

# Algoritma Genetika Ganda (AGG) untuk *Capacitated Vehicle Routing Problem* (CVRP)

Daryono Budi Utomo, Mohammad Isa Irawan, Muhammad Luthfi Shahab

Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember  
[daryono@matematika.its.ac.id](mailto:daryono@matematika.its.ac.id)

**Abstrak**—*Capacitated vehicle routing problem* (CVRP) merupakan suatu permasalahan penyaluran barang, dengan kendaraan berkapasitas tertentu, dari satu depot ke beberapa pelanggan, yang bertujuan untuk mencari rute yang meminimalkan total jarak yang ditempuh. Contoh nyata CVRP adalah permasalahan penyaluran bahan bakar minyak ke Stasiun Pengisian Bahan Bakar Umum (SPBU). Ada banyak metode yang telah dipakai untuk menyelesaikan CVRP, namun penggunaan algoritma genetika (AG) masih belum memberikan hasil yang memuaskan. Namun keberhasilan AG untuk menyelesaikan banyak permasalahan lain menunjukkan bahwa penggunaan AG akan memberikan hasil yang semakin baik apabila terus diteliti. Untuk mempermudah menyelesaikan CVRP, dapat dilakukan dekomposisi agar CVRP terbagi menjadi beberapa permasalahan yang lebih sederhana yang dapat diselesaikan secara independen. Berdasarkan hal tersebut, dirumuskan algoritma genetika ganda (AGG) yang terlebih dahulu berusaha untuk mendekomposisi CVRP dan kemudian mencari rute terpendek pada setiap permasalahan dengan menggunakan dua algoritma genetika sederhana berbeda. Untuk mengetahui seberapa baik AGG dalam menyelesaikan CVRP, dilakukan perbandingan antara AGG dengan AG. Perbandingan AGG dengan AG dilihat dari tiga segi, yaitu jarak, waktu komputasi, dan total generasi. Agar mendapatkan hasil yang akurat, perbandingan dilakukan dengan membuat empat permasalahan yaitu P50, P75, P100, dan P125, dan untuk setiap permasalahan digunakan empat belas variasi kapasitas kendaraan yang berbeda. Didapatkan hasil bahwa untuk segi waktu komputasi dan total generasi, AGG lebih baik dari AG. Sedangkan dari segi jarak, AGG juga lebih baik dari AG kecuali untuk beberapa kapasitas kendaraan yang kecil pada permasalahan P50 dan P75. Hasil juga menunjukkan bahwa penambahan kapasitas kendaraan mengakibatkan AGG menjadi jauh lebih baik dibandingkan dengan AG.

**Kata kunci:** *algoritma genetika (AG), algoritma genetika ganda (AGG), capacitated vehicle routing problem (CVRP)* PM -

## I. PENDAHULUAN

*Vehicle routing problem* (VRP) adalah salah satu permasalahan optimasi kombinatorial yang memiliki banyak aplikasi pada bidang industri [6]. VRP dapat digambarkan sebagai permasalahan penyaluran barang dari depot-depot ke pelanggan-pelanggan tertentu yang bertujuan untuk mencari suatu rute yang meminimalkan total jarak yang ditempuh. VRP memiliki banyak variasi disesuaikan pada batasan-batasan yang digunakan. Salah satu dari variasi tersebut adalah *capacitated vehicle routing problem* (CVRP) yang hanya memiliki satu depot dan menggunakan batasan kapasitas pada kendaraan yang dipakai dalam penyaluran barang. Contoh nyata CVRP di dunia nyata diantaranya adalah permasalahan penyaluran bahan bakar minyak ke Stasiun Pengisian Bahan Bakar Umum (SPBU).

Banyaknya aplikasi dari CVRP yang sesuai dengan permasalahan di dunia nyata mengakibatkan CVRP menjadi salah satu bidang ilmu yang banyak diteliti [9]. Penelitian-penelitian untuk menyelesaikan CVRP tersebut dilakukan dengan metode-metode yang berbeda. Metode-metode *metaheuristic* lebih sering digunakan karena dapat menyelesaikan CVRP dengan hasil yang cukup baik dan waktu komputasi yang lebih singkat. Beberapa metode *metaheuristic* yang dapat digunakan antara lain adalah *variable neighborhood search*, *greedy randomized adaptive search procedure*, *stochastic local search*, *iterated local search*, *particle swarm optimization*, *scatter search*, *differential evolution*, *ant colony system*, *simulated annealing*, *tabu search*, dan algoritma genetika (AG) [5].

Toth dan Vigo [8] menyatakan bahwa penggunaan AG untuk menyelesaikan CVRP masih belum memberikan hasil yang memuaskan. Namun keberhasilan AG untuk menyelesaikan permasalahan-permasalahan lain seperti *travelling salesman problem (TSP)*, *vehicle routing problem with time windows (VRPTW)*, dan lain-lain menunjukkan bahwa penggunaan AG akan memberikan hasil yang semakin baik jika terus diteliti. Untuk mempermudah menyelesaikan CVRP, Taillard [7] melakukan dekomposisi pada CVRP agar terbagi menjadi beberapa permasalahan yang lebih sederhana yang dapat diselesaikan secara independen. Dalam penelitiannya, penggabungan *tabu search* dengan dekomposisi CVRP yang dilakukan telah memberikan hasil yang sangat baik pada empat belas permasalahan klasik CVRP dan banyak dari hasil tersebut masih tetap menjadi yang terbaik hingga saat ini [6].

Berdasarkan hal tersebut, akan dirumuskan algoritma genetika ganda (AGG) yang bekerja dengan terlebih dahulu berusaha untuk mendekomposisi CVRP menjadi beberapa daerah yang independen dan kemudian mencari rute terpendek pada setiap daerah menggunakan dua algoritma genetika sederhana yang berbeda. AGG tersebut diharapkan dapat digunakan untuk menyelesaikan CVRP dengan hasil yang baik. AGG akan dirumuskan secara runtut dan akan dibandingkan dengan AG untuk mengetahui seberapa baik AGG dapat digunakan untuk menyelesaikan CVRP.

## II. METODE PENELITIAN

Penelitian diawali dengan membuat beberapa CVRP. Beberapa CVRP tersebut dibuat agar AG dan AGG dapat diimplementasikan untuk menyelesaikannya. Banyaknya CVRP yang dibuat dalam penelitian ini ada sebanyak empat permasalahan.

Setelah CVRP selesai dibuat, akan dirumuskan AG dan AGG yang dapat digunakan untuk menyelesaikan CVRP. Untuk setiap algoritma genetika, yang akan dirumuskan adalah representasi kromosom, besar populasi, fungsi *fitness*, operator seleksi, operator *crossover*, operator mutasi, skema penggantian populasi, dan kondisi pemberhentian.

Selanjutnya dirancang suatu program yang akan digunakan sebagai alat untuk melakukan perbandingan antara AGG dan AG. Perbandingan antara AGG dan AG akan dilihat dari tiga segi, yaitu dari segi jarak, waktu komputasi, dan total generasi yang dibutuhkan untuk menyelesaikan CVRP.

## III. HASIL DAN PEMBAHASAN

### A. Pembuatan CVRP

Dibuat empat CVRP yaitu P50, P75, P100, dan P125. Absis, ordinat, dan permintaan dari setiap pelanggan pada CVRP yang dibuat, dipilih secara acak dari suatu rentang tertentu. Permasalahan P50 terdiri dari depot dan 50 pelanggan yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 50, sedangkan absis dan ordinat pelanggan bernilai antara 0 sampai 100. Setiap pelanggan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua pelanggan adalah 1040. Permasalahan P75 terdiri dari depot dan 75 pelanggan yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 75, sedangkan absis dan ordinat pelanggan bernilai antara 0 sampai 150. Setiap pelanggan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua pelanggan adalah 1413. Permasalahan P100 terdiri dari depot dan 100 pelanggan yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 100, sedangkan absis dan ordinat pelanggan bernilai antara 0 sampai 200. Setiap pelanggan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua pelanggan adalah 2044. Permasalahan P125 terdiri dari depot dan 125 pelanggan yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 125, sedangkan absis dan ordinat pelanggan bernilai antara 0 sampai 250. Setiap pelanggan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua pelanggan adalah 2472.

### B. Perumusan AG untuk CVRP

Sebelum merumuskan AGG, terlebih dahulu akan dirumuskan AG yang dapat digunakan untuk menyelesaikan CVRP. AG tersebut dirumuskan dengan karakteristik sebagai berikut:

- Representasi kromosom yang digunakan adalah permutasi dari pelanggan-pelanggan. Setiap kromosom yang terbentuk adalah unik dan setiap kromosom hanya bisa merepresentasikan satu solusi CVRP. Sebagai contoh, apabila permasalahan CVRP yang digunakan terdiri dari 9 pelanggan, salah satu kromosom yang dapat digunakan adalah  $v_1 v_6 v_8 v_5 v_3 v_2 v_7 v_4 v_9$ . Untuk merubah kromosom

tersebut menjadi solusi yang diinginkan, digunakan informasi mengenai kapasitas kendaraan dan permintaan dari setiap pelanggan. Misalkan kapasitas kendaraan dalam permasalahan adalah 17 dan permintaan dari setiap pelanggan  $v_i$  adalah  $i$ ,  $i = 1, \dots, 9$ , maka rute pertama adalah  $R_1$  yaitu  $v_0 \rightarrow v_1 \rightarrow v_6 \rightarrow v_8 \rightarrow v_0$ , rute kedua adalah  $R_2$  yaitu  $v_0 \rightarrow v_5 \rightarrow v_3 \rightarrow v_2 \rightarrow v_7 \rightarrow v_0$ , dan rute ketiga adalah  $R_3$  yaitu  $v_0 \rightarrow v_4 \rightarrow v_9 \rightarrow v_2$ .

- Besar populasi yang digunakan adalah 100.
- Misalkan dari suatu kromosom didapatkan rute-rute  $R_i$ ,  $i = 1, \dots, n$ , dimana  $R_i$  adalah  $v_{i,0} \rightarrow v_{i,1} \rightarrow \dots \rightarrow v_{i,k_i+1}$  dengan  $v_{i,0} = v_{i,k_i+1} = v_0$ , maka fungsi *fitness* yang dapat digunakan adalah

$$f(X, Y) = \sum_{i=1}^n \sum_{j=1}^{k_i+1} \sqrt{(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2} \quad (1)$$

dimana  $x_{i,j}$  dan  $y_{i,j}$  adalah absis dan ordinat dari  $v_{i,j}$ ,  $X$  adalah matriks dengan elemen pada baris ke- $i$  dan kolom ke- $j$  adalah  $x_{i,j-1}$ , dan  $Y$  adalah matriks dengan elemen pada baris ke- $i$  dan kolom ke- $j$  adalah  $y_{i,j-1}$ . Semakin kecil nilai *fitness* tersebut, maka semakin baik kromosom tersebut.

- Seleksi dilakukan dengan memilih dua kromosom secara acak.
- Operator *crossover* yang digunakan adalah *ordered crossover* (OX) dengan kemungkinan terjadinya *crossover* adalah 1,0.
- Operator mutasi yang digunakan adalah *exchange* dan *inversion* dengan kemungkinan terjadinya mutasi masing-masing adalah 0,1.
- Skema penggantian populasi yang digunakan adalah *elitism replacement with filtration*.
- Kondisi pemberhentian yang digunakan adalah tidak bertambahnya *fitness* selama 2000 generasi atau banyaknya generasi telah mencapai 100000.

### C. Perumusan AGG untuk CVRP

AGG bekerja dengan menggabungkan dua algoritma genetika sederhana berbeda agar dapat digunakan untuk menyelesaikan CVRP dengan cara yang berbeda dari AG. AG berusaha untuk menyelesaikan CVRP secara langsung, sedangkan AGG akan terlebih dahulu berusaha untuk mendekomposisi CVRP menjadi beberapa daerah yang independen dengan AGG1 (algoritma genetika pertama dalam AGG) dan kemudian mencari rute terpendek pada setiap daerah yang terbentuk oleh AGG1 dengan AGG2 (algoritma genetika kedua dalam AGG). Daerah-daerah yang terbentuk dari hasil dekomposisi yang dilakukan oleh AGG1 haruslah memenuhi dua karakteristik sebagai berikut:

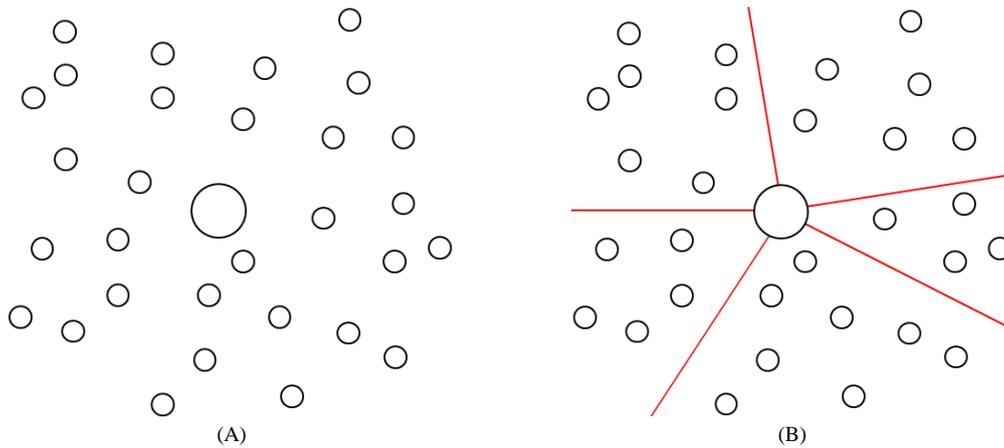
- Setiap daerah hanya membutuhkan satu kendaraan untuk melayani setiap pelanggan yang ada dalam daerah tersebut. Dengan kata lain, total permintaan pelanggan yang ada dalam setiap daerah tidak melebihi kapasitas kendaraan.
- Pelanggan-pelanggan yang terletak dalam suatu daerah harus terletak saling berdekatan.

Karakteristik yang pertama diambil dari sifat dasar CVRP yang menyatakan bahwa setiap rute yang terbentuk dalam solusi CVRP harus dilayani oleh satu kendaraan. AGG1 akan berusaha memenuhi karakteristik pertama tersebut dengan mempertimbangkan kapasitas kendaraan dan permintaan dari setiap pelanggan. Karakteristik yang kedua dibuat agar nantinya solusi CVRP yang terbentuk akan menjadi cukup baik. AGG1 akan berusaha memenuhi karakteristik kedua tersebut dengan mempertimbangkan kemiringan garis yang menghubungkan pelanggan dengan depot. Dalam hal ini, digunakannya kemiringan garis didasarkan pada kenyataan bahwa apabila kemiringan antara dua garis saling berdekatan, maka pelanggan-pelanggan yang ada pada garis tersebut juga akan cukup berdekatan.

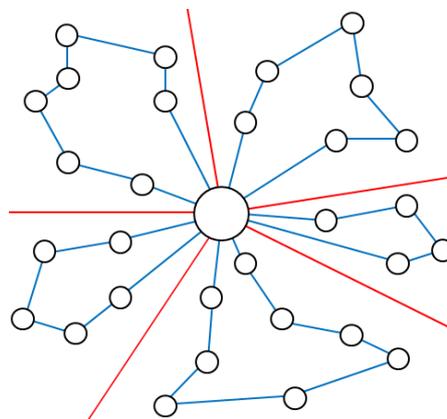
Perhatikan contoh CVRP sederhana dalam GAMBAR 1(A) dimana lingkaran besar merepresentasikan depot dan lingkaran-lingkaran kecil merepresentasikan pelanggan-pelanggan. Dari permasalahan tersebut, salah satu dekomposisi yang bisa dihasilkan oleh AGG1 dapat dilihat pada GAMBAR 1(B).

Dengan dilakukannya dekomposisi oleh AGG1, solusi untuk setiap daerah yang terbentuk akan menjadi solusi untuk CVRP. Solusi untuk setiap daerah yang terbentuk adalah rute terpendek yang berangkat dari depot, kemudian menghubungkan setiap pelanggan yang ada dalam daerah, dan kemudian kembali lagi ke depot. Pemilihan rute tersebut akan dilakukan oleh AGG2. Perhatikan bahwa karena setiap daerah yang terbentuk dari hasil dekomposisi hanya membutuhkan satu kendaraan untuk melayani setiap pelanggan yang ada dalam daerah tersebut, maka informasi mengenai permintaan dari setiap pelanggan dapat dihilangkan sehingga permasalahan pemilihan rute terpendek pada setiap daerah dapat disebut sebagai *travelling salesman problem* (TSP). Oleh karena itu, AGG2 akan mencari rute terpendek pada setiap daerah dengan hanya mempertimbangkan lokasi depot dan lokasi pelanggan. Apabila rute

terpendek untuk setiap daerah telah didapatkan dengan AGG2, maka setiap rute tersebut akan digabung menjadi satu sehingga menjadi solusi dari CVRP. Contoh CVRP pada GAMBAR 1(A) yang telah didekomposisi seperti pada Gambar 1(B), penggabungan rute yang telah diperoleh dengan AGG2 dapat dilihat pada GAMBAR 2.



GAMBAR 1. (A) CONTOH CVRP DAN (B) CONTOH DEKOMPOSISI OLEH AG1



GAMBAR 2. CONTOH SOLUSI DENGAN AGG

### 1) Perumusan AGG1

AGG1 digunakan untuk mendekomposisi CVRP menjadi beberapa daerah yang independen. Sebelum AGG1 dapat digunakan, harus dihitung terlebih dahulu setiap kemiringan garis yang menghubungkan pelanggan dengan depot. Setelah setiap kemiringan didapatkan, kemiringan tersebut diurutkan mulai dari yang terkecil hingga yang terbesar. Setelah setiap kemiringan selesai diurutkan, setiap pelanggan yang bersesuaian dengan kemiringan tersebut dilabeli dengan  $v_1, v_2, \dots, v_n$ , dimana  $n$  adalah jumlah pelanggan dalam CVRP yang digunakan, dengan urutan yang sesuai dengan urutan kemiringan. AGG1 dirumuskan dengan karakteristik sebagai berikut:

- Representasi kromosom yang digunakan adalah representasi kromosom biner. Sebagai contoh, apabila CVRP yang digunakan terdiri dari 20 pelanggan, salah satu kromosom yang dapat digunakan adalah 00010000010010000000. Kromosom tersebut menunjukkan bahwa CVRP didekomposisi menjadi tiga daerah. Banyaknya daerah yang terbentuk adalah sama dengan banyaknya digit 1 yang muncul dalam kromosom. Daerah pertama ditandai dengan subkromosom 100000. Karena subkromosom tersebut mengisi kromosom pada posisi ke-4 sampai ke-9, maka yang menjadi pelanggan-pelanggan pada daerah pertama adalah  $v_4, v_5, \dots, v_9$ . Daerah kedua ditandai dengan subkromosom 100. Karena subkromosom tersebut mengisi kromosom pada posisi ke-10 sampai ke-12, maka yang menjadi pelanggan-pelanggan pada daerah kedua adalah  $v_{10}, v_{11}, v_{12}$ . Daerah ketiga ditandai dengan subkromosom 000 dan 10000000. Karena subkromosom tersebut mengisi kromosom pada posisi ke-1 sampai ke-3 dan posisi ke-13 sampai ke-20, maka yang menjadi pelanggan-pelanggan pada daerah ketiga adalah  $v_1, v_2, v_3, v_{13}, v_{14}, \dots, v_{20}$ .

- Besar populasi yang digunakan adalah 100.
- Misalkan dari suatu kromosom didapatkan daerah-daerah  $D_i$ ,  $i = 1, \dots, n$ , dimana  $D_i$  adalah  $\{v_{i,1}, \dots, v_{i,k_i}\}$ . Didefinisikan beberapa fungsi sebagai berikut:

$$f_1(a, m) = |a - m| \quad (2)$$

$$f_2(Q, Q_i) = \max \left\{ 0, Q - \sum_{j=1}^{k_i} q_{i,j} \right\} \quad (3)$$

$$f_3(X_i, Y_i) = \sum_{j=1}^{k_i} \sqrt{\left( x_{i,j} - \frac{1}{k_i} \sum_{j=1}^{k_i} x_{i,j} \right)^2 + \left( y_{i,j} - \frac{1}{k_i} \sum_{j=1}^{k_i} y_{i,j} \right)^2} \quad (4)$$

$$f(a, m, Q, Q_i, X_i, Y_i) = 10^6 f_1(a, m) + 10^3 \sum_{i=1}^m f_2(Q, Q_i) + \sum_{i=1}^m f_3(X_i, Y_i) \quad (5)$$

dimana  $x_{i,j}$ ,  $y_{i,j}$ , dan  $q_{i,j}$  adalah absis, ordinat, dan permintaan dari  $v_{i,j}$ ,  $a$  adalah banyaknya digit 1 dalam kromosom,  $m$  adalah banyak kendaraan yang digunakan,  $Q$  adalah kapasitas kendaraan,  $Q_i$  adalah matriks baris dengan elemen pada kolom ke- $j$  adalah  $q_{i,j}$ ,  $X_i$  adalah matriks baris dengan elemen pada kolom ke- $j$  adalah  $x_{i,j}$ , dan  $Y_i$  adalah matriks baris dengan elemen pada kolom ke- $j$  adalah  $y_{i,j}$ . Fungsi  $f(a, m, Q, Q_i, X_i, Y_i)$  adalah fungsi *fitness* yang digunakan dalam AGG1. Semakin kecil nilai *fitness* dari suatu kromosom, maka semakin baik kromosom tersebut. Nilai  $10^6$  dan  $10^3$  yang digunakan dalam *fitness* tersebut akan menyebabkan tujuan utama dari AGG1 adalah memperkecil  $f_1(a, m)$ , kemudian memperkecil  $\sum_{i=1}^m f_2(Q, Q_i)$ , dan yang terakhir memperkecil  $\sum_{i=1}^m f_3(X_i, Y_i)$ .

- Seleksi dilakukan dengan memilih dua kromosom secara acak.
- Operator *crossover* yang digunakan adalah *1-point crossover* dengan kemungkinan terjadinya *crossover* adalah 1,0.
- Mutasi dilakukan dengan memilih secara acak suatu digit dalam kromosom dan kemudian merubah nilainya. Jika yang terpilih adalah digit 1, maka dirubah menjadi 0. Jika yang terpilih adalah digit 0, maka dirubah menjadi 1. Kemungkinan terjadinya mutasi adalah 0,5.
- Skema penggantian populasi yang digunakan adalah *elitism replacement with filtration*.
- Kondisi pemberhentian yang digunakan adalah tidak bertambahnya *fitness* selama 2000 generasi atau banyaknya generasi telah mencapai 100000.

## 2) Perumusan AGG2

AGG2 digunakan untuk mencari rute terpendek pada setiap daerah yang terbentuk dari hasil dekomposisi oleh AGG1. Rute-rute tersebut kemudian digabungkan untuk menjadi solusi CVRP. AGG2 dirumuskan dengan karakteristik sebagai berikut:

- Representasi kromosom yang digunakan adalah permutasi dari pelanggan-pelanggan. Sebagai contoh, apabila pelanggan-pelanggan pada suatu daerah adalah  $v_4, v_5, \dots, v_9$ , salah satu kromosom yang dapat digunakan adalah  $v_6 v_7 v_4 v_9 v_5 v_8$ . Hal itu menunjukkan bahwa rute perjalanan yang terbentuk adalah  $v_0 \rightarrow v_6 \rightarrow v_7 \rightarrow v_4 \rightarrow v_9 \rightarrow v_5 \rightarrow v_8 \rightarrow v_0$ .
- Besar populasi yang digunakan adalah 50.
- Misalkan sebuah kromosom  $v_1 v_2 \dots v_n$ , fungsi *fitness* yang dapat digunakan untuk mengolah kromosom tersebut adalah

$$f(X, Y) = \sum_{i=1}^{n+1} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (6)$$

dimana  $v_{n+1} = v_0$ ,  $x_i$  dan  $y_i$  adalah absis dan ordinat dari  $v_i$ ,  $X$  adalah matriks baris dengan elemen pada kolom ke- $j$  adalah  $x_j$ , dan  $Y$  adalah matriks baris dengan elemen pada kolom ke- $j$  adalah  $y_j$ .

- Operator seleksi yang digunakan adalah *tournament selection* dengan besar 5.
- Operator *crossover* yang digunakan adalah *sequential constructive crossover* (SCX) dengan kemungkinan terjadinya *crossover* adalah 1,0. SCX tersebut adalah operator *crossover* yang sangat baik untuk menyelesaikan TSP.
- Operator mutasi yang digunakan adalah *exchange* dengan kemungkinan terjadinya mutasi adalah 0,2.
- Skema penggantian populasi yang digunakan adalah *elitism replacement with filtration*.

- Kondisi pemberhentian yang digunakan adalah tidak bertambahnya *fitness* selama  $6 \times \frac{m}{p}$  generasi (*m* adalah banyak pelanggan pada suatu daerah dan *p* adalah banyaknya daerah) atau banyaknya generasi telah mencapai 1000.

#### D. Perbandingan AG dengan AGG

Perbandingan AG dengan AGG dilakukan dengan memanfaatkan permasalahan P50, P75, P100, P125 dan program yang telah dibuat. Perbandingan antara AG dan AGG akan dilihat dari tiga segi yang berbeda yaitu dari segi jarak, waktu komputasi, dan total generasi yang dibutuhkan untuk menyelesaikan CVRP. Dalam segi jarak, digunakan  $P = \frac{\text{jarak1} - \text{jarak2}}{\text{jarak1}} \times 100$  dimana *jarak1* adalah jarak yang diperoleh AG dan *jarak2* adalah jarak yang diperoleh AGG. Nilai *P* tersebut mempunyai arti berapa persen AGG lebih baik atau lebih buruk dari AG. Generasi dari AGG adalah total generasi yang digunakan oleh AGG1 dan semua AGG2.

##### 1) Perbandingan Algoritma pada Permasalahan P50

Untuk perbandingan AG dengan AGG pada permasalahan P50, digunakan empat belas variasi kapasitas kendaraan yang berbeda yaitu 77, 83, 90, 109, 122, 139, 160, 189, 231, dan 297. Untuk setiap kapasitas kendaraan yang berbeda, dilakukan tiga kali pengujian dan dipilih satu yang terbaik. Hasil yang didapat dari AG dan AGG disajikan dalam TABEL 1.

TABEL 1. PERBANDINGAN UNTUK PERMASALAHAN P50

Kapasitas Kendaraan	Algoritma Genetika Ganda			Algoritma Genetika			<i>P</i>
	Jarak	Waktu	Generasi	Jarak	Waktu	Generasi	
77	1591.732	<b>10</b>	7956	<b>1459.428</b>	25	<b>4420</b>	-9.06
83	1533.234	<b>10</b>	7938	<b>1411.733</b>	31	<b>5504</b>	-8.60
90	1429.179	<b>10</b>	7522	<b>1319.67</b>	38	<b>6633</b>	-8.29
99	1287.732	<b>7</b>	6299	<b>1234.57</b>	24	<b>4314</b>	-4.30
109	1247.899	<b>5</b>	4546	<b>1205.129</b>	13	<b>3034</b>	-3.54
122	1166.375	<b>6</b>	<b>4612</b>	<b>1083.415</b>	29	5834	-7.65
139	1059.293	<b>5</b>	<b>2366</b>	<b>1015.836</b>	64	11541	-4.27
160	<b>915.0866</b>	<b>4</b>	<b>3715</b>	985.2263	28	4662	7.11
189	<b>863.5713</b>	<b>3</b>	<b>2345</b>	899.1628	18	4338	3.95
231	801.5071	<b>3</b>	<b>2428</b>	<b>801.136</b>	22	5038	-0.04
297	<b>710.1832</b>	<b>3</b>	<b>2514</b>	788.5825	44	8489	9.94

Dari segi jarak, dapat dilihat bahwa AGG masih kalah dari AG untuk kapasitas kendaraan 77, 83, 90, 99, 109, 122, 139, dan 231. Namun untuk kapasitas kendaraan 160, 189, dan 297, AGG memberikan jarak yang lebih baik. Dapat dilihat dari TABEL 1 bahwa seiring dengan bertambahnya kapasitas kendaraan, nilai *P* juga ikut bertambah kecuali pada beberapa kasus tertentu. Hal ini berarti bahwa seiring dengan bertambahnya kapasitas kendaraan, AGG menjadi semakin baik dari AG. Dari segi waktu komputasi, rata-rata waktu komputasi AGG adalah 6,7 detik sedangkan untuk AG adalah 30,0 detik. Dari segi generasi, rata-rata total generasi AGG adalah 5426 sedangkan untuk AG adalah 5764. Jadi, waktu komputasi dan generasi AGG lebih baik dari AG.

##### 2) Perbandingan Algoritma pada Permasalahan P75

Untuk perbandingan AG dengan AGG pada permasalahan P75, digunakan empat belas variasi kapasitas kendaraan yang berbeda yaitu 105, 113, 123, 135, 149, 166, 188, 217, 257, 314, dan 404. Untuk setiap kapasitas kendaraan yang berbeda, dilakukan tiga kali pengujian dan dipilih satu yang terbaik. Hasil yang didapat dari AG dan AGG disajikan dalam TABEL 2.

Dari segi jarak yang diperoleh, dapat dilihat bahwa AGG masih kalah dari AG untuk kapasitas kendaraan 105, 113, 123, 135, dan 166. Namun untuk kapasitas kendaraan 149, 188, 217, 257, 314, dan 404, AGG memberikan jarak yang lebih baik. Dapat dilihat dari TABEL 2 bahwa seiring dengan bertambahnya kapasitas kendaraan, nilai *P* juga ikut bertambah kecuali pada beberapa kasus tertentu. Hal ini berarti bahwa seiring dengan bertambahnya kapasitas kendaraan, AGG menjadi semakin baik dari AG. Dari segi waktu komputasi, rata-rata waktu komputasi AGG adalah 7,3 detik sedangkan untuk AG adalah 91,1 detik. Dari segi generasi, rata-rata total generasi AGG adalah 4452 sedangkan untuk AG adalah 11749. Jadi, waktu komputasi dan generasi AGG lebih baik dari AG.

TABEL 2. PERBANDINGAN UNTUK PERMASALAHAN P75

Kapasitas Kendaraan	Algoritma Genetika Ganda			Algoritma Genetika			<i>P</i>
	Jarak	Waktu	Generasi	Jarak	Waktu	Generasi	
105	2404.516	<b>9</b>	<b>6183</b>	<b>2386.744</b>	85	10739	-0.74
113	2401.982	<b>6</b>	<b>4767</b>	<b>2309.077</b>	104	13173	-4.02
123	2203.066	<b>9</b>	<b>5955</b>	<b>2170.193</b>	65	8807	-1.51
135	2068.197	<b>7</b>	<b>4757</b>	<b>2053.659</b>	115	14277	-0.70
149	<b>1885.894</b>	<b>5</b>	<b>3336</b>	1961.569	55	7242	3.85
166	1835.419	<b>6</b>	<b>3982</b>	<b>1811.217</b>	136	18191	-1.33
188	<b>1630.429</b>	<b>4</b>	<b>2604</b>	1719.833	85	10677	5.19
217	<b>1501.133</b>	<b>5</b>	<b>2829</b>	1698.279	81	9711	11.6
257	<b>1377.45</b>	<b>5</b>	<b>3171</b>	1598.278	98	12585	13.8
314	<b>1359.159</b>	<b>5</b>	<b>2815</b>	1574.126	95	12278	13.6
404	<b>1215.419</b>	<b>6</b>	<b>2884</b>	1514.616	116	15220	19.7

### 3) *Pembandingan Algoritma pada Permasalahan P100*

Untuk perbandingan AG dengan AGG pada permasalahan P100, digunakan empat belas variasi kapasitas kendaraan yang berbeda. Kapasitas kendaraan yang digunakan adalah 151, 164, 178, 195, 215, 240, 273, 314, 372, 454, dan 584. Untuk setiap kapasitas kendaraan yang berbeda, dilakukan tiga kali pengujian dan dipilih satu yang terbaik. Hasil yang didapat dari AG dan AGG disajikan dalam TABEL 3.

TABEL 3. PERBANDINGAN UNTUK PERMASALAHAN P100

Kapasitas Kendaraan	Algoritma Genetika Ganda			Algoritma Genetika			<i>P</i>
	Jarak	Waktu	Generasi	Jarak	Waktu	Generasi	
151	<b>3664.148</b>	<b>10</b>	<b>6176</b>	3951.525	123	10961	7,27
164	<b>3428.708</b>	<b>11</b>	<b>6267</b>	3622.753	153	13819	5,35
178	<b>3119.468</b>	<b>9</b>	<b>6137</b>	3588.194	125	10521	13,0
195	<b>2958.256</b>	<b>5</b>	<b>2883</b>	3454.294	166	13524	14,3
215	<b>2726.513</b>	<b>6</b>	<b>3908</b>	3287.016	293	26427	17,0
240	<b>2604.126</b>	<b>5</b>	<b>2775</b>	3060.496	231	18616	14,9
273	<b>2463.602</b>	<b>6</b>	<b>2853</b>	2717.931	171	13382	9,35
314	<b>2216.962</b>	<b>7</b>	<b>3742</b>	2829.297	256	22065	21,6
372	<b>2094.614</b>	<b>8</b>	<b>3995</b>	2742.865	164	14749	23,6
454	<b>1939.12</b>	<b>13</b>	<b>4589</b>	2638.619	148	10825	26,5
584	<b>1814.5</b>	<b>17</b>	<b>3398</b>	2458.27	232	20722	26,1

Dari segi jarak yang diperoleh, dapat dilihat bahwa AGG memberikan hasil yang lebih baik dari AG untuk semua variasi kapasitas kendaraan yang digunakan. Dapat dilihat dari TABEL 3 bahwa seiring dengan bertambahnya kapasitas kendaraan, nilai *P* juga ikut bertambah kecuali pada beberapa kasus tertentu. Hal ini berarti bahwa seiring dengan bertambahnya kapasitas kendaraan, AGG menjadi semakin baik dari AG. Dari segi waktu komputasi, rata-rata waktu komputasi AGG adalah 10,4 detik sedangkan untuk AG adalah 171,8 detik. Dari segi generasi, rata-rata total generasi AGG adalah 5143 sedangkan untuk AG adalah 14589. Jadi, waktu komputasi dan generasi AGG lebih baik dari AG.

### 4) *Pembandingan Algoritma pada Permasalahan P125*

Untuk perbandingan AG dengan AGG pada permasalahan P125, digunakan empat belas variasi kapasitas kendaraan yang berbeda. Kapasitas kendaraan yang digunakan adalah 183, 198, 215, 235, 260, 291, 330, 380, 449, 549, dan 706. Untuk setiap kapasitas kendaraan yang berbeda, dilakukan tiga kali pengujian dan dipilih satu yang terbaik. Hasil yang didapat dari AG dan algoritma ganda disajikan dalam TABEL 4.

Dari segi jarak yang diperoleh, dapat dilihat bahwa AGG memberikan hasil yang lebih baik dari AG untuk semua variasi kapasitas kendaraan yang digunakan. Dapat dilihat dari TABEL 4 bahwa seiring dengan bertambahnya kapasitas kendaraan, nilai *P* juga ikut bertambah kecuali pada beberapa kasus

tertentu. Hal ini berarti bahwa seiring dengan bertambahnya kapasitas kendaraan, AGG menjadi semakin baik dari AG. Dari segi waktu komputasi, rata-rata waktu komputasi AGG adalah 15,6 detik sedangkan untuk AG adalah 403,7 detik. Dari segi generasi, rata-rata total generasi AGG adalah 5314 sedangkan untuk AG adalah 22254. Jadi, waktu komputasi dan generasi AGG lebih baik dari AG.

TABEL 4. PERBANDINGAN UNTUK PERMASALAHAN P125

Kapasitas Kendaraan	Algoritma Genetika Ganda			Algoritma Genetika			<i>P</i>
	Jarak	Waktu	Generasi	Jarak	Waktu	Generasi	
183	<b>4476.805</b>	<b>24</b>	<b>11107</b>	5269.657	364	22374	15,0
198	<b>4352.449</b>	<b>10</b>	<b>6188</b>	5100.448	435	24877	14,6
215	<b>4141.08</b>	<b>14</b>	<b>7145</b>	4996.808	298	16720	17,1
235	<b>3806.193</b>	<b>7</b>	<b>3790</b>	4706.922	444	26820	19,1
260	<b>3594.609</b>	<b>7</b>	<b>3166</b>	4597.851	507	30608	21,8
291	<b>3280.79</b>	<b>8</b>	<b>3955</b>	4505.065	372	16397	27,1
330	<b>3165.287</b>	<b>12</b>	<b>5258</b>	4440.228	643	30827	28,7
380	<b>3003.622</b>	<b>12</b>	<b>5045</b>	4275.217	499	28333	29,7
449	<b>2817.172</b>	<b>17</b>	<b>3627</b>	4164.521	407	23457	32,3
549	<b>2603.215</b>	<b>20</b>	<b>3505</b>	3928.582	514	28909	33,7
706	<b>2525.264</b>	<b>47</b>	<b>4063</b>	3502.791	522	30022	27,9

#### IV. SIMPULAN DAN SARAN

##### A. Simpulan

Berdasarkan pada hasil dan pembahasan yang telah dipaparkan, dapat diambil kesimpulan bahwa AGG dapat digunakan untuk menyelesaikan CVRP dengan cara yang berbeda dari AG. Dari segi jarak, AGG lebih baik dari AG kecuali untuk beberapa kapasitas kendaraan yang kecil pada permasalahan P50 dan P75. Untuk kapasitas kendaraan yang semakin besar, nilai *P* juga semakin besar yang berarti bahwa penambahan kapasitas kendaraan mengakibatkan AGG menjadi jauh lebih baik dibandingkan dengan AG. Rata-rata waktu komputasi AGG untuk permasalahan P50, P75, P100, dan P125 tidak melebihi 20 detik sedangkan untuk AG bervariasi mulai dari kisaran 30 detik hingga 400 detik. Sehingga dari segi waktu komputasi, AGG jauh lebih baik dari AG. Rata-rata generasi AGG untuk permasalahan P50, P75, P100, dan P125 berada di kisaran 5000 generasi sedangkan untuk AG bervariasi mulai dari kisaran 5000 generasi hingga 20000 generasi. Sehingga dari segi generasi, AGG lebih baik dari AG.

##### B. Saran

Saran yang dapat diberikan oleh penulis adalah sebagai berikut:

- AGG dapat digunakan untuk menyelesaikan CVRP di dunia nyata.
- AGG dapat dikembangkan oleh peneliti lain agar dapat digunakan untuk menyelesaikan permasalahan yang lain.

#### DAFTAR PUSTAKA

- [1] Karakatic, S. dan Podgorelec, V. 2015. "A Suvey of Genetic Algorithms for Solving Multi Depot Vehicle Routing Problem". *Applied Soft Computing* 27, 519-532.
- [2] Hozairi, Buda, K., Masroeri, dan Irawan, M.I. 2014. "Implementation of Nondominated Sorting Genetic Algorithm – II (NSGA-II) for Multiobjective Optimization Problems on Distribution of Indonesian Navy Warship". *Journal of Theoretical and Applied Information Technology* 64, 274-281.
- [3] Taillard, E. 1993. "Parallel Iterative Search Methods for Vehicle Routing Problem". *Network* 23, 661-673.
- [4] Toth, P. dan Vigo, D. 2002. *The Vehicle Routing Problem*. Philadelphia: University City Science Center.
- [5] Yucenur, G.N. dan Demirel, N.C. 2011. "A New Geometric Shape-Based Genetic Clustering Algorithm for The Multi-Depot Vehicle Routing Problem". *Expert System with Applications* 38, 11859-11865.
- [6] Ahmed, Z.H. 2005. "Genetic Algorithm for Travelling Salesman Problem using Sequential Constructive Crossover Operator". *International Journal of Biometrics & Bioinformatics* 3, 96-105.
- [7] Coley, D.A. 1999. *An Introduction to Genetic Algorithms for Scientists and Engineers*. New Jersey: World Scientific.
- [8] Eiben, A.E. dan Smith, J.E. 2003. *Introduction to Evolutionary Computing*. Berlin: Springer.
- [9] Nazif, H. dan Lee, L.S. 2012. "Optimised Crossover Genetic Algorithm for Capacitated Vehicle Routing Problem". *Applied Mathematical Modelling* 36, 2110-2117.