

# Pencarian Rute Terbaik Pada Travelling Salesman Problem (TSP) Menggunakan Algoritma Genetika pada Dinas Kebersihan dan Pertamanan Kota Pekanbaru

Medrio Dwi Aksara Cipta Hasibuan  
STMIK-AMIK Riau - Pekanbaru  
medrio@yahoo.com

Lusiana  
STMIK-AMIK Riau – Pekanbaru  
lusi\_dl@yahoo.co.id

## Abstrak

*Travelling Salesman Problem (TSP) merupakan salah satu permasalahan optimasi yang terjadi di kehidupan sehari-hari. Permasalahan pada kasus TSP adalah bagaimana membangun rute terpendek yang akan dilalui salesman. Pemodelan kasus ini akan diterapkan pada kegiatan pengangkutan sampah di Dinas Kebersihan dan Pertamanan Kota Pekanbaru (DKP). Saat ini rute pengangkutan sampah yang dikelola DKP masih dilakukan secara konvensional sehingga belum dapat dipastikan bahwa rute yang dibangun sudah optimal. Untuk membangun rute yang optimal pada TSP, penulis mengusulkan menggunakan algoritma genetika. Algoritma genetika merupakan suatu metode yang digunakan untuk memecahkan suatu permasalahan pencarian solusi optimal dalam sebuah masalah optimasi. Algoritma ini mengikuti proses genetik dari organisme-organisme biologi yang berdasar pada teori evolusi Charles Darwin. Implementasi algoritma genetika menguji kemampuan algoritma genetika, dalam mencari solusi rute optimal pada kegiatan pengangkutan sampah. Berdasarkan hasil pengujian yang dilakukan pada tiga wilayah kerja pengangkutan sampah yang disusun oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru, dapat disimpulkan bahwa algoritma genetika mampu menghasilkan solusi rute yang optimal.*

*Kata Kunci : TSP, Algoritma Genetika, Optimasi*

## 1. Pendahuluan

### 1.1. Latar Belakang

Optimasi merupakan suatu cara dalam pencarian hasil yang terbaik dengan tujuan untuk mendapatkan solusi nilai-nilai yang mendekati optimal dalam suatu permasalahan. Dalam kehidupan sehari-hari, permasalahan tentang optimasi cukup sering terjadi. Salah satu contoh kasus yang sering dijumpai adalah permasalahan tentang penentuan rute terpendek yang

dikenal dengan model kasus *Travelling Salesman Problem* atau disingkat dengan TSP.

Model kasus TSP yang sebenarnya yaitu terdapat seorang *salesman* yang akan mengunjungi sejumlah  $n$  kota. Namun, seluruh kota harus dikunjungi dan setiap kota hanya dapat dikunjungi tepatnya satu kali. Permasalahannya adalah bagaimana *salesman* tersebut dapat menentukan rute terpendek yang akan dilalui dalam mengunjungi seluruh kota dan kembali lagi ke kota awal. Model TSP ini dapat juga kita implementasikan pada kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru.

Pemerintah kota Pekanbaru melimpahkan wewenang pengelolaan kebersihan kota kepada Dinas Kebersihan dan Pertamanan Kota Pekanbaru yang dikelola oleh Bidang Kebersihan Kota. Kegiatan pengangkutan sampah dilakukan secara *mobile*. Artinya petugas angkut mengambil sampah dengan menyisir di sepanjang jalan raya yang rawan terjadinya tumpukan sampah. Hal ini terjadi dikarenakan kurangnya kesadaran masyarakat untuk membuang sampah ke tempat pembuangan sementara (TPS). Sehingga Bidang Kebersihan Kota dalam menyusun wilayah kerja pengangkutan sampah bukan berdasarkan lokasi TPS yang tersebar. Namun berdasarkan jalan utama yang banyak terjadi tumpukan sampah dan rute yang dibangun hanya menunjuk pada jalan-jalan utama yang harus disisir untuk setiap masing-masing petugas pengangkut sampah. Sehingga pembangunan rute masih dilakukan secara konvensional dan belum dapat dipastikan bahwa jalur rute yang dibangun sudah optimal. Dalam membangun rute pengangkutan sampah dapat dilakukan dengan memodelkannya ke dalam kasus TSP dan salah satu metode yang dapat digunakan untuk menentukan rute optimal dari pemodelan tersebut yaitu dengan menggunakan algoritma genetika.

Algoritma genetika adalah salah satu cabang ilmu komputer pada bidang *artificial intelligence* untuk teknik optimasi. Algoritma ini mengadopsi mekanisme biologi yang dikemukakan oleh Charles Darwin dengan teori evolusinya yang terkenal dan kontroversial. Beberapa contoh penerapan algoritma

genetika yakni pada bidang industri untuk penentuan tata letak mesin. Selanjutnya algoritma genetika dapat diterapkan pada bidang peternakan untuk penentuan komposisi pakan ternak, dan pada bidang teknik sipil dapat diterapkan untuk perancangan jalur distribusi pipa. Dengan demikian algoritma genetika terlihat cukup mampu untuk menyelesaikan permasalahan optimasi secara *universal*.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang masalah dan identifikasi masalah yang sudah penulis kemukakan, penulis merumuskan permasalahan sebagai berikut: “Bagaimana mengimplementasikan algoritma genetika pada kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru ke dalam pemodelan kasus TSP”.

## 1.3. Tujuan dan Manfaat

### 1.3.1. Tujuan Penelitian

Adapun tujuan dari penelitian ini yakni untuk menguji kemampuan algoritma genetika dalam pencarian solusi rute mendekati optimal pada pemodelan kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru ke dalam kasus TSP.

### 1.3.2. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

- (i.) Dapat memberikan wawasan dan menambah pengalaman penulis dalam menerapkan ilmu yang didapat selama kuliah ke dalam praktek nyata.
- (ii.) Dapat memberikan solusi rute terbaik yang mendekati optimal pada kegiatan pengangkutan sampah yang dikelola instansi.
- (iii.) Dapat mengurangi waktu operasional dalam proses penyisiran dan pengangkutan sampah yang dilakukan petugas angkut jika jalur rute yang dibangun sudah optimal.

## 2. Landasan Teori

### 2.1. Dinas Kebersihan dan Pertamanan Kota Pekanbaru

Berdasarkan Peraturan Daerah Kota Pekanbaru Nomor 7 Tahun 2001 tentang Struktur Organisasi dan Tata Kerja dinas-dinas di lingkungan Pemerintah Kota Pekanbaru, maka dibentuklah Dinas Kebersihan dan Pertamanan Kota Pekanbaru pada Tahun 2001. Tugasnya adalah membantu Walikota Pekanbaru dalam melaksanakan kewenangan otonomi di bidang persampahan, pertamanan, penghijauan, lampu penerangan jalan umum dan lampu hias. Tugas utama

Dinas Kebersihan dan Pertamanan Kota Pekanbaru adalah mengelola persampahan yang ada di Kota Pekanbaru dimulai dari penyapuan, pengangkutan, pemusnahan hingga pengelolaan sampah.

### 2.2. Travelling Salesman Problem (TSP)

Pada tahun 1800, matematikawan Irlandia William Rowan Hamilton dan matematikawan Inggris Thomas Penyngton mengemukakan permasalahan matematika yang merupakan cikal bakal dari permasalahan *Travelling Salesman Problem* (TSP). TSP disajikan dalam bentuk permainan yang bernama *Icosian Hamilton* yang mengharuskan pemain untuk menyelesaikan permainan dengan menghubungkan 20 titik perjalanan hanya dengan menggunakan jalur-jalur tertentu. Selanjutnya hal ini dikenal dalam matematika diskrit yang disebut dengan teori Sirkuit Hamilton.

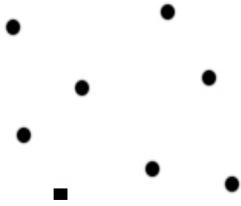
Munculnya bentuk umum dari TSP yaitu pada saat pertama kali dipelajari oleh seorang matematikawan Karl Menger di Vienna dan Harvard pada tahun 1930. Kemudian permasalahan ini dipublikasikan oleh Hassler Whitney dan Merrill Flood di Princeton. Pembicaraan detail tentang hubungan antara Menger dan Whitney, dan perkembangan dari TSP sebagai topik studi dapat dilihat di makalah Alexander Schrijver yang berjudul “*On the history of combinatorial optimization (till 1960)*” [13].

Menurut Manggolo *et al.* [8], persoalan pedagang keliling (*travelling salesman problem*) merupakan persoalan optimasi untuk mencari perjalanan terpendek bagi pedagang keliling yang ingin berkunjung ke beberapa kota. *Travelling salesman problem* dikenal sebagai salah satu masalah optimasi yang banyak menarik perhatian para ahli matematika dan khususnya ilmuwan komputer karena TSP mudah didefinisikan tetapi begitu sulit untuk diselesaikan [1].

Hal yang perlu diperhatikan di dalam kasus TSP adalah perjalanan *salesman* dimulai dari kota awal sampai seterusnya ke kota  $n$  dan akhirnya akan kembali lagi ke kota awal. Namun, aturannya adalah setiap kota selain kota awal hanya dapat dikunjungi tepat satu kali. Persoalan yang dihadapi ialah bagaimana membangun jalur rute yang optimal dengan mempertimbangkan aturan tersebut agar didapatkan total jarak tempuh minimal sehingga akan berdampak pada penghematan biaya transportasi.

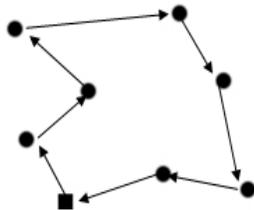
Persoalan yang dihadapi pada kasus ini tidak mudah, karena terdapat banyak kombinasi alur rute yang mungkin terjadi seiring dengan banyaknya kota yang akan dikunjungi dan juga harus memperhatikan aturan yang berlaku. Bentuk sederhana dari representasi TSP ke dalam bentuk graf dapat dilihat pada gambar 1. Berdasarkan gambar 1, terdapat tujuh titik kota yang harus dikunjungi *salesman* dan satu titik persegi yang didefinisikan sebagai rute awal dan rute akhir dalam membangun rute perjalanan. Dalam proses

pencarian rute, terlebih dahulu harus telah diketahui jarak-jarak yang menghubungkan tiap-tiap kota. Namun, jika jarak-jarak tidak diketahui, maka jarak dapat dihitung berdasarkan koordinat dari tiap-tiap titik



Gambar 1. Titik-titik kota yang akan dilewati

Setelah jarak yang menghubungkan tiap kota diketahui, maka rute terpendek dapat dicari dengan mencoba semua kombinasi dan menjumlahkan jarak dari kombinasi tersebut sehingga didapatkan rute seperti pada gambar 2.



Gambar 2. Rute terpendek yang didapatkan

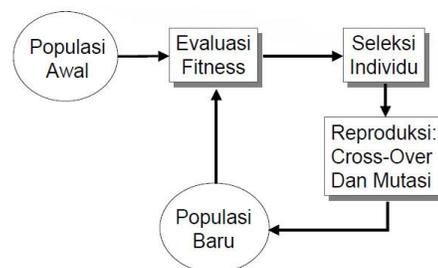
### 2.3. Algoritma Genetika

Menurut Carwoto [4], algoritma genetika adalah algoritma pencarian yang berdasarkan mekanisme seleksi alam Darwin dan prinsip-prinsip genetika untuk menentukan struktur-struktur (yang masing-masing disebut individu) berkualitas tinggi yang terdapat dalam sebuah domain (yang disebut populasi). Carwoto [4] juga menjelaskan bahwa dalam ilmu komputer, algoritma genetika termasuk dalam kajian komputasi lunak (*soft computing*) dan kecerdasan buatan (*artificial intelligence*). Algoritma Genetika dapat digunakan untuk menyelesaikan permasalahan *searching* dan optimasi yang mempunyai kompleksitas tinggi yang banyak terjadi dalam *dynamic programming* seperti TSP dan *Knapsack Problem* [2-3].

Gambaran tentang proses Algoritma Genetika [5-9] yaitu dimisalkan terdapat suatu populasi yang di dalamnya terdapat individu-individu yang merepresentasikan kumpulan solusi permasalahan. Individu-individu tersebut akan dievaluasi untuk mendapatkan nilai *fitness* dari tiap-tiap individu. Setelah dievaluasi, individu-individu ini akan menjalani proses seleksi. Hanya individu dengan nilai *fitness* tertinggi yang akan memiliki peluang lebih besar untuk terpilih ke proses berikutnya. Selanjutnya

akan dilakukan proses reproduksi untuk menambah keanekaragaman individu-individu yang telah terseleksi.

Proses reproduksi dikerjakan dengan melakukan perkawinan silang antara individu yang telah terpilih yang dinamakan proses *crossover* dengan harapan setelah mengalami perkawinan silang, akan lahir individu-individu baru yang lebih baik dari individu sebelumnya. Selanjutnya individu dari hasil *crossover* ini akan mengalami proses mutasi yang direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam kromosom. Pada akhirnya proses reproduksi ini akan menghasilkan populasi baru. Gambar 3 merupakan ilustrasi dari siklus algoritma genetika:



Gambar 3. Siklus Algoritma Genetika  
 Sumber: Basuki [2]

Setelah semua dilakukan, populasi baru ini akan kembali menjalani proses evaluasi nilai *fitness*, seleksi dan proses reproduksi sampai beberapa generasi yang telah ditentukan. Diharapkan setelah beberapa generasi akan didapatkan individu yang memiliki nilai *fitness* paling tinggi yang merupakan solusi dari permasalahan yang sedang dihadapi.

## 3. Analisa dan Desain Sistem

### 3.1. Analisa Sistem

#### 3.1.1. Metode TSP dan Algoritma Genetika

Proses pengangkutan sampah pada Dinas Kebersihan dan Pertamanan Kota Pekanbaru dilakukan dengan menyisir jalan utama sesuai dengan wilayah kerja yang telah ditetapkan untuk tiap-tiap petugas angkut dan membawa sampah ke tempat pembuangan akhir (TPA Muara Fajar). Pemodelan kasus TSP pada penelitian ini menguji tiga wilayah kerja pengangkutan sampah. Ini dilakukan karena terlalu banyaknya wilayah kerja yang ada, yakni berjumlah 27 wilayah kerja yang disusun oleh Bidang Kebersihan Kota. Karena keterbatasan space, hanya diilustrasikan untuk satu wilayah kerja (Wilayah kerja pertama). Pemodelan wilayah kerja pertama ke dalam kasus TSP. Uraian prosesnya adalah sebagai berikut:

**Tabel 1. Wilayah Kerja Pertama**  
 Sumber: Dinas Kebersihan dan Pertamanan Kota Pekanbaru 2014

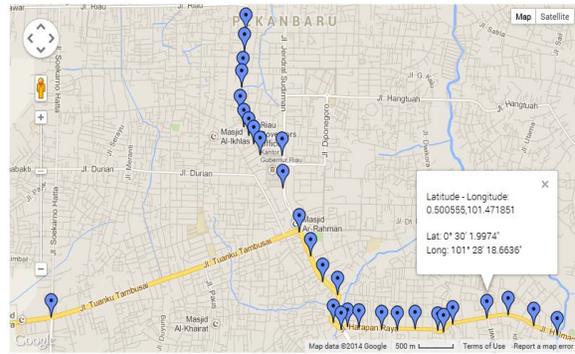
Jenis Angkutan	Wilayah Kerja	Jam Kerja
Angkutan Sampah / Mobil Sedang DKP 33 BM 9266 AP	Mulai dari Jembatan Sail Jl. Imam Munandar, memutar depan Hotel Asean, masuk Jl. Sudirman, Jl. T. Tambusai, memutar depan U-Turn ujung SKA, menuju Simpang Gramedia, masuk Jl. Sudirman sampai Pos Polisi Gedung Bank Riau Utama, masuk Jl. A. Yani kiri kanan sampai kantin Dinas PU Provinsi, TPA.	Pukul 12.00 s/d 18.00

Google Maps digunakan untuk menggambarkan wilayah kerja pengangkutan sampah yang sesuai dengan kondisi di lapangan. Berikut kondisi rute dilihat dari *google maps* berdasarkan wilayah kerja yang telah dipilih:



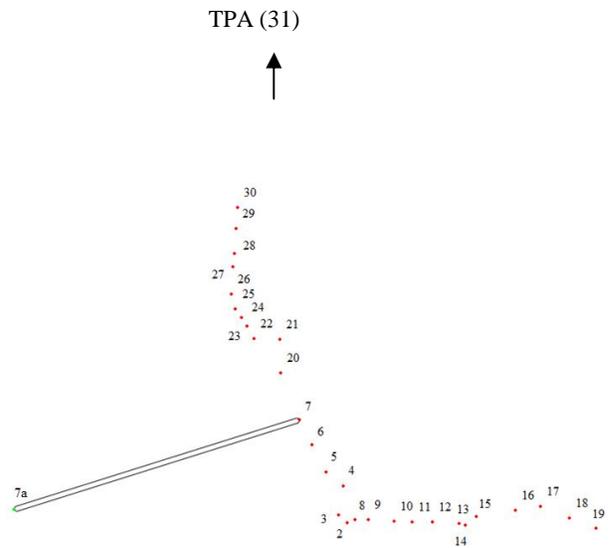
**Gambar 4. Kondisi rute wilayah kerja pertama**

Dalam memodelkan wilayah kerja tersebut ke dalam kasus TSP agar dapat diimplementasikan pada algoritma genetika, maka perlu diketahui lokasi tong sampah atau titik rawan terjadinya tumpukan sampah. Titik lokasi tong sampah atau titik tumpukan sampah akan ditambahkan. Selanjutnya akan dijadikan sebagai koordinat *decimal degrees*. Hal ini dapat dilihat pada gambar berikut:



**Gambar 5. Titik-titik Lokasi yang Ditentukan**

Berdasarkan titik-titik lokasi dan koordinat *decimal degrees* yang telah ditentukan, maka akan dilakukan *plotting* titik-titik lokasi. Dalam *plotting* ini setiap titik lokasi akan diberi inisial agar wilayah kerja tersebut nantinya dapat di-*encode* ke dalam kromosom pada algoritma genetika. Berikut gambar hasil plot dari wilayah kerja pertama:



**Gambar 6. Hasil plot wilayah kerja pertama**

Dalam *plotting*, titik lokasi pool dan TPA tidak dapat ditampilkan dikarenakan jauhnya jangkauan antara pool dan TPA dengan wilayah kerja. Titik lokasi terdiri dari 32 titik. Tetapi titik rute yang akan dituju pada saat pengujian hanya 31 titik dikarenakan titik 7a merupakan satu kesatuan dengan titik 7. Titik 7 dan titik 7a merupakan wilayah kerja pada Jalan Tuanku Tambusai. Penyisiran pada wilayah ini dilakukan pada kedua arah jalan sehingga jika rute perjalanan melewati titik 7, maka secara otomatis titik 7a juga akan dilewati dan akan kembali lagi ke titik 7 karena jalan tersebut merupakan jalan dua arah. Berdasarkan gambar plot

diatas, koordinat *decimal degrees* dari tiap-tiap titik lokasi dapat dilihat pada tabel 2 berikut:

**Tabel 2. Koordinat Titik Lokasi**

Lokasi	Koordinat	
	Latitude	Longitude
Titik 1 (Pool)	0.476521	101.469955
Titik 2	0.499265	101.454269
Titik 3	0.500083	101.453311
Titik 4	0.50313	101.453869
Titik 5	0.504589	101.452024
Titik 6	0.507422	101.450565
Titik 7	0.510082	101.449234
Titik 8	0.499611	101.455071
Titik 9	0.499568	101.456444
Titik 10	0.499397	101.459191
Titik 11	0.499311	101.461079
Titik 12	0.499354	101.463182
Titik 13	0.499182	101.465929
Titik 14	0.499053	101.466615
Titik 15	0.499912	101.467774
Titik 16	0.500555	101.471851
Titik 17	0.500942	101.474469
Titik 18	0.49974	101.477559
Titik 19	0.49871	101.480348
Titik 20	0.515019	101.44725
Titik 21	0.518559	101.447164
Titik 22	0.518645	101.44446
Titik 23	0.519911	101.443709
Titik 24	0.520812	101.44313
Titik 25	0.521735	101.44255
Titik 26	0.52328	101.442121
Titik 27	0.526133	101.44225
Titik 28	0.527528	101.4424
Titik 29	0.530146	101.442572
Titik 30	0.532356	101.442744
Titik 31 (TPA Muara Fajar)	0.643376	101.438935

### 3.1.2. Penyelesaian Masalah

Dari beberapa sampel titik lokasi yang akan dikunjungi, disajikan dalam bentuk koordinat *decimal degrees* yang dapat dilihat pada tabel 3.

**Tabel 3. Sampel Koordinat Titik Lokasi**

Lokasi	Latitude	Longitude
1 (Pool)	0.476521	101.469955
2	0.499265	101.454269
3	0.500083	101.453311
4	0.50313	101.453869
5	0.504589	101.452024
6 (TPA)	0.507422	101.450565

Jarak antara dua titik pada koordinat *decimal degrees* dihitung menggunakan rumus *Haversine*. Sintak program untuk penghitungan jarak menggunakan rumus *haversine* adalah sebagai berikut:

```
function Haversine(double lat1, double
lat2, double long1, double long2)
{
    double R = 6371; //Nilai radian
    bumi
    double dLat = toRad(lat2 - lat1);
    double dLong = toRad(long2 - long1);
    double a = Math.Pow(Math.Sin(dLat /
2), 2)
                + Math.Pow(Math.Sin(dLong
/ 2), 2)
                * Math.Cos(toRad(lat1))
                * Math.Cos(toRad(lat2));
    double c = 2 *
Math.Atan2(Math.Sqrt(a), Math.Sqrt(1 -
a));
    double d = R * c;
    return d;
}

function toRad(double nilai)
{
    return nilai * Math.PI / 180;
}
```

Setelah dilakukan penghitungan jarak antar titik-titik koordinat lokasi, maka akan menghasilkan jarak antar lokasi yang disajikan dalam tabel matriks jarak (Lihat tabel 4).

**Tabel 4. Matriks Jarak**

0	1	2	3	4	5	6
1	0	3,0721	3,2076	3,4573	3,7034	4,0564
2	3,0721	0	0,14	0,4320	0,6424	0,9961
3	3,2076	0,1400	0	0,3444	0,5210	0,8713
4	3,4573	0,4320	0,3444	0	0,2615	0,6022
5	3,7034	0,6424	0,5210	0,2615	0	0,3543
6	4,0564	0,9961	0,8713	0,6022	0,3543	0

Selanjutnya dilakukan proses optimasi menggunakan algoritma genetika. Proses algoritma genetika yang diterapkan terdiri dari 6 tahapan [12] yakni sebagai berikut:

- Pembangkitan populasi awal, ukuran populasi pada contoh kasus ini ditetapkan berjumlah 10.
- Evaluasi *fitness*, melakukan *inverse* dari total jarak setiap kromosom dengan penghitungan  $\frac{1}{\text{nilai jarak kromosom}}$
- Seleksi, proses seleksi dilakukan dengan menggunakan metode *roulette wheel selection*.
- Crossover*, proses *crossover* dilakukan dengan menggunakan metode *cycle crossover*.
- Mutasi, proses mutasi dilakukan dengan menggunakan metode *inversion mutation*.
- Elitism*, dikarenakan proses seleksi dilakukan secara acak, maka pada penelitian ini akan ditambahkan metode *elitism* untuk menyimpan salah satu kromosom terbaik pada setiap generasi untuk di salin ke populasi selanjutnya.

### 3.1.3. Proses Algoritma Genetika

#### A. Pembangkitan Populasi Awal

Pembangkitan populasi dilakukan dengan membangkitkan bilangan acak antara 1 sampai dengan banyaknya titik lokasi yakni berjumlah 6 untuk mengisi nilai-nilai gen pada setiap kromosom dan dilakukan sebanyak jumlah populasi yang ditentukan yakni 10 kali. Dimisalkan populasi awal yang terbentuk adalah sebagai berikut (tabel 5):

**Tabel 5. Populasi Awal**

Kromosom	Susunan Gen
1	1 4 3 5 2 6
2	1 2 5 3 4 6
3	1 3 2 5 4 6
4	1 2 3 4 5 6
5	1 3 2 4 5 6
6	1 2 4 3 5 6
7	1 4 2 5 3 6
8	1 5 4 3 2 6
9	1 4 3 5 2 6
10	1 2 5 4 3 6

Dikarenakan perjalanan dimulai dari pool, dan rute akan selalu berakhir di TPA sebelum kembali ke pool, maka setiap gen pertama dan terakhir pada kromosom akan bernilai 1 dan 6 berdasarkan inisial dari pool dan TPA pada sampel data.

#### B. Evaluasi *Fitness*

Selanjutnya setiap kromosom menjalani proses evaluasi untuk mendapatkan nilai *fitness* dari masing-masing kromosom. Hitung jarak rute kromosom berdasarkan tabel matriks jarak, sehingga:

$$\begin{aligned} \text{Kromosom 1} &= (1 \rightarrow 4) + (4 \rightarrow 3) + (3 \rightarrow 5) + (5 \rightarrow 2) \\ &\quad + (2 \rightarrow 6) + (6 \rightarrow 1) \\ &= 3,4573 + 0,3444 + 0,5210 \\ &\quad + 0,6424 \\ &\quad + 0,9961 \\ &\quad + 4,0564 \\ &= 10,0176 \end{aligned}$$

$$\begin{aligned} \text{Fitness Kromosom 1} &= \frac{1}{10,0176} \\ &= 0,0998243 \end{aligned}$$

Lakukan langkah penghitungan yang sama untuk mendapatkan nilai *fitness* kromosom lainnya. Sehingga *fitness* seluruh kromosom adalah sebagai berikut:

**Tabel 6. Nilai *Fitness* Kromosom**

Kromosom	Total Jarak	<i>Fitness</i>
1	10,0176	0,0998243
2	9,2385	0,1082426

3	8,9101	0,1122321
<b>4</b>	<b>8,2287</b>	<b>0,1215258</b>
5	8,4518	0,1183179
6	8,7802	0,1138926
7	9,9804	0,1001963
8	9,5018	0,1052432
9	10,0176	0,0998243
10	9,2481	0,1081303
<b>Total <i>Fitness</i></b>		<b>1,0874294</b>

Berdasarkan pada tabel 6, terlihat bahwa kromosom ke-4 memiliki nilai *fitness* paling tinggi pada generasi saat ini yaitu 0,1215258 dengan susunan kromosom = [1 2 3 4 5 6]. Jarak rute yang didapatkan pada kromosom ini adalah 8,2287. Selanjutnya kromosom ini disimpan sebagai kromosom elit yang akan disalin secara langsung ke populasi selanjutnya pada proses *elitism*.

#### C. Seleksi

Total *fitness* seluruh kromosom pada populasi yakni berjumlah 1,0874294. Lalu hitung nilai *fitness* relatif setiap kromosom. Proses penghitungannya adalah sebagai berikut:

$$\begin{aligned} P_1 &= 0,0998243/1,0874294 = 0,091798 \\ &\approx 0,09180 \end{aligned}$$

$$\begin{aligned} P_2 &= 0,1082426/1,0874294 = 0,099539 \\ &\approx 0,09954 \end{aligned}$$

$$\begin{aligned} P_3 &= 0,1122321/1,0874294 = 0,103208 \\ &\approx 0,10321 \end{aligned}$$

$$\begin{aligned} P_4 &= 0,1215258/1,0874294 = 0,111756 \\ &\approx 0,11176 \end{aligned}$$

$$\begin{aligned} P_5 &= 0,1183179/1,0874294 = 0,108805 \\ &\approx 0,1088 \end{aligned}$$

$$\begin{aligned} P_6 &= 0,1138926/1,0874294 = 0,104735 \\ &\approx 0,10473 \end{aligned}$$

$$\begin{aligned} P_7 &= 0,1001963/1,0874294 = 0,0921405 \\ &\approx 0,09214 \end{aligned}$$

$$\begin{aligned} P_8 &= 0,1052432/1,0874294 = 0,096781 \\ &\approx 0,09678 \end{aligned}$$

$$\begin{aligned} P_9 &= 0,0998243/1,0874294 = 0,091798 \\ &\approx 0,09180 \end{aligned}$$

$$\begin{aligned} P_{10} &= 0,1081303/1,0874294 = 0,099436 \\ &\approx 0,09944 \end{aligned}$$

Selanjutnya hitung *fitness* kumulatif dari tiap-tiap *fitness* relatif yang telah didapatkan:

$$C_0 = 0$$

$$C_1 = 0 + 0,09180 = 0,09180$$

$$C_2 = 0,09180 + 0,09954 = 0,19134$$

$$C_3 = 0,19134 + 0,10321 = 0,29455$$

$$C_4 = 0,29455 + 0,11176 = 0,40631$$

$$C_5 = 0,40631 + 0,1088 = 0,51511$$

$$C_6 = 0,51511 + 0,10473 = 0,61984$$

$$C_7 = 0,61984 + 0,09214 = 0,71198$$

$$C_8 = 0,71198 + 0,09678 = 0,80876$$

$$C_9 = 0,80876 + 0,09180 = 0,90056$$

$$C_{10} = 0,90056 + 0,09944 = 1$$

Dari hasil perhitungan *fitness* relatif dan kumulatif, selanjutnya proses seleksi kromosom dilakukan berdasarkan proporsi dari *fitness* kumulatifnya yang disajikan pada tabel 7 berikut:

**Tabel 7. Nilai *Fitness* Relatif dan Kumulatif Setiap Kromosom**

Kromosom	<i>Fitness</i> Relatif	<i>Fitness</i> Kumulatif
1	0,09180	0 – 0,09180
2	0,09954	> 0,09180 – 0,19134
3	0,10321	> 0,19134 – 0,29455
4	0,11176	> 0,29455 – 0,40631
5	0,1088	> 0,40631 – 0,51511
6	0,10473	> 0,51511 – 0,61984
7	0,09214	> 0,61984 – 0,71198
8	0,09678	> 0,71198 – 0,80876
9	0,09180	> 0,80876 – 0,90056
10	0,09944	> 0,90056 – 1

Bangkitkan bilangan acak antara 0 sampai 1 sebanyak jumlah populasi. Misal bilangan acak yang telah dibangkitkan sebagai berikut:

$$R_1 = 0,583$$

$$R_2 = 0,836$$

$$R_3 = 0,236$$

$$R_4 = 0,679$$

$$R_5 = 0,346$$

$$R_6 = 0,385$$

$$R_7 = 0,794$$

$$R_8 = 0,458$$

$$R_9 = 0,245$$

$$R_{10} = 0,632$$

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

**Tabel 8. Kromosom Terpilih sebagai Induk**

Induk	Ri	Kromosom Terpilih	Susunan Gen
1	0,583	Kromosom 6	1 2 4 3 5 6
2	0,836	Kromosom 9	1 4 3 5 2 6
3	0,236	Kromosom 3	1 3 2 5 4 6
4	0,679	Kromosom 7	1 4 2 5 3 6
5	0,346	Kromosom 4	1 2 3 4 5 6
6	0,385	Kromosom 4	1 2 3 4 5 6
7	0,794	Kromosom 8	1 5 4 3 2 6
8	0,458	Kromosom 5	1 3 2 4 5 6
9	0,245	Kromosom 3	1 3 2 5 4 6
10	0,632	Kromosom 7	1 4 2 5 3 6

#### D. Crossover

Pasangkan kromosom induk secara acak sebanyak setengah dari ukuran populasi dan tentukan nilai probabilitas *crossover* ( $P_c$ ). Nilai  $P_c$  yang ditentukan adalah 0,5. Misal susunan pasangan induk adalah sebagai berikut:

Pasangan 1	Induk 4 Induk 7
Pasangan 2	Induk 6 Induk 1
Pasangan 3	Induk 2 Induk 5
Pasangan 4	Induk 9 Induk 3
Pasangan 5	Induk 10 Induk 8

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

$$R_1 = 0,743$$

$$R_2 = 0,536$$

$$R_3 = 0,945$$

$$R_4 = 0,834$$

$$R_5 = 0,345$$

Dari bilangan acak yang telah dibangkitkan, terdapat satu nilai yang kurang dari nilai  $P_c$  yakni pada  $R_5$ . Artinya pasangan ke-5 akan mengalami *crossover* yaitu induk 10 dan induk 8, dan sisanya akan masuk ke tahap berikutnya tanpa mengalami *crossover*. Berikut ini merupakan proses *cycle crossover* pada pasangan induk yang telah terpilih:

- Tentukan siklus dari kedua induk. Dikarenakan setiap gen pertama bernilai 1, maka penentuan siklus dimulai dari gen kedua.

$$\text{Induk 10} = [1 \ 4 \ 2 \ 5 \ 3 \ 6]$$



$$\text{Induk 8} = [1 \ 3 \ 2 \ 4 \ 5 \ 6]$$

$$\text{Siklus} = 4 - 3 - 5 - 4$$

Sehingga gen yang termasuk dalam siklus akan disalin ke dalam *offspring*.

$$\text{Offspring 10} = [x \ 4 \ x \ 5 \ 3 \ x]$$

$$\text{Offspring 8} = [x \ 3 \ x \ 4 \ 5 \ x]$$

- Isi nilai gen yang kosong pada offspring 10 dengan mengambil sisa gen dari induk 8 dan begitu juga sebaliknya sehingga:

$$\text{Offspring 10} = [1 \ 4 \ 2 \ 5 \ 3 \ 6]$$

$$\text{Offspring 8} = [1 \ 3 \ 2 \ 4 \ 5 \ 6]$$

Proses *crossover* tidak mengalami perubahan dikarenakan jumlah gen yang terlalu sedikit, sehingga hal ini mudah terjadi. Beda halnya jika gen kromosom berjumlah puluhan, maka kromosom akan mengalami rekombinasi. Tetapi tidak menutup kemungkinan hal tersebut juga terjadi, karena penentuan penyalinan gen dilakukan berdasarkan siklus. Walaupun demikian,

*offspring* akan tetap menggantikan posisi dari induknya.

Induk 1	1 2 4 3 5 6
Induk 2	1 4 3 5 2 6
Induk 3	1 3 2 5 4 6
Induk 4	1 4 2 5 3 6
Induk 5	1 2 3 4 5 6
Induk 6	1 2 3 4 5 6
Induk 7	1 5 4 3 2 6
<b>Offspring 8</b>	<b>1 3 2 4 5 6</b>
Induk 9	1 3 2 5 4 6
<b>Offspring 10</b>	<b>1 4 2 5 3 6</b>

### E. Mutasi

Selanjutnya kromosom akan menjalani proses mutasi. Nilai probabilitas mutasi ( $P_m$ ) ditentukan sebesar 0,1. Lalu bangkitkan bilangan acak antara 0 sampai 1 sebanyak jumlah populasi. Misalkan bilangan acak yang telah dibangkitkan sebagai berikut:

$$\begin{aligned} R_1 &= 0,124 \\ R_2 &= 0,835 \\ R_3 &= 0,964 \\ R_4 &= 0,235 \\ R_5 &= 0,459 \\ R_6 &= 0,345 \\ R_7 &= 0,347 \\ R_8 &= 0,235 \\ R_9 &= 0,056 \\ R_{10} &= 0,633 \end{aligned}$$

Berdasarkan bilangan acak yang telah dibangkitkan, terlihat hanya  $R_9$  yang memiliki nilai lebih kecil dari  $P_m$ , sehingga hanya kromosom ke-9 yang terpilih untuk menjalani mutasi. Proses mutasi dilakukan dengan menggunakan metode *inversion mutation*. Dikarenakan pada kasus ini setiap gen awal dan akhir pada kromosom tidak boleh berubah, maka prosedur sedikit dimodifikasi sehingga langkah-langkahnya sebagai berikut:

- Bangkitkan bilangan bulat acak antara 2 sampai  $n-1$  sebagai titik potong  $T_1$  dan  $T_2$ , dimana  $n$  merupakan jumlah gen yakni berjumlah 6. misalkan  $T_1=3$  dan  $T_2=5$ , sehingga:

$$\text{Kromosom} = [1\ 3\ |2\ 5\ 4\ |6]$$

- Selanjutnya balikkan posisi nilai gen yang telah ditentukan dengan titik potong, sehingga:

$$\text{Kromosom} = [1\ 3\ |4\ 5\ 2\ |6]$$

Setelah mutasi dilakukan, kromosom tersebut akan dikelompokkan pada kromosom-kromosom yang tidak menjalani mutasi sehingga didapatkan populasi baru. Populasi baru hasil dari mutasi tersebut adalah sebagai berikut:

Kromosom 1	1 2 4 3 5 6
Kromosom 2	1 4 3 5 2 6
Kromosom 3	1 3 2 5 4 6
Kromosom 4	1 4 2 5 3 6
Kromosom 5	1 2 3 4 5 6
Kromosom 6	1 2 3 4 5 6
Kromosom 7	1 5 4 3 2 6
Kromosom 8	1 3 2 4 5 6
<b>Kromosom 9</b>	<b>1 3 4 5 2 6</b>
Kromosom 10	1 4 2 5 3 6

Sebelum kembali pada tahap evaluasi *fitness*, kromosom akan menjalani proses *elitism* terlebih dahulu.

### F. Elitism

Proses *elitism* ditambahkan karena terdapat kemungkinan bahwa kromosom terbaik pada generasi sebelumnya tidak termasuk pada populasi baru atau nilai *fitness* kromosom terbaik tersebut telah mengalami penurunan dikarenakan proses *crossover* dan mutasi. *Elitism* dilakukan dengan menyalin kromosom terbaik yang telah disimpan pada tahap evaluasi yang sebelumnya untuk menggantikan kromosom yang memiliki nilai *fitness* paling rendah di dalam populasi baru. Berikut ini merupakan nilai *fitness* tiap-tiap kromosom pada populasi baru:

**Tabel 9. Nilai *Fitness* Kromosom Populasi Baru**

Kromosom	<i>Fitness</i>
1	0,1138926
<b>2</b>	<b>0,0998243</b>
3	0,1122321
4	0,1001963
5	0,1215258
6	0,1215258
7	0,1052432
8	0,1183179
9	0,1051701
10	0,1001963

Berdasarkan nilai *fitness* kromosom populasi baru pada tabel 3.13, dapat dilihat bahwa kromosom ke-2 memiliki nilai *fitness* paling rendah dan akan diganti dengan kromosom terbaik yang telah disimpan pada tahap evaluasi yang sebelumnya. Sehingga populasi baru yang dihasilkan adalah sebagai berikut:

Kromosom 1	1 2 4 3 5 6
<b>Kromosom 2</b>	<b>1 2 3 4 5 6</b>
Kromosom 3	1 3 2 5 4 6
Kromosom 4	1 4 2 5 3 6
Kromosom 5	1 2 3 4 5 6
Kromosom 6	1 2 3 4 5 6

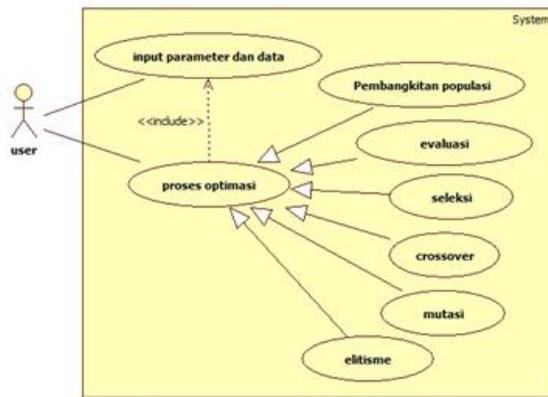
Kromosom 7	1 5 4 3 2 6
Kromosom 8	1 3 2 4 5 6
Kromosom 9	1 3 4 5 2 6
Kromosom 10	1 4 2 5 3 6

Selanjutnya proses optimasi akan kembali menjalani evaluasi *fitness*, seleksi, reproduksi dan *elitism* sampai maksimum generasi yang telah ditentukan.

### 3.2. Desain Sistem

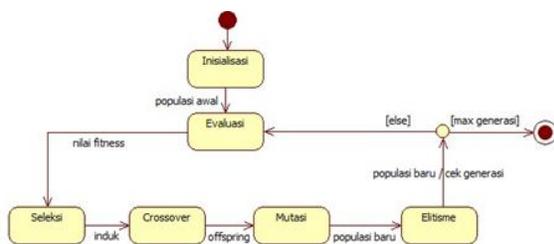
#### 3.2.1. Desain Global

Pada aplikasi yang akan dibuat terdapat beberapa kegiatan yang dapat dilakukan oleh *user*. Berikut ini gambaran diagram *use case* dari aplikasi yang akan dibangun:



Gambar 7. Use case Diagram

Berikut *state machine diagram* yang menggambarkan perubahan status di dalam aplikasi ketika user melakukan proses optimasi.

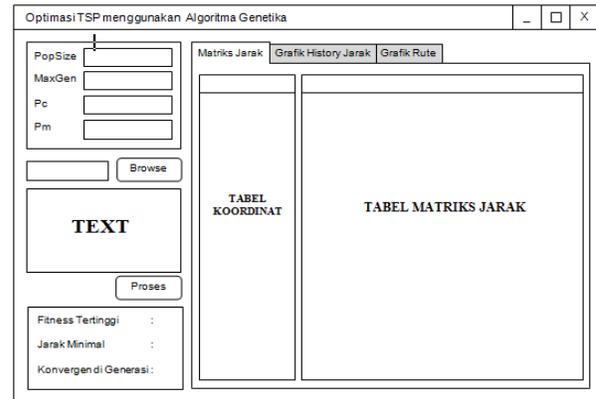


Gambar 8. State Machine Diagram

#### 3.2.2. Desain Interface

Desain *Interface* [11-12] berguna sebagai alat untuk *user* agar dapat berinteraksi dengan program. Suatu *interface* dari aplikasi dirancang sebagai gambaran struktur aplikasi yang akan dibuat untuk memudahkan user dalam menjalankan program. Berikut gambaran

*interface* dari aplikasi yang akan dibangun pada penelitian ini:



Gambar 9. Desain Interface

## 4. Implementasi dan Pengujian Sistem

### 4.1. Implementasi Sistem

Implementasi dilakukan untuk menerapkan hasil analisa dan perancangan aplikasi yang telah dibahas sebelumnya. Terdapat dua tahapan pada implementasi ini, yaitu tahapan membuat file *.tsp* untuk data pengujian dan tahap implementasi aplikasi.

#### 4.2.1 Data Pengujian

Data pengujian yang akan di-*input* pada aplikasi merupakan sebuah file yang berformat *\*.tsp*. Sehingga data wilayah kerja yang telah diolah pada tahap analisa yang berupa koordinat *decimal degrees* akan dijadikan file berformat *.tsp* dengan *EDGE\_WEIGHT\_TYPE: GEO*. Pembuatan file dilakukan dengan menggunakan aplikasi notepad. Berikut contoh isi dari file *.tsp* tersebut:

```

NAME: dkp16
TYPE: TSP
COMMENT: 16 titik lokasi pengangkutan sampah
(Medrio)
DIMENSION: 16
EDGE_WEIGHT_TYPE: GEO
DISPLAY_DATA_TYPE: COORD_DISPLAY
NODE_COORD_SECTION
1 0.476521 101.469955
2 0.499265 101.454269
3 0.500083 101.453311
4 0.50313 101.453869
5 0.504589 101.452024
6 0.507422 101.450565
7 0.510082 101.449234
8 0.499611 101.455071
9 0.499568 101.456444
10 0.499397 101.459191
11 0.499311 101.461079
12 0.499354 101.463182
    
```

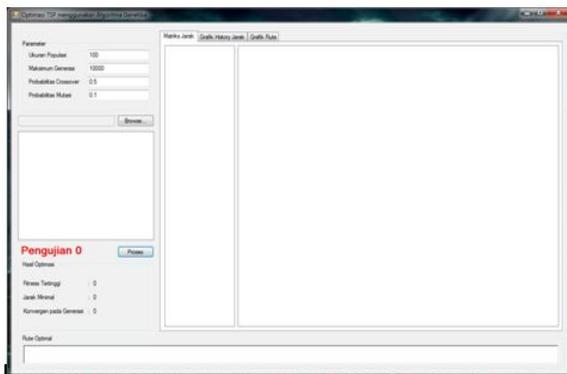
13 0.499182 101.465929  
 14 0.499053 101.466615  
 15 0.499912 101.467774  
 16 0.500555 101.471851  
 EOF

Setelah data koordinat dari ketiga wilayah kerja tersebut dijadikan file .tsp, selanjutnya pengujian dilakukan dengan meng-*input* file tersebut ke dalam aplikasi. Berikut ini merupakan nama file dari ketiga wilayah kerja yang diujikan:

1. Wilayah Kerja Pertama: Dkp31.tsp
2. Wilayah Kerja Kedua: Dkp35.tsp
3. Wilayah Kerja Ketiga: Dkp39.tsp

#### 4.2.2 Implementasi Aplikasi

Aplikasi dibangun berdasarkan hasil analisa dan perancangan yang telah dibahas sebelumnya dan dibangun menggunakan bahasa pemrograman VB.NET 2008. Berikut ini merupakan tampilan dari aplikasi:



Gambar 10. Tampilan Aplikasi

Sebelum proses optimasi dijalankan, terdapat beberapa hal yang harus diperhatikan yakni sebagai berikut:

##### 1. Pengisian Parameter

Terdapat empat parameter algoritma genetika yang harus ditentukan sebelum proses optimasi dijalankan, yakni ukuran populasi, maksimum generasi, probabilitas *crossover* dan probabilitas mutasi. Pada aplikasi telah ditentukan batas-batas nilai parameter yang dapat di *input* ke dalam aplikasi sebagai berikut:

- a. Batas nilai untuk ukuran populasi berada antara 2 sampai 1000 dan merupakan bilangan genap.
- b. Batas minimal untuk nilai maksimum generasi adalah 1 dan batas maksimal tidak ditentukan.
- c. Nilai Probabilitas *crossover* berada antara 0 sampai 1 dan merupakan bilangan desimal.
- d. Nilai Probabilitas mutasi berada antara 0 sampai 1 dan merupakan bilangan desimal.

##### 2. Input Data Pengujian

Setelah nilai parameter ditentukan, selanjutnya *input*-kan data pengujian dengan meng-klik tombol "Browse" lalu cari lokasi file yang akan diuji. Data pengujian yang dapat di-*input* ke dalam aplikasi hanya file-file .tsp yang bertipe "GEO".

Selanjutnya proses optimasi dilakukan dengan meng-klik tombol "Proses". Setelah optimasi selesai, aplikasi akan menampilkan informasi berupa nilai *fitness* tertinggi, nilai jarak minimal, grafik *history* jarak, dan rute optimal yang telah didapatkan.

#### 4.3. Pengujian Data

Pengujian dilakukan dengan menggunakan nilai parameter tetap yakni ukuran populasi = 100, maksimum generasi = 10000, probabilitas *crossover* = 0,5 dan probabilitas mutasi = 0,1. Masing-masing data akan diuji sebanyak 10 kali. Parameter untuk menentukan solusi terbaik dilihat berdasarkan nilai *fitness* yang diperoleh dari masing-masing hasil pengujian.

##### 4.3.1. Pengujian Wilayah Kerja Pertama

Data pengujian wilayah kerja pertama adalah dkp31.tsp. Dari pengujian yang telah dilakukan sebanyak 10 kali, diperoleh hasil pengujian yang dapat dilihat pada tabel berikut:

Tabel 9. Hasil Pengujian Wilayah Kerja Pertama

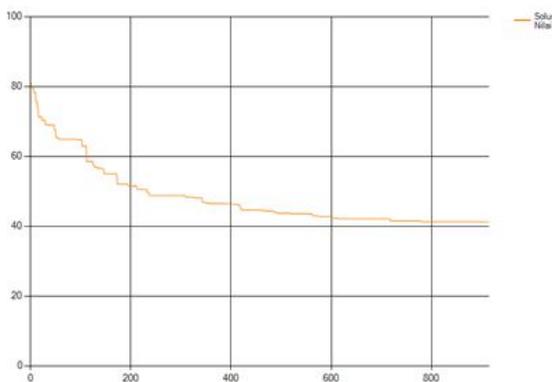
Pengujian ke	<i>Fitness</i> Tertinggi	Jarak Minimal	Generasi ke
1	0,0242390	41,2558	1460
2	0,0242390	41,2558	1309
3	0,0242390	41,2558	1048
4	0,0242390	41,2558	1249
5	0,0242390	41,2558	1529
6	0,0242390	41,2558	2014
7	0,0242390	41,2558	3017
8	0,0242390	41,2558	963
9	<b>0,0242390</b>	<b>41,2558</b>	<b>914</b>
10	0,0242390	41,2558	4853

Seluruh hasil pengujian yang diperoleh cenderung sama, yakni nilai *fitness* yang didapatkan = 0,0242390 dengan jarak minimal = 41,2558. Sehingga hasil pengujian terbaik dilihat dari perolehan solusi yang tercepat. Perolehan *fitness* tertinggi yang tercepat didapatkan pada pengujian ke-9, dimana solusi ditemukan pada generasi ke-914. Susunan rute optimal yang didapatkan adalah 1 - 19 - 18 - 17 - 16 - 15 - 14 - 13 - 12 - 11 - 10 - 9 - 8 - 2 - 3 - 4 - 5 - 6 - 7 - 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 1.

Berikut ini merupakan gambar hasil dari pengujian ke-9 untuk data dkp31.tsp pada aplikasi:

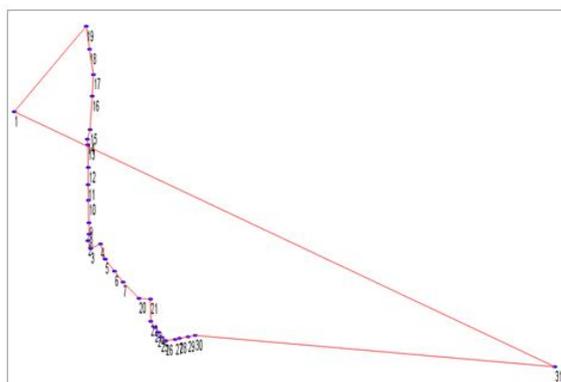
Gambar 10. Hasil Penguji ke-9

Gambar 11 merupakan grafik dari nilai jarak yang pernah didapatkan pada penguji ke-9 dari awal sampai proses optimasi berakhir:



Gambar 11. Grafik History Jarak Penguji ke-9

Pada generasi pertama, nilai jarak yang didapatkan berada pada kisaran 80. Kemudian terus mengalami penurunan sampai akhirnya didapatkan nilai jarak 41,2558 di generasi ke-914 dan sampai pada generasi ke-10000 tidak ditemukan kembali nilai yang lebih kecil dari nilai jarak tersebut. Grafik rute yang dihasilkan pada penguji dapat dilihat pada gambar 12 berikut:



Gambar 12. Grafik Rute Optimal

Perjalanan dimulai dari rute 1 (Pool) kemudian mengunjungi seluruh rute pada wilayah kerja dan berakhir pada rute 31 (TPA). Selanjutnya dari rute 31 akan kembali lagi pada rute 1.

#### 4.4. Hasil Penguji

Berdasarkan pada penguji yang telah dilakukan, dapat ditarik beberapa kesimpulan dari hasil penguji sebagai berikut:

1. Algoritma genetika mampu memberikan solusi rute optimal pada pemodelan kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru ke dalam kasus TSP.
2. Dari hasil penguji yang dilakukan sebanyak 10 kali untuk setiap data uji, mayoritas memberikan solusi kromosom dengan bobot *fitness* yang sama. Ini menandakan bahwa tidak ada lagi solusi yang ditemukan selain dari solusi yang telah didapatkan. Sehingga dapat dikatakan bahwa hasil dari proses optimasi yang didapat merupakan solusi yang paling optimal. Perolehan kromosom dengan *fitness* tertinggi pada masing-masing data uji paling cepat didapatkan pada generasi berikut:
  - a. Untuk data dkp31.tsp didapatkan pada generasi ke-914
  - b. Untuk data dkp35.tsp didapatkan pada generasi ke-937
  - c. Untuk data dkp39.tsp didapatkan pada generasi ke-1432

Algoritma genetika menghasilkan solusi pada generasi yang tidak menentu. Hasil bisa saja didapatkan pada rentang generasi maksimum atau bahkan bisa saja maksimum generasi yang ditentukan tidak mampu mengeksplorasi seluruh ruang solusi permasalahan. Sehingga nilai maksimum generasi perlu ditambahkan.

### 5. Penutup

#### 5.1. Kesimpulan

Setelah dilakukan implementasi algoritma genetika pada kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru ke dalam pemodelan kasus TSP, maka dapat diambil kesimpulan sebagai berikut:

- a. Proses pembangunan rute dilakukan dengan memodelkannya ke dalam kasus TSP. Sehingga jalur rute pengangkutan sampah langsung tertuju pada posisi titik lokasi tumpukan sampah yang tersebar.
- b. Berdasarkan pada hasil dari penguji yang dilakukan, dapat dipastikan bahwa rute pengangkutan sampah yang telah diperoleh merupakan solusi rute yang optimal.

- c. Penerapan algoritma genetika cukup membantu dalam mencari solusi rute optimal pada kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru.

## 5.2. Saran

Adapun saran untuk penelitian selanjutnya tentang penerapan algoritma genetika pada pemodelan kasus TSP adalah sebagai berikut:

- a. Penelitian yang dilakukan hanya sebatas mencari rute terpendek. Akan lebih baik apabila dalam membangun rute juga memperhatikan kapasitas angkut kendaraan, estimasi bahan bakar kendaraan, dan waktu operasional yang diperbolehkan.
- b. Penginputan data uji masih dilakukan secara manual, yakni dengan membuat file .tsp terlebih dahulu. Diharapkan untuk pengembangan aplikasi selanjutnya, proses *input* data dapat dilakukan secara langsung, yakni di dalam aplikasi sudah terdapat peta untuk menentukan koordinat titik-titik lokasi.
- c. Diharapkan setelah ini peneliti selanjutnya dapat menerapkan algoritma genetika dalam membangun suatu sistem yang digunakan untuk pengelolaan rute kendaraan atau untuk kasus yang identik.

## 6.Referensi

- [1] Amri, Faisal., Nababan, Erna Budhiarti., dan Syahputra, Mohammad Fadly. 2012. Artificial Bee Colony Algorithm untuk Menyelesaikan Travelling Salesman Problem. *Jurnal Dunia Teknologi Informasi*, Vol. 1, No. 1, pp. 8-13.
- [2] Basuki, Achmad. 2003a. Algoritma Genetika: Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi, dan Machine Learning. [http://budi.blog.undip.ac.id/files/2009/06/algoritma\\_genetika.pdf](http://budi.blog.undip.ac.id/files/2009/06/algoritma_genetika.pdf). Diakses Juli 2014.
- [3] Basuki, Achmad. 2003b. Strategi Menggunakan Algoritma Genetika. <https://lecturer.eepis-its.edu/~basuki/lecture/StrategiAlgoritmaGenetika.pdf>. Diakses Mei 2014.
- [4] Carwoto. 2007. Implementasi Algoritma Genetika untuk Optimasi Penempatan Kapasitor Shunt pada Penyulang Distribusi Tenaga Listrik. *Jurnal Teknologi Informasi DINAMIK*, Vol. 12, No. 2, pp. 122-130.
- [5] Deep, Kusum dan Mebrahtu, Hadush. 2011. Combined Mutation Operators of Genetic Algorithm for the Travelling Salesman Problem. *International Journal of Combinatorial Optimization Problems and Informatics*, Vol. 2, No. 3, pp. 1-23.
- [6] Dewi, Kania Evita. 2012. Perbandingan Metode Newton-Raphson dan Algoritma Genetik pada Penentuan *Implied Volatility* Saham. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, Vol. 1, No. 2, pp. 9-16.
- [7] Fox, Roland dan Tseng, Steven. 1997. Traveling Salesman Problem by Genetic Algorithm and Simulated Annealing. <http://www.cs.nthu.edu.tw/~jang/courses/cs5611/project/2/>. Diakses Mei 2014.
- [8] Haviluddin. 2011. Memahami Penggunaan UML (*Unified Modelling Language*). *Jurnal Informatika Mulawarman*, Vol. 6, No. 1, pp. 1-15.
- [9] Manggolo, Inu., Marzuki, Marza Ihsan., dan Alaydrus, Mudrik. 2011. Optimalisasi Perencanaan Jaringan Akses Serat Optik Fiber To The Home Menggunakan Algoritma Genetika. *InComTech, Jurnal Telekomunikasi dan Komputer*, Vol. 2, No. 2, pp. 21-36.
- [10] Rosa, A.S. dan Shalahuddin, M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- [11] Suarga. 2012. *Algoritma dan Pemrograman*. Yogyakarta: Penerbit ANDI.
- [12] Yusuf, Akhmad dan Soesanto, Oni. 2012. Algoritma Genetika pada Penyelesaian Akar Persamaan Sebuah Fungsi. *Jurnal Matematika Murni dan Terapan*, Vol. 6, No. 2, pp. 47-56.
- [13] <http://www.math.uwaterloo.ca/tsp/history/>. Diakses Juli 2014.