

BAB III

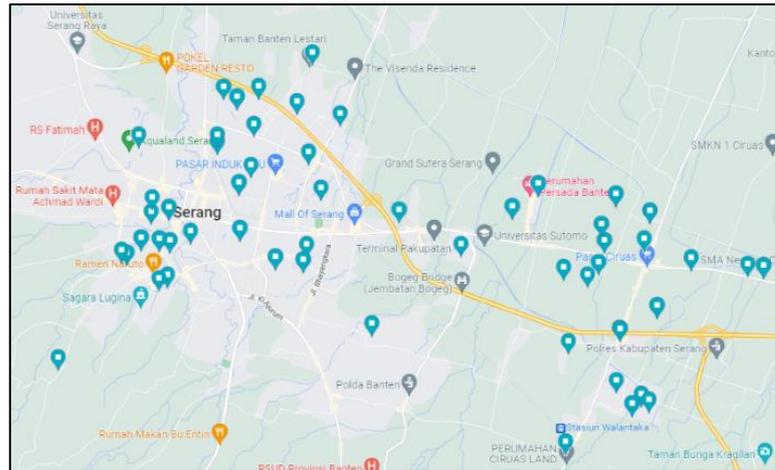
METODE PENELITIAN

Bab ini membahas metodologi yang digunakan untuk menentukan rute penjemputan sampah terpilah dari bank sampah agen oleh bank sampah induk Kota & Kabupaten Serang dengan mengaplikasikan penyelesaian *Multi Travelling Salesman Problem* (MTSP) menggunakan Algoritma Genetika. Langkah penyelesaian dimulai dengan mengidentifikasi dan mendeskripsikan masalah, lalu mengonstruksi model optimisasi dengan mendefinisikan himpunan, parameter dan variabelnya dan diakhiri dengan menyelesaikan model optimisasi tersebut menggunakan Algoritma Genetika.

3.1 Deskripsi Masalah

Penelitian ini akan membahas penentuan rute penjemputan sampah terpilah dari bank sampah agen oleh bank sampah induk kota & kabupaten Serang dengan mengaplikasikan penyelesaian *Multi Travelling Salesman Problem* (MTSP) menggunakan Algoritma Genetika. MTSP adalah masalah optimasi yang merupakan perluasan dari *Travelling Salesman Problem* (TSP) berupa penentuan rute terpendek atau rute optimal kunjungan beberapa kota oleh beberapa *salesman*. Dalam permasalahan yang diangkat oleh penulis, lokasi *bank* sampah agen di Kabupaten & Kota Serang merepresentasikan kota tujuan dan petugas bank sampah yang melakukan penjemputan merepresentasikan *salesman*.

Penelitian ini mengaplikasikan MTSP dengan tujuan agar didapat rute terpendek petugas bank sampah dalam menjemput sampah terpilah di bank sampah agen tanpa adanya tumpang tindih antar petugas dan tidak ada bank sampah agen yang luput untuk dijemput sampah terpilahnya. Untuk menyelesaikan permasalahan ini, tahapan pertama yang dilakukan ialah menganalisis masalah. Analisis masalah ini bertujuan untuk mengetahui sejumlah sumber daya yang dimiliki serta tujuan pemetaan yang harus dipenuhi. Dari analisis masalah akan diketahui variabel input yang digunakan dan variabel output yang akan diharapkan dalam penyelesaian Algoritma Genetika untuk penentuan rute penjemputan sampah terpilah.



Gambar 3.1 Peta Lokasi Bank Sampah Induk & Bank Sampah Agen

Pada Bank Sampah Digital (BSD) selaku bank sampah induk di Kota dan Kabupaten Serang, terdapat 5 petugas yang ditugaskan untuk menjemput sampah terpilah dari 60 titik bank sampah agen. Lokasi bank sampah induk dan bank sampah agen diilustrasikan melalui Gambar 3.1.

3.2 Model Matematika untuk *Multi Travelling Salesman Problem (MTSP)*

Pada bagian ini dilakukan tahapan mengonstruksi model matematika untuk mengoptimalkan rute penjemputan sampah terpilah berdasarkan permasalahan dan data yang diperoleh. Model matematika dibangun berdasarkan referensi model MTSP yang telah ada dengan memperhatikan beberapa asumsi, yakni:

1. Bank sampah agen yang disertakan dalam penelitian ini hanya 60 bank sampah agen yang bertempat di Kota dan Kabupaten Serang, sedangkan bank sampah agen dilokasi lainnya seperti Kota Cilegon diabaikan.
2. Hanya terdapat satu depot sebagai titik awal sekaligus titik akhir kunjungan petugas bank sampah yakni lokasi kantor Bank Sampah Digital (BSD).
3. Setiap lokasi bank sampah agen dikunjungi tepat sekali oleh satu petugas bank sampah.
4. Banyaknya lokasi bank sampah agen yang dikunjungi oleh petugas bank sampah ialah sama banyak.

5. Setiap jalan dapat dilewati dan jarak (berangkat) dari bank sampah agen i ke bank sampah agen j sama dengan jarak (kembali) dari bank sampah agen j ke bank sampah agen i .
6. Seluruh jalan yang dilewati dianggap dalam keadaan normal atau tidak menghiraukan kepadatan jalan (kemacetan), rambu lalu lintas dan perbaikan jalan yang rusak.
7. Kapasitas kendaraan yang digunakan dalam penjemputan sampah bernilai besar atau setiap kendaraan yang digunakan petugas dianggap mampu mengangkut sampah terpilah dari seluruh bank sampah agen dalam satu rute perjalanan.
8. Seluruh bank sampah agen dalam rute penjemputan sedia untuk melakukan penimbangan ketika petugas melakukan penjemputan sampah terpilah.

Dalam mengonstruksi model, perlu didefinisikan himpunan, parameter serta variabel yang terlibat. Pada penyelesaian penentuan rute penjemputan sampah terpilah ini himpunan yang digunakan ialah himpunan bank sampah (N : *Himpunan Bank Sampah Agen*) dan himpunan petugas bank sampah yang akan melakukan penjemputan (V : *Himpunan petugas*). Adapun parameter yang digunakan ialah C_{ij} sebagai jarak dari bank sampah agen i ke bank sampah agen j , m menyatakan banyaknya jumlah petugas bank sampah yang akan menjemput sampah terpilah dari bank sampah agen, serta n menyatakan banyaknya jumlah bank sampah agen yang harus dikunjungi.

Selanjutnya variabel keputusan pada model MTSP menentukan ada atau tidaknya perjalanan petugas bank sampah dimulai dari depot (titik awal), yang direpresentasikan bilangan 0, ke lokasi bank sampah agen i hingga ke lokasi bank sampah agen j lalu kembali ke depot sebagai akhir rute perjalanan. Variabel keputusan ini dinyatakan dalam nilai x_{ijk} dimana x_{ijk} akan bernilai 1 jika terdapat perjalanan seorang petugas dari lokasi bank sampah agen i ke lokasi bank sampah agen j dan x_{ijk} akan bernilai 0 jika tidak ada perjalanan seorang petugas dari lokasi bank sampah agen i ke lokasi bank sampah agen j . Dengan demikian, variable keputusan model adalah sebagai berikut:

$$X_{ijk} = \begin{cases} 1, & \text{jika petugas } k \text{ melakukan perjalanan dari lokasi } i \text{ ke lokasi } j \\ 0, & \text{jika petugas } k \text{ tidak melakukan perjalanan dari lokasi } i \text{ ke lokasi } j \end{cases}$$

Fungsi tujuan model MTSP yang dibangun dalam penelitian ini ialah meminimumkan total jarak seluruh petugas bank sampah dalam melakukan penjemputan sampah terpilah di bank sampah agen. Berdasarkan penjabaran tersebut fungsi tujuan tersebut dapat ditulis sebagai berikut:

$$f = \min \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ijk}$$

Adapun kendala-kendala dari model adalah sebagai berikut:

1. Setiap bank sampah agen harus dikunjungi tepat satu kali

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

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

2. Setiap petugas berangkat dari depot untuk setiap satu rute perjalanan yang dilakukan.

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

3. Setiap petugas kembali ke depot untuk setiap satu rute perjalanan yang dilakukan.

$$\sum_{i \in N} X_{i0k} = 1, \forall k \in V$$

4. Variabel keputusan X_{ijk} bernilai biner

$$X_{ijk} \in \{0,1\}, i \neq j, i, j \in N$$

5. Rute perjalanan seluruh petugas terhubung atau tidak ada sub-rute yang tidak terhubung. Kendala mengeliminasi subtour ini dikenal dengan formula Miller–Tucker–Zemlin. Dalam formula ini digunakan variabel tambahan u_{ik} dan u_{jk} yang menyatakan nilai simpul (bank sampah agen) jika petugas k bergerak dari simpul (bank sampah agen) i ke simpul (bank sampah agen) j . Selain itu didefinisikan variabel p yang menunjukkan jumlah maksimum simpul (bank sampah agen) yang dapat dikunjungi oleh petugas. Kendala ini dinyatakan sebagai:

$$u_{ik} - u_{jk} + p \cdot X_{ijk} \leq p - 1, \forall 2 \leq i \neq j \leq n, i, j \in N, k \in V$$

Selengkapnya, model MTSP pada penelitian ini dapat dimodelkan sebagai model optimasi sebagai berikut:

$$f = \min \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ijk} \quad (3.1)$$

Terhadap:

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

$$\sum_{k \in V} \sum_{j \in N} X_{ijk} = 1, \forall i \in N \quad (3.3)$$

$$\sum_{j \in N} X_{0jk} = 1, \forall k \in V \quad (3.4)$$

$$\sum_{i \in N} X_{i0k} = 1, \forall k \in V \quad (3.5)$$

$$X_{ijk} \in \{0,1\}, i \neq j, i, j \in N \quad (3.6)$$

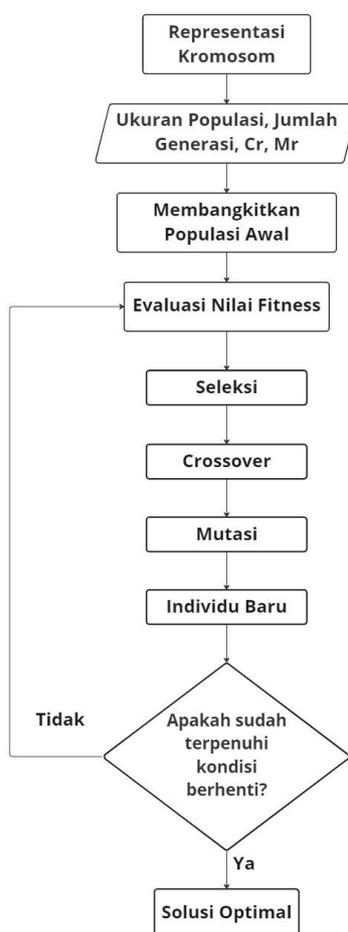
$$u_{ik} - u_{jk} + p \cdot X_{ijk} \leq p - 1, \forall 2 \leq i \neq j \leq n, i, j \in N, \quad (3.7)$$

Model optimisasi di atas termasuk dalam kategori model *integer linear programming*. Pada bagian selanjutnya dari bab ini akan dibahas penyelesaian model optimisasi dengan menggunakan Algoritma Genetika.

3.3 Penyelesaian MTSP dengan Algoritma Genetika

Berbagai metode algoritma heuristik atau metaheuristik cukup efektif untuk menyelesaikan masalah MTSP, meskipun solusi yang dihasilkan adalah solusi pendekatan. Salah satu metode yang dapat digunakan ialah Algoritma Genetika. Algoritma Genetika merupakan algoritma pencarian dan model ilmiah evolusi alami yang diangkat dari mekanisme seleksi alam oleh Darwin. Ide dasar dari Algoritma Genetika adalah untuk mensimulasikan evolusi populasi individu yang mewakili masalah pencarian tertentu, mempromosikan kelangsungan hidup dan reproduksi yang paling cocok. Algoritma Genetika menjadi metode yang relatif

banyak dipilih karena kemampuannya untuk menyelesaikan berbagai masalah kompleks.



Gambar 3.2 Langkah-Langkah Algoritma Genetika

Cara kerja Algoritma Genetika dapat dilihat pada Gambar 3.2. Pada Sub bab Selanjutnya akan dibahas lebih rinci setiap tahapan dari Algoritma Genetik yang digunakan pada penelitian ini disertai dengan ilustrasi pengerjaannya menggunakan contoh kasus.

3.3.1 Representasi Kromosom pada MTSP

Kromosom ialah string yang terdiri dari sekumpulan bit. Adapun bit tersebut merupakan nilai yang menyatakan satuan dasar tertentu dan biasa dikenal dengan gen. Dalam artian lain, kromosom ialah kumpulan gen yang merepresentasikan suatu nilai dan menyatakan solusi yang mungkin dari permasalahan. Kromosom dibangun pada tahap pengkodean atau representasi kromosom.

Kromosom yang dibangun pada tahap pengkodean akan menjadi bagian krusial implementasi Algoritma Genetika. Tahap ini harus dilalui karena pengkodean untuk setiap permasalahan dapat berbeda-beda. Misalnya gen, yang dapat direpresentasikan dalam bentuk bit, bilangan real, daftar aturan, elemen permutasi, elemen program atau representasi lainnya yang dapat diimplementasikan untuk operator genetika. Pada penelitian ini, penulis mengasumsikan bahwa banyaknya lokasi yang dikunjungi setiap petugas adalah sama. Oleh karena itu, representasi kromosom yang digunakan adalah *permutation encoding* atau pengkodean berupa kumpulan nilai *integer* yang mewakili suatu posisi dalam urutan. Kromosom yang digunakan terdiri dari n gen di mana n menyatakan banyaknya lokasi.

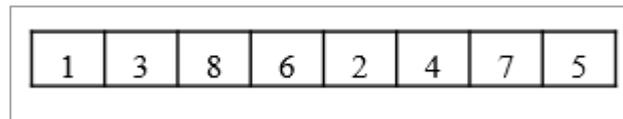
Sebagai contoh, akan dicari jarak minimum untuk 2 petugas pengantaran yang akan mengunjungi 8 lokasi. Jarak antar lokasi dan depot sebagai titik awal keberangkatan dan titik kepulangan petugas yang direpresentasikan sebagai lokasi ke-0 dapat dilihat pada Tabel 3.1.

Tabel 3.1 Contoh Tabulasi Data Jarak Antar Lokasi pada Contoh Kasus

Lokasi/ Jarak (km)	0	1	2	3	4	5	6	7	8
0	0	4,0	4,0	4,0	4,3	4,5	4,1	5,2	5,4
1	4,0	0	0,1	0,1	0,3	0,5	0,1	1,2	1,4
2	4,0	0,1	0	0,1	0,3	0,5	0,1	1,2	1,4
3	4,0	0,1	0,1	0	0,3	0,5	0,1	1,2	1,4
4	4,3	0,3	0,3	0,3	0	0,2	0,2	0,9	1,1
5	4,5	0,5	0,5	0,5	0,2	0	0,4	0,7	0,9
6	4,1	0,1	0,1	0,1	0,2	0,4	0	1,1	1,3
7	5,2	1,2	1,2	1,2	0,9	0,7	1,1	0	0,2
8	5,4	1,4	1,4	1,4	1,1	0,9	1,3	0,2	0

Masalah pengantaran tersebut termasuk dalam MTSP karena terdapat lebih dari satu petugas atau *salesman*. Adapun contoh representasi kromosom untuk masalah tersebut digambarkan pada Gambar 3.3. Dalam sebaris kromosom terdiri dari 8 gen

yang mewakili urutan lokasi yang akan dikunjungi oleh petugas. Dalam kasus ini jumlah lokasi yang dikunjungi masing-masing petugas ada sebanyak 4 lokasi.



Gambar 3.3 Contoh Representasi Kromosom dengan *Permutation Encoding*

3.3.1 Penentuan Parameter Algoritma Genetika

Algoritma Genetik bekerja bergantung dari beberapa nilai parameter yang harus ditetapkan diawal proses. Parameter-parameter tersebut adalah sebagai berikut:

1. Banyaknya generasi (iterasi)

Banyaknya jumlah iterasi satu siklus reproduksi Algoritma Genetika berupa seleksi, *crossover*, *mutasi* yang tiap generasinya menghasilkan solusi permasalahan.

2. Ukuran populasi awal

Banyaknya jumlah individu solusi yang akan diproses bersama dalam satu satuan siklus evolusi.

3. Peluang *crossover*

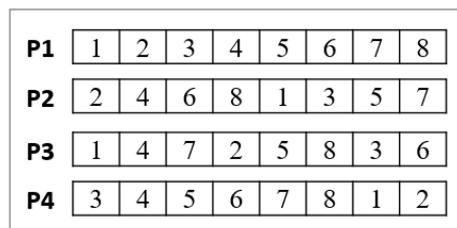
Parameter penentu jumlah gen dalam populasi yang mengalami *crossover*.

4. Peluang mutasi

Parameter penentu jumlah gen dalam populasi yang mengalami mutasi.

3.3.2 Pembangkitan Populasi Awal pada MTSP

Pada generasi ke-0 atau sebelum dimulainya tahapan algoritma genetika, dilakukan pembangkitan acak semisal dengan *random walk* untuk membentuk populasi awal. Dengan jumlah atau banyaknya populasi awal dapat ditentukan terlebih dahulu melalui berbagai cara. Misal pada kasus di atas akan dibangkitkan populasi awal sebanyak 4 individu secara acak, dari seluruh kemungkinan individu dalam populasi sebanyak 8! (8 faktorial).



Gambar 3.4 Contoh Inisialisasi Populasi Awal pada Contoh Kasus

Dari contoh inialisasi populasi pada Gambar 3.4 dapat diperoleh informasi bahwa pada P1 (individu/calon solusi 1) petugas A akan berangkat dari depot menuju lokasi 1, lalu ke lokasi 2, lalu ke lokasi 3, lalu ke lokasi 4 kemudian kembali ke depot. Sedangkan petugas B berangkat dari depot ke lokasi 5, lalu lokasi 6, lalu lokasi 7, lalu lokasi 8, kemudian kembali ke depot. Demikian pula informasi alternatif solusi perjalanan petugas A dan B lainnya yang dapat diperoleh pada Tabel 3.2.

3.3.3 Menghitung Nilai *Fitness* (Nilai Kesesuaian) pada MTSP

Hal penting selanjutnya yang perlu dilakukan dalam Algoritma Genetika adalah pemilihan individu untuk menghasilkan keturunan berikutnya. Pemilihan ini berdasarkan nilai kesesuaian atau yang biasa dikenal dengan nilai *fitness*. Selain untuk menyatakan baik atau buruknya suatu individu solusi, nilai *fitness* ini juga dijadikan acuan dalam mencapai nilai optimal. Nilai *fitness* pada masalah MTSP dapat diperoleh berdasarkan total jarak yang ditempuh seluruh *salesman* (dalam kasus di atas ialah petugas yang akan berkunjung ke 8 lokasi) dari titik awal atau depot menuju seluruh lokasi tujuan dalam satu rute perjalanan dan kembali lagi ke titik awal (depot). Nilai *fitness* ini dapat diperoleh dari persamaan berikut:

$$\text{nilai fitness} = \frac{1}{\sum C_{ij}}$$

Dengan C_{ij} adalah total jarak yang ditempuh oleh seluruh petugas.

Berikut contoh perhitungan nilai *fitness* untuk P1: 1 2 3 4 5 6 7 8 berdasarkan data jarak pada Tabel 3.1.

Tabel 3.2 Ilustrasi Perhitungan Jarak Tempuh pada Contoh Kasus

Petugas A		Petugas B	
Perjalanan	Jarak Tempuh (km)	Perjalanan	Jarak Tempuh (km)
Dari 0 ke 1	4,0	Dari 0 ke 5	4,5
Dari 1 ke 2	0,1	Dari 5 ke 6	0,4
Dari 2 ke 3	0,1	Dari 6 ke 7	1,1
Dari 3 ke 4	0,3	Dari 7 ke 8	0,2
Dari 4 ke 0	4,3	Dari 8 ke 0	5,4
Total	8,8	Total	11,6

Berdasarkan Tabel 3.3 total jarak yang ditempuh seluruh petugas pada P1 ialah 20,4 km. Dengan cara yang sama dapat diperoleh total jarak untuk P2 sebesar 21,7 km, P3 sebesar 21,4 km dan P4 sebesar 19,9 km. Dengan mengetahui jumlah total jarak yang ditempuh oleh seluruh petugas maka dapat dicari nilai *fitness* untuk setiap individu dalam populasi. Tabel 3.4 berikut menunjukkan nilai *fitness* untuk setiap individu dalam populasi

Tabel 3.3 Contoh Perhitungan Nilai *Fitness* Individu pada Contoh Kasus

Individu	Total Jarak	Fungsi <i>Fitness</i>	Nilai <i>Fitness</i>
P1	20,4	$\frac{1}{20,4}$	0,0490
P2	21,7	$\frac{1}{21,7}$	0,0460
P3	21,4	$\frac{1}{21,4}$	0,0467
P4	19,9	$\frac{1}{19,9}$	0,0502

3.3.4 Seleksi Parents pada MTSP

Proses seleksi bertujuan untuk mendapatkan calon induk yang baik untuk dilanjutkan pada proses selanjutnya yakni proses *Crossover* atau perkawinan silang. Terdapat beberapa macam cara seleksi untuk mencari calon induk yang baik diantaranya seleksi dengan mesin roulette atau *Roulette Wheel* dan seleksi dengan turnamen. Pada penelitian ini, penulis akan menggunakan seleksi dengan *Roulette Wheel* yang juga dikenal dengan metode *stochastic sampling with replacement*.

Dengan langkah-langkah yang telah dicantumkan di Bab II, proses seleksi untuk kasus di atas dapat dikerjakan sebagai berikut:

- Cari total nilai *fitness*: $0,0490 + 0,0460 + 0,0467 + 0,0502 = 0,1919$
- Cari probabilitas individu ($\frac{\text{nilai fitness individu}}{\text{total nilai fitness}}$) lalu tentukan proporsinya pada rentang 1 -100

$$P1: \frac{0,0490}{0,1919} = 0,2553 \text{ maka memiliki proporsi pada rentang } 1 - 26$$

$$P2: \frac{0,0460}{0,1919} = 0,2397 \text{ maka memiliki proporsi pada rentang } 27 - 50$$

$$P3: \frac{0,0467}{0,1919} = 0,2433 \text{ maka memiliki proporsi pada rentang } 51 - 74$$

$$P4: \frac{0,0502}{0,1919} = 0,2615 \text{ maka memiliki proporsi pada rentang } 75 - 100$$

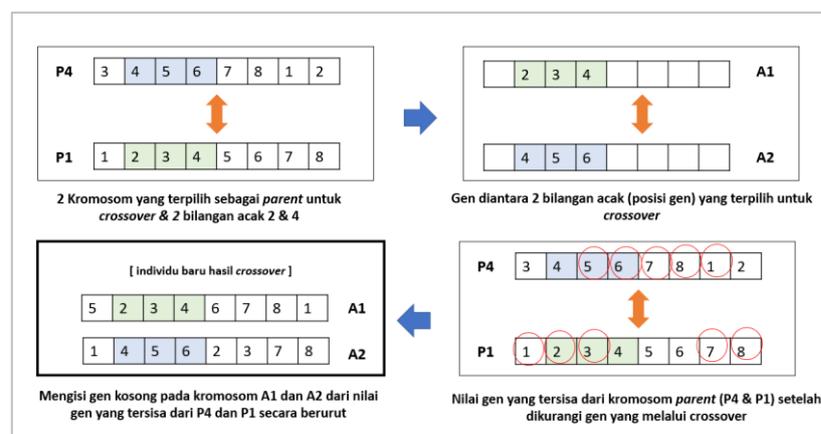
- Bangkitkan bilangan random antara 1 sampai 100. Misal diperoleh bilangan random 87 dan 19 maka akan terpilih Individu P4 dan P1 untuk melanjutkan ke tahap Algoritma Genetika selanjutnya.

3.3.5 *Crossover* (perkawinan silang) pada MTSP

Metode dan tipe *crossover* yang dilakukan tergantung pada encoding dan permasalahannya. Karena penelitian ini menggunakan *permutation encoding*, maka metode *crossover* yang akan digunakan ialah *order crossover (OX)*. Seperti langkah-langkah yang telah dicantumkan di Bab II, proses perkawinan silang untuk kasus di atas adalah sebagai berikut:

- Tentukan dua individu induk yang akan melalui *crossover*, misal P4 dan P1.
- Bangkitkan bilangan acak antara 0 dan 1
- Apabila bilangan acak tersebut kurang dari nilai probabilitas *crossover* (*crossover rate/ cr*) maka proses *crossover* dapat dilanjutkan.
- Selanjutnya bangkitkan 2 bilangan acak pada rentang 1 sampai banyaknya gen. Pada kasus sebelumnya, terdapat 8 gen dalam satu kromosom. Misal, diperoleh bilangan acak 2 dan 4.
- Buat kromosom kosong sebagai anak 1 dan anak 2
- Tempatkan gen yang berada diantara 2 bilangan acak pada induk 1 ke anak 2 dan induk 2 ke anak 1
- Sempurnakan kromosom dengan mengisi bagian kosong dengan gen yang belum termuat secara urut pada kromosom induk ke kromosom anak.

Proses di atas diilustrasikan pada Gambar 3.4.

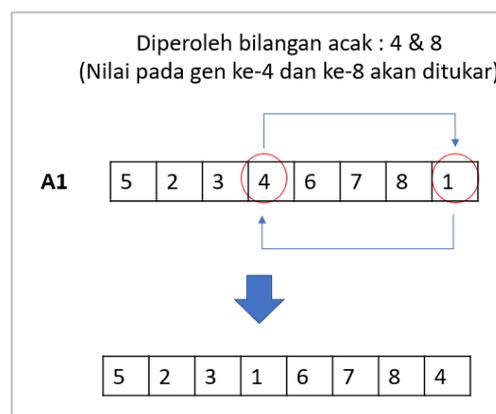


Gambar 3.5 Contoh Prosedur *Order Crossover*

3.3.6 Mutasi pada MTSP

Setelah melalui tahap *crossover*, seluruh individu baru yang dihasilkan akan melalui tahap mutasi. Tahap ini juga berguna untuk menggantikan gen yang hilang dari populasi akibat proses seleksi. Terdapat beberapa cara yang dapat digunakan untuk melakukan mutasi berdasarkan metode *encoding*-nya. Untuk *permutation encoding* metode mutasi yang akan digunakan ialah *swap mutation* atau yang juga dikenal sebagai *order-based mutation*. Berdasarkan langkah-langkah yang telah dicantumkan di Bab II, proses *swap mutation* adalah sebagai berikut:

- Tentukan individu yang akan melalui proses mutasi. Sebagai contoh, melanjutkan contoh kasus sebelumnya pada tahap mutasi akan dipilih individu A1 yang merupakan hasil dari proses *crossover*.
- Bangkitkan bilangan acak antara 0 dan 1
- Apabila bilangan acak tersebut kurang dari nilai probabilitas mutasi (*mutation rate/ mr*) maka proses mutasi dapat dilanjutkan.
- Selanjutnya bangkitkan 2 bilangan acak pada rentang 1 sampai banyaknya gen. Sebagai mana pada contoh kasus sebelumnya bahwa dalam satu kromosom individu terdapat 8 gen. Maka akan dibangkitkan bilangan acak pada rentang 1 sampai 8. Semisal didapat 2 bilangan acak 4 dan 8. Bilangan acak ini akan merepresentasikan posisi nilai gen yang akan ditukar. Dengan demikian, nilai dari gen ke-4 akan ditukar dengan nilai dari gen ke-8. Proses ini dapat digambarkan pada Gambar 3.5.



Gambar 3.6 Contoh Prosedur *Swap Mutation*

3.3.7 Evaluasi Algoritma Genetika pada MTSP

Evaluasi pada individu dilakukan untuk menentukan individu terbaik yang akan digunakan dalam menghasilkan keturunan generasi berikutnya. Iterasi generasi pada Algoritma Genetika akan terus berlangsung hingga terpenuhi kondisi berhenti (*termination condition*). Kondisi berhenti yang akan digunakan dalam penelitian ini adalah bahwa iterasi berhenti sampai generasi ke- n . Dimana nilai n ditentukan berdasarkan eksperimen pendahuluan.

Pada contoh kasus diatas, Misal kondisi berhenti ditentukan sebanyak 5 iterasi atau berjumlah 5 generasi. Setelah Algoritma Genetika dijalankan, didapat rute terbaik kasus diatas bahwa rute yang ditempuh oleh kedua petugas ialah sejauh 19,8 kilometer. Adapun rute terbaiknya tercantum pada Tabel 3.5.

Tabel 3.4 Rute Optimal pada Contoh Kasus MTSP

Petugas	Rute Perjalanan	Jarak Tempuh
A	0 – 3 – 4 – 5 – 1 – 0	9 km
B	0 – 2 – 6 – 7 – 8 – 0	10,8 km