



Studi Kasus

Optimasi Biaya Distribusi pada HFVRP Menggunakan Algoritma *Particle Swarm Optimization*

Ismanitami Lukman, Rosmalina Hanafi, Syarifuddin M. Parenreng

Departemen Program Studi Teknik Industri, Fakultas Teknik, Universitas Hasanuddin, Jl. Perintis Kemerdekaan KM.10, Kota Makassar, Sulawesi Selatan 90245, Indonesia

ARTICLE INFORMATION

Received: July 23, 19
Revised: October 29, 19
Available online: October 31, 19

KEYWORDS

Distribusi, Heterogeneous Fleet Vehicle Routing Problem, Particle Swarm Optimization

CORRESPONDENCE

Phone: +62 (0751) 12345678
E-mail: lukmani15d@student.unhas.ac.id

A B S T R A C T

Distribution systems play a significant role in logistics operations. For the companies with consumer goods products this even more important as consumer goods production has fairly cheap price compared to the distribution cost that must be spent by the company. In addition, increased fuel costs have urged the company to be more efficient in planning and schedule the transportation routes. This paper presents the application of the Particle Swarm Optimization (PSO) algorithm to minimize the travel distance and total cost of a Heterogeneous Fleet Vehicle Routing Problem (HFVRP). Experimental results from its application to a real-world case study are presented. The model in this research is the HFVRP where vehicles have different capacities, variable costs, and fixed costs. PSO algorithm was applied because of the high number of customers served, and therefore the exact methods may not be sufficient. PSO parameter setting which produced the optimum result was with the number of swarms 50, C1 1.5, and C2 2 determined through the design of the experiment. The results of computation show that using PSO can minimize the total traveled distance with an average savings of 51.55% and minimize total cost with an average savings of 44.92% from the existing vehicle routes operated by the company.

PENDAHULUAN

Distribusi memegang peran penting sebagai penghubung antara perusahaan dengan pelanggan. Permasalahan distribusi pada perusahaan dengan produk *consumer goods* yaitu produk *consumer goods* memiliki harga cukup murah dibandingkan dengan biaya distribusi yang harus dikeluarkan perusahaan dan peningkatan biaya bahan bakar telah mendesak perusahaan logistik dan distribusi untuk dapat lebih efisien dalam merencanakan rute transportasi dan menjadwalkannya. Sehingga diperlukan suatu upaya pengoptimalan pada sistem distribusi dalam rangka pencapaian keuntungan yang optimum.

Pengelolaan sistem distribusi yang optimal dapat membantu perusahaan dalam kompetisi perekonomian dengan keputusan strategis dan operasional yang tepat. Keuntungan lainnya pengoptimalan sistem distribusi menyediakan nilai tambah pada *supply chain* dengan dilakukannya perencanaan sehingga memperoleh pengurangan biaya operasi distribusi dan standar pengiriman yang lebih baik.

Menurut Parkhi *et al.* [1] sebuah rute yang baik untuk distribusi logistik dapat memotong biaya transportasi dan meningkatkan

efisiensi. Efisiensi dari rute yang baik akan berpengaruh terhadap peningkatan keseluruhan sistem logistik (*moving load, delivery speed, service quality, dan usage of facilities*) dan pengurangan biaya transportasi serta penghematan energi.

Permasalahan pengoptimalan pada aktivitas transportasi biasa juga disebut *Vehicle Routing Problem*. *Vehicle routing problem* merupakan salah satu bentuk permasalahan di bidang manajemen logistik dimana mengoptimasi proses pergerakan material dengan batasan – batasan tertentu seperti kapasitas kendaraan sehingga dapat sampai pada titik tujuan pada kondisi, waktu dan kuantitas yang tepat. Beberapa penelitian menunjukkan *vehicle routing optimization* secara signifikan berpengaruh terhadap penghematan (diperkirakan 5%-30% oleh Hasle dan Kloster atau 5%-20% oleh Toth dan Vigo) [2].

Terdapat beberapa metode penyelesaian *Vehicle Routing Problem* yang dikelompokkan ke dalam pendekatan *exact*, heuristik dan metaheuristik. Dalam beberapa literatur, pendekatan metaheuristik banyak digunakan. Hal ini dikarenakan *vehicle routing problem* adalah sebuah permasalahan yang kompleks dan tidak direkomendasikan untuk menggunakan pendekatan *exact* pada kasus yang nyata dalam permasalahan yang besar dan kompleks dikarenakan waktu komputasional yang cukup lama. Penggunaan pendekatan *exact* dilakukan untuk

menyelesaikan permasalahan yang maksimum memiliki 25 node dan pendekatan heuristik menghasilkan solusi yang tidak dapat menjamin solusi optimal matematis [3]. Maka dari itu, solusi permasalahan yang akan digunakan adalah pendekatan metaheuristik. Hal ini dilakukan untuk menjamin solusi yang dihasilkan tetap optimal secara matematis baik jumlah *customers* yang kurang dari 25 maupun jumlah *customers* skala besar atau lebih dari 25.

Metode metaheuristik yang digunakan dalam penelitian ini adalah Algoritma *Particle Swarm Optimization* (PSO). Salah satu algoritma dibidang *swarm intelligence* yang memiliki beberapa keuntungan dibandingkan dengan metode metaheuristik lainnya yaitu kecepatan akurasi kalkulasi yang tinggi [4], mekanisme pencarian yang *robust* dan efisien dalam mempertahankan keberagaman, hanya sedikit penambahan parameter, konvergensi yang cepat menuju optimal dan telah diterapkan di berbagai penelitian seperti optimasi global, pelatihan jaringan syaraf tiruan, sistem control *fuzzy*, rekayasa optimasi desain serta logistik dan sebagainya.

Penelitian ini dilakukan di PT. XYZ merupakan salah satu perusahaan distributor dan penyedia jasa logistik untuk produk farmasi dan *consumer goods* yang melayani pendistribusian produk di Pulau Sulawesi. Model distribusi produk pada PT. XYZ dikategorikan sebagai *Heterogeneous Fleet Vehicle Routing Problem* (HFVRP). *Heterogeneous* atau *mixed fleet VRP* mempertimbangkan kelompok atau tipe dari kendaraan yang berbeda dalam hal kapasitas, *variable* dan *fixed cost*, kecepatan dan *customer* yang dapat diakses.

Saat ini, PT. XYZ belum menerapkan metode optimasi dalam pengoptimalan proses pengiriman produk ke beberapa *customers*. Proses distribusi yang tidak optimal berpotensi meningkatkan biaya dan menurunkan *service level*. Jenis produk yang didistribusikan oleh PT. XYZ memiliki karakteristik harga yang cukup rendah jika dibandingkan dengan besaran biaya baik pada *variable cost* maupun *total cost* yang dikeluarkan. Sehingga perlu dilakukan pengoptimalan pada proses pendistribusian produk untuk mengefisiensikan proses perencanaan pengiriman serta mengurangi biaya operasional distribusi yang dikeluarkan oleh perusahaan.

TINJAUAN PUSTAKA

Distribusi dan Transportasi

Distribusi adalah suatu kegiatan untuk memindahkan barang dari pihak *supplier* kepada pihak agen dalam suatu *supply chain*. Distribusi merupakan suatu kunci dari keuntungan yang akan diperoleh perusahaan karena distribusi secara langsung akan mempengaruhi biaya dari *supply chain* dan kebutuhan agen. Jaringan distribusi yang tepat dapat digunakan untuk mencapai berbagai macam tujuan dari *supply chain*, mulai dari biaya yang rendah sampai respon yang tinggi terhadap permintaan agen [5].

Transportasi berarti memindahkan sejumlah *item* dari satu lokasi ke lokasi yang lain untuk memastikan bahwa produk terjangkau oleh konsumen. Transportasi merupakan komponen biaya terbesar, pengurangan biaya transportasi memiliki pengaruh besar terhadap keseluruhan biaya logistik. Sebuah rute yang baik

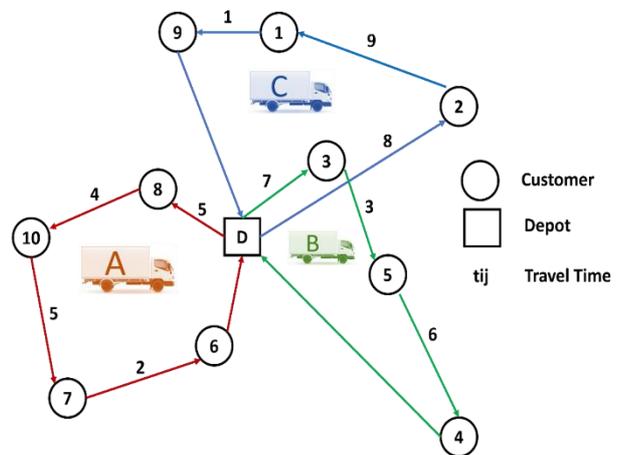
untuk distribusi logistik dapat menurunkan biaya transportasi dan meningkatkan efisiensi. Efisiensi hal tersebut berdampak pada keseluruhan peningkatan efisiensi dari sistem logistik dan menurunkan biaya transportasi. Ketika melakukan pengiriman produk, tiga hal ini harus diperhatikan yaitu perencanaan kendaraan, perencanaan rute, dan perencanaan *loading material* [1].

Vehicle Routing Problem

Dantzig dan Ramser pada tahun 1959 pertama kali memperkenalkan “*Truck Dispatching Problem*”, memodelkan bagaimana sebuah kendaraan dapat melayani permintaan untuk bahan bakar di beberapa stasiun pengisian bahan bakar dari sebuah hub pusat dan dengan jarak tempuh yang minimum. Lima tahun kemudian, Clarke dan Wright pada tahun 1964 menggeneralisasikan permasalahan ini ke dalam sebuah permasalahan linier optimasi yang biasanya digunakan pada domain logistik dan transportasi misalnya bagaimana melayani beberapa *customers*, secara geografis tersebar di sekitar depo pusat, menggunakan sebuah kendaraan dengan kapasitas. Hal ini dikenal sebagai *Vehicle Routing Problem* (VRP), salah satu topik studi yang sangat luas dalam bidang riset operasional. Sejak saat itu VRP telah banyak didiskusikan dan secara berangsur – angsur dijadikan ke dalam beberapa bentuk variasi dari VRP berdasarkan situasi praktikal distribusi [6].

Heterogeneous Fleet Vehicle Routing Problem

Bentuk dasar VRP menganggap bahwa semua kendaraan yang dimiliki mempunyai kapasitas yang sama (homogen). Padahal dalam kenyataannya, perusahaan tidak selalu mempunyai kendaraan dengan kapasitas angkut yang sama. Sebuah perusahaan baik itu perusahaan besar ataupun kecil pasti mempunyai kendaraan dengan kapasitas yang berbeda, sehingga metode penyelesaian VRP klasik sekarang ini sulit untuk diterapkan. Oleh karena itu muncul varian VRP baru dalam penyelesaian yang berbeda, varian VRP ini dikenal dengan istilah *Heterogeneous Fleet Vehicle Routing Problem* [7]. *Heterogeneous* atau *mixed fleet VRP* mempertimbangkan kelompok atau tipe dari kendaraan yang berbeda dalam hal kapasitas, *variable* dan *fixed cost*, kecepatan dan *customer* yang dapat diakses. Secara umum, *fixed cost* sangat bergantung pada apakah kendaraan dimiliki oleh perusahaan atau disewa.



Gambar 1. *Heterogeneous Fleet Vehicle Routing Problem*

Algoritma *Particle Swarm Optimization*, Algoritma Genetika dan *Memetic Algorithm* digunakan oleh Abdallah dan Ennigrou [8]

untuk menyelesaikan varian HFVRP yaitu Multi-depot Heterogeneous Fleet Vehicle Routing Problem with Time Windows (MDHFVRPTW). Ketiga Algoritma tersebut dibandingkan kinerjanya dari segi solusi dan *computational performance*. Hasil penelitian menunjukkan bahwa Algoritma *Particle Swarm Optimization* memiliki *CPU times* terbaik dibandingkan GA dan ME. Selain itu, solusi yang dihasilkan lebih baik untuk ukuran permasalahan yang berbeda-beda.

Implementasi PSO dalam penyelesaian berbagai macam jenis VRP dilakukan oleh Rabbani [9] pada penyelesaian *Periodic VRP with Pickup and Delivery*, Balmecheri [10] pada penyelesaian *Heterogeneous Fleet Vehicle Routing Problem with Mixed Backhaul and Time Windows*, Sun [6] pada penyelesaian *Heterogeneous Fleet VRP with 2-Dimensional Loading Constraint*, Alinezhad [11] dan Iswari [12] pada penyelesaian *Capacitated Vehicle Routing Problem*, dan Abdallah [8] pada penyelesaian *Muti Depot Heterogeneous Fleet VRP with Time Windows*.

Penyelesaian HFVRP dengan memperhitungkan *fixed cost* untuk meminimasi total biaya distribusi dilakukan oleh Karagul dan Gungor [13] menggunakan *Random Search Algorithm* dan Algoritma Genetika, Penna dan Ochi [14] menggunakan *Iterated Local Search Heuristics*, S.W. Lai dan C. Demirag [15] menggunakan *Tabu Search* dan Vincent et al [16] menggunakan *Simulated Annealing*.

Formulasi Model Matematis

Fungsi Tujuan:

Formulasi model matematis *Heterogeneous Fleet Vehicle Routing Problem* dinyatakan sebagai berikut merujuk pada model matematis yang dikembangkan oleh [17]:

$$MIN TC = \sum_{k=1}^T f_k z_k + \sum_{k=1}^T \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k \tag{1}$$

Fungsi Pembatas:

1. Pembatas kunjungan semua lokasi

Pembatas kunjungan lokasi memastikan bahwa setiap pelanggan hanya dilayani satu kali oleh kendaraan *k*.

$$\sum_{k=1}^K y_{ik} = 1, \quad \forall i \in V \setminus \{0\} \tag{2}$$

2. Pembatas penggunaan kendaraan

Pada pembatas ini model memastikan bahwa aktivasi kendaraan *k* yang keluar dari depot tidak boleh melebihi jumlah kendaraan yang tersedia sebanyak *K*.

$$\sum_{k=1}^K y_{0k} \leq K, \tag{3}$$

3. Pembatas pelayanan pelanggan

Pada pembatas ini model memastikan bahwa kendaraan *k* akan melayani pelanggan *i* kemudian meninggalkannya untuk menuju pelanggan *j* (tidak termasuk depot).

$$\sum_{j \in V \setminus \{i\}} x_{ij}^k = y_{ik}, \quad \forall i \in V, k = 1, \dots, k \tag{4}$$

$$\sum_{j \in V \setminus \{i\}} x_{ji}^k = y_{ik}, \quad \forall i \in V, k = 1, \dots, k \tag{5}$$

4. Pembatas kapasitas kendaraan

Pada pembatas ini model memastikan bahwa kendaraan *k* akan mengantarkan produk sesuai jumlah permintaan. Beban angkut pengantaran tidak boleh melebihi kapasitas kendaraan *U_k*.

$$\sum_{i \in V} d_i y_{ik} \leq U_k z_k \quad k = 1, \dots, k \tag{6}$$

5. Pembatas eliminasi *sub-tour*

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij}^k \leq |S| - 1 \quad \forall S \subseteq V \setminus \{0\}, \tag{7}$$

$$|S| \geq 2, k = 1, \dots, k$$

6. Pembatas integralitas variabel *x*, *y*, dan *z*

Pembatas variabel keputusan tidak boleh negatif dan hanya bernilai 0 atau 1. Nilai 1 artinya kendaraan melakukan perjalanan pada busur (*i,j*), sedangkan akan bernilai 0 jika kendaraan tidak melakukan perjalanan.

$$x_{ij}^k \in \{0,1\}, \quad \forall i, j = 1,2, \dots, n \quad \forall k = 1,2, \dots, T \tag{8}$$

$$y_k \in \{0,1\}, \quad \forall k = 1,2, \dots, T \tag{9}$$

$$z_k \in \{0,1\}, \quad \forall k = 1,2, \dots, T \tag{10}$$

Particle Swarm Optimization

Particle Swarm Optimization (PSO) didasarkan pada perilaku sebuah kawanan serangga, seperti semut, rayap, lebah atau burung. Algoritma PSO meniru perilaku sosial organisme ini. Perilaku sosial terdiri atas tindakan individu dan pengaruh dari individu – individu lain dalam suatu kelompok. Kata “partikel” menunjukkan individu, misalnya seekor burung dalam kawanan burung. Setiap individu atau partikel berperilaku saling terhubung dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan juga dipengaruhi perilaku kelompok kolektifnya. Dengan demikian, jika satu partikel atau seekor burung menemukan jalan yang tepat atau pendek menuju ke sumber makanan, sisa kelompok yang lain juga dapat segera mengikuti jalan tersebut, meskipun lokasi mereka jauh di kelompok tersebut. Metode optimasi yang didasarkan pada *swarm intelligence* ini disebut algoritma *behaviorally inspired* sebagai alternatif dari algoritma genetika yang sering disebut *evolution-based procedures* [18].

Meskipun PSO telah digunakan untuk optimasi selama hampir dua dekade, namun periode ini dikatakan relatif singkat jika dibandingkan dengan metode lain seperti *Artificial Neural Network* (ANN), Algoritma Genetika (GA) atau *Ant Colony Optimization* (ACO) karena keuntungan dari PSO. Keuntungan PSO antara lain yaitu mekanisme pencarian yang *robust* dan efisien dalam mempertahankan keberagaman, hanya sedikit penambahan parameter konvergensi yang cepat menuju optimal, kemudian dalam *encoding* dan *decoding* mudah dalam perhitungan dan telah diterapkan di berbagai penelitian seperti optimasi global, pelatihan jaringan syaraf tiruan, sistem control *fuzzy*, rekayasa optimasi desain serta logistik dan sebagainya [19].

Pada PSO, solusi didapatkan dari pencarian *random* dengan kecerdasan *swarm*. Dalam artian lain PSO merupakan sebuah algoritma *swarm intelligent search*. Pencarian ini dilakukan dengan menginisiasi kumpulan solusi yang potensial yang disebut *swarm* dan solusi potensial individual yang disebut *particle*.

Pencarian dipengaruhi oleh dua tipe pembelajaran oleh partikel. Setiap partikel belajar dari partikel-partikel yang lainnya dan juga belajar dari pengalamannya sendiri selama pergerakan.

Pembelajaran dari partikel lainnya disebut *social learning* sedangkan pembelajaran dari pengalaman sendiri disebut *cognitive learning*. Sebagai hasilnya dari *social learning*, partikel menyimpan solusi terbaik yang disebut g_{best} dan hasil dari *cognitive learning*, partikel menyimpan solusi terbaik yang disebut p_{best} . Perubahan arah dan *magnitude* dalam setiap partikel akan diputuskan berdasarkan sebuah faktor yang disebut *velocity*. Dalam hal ini, pada PSO, *velocity* didefinisikan sebagai tingkat perubahan dalam posisi yang berdasarkan iterasinya.

Dalam algoritma PSO partikel merupakan solusi potensial untuk masalah ini, dan kawanan terdiri dari partikel i . setiap partikel i dapat direpresentasikan melalui vektor D -dimensi: vektor pertama didefinisikan sebagai $X_i^t = (X_1^t, X_2^t, \dots, X_n^t)$ dengan $i = (1, 2, \dots, n)$ yang menunjukkan posisi partikel i dalam ruang pencarian di iterasi t . Vektor kedua dituliskan $V_i^t = (V_1^t, V_2^t, \dots, V_n^t)$ yang merepresentasikan kecepatan dengan partikel i bergerak. Vektor ketiga dituliskan $P_{best_i}^t = (P_{best_1}^t, P_{best_2}^t, \dots, P_{best_m}^t)$ menunjukkan posisi terbaik dari partikel ke- i dan vektor terakhir dituliskan $G_{best_i}^t = (g_{best_1}^t, g_{best_2}^t, \dots, g_{best_m}^t)$ yang merepresentasikan posisi terbaik secara global dalam kawanan sampai iterasi ke- t .

Kecepatan konvergensi dan kemampuan untuk mencari solusi optimal dari setiap algoritma yang berbasis populasi dipengaruhi oleh pemilihan parameter. Biasanya, sebuah rekomendasi secara umum untuk penyetelan parameter untuk algoritma ini tidak memungkinkan. Hal ini dikarenakan berdasarkan jenis permasalahannya. Salah satu parameter yang biasanya di set secara empiris berdasarkan jumlah dari variabel dalam sebuah permasalahan dan kompleksitas permasalahan. Namun, secara umum 20 – 50 partikel direkomendasikan [20]. Prathama [21] dan El-Shorbagy [22] berpendapat bahwa parameter yang harus dipertimbangkan adalah jumlah swarm dan koefisien akselerasi.

Modifikasi Particle Swarm Optimization

Dalam implementasinya, ditemukan bahwa kecepatan partikel dalam PSO *standard* diperbaharui terlalu cepat dan nilai minimum fungsi tujuan yang dicari sering terlewati. Karena itu kemudian dilakukan modifikasi atau perbaikan terhadap algoritma PSO *standard*. Perbaikan itu berupa penambahan koefisien berat inersia untuk mengurangi kecepatan pada formula memperbaharui kecepatan. Biasanya nilai koefisien berat inersia dibuat sedemikian hingga semakin besar iterasi yang dilalui, semakin mengecil kecepatan partikel. Nilai ini bervariasi secara linier dalam rentang 0.9 hingga 0.4. secara matematis perbaikan ini bisa dituliskan sebagai berikut:

$$V_i^{(t+1)} = \omega V_i^t + c_1 r_1 (P_{best_i}^t - X_i^t) + c_2 r_2 (G_{best_i}^t - X_i^t) \quad (11)$$

$$X_i^{(t+1)} = X_i^t + V_i^{(t+1)} \quad (12)$$

Dimana, c_1 dan c_2 adalah parameter, r_1 dan r_2 adalah bilangan acak yang berdistribusi seragam dalam selang $[0,1]$. c_1 dan c_2 bernilai konstan, disebut sebagai parameter *cognitive* dan *social scalling*, yang biasa juga disebut sebagai koefisien akselerasi.

Koefisien berat inersia diusulkan oleh Shi dan Eberhart pada tahun 1998 untuk meredam kecepatan selama iterasi, yang memungkinkan kawanan burung menuju (konvergen) titik target secara lebih akurat dan efisien dibandingkan dengan algoritma aslinya. Koefisien berat inersia yang tinggi menambah porsi

pencarian global (*global exploration*), sedangkan nilai yang rendah lebih menekankan pencarian lokal (*local search*). Untuk tidak terlalu menitikberatkan pada salah satu bagan dan tetap mencari area pencarian yang baru dalam ruang berdimensi tertentu, maka perlu dicari nilai koefisien berat inersia yang secara imbang menjaga pencarian global dan lokal. Untuk mencapai hal tersebut dan mempercepat konvergensi, suatu koefisien berat inersia yang mengecil nilainya dengan bertambahnya iterasi digunakan formula sebagai berikut:

$$\omega = \omega_{max} - \text{iterasi} \times \frac{\omega_{max} - \omega_{min}}{\text{Maksimum Iterasi}} \quad (13)$$

Dimana ω_{max} dan ω_{min} masing – masing adalah nilai awal dan nilai akhir koefisien berat inersia [23].

Particle Coding dan Encoding

Solusi dari sebuah partikel memiliki n (jumlah *customer*) vektor dimensional disebut dimensi dimana urutan dari setiap nilai elemen mewakili pengiriman atau penjemputan pesanan ke titik *customer* dari keseluruhan rute. Contohnya, kita berasumsi 6 *customer* dan 2 kendaraan. Kita juga berasumsi bahwa satu partikel i (pada iterasi t) memiliki nilai elemen – elemen dari vektor posisi seperti berikut [10]:

$$X_i^t = [3.6, 3.3, 2.1, 2.5, 1.5, 4.2]$$

Untuk membangun solusi, dua langkah ini diaplikasikan:

- Langkah pertama: seluruh nilai elemen harus diurutkan (dari terkecil ke terbesar) seperti: 1.5, 2.1, 2.5, 3.3, 3.6, 4.2. kemudian indeks seluruh *customer* sesuai dari nilai elemennya seperti urutan sebelumnya seperti: 5 3 4 2 1 6.

Customer	1	2	3	4	5	6
Coding	3.6	3.3	2.1	2.5	1.5	4.2

Encoding (customer's sequence)	5	3	4	2	1	6
---------------------------------------	----------	----------	----------	----------	----------	----------

Gambar 2. Contoh Code/Encode Partikel

- Langkah kedua: masukkan setiap *customer* ke kendaraan satu demi satu dan perhatikan visibilitas setiap menugaskan setiap *customer* ke dalam kendaraan. Berdasarkan contoh sebelumnya, *customer* 5 dan 3 dapat dilayani oleh kendaraan 1 jika batasan (*time windows*, tipe *customers*, kapasitas kendaraan) terpenuhi. Kemudian dilanjutkan *customer* selanjutnya (4, 2, 1, 6) akan dilayani oleh kendaraan 2. Jika diasumsikan bahwa 0 adalah depo, maka solusi sebagai berikut:

Kendaraan 1				Kendaraan 2					
0	5	3	0	0	4	2	1	6	0

Gambar 3. Urutan Rute

Parameter Tuning

Setiap metaheuristik memiliki kumpulan parameter yang ditentukan di awal sebelum dieksekusi. Adaptasi metaheuristik mensyaratkan kalibrasi dari beberapa parameter. Aktivitas ini disebut *parameter tuning*. Sebuah penyetelan awal parameter yang tepat berdampak signifikan dalam memecahkan permasalahan. Beberapa metode yang dapat digunakan untuk menentukan kumpulan parameter yaitu *Latin-Square*, *Taguchi Orthogonal Arrays*, *Latin Hypercube*, *Calibra*, dan *Design of Experiment* [24]. Proses penyetelan parameter mengakselerasi proses pembuatan keputusan dan digunakan untuk meningkatkan

kualitas dari pembuatan keputusan, dengan mengizinkan permasalahan *real-life* untuk diselesaikan secara efisien, tepat dan *real-time* [25].

METODE

Penelitian ini mengkaji mengenai rute pendistribusian produk pada PT. XYZ dari depot ke beberapa pelanggan yang menjadi area cakupan pemasaran dan pengiriman. Karakteristik sistem distribusi PT. XYZ adalah pendistribusian *multi-product* dengan menggunakan jenis kendaraan yang berbeda – beda dalam segi kapasitas kendaraan maupun biaya yang dikeluarkan oleh tiap kendaraan. Sehingga diketahui bahwa varian VRP dalam objek penelitian ini adalah *Heterogeneous Fleet Vehicle Routing Problem* (HFVRP) dengan fungsi tujuan untuk meminimasi *total cost* distribusi. Formulasi model matematis dapat dilihat pada persamaan (1-10) pada penelitian yang dilakukan oleh Saputra et al [17].

Penyelesaian *Heterogeneous Fleet Vehicle Routing Problem* (HFVRP) pada studi kasus PT. XYZ menggunakan pendekatan metaheuristik. Pemilihan pendekatan metaheuristik merujuk pada penelitian yang dilakukan oleh Abdallah et al [8] dan Balmecheri [10] yang meneliti varian HFVRP dengan hasil yang menunjukkan bahwa PSO memberikan solusi yang baik dengan waktu komputasi yang *feasible*.

Berikut tahapan - tahapan pengolahan data pada penelitian ini:

1. Tahap pertama yaitu pembuatan program dengan Algoritma *Particle Swarm Optimization* menggunakan *software* MATLAB R2017b dengan memperhatikan batasan – batasan sesuai dengan formulasi matematis. Selanjutnya program dilakukan validasi dengan melihat apakah program dapat dijalankan dan verifikasi dengan melihat apakah solusi yang dihasilkan tidak melanggar fungsi pembatas.

```

Pseudocode Penugasan Kendaraan
1:  For setiap kendaraan k do
2:      Kalkulasi berat produk tiap customer i yang dilayani oleh kendaraan k
3:      Kalkulasi waktu pelayanan tiap customer i yang dilayani oleh kendaraan k
4:      if total berat produk kendaraan k > kapasitas kendaraan k dan total waktu
        pelayanan kendaraan k > total jam operasional do
5:          k = k + 1 // tambah sebuah kendaraan baru k
6:          tambahkan customer i ke kendaraan baru
7:      End If
8:      Kalkulasi jarak tempuh tiap kendaraan k
9:  End For
    
```

Gambar 4. *Pseudocode* Penugasan Kendaraan

2. Tahap kedua yaitu melakukan penyetelan parameter (*parameter tuning*) menggunakan *design of experiment* untuk kombinasi parameter yang menghasilkan kualitas yang baik pada solusi/*output* dari algoritma PSO. Parameter yang diuji adalah jumlah swarm dan koefisien akselerasi yang terdiri atas *cognitive learning* (C1) dan *social learning* (C2). Selanjutnya dilakukan analisis untuk melihat pengaruh parameter terhadap *output* yang dihasilkan.

```

Pseudocode PSO
1:  Inisialisasi secara acak posisi dan kecepatan awal partikel
2:  For s = 1:n_swarm do // n_swarm yaitu total swarm
3:      Penugasan Kendaraan
4:      Kalkulasi Fungsi tujuan tiap s
5:      Kalkulasi Pbest dan Gbest
6:  End For
7:  While iterasi (t) < iterasi maksimal (t maks) do
8:      inisialisasi r1 dan r2
9:      Perbaharui bobot inersia dengan formula (13)
10:     For s = 1:n_swarm do
11:         Perbaharui Kecepatan dengan formula (11)
12:         Perbaharui Posisi dengan formula (12)
13:         Penugasan Kendaraan
14:         Kalkulasi fungsi tujuan tiap s
15:         If Pbest t > Pbest (t+1) do
16:             Pbest = Pbest (t+1)
17:         End If
18:         Gbest(t+1) = nilai terkecil dari Pbest
19:         If Gbest > Gbest (t+1) do
20:             Gbest = Gbest (t+1)
21:         End If
22:     End For
23: End While
    
```

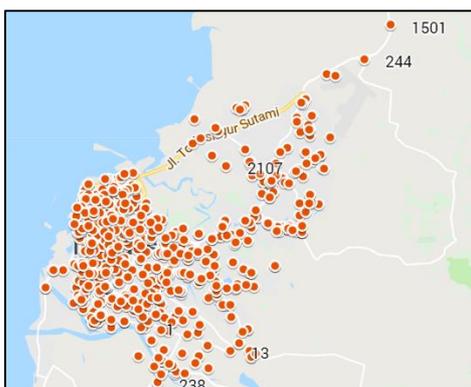
Gambar 5. *Pseudocode* PSO

3. Tahap Ketiga yaitu implementasi PSO pada data set yang didapatkan dari PT. XYZ untuk pendistribusian produk pada Bulan Maret 2019 yang terdiri atas Data Permintaan, Data *Customers*, dan Data Kendaraan serta Biaya – Biaya. Implementasi PSO pada studi kasus PT. XYZ akan dianalisa performansinya dengan merujuk pada *CPU time* yang dihasilkan dan karakteristik konvergensi.
4. Tahap Keempat yaitu melakukan analisa terhadap hasil komputasi. Pertama, analisa terhadap kualitas solusi yang dihasilkan dengan menghitung koefisien variansi untuk melihat seberapa besar variansi solusi yang dihasilkan. Kedua, perbandingan terhadap total jarak dan total biaya pada rute *existing* perusahaan dan rute yang dihasilkan oleh Algoritma *Particle Swarm Optimization*. Ketiga, analisis sensitivitas terhadap perubahan parameter *variable cost* dan kapasitas kendaraan terhadap *total cost*.

HASIL DAN PEMBAHASAN

Gambaran Umum Sistem Distribusi

Karakteristik sistem distribusi PT. XYZ adalah pendistribusian *multi-product* dengan menggunakan jenis kendaraan yang berbeda–beda dalam segi kapasitas kendaraan maupun biaya yang dikeluarkan oleh tiap kendaraan. Sistem pendistribusian produk dimulai saat *customer* melakukan pemesanan dan memperoleh faktur ID, faktur ID kemudian digabungkan untuk dilakukan *sending preparation*. Pada tahap ini, bagian terkait melakukan perencanaan pengiriman yang terdiri atas *picklist ID*, pemilihan kendaraan, daftar alamat *customer*. Gambar 6 menunjukkan peta persebaran *customers* pada Bulan Maret 2019.



Gambar 6. Peta Pesebaran *Customers* Bulan Maret 2019 Cabang Makassar

Saat ini, PT. XYZ memiliki 4 jenis kendaraan yaitu tipe A terdiri atas 3 unit yang berkapasitas 2 m³, tipe B terdiri atas 3 unit yang berkapasitas 2.5 m³, Tipe C terdiri atas 2 unit yang berkapasitas 8 m³, dan tipe D yang terdiri atas tipe D1 sejumlah 2 unit yang berkapasitas 9 m³, tipe D2 sejumlah 1 unit yang berkapasitas 12 m³, tipe D3 sejumlah 2 unit yang berkapasitas 13 m³, dan tipe D4 sejumlah 1 unit yang berkapasitas 14 m³.

Komponen penyusun biaya distribusi yang diperhitungkan dalam penelitian ini yakni *Fixed Cost* dan *Variabel Cost*. *Fixed Cost* yaitu biaya yang dipengaruhi oleh penggunaan kendaraan per hari terdiri atas biaya perawatan, biaya administrasi, dan pajak/retribusi. *Variable Cost* yaitu biaya yang dipengaruhi oleh jarak tempuh kendaraan terdiri atas biaya bahan bakar. Berikut ini data biaya – biaya distribusi:

Tabel 1. Biaya - Biaya Berdasarkan Jenis Kendaraan

Jenis Kendaraan	Kode	Fixed Cost (Rp)	Variable Cost (Rp)
VIAR	V	24059,28	1308,33
CARRY	C	51157,3	1308,33
TRUK ANGKEL (4 RODA)	TA_1	86768,26	858,33
TRUK 6 RODA	TA_2	87343,61	858,33
KECIL	TA_3	87343,61	858,33
	TA_4	88384,7	858,33
	TA_5	90439,5	858,33

Parameter Tuning

Setiap metaheuristik memiliki kumpulan parameter yang ditentukan di awal sebelum dieksekusi. Adaptasi metaheuristik mensyaratkan kalibrasi dari beberapa parameter. Aktivitas ini disebut *parameter tuning*. Sebuah penyetelan awal parameter yang tepat memiliki dampak yang signifikan dalam memecahkan permasalahan [24]. Prathama [21] dan El-Shorbagy [22] berpendapat bawah parameter yang harus dipertimbangkan adalah jumlah swarm dan koefisien akselerasi. Dalam menentukan kombinasi yang tepat dibutuhkan studi parameter untuk 3 parameter sebagai faktor percobaan dengan membandingkan rata-rata hasil akhir dengan menggunakan *design of experiment*.

Analisa parameter bertujuan untuk mengetahui efek dan interaksi antar parameter apakah mempengaruhi hasil secara signifikan atau tidak dengan menggunakan *Analysis of Variance*. Tingkat kepercayaan yang digunakan adalah sebesar 95% yang berarti nilai α (*level of significant*) adalah sebesar 0,05. level pada faktor

pada studi parameter ini dapat dilihat pada Tabel 2. Jumlah kombinasi parameter yang diuji sebesar $3 \times 3 \times 3 = 27$ kombinasi parameter dengan 10 replikasi untuk setiap kombinasi parameter.

Tabel 2. Level *Design of Experiment* Penentuan Parameter PSO

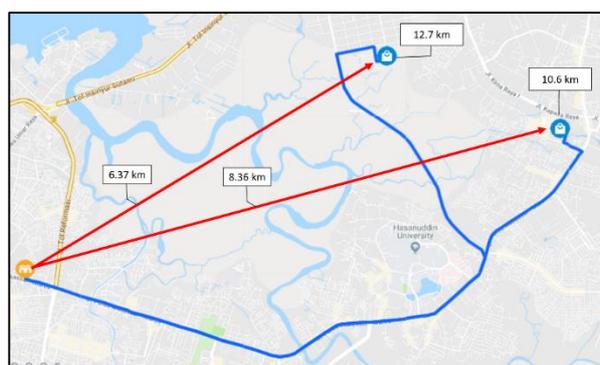
Parameter	Level		
	Low	Medium	High
Jumlah Swarm	30	40	50
<i>Cognitive Learning</i> (C1)	1,5	1,7	2
<i>Social Learning</i> (C2)	1,5	1,7	2

Dari hasil pengujian beberapa parameter maka didapatkan parameter yang optimal adalah 50 *swarm*, 1,5 *cognitive learning* C1, 2 *social learning* C2. Pada *Analysis of Variance* (ANOVA), nilai *P-value* tiap faktor berfungsi untuk mengetahui apakah perubahan level dari faktor tersebut mempunyai pengaruh yang signifikan terhadap hasil output. Apabila angka *P-value* kurang dari 0,05 maka faktor tersebut mempunyai pengaruh yang signifikan terhadap *output* yang dihasilkan. *P-value* dari faktor *Swarm* adalah 0,038, C1 adalah 0,472, dan C2 adalah 0,838. Maka dapat terlihat bawah hanya faktor jumlah *swarm* yang berpengaruh signifikan terhadap *output*.

Nilai *P-value* pada interaksi faktor *Swarm* dengan C1 adalah 0,652, *Swarm* dengan C2 adalah 0,275, dan C1 dengan C2 adalah 0,980. Interaksi dari ketiga faktor adalah 0,669. Dari ketiga interaksi tersebut tidak terdapat pengaruh yang signifikan terhadap *output*. Hasil analisis menunjukkan penyetelan parameter untuk parameter jumlah *swarm* perlu diperhatikan karena mempunyai pengaruh yang signifikan terhadap *output*.

Implementasi Particle Swarm Optimization

Model penyelesaian *vehicle routing problem* menggunakan Algoritma *Particle Swarm Optimization* dibuat dengan membuat program pada MATLAB R2017b. Input data program yaitu parameter algoritma PSO, matriks jarak, data permintaan, data kapasitas kendaraan, data kecepatan kendaraan, waktu operasional perusahaan, waktu standar *service time* di tiap *customers*, *fixed cost* dan *variable cost*. Data jarak menggunakan data yang diperoleh dari *Google Maps* dengan input koordinat dari alamat *customers*. Pengukuran data jarak yang diperoleh dari data *google maps* adalah untuk mendapatkan jarak yang sesuai dengan jarak jalan yang dilalui. Pada Gambar 7 menunjukkan perbedaan penggunaan data jarak jika menggunakan perhitungan jarak menggunakan metode *Euclidean* dengan menggunakan jarak yang sesuai dengan jalan yang dilalui.



Gambar 7. Contoh Pengambilan Data Jarak

Data permintaan dan pengiriman produk terlebih dahulu dikonversi ke dalam satuan m³. Hal ini dilakukan untuk efisiensi perhitungan karena karakteristik produk yaitu *multi-product* artinya lebih dari satu jenis. Kecepatan kendaraan dalam perhitungan diasumsikan sama yaitu 33 km/jam. Data rata – rata kecepatan kendaraan didapatkan dari survei yang dilakukan Dinas Pengelolaan Lingkungan Hidup Provinsi Sulawesi Selatan oleh Hidayat [26]. Waktu standar *service time* yaitu 30 menit yang untuk setiap *customers*. Tabel 3 menunjukkan input parameter PSO pada program yang telah dibuat.

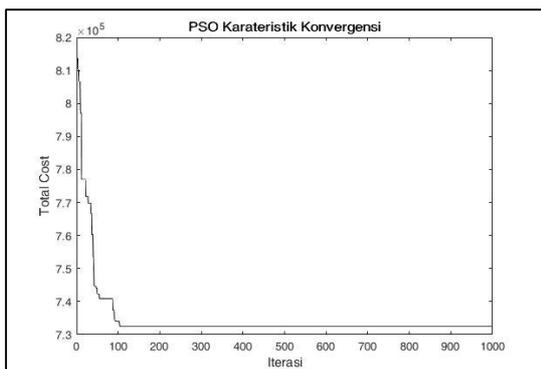
Tabel 3. Input Parameter PSO

Parameter	Nilai
Jumlah <i>Swarm</i>	50
<i>Cognitive Learning</i> (c1)	1,5
<i>Social Learning</i> (c2)	2
<i>Wmax</i>	0,9
<i>Wmin</i>	0,4
Iterasi Maksimal	1000

Hasil Komputasi

Penilaian terhadap performansi dari model yang dibangun dengan algoritma *Particle Swarm Optimization* dapat terlihat dari lama waktu komputasi (lama waktu yang dibutuhkan untuk memproses *code*). Pada Tabel 4 menunjukkan waktu komputasi (*CPU time*) dalam satuan detik. Pembuatan program dibuat pada MATLAB R2017b menggunakan Laptop Lenovo G40 dengan spesifikasi *processor* AMD A8 2.20 GHz, RAM 4 GB dan sistem operasi Windows 10 64-bit. Pada Tabel 4 terlihat bahwa jumlah *customers* mempengaruhi waktu komputasi. Semakin banyak jumlah *customers* atau titik pengantaran maka membutuhkan lebih banyak waktu komputasi dan sebaliknya. Waktu komputasi yang dihasilkan dalam model yang dibuat dengan penyelesaian menggunakan algoritma *Particle Swarm Optimization* pada data set Bulan Maret 2019 yakni dengan rata–rata sebesar 40,014 detik pada *range* titik pengiriman/jumlah *customers* 45–194.

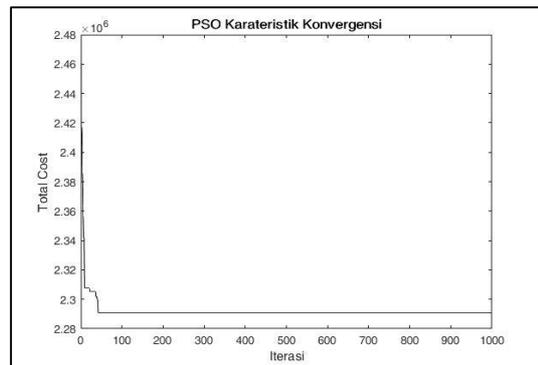
Grafik karakteristik pada gambar 8, 9 dan 10 menunjukkan pergerakan pencarian solusi optimal dari iterasi 1 hingga iterasi maksimal untuk pengiriman pada tanggal 1 Maret 2019 pada 57 *customer*, tanggal 5 Maret 2019 pada 194 *customer*, tanggal 16 Maret 2019 pada 110 *customers*. Grafik tersebut menunjukkan hasil evaluasi terhadap fungsi objektif tiap iterasi. Dimana kondisi konvergen yakni kondisi solusi mencapai titik optimal dalam hal ini minimasi terjadi pada iterasi yang mendekati maupun melebihi iterasi ke 100 dengan rata – rata waktu komputasi (*CPU time*) selama 40,041 detik.



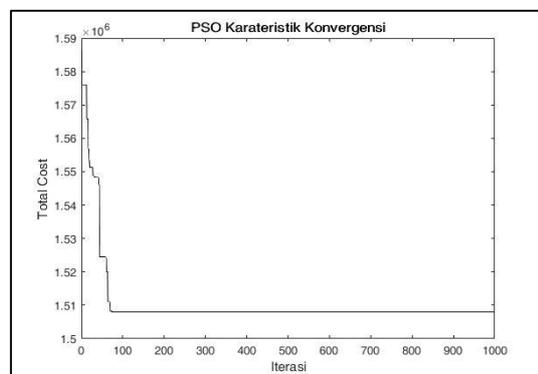
Gambar 8 Grafik Karakteristik Konvergensi 1 Maret 2019

Tabel 4. *Output* dan *Computational Time*

Tgl.	Jumlah <i>Customers</i>	Jarak total (km)	Total Cost (Rp)	<i>CPU Time</i> (s)
1	57	338,48	732426,49	28,416
2	59	433,61	901419,92	30,139
4	165	1245,30	2183856,01	82,510
5	194	1251,20	2290931,72	91,651
6	66	396,65	869696,77	30,493
8	98	696,66	1300740,17	42,429
9	87	572,00	1106976,36	38,844
11	120	905,90	817188,55	52,160
12	53	296,64	696513,83	24,355
13	45	330,15	637931,25	19,872
14	75	583,33	1116695,27	32,356
15	77	557,72	1094715,93	33,520
16	110	784,57	1508015,47	52,306
18	109	856,53	1524733,61	46,139
19	121	784,94	1512093,95	51,176
20	59	355,60	747119,44	25,723
21	67	404,20	876178,05	30,044
22	93	751,19	1434317,64	40,932
23	63	462,45	926175,97	28,382
25	124	856,45	1615927,70	53,747
26	74	453,06	918116,22	31,568
27	84	476,21	1024755,75	35,567
28	89	645,07	1256457,03	37,389
29	73	483,85	944547,73	31,091
30	67	388,65	862829,25	29,535



Gambar 9. Grafik Karakteristik Konvergensi 5 Maret 2019



Gambar 10. Grafik Karakteristik Konvergensi 16 Maret 2019

Grafik Karakteristik Konvergensi baik pada tanggal 1 Maret 2019 (57 *customer*), 5 Maret 2019 (194 *customers*) maupun 16 Maret 2019 (109 *customers*) dengan melihat jumlah *customers* yang dilayani pada tanggal tersebut mulai dari jumlah *customers* yang

sedikit, sedang hingga banyak menunjukkan bahwa model yang telah dibuat pada *Heterogeneous Fleet Vehicle Routing Problem* dengan menggunakan algoritma *Particle Swarm Optimization* (PSO) telah *robust*. Hal ini terlihat pada grafik menunjukkan gambaran *output* yang tidak terdapat perubahan secara signifikan saat mencapai titik konvergen.

Dari grafik-grafik tersebut menunjukkan jumlah titik pengiriman tidak mempengaruhi kecepatan dalam mencapai kondisi konvergen hal ini terlihat pada grafik tanggal 1 Maret 2019 dengan 57 *customers* kondisi konvergen dicapai setelah iterasi ke 100, pada grafik tanggal 5 Maret 2019 dengan 194 *customers* kondisi konvergen dicapai sebelum iterasi ke 100 dan pada grafik tanggal 16 Maret 2019 dengan 110 *customers* kondisi konvergen dicapai sebelum iterasi ke 100.

Analisa Kestabilan Solusi

Analisa kestabilan solusi dilakukan dengan tujuan untuk melihat variasi atau konsistensi kinerja dari solusi yang dihasilkan. Hal ini dikarenakan metaheuristik menghasilkan solusi yang bersifat probabilistik sehingga pengujian terhadap kestabilan dianggap perlu untuk mengetahui keseragaman solusi. Analisa kestabilan solusi dalam penelitian ini dengan menghitung nilai variansi dari 10 replikasi yang dilakukan. Data yang digunakan menggunakan data pada tanggal 1 Maret 2019.

Tabel 5 menunjukkan hasil perhitungan koefisien variansi. Terlihat bahwa nilai koefisien variansi yang dihasilkan baik pada waktu komputasi, jarak dan *total cost* mendekati nol, sehingga dapat disimpulkan bahwa solusi yang dihasilkan dari percobaan dapat dikatakan stabil.

Tabel 5. Analisa Kestabilan Solusi

Replikasi	Waktu Komputasi (detik)	Jarak (km)	Total Cost (Rp)
1	26,18	303,67	702541,90
2	26,36	320,48	716975,64
3	25,76	318,54	715313,04
4	25,40	327,02	722589,99
5	31,23	303,95	702786,53
6	30,14	315,25	712482,26
7	28,33	308,68	706848,16
8	28,79	323,09	719217,60
9	29,65	304,58	703323,84
10	25,54	315,81	712968,08
Rata-rata	27,82	314,66	711976,43
Simpangan Baku	2,12	8,08	6932,80
Koefisien Variansi	0,0761	0,0256	0,0097

Analisa Perbandingan Rute Existing dan Rute Solusi

Aktivitas pengiriman produk pada PT. XYZ merupakan salah satu aktivitas utama yang dilakukan. Hal ini merupakan tugas utama PT. XYZ sebagai distributor. Sistem pendistribusian produk *existing* dilakukan dengan sistem pengiriman per regional yang telah ditentukan oleh perusahaan. Oleh sebab itu, pengiriman produk dari *depo* ke *customer* sering kali terjadi *multi trip* untuk kendaraan yang sama dan tidak memaksimalkan kapasitas kendaraan. Tabel 6 menunjukkan rute *existing* perusahaan.

Tabel 6. Rute Existing

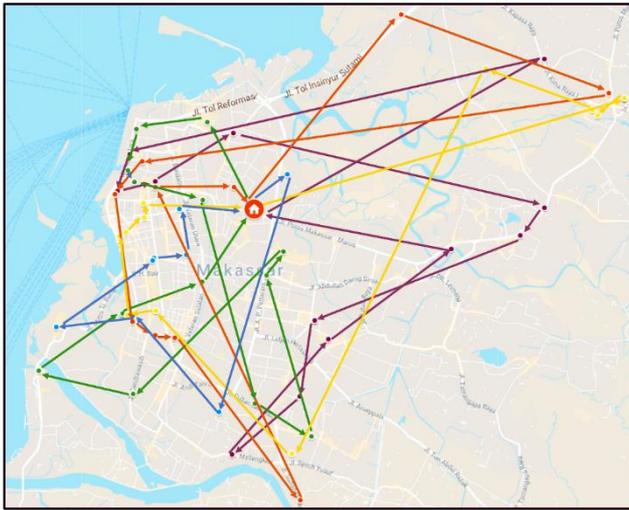
Kode	Jumlah Customer	Rute
TA_5	2	DU-2185-2487-DU
TA_5	6	DU-209-14777-208-2493-50-2500-DU
TA_5	1	DU-2092-DU
TA_4	1	DU-2481-DU
TA_4	2	DU-2093-39-DU
TA_4	3	DU-2101-740-2100-DU
TA_4	2	DU-230-284-DU
TA_4	2	DU-2092-553-DU
TA_4	1	DU-2092-DU
TA_4	2	DU-230-2487-DU
TA_4	3	DU-1175-2218-2469-DU
TA_4	2	DU-2185-262-DU
TA_4	2	DU-2098-39-DU
TA_3	8	DU-21547-2175-1626-14777-2107-2494-740-1429-DU
TA_3	2	DU-2093-999-DU
TA_2	1	DU-667-DU
TA_2	1	DU-2195-DU
TA_2	1	DU-2607-DU
TA_2	1	DU-2469-DU
TA_2	7	DU-2871-607-604-2071-610-49-2239-DU
TA_2	5	DU-741-501-2239-500-49-DU
TA_2	3	DU-444-312-744-DU
TA_2	2	DU-2487-1533-DU
TA_1	2	DU-2172-2388-DU
TA_1	2	DU-2388-2376-DU
TA_1	1	DU-32-778-DU
TA_1	2	DU-2385-2402-DU
TA_1	2	DU-2175-2093-DU
TA_1	2	DU-1965-151-DU
TA_1	2	DU-109-1965-DU

Tabel 7 menunjukkan rute solusi yang dihasilkan oleh Algoritma *Particle Swarm Optimization* dalam melakukan pengoptimalan rute pengiriman sehingga jarak tempuh dan biaya distribusi yang dikeluarkan oleh perusahaan dapat diminimasi.

Tabel 7. Rute Solusi

Kode	Jumlah Customer	Rute
TA_5	12	DU-553-607-610-999-2092-2098-2101-2494-2385-500-2093-2481-DU
TA_4	11	DU-109-2500-209-501-14777-2185-2100-230-2239-740-151-DU
TA_4	12	DU-500-1429-2107-1175-1533-1965-2175-1626-2607-2469-444-DU
TA_3	13	DU-2871-2376-604-262-2218-2172-39-49-208-284-312-667-741-DU
TA_2	9	DU-744-778-2071-2195-2388-2402-2487-2493-21547-DU

Penelitian pada studi kasus ini ditemukan bahwa pendekatan *route first cluster second* menggunakan algoritma *Particle Swarm Optimization* tidak menghasilkan pembagian *cluster* yang cukup baik dilihat pada gambar 11. Hal ini dapat terlihat dari titik pengantaran terdekat sering terlewatkan yang kemungkinan disebabkan oleh pada studi kasus ini termasuk ke dalam skala besar.



Gambar 11. Peta Rute Solusi

Penelitian lebih lanjut diperlukan untuk mendapatkan solusi yang lebih baik pada pembagian *cluster* untuk kondisi yang sama yakni kendaraan heterogen baik secara kapasitas dan biaya, batasan waktu operasional perusahaan dan berskala besar. Penelitian lanjutan disarankan untuk melakukan perbaikan mekanisme algoritma dengan penggabungan metode lain yang biasa disebut *hybrid method* seperti *heuristic* (*Large Neighbourhood Search, Local Search*), *intra route, exchange route, 2-opt*, metaheuristik lainnya (*Simulated Annealing, Tabu Search, Ant Colony Optimization, dll*), dan *clustering algorithm*.

Sebagai contoh pada penelitian yang dilakukan oleh Comert et al. [27], Guezouli dan Abdelhamid [28], Siriuk dan Tangmo [29], Comert et al. [30], dan Sung et al. [6] menggunakan pendekatan *cluster first-route second*. Namun, pemilihan algoritma atau metode *cluster* harus diperhatikan dengan baik dengan mempertimbangkan kompleksitas permasalahan HVRP dengan batasan waktu operasional.

Jarak Tempuh

Performansi Algoritma *Particle Swarm Optimization* yang telah dibahas pada sub poin sebelumnya memperlihatkan bahwa algoritma PSO merupakan salah satu algoritma yang *robust*. Tabel 8 menunjukkan implementasi Algoritma PSO pada PT. XYZ dengan membandingkan jarak tempuh pada rute *existing* dan jarak tempuh pada rute setelah dilakukan optimasi dengan menggunakan algoritma PSO. Perbandingan antara rute *existing* dan rute solusi menggunakan rumus berikut ini:

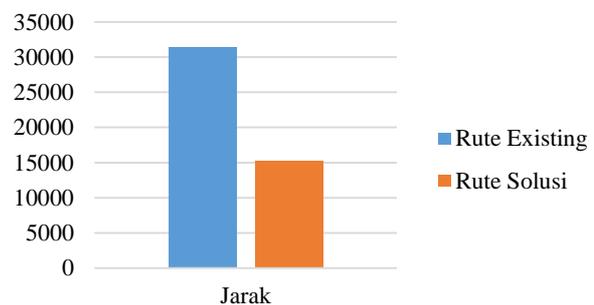
$$Gap (\%) = \frac{(PSO-Existing)}{Existing} \times 100 \tag{1}$$

Berdasarkan Tabel 8, total jarak tempuh untuk pengiriman produk di bulan Maret 2019 untuk rute *existing* adalah sebesar 31.456,63 km dan rute solusi sebesar 15.310,40 km dengan selisih 16.146,24 atau rata-rata 51,55%. Tanda dalam kurung pada Tabel 8 menunjukkan bahwa PSO memberikan hasil yang optimal atau lebih kecil dari kondisi *existing*. Penghematan jarak ini terjadi karena perencanaan pada rute *existing* belum optimal terlihat pada banyaknya rute yang dibuat serta pengelompokan pengiriman yang tidak memperhatikan total jarak tempuh saat dilakukan perencanaan pada *sending preparation*. Gambar 12

menunjukkan perbandingan secara keseluruhan terhadap jarak tempuh rute *existing* dengan rute solusi.

Tabel 8. Perbandingan Jarak Tempuh

Tgl.	Jumlah Customer	Jarak Total (km)		Gap (%)
		Existing	PSO	
1	57	698,911	338,48	(51,57)
2	59	885,493	433,61	(51,03)
4	165	2168,96	1245,30	(42,59)
5	194	2436,795	1251,20	(48,65)
6	66	955,099	396,65	(58,47)
8	98	1519,977	696,66	(54,17)
9	87	782,066	572,00	(26,86)
11	120	1901,023	905,90	(52,35)
12	53	829,564	296,64	(64,24)
13	45	874,35	330,15	(62,24)
14	75	1275,132	583,33	(54,25)
15	77	1359,85	557,72	(58,99)
16	110	1262,525	784,57	(37,86)
18	109	1761,891	856,53	(51,39)
19	121	1640,054	784,94	(52,14)
20	59	819,69	355,60	(56,62)
21	67	852,65	404,20	(52,59)
22	93	1529,02	751,19	(50,87)
23	63	860,15	462,45	(46,24)
25	124	1662,636	856,45	(48,49)
26	74	765,29	453,06	(40,80)
27	84	1110,567	476,21	(57,12)
28	89	1410,372	645,07	(54,26)
29	73	1260,688	483,85	(61,62)
30	67	833,881	388,65	(53,39)



Gambar 12. Grafik Perbandingan Jarak Tempuh

Biaya Distribusi

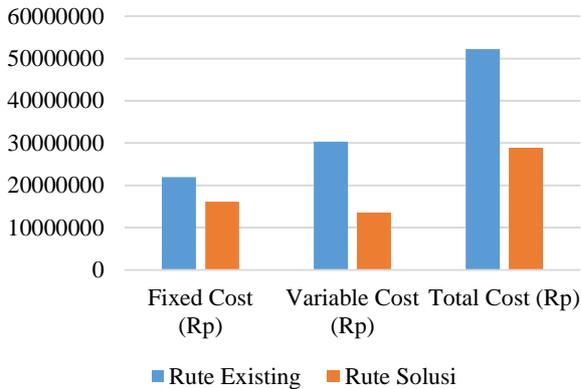
Biaya distribusi yang merupakan fungsi tujuan model *Heterogenous Fleet Routing Problem* pada penelitian ini dilakukan perbandingan untuk melihat besaran minimasi atau penghematan yang dapat didapatkan perusahaan jika menerapkan sistem pengoptimalan rute pengiriman menggunakan algoritma PSO. Tabel 9 menyajikan ringkasan perbandingan biaya pada rute *existing* perusahaan dan rute solusi.

Tabel 9. Perbandingan Biaya Distribusi

Perbandingan	Fixed Cost (Rp)	Variable Cost (Rp)	Total Cost (Rp)
Rute Existing	21908703,59	30330506,56	52239210,15
Rute Solusi	16146039,90	13571508,72	28900360,07
Selisih	5762663,68	16758997,84	23338850,08

Gambar 13 memperlihatkan perbandingan secara grafis terhadap biaya-biaya distribusi oleh rute *existing* dan rute solusi. Grafik Perbandingan Biaya menunjukkan bahwa pada *fixed cost* tidak

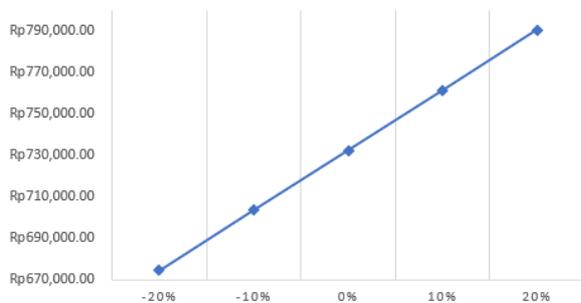
terdapat selisih yang cukup signifikan, sedangkan pada *variable cost* dan *Total Cost* menunjukkan selisih yang signifikan. Hal tersebut dikarenakan terjadinya penghematan jarak tempuh rata-rata sebesar 51,55% dari rute *existing*.



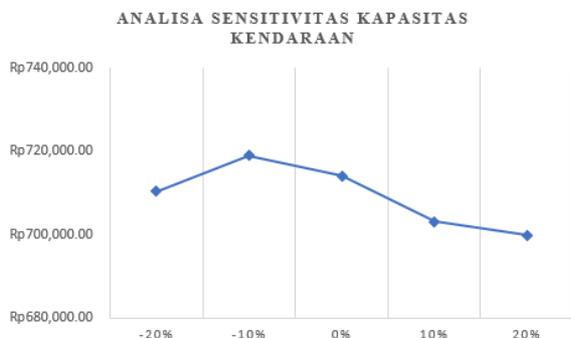
Gambar 13. Grafik Perbandingan Biaya-Biaya

Analisis Sensitivitas

Analisis sensitivitas dilakukan untuk mengetahui pengaruh perubahan parameter terhadap fungsi objektif. Parameter yang akan dianalisa yaitu *variable cost* dan kapasitas kendaraan terhadap *total cost* dengan menggunakan data pengiriman tanggal 1 Maret 2019. Analisis sensitivitas dilakukan dengan cara membuat beberapa skenario perubahan parameter dengan melakukan pengurangan dan penambahan dari kondisi *existing* dari setiap nilai sebesar -20%, -10%, 10% dan 20%. Hal ini dilakukan untuk melihat perubahan dari solusi yang dihasilkan ketika terjadi perubahan pada satu parameter.



Gambar 14. Grafik Analisis Sensitivitas *Variable Cost*



Gambar 15. Grafik Analisis Sensitivitas Kapasitas Kendaraan

Berdasarkan Gambar 14 terlihat bahwa penurunan dan peningkatan nilai *variable cost* mempengaruhi secara garis lurus

artinya ketika terjadi penurunan pada *variable cost* maka pada *total cost* akan mengalami penurunan, begitula pula sebaliknya. Pada parameter kapasitas kendaraan berdasarkan Gambar 15 terlihat bahwa setiap perubahan kapasitas kendaraan mempengaruhi *total cost* yang dikeluarkan. Meskipun demikian perubahan kapasitas kendaraan tidak mempengaruhi secara garis lurus atau secara signifikan terhadap *total cost*. *Sensitivity analysis* menunjukkan bahwa *total cost* dipengaruhi oleh *variable cost* namun tidak dipengaruhi secara signifikan oleh kapasitas kendaraan.

KESIMPULAN DAN SARAN

Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian yang dilakukan pada studi kasus PT. XYZ adalah model penyelesaian *Vehicle Routing Problem* menggunakan Algoritma *Particle Swarm Optimization* secara keseluruhan pada rute pengiriman bulan Maret 2019 dengan titik pengiriman 45-194 per hari rata-rata waktu komputasi (*CPU time*) yang dihasilkan sebesar 40,014 detik menggunakan Laptop Lenovo G40 dengan spesifikasi *processor* AMD A8 2.20 GHz dan RAM 4GB. Perhitungan nilai koefisien variansi menunjukkan solusi yang dihasilkan telah stabil. Namun, pada model penyelesaian VRP ini *output* program menghasilkan *cluster* yang belum optimal sehingga diperlukan perbaikan terhadap mekanisme algoritma. Rata-rata minimasi *Total Cost* yang dihasilkan dengan menggunakan Algoritma *Particle Swarm Optimization* sebesar Rp. 1.156.014,40 dan rata-rata *Total Cost* pada rute *existing* sebesar Rp. 2.089.568,41. Sehingga rata-rata selisih penghematan *Total Cost* sebesar 44,92%. Rata-rata minimasi jarak tempuh yang Algoritma *Particle Swarm Optimization* yaitu 612,42 km dan rata-rata jarak tempuh pada rute *existing* yaitu 1.258,27 km. Sehingga rata-rata selisih penghematan jarak tempuh sebesar 51,55%. Studi parameter dengan menggunakan *design of experiment* dengan beberapa faktor yang diuji yaitu faktor *swarm*, faktor *social learning* (C1) dan faktor *cognitive learning* (C2) menunjukkan bahwa kombinasi parameter terbaik yaitu jumlah *swarm* sebanyak 50, nilai *cognitive learning* (C1) yaitu 1.5, nilai *social learning* (C2) yaitu 2. Analisa dengan uji ANOVA menunjukkan pada pemilihan parameter jumlah *swarm* harus dipilih secara hati-hati karena berpengaruh secara signifikan terhadap *output* yang dihasilkan. Namun, interaksi dari ketiga faktor tidak menunjukkan hasil yang signifikan

Saran

Berdasarkan penelitian yang telah dilakukan maka dapat diberikan saran antara lain penelitian selanjutnya dapat menggunakan modifikasi terhadap tahapan algoritma menggunakan *hybrid metaheuristic* untuk mendapatkan solusi yang lebih baik. Pada penelitian selanjutnya untuk jenis permasalahan yang sama disarankan untuk menerapkan sistem *cluster first route second* untuk mendapatkan solusi yang lebih baik pada rute pengiriman khususnya untuk titik pengantaran yang banyak. Penelitian selanjutnya juga diharapkan dapat memasukkan variabel kemacetan sehingga data *time travel* dapat lebih akurat. Penelitian ini dapat dilanjutkan dengan menggunakan metaheuristik lainnya seperti *Ant Colony Optimization*, *Artificial Bee Colony*, *Differential Evolution* dan

lain-lain untuk dibandingkan performansinya dengan *Particle Swarm Optimization*.

UCAPAN TERIMA KASIH

Ucapan terima kasih ditujukan kepada Bapak Eric, Bapak Yusri, dan Bapak Oon atas kesediaannya membantu penulis dalam memperoleh data penelitian.

DAFTAR PUSTAKA

- [1] S. Parkhi, J. D and R. A. Kumar, "A Study on Transport Cost Optimization in Retail Distribution," *Journal of Supply Chain Management System*, pp. 31-38, 2014.
- [2] D. Cattaruzza, N. Absi, D. Feillet and J. Gonzales-Feliu, "Vehicle Routing Problems for City Logistics," *EURO Journal Transportation Logistics*, 2014. <https://doi.org/10.1007/s13676-014-0074-0>.
- [3] E. Wirdanto, D. Regenie and W. , "Aplikasi Algoritma Hybrid Dalam Penentuan Rute Pendistribusian Produk," *Jurnal Optimasi Sistem Industri* Vol. 15 No. 2, pp. 171-180, 2016. <https://doi.org/10.25077/josi.v15.n2.p171-180.2016>.
- [4] S. H. Xu, J. P. Liu, F. H. Zhang, L. Wang and L. J. Sun, "A Combination of Genetic Algorithm and Particle Swarm Optimization for Vehicle Routing Problem with Time Windows," Licensee MDPI, Basel, Switzerland, Switzerland, 2015.
- [5] F. R. Jacobs and R. Chase, *Operation and Supply Chain Management Edisi 14 Buku 1*, Jakarta: Salemba Empat, 2015.
- [6] H. Sun, K. Dong, Q. Zhang and R. Yan, "An Improved Particle Swarm Optimization Algorithm for Heterogeneous Fleet Vehicle Routing Problem with Two-Dimensional Loading Constraints," in *International Conference on Materials Engineering and Information Technology Applications (MEITA 2015)*, 2015. <https://doi.org/10.2991/meita-15.2015.74>.
- [7] A. Arvianto, S. A.H and S. S, "Model Vehicle Routing Problem dengan Karakteristik Rute Majemuk, Multiple Time Windows, Multiple Products dan Heterogeneous Fleet untuk Depot Tunggal," *Jurnal Teknik Industri* Vol. 16 No.2, pp. 83-94, 2014. <https://doi.org/10.9744/jti.16.2.83-94>.
- [8] M. B. Abdallah and M. Ennigrou, "Hybrid Multi-Agent Approach to Solve the Multi-depot Heterogeneous Fleet Vehicle Routing Problem with Time Windows (MDHFVRPTW)," *HIS*, pp. 376 - 386, 2018. https://doi.org/10.1007/978-3-030-14347-3_37.
- [9] M. Rabbani, N. Manivizadeh and A. Shamekhi, "A Particle Swarm Optimization Method for Periodic Vehicle Routing Problem with Pickup and Delivery in Transportation," *Advances in Railway Engineering, An International Journal* Vol. 1/ No.1/Autumn, pp. 51-60, 2013.
- [10] F. Belmecheri, C. Prins, F. Yalaoui and L. Amodéo, "Particle Swarm Optimization to Solve the Vehicle Routing Problem with Heterogeneous Fleet, Mixed Backhauls, and Time Windows," *IEEE*, 2010. <https://doi.org/10.1109/IPDPSW.2010.5470702>.
- [11] H. Alinezhad, S. Yaghubi, S.-M. Hoseini-Motlagh, S. Allahyari and N. S. Mojtaba, "An Improved Particle Swarm Optimization for a Class of Capacitated Vehicle Routing Problem," *International Journal of Transportation Engineering* Vol.5/No.4/Spring, pp. 331-347, 2018.
- [12] T. Iswari and A. M. S. Asih, "Comparing Genetic Algorithm and Particle Swarm Optimization for Solving Capacitated Vehicle Routing Problem," in *International Conference on Industrial and System Engineering (ICOnISE)*, 2017. <https://doi.org/10.1088/1757-899X/337/1/012004>.
- [13] K. Karagul and I. Gungor, "A Case Study of Heterogeneous Fleet Vehicle Routing Problem: Touristic Distribution Application in Alanya," *An International Journal of Optimization and Control: Theories & Applications* Vol 4 No 2, pp. 67 - 76, 2014. <https://doi.org/10.11121/ijocta.01.2014.00185>.
- [14] P. S. A. Penna and L. S. Ochi, "An Iterated Local Search Heuristic for The Heterogeneous Fleet Vehicle Routing Problem," *Journal of Heuristics* Vol. 19 No.2, pp. 201-232, 2013. <https://doi.org/10.1007/s10732-011-9186-y>.
- [15] S. D and S. D. O, "A Tabu Search Heuristic for the Heterogeneous Vehicle Routing Problem on a Multi-Graph," *Journal of Transportation Research Part E: Logistics and Transportation Review*, pp. 32-52, 2016. <https://doi.org/10.1016/j.tre.2015.12.001>.
- [16] Y. Vincent F, A. Redi, Y. Agustina and O. J. Wibowo, "A Simulated Annealing Heuristic for The Hybrid Vehicle Routing Problem," *Applied Soft Computing Journal*, 2016.
- [17] N. Saputra, P. D. Sentia and A. , "Penentuan Rute Kendaraan Heterogen Menggunakan Algoritma Insertion Heuristic," *Jurnal Optimasi Sistem Industri* Vol 17 No 1, pp. 35-45, 2018. <https://doi.org/10.25077/josi.v17.n1.p35-45.2018>.
- [18] I. Cholissodin and E. Riyandani, *Swarm Intelligence*, Malang: Fakultas Ilmu Komputer Universitas Brawijaya, 2016.
- [19] C. Pornsing, "A Particle Swarm Optimization for The Vehicle Routing Problem," *University of Rhode Island, Rhode Island*, 2014.
- [20] J. C. Bansal, P. K. Singh and N. R. Pal, *Evolutionary and Swarm Intelligence Algorithms*, Springer International Publishing Ag, 2019. <https://doi.org/10.1007/978-3-319-91341-4>.
- [21] W. A. Prathama and S. R. Sulistyono, "Penentuan Rute Distribusi Bantuan Medis untuk Bencana Erupsi Gunung Merapi di Yogyakarta," in *Seminar Nasional Teknik Industri Universitas Gajah Mada, Yogyakarta*, 2016.
- [22] M. A. El-Shorbagy and A. E. Hassanien, "Particle Swarm Optimization from Theory to Applications," *International Journal of Rough Sets and Data Analysis*, 2018. <https://doi.org/10.4018/IJRSDA.2018040101>.
- [23] B. Santosa, *Pengantar Metaheuristik: Implementasi dengan Matlab*, Surabaya: ITS Tekno Sains, 2017.
- [24] F. Dobsław, "A Parameter Tuning Framework for Metaheuristics Based on Design of Experiment and Artificial Neural Networks," *Proceeding of the International Conference on Computer Mathematics and Natural Computing*, 2010.
- [25] R. Korkmaz Tan and S. Bora, "Parameter Tuning Algorithms in Modeling and Simulation," *International Journal of Engineering Science and Application* Vol. 1 No. 2, 2017.
- [26] S. Hidayat, "Pertimbangan Isu Transportasi Dalam Perencanaan Ruang Kota Makassar," *Dinas Pengelolaan Lingkungan Hidup Provinsi Sulawesi Selatan, Makassar*, 2017.
- [27] S. E. Comert, H. R. Yazgan, S. Kir and F. Yener, "A Cluster First-Route Second Approach for A Capacitated Vehicle Routing Problem: A Case Study," *International Journal of*

Procurement Management Vol. 11 No 4, pp. 399-419, 2018. <https://doi.org/10.1504/IJPM.2018.092766>.

- [28] L. Guezouli and S. Abdelhamid, "Multi-objective Optimisation Using Genetic Algorithm Based Clustering for Multi-Depot Heterogeneous Fleet Vehicle Routing Problem with Time Windows," *Internasional Journal of Mathematics in Operational Research* Vol 13 No 3, pp. 332-349, 2018. <https://doi.org/10.1504/IJMOR.2018.094850>.
- [29] P. Siriruk and Y. Tangmo, "Heterogeneous Fleet Vehicle Routing Problem for Retail Construction Material Stores," in *4th International Conference on Industrial Engineering and Applications*, 2017.
- [30] S. E. Comert, H. R. Yazgan, I. Sertvuran and H. Sengul, "A New Approach for Solution of Vehicle Routing Problem with Hard Time Window: An Application in A Supermarket Chain," *Sadhana* Vol 42 No 12, pp. 2067-2080, 2017. <https://doi.org/10.1007/s12046-017-0754-1>.

NOMENKLATUR

PSO = *Particle Swarm Optimization*

VRP = *Vehicle Routing Problem*

HFVRP = *Heterogeneous Fleet Vehicle Routing Problem*

Himpunan Model:

N = Himpunan lokasi pelanggan termasuk depot.

K = Himpunan Kendaraan

Indeks Model:

i = indeks lokasi

j = indeks lokasi

k = indeks kendaraan

Parameter:

U_k = Kapasitas kendaraan k (m^3)

d_i = permintaan di lokasi i (m^3)

c_{ij} = biaya operasional dari i ke j (Rp)

f_k = biaya tetap kendaraan k (Rp)

Variabel – variabel Keputusan Model:

$$x_{ij}^k = \begin{cases} 1, & \text{jika kendaraan } k \text{ beroperasi dari } i \text{ ke } j \\ 0, & \text{jika lainnya} \end{cases}$$

$$z_k = \begin{cases} 1, & \text{jika kendaraan } k \text{ beroperasi} \\ 0, & \text{jika lainnya} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{jika kendaraan } k \text{ mengunjungi lokasi } i \\ 0, & \text{jika lainnya} \end{cases}$$

Notasi Algoritma Particle Swam Optimization

X_i^t = Posisi partikel i pada iterasi t

V_i^t = Kecepatan partikel i pada iterasi t

$P_{best\ i}^t$ = Posisi terbaik dari partikel i pada iterasi t

$G_{best\ i}^t$ = Posisi terbaik secara global pada iterasi t

c_1 = *Cognitive Learning*

c_2 = *Social Learning*

r_1 & r_2 = Bilangan Acak $1 - 0$

ω = Bobot Inersia