

### **BAB III**

## **PENYELESAIAN MODEL TRAVELING SALESMAN PROBLEM MENGUNAKAN ALGORITMA IMPROVED ANT COLONY OPTIMIZATION**

Bab ini membahas tentang masalah penentuan rute terpendek yang dimodelkan dalam bentuk *Traveling Salesman Problem* (TSP). Selanjutnya akan dibahas pula teknik penyelesaiannya menggunakan *Algoritma Improved Ant Colony Optimization* (IACO).

### **3.1 *Travelling Salesman Problem* (TSP)**

TSP dapat digambarkan sebagai permasalahan dimana terdapat sebuah kendaraan yang akan mendistribusikan makanan ke sejumlah konsumen yang berada pada lokasi tertentu. Untuk mengirimkan makanan tersebut, kendaraan memulai perjalanannya dari sebuah tempat, yang disebut depot, lalu melakukan perjalanan ke sejumlah lokasi. Kendaraan harus mendistribusikan makanan ke setiap lokasi tepat satu kali dan kemudian kembali lagi ke depot.

Tujuan dari penyelesaian TSP adalah menentukan rute perjalanan dari depot ke semua lokasi agar total jarak perjalanan yang ditempuh adalah minimum. Dalam penelitian ini, untuk menyelesaikan model TSP tersebut, digunakan *Algoritma Improved Ant Colony Optimization*.

### **3.2 Model Optimisasi dari TSP**

Masalah TSP dapat dimodelkan sebagai model optimasi dengan sebuah fungsi tujuan dan sejumlah kendala. Asumsi-asumsi yang digunakan pada TSP ini adalah sebagai berikut :

1. Kecepatan kendaraan dan waktu perjalanan diabaikan.
2. Terdapat rute jalan yang menghubungkan setiap lokasi.
3. Hanya terdapat satu kendaraan.
4. Perjalanan dimulai dari depot, kemudian menuju setiap lokasi dengan tepat satu kali, dan kembali lagi ke depot.
5. Jarak dari lokasi  $i$  ke lokasi  $j$  sama dengan jarak dari lokasi  $j$  ke lokasi  $i$

Berdasarkan Christofides (1979), tahapan pertama pemodelan adalah mendefinisikan himpunan, parameter dan variabel keputusan yang digunakan dalam model ini. Misalkan  $I$  adalah himpunan lokasi dan  $d_{ij}$  adalah jarak tempuh dari lokasi  $i$  ke lokasi  $j$ . Variabel keputusan  $x_{ij}$ , menentukan apakah kendaraan melakukan perjalanan atau tidak dari lokasi  $i$  ke lokasi  $j$ , yang didefinisikan sebagai berikut.

$$x_{ij} = \begin{cases} 1, & \text{jika kendaraan melakukan perjalanan dari } i \text{ ke } j \\ 0, & \text{yang lainnya} \end{cases}$$

Fungsi tujuan dari model optimasi adalah untuk mencari rute yang meminimumkan total jarak tempuh dalam pendistribusian makanan ke lokasi. Fungsi tersebut didefinisikan sebagai berikut.

Meminimumkan :

$$f = \sum_{i \in I} \sum_{j \in I} d_{ij} x_{ij} .$$

Kendala dari model adalah batasan yang harus dipenuhi dalam menentukan rute yang dilalui. Kendala pertama adalah kendala yang menjamin bahwa hanya ada satu jalan yang masuk ke setiap lokasi. Kendala ini diekspresikan sebagai berikut.

$$\sum_{i \in I} x_{ij} = 1, \forall j \in I .$$

Kendala kedua adalah kendala yang menjamin bahwa hanya ada satu jalan yang keluar dari setiap lokasi. Kendala ini diekspresikan sebagai berikut.

$$\sum_{j \in I} x_{ij} = 1, \forall i \in I .$$

Kendala ketiga adalah kendala yang menjamin bahwa  $x_{ij}$  harus membentuk perjalanan, sehingga

$$\sum_{i \in S_t} \sum_{j \in \bar{S}_t} x_{ij} \geq 1, \forall S_t \subset V$$

dimana  $\bar{S}_t = V - S_t$ , dengan  $S_t$  adalah himpunan lokasi yang telah dikunjungi dan  $\bar{S}_t$  adalah himpunan lokasi yang belum dikunjungi. Untuk selanjutnya, lokasi konsumen tersebut akan disebut sebagai titik.

### 3.3 Implementasi Algoritma *Improved Ant Colony Optimization* (IACO) pada *Traveling Salesman Problem* (TSP)

Algoritma IACO telah diimplementasikan untuk menyelesaikan masalah pencarian rute terpendek oleh beberapa peneliti. Beberapa penelitian mengenai pencarian rute terpendek tersebut dilakukan untuk model yang berbeda. Hasil dari penelitian-penelitian tersebut menunjukkan bahwa penggunaan Algoritma IACO memberikan hasil yang lebih baik dibandingkan dengan Algoritma ACO.

Pada penelitian ini, Algoritma IACO akan diimplementasikan pada TSP. Terdapat 5 tahapan dalam Algoritma IACO, yaitu inisialisasi, konstruksi rute, mutasi, *local search*, dan *update pheromone*. Penjelasan untuk setiap tahapan tersebut akan dibahas pada subbab berikutnya.

#### 3.3.1 Inisialisasi

Pada tahapan ini, parameter-parameter yang digunakan pada Algoritma IACO akan diinisialisasi terlebih dahulu. Parameter-parameter yang diperlukan pada algoritma semut adalah sebagai berikut:

1. Intensitas jejak semut antar titik ( $\tau_{ij}$ )

Parameter ini diinisialisasikan untuk iterasi pertama. Parameter ini digunakan untuk mencari peluang pemilihan titik yang akan dikunjungi selanjutnya pada suatu rute.

2. Tetapan perjalanan semut ( $Q$ )

Parameter ini merupakan konstanta yang digunakan untuk menentukan perubahan intensitas jejak semut antar titik ( $\Delta\tau_{ij}$ ) yang akan menentukan nilai  $\tau_{ij}$  pada iterasi selanjutnya. Nilai  $Q$  ditentukan oleh penulis.

3. Tetapan pengendali intensitas jejak semut ( $\alpha$ )

Parameter ini digunakan untuk mencari peluang pemilihan titik yang akan dikunjungi selanjutnya pada suatu rute dan berfungsi sebagai pengendali intensitas jejak semut. Nilai  $\alpha$  ditentukan oleh penulis.

4. Visibilitas antar titik ( $\eta_{ij}$ )

Parameter ini digunakan untuk mencari peluang pemilihan titik yang akan dikunjungi selanjutnya pada suatu rute. Nilai  $\eta_{ij}$  merupakan hasil dari perhitungan  $\frac{1}{d_{ij}}$ , dimana nilai  $d_{ij}$  adalah jarak antar titik.

5. Tetapan pengendali visibilitas ( $\beta$ )

Parameter ini digunakan untuk mencari peluang pemilihan titik yang akan dikunjungi selanjutnya pada suatu rute dan berfungsi sebagai pengendali nilai visibilitas antar titik ( $\eta_{ij}$ ). Nilai  $\beta$  ditentukan oleh penulis.

6. Banyak semut ( $m$ )

Parameter ini menentukan banyaknya semut yang akan melakukan iterasi dalam algoritma ini. Nilai  $m$  ditentukan oleh penulis.

7. Tetapan penguapan jejak semut ( $\rho$ )

Parameter ini digunakan untuk menentukan nilai  $\tau_{ij}$  untuk iterasi selanjutnya. Nilai  $\rho$  ditentukan oleh penulis.

8. Jumlah iterasi maksimum (NCmax)

NCmax adalah jumlah maksimum iterasi yang akan terjadi. Iterasi akan berhenti sesuai dengan NCmax yang telah ditentukan. Nilai NCmax ditentukan oleh penulis.

Setelah menentukan nilai untuk parameter-parameter tersebut, langkah selanjutnya adalah menempatkan setiap semut pada sebuah titik. Kemudian setiap semut akan melakukan perjalanan dan membuat rutenya masing-masing.

### 3.3.2 Konstruksi Rute

1. Mengkonstruksi Rute Awal

Semut yang sudah terdistribusi ke setiap titik akan mulai melakukan perjalanannya masing-masing. Semua semut ( $k$ ) akan melakukan perjalanan dari titik pertamanya ( $i$ ), menuju ke salah satu dari titik-titik lainnya ( $j$ ). Titik yang terpilih tersebut akan menempati sebuah tabu yang disebut dengan tabu ke- $k$ , dilambangkan dengan  $\text{tabu}_k$ . Perjalanan semua semut berlangsung terus menerus sampai

semua titik satu persatu dikunjungi dan menempati tabunya masing-masing, yang kemudian akan membentuk sebuah rute.

Untuk menentukan titik yang akan menjadi tujuan semut dari titik  $i$  ke titik  $j$  digunakan persamaan probabilitas sebagai berikut:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in \text{tabu}_k} [\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta}, & \text{jika } j \in \text{tabu}_k \\ 0, & \text{untuk lainnya} \end{cases}$$

Misalkan, terdapat 6 titik yaitu A, B, C, D, E dan F. Kemudian setiap semut akan disebar ke setiap titik sebagai titik asal. Misalkan semut 1 ditempatkan pada titik A, semut 2 ditempatkan pada titik 2, dan seterusnya. Akan dicari nilai probabilitas terbesar dari setiap titik ke titik lainnya untuk menentukan titik tujuannya. Misalkan untuk semut 1 memulai dari titik A. Kemudian diperoleh nilai  $p_{AB}^1 = 0,8$ ,  $p_{AC}^1 = 0,9$ ,  $p_{AD}^1 = 0,5$ ,  $p_{AE}^1 = 0,4$ ,  $p_{AF}^1 = 0,3$ , maka titik C terpilih menjadi titik tujuannya. Titik C akan masuk ke dalam tabu semut ke-1, yang dilambangkan dengan  $\text{tabu}_1 = \{A, C\}$ . Selanjutnya akan dihitung nilai probabilitas semut 1 dari titik C. Misalkan terpilih titik B, maka tabu akan menjadi  $\text{tabu}_1 = \{A, C, B\}$ . dan seterusnya hingga semua titik masuk ke dalam tabu dan kembali lagi ke titik awal. Lakukan cara yang sama untuk semua semut, hingga tabu-tabu tersebut akan membentuk sebuah rute perjalanan.

## 2. Menghitung Panjang Rute

Perhitungan panjang rute setiap semut yang disimbolkan dengan  $L_k$ , dilakukan setelah semua semut selesai melakukan perjalanannya dan kembali ke titik awal. Perhitungan dilakukan berdasarkan  $\text{tabu}_k$  masing-masing dengan menggunakan persamaan berikut:

$$L_k = d_{\text{tabu}_k(n), \text{tabu}_k(1)} + \sum_{i=1}^{n-1} d_{\text{tabu}_k(s), \text{tabu}_k(s+1)}$$

dimana  $d_{ij}$  adalah jarak antara titik  $i$  ke titik  $j$ ,  $s$  adalah titik yang akan dikunjungi, dan  $n$  adalah banyaknya titik.

Misalkan rute yang terbentuk dari semut 1 adalah  $\text{tabu}_1 = \{A, C, B, D, E, F\}$ . Maka nilai  $L_k$  dari  $\text{tabu}_1$  adalah  $L_1 = d_{AC} + d_{CB} + d_{BD} + d_{DE} +$

$d_{EF} + d_{FA}$ . Setelah panjang rute setiap semut dihitung, akan diperoleh nilai minimal panjang rute dari semua semut.

### 3.3.3 Mutasi

Mutasi adalah tahapan yang tidak ada dalam Algoritma ACO. Sehingga tahapan ini menjadi salah satu pembedanya dengan Algoritma IACO. Tujuan dari mutasi ini untuk mendapat nilai yang lebih baik sehingga mendapat rute perjalanan terpendek dalam penelitian ini. Mutasi yang digunakan pada penelitian ini adalah *swapping mutation*. Jenis mutasi ini adalah mutasi yang cocok dengan kasus TSP. Proses mutasi dilakukan dengan cara mengganti posisi titik yang terpilih dengan posisi titik lainnya yang dipilih secara acak. Setiap titik yang akan dimutasi ditentukan oleh nilai parameter yang dinamakan *mutation rate* ( $\rho_m$ ). Akan dipilih nilai acak  $p$  dari 0 sampai 1 pada setiap titik untuk semua rute. Apabila terpenuhi  $p < \rho_m$ , maka akan terjadi mutasi. Berdasarkan Bin (2009), persamaan yang digunakan untuk mencari nilai  $\rho_m$  adalah sebagai berikut:

$$\rho_m(k) = \rho_m^{min} + (\rho_m^{max} - \rho_m^{min})^{1-\frac{k}{n}}$$

dimana  $\rho_m^{min} = \frac{1}{n}$  dan  $\rho_m^{max} = \frac{1}{m}$ , dengan  $k$  adalah urutan semut,  $m$  adalah jumlah semut, dan  $n$  adalah jumlah titik yang ada pada satu rute.

Misal rute yang terbentuk dari 6 semut adalah sebanyak 6 rute dan dari semut 1 diperoleh rute 1 yaitu  $tabu_1 = \{A, C, B, D, E, F\}$ . Sehingga diperoleh nilai  $\rho_m(1) = \frac{1}{6} + (\frac{1}{6} - \frac{1}{6})^{1-\frac{1}{6}} = 0,2$ . Untuk setiap titik yang ada pada rute 1, akan dibangkitkan nilai acak dari 0 sampai 1. Jika pada urutan titik pertama, yaitu titik A terpilih nilai acak 0,05. Berarti nilai tersebut kurang dari nilai  $\rho_m(1) = 0,2$ , maka urutan titik pertama akan ditukar dengan titik urutan yang lain. Titik urutan yang lain tersebut akan dipilih secara acak dari 1 hingga sebanyak titik yang ada, yaitu 6 titik. Misal terpilih titik urutan keempat, maka urutan titik pertama ditukar dengan urutan titik keempat. Jika untuk titik A yang terpilih nilai acak 0,3 yang artinya lebih besar dari  $\rho_m(1) = 0,2$ , maka tidak terjadi proses mutasi.

Lakukan hal yang sama untuk setiap titik yang ada pada setiap rute yang terbentuk. Hasil dari mutasi diilustrasikan pada Gambar 3.1



Gambar 3.1 *Swapping Mutation*

Setelah menyelesaikan proses mutasi maka akan dilanjutkan ke tahap selanjutnya, yaitu tahapan *local search*.

### 3.3.4 Local Search

*Local search* yang digunakan pada Algoritma IACO ini merupakan tahapan yang bermaksud untuk menentukan nilai jarak yang terbaik. Nilai tersebut akan diperoleh dari nilai 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 diabaikan. Sedangkan setiap rute yang tidak mengalami proses mutasi, akan masuk proses *local search* [2], yaitu langsung digunakan nilai total jaraknya.

Misalkan, pada proses konstruksi rute diperoleh rute  $a$  dan dihasilkan rute baru dari hasil proses mutasi yaitu rute  $a'$ . Kemudian hitung nilai total jarak dari rute  $a$  dan rute  $a'$ . Jika rute  $a'$  menghasilkan nilai total jarak yang lebih baik dari rute  $a$ , maka dipilih rute  $a'$  menjadi rute baru.

### 3.3.5 Update Pheromone

Semua semut akan meninggalkan jejak *pheromone* pada setiap rute yang dilaluinya. Pada tahapan ini akan dilakukan perhitungan nilai intensitas jejak semut antar titik ( $\tau_{ij}$ ) yang baru untuk digunakan pada iterasi selanjutnya. Nilai intensitas jejak semut antar titik pada semua rute perjalanan kemungkinan bisa berubah. Hal tersebut terjadi karena adanya penguapan dan perbedaan jumlah semut yang melewati setiap titiknya. Perhitungan perubahan nilai intensitas jejak semut antar titik disimbolkan dengan  $\Delta\tau_{ij}$ . Persamaan perubahannya adalah sebagai berikut:

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$$

dengan  $\Delta\tau_{ij}^k$  adalah nilai perubahan nilai intensitas jejak semut ke-k yang melewati titik  $(i,j)$  dan  $m$  adalah banyaknya semut. Nilai tersebut dihitung berdasarkan persamaan yang menggunakan parameter  $Q$ , yaitu konstanta nilai tetapan perjalanan semut. Persamaan untuk mencari nilai  $\Delta\tau_{ij}^k$  adalah sebagai berikut:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{jika semut ke } k \text{ melewati titik } (i,j) \\ 0, & \text{untuk lainnya} \end{cases}$$

Nilai intensitas jejak semut antar titik untuk iterasi selanjutnya dihitung dengan persamaan menggunakan parameter  $\rho$ , yaitu nilai tetapan penguapan jejak semut. Persamaan untuk mencari nilai  $\tau_{ij}$  pada iterasi selanjutnya adalah sebagai berikut:

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_{ij}$$

Untuk memulai perjalanan baru lakukan dengan menggunakan tabu list baru, kemudian lakukan kembali langkah pembentukan rute jika diperlukan. Tabu list baru akan diisi kembali dengan urutan tempat yang baru pada iterasi selanjutnya. Lakukan semua tahapan hingga sejumlah iterasi maksimum, yaitu  $NC_{\max}$  yang nilainya ditentukan oleh penulis.