya perjalanan yang dimulai dari depot (titik awal) ke lokasi agen  $i$  hingga ke lokasi agen  $j$  lalu kembali ke depot sebagai akhir dari perjalanan. Variabel keputusan ini dinyatakan dalam  $x_{ij}$  yang didefinisikan sebagai berikut:

$$x_{ij} = \begin{cases} 1, & \text{jika jalur } (i, j) \text{ berada dalam rute} \\ 0, & \text{lainnya.} \end{cases}$$

Setelah mendefinisikan himpunan, parameter dan variabel yang terlibat dalam model, maka dilanjutkan dengan perumusan fungsi tujuan dan kendala model. Fungsi tujuan dari model optimasi ini adalah untuk meminimumkan total jarak dan waktu tempuh perjalanan. Fungsi ini dapat ditulis sebagai:

$$\text{Minimasi } F_Z = \omega_d f_d + \omega_t f_t \quad (3.1)$$

di mana:

$$f_d(x) = \sum_{i=1}^n \sum_{j=i}^n d_{ij} x_{ij}, \text{ dan} \quad (3.2)$$

$$f_t(x) = \sum_{i=1}^n \sum_{j=i}^n t_{ij} x_{ij}. \quad (3.3)$$

Adapun kendala dari model optimasi adalah sebagai berikut.

1. Setiap kota hanya dikunjungi tepat satu kali.

$$\sum_{j=0}^n x_{ij} = 1; \quad i = 0, 1, 2, \dots, n. \quad (3.4)$$

$$\sum_{i=0}^n x_{ij} = 1; \quad j = 0, 1, 2, \dots, n. \quad (3.5)$$

2. Setiap rute tidak mengandung subrute di dalamnya.

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1; \quad S \subseteq N; 2 \leq |S| \leq |N| - 1. \quad (3.7)$$

3. Variabel keputusan bernilai 0 atau 1 (biner).

$$x_{ij} \in \{0,1\}; \quad i \neq j; \quad i, j \in N. \quad (3.6)$$

4. Karena kedua kriteria dioptimasi secara bersamaan, maka bobot dari masing-masing kriterianya perlu dilibatkan dalam perhitungan fungsi tujuan MCTSP.

Kendala ini dinyatakan sebagai berikut.

$$\omega_d + \omega_t = 1. \quad (3.7)$$

Selengkapnya, model MCTSP pada penelitian ini dapat dimodelkan sebagai model optimasi sebagai berikut.

Minimasi  $F_Z = \omega_d f_d + \omega_t f_t$

di mana

$$f_d(x) = \sum_{i=1}^n \sum_{j=i}^n d_{ij} x_{ij}, \text{ dan}$$

$$f_t(x) = \sum_{i=1}^n \sum_{j=i}^n t_{ij} x_{ij}$$

terhadap

$$\sum_{j=0}^n x_{ij} = 1; \quad i = 0, 1, 2, \dots, n,$$

$$\sum_{i=0}^n x_{ij} = 1; \quad j = 0, 1, 2, \dots, n,$$

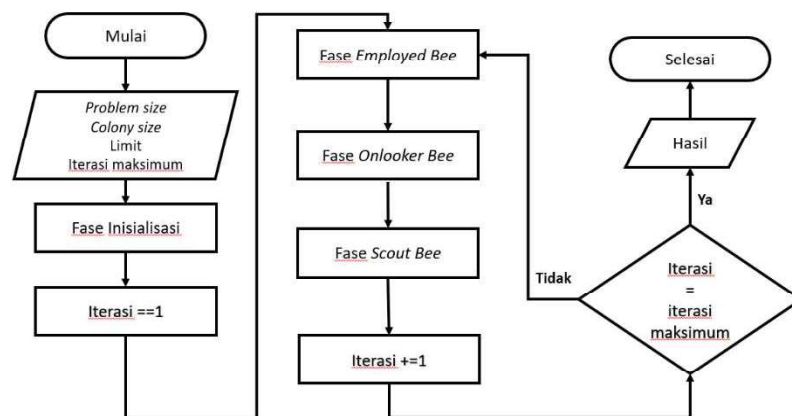
$$\sum_{i,j \in S} x_{ij} \leq |S| - 1; \quad S \subseteq N; 2 \leq |S| \leq |N| - 1,$$

$$x_{ij} \in \{0,1\}; i \neq j; i, j \in N,$$

$$\omega_d + \omega_t = 1.$$

### 3.3 Penyelesaian Model dengan Algoritma Artificial Bee Colony (ABC)

Dalam penelitian ini, penyelesaian dari model MCTSP akan diselesaikan dengan menggunakan Algoritma ABC. Algoritma ABC merupakan algoritma yang terinspirasi dari perilaku koloni lebah dalam mencari sumber makanan. Algoritma ini termasuk ke dalam metode metaheuristik yang dapat digunakan untuk menyelesaikan masalah optimasi yang cukup rumit seperti MCTSP. Diagram alur kerja dari algoritma ABC dapat digambarkan sebagai berikut.



Gambar 3. 1 Tahapan Algoritma ABC

### 3.3.1 Penentuan Parameter

Algoritma ABC dapat bekerja dengan bergantung pada beberapa nilai parameter yang harus ditetapkan di awal proses. Parameter-parameter tersebut adalah sebagai berikut (Amanah & Noviani, 2022).

1. *Colony size*, yaitu jumlah keseluruhan lebah yang digunakan dalam sistem.
2. *Problem Size* mengacu pada dimensi atau ukuran permasalahan, seperti jumlah elemen atau variabel yang menentukan tingkat kompleksitas penyelesaiannya.
3. *Limit*, yaitu jumlah siklus maksimal yang ditentukan untuk menghentikan perbaikan solusi.
4. Iterasi maksimum, yaitu kriteria pemberhentian berdasarkan banyaknya iterasi yang ditetapkan.

Dalam permasalahan multi-kriteria ada dua pendekatan yang biasa digunakan dalam menentukan bobot dari setiap kriteria, yaitu secara subjektif dan objektif. Salah satu pendekatan secara objektif yang paling tepat dalam mendapatkan signifikansi relatif adalah metode *entropy* (Sai & Babu, 2017). Metode ini dipilih karena dapat menormalisasi nilai-nilai dari setiap kriteria, walaupun terdapat perbedaan rentang dan satuan dari setiap kriterianya (Prawiro et al., 2021).

Sebelum masuk ke metode *entropy*, nilai dari setiap kriteria harus dinormalisasi terlebih dahulu agar berada dalam rentang 0 sampai 1. Misalkan terdapat  $m$  kota dan  $n$  kriteria, dan nilai kriteria untuk data ke- $i$  pada kriteria ke- $j$  adalah  $x_{ij}$ . Normalisasi data tersebut diperoleh dengan menggunakan persamaan persamaan berikut.

$$p_{ij} = \frac{x_{ij}}{\sum_{i \in m} x_{ij}} \quad (2.16)$$

di mana:

$p_{ij}$  : normalisasi data ke- $i$  pada kriteria ke- $j$

$x_{ij}$  : data jalur kota  $i$  ke kota  $j$  pada kriteria ke- $n$

Setelah data pada setiap kriteria dinormalisasi, Entropy untuk kriteria ke- $n$  ( $E_n$ ), dihitung dengan menggunakan persamaan berikut.

$$E_n = -k \sum_{i \in m} p_{ij} \ln p_{ij} \quad (2.17)$$

di mana:

- $E_n$  : nilai *entropy* untuk kriteria ke- $n$   
 $k$  : konstanta yang bernilai  $\frac{1}{\ln n}$   
 $p_{ij}$  : hasil normalisasi data ke- $i$  pada kriteria ke- $j$

Terakhir, untuk menentukan nilai bobot kriteria digunakan persamaan berikut.

$$W_j = \frac{d_j}{\sum_{j \in n} d_j} \quad (2.18)$$

di mana:

- $W_j$  : bobot untuk kriteria ke- $j$   
 $d_j$  : nilai hasil pengurangan  $1 - E_j$

### 3.3.2 Fase Inisialisasi

Fase inisialisasi dilakukan untuk memperoleh solusi awal yang dibangkitkan secara acak sesuai dengan parameter yang telah ditentukan. Setelah solusi awal diperoleh, selanjutnya adalah menghitung nilai *fitness*-nya. Pada fase ini, setiap individu memiliki nilai *trial* 0.

### 3.3.3 Fase *Employed Bee*

Fase *employed bee* bertujuan untuk memperbarui solusi awal yang dihasilkan pada fase inisialisasi. Dalam fase ini, *employed bee* akan mencari solusi baru dengan mengeksplorasi solusi tetangga dari solusi awal menggunakan mekanisme seperti *neighborhood operator*. Proses eksplorasi ini dilakukan dengan melakukan perubahan kecil pada solusi awal, misalnya melalui *swap operator* atau *swap sequence*. Karena perubahan yang dilakukan bersifat lokal dan terbatas, solusi yang dihasilkan cenderung lebih stabil, yaitu hanya mengalami sedikit perbaikan dibandingkan solusi sebelumnya. Proses ini memperbaiki solusi sebelumnya tanpa mengubah keseluruhan strukturnya secara drastis, sehingga solusi baru tetap stabil dan masih dekat dengan solusi awal. Stabilitas ini sangat penting dalam fase *employed bee* karena fokusnya adalah pada eksplorasi lokal untuk mengidentifikasi perbaikan kecil sebelum berlanjut ke fase eksplorasi yang lebih luas. Penjelasan mengenai tahapan dari *swap operator* dan *swap sequence* adalah sebagai berikut (Amri *et al.*, 2012).

- *Swap Operator* (SO)

*Swap operator* digunakan untuk menukar posisi urutan dari 2 kota berbeda dengan menggunakan metode *random*. Solusi baru yang diperoleh dari proses SO dinyatakan sebagai berikut.

$$Bee_i = Bee_i + SO, \quad (3.9)$$

Di mana:

$Bee_i$  : individu lebah ke- $i$ ,

$SO$  : *swap operator*.

Sebagai contoh, misalkan  $Bee_1 = (7,8,2,5,3)$  dan nilai dari *swap operator* adalah  $SO(1,4)$ . Maka hasil proses *swap operator* yang bekerja adalah sebagai berikut.

$$\begin{aligned} Bee_1 &= Bee_1 + SO \\ &= (7,8,2,5,3) + SO(1,4) \\ &= (5,8,2,7,3) \end{aligned}$$



**Gambar 3. 2** Ilustrasi *Swap Operator*

Pada Gambar 3.3 terlihat bahwa operator  $SO(1,4)$  bekerja dengan cara menukar posisi elemen pertama dan keempat dari  $Bee_1$

- *Swap Sequence (SS)*

*Swap sequence* adalah kumpulan dari *swap operator*, di mana setiap solusi baru untuk setiap individu diperoleh dengan berdasarkan solusi sebelumnya dan proses *swap sequence*. Proses untuk memperoleh solusi baru dengan menggunakan metode SS dijabarkan sebagai berikut.

$$\begin{aligned} Bee_i &= Bee_i + SS \\ &= Bee_i + (SO_1 + SO_2 + \dots + SO_n) \\ &= ((\dots ((Bee_i + SO_1) + SO_2) + \dots) + SO_n) \end{aligned} \quad (3.10)$$

di mana:

$SS$  : *Swap sequence*,  
 $SO_n$  : *Swap operator ke-n*,  
 $n$  : banyaknya *swap sequence*.

### 3.3.4 Fase *Onlooker Bee*

Fase *onlooker bee* bertujuan untuk menyeleksi dan mengeksploitasi solusi yang dihasilkan pada fase *employed bee*. Dalam fase ini, proses seleksi solusi dilakukan menggunakan metode *roulette wheel selection*, di mana solusi dengan nilai *fitness* yang lebih baik memiliki peluang lebih tinggi untuk dipilih. Setelah proses seleksi, dilakukan eksploitasi solusi terpilih dengan menerapkan *neighborhood operator* untuk menghasilkan solusi baru yang lebih baik. Proses eksploitasi ini bertujuan untuk memperbesar peluang mendapatkan solusi baru dengan peningkatan yang signifikan dibandingkan solusi sebelumnya. *Neighborhood operator* seperti *insert operator* dan *insert sequence* dapat memberikan perubahan yang lebih besar dibandingkan operator sederhana seperti *swap operator*. Dengan memanfaatkan *insert operator* dan *insert sequence*, algoritma dapat mengeksplorasi ruang pencarian yang lebih luas sekaligus memungkinkan perubahan signifikan pada solusi. Hal ini mendukung tujuan fase *onlooker bee*, yaitu mengarahkan algoritma menuju solusi yang lebih optimal secara global. Penjelasan mengenai tahapan dari *insert operator* dan *insert sequence* adalah sebagai berikut (Amri *et al.*, 2012).

- *Insert Operator (IO)*

*Insert Operator* digunakan memindahkan sebuah kota dari suatu posisi ke posisi lain. posisi awal dan posisi tujuan diperoleh dengan menggunakan teknik *random* di mana posisi awal  $\neq$  posisi tujuan. Setelah itu, kota-kota sisa urutan yang berada di antara posisi awal dan tujuan akan digeser. Solusi baru yang dihasilkan dalam proses IO dinyatakan sebagai berikut:

$$Bee_i = Bee_i + IO \quad (3.11)$$

di mana:

$Bee_i$  : individu lebah ke- $i$ ,  
 $IO$  : *insert operator*.



Sebagai contoh, misalkan  $Bee_1 = (9,7,4,2,5)$  dan nilai dari *insert operator* adalah IO (2,5). Maka hasil dari proses *swap operator* yang bekerja adalah sebagai berikut.

$$\begin{aligned} Bee_1 &= Bee_1 + IO \\ &= (9,7,4,2,5) + IO(2,5) \\ &= (9,4,2,5,7) \end{aligned}$$



**Gambar 3.3** Ilustrasi *Insert Operator*

Pada Gambar 3.4, terlihat bahwa operator IO (2,5) bekerja dengan cara memindahkan elemen kedua pada  $Bee_1$  ke posisi elemen kelima, lalu memindahkan menggeser elemen ketiga, keempat, dan kelima ke sebelah kiri untuk mengisi posisi yang kosong.

- *Insert Sequence* (IS)

*Insert sequence* adalah sekumpulan dari *insert operator*, di mana setiap solusi baru untuk setiap individu diperoleh dengan berdasarkan solusi sebelumnya dan proses *insert sequence*. Proses untuk memperoleh solusi baru dengan menggunakan metode IS dijabarkan sebagai berikut:

$$\begin{aligned} Bee_i &= Bee_i + IS \\ &= Bee_i + (IO_1 + IO_2 + \dots + IO_n) \\ &= ((\dots (Bee_i + IO_1) + IO_2) + \dots) + IO_n, \end{aligned}$$

di mana:

- $IS$  : *Insert sequence*,
- $IO_n$  : *Insert operator* ke-n,
- $n$  : banyaknya *swap sequence*.

### 3.3.5 Fase *Scout Bee*

Setelah melalui dua fase sebelumnya, maka akan dilakukan perhitungan kualitas pada setiap individu. Apabila nilai *trial* pada individu melewati nilai limit yang telah

ditentukan, maka solusi dari individu tersebut akan diganti dengan solusi baru yang dibangkitkan secara *random* serta mereset nilai trial pada individu tersebut.

### 3.3.6 Kriteria Pemberhentian

Pemeriksaan kriteria pemberhentian ini akan dilakukan sebanyak iterasi maksimum yang telah ditentukan sebelumnya. Iterasi akan terus dilakukan hingga kriteria pemberhentian dipenuhi. Selama kriteria pemberhentian belum terpenuhi, maka akan mengulang kembali ke langkah 3, yakni fase *employed bee*.

### 3.4 Ilustrasi Penyelesaian *Multi-Criteria Travelling Salesman Problem* Menggunakan Algoritma *Artificial Bee Colony*

Sebagai ilustrasi, akan ditentukan rute optimal untuk pendistribusian paket kiriman dari depot (sebagai titik awal keberangkatan dan titik akhir kembali) ke lima gudang agen. Rute optimal ini bertujuan untuk meminimalkan dua kriteria utama, yaitu jarak tempuh dan waktu pengiriman. Dalam konteks ini, pemilihan rute harus mempertimbangkan efisiensi perjalanan, seperti pengurangan jarak total, serta pengurangan waktu distribusi agar layanan pengiriman menjadi lebih optimal. Berikut adalah data jarak dan waktu tempuh antar gudang.

**Tabel 3. 1** Jarak Tempuh Antar Gudang

JARAK (km)						
Jarak	depot	hub 1	hub 2	hub 3	hub 4	hub 5
depot	0	5	1	5	6	2
hub 1	5	0	3	6	1	6
hub 2	1	3	0	3	6	7
hub 3	5	6	3	0	5	3
hub 4	6	1	6	5	0	10
hub 5	2	6	7	3	10	0

**Tabel 3. 2 Waktu Tempuh Antar Gudang**

WAKTU (menit)						
Waktu	depot	hub 1	hub 2	hub 3	hub 4	hub 5
depot	0	15	24	15	13	16
hub 1	15	0	25	10	19	15
hub 2	24	25	0	11	9	13
hub 3	15	10	11	0	24	25
hub 4	13	19	9	24	0	13
hub 5	16	15	13	25	13	0

Tabel 3.1 dan Tabel 3.2 menunjukkan jarak dan waktu tempuh yang dibutuhkan untuk melakukan perpindahan antar gudang agen. Untuk perpindahan dari dan menuju titik yang sama, nilai jarak dan waktunya bernilai 0 karena tidak ada perpindahan yang terjadi. Urutan langkah-langkah yang digunakan untuk menyelesaikan masalah MCTSP dengan menggunakan ABC adalah sebagai berikut.

**3.4.1 Penentuan Parameter**

Pada contoh ini, parameter yang akan digunakan adalah sebagai berikut.

1. *Max iteration* : 1
2. *Colony size* : 6
3. *Limit* : 4
4. *Size problem* : 6

Sebelum masuk ke fase inisialisasi, terlebih dahulu dihitung bobot dari kedua kriteria tersebut dengan menggunakan metode *entropy*. Pertama, data dari kedua kriteria tersebut perlu dinormalisasi terlebih dahulu menggunakan persamaan (2.16). berikut.

**Tabel 3. 3 Tabel Hasil Normalisasi Data Jarak Tempuh**

JARAK	0	1	2	3	4	5
0	0	0,0362318	0,0072463	0,0362318	0,0434782	0,0144927
1	0,0362318	0	0,0217391	0,0434782	0,0072463	0,0434782
2	0,0072463	0,0217391	0	0,0217391	0,0434782	0,0507246
3	0,0362318	0,0434782	0,0217391	0	0,0362318	0,0217391
4	0,0434782	0,0072463	0,0434782	0,0362318	0	0,0724637
5	0,0144927	0,0434782	0,0507246	0,0217391	0,0724637	0

**Tabel 3. 4** Tabel Hasil Normalisasi Data Waktu Tempuh

WAKTU	0	1	2	3	4	5
0	0	0,0303643	0,0485829	0,0303643	0,0263157	0,0323886
1	0,0303643	0	0,0506072	0,0202429	0,0384615	0,0303643
2	0,0485829	0,0506072	0	0,0222672	0,0182186	0,0263157
3	0,0303643	0,0202429	0,0222672	0	0,0485829	0,0506072
4	0,0263157	0,0384615	0,0182186	0,0485829	0	0,0263157
5	0,0323886	0,0303643	0,0263157	0,0506072	0,0263157	0

Setelah data dinormalisasi, selanjutnya adalah menghitung nilai *entropy* dari setiap kriteria dengan menggunakan persamaan (2.17) dan diperoleh nilai *entropy*-nya dari setiap kriteria adalah  $E_{jarak} = 0,90962197$  dan  $E_{waktu} = 0,934504425$ . Terakhir ditentukan bobot dari setiap kriteria menggunakan persamaan (2.18) dan diperoleh bobotnya adalah  $\omega_{jarak} = 0,579816127$  dan  $\omega_{waktu} = 0,420183873$ .

### 3.4.2 Fase Inisialisasi

Pada fase ini, setiap individu merupakan hasil pembangkitan yang diurutkan dari yang terkecil hingga yang terbesar. Setelah diurutkan, setiap individu akan mendapatkan solusi awal berupa sebuah rute yang dibangkitkan secara acak. Setelah itu, nilai jarak dan waktu dari setiap jalan yang menghubungkan setiap titik pada rute dihitung dan dijumlahkan secara terpisah. Selanjutnya, akan dihitung nilai *fitness* menggunakan total jarak dan waktu tempuh dari rute tersebut. Dari fase ini diperoleh solusi awal seperti pada Tabel 3.5.

**Tabel 3. 5** Hasil Inisialisasi

LEBAH	URUTAN KOTA						fit-MCTSP	trial
b1	0	2	3	1	5	4	0,828089089563	0
b2	0	3	2	4	1	5	0,855479658734	0
b3	0	1	3	4	2	5	0,830393364302	0
b4	0	2	4	3	1	5	0,838506193711	0
b5	0	4	3	5	1	2	0,827807722213	0
b6	0	5	4	2	1	3	0,826924198831	0

### 3.4.3 Fase *Employed Bee*

Pada tahap ini, dilakukan perbaikan solusi untuk memperbaiki solusi dengan menggunakan metode *swap operator* dan *swap sequence*. Solusi yang diperoleh dari hasil swap operator dihitung kembali nilai *fitness* tiap individunya. Hasil perhitungan

*swap operator* dan perbandingannya dengan solusi awal dapat dilihat pada Tabel 3.6 dan Tabel 3.7.

**Tabel 3. 6 Hasil Swap Operator**

Lebah	SO		Urutan Kota						fit-MCTSP
			1	2	3	4	5	6	
b1	4	5	0	2	3	5	1	4	0,851032823659375
b2	5	4	0	3	2	1	4	5	0,843850184154419
b3	4	2	0	4	3	1	2	5	0,82798355439567
b4	4	5	0	2	4	1	3	5	0,856575844542309
b5	2	4	0	1	3	5	4	2	0,825148059946163
b6	5	4	0	5	4	1	2	3	0,843850184154419

**Tabel 3. 7 Perbandingan Hasil Inisialisasi dan Hasil Swap Operator**

Lebah	Hasil Inisialisasi	Swap Operator	trial
	Nilai Fitness	Nilai Fitness	
b1	0,828089089563913	0,851032823659375	0
b2	0,855479658734375	0,843850184154419	1
b3	0,830393364302569	0,82798355439567	1
b4	0,838506193711842	0,856575844542309	0
b5	0,827807722213001	0,825148059946163	1
b6	0,82692419883166	0,843850184154419	0

Dari hasil perbandingan nilai *fitness* tersebut, dihitung nilai *trial* dengan melihat hasil dari *fitness* keduanya. Jika nilai *fitness* dari solusi baru lebih tinggi dari nilai sebelumnya, maka nilai *trial* tidak akan bertambah atau diberi nilai 0. Jika nilai *fitness* solusi terbaru lebih kecil dibanding nilai sebelumnya, maka nilai *trial* akan bertambah 1. Selanjutnya akan dipilih individu dengan nilai *fitness* yang lebih tinggi untuk masuk ke tahap selanjutnya, seperti yang ditunjukkan pada tabel 3.8.

**Tabel 3. 8 Hasil Evaluasi Swap Operator**

LEBAH	URUTAN KOTA						fit-MCTSP	trial
b1	0	2	3	5	1	4	0,85103282	0
b2	0	3	2	4	1	5	0,85547965	1
b3	0	1	3	4	2	5	0,83039336	1
b4	0	2	4	1	3	5	0,85657584	0
b5	0	4	3	5	1	2	0,82780772	1
b6	0	5	4	1	2	3	0,84385018	0

Hasil dari evaluasi *swap operator*, selanjutnya akan diperluas kembali dengan menggunakan teknik *swap sequence*. *Swap sequence* merupakan gabungan dari *swap operator* yang telah digunakan sebelumnya. Solusi baru untuk setiap *employed bee*

dibentuk berdasarkan individu sebelumnya dan hasil *swap sequence*. Berdasarkan teknik tersebut, himpunan operator yang digunakan serta hasil evaluasinya ditunjukkan dalam tabel 3.9 dan tabel 3.10 sebagai berikut.

**Tabel 3. 9** Hasil Evaluasi *Swap Operator*

Swap Sequence			Lebah	Urutan Kota						fit-MCTSP
operator	posisi 1	posisi 2		1	2	3	4	5	6	
1	3	5	b1	0	3	2	5	4	1	0,839776327522042
2	4	3	b2	0	2	3	1	5	4	0,828089089563913
3	3	2	b3	0	1	3	4	5	2	0,814955982066522
4	5	6	b4	0	4	2	1	5	3	0,823011751527993
5	2	6	b5	0	3	1	5	2	4	0,823011751527993
6	6	4	b6	0	4	5	1	3	2	0,828089089563913

**Tabel 3. 10** Perbandingan Hasil *Swap Operator* dan *Swap Sequence*

Lebah	Swap Operator	Swap Sequence	trial
	Nilai Fitness	Nilai Fitness	
b1	0,851032823659375	0,839776327522042	1
b2	0,843850184154419	0,828089089563913	2
b3	0,82798355439567	0,814955982066522	2
b4	0,856575844542309	0,823011751527993	1
b5	0,825148059946163	0,823011751527993	2
b6	0,843850184154419	0,828089089563913	1

Hasil dari perbandingan *swap operator* dan *swap sequence*, akan dihitung kembali nilai trialnya. Individu dengan fitness yang lebih baik akan menggantikan individu sebelumnya. Hasil evaluasi dari *swap sequence* ditunjukkan pada tabel 3.11.

**Tabel 3. 11** Hasil Evaluasi *Swap Sequence*

LEBAH	URUTAN KOTA						fit-MCTSP	trial
b1	0	2	3	5	1	4	0,85103282	1
b2	0	3	2	4	1	5	0,85547965	2
b3	0	1	3	4	2	5	0,83039336	2
b4	0	2	4	1	3	5	0,85657584	1
b5	0	4	3	5	1	2	0,82780772	2
b6	0	5	4	1	2	3	0,84385018	1

### 3.4.4 Fase *Onlooker Bee*

Pada tahap ini, solusi yang telah diperbaharui akan dihitung nilai probabilitas masing-masing individunya dan diseleksi menggunakan teknik *roulette wheel selection*. Selanjutnya solusi akan diperui kembali dengan menggunakan metode *insert operator* dan *insert sequence*. Nilai probabilitas dan probabilitas kumulatif dari masing

masing individu serta hasil roulette wheel selection ditampilkan pada tabel 3.12 dan tabel 3.13 sebagai berikut.

**Tabel 3. 12** Nilai Probabilitas dan Probabilitas Kumulatif Setiap Individu

LEBAH	URUTAN KOTA						fit-MCTSP	Prob	Prom kum	range
	1	2	3	4	5	6				
b1	0	2	3	5	1	4	0,85103282	0,16801764	0,16801764	1 -- 16
b2	0	3	2	4	1	5	0,85547965	0,16889557	0,33691321	17 -- 33
b3	0	1	3	4	2	5	0,83039336	0,16394283	0,50085605	34 -- 50
b4	0	2	4	1	3	5	0,85657584	0,16911199	0,66996804	51 -- 66
b5	0	4	3	5	1	2	0,82780772	0,16343236	0,83340040	67 -- 83
b6	0	5	4	1	2	3	0,84385018	0,16659959	1	83 -- 100

**Tabel 3. 13** Hasil *Roulette Wheel Selection*

LEBAH	URUTAN KOTA						fit-MCTSP	trial
	1	2	3	4	5	6		
b1	0	3	2	4	1	5	0,85547965	1
b2	0	1	3	4	2	5	0,83039336	2
b3	0	2	4	1	3	5	0,85657584	2
b4	0	5	4	1	2	3	0,84385018	1
b5	0	4	3	5	1	2	0,82780772	2
b6	0	2	3	5	1	4	0,85103282	1

Setelah diseleksi, tahap selanjutnya adalah memperbarui solusi dengan menggunakan metode *insert operator*. Hasil dari insert operator ditunjukkan pada tabel 3.14.

**Tabel 3. 14** Hasil *Insert Operator*

Lebah	IO		Urutan Kota						fit-MCTSP
			1	2	3	4	5	6	
b1	2	6	0	2	4	1	5	3	0,822366616
b2	2	3	0	3	1	4	2	5	0,829186708
b3	5	2	0	3	2	4	1	5	0,855479658
b4	3	5	0	5	1	2	4	3	0,832014556
b5	6	2	0	4	3	1	5	2	0,823414319
b6	3	6	0	2	5	1	4	3	0,834305045

Setelah dihitung nilai *fitness* dari solusi baru, hasilnya dibandingkan dengan nilai *fitness* dari solusi solusi yang didapat sebelumnya. Perbandingan hasil roulette wheel selection dan insert operator ditampilkan pada tabel 3.15.

**Tabel 3. 15** Perbandingan Hasil *Roulette Wheel Selection* dan Hasil *Insert Operator*

Lebah	Roulette Wheel Selection	Insert Operator	trial
	Nilai Fitness	Nilai Fitness	
b1	0,855479658734375	0,822366616790945	2
b2	0,830393364302569	0,829186708485452	3
b3	0,856575844542309	0,855479658734375	3
b4	0,843850184154419	0,832014556668462	2
b5	0,827807722213001	0,823414319887975	3
b6	0,851032823659375	0,834305045627061	2

Dari hasil perbandingan nilai *fitness* tersebut, dihitung kembali nilai *trial* dari masing-masing individu. Selanjutnya individu yang dipilih merupakan individu yang memiliki nilai *fitness* yang lebih tinggi. Hasil evaluasi dari insert operator dapat dilihat pada Tabel 3.16.

**Tabel 3. 16** Hasil Evaluasi *Insert Operator*

LEBAH	URUTAN KOTA						fit-MCTSP	trial
	1	2	3	4	5	6		
b1	0	3	2	4	1	5	0,85547965	2
b3	0	1	3	4	2	5	0,83039336	3
b4	0	2	4	1	3	5	0,85657584	3
b6	0	5	4	1	2	3	0,84385018	2
b5	0	4	3	5	1	2	0,82780772	3
b1	0	2	3	5	1	4	0,85103282	2

Setelah dievaluasi, solusi baru diperbarui kembali dengan menggunakan metode *insert sequence*. *Insert sequence* merupakan kumpulan dari *insert operator* dan solusi baru untuk setiap *onlooker bee* dihasilkan berdasarkan solusi sebelumnya dan hasil *insert sequence*. Berdasarkan metode tersebut diperoleh solusi baru yang ditunjukkan pada Tabel 3.17.

**Tabel 3. 17** Hasil *Insert Sequence*

LEBAH	URUTAN KOTA						fit-MCTSP	trial
	1	2	3	4	5	6		
b1	0	3	2	4	1	5	0,85547965	3
b2	0	1	3	4	2	5	0,83039336	4
b3	0	2	4	1	3	5	0,85657584	4
b4	0	5	4	1	2	3	0,84385018	3
b5	0	4	3	5	1	2	0,82780772	4
b6	0	2	3	5	1	4	0,85103282	3



Hasil tersebut selanjutnya dibandingkan dengan solusi sebelumnya. Setelah dibandingkan, solusi sebelumnya akan dievaluasi dengan mengambil individu yang memiliki nilai *fitness* yang lebih besar. Perbandingan solusi serta hasil evaluasinya ditampilkan pada tabel 3.18 dan tabel 3.19.

**Tabel 3. 18** Perbandingan Hasil *Insert Operator* dan *Insert Sequence*

Lebah	Insert Operator	Insert Sequence	trial
	Nilai Fitness	Nilai Fitness	
b1	0,855479658734375	0,83021650723603	3
b2	0,830393364302569	0,834305045627061	4
b3	0,856575844542309	0,822768554121149	4
b4	0,843850184154419	0,834305045627061	3
b5	0,827807722213001	0,824534452998555	4
b6	0,851032823659375	0,847463131288141	3

**Tabel 3. 19** Hasil Evaluasi *Insert Sequence*

LEBAH	URUTAN KOTA						fit-MCTSP	trial
	1	2	3	4	5	6		
b1	0	3	2	4	1	5	0,85547965	3
b2	0	1	3	4	2	5	0,83039336	4
b3	0	2	4	1	3	5	0,85657584	4
b4	0	5	4	1	2	3	0,84385018	3
b5	0	4	3	5	1	2	0,82780772	4
b6	0	2	3	5	1	4	0,85103282	3

Individu dengan nilai *fitness* terbesar dan nilai trial terendah akan disimpan sebagai solusi terbaik atau *global best* (Gbest). Berdasarkan hasil improvisasi fase *employed bee* dan fase *onlooker bee* pada tabel 3.19 diperoleh Gbest adalah  $b_3 = (0,2,4,1,3,5)$  dengan nilai *fitness* 0,8565758.

### 3.4.5 Fase *Scout Bee*

Pada fase *scout bee*, individu yang sulit diperbaharui akan diganti dengan individu baru. Jumlah *scout bee* bergantung pada jumlah individu yang telah memiliki nilai *trial* lebih besar dari Limit yang telah ditentukan. Apabila nilai *trial* dari lebah yang melakukan perbaikan solusi melebihi nilai *limit* yang sudah ditetapkan, maka solusi dari individu tersebut akan dihilangkan dan digantikan dengan individu baru secara acak. Selanjutnya dihitung kembali nilai *fitness* dan nilai *trial*-nya kembali menjadi 0. Jika nilai *trial* dari lebah yang melakukan perbaikan kurang dari *limit* yang

telah ditentukan, maka solusi dari tiap individu dan nilai *trial*-nya tidak berubah. Berdasarkan Tabel 3.16, lebah  $b_2$  mengalami perbaikan sehingga solusi dari individu tersebut diganti dengan individu yang baru dan nilai *trial* diatur kembali menjadi 0 seperti pada Tabel 3.20.

**Tabel 3. 20** Hasil Evaluasi *Scout Bee*

Lebah	Urutan Kota						fit-MCTSP	trial
	1	2	3	4	5	6		
b1	0	3	2	4	1	5	0,85547965	3
b2	0	5	1	3	2	4	0,84403289	0
b3	0	4	1	2	3	5	0,85591462	0
b4	0	5	4	1	2	3	0,84385018	3
b5	0	5	4	3	1	2	0,82733060	0
b6	0	2	3	5	1	4	0,85103282	3

Karena kriteria pemberhentian yaitu *max iteration* = 1 telah terpenuhi, maka Tabel 3.20 merupakan solusi akhir dengan Gbest adalah  $b_3 = (0,2,4,1,3,5)$  dengan nilai *fitness* 0,8565758