

Optimisasi *Travelling Salesman Problem* dengan Algoritma Genetika pada Kasus Pendistribusian Barang PT. Pos Indonesia di Kota Bandar Lampung

SAIFUL ROHMAN, LA ZAKARIA, ASMIATI*, *AangNuryaman*

Jurusan Matematika, Fakultas MIPA, Universitas Lampung,
Jl. Prof. Dr. Sumantri Brojonegoro No.1, Bandar Lampung, Lampung
*corresponding author : *asmianti.1976@fmipa.unila.ac.id*

Abstrak

Optimisasi merupakan sebuah proses untuk mendapatkan nilai minimum atau maksimum dari sebuah fungsi peluang. Salah satu masalah yang melibatkan proses optimisasi yaitu *Travelling Salesman Problem* (TSP). Studi kasus TSP yang dibahas dalam penelitian ini adalah berkenaan dengan permasalahan pendistribusian barang yang dilakukan oleh PT. Pos Indonesia di Kota Bandar Lampung. Beberapa permasalahan yang dimaksud antara lain keterlambatan pengiriman barang, kurang puasannya pelayanan kantor pos, rute pendistribusian barang yang rumit dan lain-lain. Penyelesaian masalah yang dihadapi oleh PT. Pos Indonesia dengan menggunakan konsep TSP yang dimaksud dapat diselesaikan dengan Metode Algoritma Genetika (algoritma pencarian yang didasarkan atas mekanisme seleksi alami dan evolusi biologis). Penggunaan metode Algoritma Genetika dalam menyelesaikan masalah yang dibahas memberikan hasil bahwa untuk mencapai posisi optimal proses pengujian yang menggunakan *order crossover* dan *inversion mutation* perlu dilakukan pengulangan proses sebanyak 8 kali. (80%).

Kata kunci: Optimisasi, TSP, PT.Pos Indonesia, Algoritma Genetika.

Abstract

Optimization is the process to get the minimum or maximum value of a function, opportunity. One of the problems involving the optimization process is Travelling Salesman Problem (TSP). The TSP case study discussed in this research is related to the problem of distribution of goods conducted by PT. Pos Indonesia in Bandar Lampung city. Some of the issues in question include delays in delivery of goods, less satisfaction in post Office services, a route to distribute complex goods and others. The resolution of the problem encountered by PT. Pos Indonesia by using the TSP concept in question can be solved by method of genetic algorithm (the search algorithm is based on natural selection mechanism and biological evolution). The use of the genetic algorithm method in resolving the problems discussed gives results that to achieve the optimal position of the testing process using the order crossover and inversion mutation need to be done repeating the process 8 times from 10 retry attempts (80%).

Keywords: Optimization, TSP, PT.Pos Indonesia, Genetic Algorithm.

1. PENDAHULUAN

Pada prinsipnya teori optimisasi secara matematis terdiri dari tiga metode pendekatan yaitu metode klasik, algoritma evolusioner, dan metode hibrid. Persoalan optimisasi pada dasarnya menentukan berbagai nilai variabel suatu fungsi menjadi maksimum atau minimum dengan memperhatikan keterbatasan yang ada. Salah satu bentuk konsep optimisasi yang terus berkembang adalah optimisasi kombinatorial. Optimisasi ini mengacu pada kasus optimisasi tertentu yang mengatur semua solusi yang mungkin dan layak dalam bentuk diskrit [9]. Salah satu jenis optimisasi kombinatorial yang sering dibahas adalah TSP. Adapun proses optimisasi pada TSP adalah menemukan rute perjalanan seorang salesman yang dimulai dari satu lokasi awal ke beberapa lokasi yang telah ditentukan, setelah itu kembali ke lokasi asal sedemikian rupa sehingga jarak total yang ditempuh minimum dan setiap kota dikunjungi tepat satu kali [4]. Tujuannya adalah menentukan panjang sirkuit terbaik dengan jarak total atau biaya yang paling minimum.

Terdapat beberapa model TSP, misalnya model TSP yang didasari oleh algoritma *k-rata-rata* melalui strategi struktur ulang rute perjalanan dengan penyambungan masing-masing *cluster* [1] dan model TSP yang dinyatakan dalam ruang 3D dengan semua titik berada pada permukaan bola [17].

Sebagai sebuah permasalahan, TSP memerlukan penyelesaian. Prosedur sederhana dalam penyelesaian masalah TSP dibagi menjadi dua metode, yaitu metode optimal dan metode aproksimasi. Metode optimal akan menghasilkan solusi yang optimal (minimum) sedangkan metode aproksimasi akan menghasilkan solusi yang mendekati optimal. Penyelesaian eksak untuk masalah TSP adalah mengharuskan perhitungan terhadap semua kemungkinan sirkuit yang dapat diperoleh, kemudian memilih sirkuit terbaik dengan jarak total atau biaya yang paling minimum [18].

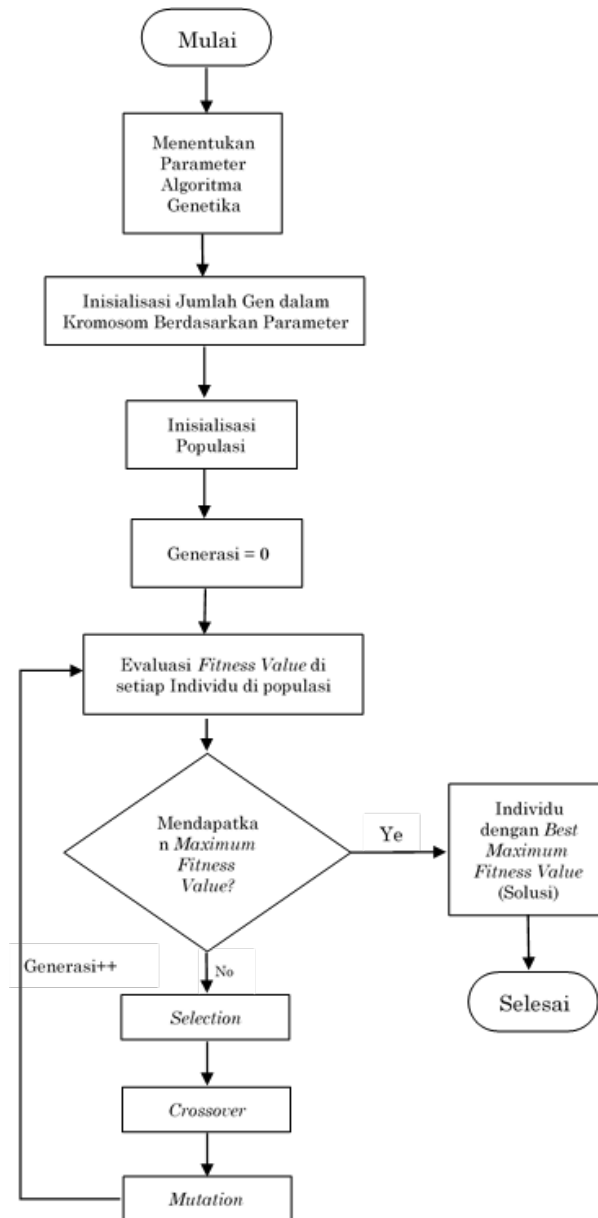
Terdapat beberapa metode yang bisa dipakai untuk melakukan pencarian sirkuit terbaik dalam sebuah TSP [6]. Algoritma genetika (GA) adalah salah satu metode yang ideal dalam menyelesaikan masalah TSP. Hubungan antara TSP dengan GA dapat diketahui melalui sejarah perkembangannya yang dapat dijumpai dalam buku [5] dan [14] dan berbagai artikel terkait, misalnya artikel [13] dan [16]. Pada prinsipnya GA dapat digunakan untuk berbagai masalah NP-Complete seperti masalah *knapsack*, *3-SAT*, *a subset*, and *vertex cover*. Salah satu aplikasi GA selain TSP adalah pengamanan jaringan [11]. Untuk model TSP yang didasari oleh algoritma *k-rata-rata* dilakukan upaya peningkatan algoritma genetik melalui strategi populasi awal baru untuk memecahkan masalah TSP [1]. Perlu dicatat bahwa algoritma genetika standar memiliki beberapa keterbatasan. Oleh karena itu tidak jarang dilakukan modifikasi Algoritma Genetika seperti yang dilakukan oleh [2].

Dalam upaya memaksimalkan kinerja GA dapat juga dikembangkan Algoritma Genetika Hibrid (HGA). Algoritma ini dirancang dengan menggabungkan varian operator *crossover* yang sudah ada dengan *Inisialisasi Heuristik* (IH), *Remove Sharp* (RS), dan *Local Opt* (LO). Sebagaimana diketahui IH hanya berlaku untuk TSP Euclidean, sementara dua heuristik lainnya, RS dan LO, dapat diterapkan untuk semua bentuk TSP simetris dan asimetris [7].

2. METODE PENELITIAN

Pandang permasalahan pendistribusian barang PT. Pos Indonesia oleh petugas di wilayah Bandar Lampung. Permasalahan ini dapat dirinci mengikuti masalah TSP sebagai berikut. Sebuah mobil pos ditugaskan mengambil dan/atau mendistribusi barang/benda pos dari kantor pos yang tersebar pada 16 buah lokasi diberbagai sudut kota di Bandar Lampung. Fenomena layanan kantor pos ini dapat dikonversikan kedalam konsep Graf. Graf dengan 17 simpul (titik-sisi) dapat digunakan untuk menyajikan persoalan ke dalam konsep TSP, Satu simpul menyatakan kantor pos tempat mobil pos mulai berangkat. Sisi (i, j) dengan $i, j \in 1, 2, \dots, 16$ diberi bobot yang sama dengan jarak dari kantor pos i ke kantor pos j . Rute yang dilalui mobil pos adalah sebuah perjalanan (*tour*) yang mengunjungi setiap kantor pos hanya satu kali dan kembali lagi ke kantor pos asal. Maka harus ditentukan rute perjalanan yang mempunyai

total jarak terpendek. Untuk mendapatkan solusi permasalahan ini akan digunakan Algoritma Genetika dengan skema penyelesaiannya mengikuti diagram alir dalam Gambar 1:



GAMBAR 1. Skema Penyelesaian Permasalahan Distribusi Barang Pos Menggunakan Algoritma Genetika

2.1. Pseudo Code Algoritma Genetika untuk Penyelesaian TSP Distribusi Barang Pos Bandar Lampung. Algoritma ini bertujuan untuk mendapatkan solusi optimal berdasarkan Algoritma Genetika untuk pendistribusian barang-barang Pos oleh Kantor Pos di Bandar Lampung menggunakan konsep TSP sehingga diperoleh sirkuit yang efisien dan efektif

Kondisi Awal: Peta koordinat posisi kantor pos diberikan/ditentukan. Selain itu, ukuran populasi, maksimum genererasi, nilai PC dan PM diberikan/ditetapkan.

Kondisi Akhir: Sirkuit/lintasan terpendek berdasarkan algoritma genetika ditampilkan dalam piranti keluaran.

Deklarasi

```

Longlat Longitude, PC, PM:Real;
Ukpop, Totgen, MaxGen:Integer;
init_data(x,y:Longitude): Real;
function Fitness = EvaluasiFit(Populasi,ukpop,totgen,Longlat) :Real;
function Seleksi = RouletteWheel(ukpop,Fitness) :Real;
function Anak = Crossover(Bapak,Ibu,totgen) :Real;
function Populasi = Kromosom(ukpop,totgen) :Real;
Deskripsi {Main Program}
Begin
ukpop      29;
totgen     length(Longlat)-1;
PC         0.95;
PM         0.01;
MaxGen     100;
Read(init_data);
For i 1 to MaxGen Do
Begin
Fitness ;
Seleksi;
Anak;
Populasi;
End{For}

Write(solusi terbaik)

End{Begin}

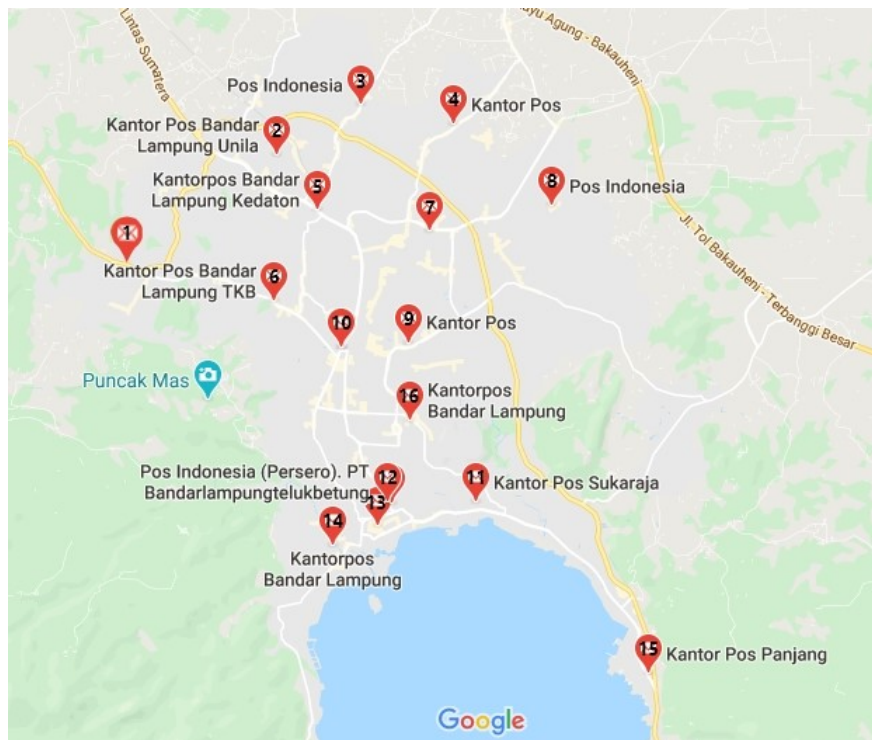
```

3. HASIL DAN PEMBAHASAN

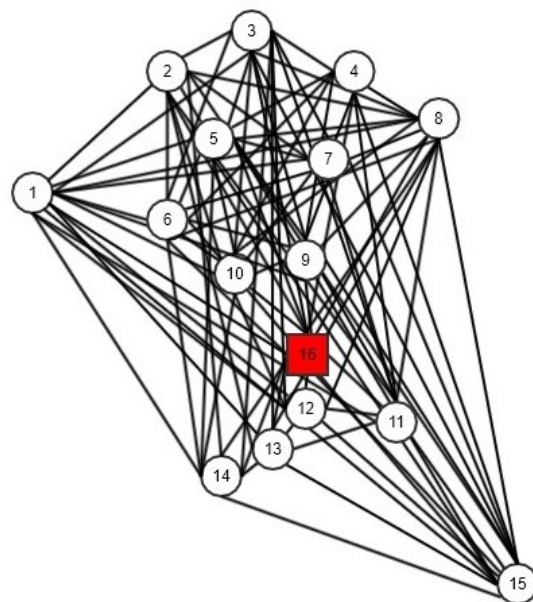
3.1. TSP dan Algoritma Genetika pada Pendistribusian Barang PT. Pos Bandar Lampung. Proses pendistribusian barang yang dilakukan oleh seorang *Postman* PT.Pos Indonesia di Kota Bandar Lampung dimulai dari Kantor Pos Pemeriksa (Kprk), selanjutnya bergerak ke-15 titik Kantor Pos Cabang Dalam Kota (KPCDK) Bandar Lampung dan kembali lagi ke Kprk. Adapun rute (*route*) yang dilalui untuk setiap KPCDK hanya dilalui satu kali. Untuk memudahkan wilayah kerjadalam proses pendistribusian barang yang dilakukan oleh seorang *Postman*, *Google Maps* digunakan untuk memodelkan titik-titik Kantor Pos Pemeriksa (Kprk) dan Kantor Pos Cabang Dalam Kota (KPCDK) Bandar Lampung.

Berdasarkan titik-titik lokasi dan koordinat *decimal degrees* yang telah ditentukan, maka akan dilakukan plotting titik-titik lokasi. Menurut [8] ada beberapa metode *encoding* yang secara umum sering digunakan salah satunya adalah *permutation encoding*. Dalam *plotting* ini setiap titik lokasi akan diberi kode dengan *permutation encoding* agar wilayah kerja tersebut nantinya dapat di-*encode* ke dalam kromosom pada algoritma genetika [15]. Berikut ini adalah model rute pendistribusian barang PT.Pos Indonesia Kota Bandar Lampung serta titik-titik koordinat yang diperoleh dari 16 titik-titik lokasi Kprk dan KPCDK.

Berdasarkan data Tabel 1 di atas Kprk yang akan menjadi titik start dan finish seorang *Postman* dalam mendistribusikan barangnya di-*encoding* pada titik ke-16 dan 15 titik lainnya merupakan KPCDK Bandar Lampung yang menjadi titik lokasi tujuan *Postman*.



GAMBAR 2. Titik-titik lokasi Kprk dan KPCDK Kota Bandar Lampung



GAMBAR 3. Model graf lengkap rute pendistribusian barang PT. Pos Indonesia Kota Bandar Lampung

3.2. Proses Penyelesaian Masalah Menggunakan Algoritma Genetika. Adapun proses penyelesaian masalah menggunakan algoritma genetika ini terdiri dari 6 tahapan, yakni sebagai berikut:

TABEL 1. Data titik-titik lokasi Kprk dan KPCDK Bandar Lampung

TITIK	ALAMAT KANTOR POS	X (<i>Longitude</i>)	Y (<i>Latitude</i>)
1	Jl. Ganjaran No.1297, Kemiling Permai, Kec. Kemiling, Kota Bandar Lampung, Lampung 35135	1,052,098	-5,389,706
2	Jl. Prof. Dr. Ir. Sumantri Brojonegoro, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung, Lampung 35141	105,243,224	-5,367,424
3	Jalan RA Basyid, Labuhan Dalam, Kec. Tj. Senang, Kota Bandar Lampung, Lampung 35141	105,261,765	-5,356,445
.	.	.	.
.	.	.	.
.	.	.	.
16	Jl. KH. Ahmad Dahlan No.21, Pahoman, Engal, Kota Bandar Lampung, Lampung 35000	105,272,523	-5,424,675

- Pembangkitan populasi awal. Berdasarkan saran dari [10] penentuan ukpop = 30, pc = 0,95, pm = 0,01, dan maksimal generasi = 100, hal ini bertujuan agar *fitnessvalue* setiap generasi digunakan sebagai indikator optimisasi.
- Evaluasi *fitness*, melakukan inverse dari total jarak setiap kromosom dengan perhitungan $\frac{1}{\text{nilaijarakkromosom}}$.
- Seleksi, proses seleksi dilakukan dengan menggunakan *elitisme* dan *roulette wheel selection*.
- Crossover* dengan menggunakan metode *order crossover* (OX).
- Mutasi, proses mutasi dilakukan dengan menggunakan metode *inversion mutation*.

3.2.1. *Pembangkitan Populasi Awal*. Pembangkitan populasi awal dilakukan dengan membangkitkan gen sebanyak titik-titik lokasi KPCDK yakni berjumlah 15 titik dalam satu populasi (kromosom) dan dibangkitkan sebanyak 30 populasi (kromosom) dalam satu generasinya. Proses pembangkitan populasi (kromosom) di Matlab 2014a menggunakan fungsi *rand*, sehingga proses didasarkan pada proses pencarian acak (*random search*). Berikut hasil populasi (kromosom) yang dibangkitkan:

TABEL 2. Hasil pembangkitan kromosom (populasi awal)

Populasi (Kromosom)	Susunan Gen
1	11 9 2 6 4 1 10 15 14 7 5 8 12 3 13
2	14 7 12 2 13 15 6 1 9 10 11 4 8 5 3
.	.
.	.
.	.
30	8 7 6 4 15 14 3 9 5 11 2 10 1 13 12

Dikarenakan perjalanan dimulai dari Kprk, dan rute akan selalu berakhir di Kprk, maka, setiap gen pertama dan terakhir akan bernilai sama yaitu 16. Sehingga inisialisasi kromosom (populasi awal) tidak ditampilkan. Hal ini berdasarkan pada inisial *permutation encoding* dari Kprk pada sampel data.

3.2.2. *Evaluasi Fitness*. Selanjutnya setiap kromosom menjalani proses evaluasi untuk mendapatkan nilai *fitness* dari masing-masing kromosom. Pertama, melakukan perhitungan jarak rute semua kromosom yang telah dibangkitkan menggunakan rumus *euclid distance*. Karena titik koordinat yang digunakan masih dalam satuan *decimal degree* (Setiawan dkk., 2018). Jika

dikonversi kedalam kilometer maka diperoleh nilai *decimal degree* = 111,319 km. Sebagaimana rumus berikut:

$$r_{i,j} = \left(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times 111.319 \text{ km} \right) \quad (1)$$

dengan:

- $r_{i,j}$: jarak dari titik i ke titik j , untuk setiap $i, j = 1, 2, \dots, n$
- x_1 : titik koordinat x (*Longitude*) pada titik i
- x_2 : titik koordinat x (*Longitude*) pada titik j
- y_1 : titik koordinat y (*Latitude*) pada titik i
- y_2 : titik koordinat y (*Latitude*) pada titik j

Misalkan proses perhitungan pada kromosom 1 dengan rute [11 9 2 6 4 1 10 15 14 7 5 8 12 3 13]. Maka jarak rute yang ditempuh adalah sebagai berikut:

$$\begin{aligned} \text{Kromosom 1} &= (11 \rightarrow 9) + (9 \rightarrow 2) + (2 \rightarrow 6) + (6 \rightarrow 1) + (1 \rightarrow 10) + (10 \rightarrow 15) + (15 \rightarrow 14) + \\ &\quad (14 \rightarrow 7) + (7 \rightarrow 5) + (5 \rightarrow 8) + (8 \rightarrow 12) + (12 \rightarrow 3) + (3 \rightarrow 13) \\ &= 4,1872 + 5,5404 + 3,5075 + 6,0751 + 8,6432 + 5,7399 + 10,8187 + \\ &\quad 8,2545 + 7,9645 + 2,7708 + 5,7057 + 8,1569 + 9,6422 + 10,1621 \\ &= 97,1686 \text{ km} \end{aligned}$$

Adapun untuk menghitung *Fitness Value* pada kromosom 1, adalah *Fitness Kromosom 1* = $1/97,1686$. Sehingga *Fitness Value* pada kromosom 1 adalah 0,0103.

Dengan melakukan langkah penghitungan yang sama, akan diperoleh jarak rute dan *Fitness Value* pada kromosom lainnya (lihat Tabel 3).

TABEL 3. Jarak dan *fitness value* populasi (kromosom)

Populasi (Kromosom)	Susunan Gen	Jarak (km)	<i>Fintess Value</i>
1	11 9 2 6 4 1 10 15 14 7 5 8 12 3 13	971,686	0,0103
2	14 7 12 2 13 15 6 1 9 10 11 4 8 5 3	912,698	0,0110
.	.	.	.
.	.	.	.
.	.	.	.
30	8 7 6 4 15 14 3 9 5 11 2 10 1 13 12	939,540	0,0106

Berdasarkan pada Tabel 3, kromosom ke-13 memiliki *Fitness Value* paling tinggi pada generasi ke-1 ini yaitu 0,0134 dengan susunan kromosom = [11 9 2 6 4 1 10 15 14 7 5 8 12 3 13]. Jarak rute yang didapatkan pada kromosom ini adalah 74.7652 km. selanjutnya kromosom ini disimpan sebagai kromosom elit yang akan disalin secara langsung ke populasi selanjutnya pada proses elitisme.

3.3. Seleksi. Total *fitness* seluruh kromosom pada populasi adalah 0,3343. Kemudian dihitung nilai *fitness* relatif setiap kromosom. Proses perhitungannya adalah sebagai berikut:

$$\begin{aligned} P_1 &= 0,0103/0,3343 = 0,0308 \\ P_2 &= 0,0110/0,3343 = 0,0329 \\ P_3 &= 0,0128/0,3343 = 0,0383 \\ &\vdots \\ P_{30} &= 0,0106/0,3343 = 0,0317 \end{aligned}$$

Selanjutnya hitung *fitness* kumulatif (probabilitas kumulatif) dari setiap *fitness* relatif yang telah didapatkan sebagai berikut:

$$\begin{aligned} C_0 &= 0 \\ C_1 &= 0 + 0,0308 = 0,0308 \\ C_2 &= 0,0308 + 0,0383 = 0,0637 \\ &\vdots \\ C_{30} &= 0,9683 + 0,0317 = 1 \end{aligned}$$

Dari hasil perhitungan *fitness* relatif dan kumulatif, selanjutnya proses seleksi kromosom dilakukan berdasarkan proporsi dari *fitness* kumulatifnya yang disajikan pada Tabel 4 berikut:

TABEL 4. *Fitness* relatif dan kumulatif

Populasi (Kromosom)	Jarak (km)	Fintess Value	Fitness Relatif	Fitness Kumulatif
1	971,686	0,0103	0,0308	0,0308
2	912,698	0,0110	0,0329	0,0637
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
30	939,540	0,0106	0,0317	10,000

Selanjutnya, melakukan proses Seleksi Elitisme dan *Roulette Wheel* (Roda Roulette) dimana metode ini semua kromosom (individu) dalam populasi ditempatkan pada roda roulette berdasarkan proporsi nilai *fitness* [3]. Pertama, melakukan proses Elitisme dengan mengkopi kromosom terbaik, yakni kromosom ke-13 dengan susunan kromosom = [11 9 2 6 4 1 10 15 14 7 5 8 12 3 13] ke sebuah temporary populasi. Karena jumlah populasi (kromosom) bernilai genap yaitu 30 populasi (kromosom). Maka, Proses pengkopian kromosom terbaik dilakukan dua kali. Sehingga Induk ke-1 dan Induk ke-2 susunan kromosomnya adalah [11 9 2 6 4 1 10 15 14 7 5 8 12 3 13]. Proses ini dilakukan bertujuan agar individu terbaik tetap ada dan tidak hilang maupun rusak saat terjadi proses genetis berupa *crossover* (pindah silang) dan mutasi. Kedua, membangkitkan bilangan acak $R(i)$ antara 0 sampai 1 dengan $i=3, \dots, 30$. Misal bilangan acak yang telah dibangkitkan sebagai berikut:

$$\begin{aligned} R_3 &= 0,0431 \\ R_4 &= 0,3895 \\ &\vdots \\ R_{30} &= 0,0586 \end{aligned}$$

Sehingga kromosom induk yang terpilih untuk menjalani proses reproduksi adalah sebagai berikut:

TABEL 5. Kromosom induk yang terpilih

Induk	$R(i)$	Kromosom Terpilih	Susunan Gen
3	0,0431	Kromosom 6	5 15 12 8 7 2 9 13 14 3 4 11 1 10 6
4	0,3895	Kromosom 12	15 8 13 3 12 1 9 11 14 4 10 6 5 2 7
$\vdots$	$\vdots$	$\vdots$	$\vdots$
30	0,0586	Kromosom 6	5 15 12 8 7 2 9 13 14 3 4 11 1 10 6

Sehingga susunan kromosom hasil proses Seleksi Elitisme dan *Roulette Wheel Selection* adalah sebagai berikut:

TABEL 6. Kromosom hasil proses seleksi elitisme dan *roulette wheel selection*

Induk	Kromosom Terpilih	Susunan Gen
1	Kromosom 13	11 9 2 6 4 1 10 15 14 7 5 8 12 3 13
2	Kromosom 13	11 9 2 6 4 1 10 15 14 7 5 8 12 3 13
$\vdots$	$\vdots$	$\vdots$
30	Kromosom 6	5 15 12 8 7 2 9 13 14 3 4 11 1 10 6

3.3.1. *Crossover*. Proses *Crossover* pada penelitian ini menggunakan metode *order crossover* [12]. Pasangkan kromosom induk secara acak sebanyak setengah dari ukuran populasi dan tentukan nilai probabilitas *crossover* (pc). Misal nilai pc yang ditentukan adalah 0,95 dan susunan pasangan induk adalah sebagai berikut:

Pasangan 1	Induk 3 Induk 4
Pasangan 2	Induk 5 Induk 6
Pasangan 3	Induk 7 Induk 8
	$\vdots$
Pasangan 14	Induk 29 Induk 30

Membangkitkan bilangan acak R_i antara 0 sampai 1 sebanyak jumlah pasangan induk. Jika bilangan acak R_i kurang atau sama dengan pc, maka pasangan ke- i akan mengalami *crossover*. Jika sebaliknya, maka pasangan tersebut masuk ke tahap berikutnya. Misal bilangan acak yang dibangkitkan adalah sebagai berikut:

$$\begin{aligned} R_1 &= 0,3840 \\ R_2 &= 0,9558 \\ &\vdots \\ R_{14} &= 0,9550 \end{aligned}$$

Dari bilangan acak yang telah dibangkitkan, terdapat 6 nilai yang bilangan acaknya yang kurang dari pc yaitu $R_1, R_3, R_7, R_8, R_{12}, R_{13}$ yang akan mengalami *crossover*, dan sisanya akan masuk ke tahap berikutnya tanpa mengalami *crossover*. Berikut ini merupakan proses *order crossover*(OX) pada pasangan 1 yang telah terpilih antara Induk 3 dengan Induk 4.

Induk 3	5	15	12	8	7	2	9	13	14	3	4	11	1	10	6
Induk 4	15	8	13	3	12	1	9	11	14	4	10	6	5	2	7

Pertama, menentukan posisi awal dan posisi akhir gen yang akan di-*crossover* dengan cara membangkitkan dua bilangan acak dengan fungsi randi pada Matlab 2014a. Dalam kasus ini posisi awal gen yang akan di-*crossover* = 4, dan posisi akhir gen yang akan di-*crossover* = 11.

Induk 3	5	15	12	8	7	2	9	13	14	3	4	11	1	10	6
Induk 4	15	8	13	3	12	1	9	11	14	4	10	6	5	2	7

Mengkopikan daerah gen terpilih yang akan dilakukan proses *crossover*, sebagai *offspring Child-1* dan *Child-2*

<i>Child-1</i>				7	2	9	13	14	3	4				
<i>Child-2</i>				12	1	9	11	14	4	10				

Untuk mengkonstruksi *Child 1* yang pertama dilakukan adalah dengan menyusun barisan pada **Induk 4**:

Induk 4	15	8	13	3	12	1	9	11	14	4	10	6	5	2	7
---------	----	---	----	---	----	---	---	----	----	---	----	---	---	---	---

Kemudian menghapus barisan **Induk 4** yang memiliki kesamaan dengan barisan pada *Child-1* yaitu 7, 2, 9, 13, 14, 3, 4.

Induk 4	15	8	13	3	12	1	9	11	14	4	10	6	5	2	7
---------	----	---	----	---	----	---	---	----	----	---	----	---	---	---	---

Maka, susunan barisan **Induk 4** yang tersisa 15 8 12 1 11 10 6 5 diletakkan pada substring yang kosong pada *Child-1*. Kemudian masukkan susunan barisan Induk 4 yang tersisa dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu dan sisanya dimasukkan pada posisi sebelum bilangan acak pertama, sehingga diperoleh *Child 1*

<i>Child-1</i>	11	10	6	5	7	2	9	13	14	3	4	15	8	12	1
----------------	----	----	---	---	---	---	---	----	----	---	---	----	---	----	---

Dengan cara yang sama untuk mengkonstruksi *Child-2*. Diperoleh sebagai berikut:

<i>Child-2</i>	2	13	3	6	12	1	9	11	14	4	10	5	15	8	7
----------------	---	----	---	---	----	---	---	----	----	---	----	---	----	---	---

Demikian proses *Crossover* ini berlangsung dengan cara yang sama pada pasangan induk terpilih lainnya yaitu Pasangan 3, Pasangan 7, Pasangan 8, Pasangan 12, Pasangan 13. Berikut ini susunan gen setelah melakukan proses *crossover*.

TABEL 7. Hasil pores *crossover*

No	Kromosom	Susunan Gen Setelah <i>Crossover</i>
1	Kromosom 1	6 13 11 14 9 10 3 4 2 1 8 15 12 7 5
2	Kromosom 2	6 13 11 14 9 10 3 4 2 1 8 15 12 7 5
⋮	⋮	⋮
30	Kromosom 30	14 6 15 5 8 7 1 11 10 12 13 2 4 3

3.3.2. *Mutasi*. Selanjutnya kromosom akan menjalani proses mutasi. Nilai probabilitas mutasi (pm) ditentukan sebesar 0,01. Lalu bangkitkan bilangan acak antara 0 sampai 1 sebanyak jumlah populasi. Pada proses mutasi ini menggunakan metode *inversion mutation*. Misalkan bilangan acak yang telah dibangkitkan sebagai berikut:

$$R_1 = 0,0607$$

$$R_2 = 0,5942$$

⋮

$$R_{30} = 0,1009$$

Karena bilangan acak yang telah dibangkitkan lebih besar dari pm. Maka, tidak ada kromosom yang menjalankan proses mutasi. Sehingga populasi baru yang didapatkan sama dengan populasi sebelum menjalankan proses mutasi.

TABEL 8. Hasil proses mutasi

No	Kromosom	Susunan Gen Setelah Mutasi
1	Kromosom 1	6 13 11 14 9 10 3 4 2 1 8 15 12 7 5
2	Kromosom 2	6 13 11 14 9 10 3 4 2 1 8 15 12 7 5
⋮	⋮	⋮
30	Kromosom 30	14 6 15 5 8 7 1 11 10 12 13 2 4 3

3.3.3. *Evaluasi Fitness*. Pada tahap ini hasil kromosom pasca dilakukan proses mutasi akan dilakukan proses evaluasi *fitness* kembali. Karena perjalanan seorang Postman dimulai dari Kprk, dan rute akan selalu berakhir di Kprk. Maka, ditambahkan gen pertama dan gen terakhir pada setiap kromosom dengan nilai 16. Hal ini berdasarkan pada tahap *encoding* (inisialisasi awal). Selanjutnya menghitung jarak dan *fitness value* masing-masing kromosom. Berikut ini hasil evaluasi *fitness* generasi ke-1:

TABEL 9. Hasil evaluasi *fitness*

Kromosom	Susunan Gen	Jarak (km)	<i>Fitness Value</i>
1	16 6 13 11 14 9 10 3 4 2 1 8 15 12 7 5 16	84,6943	0,0118
2	16 6 13 11 14 9 10 3 4 2 1 8 15 12 7 5 16	84,6943	0,0118
⋮	⋮	⋮	⋮
30	16 14 9 6 15 5 8 7 1 11 10 12 13 2 4 3 16	98,5209	0,0102

Jadi, Hasil evaluasi generasi ke-1 diperoleh kromosom terbaik ada pada kromosom ke-26 dengan nilai *fitness* adalah 0,0126 dan panjang sirkuit terbaik yang ditempuh sebesar 79,1541 km. Adapun sirkuit terbaik yang dilalui yaitu sebagai berikut: [16-3-4-2-1-5-8-15-12-7-10-9-6-13-11-14-16].

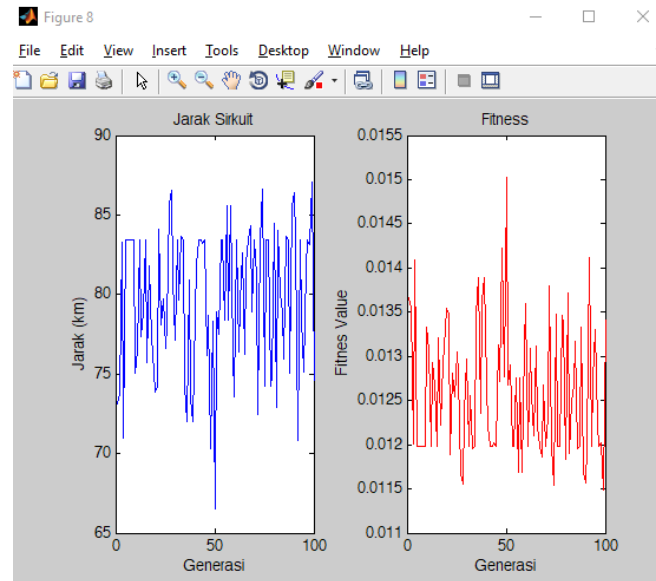
3.3.4. *Pengujian Data*. Pengujian dilakukan dengan menggunakan nilai parameter ukuran populasi = 30, maksimum generasi = 100, probabilitas *crossover* = 0,95 dan probabilitas mutasi = 0,01. Proses pengujian data dilakukan sebanyak 10 kali. Sedangkan untuk operator Algoritma Genetika menggunakan *order crossover* dan *Inversion Mutation*. Adapun solusi terbaik dilihat berdasarkan nilai *fitness* yang terbesar, sehingga diperoleh hasil pengujiannya sebagai berikut:

TABEL 10. Hasil pengujian

Pengujian Ke-	Fitness Tertinggi	Jarak Minimal (km)	Generasi	Sirkuit Terbaik
1	0,0148156	6,749,642,403	91	16 2 1 5 3 4 8 7 6 10 15 12 11 9 13 14 16
2	0,013601282	735,224,837	18	16 15 14 12 2 5 4 3 1 9 8 7 10 6 13 11 16
3	0,014021141	7,132,087,351	75	16 9 15 11 8 4 7 1 6 10 14 12 13 5 3 2 16
4	0,014862579	6,728,307,341	59	16 10 11 14 9 7 8 4 2 3 5 6 1 12 15 13 16
5	0,014192344	7,046,052,355	72	16 11 9 4 2 3 8 5 7 10 6 1 12 15 13 14 16
6	0,014515168	6,889,345,158	49	16 7 8 3 4 2 1 14 15 11 9 5 10 6 13 12 16
7	0,01394258	7,172,273,464	97	16 15 14 12 11 13 6 9 7 5 3 1 10 2 4 8 16
8	0,015032531	6,652,239,581	50	16 9 7 2 6 1 4 8 3 5 10 12 13 11 14 15 16
9	0,014898102	671,226,433	10	16 14 15 13 11 12 10 5 6 1 7 9 2 3 4 8 16
10	0,014764738	6,772,893,562	92	16 13 14 12 15 11 1 3 2 9 10 5 4 8 7 6 16

Dari Tabel 10 diatas diperoleh hasil terbaik ada pada pengujian ke-8 dengan *fitness value* = 0,015032531, dengan panjang sirkuit minimal = 66,52239581 km, solusi terbaik ini ditemukan pada generasi ke-50. Susunan rute optimal yang didapatkan adalah 16 9 7 2 6 1 4 8 3 5 10 12 13 11 14 15 16. Grafik hasil Pengujian ke-8 yang diperoleh adalah sebagai berikut:

Pada generasi pertama diperoleh nilai jarak yang didapatkan berada pada kisaran 70-75 km dengan nilai *fitness* antara 0,0135 0,0140, kemudian proses pencarian jarak terbaik terus mengalami fluktuatif akhirnya didapatkan pada generasi ke-50 dengan nilai jarak 66,52239581

GAMBAR 4. Diagram hubungan jarak dan *fitness value* hasil pengujian ke-8

km dan nilai fitness 0,015032531, sampai generasi ke-100 tidak ditemukan kembali jarak nilai fitness dan jarak yang lebih kecil dari nilai jarak tersebut. Adapun sirkuit terbaik pada percobaan ke-8 ini adalah: [16 9 7 2 6 1 4 8 3 5 10 12 13 11 14 15 16].

Adapun untuk lokasi pendistribusian barang yang harus dilakukan oleh seorang postman berdasarkan sirkuit terbaik menurut algoritma genetika adalah sebagaimana diberikan dalam Tabel 11:

TABEL 11. Urutan perjalanan *postman* pembentuk sirkuit terbaik berdasarkan Algoritma Genetika

No	No Kode Kantor Pos	Lokasi
1.	16	Kantor Pos Jl. KH. Ahmad Dahlan No.21, Pahoman, Engal, Kota Bandar Lampung,
2.	9	Kantor Pos Jl. Pangeran Antasari No.67, Kedamaian, Kota Bandar Lampung,
3.	7	Kantor Pos Jl. Griya Utama, Way Halim Permai, Way Halim, Kota Bandar Lampung,
4.	2	Kantor Pos Jl. Prof. Dr. Ir. Sumantri Brojonegoro, Gedong Meneng, Kec. Rajabasa, Kota Bandar Lampung,
5.	6	Kantor Pos Jalan Imam Bonjol No.297, Kel. Gedongair, Kec. Tanjung Karang Barat, Gedong Air, Lampung, Gedong Air, Kec. Tj. Karang Bar., Kota Bandar Lampung,
6.	1	Kantor Pos Jl. Ganjaran No.1297, Kemiling Permai, Kec. Kemiling, Kota Bandar Lampung,
7.	4	Kantor Pos Jalan Ratu Dibalau No. 04, Tj. Senang, Kec. Tj. Senang, Kota Bandar Lampung,
8.	8	Kantor Pos Jalan Letnan Kolonel Endro Suratmin, Harapan Jaya, Sukrame, Kota Bandar Lampung, Kota Bandar Lampung,
9.	3	Kantor Pos Jalan RA Basyid, Labuhan Dalam, Kec. Tj. Senang, Kota Bandar Lampung,
10.	5	Kantor Pos Jalan Zaenal Abidin Pagar Alam, Rajabasa, Labuhan Ratu, Kec. Kedaton, Kota Bandar Lampung,
11.	10	Kantor Pos Jalan Kota Raja No.12, Gunung Sari, Engal, Gunung Sari, Engal, Gn. Sari, Engal, Kota Bandar Lampung,
12.	12	Kantor Pos Jl. P. Hasanuddin, No. 41, Teluk Betung, Bandar Lampung,
13.	13	Kantor Pos Jl. Ikan Gurame No.11, Pesawahan, Kec. Telukbetung Selatan, Kota Bandar Lampung,
14.	11	Kantor Pos Jl. Ikan Tembakang No.2, Sukaraja, Kec. Telukbetung Selatan, Kota Bandar Lampung,
15.	14	Kantor Pos Jl. Banten No.8, Perwata, Tlk. Betung Bar., Kota Bandar Lampung,
16.	15	Kantor Pos Jl. Yos Sudarso, Panjang Sel., Kec. Panjang, Kota Bandar Lampung,
17.	16	Kantor Pos Jl. KH. Ahmad Dahlan No.21, Pahoman, Engal, Kota Bandar Lampung

4. SIMPULAN

Proses optimisasi TSP menggunakan algoritma genetika terdapat tiga tahapan yaitu, tahap pertama membuat model pendistribusian barang PT. Pos Indonesia di kota Bandar Lampung menggunakan graf lengkap. Tahap kedua merupakan proses optimisasi pendistribusian barang PT. Pos Indonesia di kota Bandar Lampung menggunakan Algoritma Genetika, pada tahap ini parameter algoritma genetika yang digunakan yaitu *probability crossover* = 0.95, *probability mutation* = 0.01, dan *maximal generation* = 100 generasi. Tahap ketiga merupakan proses optimisasi dengan melakukan pengujian sebanyak 10 kali. Hasil dari 10 kali pengujian diperoleh hasil terbaik pada pengujian ke-8 dengan nilai *fitness* = 0,015032531, dengan jarak minimal 66,52239581 km, solusi ini ditemukan pada generasi ke-50. Dalam kasus yang ditinjau dalam artikel ini parameter optimisasi hanya satu variabel yaitu panjang sirkuit terbaik. Saran untuk penelitian berikutnya adalah memperhatikan kapasitas angkut kendaraan, estimasi bahan bakar kendaraan, dan waktu operasional yang diperbolehkan. .

DAFTAR PUSTAKA

- [1] Yong Deng, Yang Liu, and Deyun Zhou. An improved genetic algorithm with initial population strategy for symmetric tsp. *Mathematical Problems in Engineering*, 2015, 2015.
- [2] Chunhua Fu, Lijun Zhang, Xiaojing Wang, and Liying Qiao. Solving tsp problem with improved genetic algorithm. In *AIP Conference Proceedings*, volume 1967, page 040057. AIP Publishing LLC, 2018.
- [3] David E Goldberg. Genetic algorithms in search. *Optimization, and Machine Learning*, 1989.
- [4] Gregory Gutin, Anders Yeo, and Alexei Zverovitch. Exponential neighborhoods and domination analysis for the tsp. In *The traveling salesman problem and its variations*, pages 223–256. Springer, 2007.
- [5] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [6] Muhammad Irfan. Penyelesaian travelling salesman problem (tsp) menggunakan algoritma hill climbing dan matlab. *Matematika*, 17(1), 2018.
- [7] G Andal Jayalakshmi, S Sathiamoorthy, and R Rajaram. A hybrid genetic algorithm a new approach to solve traveling salesman problem. *International Journal of Computational Engineering Science*, 2(02):339–355, 2001.
- [8] I Dewa Made Adi Baskara Joni and Vivine Nurcahyawati. Penentuan jarak terpendek pada jalur distribusi barang di pulau jawa dengan menggunakan algoritma genetika. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, 1(3):244–258, 2012.
- [9] André A Keller. *Mathematical Optimization Terminology: A Comprehensive Glossary of Terms*. Academic Press, 2017.
- [10] Sri Kusumadewi and Hari Purnomo. Penyelesaian masalah optimasi dengan teknik-teknik heuristik. *Yogyakarta: Graha Ilmu*, 2005.
- [11] Hari Mohan Pandey. Performance evaluation of selection methods of genetic algorithm and network security concerns. *Physics Procedia*, 78:13–18, 2016.
- [12] Yurika Permanasari and Ravi A Salim. Representasi jalur (path) pada traveling salesman problem untuk menentukan jarak terpendek menggunakan algoritma genetika. *Matematika*, 6(1), 2007.
- [13] Jan Scholz. Genetic algorithms and the traveling salesman problem a historical review. *arXiv preprint arXiv:1901.05737*, 2019.
- [14] SN Sivanandam and SN Deepa. Introduction to genetic algorithms, 2008.
- [15] Admi Syarif. Algoritma genetika teori dan aplikasi edisi 2. *Yogyakarta: Graha Ilmu*, 2014.
- [16] Chun-Wei Tsai, Shih-Pang Tseng, Ming-Chao Chiang, Chu-Sing Yang, and Tzung-Pei Hong. A high-performance genetic algorithm: using traveling salesman problem as a case. *The Scientific World Journal*, 2014, 2014.
- [17] Aybars Uğur, Serdar Korukoğlu, Ali Çalışkan, Muhammed Cinsdikici, and Ali Alp. Genetic algorithm based solution for tsp on a sphere. *Mathematical and Computational Applications*, 14(3):219–228, 2009.
- [18] Rhett Wilfahrt, Sangho Kim, S Shekhar, H Xiong, and X Zhou. Traveling salesman problem (tsp)., 2017.