

BAB III

METODOLOGI PENELITIAN

Bab ini membahas metodologi yang akan digunakan untuk menyelesaikan masalah *Multi Depot Vehicle Routing Problem with Time Windows* (MDVRPTW) dan Algoritma *Particle Swarm Optimization* (PSO).

3.1 Deskripsi Masalah

Penelitian ini membahas tentang *Multi Depot Vehicle Routing Problem with Time Windows* (MDVRPTW). MDVRPTW merupakan salah satu varian dari VRP dengan kendala tambahan berupa depot yang digunakan sebagai pusat distribusi sekaligus titik awal berangkat kendaraan dan titik akhir kendaraan terdapat lebih dari satu dan kegiatan distribusi dilakukan dalam rentang waktu tertentu. Terdapat sejumlah kendaraan dengan kapasitas tertentu yang harus melakukan pengantaran kepada sejumlah pelanggan dengan waktu pelayanan tertentu. Setiap kendaraan yang berangkat dari suatu depot untuk mengunjungi pelanggan harus kembali menuju depot semula.

Tujuan dari penyelesaian MDVRPTW adalah untuk menentukan himpunan rute optimal untuk mengantarkan barang ke sejumlah pelanggan menggunakan sejumlah kendaraan dengan mempertimbangkan total jarak tempuh dan waktu tempuh yang terbatas. Dalam penelitian ini, MDVRPTW akan diselesaikan menggunakan algoritma *Particle Swarm Optimization* (PSO).

3.2 Tahapan Penelitian

Penelitian ini dilakukan dalam tahapan berikut:

1. Studi Pustaka

Pada tahap ini, peneliti melakukan studi pustaka dengan mempelajari konsep atau teori-teori mengenai masalah penentuan rute dan model *Particle Swarm Optimization* bersumber dari berbagai literatur baik jurnal, buku, dan karya tulis lainnya.

2. Pengumpulan Data

Data yang digunakan pada penelitian ini terdiri dari data depot, data agen, data jumlah permintaan, data lokasi depot dan agen, data jarak depot ke agen, data jarak agen ke agen lainnya, dan data kendaraan serta kapasitasnya.

3. Pembangunan Model Optimisasi

Pada tahap ini, model permasalahan akan dibangun dengan mendefinisikan asumsi-asumsi, himpunan, parameter, variabel keputusan, fungsi tujuan, dan menetapkan kendala-kendala dari model MDVRPTW.

4. Penyelesaian Model Optimisasi

Penyelesaian model dilakukan dengan menggunakan algoritma *Particle Swarm Optimization* (PSO).

5. Validasi

Pada tahap ini, model dan penyelesaian masalah akan divalidasi dengan menggunakan data berukuran kecil yang memungkinkan diselesaikan secara manual. Solusi dari perhitungan manual akan dibandingkan dengan solusi hasil teknik penyelesaian model. Jika diperoleh perbedaan, maka tahapan diulang dari tahap pembangunan model.

6. Implementasi

Pada tahap implementasi, data yang telah dikumpulkan akan diimplementasikan kepada model optimisasi yang telah dibangun menggunakan algoritma *Particle Swarm Optimization* (PSO).

7. Penarikan Kesimpulan

Tahap terakhir adalah akan ditarik kesimpulan dari hasil rumusan masalah yang telah dirumuskan sebelumnya dan hasil dari implementasinya.

3.3 Asumsi Penelitian

Berdasarkan kasus dalam penelitian ini, maka akan diberikan asumsi penelitian sebagai berikut :

- a. Dalam pengambilan bahan baku, setiap rute yang terbentuk memiliki batasan waktu (*time windows*) yaitu 10 jam yang dimulai dari pukul 7.00 WIB sampai 17.00 WIB
- b. Setiap pelanggan memiliki batasan waktu pelayanan selama 30 menit.
- c. Setiap jarak jalan yang terbentuk dapat berlaku 2 arus.

3.4 Model Optimisasi

Untuk kepentingan pemodelan MDVRPTW, maka terlebih dahulu akan didefinisikan himpunan dan parameter sebagai berikut :

- Himpunan

V adalah himpunan gabungan dari pelanggan dan depot

N adalah himpunan pelanggan

M adalah himpunan depot

K adalah himpunan kendaraan pengangkut

- Parameter

Q adalah kapasitas maksimum setiap kendaraan pengangkut

d_i adalah jumlah permintaan pelanggan ke i

t_{ij} adalah waktu tempuh (*travel time*) dari lokasi i ke lokasi j

t_k adalah waktu tempuh (*travel time*) maksimum kendaraan pengangkut k

s_{ik} adalah waktu pelayanan (*service time*) kendaraan pengangkut k di lokasi i

a_i adalah waktu mulai pelayanan di lokasi i

e_j adalah waktu awal (buka) *time window* di lokasi j

l_i adalah waktu akhir (tutup) *time window* di lokasi i

Variabel keputusan model optimisasi didefinisikan untuk menentukan kendaraan yang akan mengunjungi pelanggan. Variabel ini didefinisikan sebagai berikut:

$$x_{ijk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ mengunjungi pelanggan } j \text{ setelah pelanggan } i \\ 0, & \text{yang lainnya.} \end{cases}$$

Tujuan dari penyelesaian MDVRPTW adalah untuk meminimumkan total waktu tempuh bagi semua kendaraan. Fungsi tujuan ini dituliskan sebagai:

$$\sum_{k \in K} \sum_{i \in V} \sum_{j \in V} x_{ijk} \cdot t_{ij} + s_{jk} \quad (3.1)$$

Adapun kendala dari model adalah sebagai berikut (Heechul & Ilkyeong, 2016) :

1. Setiap pelanggan dikunjungi sekali oleh kendaraan pengangkut

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

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

2. Setiap kendaraan berangkat dari depot dan kembali ke depot

$$\sum_{j \in N} x_{ijk} \leq 1, \forall k \in K, \forall i \in M \quad (3.4)$$

$$\sum_{i \in N} x_{ijk} \leq 1, \forall k \in K, \forall j \in M \quad (3.5)$$

2. Total produk yang diantar oleh kendaraan pengangkut tidak melebihi kapasitas maksimumnya (Q)

$$\sum_{i \in N} \sum_{j \in V} d_i x_{ijk} \leq Q, \forall k \in K \quad (3.6)$$

3. Setiap kendaraan harus patuh pada kendala *time window* di setiap *node* termasuk depot

$$a_j \geq a_i + s_{ik} + t_{ij} + \left(\sum_{k \in K} x_{ijk} - 1 \right), \forall i \in V, j \in N \quad (3.7)$$

$$t_k \geq a_j \geq a_i + s_{ik} + t_{ij} + (x_{ijk} - 1), \forall i \in N, j \in M, k \in K \quad (3.8)$$

4. Setiap kendaraan mulai *service* pada *time window* di setiap lokasi pelanggan.

$$e_i \leq a_i + s_{ik} \leq l_i, \forall i \in N \quad (3.9)$$

3.5 Penyelesaian Model Optimisasi menggunakan *Particle Swarm Optimization*

Permasalahan MDVRPTW termasuk dalam kategori *NP-Hard Problem*, yang artinya waktu komputasi yang digunakan akan semakin sulit seiring dengan meningkatnya ruang lingkup masalah (Nugroho, 2015). Ini berarti bahwa solusi *feasible* yang dihasilkan akan sangat banyak. Akibatnya, jika menggunakan solusi eksak untuk menyelesaikannya maka akan membutuhkan waktu yang lama (Arafah, 2021). Oleh karena itu, dalam penelitian ini akan digunakan solusi *metaheuristic* untuk mendapatkan penyelesaian yang lebih cepat yaitu dengan menggunakan algoritma *Particle Swarm Optimization*.

Tahapan yang akan dilakukan untuk menyelesaikan MDVRPTW terdiri dari 2 tahapan, yaitu *Classification* untuk mengelompokkan agen kepada depot terdekat, kemudian dilanjutkan dengan *Optimization* menggunakan metode *Particle Swarm Optimization* untuk menentukan urutan kunjungan sehingga mendapatkan hasil yang optimal.

3.5.1 Classification

Tahapan ini digunakan untuk mengelompokkan sejumlah agen dengan sebuah depot agar menjadi *single depot* VRPTW. Kelompok dari sebuah depot dan sejumlah agen ini selanjutnya akan disebut sebagai gugus. Dalam permasalahan ini, perhitungan akan berdasarkan *time window* dari setiap agen. Pelanggan akan dikelompokkan dengan depot yang memiliki *time window* yang berdekatan, dengan demikian suatu depot dapat memaksimalkan total tempuhnya perjalannya. Misalkan terdapat 2 depot, yaitu d_1 dan d_2 maka (Fitri, 2023):

1. Jika $d_{nd_i} < d_{nd_2}$, maka pelanggan n dikelompokkan dengan depot 1.
2. Jika $d_{nd_i} > d_{nd_2}$, maka pelanggan n dikelompokkan dengan depot 2.
3. Jika $d_{nd_i} = d_{nd_2}$, maka pelanggan n bebas dikelompokkan dengan depot 1 maupun depot 2.

3.5.2 Optimization

Tahapan ini digunakan untuk menentukan urutan kunjungan mulai dari agen pertama, kedua, ketiga, hingga agen terakhir dalam sebuah gugus. Metode yang akan digunakan adalah Algoritma *Particle Swarm Optimization*.

1) Inisialisasi Partikel

Parameter yang akan dioptimasi dalam sistem ini adalah *travel time*. Penentuan posisi awal partikel awal adalah nilai *travel time* yang ditentukan secara acak. Nilai *travel time* tersebut, pada iterasi ke-0 akan dibagikan ke setiap partikelnya. Proses pembagian nilai pada pada setiap partikelnya ini adalah inti dari PSO. Dengan menggunakan pBest dan gBest pada iterasi ke-0, nilai posisi partikel dalam proses inisialisasi ini bernilai total *travel time* setiap partikel. (Gamayanti dkk, 2015)

2) Fitness Partikel

Fase ini sudah memasuki tahap iterasi ke-1 hingga iterasi maksimum. Pada setiap iterasi, nilai *fitness* partikel sama dengan nilai dari total *travel time*. Namun *fitness* ini di tunjukkan untuk perbandingan nilai di setiap iterasi. Nilai *fitness* particle di setiap iterasi berbeda-beda. Nilai ini akan berubah dengan nilai total *travel time* pada iterasi ke-1. Apabila total *travel time* lebih optimal dari nilai iterasi sebelumnya, maka akan dipakai nilai total *travel time* yang baru. (Gamayanti dkk, 2015)

Khairunnisa Aulia Azzahra, 2024

**PENYELESAIAN MULTI DEPOT VEHICLE ROUTING PROBLEM WITH TIME WINDOWS
MENGUNAKAN PARTICLE SWARM OPTIMIZATION ALGORITHM**

(Studi Kasus: Rute Pengambilan Bahan Baku)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

3) Menentukan Nilai pBest dan gBest

pBest adalah nilai posisi partikel lokal. Penentuan pBest dilakukan pada iterasi ke-1 hingga iterasi maksimum. Nilai pBest dapat diperoleh setelah melalui fase *fitness* partikel dan bergantung pada nilai *fitness* partikel, sehingga akan berubah nilai setiap iterasi. Sedangkan gBest adalah nilai terbaik dari setiap nilai pBest (Gamayanti dkk, 2015).

4) *Updating* Kecepatan

Pada tahapan ini akan dihitung kecepatan dari semua partikel. Semua partikel bergerak menuju titik optimal dengan suatu kecepatan. Kecepatan awal dari semua partikel diasumsikan sama dengan nol dengan set iterasi $i=1$. Kecepatan partikel dihitung sebagai:

$$V_j(i) = V_j(i-1) + c_1 r_1 [P_{best,j} - x_j(i-1)] + c_2 r_2 [G_{best,j} - x_j(i-1)] \quad (3.14)$$

dengan $j = 1, 2, 3, \dots, N$. c_1 dan c_2 adalah *learning rates* untuk kemampuan individu dan pengaruh sosial. r_1 dan r_2 adalah bilangan random yang terdistribusi uniform dalam interval 0-1. Jadi parameter c_1 dan c_2 menunjukkan bobot dari memory (*position*) sebuah partikel terhadap memori dari kelompok (*swarm*). Nilai c_1 dan c_2 biasanya bernilai 2 sehingga perkalian $c_1 r_1$ dan $c_2 r_2$ memastikan bahwa partikel-partikel yang mendekati target sekitar setengah selisihnya (Gamayanti dkk, 2015).

5) *Updating* Posisi

Posisi pada partikel dihitung dengan:

$$X_j(i) = X_j(i-1) + V_j(i), j = 1, 2, 3, \dots, N \quad (3.15)$$

selanjutnya dilakukan pengecekan apakah solusi sekarang sudah konvergen, yaitu ketika semua partikel menuju ke satu nilai yang sama. Jika belum konvergen maka dilakukan *updating* kecepatan dengan memperbaharui iterasi $i=i+1$, dengan cara menghitung nilai baru dari pBest dan gBest. Proses ini akan terus berlanjut hingga semua partikel menuju satu titik solusi yang sama (Gamayanti dkk, 2015).

6) Parameter *Particle Swarm Optimization*

Dalam algoritma PSO terdapat beberapa parameter yang sudah ditentukan sejak awal, yaitu:

V = Kecepatan (*velocity*) dari partikel

c_1 dan c_2 = *Coefficient Acceleration*

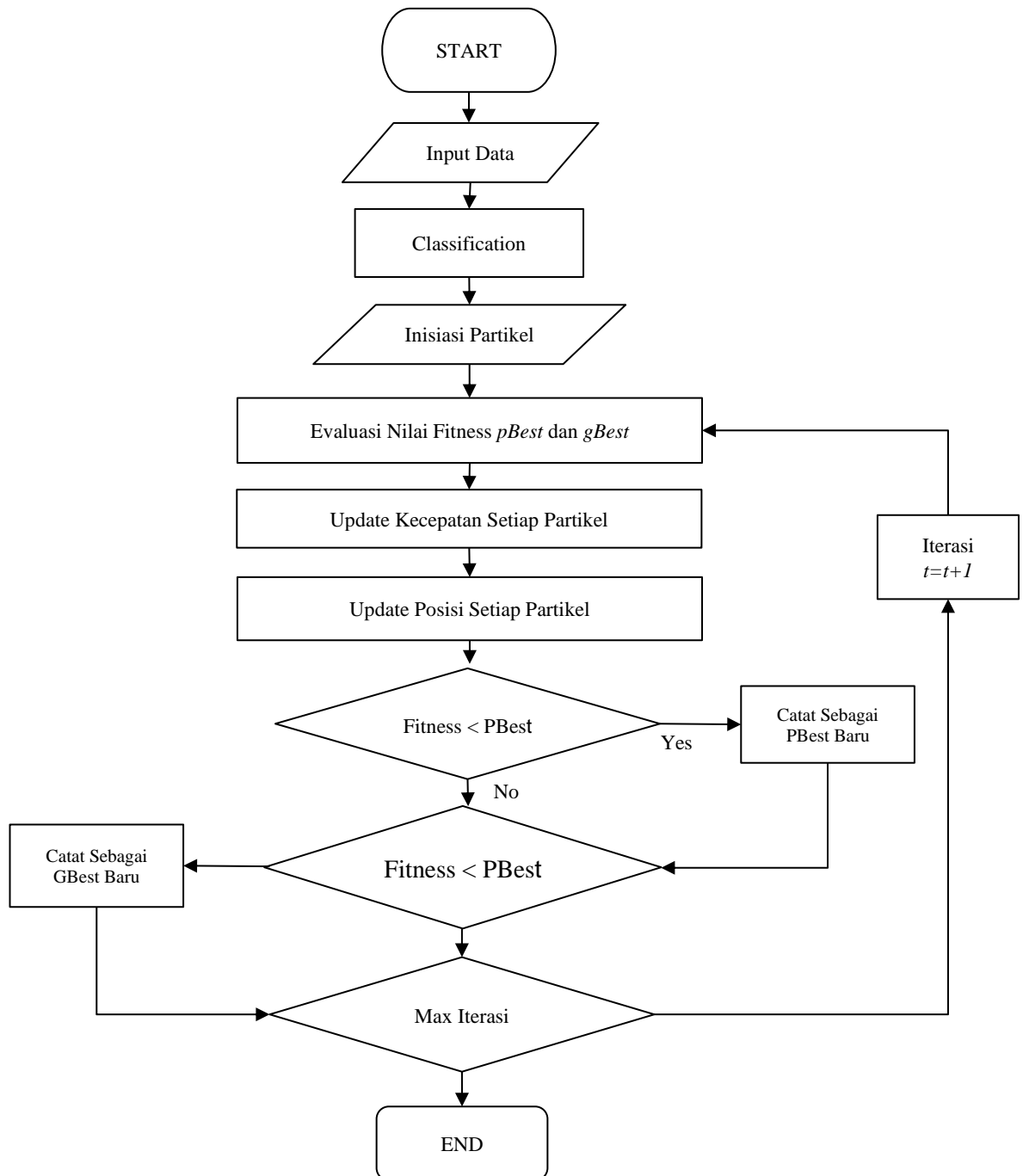
X_j = *Update posisi particle*

$x_j(i)$ = *Posisi partikel ke-i*

Swarm = *Banyaknya agen burung yang digunakan dalam optimasi*

Max Iteration = *Nilai yang menunjukkan setelah beberapa program akan dihentikan (Santosa, 2011).*

Tahapan dari Penyelesaian *Multi Depot Vehicle Routing Problem with Time Windows* Menggunakan Algoritma *Particle Swarm Optimization* digambarkan dalam bentuk flowchart pada Gambar 3.1.



Gambar 3. 1 Flowchart Algoritma PSO untuk Permasalahan MDVRPTW

3.4 Contoh Penyelesaian *Multi Depot Vehicle Routing Problem with Time Windows* Menggunakan Algoritma *Particle Swarm Optimization*

Diketahui sebuah perusahaan membutuhkan bahan baku susu untuk produksi setiap harinya. Bahan baku susu tersebut diperoleh dari 5 TPS, yang dalam perhitungan ini akan disebut sebagai agen, di beberapa tempat. Perusahaan tersebut memiliki 2 gudang penyimpanan (depot).

Dalam proses pendistribusian bahan baku, perusahaan mengharuskan ketepatan waktu distribusi. Jam kerja yang ditetapkan oleh perusahaan pada pukul 07:00 WIB hingga pukul 17:00 WIB, sehingga memiliki batasan waktu (*time window*) selama 10 jam dan waktu pelayanan setiap agen adalah 30 menit.. Data depot, pelanggan, koordinat titik, dan permintaan dari setiap pelanggan dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Data Depot, Pelanggan, Titik Koordinat, dan Permintaan

No	Tempat	Koordinat (Longitude, Latitude)	Permintaan (Liter)
0	Depot KPSBU	-6,817148,107,624716	0
1	Depot Pojok	-6,801554,107,605798	0
2	TPS Bale	-6,79686,107,650105	1050
3	TPS Pentas	-6,7974,107,652799	1265
4	TPS Nyalindung	-6,793035,107,649005	1750
5	TPS Babakan	-6,795831,107,648665	1555
6	TPS Mokla	-6,807822,107,576202	583

Langkah pertama adalah membuat matriks jarak untuk setiap lokasi antara depot dengan peternak menggunakan rumus *Haversine* berikut sehingga diperoleh jarak pada Tabel 3.2. Dengan r merupakan jari-jari bumi yaitu 6.371 Km dan Latitude – Longitude menggunakan koordinat tempat dalam data (Fitri, 2023).

$$d = 2r \cdot \arcsin \left\{ \sqrt{\sin^2 \left(\frac{Lat_1 - Lat_2}{2} \right) + \cos(Lat_1) \cdot \cos(Lat_2) \cdot \sin^2 \left(\frac{Long_1 - Long_2}{2} \right)} \right\} \quad (3.16)$$

Tabel 3. 2 Matriks Jarak (Km)

indeks	0	1	2	3	4	5	6
0	0	2,71	3,60	3,80	3,79	3,55	5,46
1	2,71	0	4,92	5,21	4,86	4,78	3,34
2	3,60	4,92	0	0,30	0,44	0,20	8,25
3	3,80	5,21	0,30	0	0,64	0,49	8,54
4	3,79	4,86	0,44	0,64	0	0,31	8,20
5	3,55	4,78	0,20	0,49	0,31	0	8,11
6	5,46	3,34	8,25	8,54	8,20	8,11	0

Khairunnisa Aulia Azzahra, 2024

**PENYELESAIAN MULTI DEPOT VEHICLE ROUTING PROBLEM WITH TIME WINDOWS
MENGUNAKAN PARTICLE SWARM OPTIMIZATION ALGORITHM**

(Studi Kasus: Rute Pengambilan Bahan Baku)

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Selanjutnya setelah matriks jarak tempuh terbentuk adalah setiap lokasi agen akan dikelompokkan ke dalam depot dengan jarak tempuh terdekat seperti berikut :

- $d_{20} = 3,6 < d_{21} = 4,92$, agen 2 dikelompokkan ke depot 0
- $d_{30} = 3,8 < d_{31} = 5,21$, agen 3 dikelompokkan ke depot 0
- $d_{40} = 3,79 < d_{41} = 4,86$, agen 4 dikelompokkan ke depot 0
- $d_{50} = 3,55 < d_{51} = 4,78$, agen 5 dikelompokkan ke depot 0
- $d_{60} = 5,46 < d_{61} = 3,34$, agen 2 dikelompokkan ke depot 1

Sehingga, depot 0 melayani agen 2, 3, 4, 5 dan depot 1 melayani agen 6.

Kemudian akan dilakukan optimisasi menggunakan Algoritma PSO pada masing-masing depot 0 dan depot 1 dengan langkah-langkah berikut.

Langkah 1: Inisialisasi Parameter

- Kode Depot = Depot 0
- Jumlah agen = 4 agen
- Jumlah swarm = 4 swarm
- Kecepatan rata-rata kendaraan = 25 km/jam
- Waktu pelayanan = 30 menit
- Batas waktu distribusi = 10 jam
- Jumlah iterasi = 4 iterasi

Langkah 2: Inisialisasi Posisi Awal (C_1)

Posisi awal akan dibangkitkan secara random antara [0,1] pada partikel-partikelnya. Pada perhitungan ini menggunakan jumlah *swarm* sebanyak 4.

$$pop_1(C) = \begin{bmatrix} 0,2077 & 0,0960 & 0,9997 & 0,8875 \\ 0,4638 & 0,7437 & 0,6952 & 0,5225 \\ 0,4235 & 0,2659 & 0,6641 & 0,5555 \\ 0,3643 & 0,6273 & 0,6271 & 0,2096 \end{bmatrix}$$

Urutkan setiap bilangan random dari yang terkecil, yang kemudian hasil pengurutan ini akan menghasilkan sebuah rute.

$$pop_1(C) \text{ urut} = \begin{bmatrix} 0,2077 & 0,8875 & 0,0960 & 0,9997 \\ 0,4638 & 0,5225 & 0,6952 & 0,7437 \\ 0,2659 & 0,4235 & 0,5555 & 0,6641 \\ 0,2096 & 0,3643 & 0,6271 & 0,6273 \end{bmatrix}$$

Setelah masing-masing partikel diurutkan, maka akan diperoleh rute yang dibentuk dari posisi bilangan random tersebut sebelum diurutkan.

$$rute = \begin{bmatrix} 2 & 5 & 3 & 4 \\ 2 & 5 & 4 & 3 \\ 3 & 2 & 5 & 4 \\ 5 & 2 & 4 & 3 \end{bmatrix}$$

Sebagai ilustrasi, misalkan *swarm* partikel pertama sebelum diurutkan adalah $C_1 = [0,2077 \quad 0,0960 \quad 0,9997 \quad 0,8875]$. Dari urutan bilangan random ini menghasilkan rute 1 yaitu $0 - 2 - 5 - 3 - 4 - 0$. Untuk solusi partikel lain dilakukan cara yang sama untuk mendapatkan rute.

Langkah 3: Menentukan nilai *fitness*

Dalam kasus ini, terdapat *time windows* yang perlu diperhatikan, maka nilai *fitness* di tentukan berdasarkan lamanya *travel time* dari setiap rute. Untuk mendapatkan *travel time* yang terdiri dari waktu tempuh antar depot ke agen, agen satu dengan agen lainnya, dan dari agen ke depot kembali ditambahkan dengan total waktu pelayanan dari setiap agen, sebagai ilustrasi akan dihitung *travel time* pada *swarm* pertama sebagai berikut :

$$T_1 = t_1 + a_1 \quad (3.17)$$

$$t_1 = \frac{\text{jarak}}{\text{kecepatan kendaraan}} \quad (3.18)$$

$$a_1 = \text{banyak depot} \times 0,5 \text{ jam} \quad (3.19)$$

Jadi $T_1 = t_1 + a_1 = 0,3468 + 2 = 2,349 \text{ jam}$.

Artinya *swarm* pertama membentuk rute dengan *travel time* 2,349 jam, di mana dalam kasus ini *travel time* tersebut tidak melanggar ketentuan *time windows* yang ditentukan yaitu 10 jam. Untuk nilai *fitness* pada *swarm* lain dilakukan cara yang sama untuk mendapatkan *travel time*.

Tabel 3. 3 Rute Solusi Iterasi Pertama

Swarm	Rute	Travel Time (jam)
1	0 - 2 - 5 - 3 - 4 - 0	2,3486
2	0 - 2 - 5 - 4 - 3 - 0	2,3419
3	0 - 3 - 2 - 5 - 4 - 0	2,3362
4	0 - 5 - 2 - 4 - 3 - 0	2,3452

Langkah 4: Menentukan p_{best} awal dan g_{best} awal

Untuk setiap partikel, p_{best} awal akan sama dengan nilai awal partikel. Sedangkan g_{best} awal diperoleh setelah mengevaluasi setiap partikel dan memperoleh solusi yang sesuai dengan kendala MDVRPTW. Sehingga diperoleh:

$$p_{best} = [0,2077 \quad 0,0960 \quad 0,9997 \quad 0,8875]$$

dan g_{best} diperoleh pada *swarm* ke-3 yang memiliki nilai *travel time* terkecil, yaitu:

$$g_{best} = [0,4235 \quad 0,673 \quad 0,6271 \quad 0,2096]$$

Langkah 5: *Update* kecepatan

Untuk kecepatan awal akan diberikan V_0 sebagai berikut:

$$V_0 = \begin{bmatrix} 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 \\ 0,10 & 0,10 & 0,10 & 0,10 \end{bmatrix}$$

Kemudian untuk *update* kecepatan akan diberikan persamaan sebagai berikut :

$$r_1 = 0,2$$

$$c_1 = 2$$

$$r_1 = 0,4$$

$$c_1 = 2$$

$$\omega = 0,9 \text{ (constant inertia weight)}$$

$$V_1(C_1) = \omega V_0 + c_1 r_1 (p_{best} - C_1) + c_2 r_2 (g_{1best} - C_1) \quad (3.20)$$

$$V_1'(C_1) = 0,9 [0,10 \quad 0,10 \quad 0,10 \quad 0,10]$$

$$+ (2)(0,2)([0,2077 \quad 0,0960 \quad 0,9997 \quad 0,8875]$$

$$- [0,2077 \quad 0,0960 \quad 0,9997 \quad 0,8875])$$

$$+ (2)(0,4)([0,4235 \quad 0,673 \quad 0,6271 \quad 0,2096]$$

$$- [0,2077 \quad 0,0960 \quad 0,9997 \quad 0,8875])$$

$$V_1(C_1) = [0,2626 \quad 0,2259 \quad -0,1785 \quad -0,1756]$$

Untuk perhitungan $V_2(C_2)$ sampai $V_4(C_4)$ sama seperti perhitungan $V_1(C_1)$

Langkah 6: *Update* posisi partikel baru

Untuk setiap partikel, nilainya akan di *update* dengan persamaan berikut :

$$C_1' = C_1 + V_1' \quad (3.21)$$

$$C_1' = [0,2077 \quad 0,0960 \quad 0,9997 \quad 0,8875]$$

$$+[0,2626 \quad 0,2259 \quad -0,1785 \quad -0,1756]$$

$$C_1' = [0,4703 \quad 0,3219 \quad 0,8212 \quad 0,7119]$$

Untuk perhitungan C'_2 sampai C'_4 dilakukan cara yang sama seperti perhitungan C'_1 . Kemudian untuk menentukan posisi partikel baru diperoleh dari nilai baru dibagi dengan jumlah totalnya setiap partikel. Dalam hal ini jumlah total dari $C'_1 = 2,3254$, sehingga nilai baru setelah setiap partikel dibagi dengan jumlah totalnya adalah berikut:

$$C'_{1baru} = [0,2023 \quad 0,1384 \quad 0,3531 \quad 0,3062]$$

Untuk perhitungan C'_{2baru} sampai C'_{4baru} dilakukan cara yang sama seperti perhitungan C'_{1baru} . Dari perhitungan nilai partikel baru ini akan diperoleh populasi partikel baru untuk iterasi selanjutnya, yaitu:

$$pop_2(C) = \begin{bmatrix} 0,2023 & 0,1384 & 0,3531 & 0,3062 \\ 0,1839 & 0,844 & 0,371 & 0,3445 \\ 0,1794 & 0,1209 & 0,3730 & 0,3268 \\ 0,1831 & 0,0899 & 0,3735 & 0,3534 \end{bmatrix}$$

Kemudian akan dilakukan kembali perhitungan pada langkah 2 hingga iterasi optimal, maka algoritma akan berhenti. Dalam perhitungan ini, pada iterasi ke-4 diperoleh solusi rute optimal sebagai berikut :

Tabel 3. 4 Rute Solusi Iterasi Keempat

Swarm	Rute	Travel Time (jam)
1	0 - 5 - 3 - 4 - 2 - 0	2,3363
2	0 - 3 - 2 - 4 - 5 - 0	2,3364
3	0 - 3 - 2 - 5 - 4 - 0	2,3362
4	0 - 3 - 2 - 4 - 5 - 0	2,3364

Jadi, algoritma telah berhenti pada iterasi ke-4 dengan nilai g_{best} terbaik pada swarm ke-3. Sehingga rute yang terbentuk untuk Depot 0 adalah 0 – 3 – 2 – 5 – 4 – 0 dengan *travel time* 2,3362 jam, artinya rute ini tidak melanggar ketentuan *time windows* yang ditentukan. Sedangkan untuk mendapatkan rute solusi Depot 1 dilakukan langkah yang sama seperti perhitungan Depot 0. Dari perhitungan tersebut rute yang terbentuk untuk depot 1 adalah 1 – 6 – 1 dengan *travel time* 2,2673 jam yang artinya tidak melanggar ketentuan *time windows* yang ditentukan.