

# Algoritma Genetika Ganda untuk *Capacitated Vehicle Routing Problem*

Muhammad Luthfi Shahab dan Mohammad Isa Irawan

Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

*E-mail:* mii@its.ac.id

**Abstrak** – *Capacitated vehicle routing problem (CVRP)* adalah salah satu variasi dari *vehicle routing problem (VRP)* yang menggunakan batasan kapasitas pada kendaraan yang dipakai. Ada banyak metode yang telah diteliti untuk bisa menyelesaikan CVRP, namun penggunaan algoritma genetika masih belum memberikan hasil yang memuaskan. Untuk mempermudah menyelesaikan CVRP, dapat dilakukan dekomposisi pada CVRP agar terbagi menjadi beberapa daerah yang dapat diselesaikan secara independen. Berdasarkan hal tersebut, dirumuskan algoritma genetika ganda yang terlebih dahulu berusaha untuk mendekomposisi CVRP dan kemudian mencari rute terpendek pada setiap daerah menggunakan dua algoritma genetika sederhana yang berbeda. Algoritma genetika ganda kemudian dibandingkan dengan algoritma genetika. Untuk membandingkan dua algoritma tersebut, dibuat empat permasalahan yaitu P50, P75, P100, dan P125 dengan pengujian pada setiap permasalahan menggunakan empat belas variasi kapasitas kendaraan yang berbeda. Didapatkan hasil bahwa algoritma genetika ganda lebih baik dari algoritma genetika dari segi waktu komputasi dan generasi. Dari segi jarak, algoritma genetika ganda juga lebih baik dari algoritma genetika kecuali untuk beberapa kapasitas kendaraan yang kecil pada permasalahan P50 dan P75.

**Kata Kunci** – CVRP, algoritma genetika, algoritma genetika ganda

## I. PENDAHULUAN

**V**RP (*Vehicle routing problem*) adalah salah satu permasalahan optimasi kombinatorial yang memiliki banyak aplikasi pada bidang industri [7]. VRP memiliki banyak variasi disesuaikan pada batasan-batasan yang digunakan. Salah satu dari variasi tersebut adalah *capacitated vehicle routing problem (CVRP)* yang menggunakan batasan kapasitas pada kendaraan yang dipakai. Permasalahan penentuan rute distribusi bahan bakar minyak [4], permasalahan pengangkutan sampah oleh truk sampah, dan permasalahan pengambilan barang dari pemasok ke gudang pusat adalah contoh-contoh CVRP di dunia nyata.

Banyaknya aplikasi dari CVRP yang sesuai dengan permasalahan di dunia nyata mengakibatkan CVRP menjadi salah satu bidang ilmu yang banyak diteliti [12]. Penelitian-penelitian untuk menyelesaikan CVRP tersebut dilakukan dengan berbagai metode yang berbeda. Metode eksak dapat menyelesaikan CVRP yang kecil dengan tepat, namun tidak dapat menyelesaikan CVRP yang besar. 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, simulated annealing, tabu search* dan algoritma genetika [6]. Metode *metaheuristic* yang lain seperti *ant colony system* juga dapat digunakan untuk menyelesaikan CVRP [4].

Toth dan Vigo [11] menyatakan bahwa penggunaan algoritma genetika untuk menyelesaikan CVRP masih belum memberikan hasil yang memuaskan. Namun, keberhasilan algoritma genetika untuk menyelesaikan permasalahan-permasalahan lain seperti *travelling salesman problem (TSP)* dan *vehicle routing problem with time windows (VRPTW)* menunjukkan bahwa penggunaan algoritma genetika akan memberikan hasil yang semakin baik jika terus diteliti. Algoritma genetika juga telah berhasil digunakan untuk menyelesaikan permasalahan penempatan *base transceiver station* [8] dan permasalahan transportasi nonlinier [9]. Dalam penelitian lain, pengembangan dari algoritma genetika yaitu NSGA-II juga telah berhasil digunakan untuk menyelesaikan permasalahan optimasi dalam distribusi kapal perang di wilayah perairan Indonesia [5].

Untuk mempermudah menyelesaikan CVRP, Taillard [10] melakukan dekomposisi pada CVRP agar terbagi menjadi beberapa daerah 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 dan banyak dari hasil tersebut masih tetap menjadi yang terbaik hingga saat ini [7].

Berdasarkan hal tersebut, akan dirumuskan algoritma genetika ganda 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. Algoritma genetika ganda tersebut diharapkan dapat digunakan untuk menyelesaikan CVRP dengan hasil yang baik. Algoritma genetika ganda akan dirumuskan secara runtut dan akan dibandingkan dengan algoritma genetika untuk mengetahui seberapa baik algoritma genetika ganda dapat digunakan untuk menyelesaikan CVRP.

## II. METODOLOGI PENELITIAN

Penelitian diawali dengan membuat CVRP agar algoritma genetika dan algoritma genetika ganda dapat diimplementasikan untuk menyelesaikan CVRP tersebut. Banyaknya CVRP yang dibuat adalah empat permasalahan.

Setelah CVRP selesai dibuat, akan dirumuskan algoritma genetika dan algoritma genetika ganda yang dapat digunakan untuk menyelesaikan CVRP. Untuk masing-masing 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 algoritma genetika ganda dan algoritma genetik. Perbandingan antara algoritma genetika ganda dan algoritma genetik 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

Pembuatan CVRP dilakukan agar algoritma genetika dan algoritma genetika ganda dapat diimplementasikan untuk menyelesaikan CVRP tersebut. Agar dapat melakukan penarikan kesimpulan dengan cukup baik, dibuat empat CVRP yaitu P50, P75, P100, dan P125. Absis, ordinat, dan permintaan dari setiap titik tujuan pada CVRP yang dibuat, dipilih secara acak dari suatu rentang tertentu.

Permasalahan P50 terdiri dari depot dan 50 titik yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 50, sedangkan absis dan ordinat titik tujuan bernilai antara 0 sampai 100. Setiap titik tujuan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua titik tujuan adalah 1040.

Permasalahan P75 terdiri dari depot dan 75 titik yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 75, sedangkan absis dan ordinat titik tujuan bernilai antara 0 sampai 150. Setiap titik tujuan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua titik tujuan adalah 1413.

Permasalahan P100 terdiri dari depot dan 100 titik yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 100, sedangkan absis dan ordinat titik tujuan bernilai antara 0 sampai 200. Setiap titik tujuan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua titik tujuan adalah 2044.

Permasalahan P125 terdiri dari depot dan 125 titik yang harus dituju. Dalam permasalahan tersebut, absis dan ordinat depot adalah 125, sedangkan absis dan ordinat titik tujuan bernilai antara 0 sampai 250. Setiap titik tujuan memiliki permintaan yang bernilai antara 10 sampai 30. Total permintaan dari semua titik tujuan adalah 2472.

#### B. Perumusan Algoritma Genetika untuk CVRP

Algoritma genetika dirumuskan sebagai berikut:

- Representasi kromosom yang digunakan dalam algoritma genetika adalah permutasi dari titik-titik tujuan. 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 titik tujuan, 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 titik tujuan. Misalkan kapasitas kendaraan dalam permasalahan adalah 17 dan permintaan dari setiap titik tujuan  $v_i$  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.
- Fungsi *fitness* yang digunakan adalah total jarak yang dibutuhkan untuk melalui setiap titik tujuan dengan urutan yang ada pada kromosom yang bersesuaian.
- Seleksi dilakukan dengan memilih dua kromosom secara acak. Hal ini dilakukan karena pemilihan dua kromosom yang baik tidak menjamin akan didapatkannya keturunan yang baik pula.
- 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.

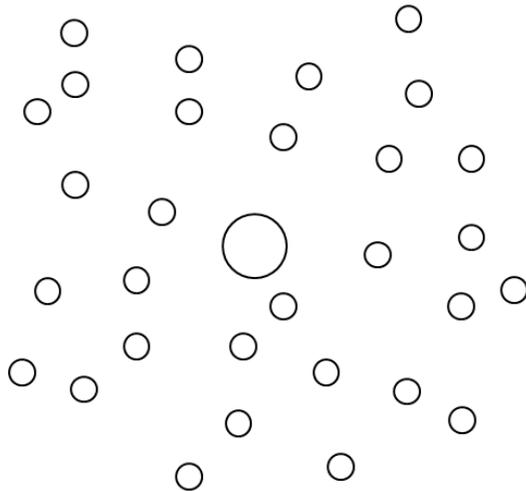
#### C. Perumusan Algoritma Genetika Ganda untuk CVRP

Algoritma genetika ganda bekerja dengan menggabungkan dua algoritma genetika sederhana agar dapat digunakan untuk menyelesaikan CVRP dengan cara yang berbeda dari algoritma genetika yang biasa. Algoritma genetika berusaha untuk menyelesaikan CVRP secara langsung, sedangkan algoritma genetika ganda akan terlebih dahulu berusaha untuk mendekomposisi CVRP menjadi beberapa daerah yang independen dengan AG1 (algoritma genetika pertama dalam algoritma genetika ganda) dan kemudian mencari rute terpendek pada setiap daerah yang terbentuk oleh AG1 dengan AG2 (algoritma genetika kedua dalam algoritma genetika ganda). Daerah-daerah yang terbentuk dari hasil dekomposisi yang dilakukan oleh AG1 haruslah memenuhi dua karakteristik sebagai berikut:

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

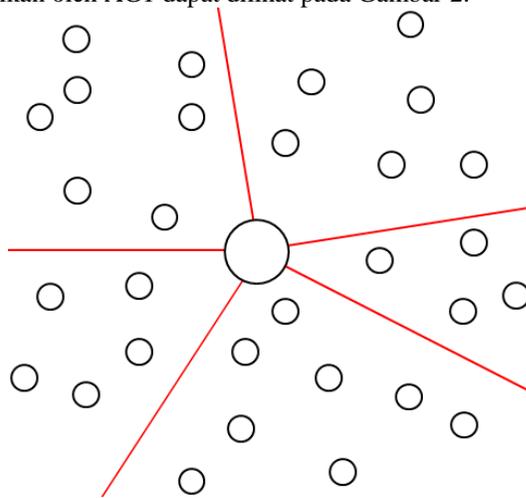
Karakteristik yang pertama diambil dari sifat dasar CVRP yang menyatakan bahwa setiap rute yang terbentuk dalam solusi CVRP harus dilayani oleh satu kendaraan. AG1 akan berusaha memenuhi karakteristik pertama tersebut dengan mempertimbangkan kapasitas kendaraan dan permintaan dari setiap titik tujuan. Karakteristik yang kedua dibuat agar nantinya solusi CVRP yang terbentuk akan menjadi cukup baik. AG1 akan berusaha memenuhi karakteristik kedua tersebut dengan mempertimbangkan kemiringan garis yang menghubungkan titik tujuan dengan depot. Dalam hal ini,

digunakannya kemiringan garis didasarkan pada kenyataan bahwa apabila kemiringan antara dua garis saling berdekatan, maka titik-titik yang ada pada garis tersebut juga akan cukup berdekatan.



Gambar 1. Contoh CVRP

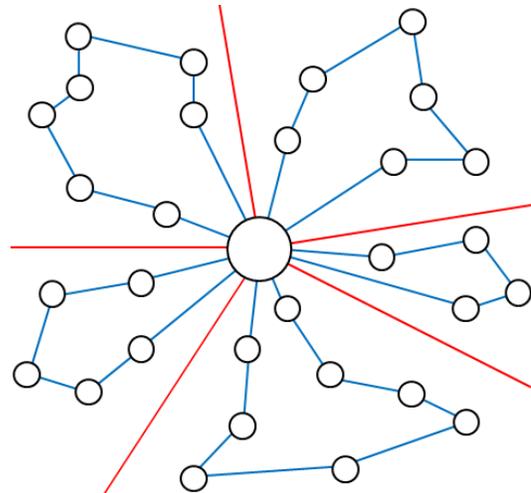
Perhatikan contoh CVRP sederhana dalam Gambar 1 dimana lingkaran besar merepresentasikan depot dan lingkaran-lingkaran kecil merepresentasikan titik-titik tujuan. Dari permasalahan tersebut, salah satu dekomposisi yang bisa dihasilkan oleh AG1 dapat dilihat pada Gambar 2.



Gambar 2. Contoh Dekomposisi oleh AG1

Dengan dilakukannya dekomposisi oleh AG1, 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 titik yang ada dalam daerah, dan kemudian kembali lagi ke depot. Perhatikan bahwa karena setiap daerah yang terbentuk dari hasil dekomposisi hanya membutuhkan satu kendaraan untuk melayani setiap titik tujuan yang ada dalam daerah tersebut, maka informasi mengenai permintaan dari setiap titik tujuan dapat dihilangkan sehingga permasalahan pemilihan rute terpendek pada setiap daerah dapat disebut sebagai *travelling salesman problem* (TSP). Untuk contoh CVRP pada Gambar 1 yang telah didekomposisi seperti pada Gambar 2, penggabungan rute

yang telah diperoleh dengan AG2 dapat dilihat pada Gambar 3.



Gambar 3. Contoh Solusi CVRP dengan Algoritma Genetika Ganda

Sebelum AG1 dapat digunakan, harus dihitung terlebih dahulu setiap kemiringan garis yang menghubungkan titik tujuan dengan depot. Kemiringan tersebut diurutkan mulai dari yang terkecil hingga yang terbesar. Setelah kemiringan selesai diurutkan, setiap titik tujuan yang bersesuaian dengan kemiringan tersebut dilabeli dengan  $v_1, v_2, \dots, v_n$ , dimana  $n$  adalah jumlah titik tujuan dalam CVRP yang digunakan. AG1 dirumuskan dengan karakteristik sebagai berikut:

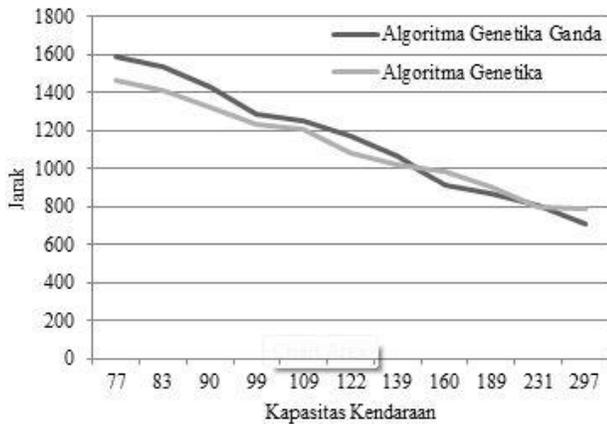
- a) Representasi kromosom yang digunakan dalam AG1 adalah representasi kromosom biner. Sebagai contoh, apabila CVRP yang digunakan terdiri dari 20 titik tujuan, salah satu kromosom yang dapat digunakan adalah 00010000010010000000. Kromosom tersebut menunjukkan bahwa CVRP didekomposisi menjadi tiga daerah. Banyaknya daerah yang terbentuk adalah sama dengan banyaknya digit 1. Daerah pertama ditandai dengan subkromosom 100000. Karena subkromosom tersebut mengisi kromosom pada posisi ke-4 sampai ke-9, maka yang menjadi titik-titik tujuan 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 titik-titik tujuan 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 titik-titik tujuan pada daerah ketiga adalah  $v_1, v_2, v_3, v_{13}, v_{14}, \dots, v_{20}$ .
- b) Besar populasi yang digunakan adalah 100.
- c) Fungsi fitness yang digunakan adalah

$$10^6|A - \lfloor n/D \rfloor| + 10^3 \sum_{i=1}^m \max\{0, D_i - D\} + \sum_{i=1}^m B_i \quad (1)$$

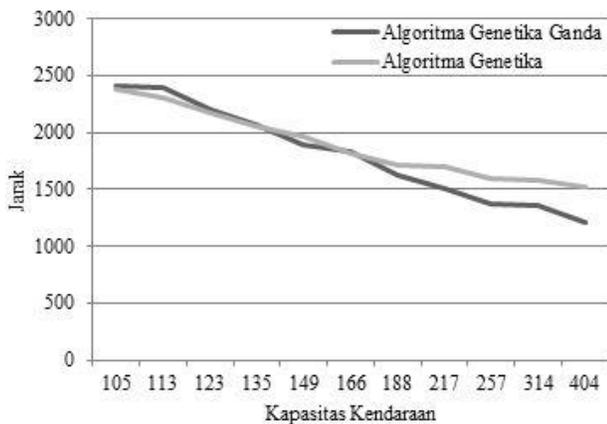
dimana  $A$  adalah banyaknya digit 1 dalam kromosom,  $n$  adalah total permintaan semua titik tujuan,  $D$  adalah kapasitas kendaraan,  $D_i$  adalah total permintaan titik tujuan dalam daerah ke- $i$ , dan  $B_i$  adalah total jarak setiap titik tujuan dengan pusat daerah dalam daerah ke- $i$ .

- d) Seleksi dilakukan dengan memilih dua kromosom secara acak. Hal ini dilakukan karena pemilihan dua kromosom

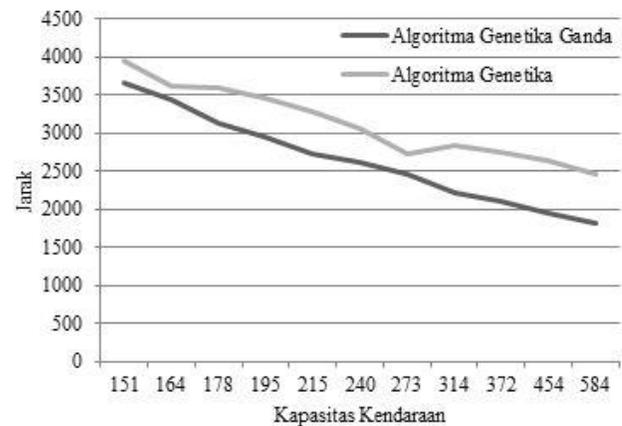
yang baik tidak menjamin akan didapatkannya keturunan yang baik pula.



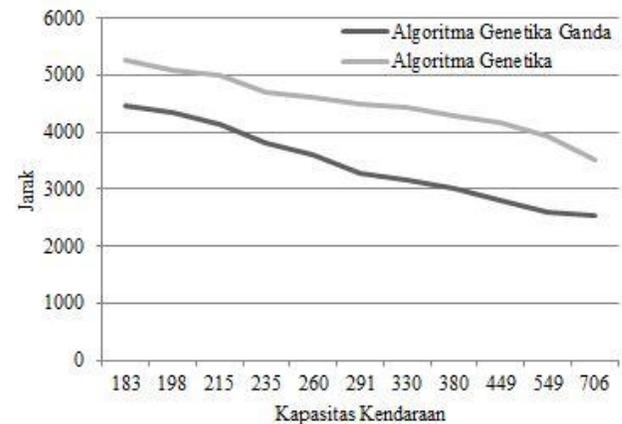
Gambar 4. Perbandingan Hasil untuk Permasalahan P50



Gambar 5. Perbandingan Hasil untuk Permasalahan P75



Gambar 6. Perbandingan Hasil untuk Permasalahan P100



Gambar 7. Perbandingan Hasil untuk Permasalahan P125

- e) Operator *crossover* yang digunakan adalah *1-point crossover* dengan kemungkinan terjadinya *crossover* adalah 1,0.
- f) 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.
- g) Skema penggantian populasi yang digunakan adalah *elitism replacement with filtration*.
- h) Kondisi pemberhentian yang digunakan adalah tidak bertambahnya *fitness* selama 2000 generasi atau banyaknya generasi telah mencapai 100000.

AG2 digunakan untuk mencari rute terpendek pada setiap daerah yang terbentuk oleh AG1. AG2 memiliki karakteristik-karakteristik sebagai berikut:

- a) Representasi kromosom yang digunakan adalah permutasi dari titik-titik tujuan. Sebagai contoh, apabila titik-titik tujuan 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  $v_0 \rightarrow v_6 \rightarrow v_7 \rightarrow v_4 \rightarrow v_9 \rightarrow v_5 \rightarrow v_8 \rightarrow v_0$ .
- b) Besar populasi yang digunakan adalah 50.
- c) Fungsi fitness yang digunakan adalah

$$\sum_{i=1}^m \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} + \sqrt{(x_m - x_0)^2 + (y_m - y_0)^2}$$

dimana  $x_i$  dan  $y_i$  adalah absis dan ordinat dari titik tujuan ke- $i$  dan  $x_0$  dan  $y_0$  adalah absis dan ordinat dari depot.

- d) Operator seleksi yang digunakan adalah *tournament selection* dengan besar 5.
- e) Operator *crossover* yang digunakan adalah *sequential constructive crossover* (SCX) dengan kemungkinan terjadinya *crossover* adalah 1,0.
- f) Operator mutasi yang digunakan adalah *exchange* dengan kemungkinan terjadinya mutasi adalah 0,2.
- g) Skema penggantian populasi yang digunakan adalah *elitism replacement with filtration*.
- h) Kondisi pemberhentian yang digunakan adalah tidak bertambahnya *fitness* selama  $6 \times \frac{m}{p}$  generasi ( $m$  adalah banyak titik tujuan pada suatu daerah dan  $p$  adalah banyaknya daerah) atau banyaknya generasi telah mencapai 1000.

#### D. Perbandingan Algoritma Genetika dengan Algoritma Genetika Ganda

Perbandingan algoritma genetika dengan algoritma genetika ganda dilakukan dengan memanfaatkan permasalahan P50, P75, P100, P125 dan program yang telah dibuat. Perbandingan antara algoritma genetika dan algoritma genetika ganda akan dilihat dari tiga segi yang berbeda yaitu dari segi jarak, waktu komputasi, dan total generasi yang dibutuhkan untuk menyelesaikan CVRP.

Untuk perbandingan algoritma pada permasalahan P50, digunakan empat belas variasi kapasitas kendaraan yang berbeda. Kapasitas kendaraan yang digunakan adalah 77, 83,

Tabel 1. Perbandingan untuk Permasalahan P50

KK	Algoritma Genetika Ganda			Algoritma Genetika Biasa		
	Hasil	Waktu	Gen	Hasil	Waktu	Gen
77	1591.732	<b>10</b>	7956	<b>1459.428</b>	25	<b>4420</b>
83	1533.234	<b>10</b>	7938	<b>1411.733</b>	31	<b>5504</b>
90	1429.179	<b>10</b>	7522	<b>1319.67</b>	38	<b>6633</b>
99	1287.732	<b>7</b>	6299	<b>1234.57</b>	24	<b>4314</b>
109	1247.899	<b>5</b>	4546	<b>1205.129</b>	13	<b>3034</b>
122	1166.375	<b>6</b>	<b>4612</b>	<b>1083.415</b>	29	5834
139	1059.293	<b>5</b>	<b>2366</b>	<b>1015.836</b>	64	11541
160	<b>915.0866</b>	<b>4</b>	<b>3715</b>	985.2263	28	4662
189	<b>863.5713</b>	<b>3</b>	<b>2345</b>	899.1628	18	4338
231	801.5071	<b>3</b>	<b>2428</b>	<b>801.136</b>	22	5038
297	<b>710.1832</b>	<b>3</b>	<b>2514</b>	788.5825	44	8489

KK – Kapasitas kendaraan, Gen – Generasi

Tabel 2. Perbandingan untuk Permasalahan P75

KK	Algoritma Genetika Ganda			Algoritma Genetika Biasa		
	Hasil	Waktu	Gen	Hasil	Waktu	Gen
105	2404.516	<b>9</b>	<b>6183</b>	<b>2386.744</b>	85	10739
113	2401.982	<b>6</b>	<b>4767</b>	<b>2309.077</b>	104	13173
123	2203.066	<b>9</b>	<b>5955</b>	<b>2170.193</b>	65	8807
135	2068.197	<b>7</b>	<b>4757</b>	<b>2053.659</b>	115	14277
149	<b>1885.894</b>	<b>5</b>	<b>3336</b>	1961.569	55	7242
166	1835.419	<b>6</b>	<b>3982</b>	<b>1811.217</b>	136	18191
188	<b>1630.429</b>	<b>4</b>	<b>2604</b>	1719.833	85	10677
217	<b>1501.133</b>	<b>5</b>	<b>2829</b>	1698.279	81	9711
257	<b>1377.45</b>	<b>5</b>	<b>3171</b>	1598.278	98	12585
314	<b>1359.159</b>	<b>5</b>	<b>2815</b>	1574.126	95	12278
404	<b>1215.419</b>	<b>6</b>	<b>2884</b>	1514.616	116	15220

KK – Kapasitas kendaraan, Gen – Generasi

90, 109, 122, 139, 160, 189, 231, dan 297. Kapasitas-kapasitas tersebut dipilih agar jumlah kendaraan minimal yang dapat digunakan untuk menuju semua titik tujuan bervariasi mulai dari 4 hingga 14 kendaraan. Sebagai contoh, apabila kapasitas kendaraan adalah 77 maka jumlah kendaraan minimal yang dapat digunakan adalah  $\lceil \frac{1040}{77} \rceil = \lceil 13,507 \rceil = 14$ , dan apabila kapasitas kendaraan adalah 297 maka jumlah kendaraan minimal yang dapat digunakan adalah  $\lceil \frac{1040}{297} \rceil = \lceil 3,502 \rceil = 4$ . Untuk setiap kapasitas kendaraan yang berbeda, dilakukan tiga kali pengujian dan dipilih satu yang terbaik. Hasil yang didapat dari algoritma genetika ganda dan algoritma biasa disajikan dalam Gambar 4 dan Tabel 1. Algoritma genetika ganda masih kalah dari algoritma genetika biasa untuk kapasitas kendaraan 77, 83, 90, 99, 109, 122, 139, dan 231. Namun untuk kapasitas kendaraan 160, 189, dan 297, algoritma genetika ganda memberikan hasil yang lebih baik. Rata-rata waktu komputasi algoritma genetika ganda adalah 6,7 detik sedangkan untuk algoritma genetika biasa adalah 30,0 detik. Rata-rata total generasi algoritma genetika ganda adalah 5426 sedangkan untuk algoritma genetika biasa adalah 5764.

Untuk perbandingan algoritma pada permasalahan P75, digunakan empat belas variasi kapasitas kendaraan yang

berbeda. Kapasitas kendaraan yang digunakan adalah 105, 113, 123, 135, 149, 166, 188, 217, 257, 314, dan 404. Untuk setiap kapasitas kendaraan yang berbeda, dilakukan tiga kali

Tabel 3. Perbandingan untuk Permasalahan P100

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

KK – Kapasitas kendaraan, Gen – Generasi

Tabel 4. Perbandingan untuk Permasalahan P125

KK	Algoritma Genetika Ganda			Algoritma Genetika Biasa		
	Hasil	Waktu	Gen	Hasil	Waktu	Gen
183	<b>4476.805</b>	<b>24</b>	<b>11107</b>	5269.657	364	22374
198	<b>4352.449</b>	<b>10</b>	<b>6188</b>	5100.448	435	24877
215	<b>4141.08</b>	<b>14</b>	<b>7145</b>	4996.808	298	16720
235	<b>3806.193</b>	<b>7</b>	<b>3790</b>	4706.922	444	26820
260	<b>3594.609</b>	<b>7</b>	<b>3166</b>	4597.851	507	30608
291	<b>3280.79</b>	<b>8</b>	<b>3955</b>	4505.065	372	16397
330	<b>3165.287</b>	<b>12</b>	<b>5258</b>	4440.228	643	30827
380	<b>3003.622</b>	<b>12</b>	<b>5045</b>	4275.217	499	28333
449	<b>2817.172</b>	<b>17</b>	<b>3627</b>	4164.521	407	23457
549	<b>2603.215</b>	<b>20</b>	<b>3505</b>	3928.582	514	28909
706	<b>2525.264</b>	<b>47</b>	<b>4063</b>	3502.791	522	30022

KK – Kapasitas kendaraan, Gen – Generasi

pengujian dan dipilih satu yang terbaik. Hasil yang didapat dari algoritma genetika ganda dan algoritma biasa disajikan dalam Gambar 5 dan Tabel 2. Algoritma genetika ganda masih kalah dari algoritma genetika biasa untuk kapasitas kendaraan 105, 113, 123, 135, dan 166. Namun untuk kapasitas kendaraan 149, 188, 217, 257, 314, dan 404 algoritma genetika ganda memberikan hasil yang lebih baik. Rata-rata waktu komputasi algoritma genetika ganda adalah 7,3 detik sedangkan untuk algoritma genetika biasa adalah 91,1 detik. Rata-rata total generasi algoritma genetika adalah 4452 sedangkan untuk algoritma genetika biasa adalah 11749.

Untuk perbandingan algoritma 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 algoritma genetika ganda dan algoritma biasa disajikan dalam Gambar 6 dan Tabel 3. Algoritma genetika ganda memberikan hasil yang lebih baik dari algoritma genetika biasa untuk semua variasi kapasitas kendaraan yang digunakan. Rata-rata waktu komputasi algoritma genetika ganda adalah 10,4 detik sedangkan untuk algoritma genetika biasa adalah 171,8 detik. Rata-rata total generasi algoritma

genetika adalah 5143 sedangkan untuk algoritma genetika biasa adalah 14589.

Untuk perbandingan algoritma 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 algoritma genetika ganda dan algoritma biasa disajikan dalam Gambar 7 dan Tabel 4. Algoritma genetika ganda memberikan hasil yang lebih baik dari algoritma genetika biasa untuk semua variasi kapasitas kendaraan yang digunakan. Rata-rata waktu komputasi algoritma genetika ganda adalah 15,6 detik sedangkan untuk algoritma genetika biasa adalah 403,7 detik. Rata-rata total generasi algoritma genetika adalah 5314 sedangkan untuk algoritma genetika biasa adalah 22254.

#### IV. KESIMPULAN DAN SARAN

##### A. Kesimpulan

Kesimpulan yang bisa diambil dari hasil dan pembahasan adalah sebagai berikut:

1. Algoritma genetika ganda dapat digunakan untuk menyelesaikan CVRP dengan cara yang berbeda dari algoritma genetika.
2. Dari segi jarak, algoritma genetika ganda lebih baik dari algoritma genetika 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 algoritma genetika ganda menjadi jauh lebih baik dibandingkan dengan algoritma genetika.
3. Rata-rata waktu komputasi algoritma genetika ganda untuk permasalahan P50, P75, P100, dan P125 tidak melebihi 20 detik sedangkan untuk algoritma genetika bervariasi mulai dari kisaran 30 detik hingga 400 detik. Sehingga dari segi waktu komputasi, algoritma genetika ganda jauh lebih baik dari algoritma genetika.
4. Rata-rata generasi algoritma genetika ganda untuk permasalahan P50, P75, P100, dan P125 berada di kisaran 5000 generasi sedangkan untuk algoritma genetika bervariasi mulai dari kisaran 5000 generasi hingga 20000 generasi. Sehingga dari segi generasi, algoritma genetika ganda lebih baik dari algoritma genetika.

##### B. Saran

Saran yang dapat diberikan oleh penulis adalah sebagai berikut:

1. Algoritma genetika ganda dapat digunakan untuk menyelesaikan CVRP di dunia nyata.
2. Algoritma genetika ganda dapat dikembangkan oleh peneliti lain agar dapat digunakan untuk menyelesaikan permasalahan yang lain.

#### DAFTAR PUSTAKA

1. Ahmed, Z.H. 2005. "Genetic Algorithm for Travelling Salesman Problem using Sequential Constructive Crossover Operator". **International Journal of Biometrics & Bioinformatics** 3, 96-105.
2. Coley, D.A. 1999. **An Introduction to Genetic Algorithms for Scientists and Engineers**. New Jersey: World Scientific.
3. Eiben, A.E. dan Smith, J.E. 2003. **Introduction to Evolutionary Computing**. Berlin: Springer.
4. Hermawan, A. 2012. "Penentuan Rute yang Optimal pada Kegiatan Distribusi BBM Menggunakan *Ant Colony System*". **Surabaya: Tugas Akhir, Jurusan Matematika, Institut Teknologi Sepuluh Nopember**.
5. 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.
6. Karakatic, S. dan Podgorelec, V. 2015. "A Survey of Genetic Algorithms for Solving *Multi Depot Vehicle Routing Problem*". **Applied Soft Computing** 27, 519-532.
7. Nazif, H. dan Lee, L.S. 2012. "Optimised Crossover Genetic Algorithm for *Capacitated Vehicle Routing Problem*". **Applied Mathematical Modelling** 36, 2110-2117.
8. Pramsistya, Y. 2010. "Optimasi Penempatan BTS dengan Menggunakan Algoritma Genetika". **Surabaya: Tugas Akhir, Jurusan Matematika, Institut Teknologi Sepuluh Nopember**.
9. Soelistyowati, R. 2010. "Pendekatan Algoritma Genetika untuk Menyelesaikan Masalah Transportasi Nonlinier". **Surabaya: Tugas Akhir, Jurusan Matematika, Institut Teknologi Sepuluh Nopember**.
10. Taillard, E. 1993. "Parallel Iterative Search Methods for *Vehicle Routing Problem*". **Network** 23, 661-673.
11. Toth, P. dan Vigo, D. 2002. **The Vehicle Routing Problem**. Philadelphia: University City Science Center.
12. 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.