

BAB III METODOLOGI PENELITIAN

Bab ini membahas metodologi yang digunakan dalam penyelesaian pendistribusian dan persediaan FIRP dengan menggunakan AGA. Pembahasan ini mencakup deskripsi masalah, tahapan penelitian, formulasi model optimasi, dan teknik penyelesaian.

3.1 Deskripsi Masalah

Distribusi produk dari pemasok ke pelanggan memerlukan perencanaan yang matang karena melibatkan banyak keterbatasan dan ketidakpastian. Dalam praktiknya, pemasok harus memastikan setiap pelanggan mendapatkan pasokan produk sesuai kebutuhannya, tetapi pada saat yang sama tetap memperhatikan kapasitas penyimpanan pelanggan yang terbatas, kapasitas muatan kendaraan, serta biaya operasional distribusi. Permintaan pelanggan sering kali tidak pasti karena dipengaruhi oleh faktor-faktor seperti tingkat penggunaan, perubahan musim, dan pola konsumsi. Oleh sebab itu, pemasok memerlukan metode perencanaan distribusi yang efisien agar mampu memenuhi permintaan.

Penelitian ini membahas *fuzzy inventory routing problem* (FIRP), yaitu masalah persediaan yang mempertimbangkan ketidakpastian permintaan dari pelanggan serta keterbatasan kapasitas kendaraan dan penyimpanan di lokasi pelanggan. Misalkan terdapat n pelanggan, yaitu $R = [R_1, R_2, \dots, R_n]$ dan t periode distribusi $P = \{P_1, P_2, \dots, P_n\}$, di mana pemasok bertanggung jawab untuk mendistribusikan produk kepada seluruh pelanggan selama periode perencanaan. Setiap pelanggan memiliki kapasitas penyimpanan yang terbatas dan akan mengajukan permintaan barang pada setiap periode. Untuk memodelkan ketidakpastian permintaan, penelitian ini menggunakan pendekatan bilangan *fuzzy*.

Distribusi produk dilakukan dengan menggunakan satu kendaraan yang memiliki kapasitas muatan tetap. Dalam satu periode, kendaraan diperbolehkan melakukan beberapa perjalanan, selama setiap perjalanan tidak melebihi kapasitas kendaraan.

Pemasok bertugas menentukan jumlah produk yang akan dikirim kepada setiap pelanggan serta merencanakan rute kendaraan. Pengiriman hanya dilakukan kepada pelanggan yang membutuhkan, berdasarkan permintaan aktual pada periode tersebut, dengan memperhatikan batas kapasitas penyimpanan masing-masing pelanggan. Dalam penelitian ini, pemasok hanya bertanggung jawab sampai pada pengiriman produk ke pelanggan, dan tidak bertanggung jawab atas pergerakan barang dari pelanggan ke konsumen akhir.

Tujuan utama dari penelitian ini adalah menentukan jumlah pengiriman dan rute kendaraan yang optimal di setiap periode untuk meminimalkan total biaya operasional yang meliputi biaya transportasi, biaya penyimpanan, dan biaya akibat permintaan yang tidak terpenuhi (*lost sales*). Algoritma genetika adaptif akan diimplementasikan pada penyelesaian FIRP ini dan akan diimplementasikan pada suatu agen gas LPG.

3.2 Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan, yaitu sebagai berikut.

1. Studi literatur

Pada tahapan ini, dilakukan studi pustaka dengan mempelajari teori-teori yang terkait pada penelitian ini, yang bersumber dari buku, jurnal, dan sumber lainnya. Beberapa topik yang dipelajari adalah FIRP, bilangan *fuzzy*, dan AGA. Selain itu, dilakukan pengumpulan informasi yang relevan untuk memahami lebih dalam tentang FIRP dengan permintaan tidak pasti, dan penerapan AGA.

2. Pengumpulan Data

Pada tahap ini, akan dilakukan pengumpulan data yang dibutuhkan untuk analisis. Data yang digunakan dalam penelitian ini meliputi jarak lokasi antara depot dan pelanggan, jumlah permintaan pelanggan, biaya penyimpanan dan transportasi, serta kapasitas maksimum kendaraan maupun penyimpanan pelanggan. Data-data tersebut diperoleh melalui hasil penelitian oleh Putri (2021). Perbedaan utama dengan penelitian Putri (2021) adalah pada model ini mengintegrasikan aspek inventori dengan *vehicle routing* untuk menangani pengelolaan persediaan sekaligus

penjadwalan pengiriman, sedangkan penelitian Putri (2021) hanya fokus pada VRP tanpa mempertimbangkan manajemen inventori.

3. Pembangunan Model Optimasi

Pada tahapan ini, akan dibangun model optimasi FIRP dengan terlebih dahulu mendefinisikan asumsi-asumsi.

4. Penyelesaian Masalah

Pada tahapan ini, AGA digunakan untuk menyelesaikan FIRP secara iteratif.

5. Validasi

Pada tahapan ini, dilakukan validasi untuk menguji model optimasi algoritma genetika yang sudah digunakan valid atau tidak. Validasi dilakukan dengan membandingkan solusi optimal yang diperoleh dari perhitungan secara manual dari satu kasus FIRP berukuran kecil dengan solusi optimal dari bahasa pemrograman, apabila hasilnya valid dan sesuai, maka akan dilanjut ke tahap selanjutnya. Jika hasilnya tidak valid atau solusi tidak sesuai, maka tahapan akan diulang kembali.

6. Implementasi

Setelah validasi berhasil dilakukan, selanjutnya model dan teknik penyelesaian tersebut akan diimplementasikan untuk menyelesaikan FIRP pada suatu agen gas LPG.

7. Penarikan kesimpulan

Pada tahapan ini, akan dilakukan penarikan kesimpulan berdasarkan hasil implementasi pada penelitian yang dilakukan. Hasil dari penelitian mencakup sejauh mana ketepatan rumusan masalah dan implementasi dari optimasi FIRP dengan algoritma genetika adaptif.

3.3 Model Optimasi

Asumsi yang digunakan dalam pemodelan adalah sebagai berikut.

1. Permintaan pelanggan bersifat tidak pasti dan dimodelkan sebagai variabel *fuzzy*.
2. Pemasok diasumsikan dapat memenuhi permintaan tanpa terjadi penumpukan stok di depot, karena barang yang masuk langsung didistribusikan.

3. Barang yang dikirim tidak mengalami kerusakan atau perubahan jumlah selama proses pengiriman.
4. Tersedia satu kendaraan untuk pendistribusian, dengan kapasitas muatan terbatas.
5. Setiap kendaraan selalu memulai dan mengakhiri rute di depot utama.
6. Kendaraan diperbolehkan melakukan beberapa perjalanan dalam satu periode, selama setiap perjalanan tidak melebihi kapasitas kendaraan.
7. Biaya transportasi dihitung berdasarkan jarak antar lokasi.
8. Jarak antara pelanggan dan depot sudah diketahui dan tetap.
9. Pelanggan hanya dikunjungi jika perlu pengiriman.
10. Permintaan di periode t tidak bergantung pada periode sebelumnya.
11. Waktu tempuh antar lokasi tidak mempengaruhi jadwal pengiriman.
12. Pelanggan tidak memiliki batasan waktu spesifik untuk dikunjungi.
13. Distribusi tidak mempertimbangkan permintaan yang masuk setelah keputusan pengiriman dibuat.

Adapun tahapan formulasi model optimasi ini adalah sebagai berikut.

1. Pendefinisian himpunan dan parameter

Himpunan-himpunan model optimasi terdiri dari himpunan pelanggan, himpunan periode waktu, dan himpunan jalur antar lokasi. Selain itu, terdapat parameter yang digunakan pada model optimasi meliputi biaya penyimpanan di depot dan pelanggan, biaya transportasi antar lokasi, kapasitas maksimum pelanggan dan kendaraan, jumlah persediaan di pelanggan pada tiap periode, jumlah pengiriman ke pelanggan, serta permintaan pelanggan dalam bentuk *fuzzy*.

2. Pendefinisian variabel keputusan

Variabel keputusan model didefinisikan untuk menunjukkan apakah suatu pelanggan dikunjungi oleh kendaraan pada periode tertentu, serta apakah kendaraan melakukan perjalanan dari satu lokasi ke lokasi lainnya dalam periode tersebut.

3. Pendefinisian fungsi tujuan

Fungsi tujuan dari model optimasi didefinisikan untuk meminimalkan total biaya yang meliputi biaya transportasi, biaya penyimpanan, dan biaya kekurangan

pengiriman (*lost sales*).

4. Menetapkan fungsi kendala

Kendala-kendala model optimasi menjamin bahwa:

- a) stok pelanggan diperbarui setiap periode;
- b) *lost sales* terjadi jika stok tidak mencukupi;
- c) stok tidak boleh melebihi kapasitas;
- d) pengiriman tidak melebihi kapasitas;
- e) pengiriman hanya jika pelanggan dikunjungi;
- f) hanya satu kendaraan yang tersedia;
- g) total pengiriman tidak melebihi kapasitas kendaraan;
- h) kendaraan yang mengunjungi pelanggan harus masuk dan keluar satu kali;
- i) subtour tanpa depot tidak diperbolehkan.

5. Menentukan batasan model

Batasan dari model optimasi menjamin bahwa seluruh variabel bernilai non negatif.

3.4 Teknik Penyelesaian

Penelitian ini menggunakan AGA untuk menyelesaikan *fuzzy inventory routing problem*. AGA merupakan pengembangan dari AGA yang dirancang untuk meningkatkan efektivitas pencarian solusi dalam proses optimasi. Perbedaan utama AGA dibandingkan AG standar terletak pada kemampuan penyesuaian (adaptasi) terhadap parameter utama, yaitu probabilitas *crossover* dan mutasi. Dalam AG standar, kedua nilai ini bersifat tetap selama seluruh proses evolusi. Namun, AGA mengadaptasi nilainya secara dinamis berdasarkan kondisi evolusi populasi, seperti selisih antara nilai *fitness* terbaik dan rata-rata, atau berdasarkan tingkat konvergensi solusi. Tujuannya adalah untuk menjaga keragaman populasi serta menghindari konvergensi terlalu dini.

Sebelum algoritma dijalankan, terlebih dahulu dilakukan proses *defuzzifikasi* terhadap data permintaan yang dinyatakan dalam bentuk bilangan *fuzzy* segitiga. Dalam penelitian ini, nilai b pada bilangan *fuzzy* ditentukan berdasarkan data historis yang mencerminkan pola permintaan produk. Untuk menentukan nilai b , nilai ini akan lebih condong pada nilai yang sering terjadi dalam data historis, apakah lebih mendekati nilai

minimum a atau maksimum c . Jika permintaan lebih sering rendah, b akan lebih dekat dengan a , begitu juga sebaliknya. Setelah a , b , dan c ditentukan, proses *defuzzifikasi* dilakukan untuk mengubah nilai *fuzzy* menjadi nilai *crisp* dengan rumus:

$$Crisp\ Value = \frac{a + 2b + c}{4}.$$

Proses pencarian solusi optimum menggunakan AGA merujuk pada penelitian Park, dkk. (2016) dan Cho, dkk. (2012). Proses AGA pada penelitian ini dimulai dengan membentuk populasi awal yang merepresentasikan sejumlah solusi kandidat. Masing-masing individu kemudian dievaluasi berdasarkan nilai *fitness*-nya. Variasi solusi baru dihasilkan melalui tahapan *crossover* dan mutasi, yang kemudian diperbaiki agar tetap memenuhi batasan model. Setelah itu, dilakukan proses penyempurnaan untuk meningkatkan kualitas *fitness*. Nilai parameter seperti probabilitas *crossover* dan mutasi disesuaikan secara adaptif mengikuti dinamika populasi. Seluruh proses berlangsung secara iteratif hingga tercapai kondisi penghentian dan diperoleh solusi dengan *fitness* terbaik. Langkah penyelesaian FIRP menggunakan AGA adalah sebagai berikut.

1. Inisialisasi Parameter Awal

Sebelum AGA dijalankan, beberapa parameter perlu ditentukan terlebih dahulu.

Parameter yang digunakan adalah sebagai berikut.

a. Ukuran populasi (*pop size*)

Ukuran populasi merupakan jumlah individu atau solusi yang dihasilkan dalam setiap generasi. Semakin besar ukuran populasi, maka semakin besar pula ruang solusi yang dapat dieksplorasi dalam proses pencarian solusi optimal. Namun, ukuran populasi juga perlu disesuaikan agar tidak memperlambat waktu komputasi secara signifikan.

b. Iterasi maksimum (P_{maks})

Iterasi maksimum adalah jumlah generasi terbanyak yang diperbolehkan dalam algoritma. Nilai ini menjadi kriteria penghentian algoritma jika solusi terbaik belum ditemukan. Pada setiap iterasi, akan dilakukan seleksi, *crossover*, mutasi,

dan evaluasi *fitness*.

c. *Crossover rate* adaptif (ρ_c)

Crossover rate adaptif merupakan probabilitas sebuah kromosom untuk menjalani operasi *crossover* yang awalnya ditetapkan dengan nilai tetap. Dalam penelitian ini, nilai awal *crossover rate* ditetapkan sebesar 0,8 sebagai titik awal yang digunakan di seluruh populasi. Nilai ini menjadi basis sebelum dilakukan penyesuaian dinamis selama proses evolusi.

d. Probabilitas pertukaran baris (p_{row_c})

Probabilitas ini mengatur peluang terjadinya pertukaran baris antar kromosom pada bagian matriks pengisian (*replenishment matrix*). Setiap baris (pelanggan) memiliki peluang untuk dipertukarkan antara dua induk. Nilai probabilitas ini mengikuti nilai *crossover rate* adaptif setiap generasi. Semakin besar nilai ρ_c , semakin besar kemungkinan baris-baris kromosom ditukar, sehingga mendorong variasi pada strategi pengisian.

e. Probabilitas *crossover* rute (P_{route_c})

Probabilitas *crossover* rute mengacu pada kemungkinan pertukaran bagian urutan kunjungan pelanggan antar dua induk. probabilitas ini juga mengikuti nilai *crossover rate* adaptif (ρ_c).

f. *Mutation rate* adaptif (ρ_m)

Mutation rate adaptif merupakan probabilitas terjadinya mutasi pada kromosom yang awalnya ditetapkan dengan nilai tetap. Pada penelitian ini, nilai awal mutasi *rate* ditetapkan sebesar 0,1 sebagai nilai dasar yang digunakan pada seluruh populasi sebelum dilakukan penyesuaian secara adaptif. Nilai ini menjadi basis sebelum dilakukan penyesuaian dinamis selama proses evolusi.

2. Representasi Kromosom

Kromosom terdiri dari dua bagian utama, yaitu matriks *replenishment* dan rute kendaraan:

a. Kromosom matriks pengisian (*Replenishment matrix*)

Bagian ini menyimpan informasi tentang jumlah pengisian ulang persediaan

untuk setiap pelanggan dalam setiap periode. Setiap elemen dalam matriks ini menunjukkan berapa banyak produk yang dikirim ke pelanggan tertentu pada periode tertentu. Misalnya, elemen $q_{i,t}$ merepresentasikan jumlah barang yang dikirim ke pelanggan i pada periode t . Contoh representasi matriks pengisian dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Representasi Kromosom Matriks Pengisian

Pelanggan (i)	Periode 1	Periode 2	Periode 3	Periode 4
Pelanggan 1	q_{11}	q_{12}	q_{13}	q_{14}
Pelanggan 2	q_{21}	q_{22}	q_{23}	q_{24}

b. Kromosom rute kendaraan

Bagian ini direpresentasikan dalam bentuk urutan permutasi pelanggan yang akan dikunjungi oleh kendaraan dalam satu periode. Depot tidak dicantumkan secara eksplisit dalam kromosom karena dianggap sebagai titik awal dan akhir setiap perjalanan.

Karena kapasitas kendaraan terbatas, satu rute dalam satu periode dapat terdiri dari beberapa trip. Untuk menjadikan kromosom valid, dilakukan pembagian trip berdasarkan batas kapasitas kendaraan. Mekanisme pembacaan dan pembagian trip dilakukan dengan langkah sebagai berikut:

- 1) Proses pembacaan dimulai dari depot, kemudian mengunjungi pelanggan sesuai urutan dalam kromosom.
- 2) Selama jumlah permintaan kumulatif pelanggan yang sudah dikunjungi masih di bawah kapasitas maksimum kendaraan, pelanggan berikutnya dapat ditambahkan ke trip yang sama.
- 3) Jika penambahan pelanggan berikutnya menyebabkan total permintaan melebihi kapasitas, trip tersebut ditutup, kendaraan kembali ke depot, dan perjalanan baru dimulai untuk pelanggan berikutnya.

Proses ini diulangi hingga seluruh pelanggan dalam satu periode terlayani. Contoh representasi kromosom rute kendaraan dapat dilihat pada Tabel 3.2, di mana trip 1 melayani pelanggan 4, 8, dan 1; trip 2 melayani pelanggan 7 dan 5; serta trip 3

melayani pelanggan 3, 2, dan 6. Pembagian trip ini memastikan setiap perjalanan kendaraan tidak melebihi kapasitas maksimum yang diizinkan, sehingga solusi yang dihasilkan tetap memenuhi batasan yang ditetapkan.

Tabel 3. 2 Representasi Encoding Kromosom Rute Kendaraan

Periode	Kromosom (Encoding)							
P1	4	8	1	7	5	3	2	6
	Trip 1			Trip 2		Trip 3		

3. Inisialisasi Populasi

Pada tahap ini, dibentuk sejumlah solusi awal yang merepresentasikan skenario distribusi produk ke pelanggan. Setiap solusi diwakili oleh kromosom yang terdiri atas matriks pengisian (*replenishment matrix*), merepresentasikan jumlah pengisian ulang persediaan ke setiap pelanggan pada masing-masing periode. Nilai pengisian ini didasarkan pada permintaan pelanggan yang telah melalui proses *defuzzifikasi*, dengan tetap memenuhi dua syarat utama: (1) total pengiriman tidak melebihi kapasitas kendaraan, dan (2) jumlah persediaan tidak melebihi kapasitas maksimum penyimpanan pelanggan.

Untuk bagian rute kendaraan, digunakan metode *r-savings* dalam proses konstruksi awal. Metode ini bertujuan menggabungkan pelanggan-pelanggan yang menghasilkan penghematan jarak (*saving*) terbesar, berdasarkan prinsip bahwa penggabungan rute antara dua pelanggan lebih hemat dibanding pelayanan terpisah. Proses penggabungan dilakukan secara bertahap, dengan tetap memerhatikan batasan kapasitas kendaraan, hingga tidak ada lagi pasangan pelanggan yang dapat digabungkan tanpa melanggar batasan, pendekatan ini memungkinkan terbentuknya populasi awal yang efisien dan layak, sebagai titik awal sebelum dilakukan proses evolusi dan pengoptimalan lebih lanjut menggunakan AGA. Rumus *r-savings* adalah sebagai berikut:

$$S_{ij} = c_{i0} + c_{0j} - c_{ij}$$

dengan s_{ij} adalah nilai *saving*, c_{ij} adalah jarak dari depot ke pelanggan i , c_{0j} adalah jarak dari depot ke pelanggan j , dan c_{ij} adalah jarak antara pelanggan i ke j .

4. Evaluasi *Fitness*

Dalam FIRP, fungsi *fitness* digunakan untuk mengukur kualitas solusi biaya kromosom berdasarkan total biaya distribusi. Nilai *fitness* yang lebih besar menunjukkan solusi yang lebih efisien. Komponen biaya dalam fungsi *fitness* terdiri dari tiga bagian utama:

- a. Biaya transportasi ($C_{transport}(X)$), biaya ini dihitung dari total jarak tempuh kendaraan dikalikan dengan biaya per kilometer, dan disesuaikan dengan faktor periode λ_t . Nilai λ_t bervariasi berdasarkan kondisi hari di mana hari diskon $\lambda_t = 0,8$, hari normal $\lambda_t = 1,0$, dan hari sibuk $\lambda_t = 1,2$.
- b. Biaya inventori ($C_{inventory}(X)$), merupakan akumulasi biaya penyimpanan produk di setiap pelanggan. Perhitungannya didasarkan pada jumlah stok awal, jumlah produk yang dikirim (didasarkan pada kromosom), serta permintaan pelanggan (hasil *defuzzifikasi* permintaan *fuzzy*), dengan rumus:

$$HC_{it} = h_i \times (I_{i,t-1} + q_{it} - d_{it}).$$

- c. Biaya penalti *lost sales*, biaya ini memperkirakan potensi kerugian akibat ketidakmampuan memenuhi permintaan pelanggan. Penambahan komponen ini dalam fungsi biaya bertujuan untuk meminimalkan ketidakterpenuhan permintaan pelanggan, serta mendorong solusi yang lebih responsif terhadap kebutuhan pelanggan pada tiap periode. *Lost sales* terjadi ketika:

$$LS_{it} = \max(D_i^* - (I_{i,t-1} + q_{it}), 0).$$

Jika solusi melanggar kapasitas kendaraan, maka tidak langsung diberikan penalti besar. Sebaliknya, penalti tersebut akan diperbaiki melalui mekanisme *chromosome repair*, seperti membagi kembali pelanggan ke dalam beberapa rute berbeda agar tidak melebihi kapasitas kendaraan, namun tetap menjaga agar setiap pelanggan hanya dikunjungi satu kali dalam satu periode, sehingga rumus untuk meminimalkan total biaya ditulis sebagai:

$$Fitness(X) = C_{transport}(X) + C_{inventory}(X) + Penalty_{LostSales}(X).$$

Dengan demikian, fungsi *fitness* ditulis sebagai:

$$Fitness(X) = \frac{1}{Total\ biaya(X)}.$$

5. Seleksi

Proses seleksi dilakukan untuk memilih kromosom (solusi) terbaik yang akan digunakan dalam generasi berikutnya. Metode seleksi yang digunakan adalah *Roulette-Wheel Selection*, di mana semakin kecil total biaya distribusi suatu solusi, semakin tinggi peluangnya untuk terpilih. Langkah-langkah dari metode seleksi *roulette-wheel* adalah sebagai berikut:

- a. Hitung nilai *fitness* dari setiap kromosom X_k dalam populasi.
- b. Hitung *fitness* relatif f'_k agar dapat digunakan dalam seleksi *roulette-wheel*, dengan menggunakan rumus:

$$f'_k = \frac{f_k - f_{min}}{f_{max} - f_{min}}$$

di mana f_k adalah nilai *fitness* dari kromosom ke- k , dan f_{min} , f_{max} masing-masing adalah *fitness* minimum dan maksimum dalam populasi. Normalisasi ini bertujuan untuk memetakan nilai *fitness* ke rentang 0 – 1, memperjelas perbedaan relatif antar kromosom, dan mencegah dominasi berlebihan dari kromosom dengan *fitness* tertinggi sehingga tekanan seleksi lebih terkontrol.

- c. Hitung probabilitas seleksi p_k untuk setiap kromosom berdasarkan kontribusi *fitness* relatif terhadap total:

$$p_k = \frac{f'_k}{\sum_{i=1}^N f'_i}$$

Semakin tinggi nilai *fitness* relatif, semakin besar peluang kromosom tersebut untuk dipilih.

- d. Bentuk *roulette-wheel* berdasarkan probabilitas kumulatif dari setiap kromosom. Setiap kromosom memiliki rentang peluang pada skala 0 sampai 1, sesuai nilai p_k -nya.
 - e. Lakukan seleksi dengan menghasilkan bilangan acak $r \in [0, 1]$, kromosom terpilih adalah yang rentang kumulatifnya memuat angka r .
6. Crossover

Crossover digunakan untuk menghasilkan kromosom baru (*offspring*) dengan menggabungkan sebagian karakteristik dari dua individu induk. Dalam penelitian ini, setiap kromosom terdiri atas dua bagian utama, yaitu matriks pengisian dan rute kendaraan.

Proses *crossover* dilakukan secara adaptif mengikuti nilai *crossover rate* adaptif yang telah dijelaskan pada bagian inisialisasi parameter awal. Nilai *crossover rate* ini akan menyesuaikan secara otomatis di setiap generasi berdasarkan kinerja solusi anak dibandingkan dengan induknya. Tiga bentuk probabilitas *crossover* yang digunakan dalam algoritma ini adalah:

- a. *Crossover rate* adaptif, yaitu probabilitas dasar terjadinya *crossover* antar dua induk, yang disesuaikan secara dinamis selama evolusi.
- b. Probabilitas pertukaran baris, yang digunakan dalam *row-based uniform crossover* pada matriks pengisian, dengan menukar sebagian baris antar-induk secara acak.
- c. Probabilitas *crossover* rute, yang diterapkan pada kromosom rute kendaraan dengan metode *one-point crossover*, yaitu pertukaran sebagian urutan kunjungan pelanggan tanpa duplikasi.

Langkah-langkah metode *row-based uniform crossover* matriks pengisian pada AGA adalah sebagai berikut:

- a. Pemilihan induk

Dua induk dipilih dari populasi berdasarkan hasil seleksi. Masing-masing kromosom direpresentasikan dalam bentuk matriks pengisian berukuran $n \times T$ (pelanggan \times periode), di mana setiap sel berisi jumlah pengisian ulang

persediaan ke pelanggan tertentu pada periode tertentu. Contoh representasi kromosom induk dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Representasi Kromosom Induk Matriks Pengisian Sebelum *Crossover*

Pelanggan	P1	P2	P3	Pelanggan	P1	P2	P3
R1	1	2	3	R1	5	4	3
R2	5	3	2	R2	1	3	5
R3	4	5	1	R3	3	2	1
R4	5	1	3	R4	1	1	2
R5	4	4	2	R5	4	2	3
Induk 1				Induk 2			

b. Penentuan terjadinya *crossover*

Pada setiap pasangan induk, dibangkitkan bilangan acak $r \in [0, 1]$ untuk menentukan apakah *crossover* dilakukan. Proses ini dikontrol oleh probabilitas *crossover* adaptif (ρ_c), di mana pada generasi pertama biasanya dimulai dari nilai batas bawah. Lalu pada generasi berikutnya, nilainya diperbarui otomatis mengikuti skema adaptif, dengan nilai ρ_c selalu dijaga pada rentang yang ditentukan. Jika $r < \rho_c$, maka *crossover* dilakukan pada kedua induk. Jika tidak, *offspring* yang dihasilkan adalah salinan langsung dari induk.

c. Pertukaran baris (*row swapping*)

Jika *crossover* terjadi, untuk setiap baris (pelanggan ke- i) pada matriks pengisian, kembali diacak peluang pertukaran baris antar-induk menggunakan bilangan acak yang mengacu ke ρ_c pada generasi tersebut. Jika $r < P_{row,c}$, seluruh isi baris ke- i untuk periode tersebut dipertukarkan antara induk 1 dan induk 2. Jika tidak, baris pada *offspring* tetap sama dengan induk semula. Proses ini dilanjutkan untuk setiap baris (pelanggan), sehingga tercipta kombinasi baru jumlah pengisian ulang persediaan dalam rentang periode tertentu. Contoh penentuan baris untuk ditukar

dapat dilihat pada Tabel 3.4.

Tabel 3. 4 Baris R1 dari Induk 1 dan Induk 2 yang akan Ditukar

Pelanggan	P1	P2	P3	Pelanggan	P1	P2	P3
R1	1	2	3	R1	5	4	3
Induk 1				Induk 2			

d. Pembentukan *offspring*

Setelah semua baris pada kedua induk diproses, terbentuk dua *offspring* yang mengandung kombinasi rencana pengisian ulang setiap pelanggan dalam berbagai periode. Contoh hasil *crossover* dapat dilihat pada Tabel 3.5 sebagai representasi *offspring* hasil pertukaran beberapa baris antar-induk.

Tabel 3. 5 Contoh Pembentukan *Offspring* bagian Matriks Pengisian pada Proses *Row-based Uniform Crossover*.

Pelanggan	P1	P2	P3	Pelanggan	P1	P2	P3
R1	5	4	3	R1	1	2	3
R2	5	3	2	R2	1	3	5
R3	4	5	1	R3	3	2	1
R4	5	1	3	R4	1	1	2
R5	4	4	2	R5	4	2	3
<i>Offspring 1</i>				<i>Offspring 2</i>			

e. Perbaiki solusi (*repair solution*)

Sebagai langkah akhir, solusi *offspring* akan diperiksa untuk memastikan seluruh batasan terpenuhi, yaitu batasan kapasitas kendaraan. Jika hasil *crossover* menghasilkan solusi yang melampaui kapasitas kendaraan, maka dilakukan perbaikan menggunakan *vehicle capacity fixation* (VCF). Perbaikan dilakukan dengan mengidentifikasi pelanggan pada rute dengan total muatan terkecil, lalu melakukan redistribusi pengiriman ke periode lain—baik ke periode yang memiliki *lost sales* ataupun periode sebelumnya yang masih memiliki sisa kapasitas kendaraan. Jika redistribusi lintas periode tersebut tidak memungkinkan, maka pengurangan atau penghapusan sebagian pengiriman dilakukan, sehingga

pengiriman pada rute tersebut kembali berada dalam batasan kapasitas maksimal yang diperbolehkan.

Langkah-langkah metode *One-point crossover* rute kendaraan pada AGA adalah sebagai berikut:

a. Pemilihan induk

Dua induk dipilih dari populasi berdasarkan hasil seleksi. Masing-masing kromosom rute kendaraan memuat informasi urutan kunjungan pelanggan oleh setiap kendaraan pada setiap periode perencanaan. Representasi kromosom induk dapat dilihat pada Tabel 3.6.

Tabel 3. 6 Representasi Kromosom Induk Rute Kendaraan sebelum *Crossover*

Periode	Trip 1	Trip 2	Periode	Trip 1	Trip 2
P1	[1, 3, 5]	[2, 4]	P1	[3, 2, 1]	[4, 5]
P2	[3, 1, 2]	[4, 5]	P2	[5, 1, 4]	[2, 3]
P3	[5, 1]	[2, 4, 3]	P3	[1, 2, 3]	[5, 4]
Induk 1			Induk 2		

b. Penentuan terjadinya *crossover*

Pada setiap pasangan induk, Proses ini dikontrol oleh probabilitas *crossover* adaptif (ρ_c) dengan ρ_c tetap berada dalam interval yang ditentukan. Di mana pada generasi pertama biasanya dimulai dari nilai batas bawah. Jika $r < \rho_c$, maka *crossover* dilakukan pada kedua induk. Jika tidak, *offspring* yang dihasilkan adalah salinan langsung dari induk.

c. Pemilihan titik *crossover*

Jika *crossover* diputuskan akan dilakukan, maka satu titik potong dipilih secara acak pada kromosom rute kendaraan. Titik *crossover* ini menentukan di mana pemisahan gen antar induk dilakukan untuk selanjutnya digabungkan menjadi kromosom baru (*offspring*). Contoh pemilihan titik *crossover* dapat dilihat pada

Tabel 3.7.

Tabel 3. 7 Contoh Pemilihan Titik *Crossover* pada Kromosom Rute Kendaraan

Periode	Trip 1	Trip 2	Periode	Trip 1	Trip 2
P1	[1, 3, 5]	[2, 4]	P1	[3, 2, 1]	[4, 5]
P2	[3, 1, 2]	[4, 5]	P2	[5, 1, 4]	[2, 3]
P3	[5, 1]	[2, 4, 3]	P3	[1, 2, 3]	[5, 4]
Induk 1			Induk 2		

d. Pembentukan *offspring*

Setelah titik *crossover* dipilih, pembentukan *offspring* dilakukan dengan mengambil bagian awal kromosom dari induk pertama hingga titik potong, dengan melengkapi bagian sisanya dari induk kedua, sambil memastikan urutan pelanggan tetap valid tanpa duplikasi. Misalnya, titik potong adalah setelah periode P1. Artinya, *offspring* pertama mengambil periode 1 dari induk utama, sedangkan sisanya dari induk kedua, begitu pula sebaliknya. Proses ini menghasilkan dua kromosom anak dengan kombinasi urutan kunjungan yang baru dan valid, seperti ditunjukkan pada Tabel 3.8.

Tabel 3. 8 Contoh Pembentukan *Offspring* bagian Matriks Rute Kendaraan pada Proses *One-point Crossover*

Periode	Trip 1	Trip 2	Periode	Trip 1	Trip 2
P1	[1, 3, 5]	[2, 4]	P1	[3, 2, 1]	[4, 5]
P2	[5, 1, 4]	[2, 3]	P2	[3, 1, 2]	[4, 5]
P3	[1, 2, 3]	[5, 4]	P3	[5, 1]	[2, 4, 3]
<i>Offspring 1</i>			<i>Offspring 2</i>		

e. Perbaiki solusi (*repair solution*)

Setelah *offspring* terbentuk, dilakukan pemeriksaan apakah solusi melanggar batasan, yaitu kapasitas kendaraan. Bila terjadi pelanggaran karena kelebihan muatan pada suatu periode, dilakukan perbaikan dengan menyesuaikan jumlah pengiriman per pelanggan dalam periode tersebut, penyesuaian ini dapat berupa redistribusi pengiriman ke periode lain yang memiliki kapasitas tersisa ataupun

pengurangan jumlah pengiriman agar total muatan pada satu trip tidak melebihi kapasitas maksimum kendaraan. Dengan demikian, setiap *offspring* yang dihasilkan tetap memenuhi semua batasan dan layak digunakan pada tahap evolusi selanjutnya.

7. Mutasi

Setelah proses *crossover*, tahap berikutnya adalah melakukan mutasi pada kromosom untuk memperkenalkan variasi baru dalam solusi dan mencegah konvergensi terhadap solusi lokal. Proses mutasi dilakukan secara adaptif mengikuti nilai mutasi *rate* adaptif, yaitu probabilitas mutasi yang nilainya disesuaikan secara otomatis pada setiap generasi. Nilai ρ_m tidak bersifat tetap, melainkan berubah sesuai dengan kinerja solusi. Jika mutasi menghasilkan solusi yang lebih baik, maka nilai ρ_m cenderung meningkat, sedangkan jika hasilnya kurang baik, maka ρ_m akan diturunkan. Nilai ρ_m ini dijaga agar tetap berada dalam rentang tertentu, sehingga proses mutasi dapat menyesuaikan diri secara dinamis selama proses evolusi berlangsung. jenis mutasi yang digunakan pada penelitian ini adalah *swap* mutation.

Langkah-langkah *swap mutation* pada kromosom matriks pengisian dan kromosom rute kendaraan adalah sebagai berikut:

a. Pemilihan *offspring* untuk mutasi

Dipilih *offspring* hasil *crossover* yang akan mengalami mutasi. Sebuah bilangan acak $r \in [0, 1]$ dibangkitkan. Jika nilainya kurang dari ρ_m , maka kromosom *offspring* tersebut akan dimutasi. Contoh kromosom yang akan dimutasi pada kromosom matriks pengisian dan kromosom rute kendaraan dapat dilihat pada Tabel 3.9 dan Tabel 3.10.

Tabel 3. 9 Contoh Kromosom Induk Matriks Pengisian pada Proses Mutasi

Pelanggan	Periode 1	Periode 2	Periode 3
R1	5	4	3
R2	5	3	2
R3	4	5	1

R4	5	1	3
R5	4	4	2

Tabel 3. 10 Contoh Kromosom Induk Rute Kendaraan pada Proses Mutasi

Periode	Trip 1	Trip 2
P1	[1, 3, 5]	[2, 4]
P2	[3, 1, 2]	[4, 5]
P3	[5, 1]	[2, 4, 3]

b. Pemilihan periode dan pelanggan yang ditukar

Mutasi dilakukan dengan memilih satu periode secara acak, kemudian memilih dua pelanggan dalam periode tersebut untuk dipertukarkan. Pada matriks pengisian, pertukaran berarti menukar nilai pengisian antara dua pelanggan di periode tersebut. Pada kromosom rute kendaraan, pertukaran berarti menukar urutan kunjungan kedua pelanggan tersebut dalam rute periode yang sama. Contoh titik acak pada kromosom matriks pengisian dapat dilihat pada Tabel 3.11.

Tabel 3. 11 Contoh Titik Acak pada Kromosom Matriks Pengisian

Pelanggan	Periode 1	Periode 2	Periode 3
R1	5	4	3
R2	5	3	2
R3	4	5	1
R4	5	1	3
R4	4	4	2

c. Pertukaran (*swap*)

Dilakukan pertukaran posisi antara dua pelanggan yang terpilih. Misalnya, pada periode P2, pelanggan R2 dan R3 dipilih untuk ditukar; nilai pengisian pada matriks pengisian dan posisi pada rute kendaraan ditukar antara R2 dan R3. Contoh hasil mutasi pada matriks pengisian dapat dilihat pada Tabel 3.12 dan hasil mutasi pada rute pengiriman ditunjukkan pada Tabel 3.13.

Tabel 3. 12 Contoh *Offspring* Kromosom Matriks Pengisian setelah Proses Mutasi

Pelanggan	Periode 1	Periode 2	Periode 3
R1	5	4	3
R2	5	5	2
R3	4	3	1
R4	5	1	3
R5	4	4	2

Tabel 3. 13 Contoh *Offspring* Kromosom Rute Kendaraan setelah Proses Mutasi

Periode	Trip 1	Trip 2
P1	[1, 3, 5]	[2, 4]
P2	[2, 1, 3]	[4, 5]
P3	[5, 1]	[2, 4, 3]

d. Pemeriksaan kelayakan kromosom hasil mutasi

Setelah pertukaran dilakukan, hasil mutasi dicek untuk memastikan seluruh batasan tetap terpenuhi, seperti batasan kapasitas kendaraan, kapasitas penyimpanan pelanggan, serta konsistensi antara rencana pengiriman dan rute kendaraan. Jika solusi valid, *offspring* disimpan dalam populasi baru untuk generasi selanjutnya

8. Penyempurnaan Solusi (*Solution Improvement*)

Setelah semua proses evolusi selesai, dilakukan tahap penyempurnaan solusi (*solution improvement*) untuk menyempurnakan solusi yang dihasilkan. Tahapan ini bertujuan mengurangi biaya total (transportasi, penyimpanan, dan *lost sales*) dengan konsolidasi pengiriman antar-periode yang memungkinkan, tanpa melanggar kapasitas kendaraan dan kapasitas maksimum gudang pelanggan. Berikut adalah langkah-langkah perbaikannya:

a. Konsolidasi pengiriman ke periode sebelumnya

Untuk setiap pelanggan pada setiap periode t , dilakukan pemeriksaan apakah sebagian atau seluruh jumlah pengiriman dapat dipindahkan ke periode sebelumnya ($t - 1$). Pemindahan ini hanya dilakukan jika total pengiriman di

periode sebelumnya tidak melebihi kapasitas maksimum pelanggan, serta stok setelah penambahan tidak melampaui batas penyimpanan. Selama syarat permintaan minimum hingga maksimum terpenuhi, pengiriman bisa digabung ke periode sebelumnya sehingga jadwal rute kendaraan dapat diperbarui.

b. Konsolidasi trip/rute dalam satu periode

Jika total muatan dalam beberapa trip pada periode yang sama masih berada di bawah kapasitas kendaraan, maka rute-rute tersebut dapat digabungkan menjadi satu perjalanan. Langkah ini bertujuan untuk menekan biaya transportasi dengan mengurangi jumlah perjalanan

c. Revisi rute kendaraan

Setelah konsolidasi, rute kendaraan diperbarui kembali. Pelanggan yang sudah tidak menerima pengiriman pada suatu periode dihapus dari rute pada periode tersebut. Apabila di satu periode tidak ada lagi pelanggan yang akan dikunjungi, trip pada periode tersebut dihapus untuk menghindari biaya transportasi yang tidak perlu.

d. Pemeriksaan kelayakan solusi

Setelah konsolidasi selesai, dilakukan pemeriksaan kembali terhadap seluruh batasan, baik kapasitas kendaraan, kapasitas penyimpanan, maupun aturan *fuzzy* pengiriman. Jika ditemukan pelanggaran, maka konsolidasi dibatalkan atau diatur ulang ke periode lain yang masih memungkinkan, sehingga solusi tetap layak dan seluruh batasan dapat terjaga.

Setelah proses penyempurnaan solusi, dilakukan perbandingan antara total biaya solusi hasil penyempurnaan dengan biaya solusi *offspring* hasil mutasi yang sudah melalui perbaikan. Jika solusi hasil penyempurnaan memiliki total biaya lebih rendah (lebih baik), maka solusi tersebut yang dipilih untuk iterasi berikutnya. Sebaliknya, jika tidak, solusi *offspring* hasil mutasi tetap dipertahankan sebagai kandidat populasi selanjutnya.

9. Penyesuaian Parameter Adaptif

Pada proses evolusi AGA, probabilitas *crossover* dan mutasi tidak digunakan secara tetap, melainkan disesuaikan secara dinamis di setiap generasi sesuai performa solusi yang dihasilkan. Penyesuaian ini bertujuan menjaga keragaman populasi dan mencegah algoritma terjebak pada solusi lokal.

Crossover rate adaptif diperbarui dengan membandingkan kualitas (*fitness*) *offspring* terhadap induknya:

- a. Jika *offspring* memiliki *fitness* lebih baik daripada induk, *crossover rate* dinaikkan secara bertahap, yaitu dengan:

$$\rho_{c_{t+1}} = \min(\rho_{c_t} + 0,05; 0,8).$$

- b. Jika *fitness offspring* lebih buruk, maka:

$$\rho_{c_{t+1}} = \max(\rho_{c_t} - 0,05; 0,5).$$

Nilai *crossover rate* selalu dijaga dalam rentang $[0,5 - 0,8]$ agar tetap seimbang antara eksplorasi dan eksploitasi.

Mutasi *rate* adaptif juga diatur dengan pola serupa:

- a. Jika *offspring* memiliki *fitness* lebih baik daripada induk, mutasi *rate* dinaikkan secara bertahap, yaitu dengan:

$$\rho_{m_{t+1}} = \min(\rho_{m_t} + 0,005; 0,8).$$

- b. Jika *fitness offspring* lebih buruk, maka:

$$\rho_{m_{t+1}} = \max(\rho_{m_t} - 0,005; 0,5).$$

Nilai mutasi *rate* tetap berada dalam rentang $[0,1 - 0,3]$ agar tetap efektif dalam menghasilkan variasi solusi.

10. Pembaruan Populasi (*Population Replacement*)

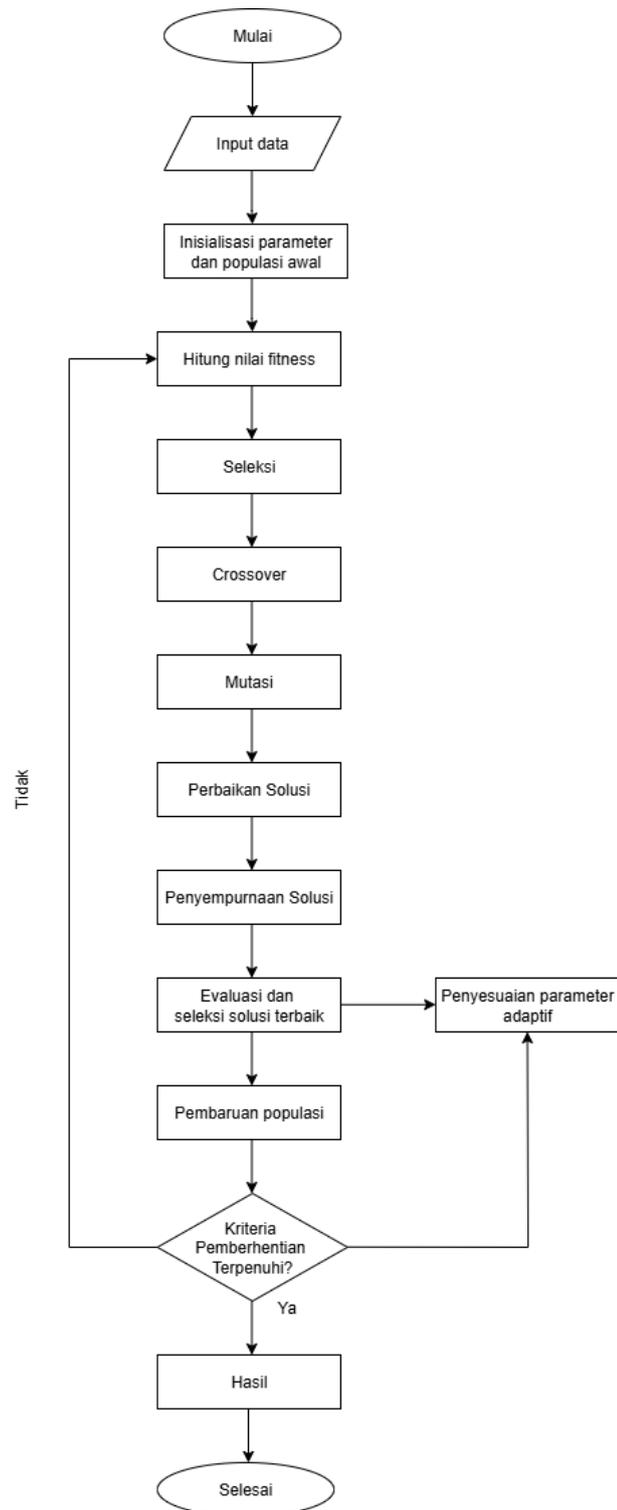
Pembaruan populasi dilakukan setelah seluruh proses evolusi dalam satu generasi selesai, yaitu setelah seleksi, *crossover*, mutasi, penyempurnaan solusi, dan penyesuaian parameter adaptif. Pada tahap ini, solusi-solusi baru (*offspring*) yang telah diperbaiki dan disempurnakan dipilih untuk menggantikan individu yang kurang baik di populasi, dengan tetap menjaga ukuran populasi tetap sama. Selain

itu, mekanisme elitisme diterapkan dengan mempertahankan solusi terbaik dari generasi sebelumnya agar tidak tergantikan oleh solusi yang lebih buruk. Proses ini menjamin bahwa populasi generasi berikutnya berisi solusi-solusi yang lebih baik dan layak untuk evolusi selanjutnya, sehingga arah pencarian solusi tetap optimal.

11. Kriteria Penghentian Algoritma

Pencarian dengan algoritma genetika adaptif akan dihentikan jika jumlah generasi yang ditentukan sudah tercapai dan solusi terbaik dipilih. Selain itu, jika dalam beberapa generasi terakhir tidak ada peningkatan signifikan pada nilai *fitness*, maka solusi sudah cukup optimal dan tidak perlu iterasi lebih lanjut.

Untuk memberikan gambaran terkait langkah-langkah AGA, berikut disajikan *flowchart* yang dapat dilihat pada Gambar 3.1.



Gambar 3. 1 *Flowchart* AGA