

ANALISIS PERBANDINGAN ALGORITMA GENETIKA DAN ALGORITMA FUZZY EVOLUSI DALAM PENYELESAIAN TRAVELING SALESMAN PROBLEM

Tedi Sefuro¹, Slamet Sudaryanto N., ST, M. Kom²

Teknik Informatika, Ilmu Komputer, Universitas Dian Nuswantoro

Jl. Nakula 1 No. 5-11, Semarang, 50131, (024) 3517261

E-mail : Tedyrocknrollpalsu@yahoo.com

Abstrak

Traveling Salesman Problem (TSP) merupakan sebuah permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan seperti *routing* dan penjadwalan produksi. Adanya Algoritma Fuzzy Evolusi yang memberikan suatu solusi untuk dapat dibandingkan dengan Algoritma Genetika yang dimana kedua algoritma tersebut mempunyai kesamaan pada Tahapan-tahapan algoritma genetika, namun untuk untuk penentuan parameter-parameter genetika seperti halnya nilai probabilitas rekombinasi dan nilai probabilitas mutasi dihasilkan melalui sistem fuzzy menjadikan daya Tarik untuk melakukan perbandingan kedua algoritma tersebut. Setelah proses pengaplikasian dengan data pada PT Jalur Nugraha Ekakurir (JNE) dan pengujian lebih lanjut dengan mengubah parameter populasi diperoleh hasil algoritma fuzzy evolusi mendapat nilai yang lebih baik dibandingkan dengan algoritma genetika dalam penyelesaian TSP.

Kata kunci : *Perbandingan algoritma genetika, dan algoritma fuzzy evolusi, Traveling salesman problem.*

Abstract

Traveling Salesman Problem (TSP) is an optimization problems that can be applied to a variety of activities such as routing and scheduling production. The existence of Fuzzy Evolutionary Algorithm that provides a solution to be compared with the genetic algorithm in which the algorithms are having similarities in stages genetic algorithms, however, for the determination of parameters for genetic as well as the value of the probability of recombination and mutation probability values generated through a fuzzy system makes attractiveness to do a comparison of the two algorithms. Once the application process with the data on the PT Jalur Nugraha Ekakurir (JNE) and further testing by composing the population parameters obtained results of fuzzy evolutionary algorithm gets better value compared with genetic algorithm in solving the TSP.

Keywords : *Comparison of genetic algorithms, evolutionary fuzzy algorithms, the Traveling Salesman Problem.*

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komputer dan ilmu pengetahuan saat ini mengalami perkembangan yang sangat pesat terutama dalam hal proses komputasi yang memungkinkan sebuah masalah dapat diselesaikan dengan bantuan komputer sebagai solusinya. Seiring dengan itu pula peningkatan ilmu pengetahuan telah melahirkan begitu banyak algoritma-algoritma yang sangat membantu bagi pekerjaan manusia.

Traveling Salesman Problem (TSP) merupakan sebuah permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan seperti *routing* dan penjadwalan produksi. Masalah optimasi TSP terkenal dan telah menjadi standar untuk mencoba algoritma yang komputational. Pokok permasalahan dari TSP adalah seorang *salesman* harus mengunjungi sejumlah kota yang diketahui jaraknya satu dengan yang lainnya. Semua kota yang ada harus dikunjungi oleh *salesman* tersebut dan kota tersebut hanya boleh dikunjungi tepat satu kali. Permasalahannya adalah bagaimana *salesman* tersebut dapat mengatur rute perjalanannya sehingga jarak yang ditempuhnya merupakan jarak minimum [1].

Banyak metode yang dapat dipakai untuk menyelesaikan TSP yaitu *Hill Climbing Method*, *Ant Colony System* dan *Dynamic Programming*. Metode lain yang dapat dipakai untuk menyelesaikan TSP adalah algoritma genetika. Algoritma genetic merupakan sebuah

algoritma yang meniru cara kerja proses genetika pada makhluk hidup, dimana terdapat proses seleksi, rekombinasi dan mutasi untuk mendapatkan kromosom terbaik pada suatu generasi [1]. Dengan meniru teori evolusi, algoritma genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata. Sebelum algoritma dapat dijalankan, maka sebuah kode yang sesuai untuk persoalan harus dirancang. Untuk ini maka solusi layak dalam ruang permasalahan di kodekan dalam bentuk kromosom yang terdiri atas komponen genetika terkecil yaitu gen. Dengan teori evolusi dan teori genetika, di dalam penerapan algoritma genetika akan melibatkan beberapa operator, yaitu reproduksi, *crossover*, dan mutasi.

Jenis dari operator dasar algoritma genetika bermacam-macam dan terus berkembang. Pada skripsi ini, operator genetika yang akan digunakan adalah *roda roulette selection* untuk reproduksi, *Order Crossover* (OX) untuk *crossover*, dan *insertion mutation* untuk mutasi. Secara umum, tahapan dari algoritma genetika diawali dengan pembentukan populasi awal. Selanjutnya dilakukan proses yang menggunakan ketiga operator genetika tersebut untuk membentuk populasi baru yang akan digunakan untuk generasi berikutnya. Banyaknya proses *crossover* dan mutasi bergantung pada masing masing nilai parameter probabilitas yang telah di tentukan sebelumnya. Proses atau tahapan pada algoritma genetika tersebut dilakukan berulang

kali sampai mencapai kriteria berhenti, dalam hal ini batas generasi yang ditentukan[2].

Adanya Algoritma Fuzzy Evolusi yang memberikan suatu solusi untuk dapat dibandingkan dengan Algoritma Genetika yang dimana kedua algoritma tersebut mempunyai kesamaan pada Tahapan-tahapan algoritma genetika, namun untuk penentuan parameter-parameter genetika seperti halnya nilai probabilitas rekombinasi dan nilai probabilitas mutasi dihasilkan melalui sistem fuzzy menjadikan daya Tarik untuk melakukan perbandingan kedua algoritma tersebut.

Berdasarkan latar belakang yang telah dikemukakan di atas, maka penulis tertarik, untuk melakukan penelitian penerapan *Genetic Algorithm* dan *Algoritma Fuzzy Evolusi* dengan judul “ANALISIS PERBANDINGAN ALGORITMA GENETIKA DAN ALGORITMA FUZZY EVOLUSI DALAM PENYELESAIAN TRAVELING SALESMAN PROBLEM”.

1.2 Rumusan Masalah

1. Sesuai dengan latar belakang yang telah di jelaskan sebelumnya, maka dapat di simpulkan, Bagaimana algoritma genetika dan algoritma fuzzy evolusi dapat menyelesaikan masalah TSP sehingga mempunyai jarak minimum dalam pengiriman barang di PT Jalur Nugraha Ekakurir (JNE) Semarang?
2. Dari rumusan masalah yang pertama dapat di Bentuk perwujudan dari

Traveling Salesman Problem dengan menggunakan perbandingan algoritma genetika dan algoritma fuzzy evolusi, yaitu: seperti apa perbandingan menggunakan perangkat lunak matlab?

1.3 Batasan Masalah

Batasan masalah yang ditekankan pada penulisan Tugas Akhir ini yaitu:

1. Algoritma genetika dan algoritma fuzzy evolusi diterapkan hanya untuk pencarian jarak terpendek pada masalah TSP.
2. Jalur transportasi yang digunakan yaitu jalur pengambilan barang dari agen JNE di kota Semarang Barat.
3. Tidak ada prioritas agen mana yang akan dilalui terlebih dahulu.
4. Menggunakan jarak yang sebenarnya.
5. Pembangkitan bilangan acak pada interval 0 sampai 1
6. Parameter-parameter yang digunakan : Jumlah populasi = Jumlah Agen JNE 10, Jumlah Generasi=1...100, Jumlah Kromosom = Jumlah Agen, Probabilitas Crossover, Probabilitas mutasi.
7. Menggunakan aturan-aturan fuzzy model Xu.

1.4 Maksud dan Tujuan Penelitian

Maksud dari penulisan skripsi ini adalah menerapkan algoritma genetika dan algoritma fuzzy evolusi sebagai metode untuk mencari jarak terpendek dalam penyelesaian *Traveling Salesman Problem*.

Tujuan dari penelitian adalah :

1. Untuk mengetahui cara kerja algoritma genetika dan algoritma fuzzy evolusi dalam memecahkan masalah *Traveling Salesman Problem*.
2. Membuat Aplikasi atau perangkat lunak untuk mensimulasikan kedua algoritma dalam menyelesaikan masalah TSP.
3. Melakukan analisis hasil eksekusi dari perangkat lunak yang dibangun terhadap beberapa contoh kasus dalam menyelesaikan *Traveling salesman problem*. Analisis bertujuan untuk mengetahui perbandingan performansi kedua algoritma tersebut.

1.5 Manfaat Penelitian

Hasil dari penelitian ini bermanfaat untuk mengetahui perbandingan algoritma genetika dan algoritma fuzzy evolusi untuk penyelesaian *Traveling Salesman Problem*. Dapat dilihat dari sudut pandangan antara dua algoritma yang berbeda tetapi saling berhubungan ini, pembaca akan mengetahui perbedaan studi komparatif antara dua algoritma ini dengan studi komparatif algoritma yang lain.

2. METODOLOGI

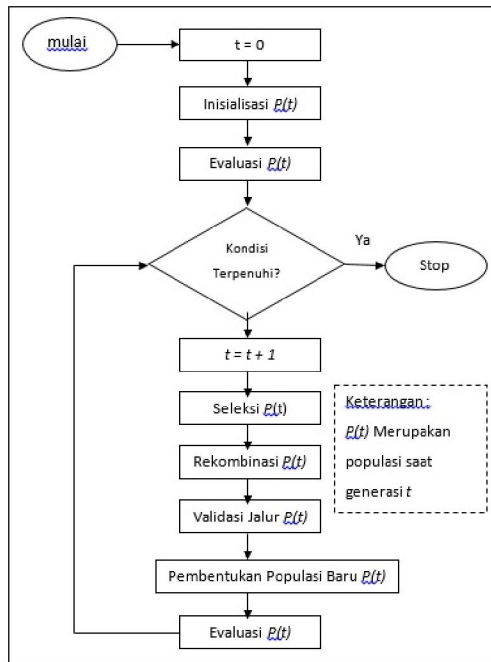
Pada bab ini akan dibahas mengenai penggunaan algoritma genetika dan algoritma fuzzy evolusi untuk menyelesaikan masalah

jalur terpendek. Beberapa proses penting yang harus dilakukan untuk mengimplementasikan algoritma genetika dan algoritma fuzzy evolusi dalam mencari jalur terpendek yaitu sebagai berikut [3] :

1. Representasi kromosom.
2. Inisialisasi Populasi.
3. Fungsi evaluasi/fitness.
4. Seleksi.
5. Operator genetika, yaitu operator rekombinasi (*crossover*) dan mutasi.
6. Penentuan parameter, yaitu parameter control algoritma genetika, yaitu : ukuran populasi(*popsize*), peluang *crossover* (*pc*) dan peluang mutasi (*pm*) Dalam penentuan parameter ini dilakukan proses system fuzzy untuk mendapatkan nilai yang akan digunakan sebagai parameter.

Dalam hal ini kedua algoritma mempunyai kesinambungan yang dimana pada algoritma fuzzy evolusi hanya berbeda pada penentuan parameternya, oleh karena itu proses yang dilakukan sama dengan algoritma genetika.

2.1 Proses Algoritma Genetika



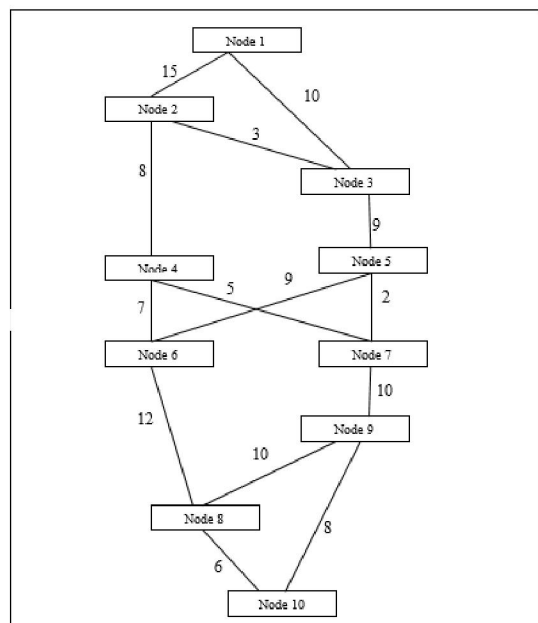
Gambar 3.2 Diagram alir proses algoritma genetika

2.1.1 Pengkodean Atau Reprerentasi Kromosom

Pada pengaplikasian algoritma genetika yang akan dijelaskan pada skripsi ini adalah jenis representasi kromosom yang digunakan adalah pengkodean permutasi. Simpul-simpul yang akan dikodekan pada jaringan bilangan bulat positif $1, 2, 3, 4, \dots, n$, dimana n adalah banyaknya simpul pada jaringan. Tiap kode atau simpul dianggap sebagai gen pada kromosom, sehingga kromosom merupakan untaian kode-kode dari simpul pada jaringan yang tidak berulang dan mempresentasikan suatu urutan atau jalur.

Contoh 3.1 :

Berikut adalah persoalan masalah jalur terpendek dari suatu jaringan data yang terdapat 10 titik yang disimbolkan menjadi simpul. Dengan *Node* Sumber adalah *Node* 1 dan *Node* 10 adalah node akhir. Karena banyaknya simpul 10, maka panjang kromosom atau gen pada satu kromosom adalah 10.



Gambar 3.3 Topologi jaringan

2.1.2 Inisialisasi Populasi

Dalam tahap ini akan dibangkitkan sebuah populasi dengan jumlah kromosom yang telah di tentukan jumlahnya.

Dengan data yang sudah ada yaitu terdapat 10 simpul dan 10 kromosom yang terbentuk seperti terlihat pada tabel 3.2. Kromosom yang terbentuk secara acak dengan

menetapkan gen pertama sebagai *node* sumber, dalam hal ini adalah *node* 1.

Tabel 2.1 Populasi Awal Terbentuk

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

2.1.3 Evaluasi Fungsi Fitness

Fungsi *fitness* digunakan untuk menentukan seberapa baik individu yang direpresentasikan oleh suatu kromosom. Dalam kasus ini permasalahan jalur terpendek yaitu untuk mencari jarak terpendek dari 10 *node* dan 14 busur yang telah disebutkan pada Contoh 3.1 Nilai *fitness* yang dapat digunakan adalah $1 / \text{total jarak}$ (satu pertotal jarak). Dalam hal ini yang dimaksud total jarak adalah jumlah jarak antara satu *node* dengan *node* lainnya yang dapat dilalui. Semakin tinggi nilai *fitness* dari suatu individu atau kromosom, maka semakin baik individu tersebut [2]. Nilai *fitness* ini juga bergantung pada keabsahan dari jalur yang terkandung dalam kromosom yang bersangkutan. Jika ada kromosom yang memiliki jalur dari *node* sumber ke *node*

tujuan yang valid, maka nilai *fitness* akan sama dengan nilai dari fungsi *fitness* yang telah ditentukan. Berikut adalah penentuan nilai *fitness* pada skripsi ini [2]:

$$F = \begin{cases} \frac{1}{n-1 \sum_{i=1} C_i(g_i, g_{i+1})}; & \text{Jalur valid} \\ 0 & ; \text{Jalur tidak valid} \end{cases}$$

Di mana $C_i(g_i, g_{i+1})$ adalah cost antara gen g_i dan gen tetangganya g_{i+1} dalam kromosom dari n gen (simpul).

2.1.4 Seleksi

Setelah terbentuk populasi awal, selanjutnya hasil populasi tersebut akan di seleksi. Metode seleksi yang digunakan dalam algoritma genetika untuk pencarian jalur terpendek ini adalah seleksi roda *roulette*.

Dalam metode roda *roulette*, proses seleksi individu diibaratkan seperti dalam permainan judi roda *roulette*. Dimana pemain akan memutar roda yang telah terpartisi menjadi beberapa bagian untuk mendapatkan suatu hadiah. Kaitanya dengan metode seleksi yang dibahas ini adalah suatu kromosom diibaratkan sebagai hadiah [2]. Partisi-partisi pada roda *roulette* merupakan interval dari nilai kumulatif probabilitas masing-masing kromosom. Kemudian proses memutar roda dinyatakan dengan menentukan suatu bilangan acak pada interval $[0 - 1]$. Pada proses seleksi ini suatu kromosom kadang terpilih lebih dari sekali dan lebih dari satu. kromosom-kromosom

yang terpilih tersebut akan membentuk populasi orang tua.

Adapun tahapan dari proses seleksi sebagai berikut [2]:

Tahap 1: Hitung nilai *fitness* dari masing-masing kromosom

Tahap 2: Hitung total *fitness* dari masing-masing kromosom.

Tahap 3: Hitung probabilitas dan nilai kumulatif probabilitas masing-Masing kromosom.

Tahap 4: Dari probabilitas tersebut hitung jatah masing-masing individu Pada angka 0 sampai 1, atau dengan kata lain menentukan intereval kumulatif probalitas masing-masing kromosom.

Tahap 5: Bangkitkan bilangan acak antara 0 – 1.

Tahap 6: Dari bilangan acak yang dihasilkan,tentukan kromosom mana Yang terpilih dalam proses seleksi menurut interval yang Bersesuaian yang telah ditentukan sebelumnya pada tahap 4.

2.1.5 Crossover

Salah satu komponen yang paling penting dalam algoritma genetika adalah *Crossover* atau pindah silangan. *Crossover* merupakan suatu proses persilangan sepasang kromosom orang tua untuk menghasilkan *offspring* yang akan menjadi individu pada populasi di generasi berikutnya. *Offspring* yang di hasilkan dari proses *Crossover* diharapkan mewarisi sifat-sifat unggul yang dimiliki oleh

kromosom orang tua. Pindah silang pada masalah jalur terpendek ini menggunakan *Order Crossover*. Banyaknya kromosom yang di *crossover* di pengaruhi oleh parameter probabilitas *crossover* (*pc*).

Adapun tahapan dari proses *Crossover* sebagai berikut [2]:

Tahap 1: Pilih dua kromosom berbeda pada populasi orang tua secara Berurutan

Tahap 2: Pilih *Substring* dari orang tua secara berurut.

Tahap 3: Salin *Substring* dan *node* sumber ke *offspring* yang akan Di generasi dengan posisi yang sama

Tahap 4: Abaikan gen dengan nilai yang sama dengan yang sudah ada di Tahap 2.

Tahap 5: Tempatkan sisa *Substring* kromosom orang tua ke *offstring* Setelah daerah *Substring* dari *Offstring* dengan urutan yang sama.

Setelah *Offstring* terbentuk dari proses *Crossover* maka selanjutnya adalah memvalidasi jalur yang terkandung didalamnya, karena bisa jadi *Offstring* yang terbentuk merepresentasikan jalur yang tidak valid [2]. Jika jalur tersebut tidak valid, maka *Offspring* tersebut tidak akan menjadi bagian dari generasi berikutnya. Sebaliknya, jika jalur valid, maka *offspring* tersebut dapat menjadi bagian dari generasi berikutnya.

2.1.6 Mutasi

Mutasi adalah suatu proses yang dilakukan untuk mempertahankan keanekaragaman genetic populasi. Hal tersebut dilakukan untuk

mencegah populasi terjebak dalam solusi optimal lokal.

Daftar populasi baru hasil *Crossover* dipilih secara acak untuk dilibatkan dalam proses mutasi. Pada algoritma genetika, mutasi memainkan peran penting, yaitu menggantikan gen-gen yang hilang dari populasi selama proses seleksi atau mengembalikan kromosom optimal yang hilang akibat proses *Crossover* [2]. Dan muncul kromosom yang tidak di tampilkan pada populasi awal yang bisa jadi lebih baik dari kromosom pada populasi saat itu.

Adapun tahapan dari proses mutasi sebagai berikut berikut [2]:

Tahap 1: Pilih satu kromosom pada populasi anak hasil *Crossover*.

Tahap 2: Pilih dua posisi secara acak. Posisi pertama digunakan untuk Menandakan gen mana yang akan dimutasi atau disisipkan Keposisi kedua.

Tahap 3: Sisipkan gen pada posisi pertama ke posisi kedua.

Setelah *Offspring* hasil mutasi terbentuk maka selanjutnya adalah memvalidasi jalur yang terkandung di dalamnya dengan teknik yang sama pada setelah *Crossover*, karena bisa jadi *Offspring* hasil mutasi tersebut mempresentasikan jalur yang tidak valid [2]. Jika jalur tersebut tidak valid, maka *offspring* tersebut tidak akan menjadi bagian dari generasi berikutnya. Sebaliknya, jika jalur valid, maka *Offspring*

tersebut dapat menjadi bagian dari generasi berikutnya.

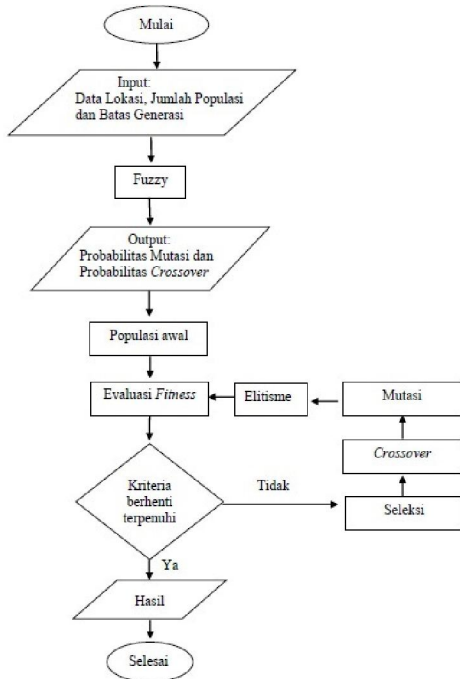
2.1.7 Pembentukan Populasi Untuk Generasi Berikutnya

Langkah awal dari pemilihan individu untuk generasi berikutnya adalah dengan menggabungkan semua kromosom orang tua dan semua kromosom anak baik yang mengalami mutasi maupun tidak mengalami mutasi. Selanjutnya hitung nilai *Fitness* gabungan kromosom tersebut. Lalu sorting kromosom dari yang memiliki *fitness* tertinggi sampai terendah. Terakhir ambil kromosom yang memiliki nilai *fitness* tertinggi sebanyak ukuran populasi yang telah ditentukan di awal.

2.1.8 Kriteria Berhenti

Proses-proses pada algoritma genetika akan terus berulang sampai mencapai suatu kriteria berhenti tertentu. Kriteria berhenti yang digunakan pada algoritma genetika untuk pencarian jalur terpendek ini adalah *generations*, yaitu algoritma genetika akan berhenti setelah mencapai batas generasi yang telah ditentukan.

2.2 Proses Algoritma Fuzzy Evolusi



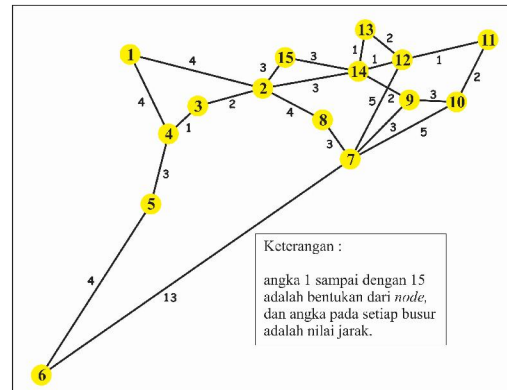
Gambar 3.1 Diagram alir proses algoritma fuzzy evolusi

2.2.1 Alur proses Algoritma Fuzzy Evolusi

1. Tahap Pemasukan Data
2. Tahap Proses Fuzzy
3. Tahap proses populasi awal/inisialisasi populasi
4. Tahap Evaluasi *Fitness*
5. Tahap Kriteria Berhenti
6. Tahap seleksi
7. Tahap *Crossover*
8. Tahap Mutasi
9. Tahap *Elitisme*
10. Perolehan Hasil

4. ANALISIS HASIL PENELITIAN DAN PEMBAHASAN

4.1 Objek Masalah Pengujian



Gambar 4.1 Peta lokasi agen JNE semarang barat yang sudah ditandai dengan bentuk “node”

Contoh 4.1

Contoh ini akan menyelesaikan optimasi / pencarian jalur terpendek pada agen - agen PT. Jalur Nugraha Ekakurir (JNE) Semarang barat menggunakan algoritma genetika dan algoritma fuzzy evolusi.

Ketik “proga” pada MATLAB *command window* dan tekan enter. Maka program akan meminta pengguna untuk memasukan input berupa *node* sumber, *node* tujuan dan parameter-parameter algoritma genetika. Setelah menggunakan program algoritma genetika (proga) gunakan algoritma fuzzy evolusi, untuk menggunkanya terlebih dahulu buka program FUZZY XU. ketik “fuzzy” pada MATLAB *command window* selanjutnya tekan enter, *open file* - lalu pilih *view – rules* (ctrl + 5) setelah itu gunakan inputan populasi dan generasi untuk menghasilkan *ProbCrossover* dan *ProbMutasi* yang digunakan pada masukan “profuzzy”.

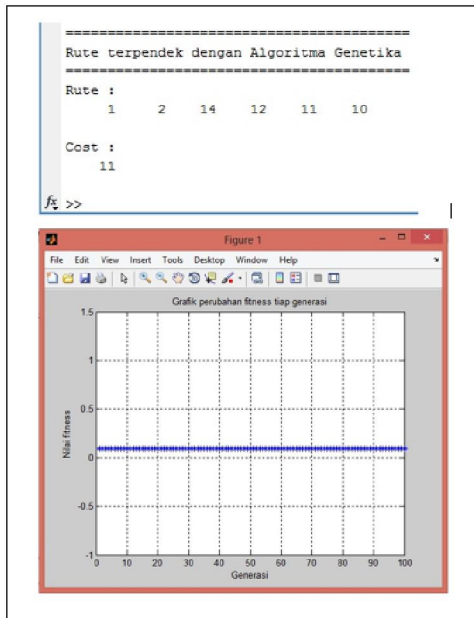
```

Command Window
=====
Masukkan Parameter Program
=====
>> Nama Input File : jne
>> Node sumber : 1
>> Node tujuan : 10
>> Ukuran populasi : 80
>> Probabilitas crossover : 0.45
>> Probabilitas mutasi : 0.01
>> Batas generasi : 100

```

Gambar 4.2.a Tampilan *command window* data masukan algoritma genetika (proga) untuk Contoh 4.1

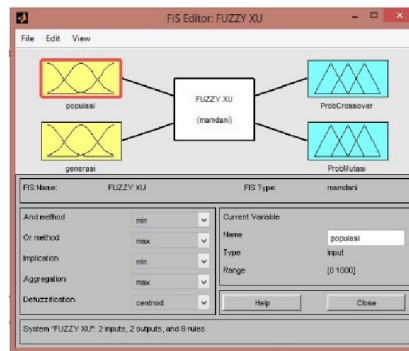
Terdapat input file “jne” yang berisikan data seperti dijelaskan pada Lampiran 2, Adapun nilai parameter yang digunakan untuk masalah ini adalah ukuran populasi 80, probabilitas *crossover* 0.45, probabilitas mutasi 0.01, batas generasi 100, sedangkan *node* sumber 1, dan *node* tujuan 10. Kemudian program mengeluarkan solusi akhir yang didapat dan grafik yang menampilkan perubahan nilai *fitness* terbaik terhadap pertambahan generasi.



Gambar 4.2.b Tampilan keluaran untuk contoh 4.1

Dari Gambar 4.2.b terlihat bahwa solusi untuk Contoh 4.1 adalah jalur 1 - 2 - 14 - 12 - 11 - 10 yang memiliki total Cost 11, dan menghasilkan grafik yang menunjukkan nilai *fitness* terbaik pada tiap generasi.

Dengan ini hasil dari algoritma genetika (proga) menghasilkan solusinya, selanjutnya implementasi pada tool fuzzy kedalam algoritma fuzzy evolusi (profuzzy).



Gambar 4.3.a Tampilan FUZZY XU untuk Contoh 4.1.



Gambar 4.3.b Tampilan input dan output pada FUZZY XU untuk Contoh 4.1.

Seperti yang terlihat pada Gambar 4.3.b terdapat input populasi 80, dan generasi 100 yang sama pada saat penginputan parameter algoritma genetika (proga) dan menghasilkan output probabilitas *crossover* 0.716, dan probabilitas mutasi 0.202 pada FUZZY XU yang akan digunakan pada masukan “profuzzy”.

```

Command Window
=====
Masukkan Parameter Program
=====
>> Nama Input File : jne
>> Node sumber : 1
>> Node tujuan : 10
>> Ukuran populasi : 80
>> Probabilitas crossover : 0.716
>> Probabilitas mutasi : 0.202
>> Batas generasi : 100

=====
Rute terpendek dengan Algoritma Fuzzy Evolusi
=====
Rute :
    1   2   14   12   11   10

Cost :
    11

fx >>

```

Gambar 4.3.c Tampilan *command window* data masukan algoritma fuzzy evolusi (profuzzy) sampai dengan hasilnya untuk Contoh 4.1.

Dari Gambar 4.3.c terlihat bahwa masukan yang sama pada ukuran populasi dan generasi, sedangkan untuk probabilitas *crossover* dan probabilitas mutasi berbeda, hal tersebut dikarenakan masukan probabilitas *crossover* dan probabilitas mutasi dihasilkan dari system FUZZY XU.

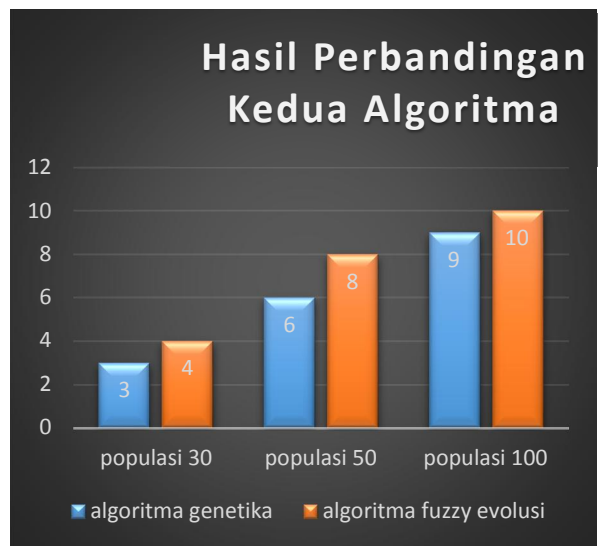
Dan terlihat bahwa solusi untuk Contoh 4.1 adalah jalur 1 - 2 - 14 - 12 - 11 - 10 yang memiliki total Cost 11, dan menghasilkan grafik yang menunjukkan nilai *fitness* terbaik pada tiap generasi, hasil yang sama dengan algoritma genetika(proga) pada gambar 4.1.b.

4.2 Hasil Percobaan

Hasil percobaan dilakukan dengan menggunakan parameter control Probabilitas *crossover* = 0.45, Probabilitas *mutasi* = 0.01, dan mengubah ukuran populasi *nind* yang berubah-ubah. Nilai *nind* yang digunakan adalah 30, 50, dan 100 serta menggunakan batas generasi = 100. Pengujian dilakukan percobaan sebanyak 10 kali.

Program dijalankan pada *Personal Computer* (PC) dengan prosesor Intel Core i3 2,13GHz, memori 2GB dan system operasi windows 8.

Dalam hal ini masalah pengujian ada dua yaitu pengujian algoritma genetika dan pengujian algoritma fuzzy evolusi, pada pengujian algoritma fuzzy evolusi akan menggunakan probabilitas *crossover* dan probabilitas *mutasi* versi algoritma itu sendiri. Selanjutnya hasil percobaan algoritma genetika.



Gambar 4.4 Grafik Hasil Perbandingan Algoritma Genetika dan algoritma fuzzy evolusi

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat dari penelitian ini adalah :

1. Algoritma genetika dan algoritma fuzzy evolusi dapat dibandingkan dengan menggunakan parameter *nind* pada penyelesaian Traveling Salesman Problem.

2. Algoritma fuzzy evolusi lebih unggul dalam hal pencarian solusi optimal dibandingkan dengan algoritma genetika pada penyelesaian Traveling Salesman Problem.

5.2 Saran

Mengingat keterbatasan waktu untuk mengembangkan lebih jauh Tugas Akhir ini, maka saran untuk mengembangkan adalah :

1. Diharapkan adanya perbandingan kedua algoritma genetika dan algoritma fuzzy evolusi selain pencarian solusi optimal.
2. Pada program tidak dapat diberikan tanda koma(,)/titik(.) untuk jarak yang ada pada lampiran 2, sehingga jarak antara agen tidak dapat menggunakan jarak yang sebenarnya.
3. Diharapkan adanya object lain untuk dibandingkan selain traveling salesman problem.
4. Diharapkan Adanya Penambahan Sistem program untuk mengatasi jalan satu arah.

- [2] R. M. Sukaton, "PENGUNAAN ALGORITMA GENETIKA DALAM MASALAH JALUR TERPENDEK PADA JARINGAN DATA," 2011.
- [3] S. Muzid, "PEMANFAATAN ALGORITMA FUZZY EVOLUSI UNTUK PENYELESAIAN KASUS TRAVELING SALESMAN PROBLEM," *Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008)*, 2008.
- [4] I. M. H. F. Fajar Saptono, "PERBANDINGAN PERFORMASI ALGORITMA GENETIKA DAN ALGORITMA SEMUT UNTUK PENYELESAIAN SHORTER PATH PROBLEM," *Seminar Nasional Sistem dan Informatika*, 2007.
- [5] S. Puspitorini, "PENYELESAIAN MASALAH TRAVELING SALESMAN PROBLEM DENGAN JARINGAN SARAF SELF ORGANIZING," Vol.6, no. 39-55, p. No. 1, 2008.
- [6] D. A. Wicaksana, "SOLUSI TRAVELING SALESMAN PROBLEM MENGGUNAKAN ALGORITMA FUZZY EVOLUSI," Semarang, 2013, p. 50.

DAFTAR PUSTAKA

- [1] T. A. Y. Samuel Lukas, "PENERAPAN ALGORITMA GENETIKA UNTUK TRAVELING SALESMAN PROBLEM DENGAN MENGGUNAKAN METODE ORDER CROSSOVER DAN INSERT MUTATION," *Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005)*, 2005.