



Algoritma Genetika Menentukan Jalur Jalan dengan Lintasan Terpendek (Shortest Path)

Melladia

Teknik Informatika, fakultas Teknik, Universitas Nahdlatul Ulama Sumatera Barat
Melladia1311@gmail.com

Abstract

The shortest route (shortest path) is a problem to find the minimum route from the initial point (node) to the destination point (node). One of the artificial intelligence that can be used to solve the problem of finding the shortest route is the Genetic Algorithm. To get the right solution for optimization problems with one variable or multiple variables. The problem of traveling salesman problem (TSP) is one of the combinatorial optimization problems. TSP is a difficult problem when viewed from the point of computation. Several methods have been used to solve the problem and are a solution in determining the shortest trip through another city only once and returning to the city of origin of departure. Search techniques are carried out at the same time on a number of solutions known as populations. Individuals in a population are called chromosomes. This genetic algorithm consists of several main procedures, namely the selection procedure, crossover, mutation and elitism. Based on research results, the shortest path is 1-2-3-6-5-4-7-8-9-10 where the path is Sunur, Kurai Taji, Lapai, Jati, Pasar Pariaman, Gelombang, Rawang, Pauh, Sei Pasak, dan Koto Marapak with a path length of 55.8342.

Keywords: genetic algorithm, shortest path, TSP

Abstrak

Rute terpendek (shortest path) adalah permasalahan untuk mencari rute minimum dari titik (node) awal ke titik (node) tujuan. Salah satu kecerdasan buatan yang dapat digunakan untuk menyelesaikan masalah pencarian rute terpendek adalah Algoritma Genetika. Untuk mendapatkan solusi yang tepat untuk permasalahan optimasi dengan satu variabel atau multi variabel. Permasalahan travelling salesman problem (TSP) merupakan salah satu persoalan optimasi kombinatorial. TSP merupakan persoalan yang sulit bila dipandang dari sudut komputasinya. Beberapa metode telah digunakan untuk memecahkan persoalan dan merupakan solusi dalam menentukan perjalanan terpendek yang melalui kota lainnya hanya sekali dan kembali ke kota asal keberangkatan. teknik pencarian dilakukan sekaligus atas sejumlah solusi yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Algoritma genetika ini terdiri dari beberapa prosedur utama yaitu prosedur seleksi, crossover, mutasi dan elitisme. Berdasarkan hasil penelitian, jalur terpendek adalah 1-2-3-6-5-4-7-8-9-10 dimana jalur tersebut adalah Sunur, Kurai Taji, Lapai, Jati, Pasar Pariaman, Gelombang, Rawang, Pauh, Sei Pasak, dan Koto Marapak dengan panjang jalur 55,8342.

Kata kunci: algoritma genetika, shortest path, TSP

1. Pendahuluan

Travelling Salesman Problem (TSP) dikenal sebagai salah satu masalah optimasi yang menarik perhatian para peneliti sejak beberapa dekade terdahulu. Travelling Salesman Problem (TSP) termasuk ke dalam persoalan yang sangat terkenal dalam teori graf. Nama persoalan ini diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota[1]. Masalah dalam menentukan rangkaian terpendek diantara pasangan *node* (titik) tertentu dalam suatu *graph* telah banyak menarik perhatian. Permasalahan pencarian rute optimum dalam suatu perjalanan yang mencari jarak terpendek untuk mencapai tempat tujuan yang akan dikunjungi

semuanya dan setiap tempat yang hanya sekali dikunjungi[2].

Shortest Path merupakan metode pencarian lintasan terpendek pada sebuah graf untuk pencarian lintasan terpendek dari sebuah graf bersambung dan berarah, dapat digunakan perhitungan cara manual atau dengan melalui komputer Shortest Path yang diperoleh akan meminimumkan fungsi linear khusus dari Path seperti jarak, waktu dan biaya dihadapi selama melakukan perjalanan. Perumusan persoalan ini akan menjadi salah satu kegunaan dari lintasan dengan jarak (waktu) diminimumkan terhadap biaya yang dianggarkan[3].

Permasalahan yang dihadapi adalah bagaimana cara mengunjungi *vertek* pada *graph* dari *vertek* awal ke

vertex akhir dengan bobot minimum, dimana dalam hal ini bobot yang digunakan adalah jarak dan kota-kota yang dikunjungi diasumsikan sebagai *graph* yang saling terhubung (*connected graph*) antar suatu kota dengan kota lainnya. Suatu *graph* G disebut terhubung jika untuk setiap *vertex* dari *graph* terdapat jalur yang menghubungkan kedua *vertex* tersebut, atau dengan kata lain *graph* terhubung jika setiap *vertex* yaitu V_i dan V_j dalam suatu *graph* terdapat sedikitnya sebuah *edge*. *Edge* pada *graph* berarah disebut *arc*[4].

Beberapa metode algoritma yang telah dikembangkan untuk menyelesaikan persoalan jalur terpendek diantaranya Algoritma Dijkstra, Algoritma Floyd-Warshall dan Algoritma Bellman-Ford. Algoritma ini dapat diselesaikan dengan cepat jika kota-kota yang akan dikunjungi sedikit. Seiring dengan itu muncul permasalahan bagaimana menentukan jalur terpendek jika terdapat banyak jalur alternatif ke kota tujuan dengan mempertimbangkan efisiensi dan waktu sehingga diperlukan ketepatan dalam menentukan jalur terpendek antar suatu kota. Semakin banyak alternatif jalur ke kota tujuan, semakin rumit cara untuk menghitung jalur terpendek. Untuk itu diperlukan metode/cara yang handal untuk dapat menentukan jalur terpendek dari kota asal ke kota tujuan sehingga diperoleh solusi yang terbaik.

Kecerdasan Buatan (*Artificial intelligence*) merupakan salah ilmu pengetahuan yang digunakan agar dapat menyelesaikan masalah manusia dengan cara memahami, memprediksi dan memanipulasi. Kecerdasan dibuat agar sistem yang menggunakan algoritma tertentu sehingga sistem seolah-olah dapat berpikir menyamai manusia[5]. Maka penggunaan metode AI (*Artificial Intelligent*) atau kecerdasan buatan dalam perhitungan jalur terpendek merupakan salah satu solusi untuk dapat menyelesaikan masalah dengan jalur yang banyak dan rumit.

Pada tahun 70-an muncul sebuah algoritma baru yang dikenal dengan Algoritma Genetika (*Genetic Algorithm, GA*) yang merupakan salah satu cabang dari AI. Algoritma Genetika ini diperkenalkan oleh John Holland dari University of Michigan yang kemudian dipopulerkan oleh salah satu muridnya yaitu David Goldberg, sehingga Algoritma Genetika mulai digunakan secara luas ke berbagai bidang, termasuk untuk memecahkan permasalahan-permasalahan optimasi. Untuk menentukan jalur jalan dengan lintasan terpendek (*shortest path*) ini digunakan algoritma genetika. Algoritma genetika merupakan salah satu algoritma yang digunakan untuk menyelesaikan masalah optimasi. Algoritma genetika meniru cara kerja proses genetika pada makhluk hidup, dimana terdapat proses seleksi, *rossover* dan mutasi untuk mendapatkan kromosom terbaik pada suatu generasi[6-8].

Tujuan penelitian ini untuk menerapkan suatu konsep algoritma genetika untuk penyelesaian dalam

menentukan *shortest path problem* dalam kasus ini yang diangkat adalah memberi solusi optimasi dalam masalah TSP di wilayah Kota Pariaman,

2. Metode Penelitian

Metode penelitian merupakan sebuah langkah atau cara yang digunakan dalam melakukan penelitian yang dapat dipertanggung jawabkan hasilnya

Pada pelaksanaan penelitian ini pengambilan data dilakukan selama 2 hari dan data di dapatkan dari Dinas Perhubungan Kota Pariaman dan Dinas Pekerjaan Umum Kota Pariaman.

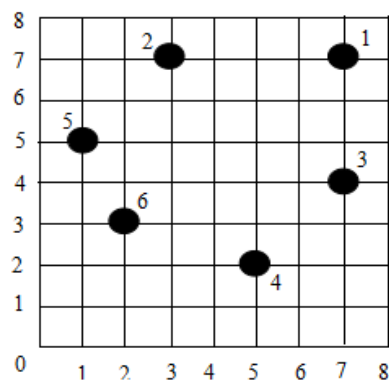
Dalam hal ini metode penelitian yang digunakan adalah metode *deskriptif*. Penelitian dengan metode *deskriptif* dapat diartikan sebagai proses pemecahan masalah yang diselidiki dengan melukiskan keadaan subyek dan obyek penelitian pada saat sekarang berdasarkan fakta-fakta yang tampak dan bagaimana adanya. Pelaksanaan metode penelitian *deskriptif* tidak terbatas sampai pada pengumpulan dan penyusunan data, tetapi meliputi analisis dan interpretasi tentang data tersebut[9-10].

3. Hasil dan Pembahasan

Langkah-langkah dalam penyelesaian TSP dengan algoritma genetika adalah sebagai berikut :

Skema pengkodean

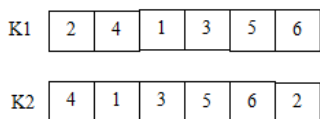
TSP dapat dirumuskan sebagai berikut : terdapat sekumpulan N node dengan posisi-posisi koordinatnya $\{X_i, Y_i\}$, $i = 1, 2, \dots, N$ perhatikan gambar 1 dibawah ini. Terdapat 6 node yang harus dikunjungi.



Gambar 1. Peta dua dimensi untuk TSP.

Suatu solusi dipresentasikan ke dalam suatu kromosom yang berisi nomor urut dari semua kota yang ada. Masing-masing nomor urut kota hanya boleh muncul satu kali didalam kromosom sehingga satu kromosom mempresentasikan satu rute perjalanan (satu solusi) yang valid. Dimana suatu kromosom mempresentasikan suatu permutasi dari nomor urut kota $1, 2, 3, \dots, N$.

Dengan demikian untuk gambar 1, suatu contoh kromosom adalah seperti pada gambar 2 dibawah ini :

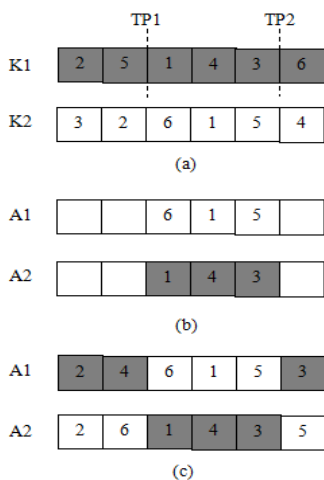


Gambar 4.2 Representasi kromosom untuk TSP

Pada gambar 2 kromosom K1 dan K2 mempresentasikan rute perjalanan yang sama. Hal ini bisa dipahami karena, secara siklus, K1 dan K2 memang memberikan rute perjalanan yang sama.

Pindah Silang

Pindah silang dapat diimplementasikan dengan skema *order crossover*. Pada skema ini satu bagian kromosom dipertukarkan dengan tetap menjaga urutan kota yang bukan bagian dari kromosom tersebut. Ilustrasi skema *order crossover* dapat dilihat pada gambar dibawah ini :



Gambar 3 Pindah silang menggunakan skema *order crossover*

Mula-mula 2 buah titik potong, TP1 dan TP2 dibangkitkan secara random untuk memotong 2 buah kromosom orang tua, K1 dan K2 (gambar 4.3a) kemudian 2 kromosom anak, A1 dan A2 mendapatkan gen-gen dari bagian kromosom K1 dan K2 secara menyilang kromosom A1 mendapatkan {6,1,5} dan A2 mendapatkan {1,4,3} (gambar 4.3b). Posisi-posisi gen yang masih kosong pada kromosom A1 diisi dengan gen-gen dari K1, secara berurutan dari gen 1 sampai gen 6 yang belum ada pada A1. Hal yang sama juga dilakukan untuk kromosom A2 (gambar 4.3c).

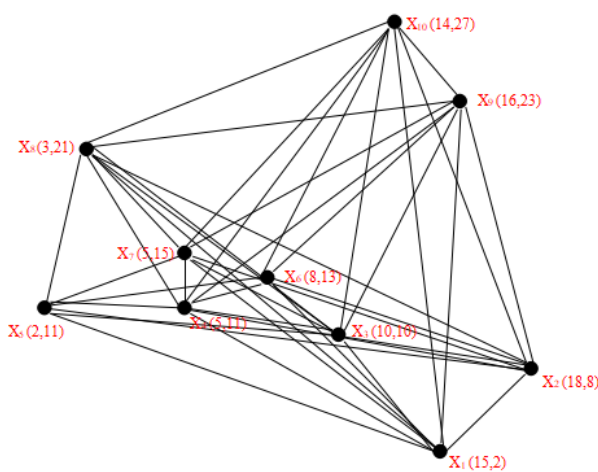
Mutasi

Operator mutasi biasanya diimplementasikan dengan menukarkan gen termutasi dengan gen lain yang dipilih secara random. Misalnya, kromosom {2,3,4,1,5} dapat termutasi menjadi kromosom {4,3,2,1,5}. Dalam hal ini gen 1 dan gen 3 saling ditukarkan. Skema mutasi ini dikenal sebagai *swapping mutation*.

3.1 Pengolahan Data

Berikut adalah contoh penyelesaian permasalahan *Travelling Salesman Problem* dengan menggunakan

algoritma genetika. Misalkna terdapat 10 kota yang akan dikunjungi yaitu kota 1 (sunur), kota 2 (kurai taji), kota 3 (lapai), kota 4 (gelombang), kota 5 (pasar pariaman), kota 6 (jati), kota 7 (rawang), kota 8 (pauh), kota 9 (sei.pasak), dan kota 10 (koto marapak). Letak masing-masing kota dinyatakan dalam koordinat. Perjalanan dimulai dari kota pertama dan akhirnya juga akan berakhir dikota pertama. Akan ditentukan jalur terpendek atau total bobot minimum yang akan ditempuh untuk mengunjungi 10 kota tersebut. Kota tersebut dipresentasikan dalam graf G berikut ini :



Gambar 4 Graph dengan G vertex

Penyelesaian :

Langkah 1 : inisialisasi

Koordinat masing-masing kota dapat dilihat pada graf G pada gambar 4. Dengan menggunakan persamaan *crossover* diperoleh jarak antar kota sebagaimana dipresentasikan dalam matrik bobot sisi graf G berukuran 10x10.

Langkah 2 :

Bentuk populasi awal dengan cara membangkitkan kromosom secara acak sebanyak ukuran populasi.

Hitung panjang jalur masing-masing kromosom dengan cara :

Untuk ukuran populasi :

$$\begin{aligned} \text{Kromosom [1]} &= 3-6-8-7-5-4-1-2-9-10 \\ \text{Panjang jalur [1]} &= \text{jarak}(3-6) + \text{jarak}(6-8) + \text{jarak}(8-7) \\ &\quad + \text{jarak}(7-5) + \text{jarak}(5-4) + \text{jarak}(4-1) \\ &\quad + \text{jarak}(1-2) + \text{jarak}(2-9) + \text{jarak}(9-10) + \text{jarak}(10-3) = 3,605 \\ &\quad + 9,434 + 6,325 + 5,000 + 3,000 \\ &\quad + 13,454 + 6,708 + 15,133 + 4,472 \\ &\quad + 17,464 = 84,595 \end{aligned}$$

Hitung nilai fitness masing-masing kromosom dengan menggunakan persamaan sebagai berikut :

$$\begin{aligned} \text{Nilai } \text{Nilaifitness}[i] &= 1 / \text{Panjangjalur}[i] \\ \text{Nilaifitness}[1] &= 1 / 84,595 = 0,0118 \end{aligned} \tag{1}$$

Dan seterusnya sampai populasi ke-30.

Langkah 3 :

Menghitung total *fitness*

$$\begin{aligned}
 TotalFitness &= \sum_{i=1}^{30} fitness[i] \\
 &= fitness(1) + fitness(2) + fitness(3) + \dots \\
 &\quad + fitness(30) \\
 &= 0,0118 + 0,0103 + 0,0095 + \dots \\
 &\quad + 0,0076 \\
 &= 0,3551
 \end{aligned}
 \tag{2}$$

Hitung *fitness* relatif

Fitness relatif dihitung dengan menggunakan persamaan

$$P[i] = \frac{Nilai fitness}{Total fitness}
 \tag{3}$$

Untuk *fitness* (1)

$$P[1] = 0,0118 / 0,3551 = 0,0332$$

Dan seterusnya sampai populasi ke-30.

Hitung *fitness* kumulatif

Fitness kumulatif dihitung dengan menggunakan persamaan berikut :

$$Q[i] = Q[i - 1] + P[i]
 \tag{4}$$

Untuk *fitness* (1) dan (2)

$$Q[1] = 0,0332$$

$$Q[2] = 0,0332 + 0,0290 = 0,0622$$

Seleksi Roda *Roulette-Wheel*

Bangkitkan nilai acak *r* yang bernilai antara 0 dan 1 sebanyak ukuran populasi (30)

Jika $R[k] < C[k]$ maka kromosom ke-*k* sebagai induk, selain itu pilih kromosom ke-*k* sebagai induk dengan syarat $C[k-1] < R[k] < C[k]$. Putar *roulette-wheel* sebanyak jumlah populasi yaitu 30 kali. Untuk $i = 1, 2, \dots, 30$, jika $R[i] \leq Q[1]$ maka pilih kromosom (1) pada urutan ke-1. Untuk $j = 2, 3, \dots, 29$, jika $Q[j] < R[i] \leq Q[j+1]$ maka pilih kromosom ke $[j+1]$ pada urutan ke-*i*.

Untuk populasi ke-1 :

$Q[9] < R[i] \leq Q[10]$ yaitu $0,2995 < 0,3120 \leq 0,3344$ maka pilih kromosom pada populasi ke-10 pada urutan ke-1. Begitu seterusnya sampai populasi ke-30.

Untuk $i = 1, 2, \dots, 30$

Jika $R[i] < P_c$ atau $R[i] < 0,500$ maka pilih kromosom ke-*i* sebagai induk.

Populasi yang bilangan yang kurang dari P_c adalah populasi ke : 2, 8, 11, 14, 17, 18, 21, 23, 26, 27, dan 28. Hal ini berarti bahwa kromosom yang berhak untuk melakukan *crossover* adalah kromosom pada populasi seperti yang telah disebutkan diatas.

Untuk $k=1, 2, \dots, 11$, karena $\text{mod}(11,2) \neq 0$, sehingga $k=k-1 = 11-1 = 10$. Jadi buang salah satu kromosom, misalkan kromosom ke 28 dibuang.

Lakukan *crossover* antara kromosom 2 dengan 8, 11 dengan 14, 17 dengan 18, 21 dengan 23, dan kromosom 26 dengan 27.

Crossover antara kromosom 2 dengan 8 :

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

Anak 1	x	x	x	x	5	3	x	x	x	x
Anak 2	x	x	x	x	5	7	x	x	x	x

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

Lakukan proses *crossover* dengan cara yang sama untuk kromosom 11 dengan 14, 17 dengan 18, 21 dengan 23, dan kromosom 26 dengan 27.

Proses Mutasi

Bangkitkan bilangan acak *r* yang bernilai antara 0 dan 1 sebanyak ukuran populasi (30).

Untuk $i=1, 2, \dots, 30$ maka, jika $R[i] < 0,100$ maka kromosom ke-*i* terkena mutasi. karena tidak ada $R[i] < 0,100$ maka tidak ada kromosom yang terkena mutasi.

Proses Elitisme

Untuk $i=1, 2, \dots, 30$ maka, jika $R[i] < 0,100$ maka lakukan penggantian pada kromosom ke-*i*. bahwa bilangan acak yang nilainya kurang dari 0,100 adalah bilangan acak pada populasi ke-4 maka lakukan penggantian pada kromosom ke-4.

Tabel 1 Populasi akhir ke-i

Pop ular	Kromosom	Panjang Jalur	Fitness	Pop ular	Kromosom	Panjang Jalur	Fitness
1	3-6-8-7-5-4-1-2-9-10	84,595	0,0118	16	5-6-9-10-8-7-3-2-1-4	80,935	0,0124
2	9-7-4-6-5-3-1-2-10-8	96,832	0,0103	17	5-4-3-1-2-6-7-9-10-8	79,678	0,0126
3	4-7-8-6-3-5-1-9-10-2	105,491	0,0095	18	8-7-5-4-6-3-2-1-9-10	74,514	0,0134
4	10-8-7-6-5-4-3-1-2-9	72,63	0,0138	19	4-5-7-6-8-10-9-3-2-1	80,767	0,0124
5	10-6-4-5-7-3-8-2-1-9	98,995	0,0101	20	3-2-1-4-5-6-7-9-10-8	84,975	0,0118
6	7-8-6-5-4-3-1-2-9-10	80,929	0,0124	21	10-6-7-5-4-3-2-1-9-8	93,596	0,0107
7	6-7-5-4-3-1-2-9-10-8	74,415	0,0134	22	6-9-10-8-7-5-4-3-2-1	77,223	0,0129
8	2-10-8-5-4-7-6-9-3-1	95,866	0,0104	23	2-8-10-9-7-6-4-5-3-1	84,866	0,0118
9	1-7-8-10-6-9-4-5-3-2	108,058	0,0093	24	7-6-4-5-3-1-2-9-10-8	72,874	0,0137
10	3-2-9-10-8-7-5-4-6-1	80,782	0,0124	25	10-8-7-6-5-4-3-1-2-9	72,63	0,0138
11	9-10-8-7-4-6-3-1-2-5	85,396	0,0117	26	4-10-8-9-7-5-6-3-1-2	102,053	0,0098
12	7-8-5-4-6-2-1-3-9-10	84,091	0,0119	27	9-7-5-4-3-1-2-6-8-10	80,457	0,0124
13	6-10-9-3-2-1-4-5-7-8	86,188	0,0116	28	2-3-1-4-7-5-6-9-8-10	104,362	0,0096
14	2-1-3-4-5-7-6-8-10-9	74,415	0,0134	29	9-5-2-8-7-3-6-4-1-10	118,12	0,0085
15	7-5-4-6-3-1-2-9-10-8	69,811	0,0143	30	8-2-10-5-6-1-7-4-3-9	131,598	0,0076

Jadi hasil populasi akhir pada generasi ke-i pada tabel 1 terlihat bahwa :

Fitness terbaik berada pada populasi ke-4 dengan nilai fitness 0,0138.

Fitness terburuk berada pada populasi ke-30 dengan nilai fitness 0,0076.

Populasi akhir pada generasi ke-1 ini akan dijadikan sebagai populasi awal untuk generasi ke-2 dan lakukan langkah 1 sampai 3 untuk populasi ke-2 sampai generasi maksimum (generasi ke-50).

Langkah 4 : Selesai

Setelah dilakukan pengujian dengan Matlab 7.9 diperoleh rekap hasil masing-masing generasi seperti terlihat pada table 2 berikut :

Tabel 2 Rekap Hasil Setelah Diuji Dengan Matlab 7.9

No	Pola jalur TSP	F. Terbaik	F. Terburuk	Rata - rata
1	10 9 5 1 2 6 4 3 7 8	0.0127	0.0076	0.0092
2	10 9 5 1 2 6 4 3 7 8	0.0127	0.0078	0.0097
3	10 9 5 1 2 6 4 3 7 8	0.0127	0.0076	0.0100
4	1 2 6 3 7 5 4 8 9 10	0.0155	0.0076	0.0106
5	1 2 6 3 7 5 4 8 9 10	0.0155	0.0083	0.0109
6	1 2 6 3 7 5 4 8 9 10	0.0155	0.0083	0.0112
7	1 2 6 3 7 5 4 8 9 10	0.0155	0.0085	0.0117
8	1 2 6 3 7 5 4 8 9 10	0.0155	0.0085	0.0118
9	1 2 6 3 7 5 4 8 9 10	0.0155	0.0085	0.0121
10	1 2 6 3 7 5 4 8 10 9	0.0157	0.0085	0.0125
11	1 2 6 3 7 5 4 8 10 9	0.0157	0.0088	0.0131
12	1 2 3 7 6 5 4 8 9 10	0.0159	0.0087	0.0138
13	1 2 3 7 6 5 4 8 9 10	0.0159	0.0085	0.0137
14	1 2 3 7 6 5 4 8 9 10	0.0159	0.0086	0.0136
15	1 2 3 7 6 5 4 8 9 10	0.0159	0.0095	0.0138
16	1 2 3 6 7 5 4 8 9 10	0.0172	0.0096	0.0138
17	1 2 3 6 7 5 4 8 9 10	0.0172	0.0092	0.0135
18	1 2 3 6 7 5 4 8 9 10	0.0172	0.0097	0.0145
19	1 2 3 6 7 5 4 8 9 10	0.0172	0.0097	0.0150
20	1 2 3 6 7 5 4 8 9 10	0.0172	0.0096	0.0142
21	1 2 3 6 7 5 4 8 9 10	0.0172	0.0096	0.0151
22	1 2 3 6 7 5 4 8 9 10	0.0172	0.0105	0.0146
23	1 2 3 6 7 5 4 8 9 10	0.0172	0.0099	0.0153
24	1 2 3 6 7 5 4 8 9 10	0.0172	0.0097	0.0147
25	1 2 3 6 5 4 7 8 9 10	0.0179	0.0086	0.0149
26	1 2 3 6 5 4 7 8 9 10	0.0179	0.0092	0.0156
27	1 2 3 6 5 4 7 8 9 10	0.0179	0.0089	0.0154
28	1 2 3 6 5 4 7 8 9 10	0.0179	0.0104	0.0163
29	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0165
30	1 2 3 6 5 4 7 8 9 10	0.0179	0.0111	0.0170
31	1 2 3 6 5 4 7 8 9 10	0.0179	0.0123	0.0168
32	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0163
33	1 2 3 6 5 4 7 8 9 10	0.0179	0.0106	0.0161
34	1 2 3 6 5 4 7 8 9 10	0.0179	0.0117	0.0164
35	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0162
36	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0162
37	1 2 3 6 5 4 7 8 9 10	0.0179	0.0123	0.0170
38	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0167
39	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0168
40	1 2 3 6 5 4 7 8 9 10	0.0179	0.0099	0.0161
41	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0169

42	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0163
43	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0168
44	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0160
45	1 2 3 6 5 4 7 8 9 10	0.0179	0.0105	0.0170
46	1 2 3 6 5 4 7 8 9 10	0.0179	0.0106	0.0175
47	1 2 3 6 5 4 7 8 9 10	0.0179	0.0179	0.0179
48	1 2 3 6 5 4 7 8 9 10	0.0179	0.0179	0.0179
49	1 2 3 6 5 4 7 8 9 10	0.0179	0.0179	0.0179
50	1 2 3 6 5 4 7 8 9 10	0.0179	0.0179	0.0179

Fitness rata-rata generasi ke-i diperoleh dengan :
Fitness rata-rata (47)

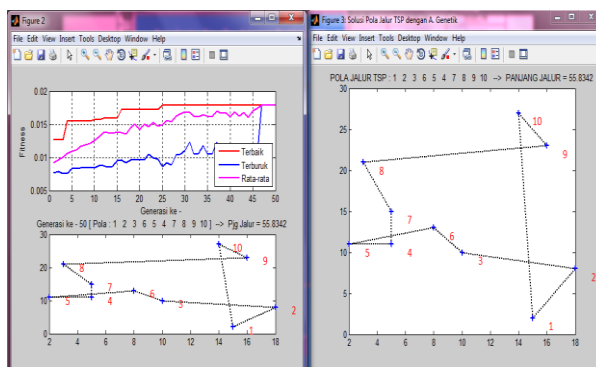
$$= \frac{Fitness_{Terbaik}(47) + fitness_{Terburuk}(47)}{2}$$

$$= \frac{0,0179 + 0,0179}{2}$$

$$= \frac{0,0358}{2}$$

Dari tabel 2 terlihat bahwa nilai fitness paling maksimum adalah 0,0176 dengan kromosom 1-2-3-6-5-4-7-8-9-10. Ini berarti bahwa jalur terpendek setelah dilakukan pencarian oleh algoritma genetika untuk 50 generasi adalah 1-2-3-6-5-4-7-8-9-10 dengan panjang jalur 55,8342.

Hasil pemrosesan Algoritma Genetika menggunakan Matlab sebagai berikut:



Gambar 5 Grafik Hasil Pemrosesan Algoritma Genetika

Pada gambar 5 terdapat dua buah grafik, dimana pada figure 2 memperlihatkan fitness terbaik, fitness terburuk, dan fitness rata-rata. Rute optimum yang harus dilalui salesman adalah 1-2-3-6-5-4-7-8-9-10 dimana jalur tersebut adalah Sunur, Kurai Taji, Lapai, Jati, Pasar Pariaman, Gelombang, Rawang, Pauh, Sei Pasak, dan Koto Marapak dengan panjang jalur 55,8342 seperti yang terlihat pada figure 3 pada gambar 5.

4. Kesimpulan

Kesimpulan yang di ambil dari hasil penelitian terhadap persoalan Travelling Salesman Problem (TSP) dapat diselesaikan dengan menggunakan algoritma genetika. Walaupun solusi TSP yang dihasilkan oleh algoritma genetika belum tentu merupakan solusi paling optimal (misalnya apabila yang dilalui sangat banyak), namun algoritma genetika akan menghasilkan solusi yang

lebih optimal pada setiap generasinya. Hal tersebut terlihat dari nilai *fitness* tiap generasi.

Kelebihan algoritma genetika sangat terlihat dari adaptivitasnya dalam menyelesaikan masalah. Begitu kita bisa mengkodekan masalah ke dalam kromosom dan bisa membangun fungsi *fitness*, maka kita dapat membangun algoritma genetika untuk masalah tersebut. Beberapa komponen algoritma genetika, misalnya Inisialisasi Populasi, *Linear Fitness Ranking*, *Roulette-Wheel*, Pindah Silang dan Mutasi bisa digunakan untuk beberapa masalah berbeda termasuk masalah TSP.

Berdasarkan hasil penelitian, jalur terpendek adalah 1-2-3-6-5-4-7-8-9-10 dimana jalur tersebut adalah Sunur, Kurai Taji, Lapai, Jati, Pasar Pariaman, Gelombang, Rawang, Pauh, Sei Pasak, dan Koto Marapak dengan panjang jalur 55,8342.

Diharapkan nantinya dapat dikembangkan sebuah pencarian TSP dengan algoritma genetika dan pengujian menggunakan aplikasi lain yang dapat ditampilkan dalam bentuk peta sebenarnya dari suatu daerah yang diteliti dengan cakupan jarak dan wilayah yang lebih besar dan luas.

Daftar Rujukan

- [1] Amozhita, K. K., Suyitno, S., Mashuri., (2019). Menyelesaikan *Travelling Salesman Problem* Dengan Metode Dua Sisi Optimal Pada PT. Es Malindo Boyolali. *UNNES Journal of Mathematics*, 8(1), 21-29.
- [2] Utomo, R. G., Maylawati, D. S., Alam, C. N., (2018). Implementasi Algoritma Cheapest Insertion Heuristic (CIH) dalam Penyelesaian Travelling Salesman Problem (TSP). *JOIN (Jurnal Online Informatika)*, 3 (1), 61-67.
- [3] Simanjuntak, O.S., Seminar Nasional Informatika 2012 (semnasIF 2012). In : UPN "Veteran", *Pengembangan Shortest Path Algorithm (SPA) Dalam Rangka Pencarian Lintasan Terpendek Pada Graf Bersambung Berarah Terurai*, Yogyakarta, 30 Juni 2012. UPN "Veteran": Yogyakarta.
- [4] Salaki, D. T., (2011). Penentuan Lintasan Terpendek Dari FMIPA ke Rektorat dan Fakultas di UNSRAT Manado Menggunakan Algoritma Dijkstra. *Jurnal Ilmiah Sains*. 11(1), 73-78
- [5] Melladia, M., & Mardani, I. R., (2018). Implementasi Algoritma Backpropagation Prediksi Kegagalan Siswa Pada Mata Pelajaran Matematika. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 2(3), 753 - 759. <https://doi.org/10.29207/resti.v2i3.588>.
- [6] Suyanto. (2005). *Algoritma Genetika dalam MATLAB*. Yogyakarta : Andi.
- [7] Saputro, Nico., (2003). *Pengenalan Huruf dengan Memakai Algoritma Genetik*. Integral, Vol 8 No.2, Oktober 2003.
- [8] Syamsudin, Aries., (2004). *Pengenalan Algoritma Genetik. (IlmuKomputer.com)*
- [9] Utami, P. Y., Suhery, C., Ilhamsyah., (2014). Aplikasi Pencarian Rute Terpendek Menggunakan Algoritma Genetika. *Jurnal Coding Sistem Komputer Universitas Tangjungpura*, 02(1), 19-25.
- [10]Febriyana, R., Mahmudy, W. F., (2016). Penjadwalan Kapal Penyeberangan Menggunakan Aloritma Genetika. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 3(1), 43-50.