

BAB III

METODE PENELITIAN

Bab ini membahas metodologi yang digunakan untuk menyelesaikan masalah DVRP dengan waktu pelayanan tak pasti yang diawali dengan mendeskripsikan masalah, menjelaskan tahapan penelitian, membangun model matematika, dan penyelesaiannya dengan *fuzzy*-ACS.

3.1 Deskripsi Masalah

Penelitian ini membahas DVRP, yaitu masalah penentuan rute sejumlah kendaraan dari depot ke sejumlah pelanggan. Setiap kendaraan mempunyai jam kerja yang terbatas. Pelanggan terbagi menjadi dua kategori, yaitu pelanggan awal dan pelanggan tambahan. Pelanggan awal adalah pelanggan yang sudah diketahui sebelum kendaraan mulai bekerja, sedangkan pelanggan tambahan adalah pelanggan baru yang muncul selama kendaraan sedang bekerja. Lamanya waktu yang diperlukan untuk melayani pelanggan tidak diketahui di awal dan bersifat tidak pasti. Pada tahap awal, sistem harus membuat rute optimal untuk mengunjungi semua pelanggan awal. Selama jam kerja, pelanggan tambahan mungkin muncul. Sistem harus memutuskan apakah pelanggan tambahan tersebut akan dilayani atau tidak. Keputusan ini dibuat dengan mempertimbangkan estimasi total waktu pelayanan yang dibutuhkan dan sisa waktu kerja yang tersedia. Setelah sebuah kendaraan memutuskan untuk melayani pelanggan, maka tugas tersebut tidak dapat dibatalkan. Selanjutnya, rute yang sedang berjalan diperbarui berdasarkan keputusan tersebut. Karena pelanggan tambahan mungkin muncul selama jam kerja, kendaraan yang telah menyelesaikan tugasnya sebelum jam kerja berakhir harus menunggu di lokasi pelanggan terakhir hingga tugasnya diperbarui atau jam kerja selesai. Sistem memperbarui rute yang sedang berjalan secara berkala dalam interval waktu tertentu, dengan memperlakukan seluruh pelanggan yang sudah terdaftar tetapi belum dilayani sebagai masalah VRP statis. Dalam penelitian ini, masalah DVRP akan diselesaikan dengan menggunakan algoritma *fuzzy*-ACS, yang

dirancang untuk menangani ketidakpastian dalam waktu pelayanan dan dinamika rute secara efektif.

3.2 Tahapan Penelitian

Penelitian ini dilakukan melalui tahapan berikut.

1. Studi Pustaka

Pada tahap ini, studi pustaka dilakukan dengan mempelajari teori-teori yang relevan dengan penelitian ini dari buku, jurnal, maupun sumber lainnya. Beberapa konsep yang dipelajari dalam penelitian ini meliputi DVRP, bilangan *fuzzy*, dan ACS. Kemudian, dilakukan pengumpulan informasi yang relevan serta sumber kajian ilmiah untuk memperdalam pemahaman mengenai DVRP dengan waktu pelayanan tak pasti dan *fuzzy*-ACS.

2. Pengumpulan Data

Pada tahap ini, akan dilakukan pengumpulan data yang dibutuhkan untuk analisis. Data-data yang dibutuhkan dalam penelitian ini meliputi lokasi pelanggan dan depot, titik koordinat tiap pelanggan dan depot, jumlah kendaraan yang tersedia, jam operasional dan batas waktu kerja, waktu temu pelanggan, serta data historis waktu layanan tiap pelanggan.

3. Formulasi Model Optimasi

Pada tahapan ini, DVRP akan diformulasikan sebagai model optimasi dengan waktu pelayanan yang tidak pasti menggunakan pendekatan *fuzzy* VRP.

4. Penyelesaian Model Optimasi

Pada tahapan ini, algoritma *fuzzy*-ACS digunakan untuk menyelesaikan DVRP secara iteratif.

5. Validasi

Pada tahapan ini, model optimisasi dan teknik penyelesaiannya akan divalidasi dengan membandingkan solusi optimal yang dihasilkan dari perhitungan manual dengan solusi hasil dari program komputer. Jika hasil dari kedua pendekatan menunjukkan konsistensi atau perbedaan yang kecil dan dalam batas toleransi, maka penelitian dilanjutkan ke tahap berikutnya. Namun, jika ditemukan perbedaan yang signifikan, model akan diperbaiki dan diuji kembali.

6. Implementasi

Setelah validasi berhasil dilakukan, algoritma *fuzzy*-ACS akan diimplementasikan pada masalah layanan *homecare* pada suatu perusahaan.

7. Penarikan Kesimpulan

Pada tahapan ini, hasil penelitian dirangkum untuk menjawab tujuan penelitian dan untuk membuat rekomendasi untuk penelitian selanjutnya.

3.3 Model Optimasi

Asumsi yang digunakan dalam pemodelan adalah sebagai berikut.

1. Setiap kendaraan selalu memulai dan mengakhiri rute di depot utama.
2. Kendaraan tidak dapat membatalkan layanan yang telah dimulai.
3. Setiap kendaraan hanya dapat melayani satu pelanggan pada satu waktu dan semua pelanggan hanya dapat dilayani satu kali.
4. Jumlah kendaraan yang tersedia terbatas.
5. Waktu layanan setiap pelanggan bersifat tidak pasti dan dimodelkan menggunakan bilangan *fuzzy* segitiga.
6. Pesanan baru dapat muncul kapan saja selama jam kerja.
7. Kendaraan memiliki batas waktu kerja maksimum yang tidak boleh dilampaui.
8. Rute kendaraan diperbarui secara berkala dalam interval waktu tertentu.

Penelitian ini menyelesaikan permasalahan dengan menerapkan DVRP sebagai representasi model. Terdapat dua fokus permasalahan utama, yaitu pelanggan yang bersifat dinamis dan waktu pelayanan yang tidak pasti. Untuk menangani pelanggan dinamis, penelitian ini mengadaptasi pendekatan VRP statis yang diperkenalkan oleh Kuo dkk. (2016), di mana DVRP dimodelkan sebagai serangkaian VRP statis yang diperbarui secara berkala.

Misalkan total jam kerja dinotasikan sebagai T . Pendekatan VRP statis membagi T ke dalam beberapa interval waktu sebanyak i . Untuk setiap interval t_i dengan $i = 1, \dots, i$, algoritma VRP statis digunakan untuk menentukan rute optimal berdasarkan informasi terbaru, termasuk daftar pelanggan yang belum dilayani dan lokasi mereka.

Misalkan:

- T : Total jam kerja.
- H : Jumlah kendaraan.
- t_i : Interval waktu i , $i = 1, \dots, i$.
- C_i : Himpunan pelanggan yang harus dipertimbangkan pada interval waktu t_i .
- N_i : Himpunan pelanggan yang datang selama interval t_i , di mana N_0 adalah himpunan pelanggan yang belum dilayani pada waktu kerja sebelumnya. Himpunan ini didefinisikan sebagai $N = \{N_0, N_1, \dots, N_m\}$.
- S_i : Himpunan pelanggan yang dilayani pada interval waktu t_i . Himpunan ini didefinisikan sebagai $S_i = \{S_0, S_1, \dots, S_m\}$, dimana S_0 melambangkan depot dan $|S|$ adalah jumlah pelanggan yang dilayani.
- e_{ij} : Waktu perjalanan dari pelanggan i ke j .
- u_i : Waktu pelayanan pelanggan i .
- a_i^k : Waktu sampai kendaraan k pada pelanggan i .
- w_i : Waktu tunggu pelanggan i dan \bar{W} adalah rata-rata waktu tunggu pelanggan.
- \tilde{r}_i : Waktu pelanggan siap dilayani
- m : Jumlah semut

Pada awal jam kerja, semua kendaraan memulai perjalanan dari depot, sehingga titik awal kendaraan untuk interval waktu t_1 adalah S_0 . Pada interval waktu berikutnya, t_i dengan $i = 2, \dots, t_i$, setiap kendaraan memulai perjalanan dari posisinya saat itu. Selama interval t_i , rute yang dibentuk harus mempertimbangkan seluruh pelanggan yang belum dilayani pada interval sebelumnya serta pelanggan baru yang muncul selama interval t_{i-1} . Selain itu, S_0 tetap diperhitungkan sebagai tujuan akhir bagi setiap kendaraan (Kuo, dkk., 2016).

$$C_i = \bigcup_{j=0}^i N_j \setminus \left\{ \bigcup_{j=1}^{i-1} S_j, S_0 \right\} \quad (3.1)$$

Karena keterbatasan jam kerja, beberapa pelanggan mungkin tidak dapat dilayani ($N - S = \emptyset$). Dalam hal ini, pelanggan yang belum dilayani akan dilayani pada jam kerja berikutnya. Oleh karena itu, perusahaan tidak hanya menentukan rute optimal untuk semua pelanggan, tetapi juga memilih pelanggan mana yang akan dilayani dalam jam kerja saat ini.

Untuk merepresentasikan permasalahan ini secara matematis, digunakan variabel keputusan berikut.

$$x_{ik} = \begin{cases} 1, & \text{jika kendaraan } k \text{ melayani pelanggan } s_i \\ 0, & \text{lainnya} \end{cases}$$

$$y_{ilk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ melalui rute pelanggan } s_i \text{ ke } s_l \\ 0, & \text{lainnya} \end{cases}$$

Dalam penelitian ini, terdapat dua fungsi objektif utama, yaitu memaksimalkan jumlah pelanggan yang dilayani dan meminimalkan rata-rata waktu tunggu pelanggan.

$$\max |S| = \sum_{k \in H} \sum_{i \in S} x_{ik} \quad (3.2)$$

$$\min \bar{W} = \frac{1}{|S|} \sum_{i \in S} w_i \quad (3.3)$$

Untuk menyederhanakan masalah, kedua fungsi objektif tersebut digabung menjadi satu fungsi objektif (F) seperti berikut.

$$\max F = \frac{|S|}{\bar{W}} = \frac{\sum_{k \in H} \sum_{i \in S} x_{ik}}{\frac{1}{|S|} \sum_{i \in S} w_i} \quad (3.4)$$

Selain itu, model ini juga memiliki sejumlah kendala yang harus dipenuhi. Kendala tersebut adalah sebagai berikut.

1. Setiap kendaraan k hanya dapat memulai perjalanan dari depot (S_0) ke satu pelanggan di subset S sebanyak satu kali.

$$\sum_{j \in S} y_{0jk} \leq 1, \quad \forall k = 1, \dots, H. \quad (3.5)$$

2. Jumlah kendaraan yang keluar dari depot harus sama dengan jumlah kendaraan yang kembali ke depot.

$$\sum_{i \in S} y_{i0k} - \sum_{i \in S} y_{0jk} = 0, \quad \forall k = 1, \dots, H. \quad (3.6)$$

3. Setiap pelanggan dilayani tepat satu kali oleh kendaraan dalam set H .

$$\sum_{i \in S} \sum_{k \in H} y_{ijk} = 1, \quad \forall j \in S/\{S_0\}. \quad (3.7)$$

$$\sum_{j \in S} \sum_{k \in H} y_{ijk} = 1, \quad \forall i \in S/\{S_0\}. \quad (3.8)$$

4. Jumlah waktu tempuh dan waktu layanan untuk setiap kendaraan k tidak melebihi batas total jam kerja T .

$$\sum_{i \in S} \sum_{j \in S} (e_{ij} + u_j) y_{ijk} \leq T, \quad \forall k = 1, \dots, H. \quad (3.9)$$

Batasan waktu dalam total jam kerja perlu disesuaikan untuk setiap interval waktu i . Pada interval i , jam kerja yang tersisa adalah $T - t_i$, yang mencakup waktu yang tersedia bagi setiap kendaraan untuk berpindah dari pelanggan terakhir pada interval sebelumnya ke pelanggan berikutnya. Selain itu, waktu perjalanan dari pelanggan terakhir kembali ke depot juga harus diperhitungkan. Dengan demikian batasan waktu pada setiap interval dapat dirumuskan sebagai berikut.

$$\sum_{i \in S_i} e_{li} y_{lik} + \sum_{i \in S_i} \sum_{j \in S_i} (e_{ij} + u_j) y_{ijk} + \sum_{i \in S_i} e_{i0} y_{i0k} \leq T - t_i, \quad (3.10)$$

$$\forall k = 1, \dots, H.$$

di mana indeks l merupakan pelanggan terakhir yang dikunjungi pada interval waktu t_{i-1} sebelumnya. Kemudian batasan waktu ini dikembangkan kembali dengan mengubah waktu layanan pasti u_i menjadi bilangan *fuzzy* segitiga $\tilde{u}_i = (u_{i1}, u_{i2}, u_{i3})$. Batasan waktu *fuzzy* ini dirumuskan dalam persamaan berikut.

$$\sum_{i \in S_i} e_{li} y_{lik} + \sum_{i \in S_i} \sum_{j \in S_i} (e_{ij} + \tilde{u}_j) y_{ijk} + \sum_{i \in S_i} e_{i0} y_{i0k} \leq T - t_i, \quad (3.11)$$

$$\forall k = 1, \dots, H.$$

$$\tilde{a}_{jk} = \tilde{a}_{ik} + \tilde{u}_i + e_{ij}, \quad \text{jika } x_{ijk} = 1. \quad (3.12)$$

Batasan waktu *fuzzy* membandingkan antara bilangan *fuzzy* dengan bilangan non-*fuzzy*. Penelitian ini menggunakan teori kredibilitas *fuzzy* untuk menyelesaikan batasan tersebut. Kredibilitas yang mengukur kemungkinan bahwa waktu layanan yang diperlukan tidak akan melebihi jam kerja yang tersedia dapat dirumuskan sebagai berikut.

$$Cr \left(\sum_{i \in S_i} e_{li} y_{lik} + \sum_{i \in S_i} \sum_{j \in S_i} (e_{ij} + \tilde{u}_j) y_{ijk} + \sum_{i \in S_i} e_{io} y_{iok} \leq T - t_i \right) \quad (3.13)$$

$$\geq Cr^*, \quad \forall k = 1, \dots, H.$$

Parameter Cr^* merepresentasikan indeks preferensi pengatur terhadap tingkat risiko dalam perencanaan rute. Nilai Cr^* yang tinggi akan menghasilkan solusi yang lebih optimal, tetapi juga meningkatkan kemungkinan terjadinya kegagalan. Misalnya, kendaraan mungkin tidak dapat menyelesaikan pelayanan kepada semua pelanggan atau tidak dapat kembali ke depot tepat waktu. Karena waktu layanan berbentuk *fuzzy*, waktu kedatangan dan waktu tunggu juga berubah menjadi bilangan *fuzzy*. Oleh karena itu, rata-rata waktu tunggu yang dijelaskan dalam Persamaan (3.3) juga diubah menjadi bilangan *fuzzy*. Untuk menghindari nilai objektif yang bersifat *fuzzy*, digunakan teknik defuzzifikasi untuk mengubah \bar{W} menjadi bilangan pasti. Dalam model ini, waktu tunggu *fuzzy* (\tilde{w}_i) untuk pelanggan i dihitung berdasarkan waktu kedatangan *fuzzy* (\tilde{a}_i) dan waktu siap pelanggan untuk dilayani (\tilde{r}_i) dengan rumus sebagai berikut.

$$\tilde{w}_i = \max(0, \tilde{a}_i - \tilde{r}_i) \quad (3.14)$$

di mana \tilde{w}_i didefinisikan sebagai bilangan *fuzzy* segitiga (w_{i1}, w_{i2}, w_{i3}). Rata-rata waktu tunggu hitung dengan rumus sebagai berikut.

$$\bar{W} = \frac{\sum_{i \in S/\{S_0\}} (w_{i1} + 2(w_{i2}) + w_{i3})}{4|S|} \quad (3.15)$$

Dengan mensubstitusikan Persamaan (3.14) kedalam fungsi objektif pada Persamaan (3.4) menjadi sebagai berikut:

$$\max F = \frac{|S|}{\bar{W}} = \frac{\sum_{k \in H} \sum_{i \in S} x_{ik}}{\frac{1}{4|S|} \sum_{i \in S/\{S_0\}} (w_{i1} + 2(w_{i2}) + w_{i3})} \quad (3.16)$$

Penelitian ini menggunakan *fuzzy-ACS* untuk menyelesaikan *fuzzy-DVRP*. Pada tahap awal, algoritma *fuzzy-ACS* digunakan untuk menentukan rute optimal dengan mempertimbangkan pelanggan dalam himpunan awal C_0 . Selanjutnya, dalam setiap interval waktu i , *fuzzy-ACS* digunakan kembali untuk mencari rute optimal yang diperbarui. Dengan total T jam kerja yang dibagi menjadi i interval waktu, algoritma ini akan dijalankan sebanyak i kali sepanjang periode kerja.

Pendekatan *fuzzy-ACS* yang digunakan dalam penelitian ini menyempurnakan tahap inisialisasi yang ada pada algoritma ACS biasa, yang umumnya menghasilkan nilai *pheromone* awal secara acak. Karena *pheromone* memiliki pengaruh yang signifikan dalam membentuk rute, penggunaan *pheromone* awal yang lebih baik dapat mempercepat pencapaian solusi yang lebih optimal (Zulvia, dkk., 2012). Oleh karena itu, *fuzzy-ACS* ini memanfaatkan metode penyisipan klaster (*cluster insertion method*) untuk menghasilkan *pheromone* awal yang lebih baik daripada menggunakan *pheromone* acak.

Dalam metode penyisipan klaster, pelanggan dikelompokkan ke dalam beberapa klaster kecil berdasarkan jarak antar pelanggan. Jumlah klaster disesuaikan dengan jumlah kendaraan, karena setiap kendaraan hanya melayani satu klaster. Untuk setiap klaster, dibuat rute awal secara kasar. Selama jam kerja, metode ini terus berjalan dengan menyisipkan pelanggan baru ke dalam rute yang sedang berlangsung. Jika terdapat pelanggan baru yang belum dilayani, metode ini menentukan kendaraan mana yang sebaiknya melayani pelanggan tersebut dan memperbarui rutenya.

Pada metode ini, rute yang menghubungkan pelanggan dalam satu klaster memiliki tingkat *pheromone* lebih tinggi dibandingkan rute yang menghubungkan pelanggan dari klaster berbeda. Hal ini menyebabkan kendaraan cenderung melayani pelanggan dalam klaster yang sama. Jika terdapat pelanggan baru, ia akan dimasukkan ke dalam klaster terdekat berdasarkan jarak *Euclidean*. Setiap klaster memiliki titik pusat, yang dihitung berdasarkan rata-rata posisi pelanggan dengan persamaan sebagai berikut.

$$x_{mean} = \frac{\sum x_i}{n} \quad (3.17)$$

$$y_{mean} = \frac{\sum y_i}{n} \quad (3.18)$$

di mana jarak antara pelanggan baru dengan kluster dihitung menggunakan jarak *Euclidean* yang dirumuskan sebagai berikut.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.19)$$

Dengan x_i, y_i dan x_j, y_j adalah koordinat pelanggan i dan j . Pelanggan baru kemudian dimasukkan ke kluster terdekat.

Metode penyisipan kluster ini bersifat non-hierarki, di mana jumlah kluster telah ditentukan sejak awal dan pembagian kluster dapat berubah secara dinamis seiring dengan penyisipan pelanggan baru.

Setelah pelanggan baru dimasukkan ke kluster, rute kendaraan yang sedang berjalan perlu diperbarui. Misalkan pelanggan baru s_n akan dimasukkan ke dalam rute kendaraan S_p^i , maka pelanggan tersebut akan dimasukkan ke dalam rute yang menghubungkan dua pelanggan sebelumnya, yaitu s_a dan s_b . Pelanggan baru dapat dimasukkan di antara s_a dan s_b pada posisi yang menghasilkan total jarak atau biaya perjalanan tambahan paling rendah. Selain itu, kelayakan penyisipan pelanggan baru ke dalam rute yang sedang berjalan juga diperhitungkan. Penyisipan hanya diterima jika tidak ada kendala yang dilanggar.

Agar rute tetap optimal, algoritma *fuzzy-ACS* dijalankan pada setiap interval waktu untuk memperbarui rute yang sedang berlangsung. Pada awal setiap interval, kendaraan akan memulai perjalanan dari pelanggan terakhir yang dikunjungi pada interval sebelumnya, yaitu s_i . Selanjutnya, setiap semut akan memilih pelanggan berikutnya (s_j) berdasarkan aturan berikut.

$$s_j = \begin{cases} \operatorname{argmax}_{i,j \in S} \tau_{ij}^\alpha \eta_{ij}^\beta, & \text{jika } q \leq q_0 \\ s_k, & \text{lainnya} \end{cases} \quad (3.20)$$

di mana s_k merupakan pelanggan yang belum dikunjungi. Jika seekor semut memutuskan untuk mengabaikan informasi yang ada dalam *pheromone* ($q \leq q_0$), maka semut tersebut akan memilih pelanggan berikutnya secara acak.

Setelah semut menyelesaikan rute, *pheromone* diperbarui dengan persamaan berikut.

$$\tau_{ij_{lokal}} = (1 - \rho)\tau_{ij} + \eta_{ij} \quad (3.21)$$

di mana ρ merupakan tingkat penguapan dan η_{ij} kebalikan dari jarak i ke j . Selanjutnya pembaruan *pheromone* global dilakukan di akhir iterasi, di mana rute terbaik diperbarui dengan persamaan berikut.

$$\tau_{ij_{global}} = \begin{cases} (1 - \rho)\tau_{ij_{lokal}} + \rho\zeta, & \text{rute } ij \text{ bagian dari rute terbaik} \\ \tau_{ij_{lokal}}, & \text{lainnya} \end{cases} \quad (3.22)$$

di mana $\zeta = \frac{1}{L_{best}}$ dengan L_{best} sebagai panjang rute terbaik yang ditemukan selama iterasi.

Model matematika yang diusulkan menggunakan variabel keputusan biner, sebagaimana tercantum pada persamaan variabel keputusan. Namun, solusi yang dihasilkan oleh algoritma *fuzzy-ACS* sudah berbentuk rute tetap. Oleh karena itu, tidak perlu mengonversi rute tetap tersebut menjadi nilai biner, karena aturan dalam *fuzzy-ACS* telah mempertimbangkan kelayakan solusi sesuai dengan kendala dalam Persamaan (3.5 – 3.8).

Untuk menentukan pelanggan berikutnya, algoritma *fuzzy-ACS* hanya mengevaluasi kendala waktu yang tercantum dalam Persamaan (3.10 – 3.13). selain itu, Persamaan (3.20) hanya mempertimbangkan pelanggan yang belum dikunjungi, sehingga memastikan bahwa setiap pelanggan tidak dikunjungi lebih dari satu kali. kendala waktu dalam Persamaan (3.13) juga memperhitungkan waktu perjalanan kembali ke depot setelah semua pelanggan telah dilayani. Dengan demikian, seluruh kendala tetap terpenuhi tanpa perlu mengubah format rute tetap menjadi variabel biner.

3.4 Teknik Penyelesaian

DVRP merupakan masalah penentuan rute kendaraan dalam situasi yang berubah secara dinamis. Algoritma umum untuk menyelesaikan DVRP merupakan pendekatan komputasional yang dikembangkan untuk mengoptimalkan rute. Algoritma DVRP mencakup beberapa metode, seperti algoritma genetika, ACO, dan pendekatan heuristik lainnya, yang bertujuan untuk menemukan solusi optimal dalam waktu yang terbatas, terutama ketika parameter seperti permintaan pelanggan atau waktu layanan dapat berubah secara tidak terduga.

Dalam penelitian ini, DVRP akan diselesaikan dengan menggunakan metode *fuzzy*-ACS merujuk pada penelitian Kuo dkk. (2016). *Fuzzy*-ACS merupakan pengembangan dari algoritma ACS yang dilengkapi dengan pendekatan *fuzzy* untuk menangani ketidakpastian dalam permasalahan DVRP. Langkah kerja *fuzzy*-ACS dalam menyelesaikan DVRP dapat diurutkan seperti berikut.

1. Koversi Ketidakpastian Pada Waktu Pelayanan Menggunakan Bilangan *Fuzzy*
Waktu pelayanan yang diperoleh dari data historis dikonversi menjadi bilangan *fuzzy* segitiga yang memiliki tiga parameter utama $\{a, b, c\}$ yang merepresentasikan waktu pelayanan masing-masing pelanggan.

2. Inisialisasi Parameter Awal dan Penyisipan Kluster

Pada tahap ini, dilakukan inisialisasi parameter $q_0, \alpha, \beta, \rho, Cr^*$ serta menetapkan k kendaraan yang akan melayani pelanggan. Kemudian, metode *cluster insertion* digunakan untuk membentuk solusi awal, menggunakan jarak *Euclidean*. Pelanggan dikelompokkan ke kluster yang memiliki jarak geometris terdekat dengan pusat kluster awal. Setiap kluster diasosiasikan dengan satu kendaraan.

2. Konstruksi Solusi oleh Setiap Kelompok Semut (G) dan Evaluasi Rute

Pada tahap ini, konstruksi solusi dilakukan oleh beberapa kelompok semut (G), di mana setiap kelompok terdiri dari semut-semut yang mewakili kendaraan. Pada setiap interval waktu, semut membangun rute dengan memilih pelanggan berdasarkan probabilitas yang dipengaruhi oleh tingkat *pheromone* dan daya tarik, sesuai dengan Persamaan (3.20). Proses dimulai dengan kendaraan yang membangun rutenya dengan memilih pelanggan berikutnya selama masih ada pelanggan yang belum dilayani dan waktu kerja kendaraan memungkinkan. Evaluasi dilakukan untuk memastikan waktu perjalanan dan layanan ke pelanggan berikutnya tidak melebihi batas waktu interval. Pada akhir interval, sebelum kendaraan kembali ke depot, dilakukan evaluasi total waktu *fuzzy* untuk memastikan validitas rute. Evaluasi ini menggunakan kredibilitas *fuzzy* (Cr^*) untuk memeriksa apakah total waktu *fuzzy* masih dalam batas waktu kerja yang ditentukan. Jika tidak memenuhi Cr^* , rute dianggap tidak valid atau tidak diterima.

3. Penanganan Pelanggan Baru dengan Metode Penyisipan Klaster

Ketika pelanggan baru muncul, pelanggan tersebut dimasukkan ke dalam klaster terdekat menggunakan metode penyisipan klaster. Jarak antara pelanggan baru dengan pusat klaster dihitung menggunakan jarak *Euclidean*. Pelanggan dimasukkan ke posisi dalam rute yang menghasilkan jarak terkecil. Penyisipan dilakukan dengan mempertimbangkan semua kemungkinan posisi dalam rute, sambil memastikan tidak ada pelanggaran kendala pada batas waktu kerja. Setelah penyisipan selesai, rute kendaraan diperbarui.

4. Evaluasi Kendala Waktu

Selanjutnya, dilakukan evaluasi terhadap kendala waktu. Evaluasi ini mencakup perhitungan waktu *fuzzy*, yang mencakup waktu kedatangan kendaraan dan waktu tunggu pelanggan. Jika seluruh rute valid, nilai *fitness* dihitung berdasarkan jumlah pelanggan yang dilayani dan rata-rata waktu tunggu *fuzzy*.

5. Pembaruan *Pheromone* Lokal

Setelah rute divalidasi dan nilai *fitness* dihitung, *pheromone* pada rute yang dilalui semut diperbarui secara lokal untuk meningkatkan eksplorasi selama iterasi berikutnya menggunakan Persamaan (3.21).

6. Konstruksi Solusi oleh Semut dan Evaluasi Solusi

Selanjutnya, semut kembali mengkonstruksi solusi menggunakan *pheromone* yang telah diperbarui secara lokal. Solusi yang dihasilkan dievaluasi kembali berdasarkan nilai *fitness*. Jika solusi tersebut lebih baik dibandingkan dengan solusi terbaik sebelumnya, maka solusi tersebut disimpan sebagai solusi terbaik.

7. Pembaruan *Pheromone* Global

Pheromone diperbarui secara global untuk rute yang termasuk dalam solusi terbaik. Pembaruan *pheromone* global dilakukan dengan Persamaan (3.22). *Pheromone* yang didapat, digunakan untuk iterasi selanjutnya. Proses berulang hingga kriteria penghentian tercapai, seperti jumlah iterasi tertentu atau tidak adanya perubahan signifikan pada solusi. Solusi terbaik yang

ditemukan selama iterasi merupakan rute distribusi yang optimal dengan mempertimbangkan semua kendala.