Implementasi Algoritma *Discrete Particle Swarm Optimization* untuk Permasalahan *Capacitated Vehicle Routing Problem*

Aisyahna Nurul Mauliddina[1], Faris Ahmad Saifuddin[2], Adesatya Lentera Nagari[3]
[1]Fakultas Teknologi Industri, Universitas Pertamina, Jakarta, Indonesia
[2]Fakultas Teknologi Industri, Universitas Pertamina, Jakarta, Indonesia
[3]Fakultas Teknologi Industri, Universitas Pertamina, Jakarta, Indonesia

Abstrak - ***Capacitated Vehicle Routing Problem* (CVRP) merupakan salah satu permasalahan kompleksitas tinggi yang dalam skala besar sulit diselesaikan dengan metode eksak. Oleh karena itu, pendekatan algoritma metaheuristik dikembangkan guna menyelesaikan permasalahan NP-hard sepesrti CVRP. Penelitian ini akan mengimplementasikan algoritma *Discrete Particle Swarm Optimization* (DPSO) untuk sepuluh dataset dengan jumlah titik dan kendaraan yang bervariasi. Proses *parameter tuning* digunakan untuk memilih kombinasi parameter terbaik dalam memperoleh hasil yang optimal. Selain itu, tes statistik digunakan untuk mengetahui performansi antarparameter PSO dan perbandingan efektifitas nilai objektif yang dihasilkan dengan algoritma lainnya. Hasil komputasi menunjukkan algoritma DPSO dengan parameter terpilih dan waktu komputasi minimal belum mampu mencapai nilai terbaik terutama untuk data dengan skala besar.**

PENDAHULUAN

Vehicle Routing Problem (VRP) merupakan sebuah desain permasalahan yang sangat berkaitan dengan kegiatan logistik dan distribusi (Augerat, Belenguer, Benavent, Corberan, & Naddef, 1998). Salah satu bentuk permasalahan dasar dari VRP yaitu *Capacitated Vehicle Routing Problem* (CVRP) yang pertama kali diperkenalkan oleh Dantzig dan Ramser pada tahun 1959 (Dantzig & Ramser, 1959). CVRP merupakan sebuah desain permasalahan yang memiliki fungsi tujuan meminimalkan total biaya distribusi dari depot ke sejumlah lokasi dalam rangka memenuhi permintaan konsumen. Dalam model CVRP diasumsikan proses distribusi dilakukan dengan menggunakan beberapa kendaraan sejenis yang memiliki kapasitas seragam (Marinakis, Iordanidou, & Marinaki, 2013).

CVRP merupakan permasalahan yang luas dan umum, oleh karena itu banyak peneliti mengembangkan model CVRP menjadi lebih spesifik sesuai dengan permasalahan yang dihadapi. Beberapa pengembangan model CVRP diantaranya bertujuan agar mampu menangani beberapa konstrain tambahan seperti jumlah permintaan yang stokastik (Marinakis, Iordanidou, & Marinaki, 2013), konstrain waktu atau *time windows* (Hannan, et al., 2017), konstrain *fuzzy demand* (Kuo, Zulvia, & Suryadi, 2012), dan lain sebagainya. Dalam kaitannya dengan proses distribusi dan logistik, CVRP dinilai menjadi kunci utama yang harus diselesaikan untuk menangani masalah operasional terkait penentuan rute kendaraan. Penyelesaian permasalahan CVRP memliki tingkat kesulitan yang berbeda tergantung pada kompleksitas permasalahan yang dihadapi.

Dari segi kompleksitas permasalahan, pada dasarnya CVRP tergolong permasalahan NP-hard. Hal ini mengakibatkan penyelesaian permasalahan CVRP sering dihadapkan dengan sulitnya mendapatkan hasil yang optimal dan waktu komputasi yang lama (Haimovich, Rinnooy Kan, & Stougie, 1988). Oleh karena itu, dibutuhkan penelitian lebih lanjut mengenai metode pendekatan yang tepat untuk memperoleh hasil yang optimal dengan waktu komputasi yang efisien (Ai & Kachitvichyanukul, 2009). Pertimbangan banyaknya titik dan konstrain membuat permasalahan CVRP sangat sulit untuk diselesaikan menggunakan metode eksak atau metode konvensional. Proses pencarian nilai solusi yang optimal dari permasalahan NP-hard dengan *instances* yang besar biasanya dilakukan menggunakan pendekatan algoritma metaheuristik. Beberapa penelitian sebelumnya telah mengembangkan pendekatan metaheuristik dalam penyelesaian permasalahan CVRP diantaranya menggunakan algoritma genetika (Baker & Ayechew, 2003), *particle swarm optimization* (PSO) (Chen, Yang , & Wu, 2006), (Ai & Kachitvichyanukul, 2009), *tabu search* (Augerat, Belenguer, Benavent, Corberan, & Naddef, 1998), dan beberapa algoritma lainnya. Penelitian ini akan membahas lebih lanjut mengenai penyelesaian permasalahan CVRP menggunakan model dasar algoritma *Particle Swarm Optimization* (PSO).

PSO merupakan salah satu algoritma metaheuristik berbasis populasi yang ditemukan oleh Kennedy dan Eberhart (Eberhart & Kennedy, 1995). Algoritma PSO mengadaptasi perilaku sekumpulan burung atau ikan di alam dengan mensimulasikan pergerakan individu dalam kumpulan untuk mendapatkan nilai optimal. Algoritma PSO memiliki beberapa kelebihan diantaranya dapat digunakan untuk fungsi objektif yang stokastik, dapat dengan mudah keluar dari lokal optimal, mudah dalam pemrograman dan implementasinya, terdapat beberapa parameter yang dapat diatur serta penentuan inisial solusi tidak akan mempengaruhi hasil optimal yang didapatkan (Marinakis, Iordanidou, & Marinaki, 2013). Selain itu, algoritma PSO menggunakan pergerakan fisik individu dalam *swarm* dan memiliki mekanisme yang fleksibel dan seimbang untuk mampu mengadaptasi nilai terbaik yang didapat individu maupun *swarm* (Chen, Yang , & Wu, 2006). Pemanfaatan algoritma PSO pada dasarnya ditujukan untuk permasalahan kontinyu, namun beberapa penelitian telah mengembangkan algoritma PSO untuk mampu diaplikasikan pada permasalahan diskrit. Penelitian terkait dengan aplikasi *Discrete Particle Swarm Optimization*

(DPSO) pada permasalahan CVRP telah dilakukan oleh (Chen, Yang , & Wu, 2006) yang mengkombinasikan DPSO dengan algoritma *Simulated Annealing* (SA) serta penelitian (Ai & Kachitvichyanukul, 2009) yang mengembangkan DPSO dengan 2 respresentasi solusi yang berbeda.

Penelitian ini akan terfokus untuk menyelesaikan permasalahan CVRP dengan algoritma DPSO menggunakan data yang digunakan oleh (Chen, Yang , & Wu, 2006) dalam penelitiannya. Penelitian ini akan meneliti lebih lanjut mengenai algoritma dasar DPSO dengan mengatur parameternya sedemikian rupa sehingga mampu menghasilkan nilai yang optimal. Proses *parameter tuning* pada penelitian ini didasarkn pada beberapa literatur dan menggunakan metode *One Factor at Time* (OFAT) untuk mendapatkan parameter terbaik yang akan diimplementasikan pada algoritma DPSO. Hasil yang didapatkan pada penelitian ini akan dibandingkan dengan dua penelitian sebelumnya menggunakan analisis statistik

Penelitian ini terbagi menjadi beberapa yaitu bagian satu berupa pendahuluan terkait permasalahan dalam penelitian, bagian dua merupakan studi literatur terkait CVRP dan algoritma PSO. Setelah itu, bagian tiga menjelaskan mengenai metodologi penelitian. Bagian empat memaparkan secara rinci mengenai data yang digunakan, proses pengolahan data serta analisis hasil yang didapatkan. Terkahir, bagian lima akan merangkum hasil penelitian dan saran untuk penelitian selanjutnya.

STUDI LITERATUR

1. *Capacitated Vehicle Routing Problem* (CVRP)

Vehicle Routing Problem (VRP) merupakan suatu permasalahan optimasi dalam penentuan rute pendistribusian barang. Tujuan utama dari VRP adalah untuk menentukan rute kendaraan dalam melayani semua permintaan konsumen. Kendaraan akan bergerak dari depot menuju ke lokasi pelanggan dan kemudian akan kembali ke depot (Kuo, Zulvia, & Suryadi, 2012).

Ada beberapa variasi masalah VRP, salah satunya adalah *Capacitated Vehicle Routing Problem* (CVRP). Pada permasalahan CVRP terdapat penambahan kendala berupa kapasitas kendaraan. CVRP dapat didefinisikan sebagai masalah penentuan rute optimal dalam memenuhi permintaan konsumen dengan tujuan untuk mendapatkan biaya perjalanan, waktu perjalanan, dan jumlah kendaraan yang optimal dan seminimal mungkin (Irnich & Vigo, 2014).

Model matematika dari permasalahan CVRP secara umum dimodelkan sebagai berikut (Bodin, Golden, Assad, & Ball, 1983):
Fungsi tujuan dari model matematika CVRP:

Minimize

$$Z = \sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij}^{k} X_{ij}^{k} \quad (1)$$

Subject to:

$$X_{ij}^{k} = \begin{cases} 1, & \text{jika kendaraan } k \text{ memiliki rute dari } i \text{ ke } j \\ 0, & \text{jika tidak ada} \end{cases} \quad (2)$$

$$\sum_{k=1}^{K}\sum_{i=0}^{N} X_{ij}^{k} = 1, \qquad j = 1, 2, \dots, N \quad (3)$$

$$\sum_{k=1}^{K}\sum_{j=0}^{N} X_{ij}^{k} = 1, \qquad i = 1, 2, \dots, n \quad (4)$$

$$\sum_{i=0}^{N} X_{it}^{k} - \sum_{j=0}^{N} X_{tj}^{k} = 0, \qquad k = 1, 2, \dots, K; t = 1, 2, \dots, N \quad (5)$$

$$\sum_{j=0}^{N} q_j \left(\sum_{i=0}^{N} X_{ij}^{k} \right) \le Q_k, \qquad k = 1, 2, \dots, K \quad (6)$$

$$\sum_{j=1}^{N} X_{0j}^{k} \le 1, \qquad k = 1, 2, \dots, K \quad (7)$$

$$\sum_{i=1}^{N} X_{i0}^{k} \le 1, \qquad k = 1, 2, \dots, Ks \quad (8)$$

$$X_{ij}^{k} \in \{0,1\}, i, j = 0, 1, 2, \dots, Nn;\ k = 1, 2, \dots, K \quad (9)$$

Pada model matematis di atas, notasi *N* mewakili jumlah konsumen, *K* adalah jumlah kendaraan yang digunakan, C_{ij}^{k} adalah jarak perjalanan dari konsumen *i* ke konsumen *j* dengan kendaraan *K*, dan d_{ij}^{k} merupakan jarak tempuh dari konsumen *i* ke konsumen *j* dengan menggunakan kendaraan *K*. selain itu, notasi Q_k, mewakili kapasitas kenaraa

Fungsi objektif dari persamaan (1) adalah untuk meminimalkan total jarak yang yang ditempuh kendaraan. *Constraints* pada persamaan (3) dan (4) menjelaskan bahwa setiap pelanggan hanya akan dikunjungi oleh kendaraan sebanyak satu kali. Persamaan (5) dapat diartikan bahwa jumlah kendaraan yang pergi dan kembali ke depot adalah sama. Persamaan (6) menjelaskan bahwa muatan yang diangkut kendaraan tidak akan melebihi kapasitas yang sudah ditetapkan. Persamaan (7) dan (8) menjelaskan bahwa kendaraan hanya digunakan sekali dalam satu siklus distribusi dan persamaan (9) digunakan untuk memastikan bahwa variabel yang diambil merupakan bilangan integer (0 atau 1).

2. *Particle Swarm Optimization* (PSO)

Algoritma PSO pertama kali dikembangkan oleh (Eberhart & Kennedy, 1995) yang didasarkan pada perilaku *swarm* di alam, seperti sekumpulan ikan maupun burung (Yang, 2014). Algoritma PSO dapat digunakan hampir di seluruh kasus optimasi, *computational intelligence*, dan desain aplikasi. Algoritma ini juga biasanya dikombinasikan dengan algoritma lain sesuai dengan spesifikasi kasus yang akan diselesaikan. Konsep PSO dapat diaplikasikan pada permasalahan optimasi non-linear hingga permasalahan yang sangat sulit untuk dapat menemukan global optimum dan keluar dari jebakan lokal optimal (Eberhart & Kennedy, 1995).

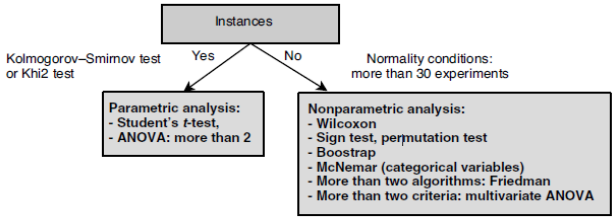
Dasar dari algoritma PSO untuk permasalahan optimasi non-linear ini adalah dengan mempertimbangkan berbagai kemungkinan kandidat solusi yang optimal di dalam suatu *swarm* (Yang, 2014). Di dalam *swarm* itu sendiri terdapat berbagai objek yang diamati yang dinamakan partikel. Masing-masing kandidat solusi yang ditemukan partikel dalam *swarm* selalu diperbaharui dan terus bergerak di wilayah pencarian secara acak di dalam *feasible region* sehingga dapat ditemukan solusi optimal dari suatu fungsi tujuan.

Algoritma PSO memiliki beberapa komponen penting antara lain jumlah partikel, *velocity* partikel, *personal best*, serta *global best*. Komponen utama dari algoritma PSO adalah posisi dan *velocity* (Lindfield & Penny, 2017). Posisi dari partikel dalam populasi ditentukan secara acak di dalam batas daerah pencarian dan nilai fungsi objektif dihitung dari titik tersebut untuk setiap partikel. Sedangkan kecepatan awal (*initial velocity*) ditentukan secara acak dalam rentang *velocity* yang ditentukan. Berdasarkan nilai objektif yang didapatkan masing-masing partikel kemudian dapat ditentukan nilai terbaik masing-masing individu yaitu *Personal Best* (PBest) dan nilai terbaik dalam kelompok yang disebut *Global Best* (GBest).

Prosedur dalam mengimplementasikan algoritma PSO diawali dnegan menentukan lokasi dan kecepatan awal, menentukan parameter lain, melakukan *constraint handling*, penentuan nilai *P-Best* dan *G-Best*, kemudian memperbarui kecepatan dan lokasi individu. Langkah tersebut dilakukan hingga *termination criteria* tercapai (Kuo, Zulvia, & Suryadi, 2012). Hasil yang didapatkan dari algoritma PSO dipengaruhi oleh nilai parameter yang digunakan (Ai & Kachitvichyanukul, 2009). Parameter tersebut antara lain *swarm size*, *learning factors* yang terdiri dari *cognitive learning* dan *social learning*, *inertia weight*, dan lain sebagainya. Oleh karena itu, dalam implementasi PSO dibutuhkan proses *parameter tuning* untuk menentukan nilai parameter yang paling baik untuk mendapatkan hasil yang optimal.

3. Uji Performansi Metaheuristik

Uji statistik dapat dilakukan setelah memperoleh hasil percobaan dari suatu algoritma dengan beberapa indikator yang berbeda. Tujuan dilakukan uji statistik adalah untuk mengetahui kinerja metaheuristik yang telah dirancang, seperti kualitas solusi yang diperoleh, serta dapat memperkirakan hasil dari suatu algoritma valid secara ilmiah. Penentuan alat penguji hipotesis dipilih berdasarkan karakteristik data (contoh: variansi dan ukuran sampel), seperti yang ditunjukan pada **Gambar 1.**



Gambar 1 Analisis Statistik Berdasal Hasil yang Diperoleh *(Talbi, 2009)*

Hypothesis testing merupakan metode dalam pengambilan keputusan berdasarkan analisis data, baik dari percobaan yang terkontrol ataupun tidak. Terdapat dua cara dalam dalam melakukan *hypothesis testing*, yaitu *parametric analysis* dan *non-parasmetric analysis*. Kedua strategi tersebut dibedakan dengan dengan asumsi-asumsi yang harus dipenuhi sebelumnya. Asumsi yang harus dipenuhi untuk *parametric analysis* diantaranya adalah sampel diperoleh secara *random* dan independen, fungsi objektif harus berupa interval atau rasio, data harus berdistribusi normal, dan nilai variansi pada sampel harus sama. Jika suatu permasalah dapat memenuhi semua asumsi yang ada pada *parametric analysis* maka akan dianggap lebih baik dan dapat terhindar dari error tipe II. Sedangkan asumsi yang harus dipenuhi pada *non-parametric analysis* adalah sampel diambil secara *random*, memiliki bentuk distribusi yang sama, dan jumlah variansi pada sampel harus sebanding. Namun penggunaan *non-parametric analysis* dalam suatu permasalahan dapat meningkatkan kemungkinan terjadinya error tipe II dibanding dengan menggunakan *parametric analysis.*

Ketika permasalahan yang akan diuji tidak dapat memenuhi asumsi uji statistik pada *parametric analysis*, maka metode yang akan diterapkan adalah pasangan uji statistik pada *non-parametric analysis.* Setiap metode uji statistik memiliki pasangan yang sepadan. Berikut merupakan pasangan metode uji statistik antara *parametric analysis* dan *non-parametric analysis*: *independent sample t-test* dengan *mann-whitney test, paired samples t-test* dengan *wilcoxon signed-rank test, one-way anova* dengan *kruskal-wallis test*, dan *one-way repeated measures anova* dengan *friedman's anova.*

Dalam melakukan pengujian terhadap asumsi normalitas dan homogenitas juga dapat menggunakan berbagai metode, *Normality* data dapat diketahui dengan melihat histogram dari data tersebut atau melakukan perhitungan *kurtosis* dan *skewness* data. Metode lain yang dapat digunakan untuk menguji normalitas data adalah *kolmogorov-smirnov* ayau *shapiro-wilkins.* Selain itu, metode *levene test* digunakan untuk menguji homogenitas dari varian setiap sampel.

METODOLOGI

Secara garis besar, penelitian ini akan dilakukan dengan langkah-langkah yaitu:

1. Implementasi DPSO untuk CVRP

Secara umum, implementasi algoritma PSO diawali dengan inisisasi awal partikel dan *velocity* partikel dalam *swarm*, pengaturan parameter dan *constraint handling*, hitung *fitness value*, tentukan PBest dan GBest, perbarui kecepatan dan lokasi partikel. langkah-langkah tersebut dilakukan secara berulang hingga menemui kriteria terminasi. Algoritma DPSO pada permasalahan CVRP secara rinci adalah sebagai berikut:

Start

***Setup* Parameter**: *swarm size, inertia weight, learning factors, velocity limit,* jumlah iterasi maksimal, waktu komputasi maksimal

Inisialisasi *Swarm*;

Evaluasi fungsi objektif tiap partikel;

Menentukan nilai PBest dan GBest awal

Repeat

do

for all particle i

- Hitung *velocity* baru dari partikel
- Hitung posisi baru partikel
- *Constraint Handling*
- Evaluasi nilai fungsi objektif partikel
- Cari PBest dan Gbest baru partikel
  If f(xi) < f(PBesti) **Then** PBesti = xi
  If f(xi) < f(GBesti) **Then** GBesti = xi
- Perbarui lokasi dan *velocity* partikel
- **Until** Kriteria Terminasi

End for

Hitung *nonimprovement*

Until *Stopping Criteria*

Print Hasil

End

2. *Parameter Tuning*

Dalam algoritma PSO, terdapat bebrapa parameter yang digunakan anatara lain *swarm size*, *learning factors*, jumlah iterasi maksimum, serta *inertia weight* (Isiet & Gadala, 2020). Seperti yang telah dipaparkan pada bagian sebelumnya mengenai parameter yang digunakan dalam PSO, proses penentuan besar masing-masing parameter disesuaikan dengan jenis permasalahan dan tidak ditentukan aturan bakunya. Pada penelitian ini, proses *parameter tuning* dilakukan untuk empat parameter sebagai berikut:

- Inertia weight (w)

  Penentuan besar nilai inertia weight yang optimal agar mampu meningkatkan performansi algoritma PSO telah banyak diteliti dan dikembangkan. Shi dan Eberhardt menjelaskan bahwa terbukti nilai *inertia weight* berkisar antara 0.9 hingga 1.2 mampu meningkatkan performansi algoritma PSO (Y.Shi & R. Eberhart, 1998). Nilai *inertia weight* harus dipertimbangkan dengan hati-hati sehubungan dengan nilai ini berpengaruh terhadap *velocity* partikel yang tentunya menentukan kemungkinan bergerak dan pencarian partikel. Penelitian lain memaparkan bahwa nilai *inertia weight* diantara 0.4-0.9 disarankan untuk problem yang cukup rumit (M. Pant, R. Thangaraj, & A. Abraham, 2009).

- *Learning Factor*

  Learning factors pada algoritma PSO meliputi *cognitive learning* dan *social learning*. *Cognitive learning* merepresentasikan seberapa besar pertimbangan dari nilai PBest dalam perhitungan *velocity* partikel. Sedangkan *social learning* menggambarkan besar pengaruh GBest dalam perhitungan nilai *velocity*. Kecepatan partikel sangat penting untuk diperhatikan untuk meghindari adanya *premature convergence*. Nilai *learning factors* $c_1 = 2, c_2 = 2$ direkomendasikan untuk perolehan hasil yang optimal (M. Pant, R. Thangaraj, & A. Abraham, 2009). Untuk parameter DPSO, (Eberhart & Shi, 2001) menggunakan 0,5 untuk masing-masing *cognitive learning* (c_1) dan *social learning* (c_2). Penelitian lain oleh (Wang, Geng, & Qiao, 2013) menggunakan nilai *learning factor* mulai dari 0,5 hingga 2,5.

- *Swarm Size*

  Penentuan banyak sedikitnya jumlah partikel berpengaruh langsung terhadap kecepatan komputasi serta kecepatan menemukan hasil yang optimal. Terdapat beberapa penelitian yang menjelaskan mengenai teknik penentuan jumlah partikel, diantaranya jumlah partikel optimal dapat ditentukan dalam kisaran $5x_N - 10x_N$, di mana x_N merupakan jumlah variabel yang dicari (J.S.Arora, 2017). Penelitian lain memaparkan bahwa jumlah partikel optimal adalah berkisar antara 10-30 partikel (I. C. Trelea, 2003). Sedangkan penelitian (Wang, Geng, & Qiao, 2013) menggunakan jumlah *swarm* bervariasi mulai dari 25 hingga 125.

3. Analisis Statistik

Pada penelitian ini, analisis statistik digunakan untuk menilai performansi parameter PSO yang dipilih dan menguji performansi algoritma DPSO dibandingkan dengan algoritma lainnya. Analisis statistik data meliputi statistik deskriptif dan uji hipotesis. Statistik deskriptif dibutuhkan untuk melihat detail dari data yang diamati. Sedangkan uji hipotesis pada penelitian ini digunakan untuk melihat perbedaan nilai rata-rata diantara data percobaan.

Uji hipotesis yang digunakan dalam penelitian ini adalah *Repetead Measure One Way* ANOVA. Uji ini dipilih karena bertujuan untuk membandingkan lebih dari dua rata-rata percobaan dengan perlakuan berbeda namun menggunakan *instance* yang sama. Sebelum melakukan uji *Repetead Measure One Way* ANOVA dilakukan uji normalitas dan pengujian terhadap *sphericity* data. Uji normalitas dilakukan menggunakan Uji Kolmogorov-Smirnov dan Uji Saphiro-Wilkins, sedangkan uji *sphericity* dilakukan dengan menggunakan uji *Muchly's test*.

PENGOLAHAN DATA DAN ANALISIS HASIL

Pada bagian ini, DPSO digunakan untuk menyelesaikan permasalahan CVRP dasar dengan menggunakan *benchmark data*. Parameter yang digunakan dalam implementasi DPSO merupakan parameter terbaik yang ditentukan berdasarkan *parameter tuning* menggunakan metode OFAT. Hasil optimal yang didapatkan dari algoritma ini kemudian akan dibandingkan dengan algoritma DPSO-SA, SR-1, dan SR-2.

1. Data

Dataset yang digunakan pada penelitian ini meliputi 10 *instances* data yang juga digunakan oleh (Chen, Yang , & Wu, 2006) dan (Ai & Kachitvichyanukul, 2009) seperti

terlihat pada **Tabel 1**. Sepuluh *instances* tersebut kemudian diselesaikan menggunakan DPSO dengan menggunakan aplikasi Visual Studio dan bahasa pemrograman C#. Dalam peneltiian ini, simulasi dilakukan dengan menggunakan laptop Intel Core i5 @1,8GHz quad-core with Turbo Boost-8GB RAM.

Tabel 1 *Benchmark Datasets*

Instances	Jumlah Customer	Jumlah Vehicles
An33k5	32	5
An46k7	45	7
An60k9	59	9
Bn35k5	34	5
Bn45k5	44	5
Bn68k9	67	9
Bn78k10	77	10
En30k3	29	3
En51k5	50	5
En76k7	75	7

2. Hasil Parameter Tuning

Pada bagian sebelumnya telah dijelaskan bahwa pengaturan parameter pada penelitian ini dilakukan untuk empat parameter yaitu *inertia weight* (w), *cognitive learning factor* (c_1), *social learning factor* (c_2), dan *swarm size* (N). Pengaturan parameter dilakukan dengan memilih sejumlah kandidat nilai untuk masing-masing parameter. Setelah itu, dilakukan eksperimen dengan OFAT untuk menentukan nilai parameter terpilih yang akan digunakan. Kandidat nilai masing-masing parameter dituliskan secara rinci pada **Tabel 2**. Pemilihan kombinasi nilai parameter tersebut didasarkan pada studi literatur dan merujuk kepada parameter yang digunakan pada penelitian (Wang, Geng, & Qiao, 2013).

Tabel 2 Daftar Nilai Parameter

Parameter					
	***Inertia Weight* (w)**	***Learning Factor* (c1)**	***Learning factor* (c2)**	***Swarm Size* (N)**	
***Inertia Weight* (w)**	0,9	2	1	25	Percobaan 1
	0,3	2	1	25	Percobaan 2
	0,4	2	1	25	Percobaan 3
	0,5	2	1	25	Percobaan 7*
***Learning Factor* (c1)**	0,5	1	1	25	Percobaan 4
	0,5	1,5	1	25	Percobaan 5
	0,5	0,5	1	25	Percobaan 6
	0,5	2	1	25	Percobaan 7*
***Learning factor* (c2)**	0,5	2	1	25	Percobaan 7*
	0,5	2	1,5	25	Percobaan 8
	0,5	2	2	25	Percobaan 9
***Swarm Size* (N)**	0,5	2	1	25	Percobaan 10
	0,5	2	1	50	Percobaan 11
	0,5	2	1	75	Percobaan 12

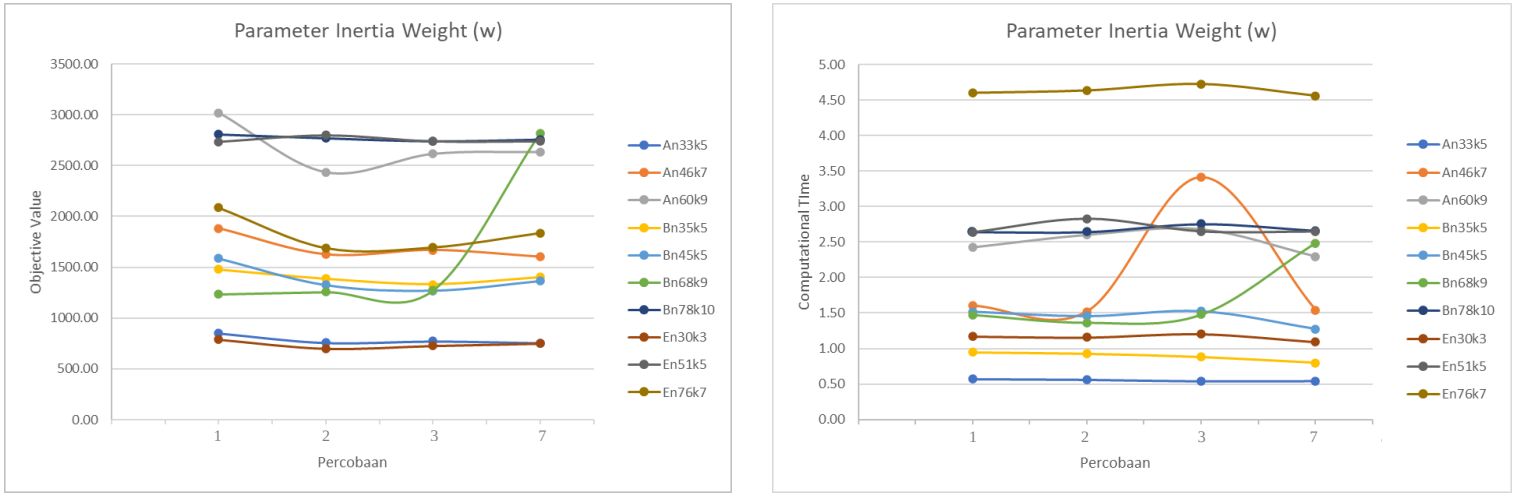
Tabel di atas menjelaskan mengenai kombinasi parameter untuk masing-masing percobaan di setiap faktor yang diuji. Pengaturan parameter pada OFAT dilakukan dengan cara mengubah nilai dari variabel uji dan memberikan nilai tetap untuk semua percobaan di variabel lainnya. Sebagai contoh, ketika ingin menentukan nilai terbaik untuk parameter c_1, maka dilakukan empat percobaan dengan mengubah nilai c_1 untuk masing-masing percobaan, sedangkan menentukan nilai tetap untuk parameter *inertia weight*, *swarm size*, dan c_2. Dalam penelitian ini dilakukan 4 percobaan untuk faktor *inertia weight*, 4 perobaan untuk faktor c_1, 3 percobaan untuk faktor c_2, dan 3 percobaan untuk faktor *swarm size*.

Sesuai dengan parameter pada Tabel 2, masing-masing kombinasi parameter pada setiap percobaan diimplementasikan pada algoritma DPSO dan digunakan untuk menentukan nilai optimal di semua *instances*. Setiap percobaan dilakukan dengan melakukan replikasi sebanyak 30 kali. Kemudian dihitung nilai rata-rata untuk fungsi objektif dan waktu komputasi setiap percobaan di masing-masing *instances*. Rekapitulasi hasil percobaan OFAT dapat dilihat pada **Tabel 3.** Setelah itu, untuk setiap parameter dilakukan perbandingan hasil percobaan untuk melihat pengaruh faktor tersebut pada nilai objektif dan waktu komputasi. Grafik perbandingan nilai objektif dan waktu komputasi untuk masing-masing faktor dapat dilihat pada **Gambar 2-5.**

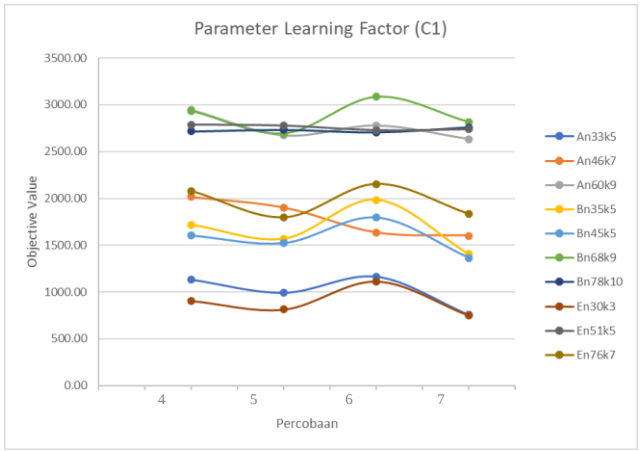
Gambar 2-5 menunjukan bahwa untuk setiap percobaan pada masing-masing parameter memiliki nilai yang berbeda. Hal ini dapat dipengaruhi karena perbedaan nilai parameter. Namun, untuk setiap percobaan dalam satu kelompok faktor, terdapat kemiripan pola untuk beberapa *instances*. Selain itu, terdapat perbedaan nilai objektif dan waktu komputasi antarpercobaan yang tidak terlalu jauh sehingga sulit untuk langsung menentukan apakah terdapat perbedaan Maka diperlukan proses *statistical analysis* menggunakan perangkat lunak SPSS untuk melihat detail secara statistik perbedaan nilai di masing-masing percobaan pada setiap faktor.

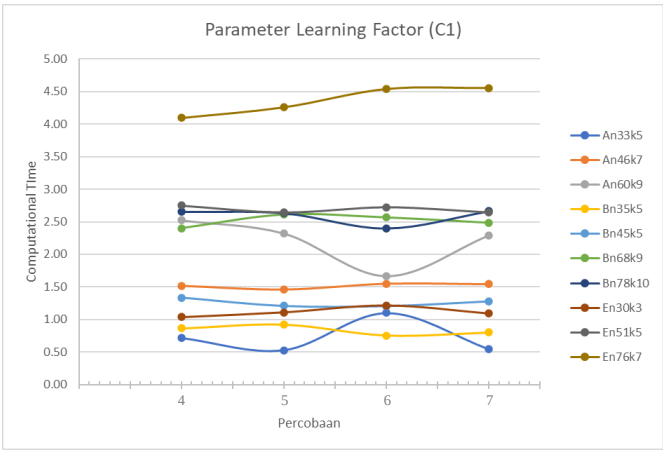
Tabel 3 Rekapitulasi Percobaan DPSO

Instances	Jumlah Cust	Jumlah Vehices	Percobaan																					
			1		2		3		4		5		6		7		8		9		10		11		12	
			Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time	Obj	Time
An33k5	**32**	**5**	850,51	0,57	756,43	0,56	771,50	0,54	1132,89	0,72	993,21	0,53	1165,43	1,10	754,32	0,54	779,47	0,81	731,56	1,47	754,32	0,54	803,09	0,53	800,00	1,00
An46k7	**45**	7	1880,75	1,60	1624,82	1,52	1668,32	3,42	2015,68	1,52	1901,94	1,46	1633,98	1,55	1601,70	1,54	1510,49	0,72	1527,72	0,88	1601,70	1,54	1579,52	1,33	1540,59	1,44
An60k9	**59**	**9**	3019,03	2,42	2432,91	2,60	2613,80	2,68	2945,46	2,52	2671,64	2,32	2779,04	1,67	2632,26	2,29	2674,21	1,12	2664,22	1,94	2632,26	2,29	2657,38	2,32	2640,77	2,34
Bn35k5	**34**	**5**	1480,33	0,95	1386,39	0,93	1331,65	0,88	1719,72	0,86	1570,44	0,93	1984,82	0,75	1402,92	0,80	1314,03	0,95	1323,50	0,92	1402,92	0,80	1307,46	0,79	1294,41	1,01
Bn45k5	**44**	**5**	1585,54	1,52	1324,07	1,46	1268,82	1,53	1602,42	1,33	1522,09	1,21	1795,90	1,21	1363,95	1,28	1337,91	1,26	1264,28	1,26	1363,95	1,28	1297,26	1,37	1278,54	1,48
Bn68k9	**67**	**9**	1235,29	1,48	1257,34	1,37	1272,01	1,48	2934,14	2,40	2692,91	2,62	3085,10	2,57	2813,08	2,49	2753,08	2,58	2794,54	2,54	2813,08	2,49	2753,76	2,76	2768,87	2,98
Bn78k10	77	**10**	2805,79	2,64	2769,48	2,64	2740,64	2,75	2715,47	2,65	2731,13	2,63	2707,86	2,39	2756,91	2,66	2822,92	2,37	2779,57	1,46	2756,91	2,66	2794,92	1,34	2759,20	2,45
En30k3	**29**	**3**	788,01	1,17	698,62	1,16	725,91	1,20	906,28	1,04	814,32	1,11	1112,64	1,21	748,78	1,09	727,77	1,08	723,66	0,95	748,78	1,09	724,51	0,63	705,88	0,62
En51k5	**50**	**5**	2733,47	2,63	2799,47	2,82	2738,63	2,64	2785,90	2,75	2775,73	2,64	2728,47	2,72	2737,95	2,64	720,87	1,03	2734,29	2,72	2743,77	2,66	2729,36	2,68	2728,06	2,65
En76k7	**75**	7	2085,95	4,60	1686,14	4,63	1693,02	4,72	2079,14	4,10	1800,27	4,26	2154,35	4,54	1835,74	4,56	2743,53	2,71	1771,75	4,24	1835,74	4,56	1835,00	4,91	1793,68	4,97
		Mean	**1846,47**	**1,96**	**1673,57**	**1,97**	**1682,43**	**2,18**	**2083,71**	**1,99**	**1947,37**	**1,97**	**2114,76**	**1,97**	**1864,76**	**1,99**	**1738,43**	**1,46**	**1831,51**	**1,84**	**1865,34**	**1,99**	**1848,23**	**1,87**	**1831,00**	**2,09**

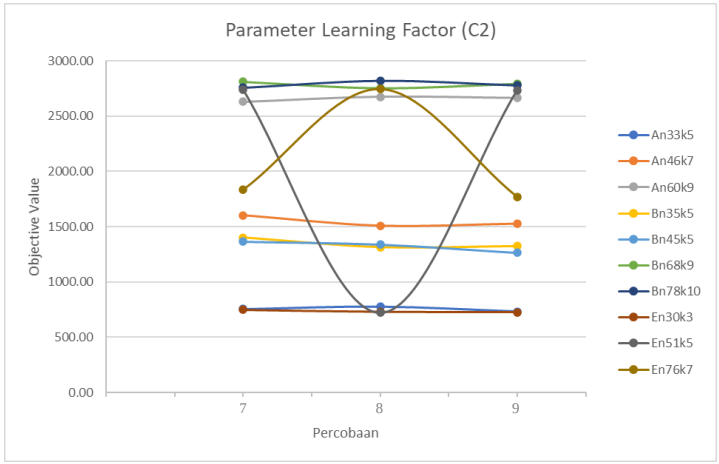


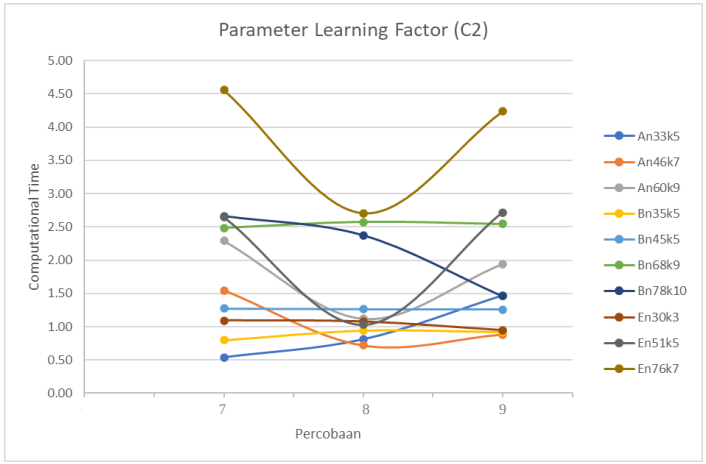
Gambar 2 Grafik Perbandingan Nilai Objektif dan Waktu Komputasi pada Faktor *Inertia Weight* (w)



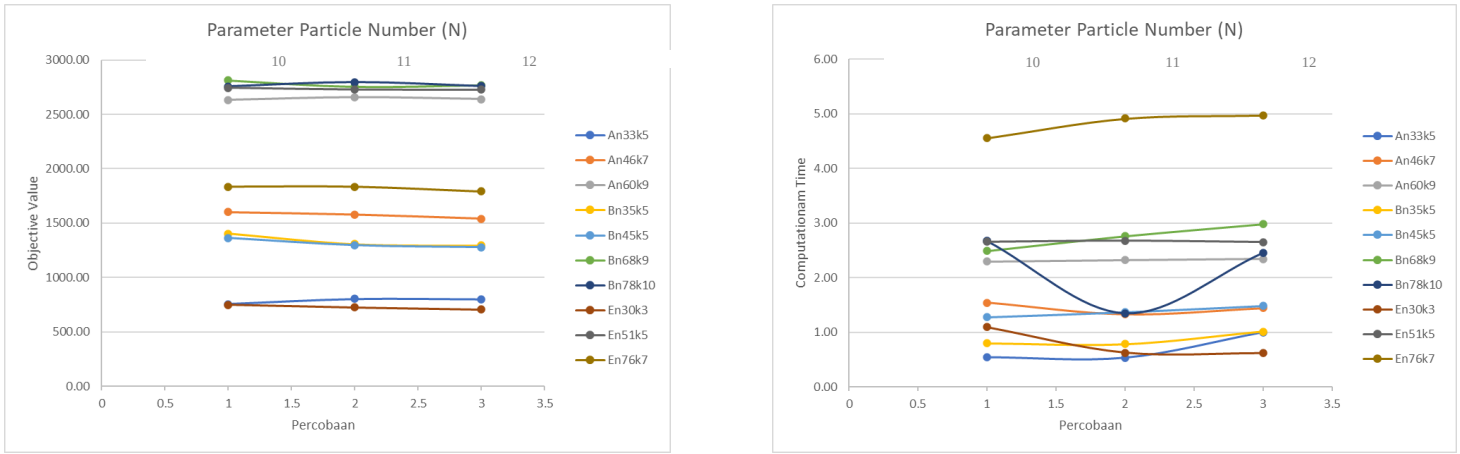


Gambar 3 Grafik Perbandingan Nilai Objektif dan Waktu Komputasi pada Faktor *Learning Factor* (c_1)





Gambar 4 Grafik Perbandingan Nilai Objektif dan Waktu Komputasi pada Faktor *Learning Factor* (c_1)



Gambar 5 Grafik Perbandingan Nilai Objektif dan Waktu Komputasi pada Faktor *Swarm Size* (w)

Proses selanjutnya adalah melakukan uji statistik untuk mengetahui perbedaan nilai rata-rata antara percobaan dalam satu faktor. Proses uji statistik untuk mengetahui perbedaan rata-rata antarpercobaan dalam satu faktor dilakukan dengan menggunakan Uji *Repeated Measures One Way* ANOVA. Uji statistik tersebut digunakan karena pada penelitian ini akan dibandingkan lebih dari 2 percobaan dengan *instances* yang sama. Sebelum melakukan uji *Repeated Measures One Way* ANOVA, dilakukan pengecekan asumsi yang harus dipenuhi melalui uji *normality* dan uji *homogeneity* dari variansi. Selanjutnya, uji *Repeated Measures One Way* ANOVA dilakukan dengan hipotesis sebagai berikut:

H_0: $\mu_A = \mu_B = \cdots = \mu_n \; ; n = jumlah\ percobaan$

$H_{1:}$ setidaknya terdapat satu rata-rata percobaan yang berbeda

Hipotesis di atas menjelaskan bahwa ketika nilai signifikansi P-*value* kurang dari level signifikansi yaitu 0,05, maka keputusan yang diambil yaitu tolak H_o yang berarti terdapat perbedaan rata-rata pada satu atau lebih percobaan. Namun, sebaliknya bila nilai signifkansi P-*value* lebih dari 0,05, maka keputusan yang diambil adalah gagal tolak H_0 yang berarti tidak terdapat perbedaan rata-rata antarpercobaan. Uji *Repeated Measures One Way* ANOVA dilakukan untuk keempat faktor parameter. Hasil yang didapatkan adalah:

- Faktor *inertia weight* (w), tidak terdapat perbedaan rata-rata yang signifikan diantara percobaan 1, 2, 3, dan 7.
- Faktor *cognitive learning* (c_1), terdapat perbedaan rata-rata yang signifikan pada beberapa percobaan. Perbedaan rata-rata terdapat antara percobaan 4 dan 5, percobaan 4 dan 7, percobaan 6 dan 7, serta percobaan 5 dan 6.
- Faktor *cognitive learning* (c_2), tidak terdapat perbedaan rata-rata yang signifikan diantara percobaan 7, 8, dan 9.
- Faktor *swarm size*, terdapat perbedaan rata-rata diantara percobaan yang dilakukan. Perbedaan rata-rata terdapat pada percobaan 10 dan 12, serta percobaan 11 dan 12.

Hasil uji hipotesis di atas kemudian digunakan untuk penentuan pearameter terbaik yang akan diimplementasikan. Faktor *inertia weight* dan *cognitive learning* pada dasarnya tidak terpengaruh oleh perubahan nilai parameter yang diujicobakan. Namun, penentuan parameter terbaik dilakukan dengan mempertimbangkan rata-rata nilai optimal dan rata-rata waktu komputasi yang dibutuhkan. Parameter terpilih merupakan percobaan dengan parameter yang mampu menghasilkan hasil yang optimal dengan waktu yang efisien. Percobaan pada faktor *inertia weight* dengan nilai objektif terkecil adalah percobaan 2 yaitu sebesar 1673,57, sedangkan waktu komputasi terkecil adalah percobaan 1 yaitu 1,96. Namun, dikarenakan perbedaan waktu komputasi antara percobaan 2 dan percobaan 1 hanya selisih 0,01, maka parameter terpilih adalah percobaan 2 dengan nilai *inertia weight* sebesar 0,3. Selanjutnya, untuk faktor *social learning* (c_2) juga tidak terdapat perbedaan rata-rata yang signifikan. Nilai objektif terendah didapatkan pada percobaan 8 yaitu 1738,43, sedangkan waktu komputasi tercepat didapatkan percobaan 8 dengan rata-rata waktu 1,46. Oleh karena itu, parameter terpilih adalah percobaan 8 dengan nilai parameter $c_2 = 1{,}5$.

Selanjutnya, parameter *cognitive learning* (c_1) dan *swarm size* dapat dilihat pada hasil uji *Repeated Measures One Way* ANOVA tabel *test of within subject effect* menunjukan terdapat percobaan yang memiliki perbedaan rata-rata dibanding percobaan lainnya. Namun, hasil uji hipotesis hanya dapat menyimpulkan bahwa setidaknya terdapat satu nilai rata-rata yang berbeda. Oleh karena itu, penentuan parameter terpilih didasarkan pada perbedaan rata-rata percobaan yang tertulis pada kolom Mean Difference(i-j) di Tabel *Pairwise Comparison*. Berdasarkan tabel tersebut, ditentukan parameter terbaik untuk *cognitive learning* (c_2) yaitu 1,5 dan *swarm size* yaitu 50.

3. Hasil Implementasi DPSO

Parameter terpilih yang telah ditentukan pada subbab sebelumnya kemudian dijadikan parameter terpilih dalam implementasi algoritma DPSO. Algoritma tersebut digunakan untuk menyelesaiakn sepuluh *instance* yang tertulis pada **Tabel 1**. Setiap *instance* dilakukan replikasi sebanyak 10 kali dengan nilai parameter w = 1,3, c_1 = 1,5, c_2 = 1,5, dan N = 50. Kemudian, komputasi dilakukan untuk jumlah iterasi maksimal dengan waktu komputasi maksimal sama dengan percobaan SR-1 oleh (Ai & Kachitvichyanukul, 2009). Hal ini dikarenakan hasil percobaan DPSO kemudian akan diabndingkan dengan hasil komputasi DPSO-SA oleh (Chen, Yang , & Wu, 2006), SR-1 dan SR-2 oleh (Ai & Kachitvichyanukul, 2009). Diantara ketiga percobaan tersebut, rata-rata waktu komputasi yang dihasilkan algoritma SR-1 paling rendah jika dibandingkan dua algoritma lainnya. Oleh karena itu, komputasi DPSO menggunakan waktu SR-1 dengan tujuan untuk mengetahui dengan waktu tersingkat apakah terdapat perbedaan rata-rata yang signifikan dari algoritma DPSO dan algoritma lainnya. Perbandingan hasil komputasi antar algoritma secara rinci terdapat pada **Tabel 4**.

Tabel 4 Perbandingan Hasil Komputasi

Instances	No. Cust	No Veh.	Objective Function					Computational Time				
			BKS	Chen	SR-1	SR-2	DPSO	BKS	Chen	SR-1	SR-2	DPSO
An33k5	32	5	661	661	661	661	661,35	11	32	11	13	11
An46k7	45	7	914	914	914	914	1554,89	19	129	19	23	19
An60k9	59	9	1354	1354	1366	1355	2246,50	28	309	28	40	28
Bn35k5	34	5	955	955	955	955	1271,62	12	38	12	14	12
Bn45k5	44	5	751	751	751	751	1319,58	17	134	17	20	17
Bn68k9	67	9	1272	1272	1278	1274	2409,63	33	344	33	50	33
Bn78k10	77	10	1223	1239	1239	1223	2487,45	41	429	41	64	41
En30k3	29	3	534	534	541	534	779,13	11	28	11	16	11
En51k5	50	5	521	528	521	521	950,24	21	301	21	22	21
En76k7	75	7	682	688	691	682	1481,93	38	527	38	60	38

Pada **Tabel 4** dapat dilihat bahwa nilai objektif dari algoritma Chen, SR-1, dan SR-2 menghasilkan nilai yang sama untuk beberapa *instances* dan menunjukkan nilai berbeda yang tidak terlalu jauh pada *instances* lainnya. Sedangkan nilai objektif yang dihasilkan algoritma DPSO dengan waktu komputasi minimal ternyata belum mampu

mencapai nilai terbaik dari algoritma lainnya. Selanjutnya, dilakukan proses uji performansi metaheuristik menggunakan tes statistik untuk mengetahui perbedaan nilai rata-rata dari empat algoritma tersebut dan mengetahui algoritma yang memiliki penyelesaian lebih baik untuk data yang sama.

Uji statistik yang digunakan untuk menganalisis performansi algoritma pada penelitian ini adalah Uji *Repeated Measure One Way* ANOVA. Uji ini dipilih karena bertujuan untuk membandingkan lebih dari 2 model algoritma namun dengan menggunakan data *instances* yang sama. Uji *Repeated Measure One Way* ANOVA dapat dilakukan saat data memenuhi asumsi terdistribusi normal dan memiliki variansi yang setara. Deskripsi statistik data dapat dilihat pada **Tabel 5** sedangkan uji normalitas data dapat dilihat secara rinci pada **Tabel 6**.

Tabel 5 Statistik Deskriptif Data

Descriptive Statistics

	Mean	Std. Deviation	N
Chen	889,60000	308,790904	10
SR1	891,7000	311,45146	10
SR2	887,0000	308,61771	10
DPSO	1516,2320	664,19403	10

Tabel 6 Uji Normalitas Data

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Algoritma	Statistic	df	Sig.	Statistic	df	Sig.
Objective	Chen	,173	10	,200*	,902	10	,231
	SR-1	,174	10	,200*	,907	10	,264
	SR-2	,170	10	,200*	,908	10	,270
	DPSO	,177	10	,200*	,912	10	,297

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Uji normalitas data dilakukan bertujuan untuk membuktikan bahwa data yang dimiliki terdistribusi dengan normal. Hipotesis pada uji normalitas ini adalah sebagai berikut:

H_0: data terdistribusi normal
H_A: data tidak terdistribusi normal

Berdasarkan data **Tabel 6** dapat dilihat bahwa nilai signifikansi masing-masing model baik untuk Kolmogorov-Smirnov Test maupun Shapiro-Wilkin Test bernilai lebih dari 0,05. Oleh karena itu, keputusan yang dapat diambil adalah gagal tolak H_0 yang berarti membuktikan data untuk setiap model terdistribusi dengan normal. Selanjutnya, pada Uji *Repeated Measure One Way* ANOVA dengan menggunakan SPSS mengahsilkan beberapa tabel sebagai berikut:

Tabel 7 Mauchly's Spericity Test

Mauchly's Test of Sphericity[a]

Measure: MEASURE_1

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b] Greenhouse-Geisser	Huynh-Feldt	Lower-bound
model	,000	117,584	5	,000	,333	,334	,333

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: model

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Mauchly's Spericity Test merupakan pengukuran yang digunakan untuk validasi pada *repeated measures* ANOVA. *Sphericity* merupakan asumsi penting pada uji *repeated measure* ANOVA yang mengukur apakah terdapat perbedaan variansi dari seluruh kemungkinan pasangan *independent variable*. Hipotesis pada uji Mauchly's Spericity Test adalah sebagai berikut:

H_0: variansi data *setara*
H_A: variansi data tidak setara

Berdasarkan **Tabel 7** dapat dilihat bahwa nilai P-*value* bernilai 0,000 dimana nilai tersebut kurang dari level signifikansi yang bernilai 0,05. Oleh karena itu, keputusan yang dapat diambil adalah tolak H_0 yang berarti variansi data tidak setara.
Pada Uji *repeated measure* ANOVA, apabila hasil Mauchly's Spericity Test tidak memenuhi asumsi variansi yang setara, maka penentuan hasil uji *repeated measure* ANOVA didasarkan pada nilai perbaikan *Greenhouse-Geisser*. *Greenhouse-Geisser* merupakan metode yang digunakan untuk memperbaiki nilai *degree of freedom* sehingga mengurangi tingkat Type I Error. Pengambilan keputusan dari uji *repeated measure* ANOVA didasarkan pada Tabel *Tests of Within Subjects Effects* sebagai berikut:

Tabel 8 Hasil Uji *Repeated Measures* ANOVA

Tests of Within-Subjects Effects

Measure: MEASURE_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
model	Sphericity Assumed	2946685,131	3	982228,377	24,763	,000	,733
	Greenhouse-Geisser	2946685,131	1,000	2945545,768	24,763	,001	,733
	Huynh-Feldt	2946685,131	1,001	2945118,659	24,763	,001	,733
	Lower-bound	2946685,131	1,000	2946685,131	24,763	,001	,733
Error(model)	Sphericity Assumed	1070954,622	27	39664,986			
	Greenhouse-Geisser	1070954,622	9,003	118948,948			
	Huynh-Feldt	1070954,622	9,005	118931,700			
	Lower-bound	1070954,622	9,000	118994,958			

Tabel di atas merupakan *output* yang sangat penting dalam Uji *Repeated Measures* ANOVA. Pada penelitian ini, uji *repeated Measures* ANOVA memiliki hipotesis sebagai berikut:

H_0: tidak terdapat perbedaan rata-rata antar model algoritma
H_A: terdapat perbedaan rata-rata antar model algoritma

Seperti yang kita ketahui melalui bahwa dasar pengambilan keputusan *repeated measures* ANOVA pada penelitian ini merujuk pada nilai *Greenhouse-Geisser*. Pada **Tabel 8** diketahui bahwa nilai signifikansi bernilai 0,001 yang berarti keputusan yang dapat diambil adalah tolak H_0. Oleh karena itu dapat disimpulkan bahwa berdasarkan uji *repeated measures* ANOVA terdapat perbedaan rata-rata antar model algoritma yang diuji. Selanjutnya, untuk mengetahui rata-rata model yang

berbeda serta menentukan model mana dengan rata-rata yang lebih baik dapat dilakukan dengan melihat *Pairwise Comparison* seperti yang terdapat pada **Tabel 9.**

Tabel 9 Analisis *Post-hoc*

Pairwise Comparisons

Measure: MEASURE_1

(I) model	(J) model	Mean Difference (I-J)	Std. Error	Sig.[b]	95% Confidence Interval for Difference[b]	
					Lower Bound	Upper Bound
1	2	-2,100	1,643	1,000	-7,627	3,427
	3	2,600	1,759	1,000	-3,317	8,517
	4	-626,632*	125,861	,005	-1050,053	-203,211
2	1	2,100	1,643	1,000	-3,427	7,627
	3	4,700	1,832	,183	-1,464	10,864
	4	-624,532*	125,383	,005	-1046,347	-202,717
3	1	-2,600	1,759	1,000	-8,517	3,317
	2	-4,700	1,832	,183	-10,864	1,464
	4	-629,232*	126,598	,005	-1055,132	-203,332
4	1	626,632*	125,861	,005	203,211	1050,053
	2	624,532*	125,383	,005	202,717	1046,347
	3	629,232*	126,598	,005	203,332	1055,132

Based on estimated marginal means

*. The mean difference is significant at the ,05 level.

b. Adjustment for multiple comparisons: Bonferroni.

Berdasarkan tabel di atas, dapat dilihat pada kolom signifkansi yang membuktikan bahwa terdapat perbedaan rata-rata pada pasangan model Chen-model DPSO, model SR1-model DPSO, model SR2-model DPSO. Kemudian untuk mengetahui rata-rata mana yang berbeda serta model mana yang memiliki rata-rata lebih baik dibandingkan model lainnya dapat dilihat pada kolom *Mean Difference* (I-J). *Pairwise Comparison* ini didapatkan dari analisis *post-hoc* dengan metode *Bonferroni Procedure*. Pada **Tabel 9** diketahui bahwa model Chen, SR1, dan SR2 lebih baik dibandingkan model DPSO. Selain itu, algoritma SR2 memiliki performansi lebih baik dibandingkan dengan algoritma Chen dan SR1.

KESIMPULAN

Penelitian ini menjelaskan mengenai penggunaan algoritma DPSO untuk permasalahan CVRP menggunakan PSO. Implementasi DPSO pada seluruh *instances* dilakukan dengan menggunakan parameter terpilih yang dihasilkan dari *parameter tuning* menggunakan metode OFAT. Hasil yang didapatkan kemudian dibandingkan dengan Chen, SR-1, dan SR-2. *Parameter tuning* yang dilakukan menghasilkan parameter terbaik yaitu nilai *inertia weight* 0,3, nilai *cognitive learning* 1,5, *social learning* 1,5, serta *swarm size* 50.

Hasil komputasi menggunakan DPSO menunjukkan bahwa rata-rata yang didapatkan dengan waktu minimal belum mampu setara dengan ketiga algoritma lainnya. Nilai obyektif optimal menggunakan DPSO hanya menunjukkan kemiripan dengan nilai objektif terbaik pada satu *instances* yaitu A-n33-k5. Hasil uji statistik menunjukkan terdapat perbedaan rata-rata nilai objektif. Berdasarkan analisis *Post-hoc* dapat disimpulkan bahwa algoritma DPSO dengan parameter yang dipilih dan waktu komputasi terkecil belum mampu menghasilkan hasil yang optimal dibandingkan dengan algoritma Chen, SR-1, dan SR-2. Algoritma DPSO tidak lebih baik dibandingkan algoritma Chen, SR-1, dan SR-2. Selain itu, algoritma SR2 memiliki performansi lebih baik dibandingkan dengan algoritma Chen dan SR1.

REFERENSI

Ai, T., & Kachitvichyanukul, V. (2009). Particle Swarm Optimization and Two Solution Representations for Solving The Capacitated Vehicle Routing Problem. *Computers & Industrial Engineeringq*, 380-387.

Augerat, P., Belenguer, J., Benavent, E., Corberan, A., & Naddef, D. (1998). Separating Capacity Constraints in The CVRP using Tabu Search. *European Journal of Operational Research*, 546-557.

Baker, B., & Ayechew, M. (2003). A Genetic Algorithm for The Vehicle Routing Problem . *Computers & Operations Research, 30*, 787-800.

Bodin, L., Golden, B., Assad, A., & Ball, M. (1983). The State of The Art in Routing and Scheduling of Vehicles and Crews. *Computers and Operation Research*, 69-221.

Chen, A., Yang , G., & Wu, Z. (2006). Hybrid Discrete Particle Swarm Optimization Algorithm for Capacitated Vehicle Routing Problem. *Journal of Xheijiang University Science A*, 607-614.

Dantzig, G., & Ramser, J. (1959). The Truck Dispatching Problem. *Management Science, 6*, 80-91.

Eberhart, R., & Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. *Proceeding of Sixth International Symposium on Micro Machine and Human Science*, 39-43.

Eberhart, R., & Shi, Y. (2001). Particle Swarm optimizations: developments, applications and resources. *Proceedings on Congress of Evolutionary Computation*, (hal. 81-86). Seoul, Korea.

Gunawan, V. (2017). Studi Tnetang Pengaruh Nilai Pelanggan Inti dan Peripheral Terhadap Minat Menggunakan Kartu Kredit untuk Meningkatkan Keputusan Menggunakan Kartu Kredit.

Haimovich, M., Rinnooy Kan, A., & Stougie, L. (1988). Analysis of heuristics for vehicle routing problems.

Hannan, M., Akhtar, M., Begum, R., Basri, H., Hussain, A., & Scavino, E. (2017). Capacitated Vehicle Routing Problem Model for Scheduled Solid Waste Collection and Route Optimization using PSO Algorithm. *Wste Management*.

I. C. Trelea. (2003). The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection. *Information Processing Letters, 85*(6), 317-325.

Irnich, S., & Vigo, D. (2014). Chapter 1: The Family of Vehicle Routing Problems. *Research Gate*, 3.

Isiet, M., & Gadala, M. (2020). Sensitivity Analysis of Control Parameters in Particle Swarm Optimization. *Journal of Computational Science*.

J.S.Arora. (2017). *Introduction to Optimum Design* (4th ed.). Elsevier, Ed.

Jamaludin, F. (2018). Mitigasi Resiko Kredit Perbankan. *Journal of Islamic Economic Law*.

Kuo, R., Zulvia, F., & Suryadi, K. (2012). Hybrid Particle Swarm Optimization with Genetic Algorithm for Solving Capacitated Vehicle Routing Problem with Fuzzy Demand - A Case Study on Garbage Collection System. *Applied Mathematics and Computation, 219*, 2574-2588.

Lindfield, G., & Penny, J. (2017). Particle Swarm Optimization Algorithm. Dalam *Introduction to Nature-Inspired Optimizatiom* (hal. 49-68).

M. Pant, R. Thangaraj, & A. Abraham. (2009). Particle swarm optimization: Performance tuning and empirical analysis. *Foundations of Computational Intelligence, 3*, 101-128.

Marinakis, Y., Iordanidou, G.-R., & Marinaki, M. (2013). Particle Swarm Optimization for The Vehicle Routing Problem with Stochastic Demands. *Applied Soft Computing, 13*, 1693-1704.

Sari, L. K. (t.thn.). Penerapan Manajemen Resiko pada Perbankan Indonesia.

Sombutham, P., & Kachitvichayanukul, V. (2010). A particle swarm optimization algorithm for multidepot vehicle routing problem with pickup and delivery request. *Proceedings of The International Multi Conference of Engineers and Computer Sciencetists*, (hal. 71-85). Hongkong.

Talbi, E.-G. (2009). *Metaheuristics From Design to Implementation.* New Jersey: John Wiley & Sons.

Waluyo, B., Dedy, & Agung, J. (2013). *Menjaga Stabilitas, Mendorong Reformasi Struktural untuk Pertumbuhan Ekonomi yang Berkelanjutan.* Bank Indonesia.

Wang, H., Geng, Q., & Qiao, Z. (2013). Parameter Tuning of Particle Swarm Optimization by Using Taguchi Method and Its Aplication to Motor Design.

Wijayanti, R., & Sulastri. (2018). Analisa Klasifikasi Kartu Kredit Menggunakan Algoritma Naive Bayes. *Prosiding Sintak*.

Y.Shi, & R. Eberhart. (1998). A Modified Particle Swarm Optimizer. *IEEE World Congress on Computational Intelligence* (hal. 69-73). Anchorage, Alaska, USA: IEEE International Conference on Evolutionary Computation Proceedings.

Yang, X. (2014). Particle Swarm Optimization. Dalam *Nature-Inspired Optimization Algorithms* (hal. 99-110).

Yeh, I.-C., & Lien, C.-h. (2009). The Comparisons of Data Mining Techniques for The Predictive Accuracy of Probability of Default of Credit Card Clients. *Expert System with Application*, 2473-2480. Diambil kembali dari www.sciencedirect.com

Yung, S. (2006). Manajemen Resiko dalam Dunia Perbankan. *Jurnal Sistem Informasi UKM, 1*, 63-71.

Yunindya, R., Kudus, A., & Yanti, T. S. (2016). Model Credit Scoring Menggunakan Metode Classification dan Regression Trees (CART) pada Data Kartu Kredit. *Prosiding Statistika*.