



## Model Penentuan Rute Terpendek Penjemputan Sampah Menggunakan Metode MTSP dan Algoritma Genetika

Aswandi<sup>1</sup>, Sugiarto Cokrowibowo<sup>2</sup>, Arnita Irianti<sup>3</sup>

<sup>1,2,3</sup>Informatika, Teknik, Universitas Sulawesi Barat

<sup>1</sup>aswandiit@gmail.com\*, <sup>2</sup>sugiarto.cokrowibowo@unsulbar.ac.id, <sup>3</sup>arnitairianti@unsulbar.ac.id

### Abstract

Garbage pick-ups performed by two or more people must have a route in their pickup. However, it is not easy to model the route of the pickup that each point must be passed and each point is only passed once. Now, the method to create a route has been done a lot, one of the most commonly used methods is the creation of routes using the Traveling Salesman Problem method. Traveling Salesman Problem is a method to determine the route of a series of cities where each city is only traversed once. In this study, the shortest route modeling was conducted using Multiple Traveling Salesman Problem and Genetic Algorithm to find out the shortest route model that can be passed in garbage pickup. In this study, datasets will be used as pick-up points to then be programmed to model the shortest routes that can be traveled. The application of Multiple Traveling Salesman Problem method using Genetic Algorithm shows success to model garbage pickup route based on existing dataset, by setting the parameters of 100 generations and 100 population and 4 salesmen obtained 90% of the best individual opportunities obtained with the best individual fitness value of 0.05209. The test was conducted using BlackBox testing and the results of this test that the functionality on the system is 100% appropriate.

Keywords: genetic algorithm, multiple traveling salesman problem, shortest route, traveling salesman problem.

### Abstrak

Penjemputan sampah yang dilakukan oleh dua orang atau lebih harus mempunyai rute dalam penjemputannya. Namun, tidak mudah untuk memodelkan rute dari penjemputan yang setiap titik harus dilalui dan setiap titik hanya dilalui sekali. Sekarang, metode untuk membuat rute sudah banyak dilakukan, salah satu metode yang paling sering digunakan yaitu pembuatan rute menggunakan metode *Traveling Salesman Problem*. *Traveling Salesman Problem* merupakan suatu metode untuk menentukan rute dari serangkaian kota tertentu dimana setiap kota hanya dilalui sekali. Pada penelitian ini, dilakukan pemodelan rute terpendek menggunakan *Multiple Traveling Salesman Problem* dan Algoritma Genetika untuk mengetahui model rute terpendek yang bisa dilalui dalam penjemputan sampah. Pada penelitian ini akan digunakan dataset sebagai titik-titik penjemputan untuk kemudian dimasukkan keprogram untuk dimodelkan rute terpendek yang bisa dilalui. Penerapan metode *Multiple Traveling Salesman Problem* menggunakan Algoritma Genetika ini menunjukkan keberhasilan untuk memodelkan rute penjemputan sampah berdasarkan dataset yang ada, dengan mengatur parameter 100 generasi dan 100 populasi serta 4 salesman didapatkan 90% peluang individu terbaik didapatkan dengan nilai fitness individu terbaik yaitu 0,05209. Pengujian yang dilakukan yaitu menggunakan pengujian BlackBox dan hasil dari pengujian ini bahwa fungsi pada sistem 100% sudah sesuai.

Kata kunci: algoritma genetika, mtsp, rute terpendek, tsp.

### 1. Pendahuluan

Penentuan rute penjemputan sampah dengan dua tau lebih penjemput pada wilayah dengan titik penjemputan sampah yang banyak akan sangat susah untuk dilakukan mengingat dalam melakukan penjemputan sampah ada beberapa aturan yang harus dipenuhi yaitu semua titik harus dilalui dan tidak ada titik yang dilalui lebih dari sekali. Pembuatan model yang dilakukan secara manual akan membutuhkan waktu yang sangat lama dalam pengerjaannya mengingat perlu diperhatikan jarak yang ditempuh setiap salesman serta waktu yang diperlukan dalam penjemputan sesuai dengan rute yang dibuat.

Strategi yang tepat yaitu membuat sistem pembuatan model rute terpendek untuk memodelkan rute dari penjemputan sampah. Model rute tersebut nantinya akan menghasilkan rute dengan jarak terpendek dan setiap titik penjemputan hanya akan dilalui sekali kecuali titik asal dan semua titik akan dilalui sehingga memudahkan dalam pemodelan rute penjemputan[1].

Terdapat metode *multiple traveling salesman problem*(MTSP) menggunakan algoritma genetika untuk pemodelan rute tersebut. MTSP merupakan metode untuk menentukan rute terpendek dari serangkaian kota tertentu dengan salesman lebih dari satu, dimana setiap kota dilalui tepat 1 kali kecuali kota



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

asal [2]. MTSP merupakan model turunan dari *traveling salesman problem* yang merupakan masalah optimisasi kombinatorial klasik yang pengaplikasiannya banyak ditemukan dalam kehidupan nyata, seperti pemetaan, penjadwalan, dan perencanaan [3].

Algoritma genetika merupakan algoritma pencarian dan model ilmiah evolusi alami yang diangkat dari mekanisme seleksi alam oleh Darwin. Ide dasar dari algoritma genetika adalah untuk mensimulasikan evolusi populasi individu yang mewakili masalah pencarian tertentu, mempromosikan kelangsungan hidup dan reproduksi yang paling cocok. Algoritma genetika merupakan algoritma pencarian tujuan umum serta sebagai algoritma optimasi dan model ilmiah evolusi yang sering digunakan diberbagai bidang sains dan teknik [4].

Pembuatan model MTSP menggunakan algoritma genetika memiliki tahapan yaitu pertama melakukan input parameter. Tahap ini merupakan tahap penambahan nilai parameter untuk digunakan dalam pembuatan individu[5]. Tahap selanjutnya yaitu melakukan perhitungan nilai *fitness*. Tahap ini merupakan perhitungan nilai *fitness* individu untuk diketahui individu terbaik dari populasi. Individu dengan nilai *fitness* tertinggi merupakan individu terbaik pada populasi tersebut [1]. Tahap selanjutnya yaitu *Sorting* individu. Tahap *sorting* merupakan tahap untuk menempatkan individu dengan nilai *fitness* tertinggi berada paling atas sehingga memudahkan untuk melihat individu terbaik [4]. Tahap selanjutnya yaitu, *Elitism*. *Elitism* merupakan tahapan untuk menyimpan individu terbaik dalam proses evolusi untuk menjaga atau mencegah individu mengalami kerusakan akibat proses pindah silang dan mutasi [4]. Tahap selanjutnya yaitu proses pindah silang atau *crossover*. Tahap ini merupakan tahap pindah silang dua individu yang terpilih sebagai *parent* untuk menghasilkan dua individu baru [5]. Tahap selanjutnya yaitu *mutation*. Mutasi merupakan proses perubahan pada kromosom individu dengan cara melakukan *swap position* sehingga terbentuk individu baru. Setelah tahap pindah silang dan mutasi selesai maka akan tercipta populasi baru dalam generasi. Individu terbaik akan didapatkan pada tahap terakhir dimana individu terbaik diambil dari proses *elitism*.

Dalam penelitian ini penulis memaparkan beberapa penelitian terdahulu yang digunakan sebagai ajuan dari penelitian ini yaitu memberikan rekomendasi untuk menggunakan metode *multiple traveling salesman problem* menggunakan algoritma genetika diantaranya yaitu : algoritma genetika untuk *traveling salesman problem* modifikasi operator *cycle crossover* [6], penerapan algoritma genetika dengan *random crossover* dan *dynamic mutation* pada masalah *traveling salesman problem* [2], *multiple traveling salesman problem* menggunakan algoritma *ant colony optimization* dengan operasi *elitism* [7], algoritma genetika untuk skala besar

menggunakan *color balanced traveling salesman problem* [8], analisis komparatif beberapa formulasi *traveling salesman problem* asimetris [9].

Dari permasalahan yang ada tujuan yang ingin dicapai pada penelitian ini yaitu membangun sistem yang dapat mengelola pembuatan model rute penjemputan sampah serta memberikan model rute terbaik yang bisa dilalui dengan jarak terpendek dalam penjemputan sampah dengan menerapkan metode *multiple traveling salesman problem* menggunakan algoritma genetika sesuai dengan dataset yang ada. Adapun manfaat yang diharapkan dengan adanya sistem ini yaitu dapat memberikan kemudahan bagi pihak penjemput sampah untuk menentukan rute dalam penjemputan sampah.

## 2. Metode Penelitian

Metode penelitian berisi langkah-langkah yang akan dilakukan dalam penelitian ini yang terdiri dari pengumpulan data dan metode penentuan rute terpendek.

### 2.1 Pengumpulan Data

Pengumpulan data merupakan proses untuk mengumpulkan data yang berguna untuk penelitian yang dilakukan. Metode yang digunakan dalam pengumpulan data pada penelitian ini yaitu studi literatur. Studi literatur dilakukan untuk mendapatkan data dari berbagai sumber tertulis. Pada penelitian ini dilakukan pengumpulan data berupa dataset, rute, proses, serta hasil dari penelitian yang dilakukan sebelumnya. Data yang dipakai pada penelitian ini menggunakan data dat TSPLIB yang juga telah digunakan pada penelitian sebelumnya yang membahas mengenai perbandingan algoritma dalam menyelesaikan masalah *traveling salesman problem* [10].

### 2.2 Menentukan Rute Penjemputan

Metode *multiple traveling salesman problem* menggunakan algoritma genetika digunakan untuk membuat pemodelan rute penjemputan sampah berdasarkan dataset yang digunakan. Tahapan pembuatan model rute yaitu sebagai berikut :

#### 2.2.1 Input parameter

Tahap ini merupakan tahap penambahan nilai parameter yaitu parameter populasi, generasi, individu terseleksi, *crossover point*, probabilitas mutasi, dan maksimal mutasi. Parameter populasi merupakan banyaknya individu yang akan dibuat dalam satu generasi. Sedangkan parameter generasi merupakan banyaknya generasi yang akan dibuat. Sedangkan parameter individu terseleksi merupakan jumlah individu yang akan digunakan dalam proses *crossover* dan *mutation*. Sedangkan parameter *crossover point* merupakan jumlah titik kromosom yang akan dilakukan pindah silang. Parameter probabilitas mutasi merupakan nilai perbandingan untuk digunakan dalam proses mutasi. Dan

parameter maksimal mutasi merupakan indeks maksimal kromosom yang akan dilakukan mutasi.

### 2.2.2 Hitung nilai *fitness*

Tahap ini merupakan tahapan perhitungan nilai *fitness* individu untuk diketahui individu terbaik dari populasi. Individu dengan nilai *fitness* tertinggi merupakan individu terbaik pada populasi tersebut. Rumus perhitungan nilai *fitness* pada penelitian ini yaitu :

$$f = \frac{1}{dMax} \quad (1)$$

### 2.2.3 Crossover

Tahap ini merupakan tahap pindah silang dua individu yang terpilih sebagai *parent* untuk menghasilkan dua individu baru. Proses pindah silang pada penelitian ini menggunakan *partially mapped crossover* (PMX). Proses pindah silang dengan operator PMX dapat dilihat dibawah ini :

$$P_1 = (3 \ 4 \ 8 \ | \ 2 \ 7 \ 1 \ | \ 6 \ 5),$$

$$P_2 = (4 \ 2 \ 5 \ | \ 1 \ 6 \ 8 \ | \ 3 \ 7).$$

Bagian yang akan dilakukan pindah silang berada diantara titik potong yang ditandai dengan “|”. Dalam hal ini, titik yang akan dipindah silang yaitu 2 ↔ 1, 7 ↔ 6, dan 1 ↔ 8. Kemudian titik potong dipindahsilangkan untuk menghasilkan keturunan sebagai berikut :

$$O_1 = (\times \ \times \ \times \ | \ 1 \ 6 \ 8 \ | \ \times \ \times),$$

$$O_2 = (\times \ \times \ \times \ | \ 2 \ 7 \ 1 \ | \ \times \ \times).$$

Kemudian kita bisa mengisi kromosom lebih lanjut dengan gen yang tidak memiliki konflik dari orang tua asli :

$$O_1 = (3 \ 4 \ \times \ | \ 1 \ 6 \ 8 \ | \ \times \ 5),$$

$$O_2 = (4 \ \times \ 5 \ | \ 2 \ 7 \ 1 \ | \ 3 \ \times).$$

Untuk × pertama merupakan kromosom 8 pada *parent* pertama, tetapi kromosom 8 sudah ada pada *offspring* ini, jadi kita memeriksa proses pindah silang 8 ↔ 1, tetapi 1 juga sudah ada pada *offspring* ini kemudian dilakukan pengecekan pindah silang lagi 1 ↔ 2, karena 2 belum ada pada *offspring* ini jadi 2 menempati × pertama. Begitu juga dengan × yang kedua dimana ini merupakan posisi 6 pada kromosom *parent* pertama tetapi 6 sudah ada dalam *offspring* ini jadi dilakukan pengecekan pindah silang 6 ↔ 7, karena 7 belum ada pada kromosom ini maka 7 menempati × yang kedua. Maka *offspring* pertama adalah :

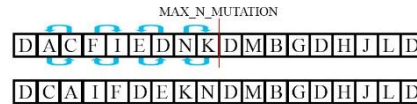
$$O_1 = (3 \ 4 \ 2 \ | \ 1 \ 6 \ 8 \ | \ 7 \ 5). \quad (5)$$

Dengan cara yang sama maka kita akan mendapatkan *offspring* kedua yaitu :

$$O_2 = (4 \ 8 \ 5 \ | \ 2 \ 7 \ 1 \ | \ 3 \ 6). \quad (6)$$

### 2.2.4 Mutation

Mutasi merupakan proses perubahan kromosom individu dengan cara *swap position* sehingga tercipta individu baru. Jumlah kromosom yang akan dilakukan *swap position* dibatasi oleh maksimal mutasi. Contoh proses mutasi dapat dilihat pada gambar 1.



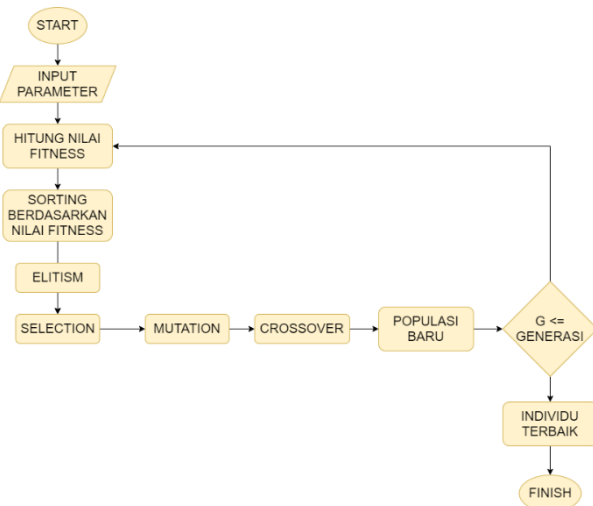
Gambar 1. Proses Mutasi

## 3. Hasil dan Pembahasan

Pada tahap ini menjelaskan tentang mengenai rancangan dari sitem serta menjelaskan tentang implementasi yang ada.

### 3.1 Rancangan Sistem

Penjelasan mengenai rancangan sistem *multiple traveling salesman problem* menggunakan algoritma genetika dapat dilihat pada gambar 2.



Gambar 2. Rancangan Sistem

Dari Gambar 2. Dapat dilihat bahwa sistem akan dimulai dari penginputan parameter yang dibutuhkan dalam algoritma genetika kemudian sistem akan berjalan dan menciptakan populasi awal. Setiap individu yang ada pada populasi kemudian akan dihitung nilai *fitness* masing-masing. Dari nilai *fitness* individu kemudian akan di *sorting* sehingga individu dengan nilai *fitness* tertinggi akan berada paling atas, hal ini bertujuan untuk memudahkan dalam proses penyimpanan individu terbaik atau proses *elitism*. Proses *elitism* merupakan proses dimana individu terbaik disimpan untuk mencegah adanya kerusakan pada individu ketika memasuki proses pindah silang dan mutasi. Proses *crossover* dilakukan untuk mendapatkan individu baru dengan cara memindahsilangkan dua individu yang terpilih sebagai orang tua. Proses mutasi dilakukan dengan proses *swap position* pada kromosom individu sehingga tercipta individu baru. Setelah proses *crossover*

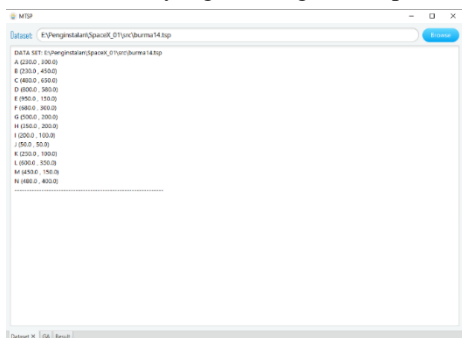
dan mutasi selesai maka akan tercipta populasi baru. Ketika generasi yang tercipta masih lebih kecil dari parameter generasi maka akan dilakukan perhitungan nilai fitness kembali untuk kembali melakukan pembuatan populasi baru. Ketika generasi sudah sama dengan parameter generasi maka akan tercipta individu dengan nilai *fitness* terbaik.

### 3.2 Implementasi

Tahap implementasi merupakan tahap untuk menjelaskan implementasi dari tahap analisa yang dilakukan. Pengimplementasian penelitian ini menggunakan *JavaFx* untuk membuat tampilan sistem.

#### 3.2.1 Antarmuka Form Tambah Data

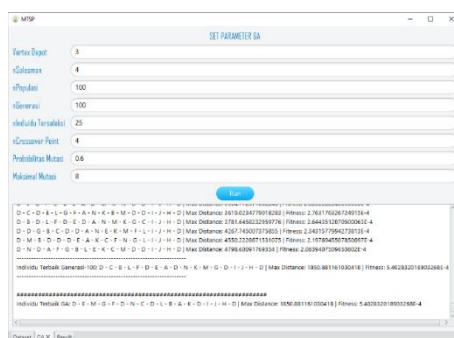
Antarmuka form tambah data digunakan untuk menambahkan dataset yang akan digunakan pada sistem.



Gambar 3. Antarmuka Form Tambah Data

#### 3.2.2 Antarmuka Form Input Parameter

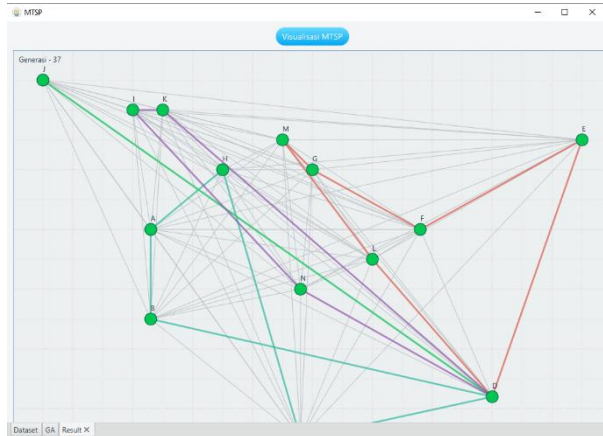
Antarmuka form input parameter digunakan untuk penginputan nilai-nilai parameter yaitu vertex depot, jumlah salesman, jumlah generasi, jumlah populasi, jumlah crossover point, probabilitas mutasi, jumlah individu terseleksi, dan maksimal mutasi.



Gambar 4. Antarmuka Form Input Parameter

#### 3.2.3 Antarmuka Visualisasi Rute

Antarmuka Visualisasi Rute digunakan untuk menampilkan peta kota dengan titik-titik penjemputan yang digambarkan dalam bentuk node. Pada halaman ini pula akan ditampilkan rute terbaik yang bisa dilalui.



Gambar 5. Antarmuka Visualisasi Rute

### 3.3 Algoritma Genetika

Dataset yang digunakan yaitu *burma14.tsp* yang memiliki 14 *vertex* sebagai titik penjemputan. Pada penelitian ini diambil parameter jumlah salesman sebanyak 4 dan *vertex depot* berada pada *vertex 3* atau kota D. nilai *fitness* individu terbaik dari dataset *burma14.tsp* untuk 4 salesman yaitu 0.05209.

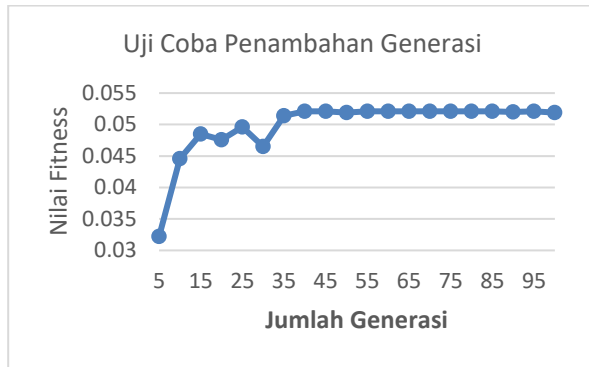
#### 3.3.1 Hasil Uji Coba Jumlah Generasi dan Populasi

Uji coba jumlah generasi dan populasi dilakukan untuk menentukan jumlah generasi dan populasi terbaik untuk dapat menghasilkan solusi terbaik pada penelitian ini.

Tabel 1. Hasil Perbandingan Jumlah Generasi dan Populasi

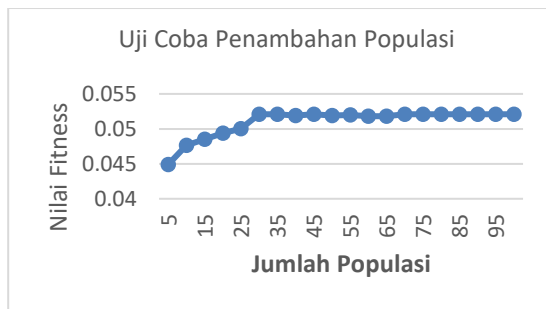
No	Generasi	Populasi	Persentase
1.	100	100	70 %
2.	1000	1000	100%
3.	10000	10000	100%

Pada Tabel 1 Dapat dilihat bahwa untuk penambahan jumlah pada generasi dan populasi maka akan semakin besar peluang untuk mendapatkan individu dengan nilai fitness terbaik. Untuk lebih jelasnya mengenai perbandingan jumlah generasi dan populasi dapat dilihat pada Gambar 6 dan Gambar 7. dimana grafik uji coba pada Gambar 6 menunjukkan bahwa semakin banyak parameter generasi yang dipakai maka nilai fitness yang didapatkan semakin meningkat. Jumlah generasi yang diuji yaitu 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, dan 100. Dari grafik hasil pengujian dapat dilihat rata-rata nilai *fitness* yang didapatkan dari generasi sebanyak 10 sampai generasi sebanyak 40 terus mengalami peningkatan. Sedangkan nilai *fitness* yang dihasilkan dengan generasi diatas 40 cenderung stabil. Hal ini menunjukkan bahwa 40 merupakan jumlah generasi yang optimal.



Gambar 6. Grafik Hasil Uji Penambahan Jumlah Generasi

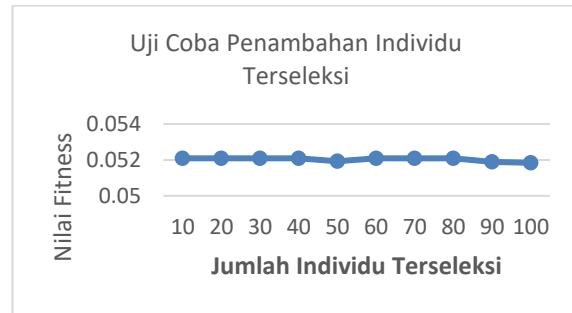
Pada grafik di gambar 7 ditunjukkan bahwa penambahan jumlah populasi membuat nilai *fitness* terbaik meningkat. Dengan melakukan pengujian dengan jumlah populasi sebanyak 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 50, 60, 65, 70, 75, 80, 85, 90, 95, dan 100. Pada jumlah populasi 5 sampai 30 mengalami peningkatan yang signifikan. Pada jumlah Populasi sebanyak 30 sampai 100 mengalami kestabilan. Ini menunjukkan bahwa jumlah populasi paling optimal adalah 30.



Gambar 7. Grafik Hasil Uji Coba Jumlah Populasi

### 3.3.2 Hasil Uji Coba Jumlah Individu Terseleksi

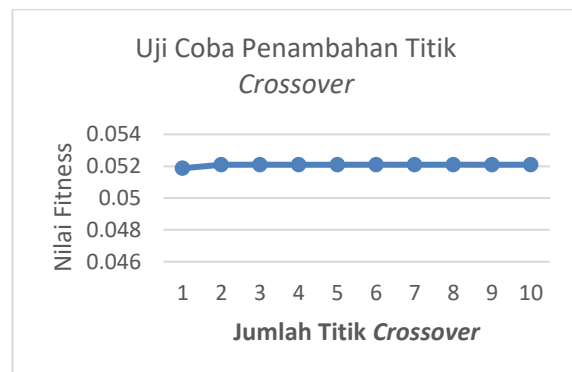
Uji coba jumlah individu terseleksi dilakukan untuk mengetahui jumlah individu terseleksi terbaik untuk dapat menghasilkan solusi terbaik pada penelitian ini. Pada gambar 8 dapat dilihat grafik ujicoba menunjukkan bahwa semakin sedikit parameter individu terseleksi yang dipakai maka nilai *fitness* yang didapatkan semakin meningkat. Jumlah individu terseleksi yang di uji yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100. Dari grafik dapat dilihat bahwa nilai *fitness* yang didapatkan dengan individu terseleksi sebanyak 10 sampai sebanyak 40 memiliki kestabilan. Kemudian mengalami penurunan pada individu terseleksi sebanyak 50. Kemudian pada individu terseleksi sebanyak 60 sampai 100 mengalami penurunan nilai *fitness*. Hal ini menunjukkan bahwa individu terseleksi sebanyak 10 merupakan jumlah individu terseleksi yang optimal. Penurunan nilai *fitness* dengan penambahan jumlah individu terseleksi terjadi karena terlalu banyak individu yang diseleksi dan digunakan dalam proses pindah silang sehingga individu dengan nilai *fitness* rendah juga ikut dalam proses pindah silang dan mutasi.



Gambar 8. Grafik Hasil Uji Coba Penambahan Individu terseleksi

### 3.3.3 Hasil Uji Coba Penambahan Titik *Crossover*

Uji coba jumlah titik *crossover* dilakukan untuk mengetahui jumlah titik pindah silang terbaik untuk dapat menghasilkan solusi terbaik pada penelitian ini. Pada gambar 9 dapat dilihat uji coba penambahan titik *crossover* tidak terlalu berpengaruh untuk mendapatkan individu *fitness* terbaik. Jumlah titik *crossover* yang di uji yaitu 1, 2, 3, 4, 5, 6, 7, 8, 9, dan 10. Dari grafik dapat dilihat bahwa nilai *fitness* yang didapatkan dengan individu terseleksi sebanyak 1 sampai sebanyak 2 mengalami kenaikan. Kemudian pada titik *crossover* sebanyak 2 sampai 10 mengalami kestabilan nilai *fitness*. Hal ini menunjukkan bahwa titik *crossover* sebanyak 2 merupakan jumlah titik *crossover* yang optimal.



Gambar 9. Grafik Hasil Uji Coba Penambahan Titik *Crossover*

## 4.2 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan dengan mengamati hasil eksekusi dari perangkat lunak. Hasil pengujian melakukan pengujian *black box* dapat dilihat pada tabel 2 berikut.

Tabel 2. Hasil Pengujian *Black Box*

No	Model Uji	Keberhasilan
1.	Input dataset dari <i>library</i>	100%
2.	Tampilkan isi dari dataset	100%
3.	Input Parameter	100%
4.	Run aplikasi	100%
5.	Tampilkan Visualisasi Rute	100%

#### 4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan dapat disimpulkan bahwa algoritma genetika dapat menyelesaikan permasalahan *Multiple Traveling Salesman Problem* hal ini dapat dilihat pada sistem yang dapat membuat rute yang optimal. Untuk dataset burma14.tsp dengan 14 titiknya yang dijadikan sebagai titik penjemputan dengan titik D sebagai titik asal dan mempunyai 4 salesman didapatkan bahwa individu terbaik memiliki nilai *fitness* sebesar 0,05209. Dengan melakukan pengujian terhadap perubahan parameter yang digunakan dapat dilihat bahwa penambahan parameter generasi dan populasi membuat peluang untuk mendapatkan individu nilai *fitness* terbaik dapat didapatkan. Untuk parameter yang paling optimal yang dapat digunakan dalam membuat penentuan rute untuk burma14 dengan 14 titik penjemputan yaitu jumlah generasi sebanyak 40, jumlah populasi sebanyak 30, jumlah individu terseleksi sebanyak 10, dan jumlah titik *crossover* sebanyak 2.

#### Daftar Rujukan

- [1] X. Chen, P. Zhang, G. Du, and F. Li, "Ant Colony Optimization Based Memetic Algorithm to Solve Bi-Objective Multiple Traveling Salesmen Problem for Multi-Robot Systems," *IEEE Access*, vol. 6, no. c, pp. 21745–21757, 2018, doi: 10.1109/ACCESS.2018.2828499.
- [2] J. Xu, L. Pei, and R. Z. Zhu, "Application of a genetic algorithm with random crossover and dynamic mutation on the travelling salesman problem," *Procedia Comput. Sci.*, vol. 131, pp. 937–945, 2018, doi: 10.1016/j.procs.2018.04.230.
- [3] Z. Xing and S. Tu, "A Graph Neural Network Assisted Monte Carlo Tree Search Approach to Traveling Salesman Problem," *IEEE Access*, vol. 8, pp. 108418–108428, 2020, doi: 10.1109/ACCESS.2020.3000236.
- [4] D. D'Ambrosio, W. Spataro, R. Rongo, and G. G. R. Iovine, *Genetic Algorithms, Optimization, and Evolutionary Modeling*, vol. 2. Elsevier Ltd., 2013.
- [5] V. Singh and S. Choudhary, "Genetic Algorithm for Traveling Salesman Problem : Using Modified Partially-Mapped Crossover Operator," pp. 20–23, 2009.
- [6] A. Hussain, Y. S. Muhammad, M. N. Sajid, I. Hussain, A. M. Shoukry, and S. Gani, "Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator," vol. 2017, pp. 1–8, 2017.
- [7] I. Sugiarto Cokrowibowo, Ismail, "Multiple Traveling Salesman Problem Menggunakan Algoritma Ant Colony Optimization dengan Operasi Elitism," *Univ. Sulawesi Barat*, vol. 1, no. 2, p. 11, 2019, doi: 10.31605/jcis.v2i1.
- [8] X. Dong and Y. Cai, "A novel genetic algorithm for large scale colored balanced traveling salesman problem," *Futur. Gener. Comput. Syst.*, vol. 95, pp. 727–742, 2019, doi: 10.1016/j.future.2018.12.065.
- [9] T. Öncan, I. K. Altinel, and G. Laporte, "A comparative analysis of several asymmetric traveling salesman problem formulations," *Comput. Oper. Res.*, vol. 36, no. 3, pp. 637–654, 2009, doi: 10.1016/j.cor.2007.11.008.
- [10] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete Spider Monkey Optimization for Travelling Salesman Problem," *Appl. Soft Comput. J.*, vol. 86, p. 105887, 2020, doi: 10.1016/j.asoc.2019.105887.