

## PENERAPAN ALGORITMA GENETIKA UNTUK *TRAVELING SALESMAN* *PROBLEM* DENGAN MENGGUNAKAN METODE *ORDER CROSSOVER* DAN *INSERTION MUTATION*

Samuel Lukas<sup>1</sup>, Toni Anwar<sup>1</sup>, Willi Yuliani<sup>2</sup>

<sup>1</sup>Dosen Teknik Informatika, Universitas Pelita Harapan

<sup>2</sup>Alumni Jurusan Teknik Informatika Universitas Pelita Harapan

Menara UPH Lippo Karawaci Jl. M.H. Thamrin Boulevard 1100, Tangerang

E-mail: sml\_lks@yahoo.com, dan toni\_anwar@yahoo.com

### Abstract

*Traveling Salesman Problem* is one of the most famous optimization problem and genetic algorithm is the most famous optimization problem solver. This paper demonstrates how the genetic algorithms can solve this *Traveling Salesman Problem* effectively. The four variables of genetic algorithm that are number of chromosomes, number of generations, crossover probability and mutation probability are set in such a way to solve the problem.

**Kata kunci:** *Traveling Salesman Problem*, genetic algorithm, optimization

### 1. Pendahuluan

*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 komputasional. 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.

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 genetik. Algoritma genetik 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.

Makalah ini membahas bagaimana algoritma genetik menyelesaikan TSP dengan menggunakan metode *order crossover* sebagai teknik rekombinasi dan metode *insertion mutation* sebagai teknik mutasi yang digunakan pada algoritma genetik. Untuk mengetahui bagaimana penerapan algoritma genetik dalam menyelesaikan *Traveling Salesman Problem*, dibuatkan sebuah program simulasi sederhana dengan menggunakan piranti lunak Microsoft Visual Basic 6.0. Dalam program simulasi tersebut, *Traveling Salesman Problem* yang akan digunakan adalah *Symmetric Traveling Salesman Problem* dimana jarak kota A ke kota B adalah sama dengan jarak kota B ke kota A.

### 2. Algoritma Genetika

Algoritma genetika adalah sebuah teknik optimasi yang berdasarkan pada proses evolusi alam. Di alam, kromosom yang terbaik akan bertahan hidup sehingga generasi berikutnya akan lebih baik karena kromosom pada generasi tersebut diturunkan dari orang tua yang baik pula. Konsep yang sama dikembangkan untuk penyelesaian masalah dengan cara mencari himpunan solusi terbaik yang bertahan hidup dan melakukan rekombinasi solusi yang kurang baik untuk mendapatkan kromosom lain yang lebih baik pada generasi berikutnya.

Proses algoritma genetika terdiri dari beberapa langkah, yaitu pengkodean (*encoding*), seleksi (*selection*), persilangan (*crossover*), mutasi (*mutation*), *decoding*. Pertama-tama, proses *encoding* adalah suatu proses kodifikasi atas solusi dari permasalahannya. Hasil *encoding* adalah berbentuk *string* yang merupakan representasi dari suatu kromosom. Proses *selection* menentukan kromosom mana yang tetap tinggal pada generasi berikutnya. Proses *crossover* akan menghasilkan kromosom baru yang merupakan pengganti dari kromosom yang hilang sehingga total kromosom pada satu generasi berjumlah tetap. Proses *mutation* memungkinkan terjadinya kromosom baru secara *unpredictable*. Proses terakhir adalah *decoding* yaitu mengambil makna dari hasil kromosom terbaik untuk menjawab permasalahannya.

#### 2.1 Teknik Encoding

Proses *encoding* adalah salah satu proses yang sulit dalam algoritma genetika. Hal ini disebabkan karena proses *encoding* untuk setiap permasalahan berbeda-beda karena tidak semua teknik *encoding* cocok untuk setiap permasalahan. Proses *encoding* menghasilkan *string* yang kemudian disebut kromosom. *String* terdiri dari

sekumpulan bit. Bit ini dikenal sebagai gen. Jadi satu kromosom terdiri dari sejumlah gen.

Ada bermacam-macam teknik *encoding* yang dapat dilakukan dalam algoritma genetika. Beberapa teknik-teknik *encoding* itu antara lain adalah *binary encoding*, *permutation encoding*, *value encoding* serta *tree encoding*. Teknik *encoding* yang digunakan pada *Traveling Salesman Problem* adalah *permutation encoding*. Selain digunakan pada *Traveling Salesman Problem*, teknik ini juga dapat digunakan pada *Task Ordering Problem* [1]. Pada *permutation encoding*, kromosom-kromosom adalah kumpulan angka yang mewakili posisi dalam sebuah rangkaian. Pada *Traveling Salesman Problem*, kromosom mewakili urutan kota yang dikunjungi *salesman*. Jadi apabila satu kromosom berbentuk sebagai berikut  $P1 = (X_1, X_2, X_3, \dots, X_n)$  berarti *salesmen* bergerak dari kota bernomor  $X_1$  ke  $X_2$  dst hingga ke kota ke  $X_n$ .

## 2.2 Proses Seleksi

Proses seleksi adalah proses yang memegang peranan penting dalam algoritma genetika. Proses seleksi ini digunakan agar hanya kromosom-kromosom yang berkualitas yang dapat melanjutkan perannya dalam proses algoritma genetika berikutnya. Ada bermacam-macam teknik untuk melakukan proses seleksi pada suatu permasalahan. Teknik seleksi yang akan digunakan tergantung pada permasalahan yang akan diselesaikan. Ada bermacam-macam teknik seleksi, diantaranya adalah *Roulette Wheel Selection*, *Rank Base Selection*, dan *Steady State Selection* [2].

Proses penseleksian pada makalah ini menggunakan teknik peringkat atau *Rank Base Selection*. Pada proses penseleksian digunakan suatu parameter yang disebut kesesuaian atau *fitness*. *Fitness* digunakan untuk menentukan seberapa baik kromosom akan bertahan hidup. Makin tinggi nilai *fitness*,  $0 \leq fitness \leq 1$ , suatu kromosom maka makin baik kromosom itu akan bertahan hidup. Nilai *fitness* tertinggi merepresentasikan jawab terbaik atas persoalan itu sendiri. Penentuan berapa besar nilai *fitness* suatu kromosom berdasarkan fungsi *fitness* yang didefinisikan tersendiri. Untuk makalah ini maka fungsi *fitness* didefinisikan [5] sebagai:

$$fitness = \frac{1}{TotalJarak} \dots\dots\dots(1)$$

Pada *Rank Base Selection*, hanya kromosom yang mempunyai nilai *fitness* yang tinggilah yang dapat bertahan pada generasi berikutnya sebaliknya yang mempunyai nilai *fitness* rendah akan hilang pada generasi berikutnya. Untuk mempertahankan jumlah kromosom tetap pada satu generasi maka perlu dibangkitkan cromosom baru yang merupakan hasil penyilangan dari kromosom yang hidup. Untuk itu dilakukan proses rekombinasi.

## 2.3 Proses Rekombinasi

Proses rekombinasi atau yang lebih dikenal dengan nama proses *crossover* adalah menyilangkan dua kromosom sehingga membentuk kromosom baru yang harapannya lebih baik dari pada induknya. Tidak semua kromosom pada suatu populasi akan mengalami proses rekombinasi. Kemungkinan suatu kromosom mengalami proses rekombinasi didasarkan pada probabilitas *crossover* yang telah ditentukan terlebih dahulu. Probabilitas *crossover* menyatakan peluang suatu cromosom akan mengalami *crossover*. Ada beberapa teknik rekombinasi yang dapat digunakan untuk menyelesaikan *Traveling Salesman Problem*, antara lain adalah *partially mapped crossover* (PMX), *order crossover* dan *cycle crossover*.

Teknik rekombinasi yang digunakan adalah teknik *order crossover*. *Order crossover* (OX) diperkenalkan oleh Davis [3]. Teknik ini diawali dengan membangkitkan dua bilangan acak. Kemudian gen yang berada diantara kedua bilangan acak akan disalin ke *offspring* dengan posisi yang sama. Langkah berikutnya untuk mendapatkan *offspring* pertama adalah mengurutkan gen yang berada pada *parent* kedua dengan urutan gen yang berada pada posisi setelah bilangan acak kedua diikuti dengan gen yang berada pada posisi sebelum bilangan acak pertama dan diakhiri dengan gen yang berada pada posisi diantara kedua bilangan acak. Kemudian gen yang telah diurutkan tersebut dibandingkan dengan *offspring* pertama. Apabila gen tersebut ada pada *offspring* kedua maka abaikan gen tersebut dari urutan itu. Kemudian masukkan urutan yang baru saja didapat pada *offspring* dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu dan sisanya dimasukkan pada posisi sebelum bilangan acak pertama. Begitu juga untuk menghasilkan *offspring* kedua. Contoh *order crossover* adalah sebagai berikut

$$\begin{aligned}
 p1 &= (1\ 2\ 3\ |4\ 5\ 6\ 7\ |8\ 9) \\
 p2 &= (4\ 5\ 2\ |1\ 8\ 7\ 6\ |9\ 3) \\
 \Rightarrow \text{copy the selected segments} \\
 o1 &= (x\ x\ x\ |4\ 5\ 6\ 7\ |x\ x) \\
 o2 &= (x\ x\ x\ |1\ 8\ 7\ 6\ |x\ x) \\
 \Rightarrow \text{the seq of p2 : } &9-3-4-5-2-1-8-7-6 \\
 \text{remove 4, 5, 6, 7 in the seq and put it in o1 the} \\
 \text{remaining seq : } &9-3-2-1-8 \\
 o1 &= (x\ x\ x\ |4\ 5\ 6\ 7\ |x\ x) = (2\ 1\ 8\ |4\ 5\ 6\ 7\ |9\ 3) \\
 \text{construct o2 in the same way:} \\
 o2 &= (3\ 4\ 5\ |1\ 8\ 7\ 6\ |9\ 2)
 \end{aligned}$$

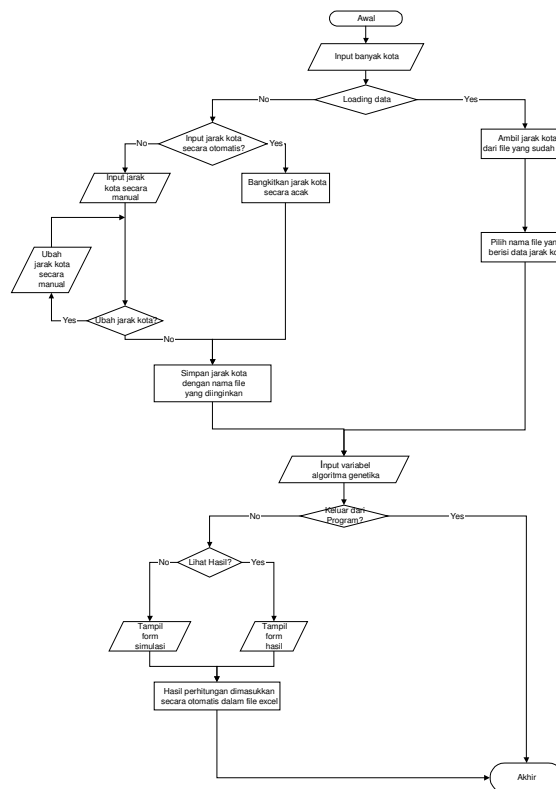
## 2.4 Proses Mutasi

Proses mutasi ini dilakukan setelah proses rekombinasi dengan cara memilih kromosom yang akan dimutasi secara acak, dan kemudian menentukan titik mutasi pada kromosom tersebut secara acak pula. Banyaknya kromosom yang akan

mengalami mutasi dihitung berdasarkan probabilitas mutasi yang telah ditentukan terlebih dahulu [4]. Apabila probabilitas mutasi adalah 100% maka semua kromosom yang ada pada populasi tersebut akan mengalami mutasi. Sebaliknya, jika probabilitas mutasi yang digunakan adalah 0% maka tidak ada kromosom yang mengalami mutasi pada populasi tersebut.

Ada bermacam-macam teknik mutasi yang dapat digunakan untuk menyelesaikan suatu masalah dengan algoritma genetika. Seperti pada teknik rekombinasi, teknik mutasi juga dirancang untuk digunakan pada suatu masalah yang spesifik sehingga tidak setiap teknik mutasi dapat diterapkan pada suatu masalah yang akan diselesaikan. Selain itu, teknik mutasi yang digunakan juga harus sesuai dengan teknik *encoding* yang digunakan untuk menyelesaikan permasalahan tersebut. Beberapa teknik mutasi yang dapat digunakan dalam penyelesaian *Traveling Salesman Problem* adalah *inversion mutation*, *insertion mutation*, dan *reciprocal mutation*.

Teknik mutasi yang digunakan dalam makalah adalah teknik *insertion mutation*. Teknik ini diawali dengan memilih dua bilangan acak kemudian gen yang berada pada posisi bilangan acak pertama ditukar dengan gen yang berada pada bilangan acak kedua.

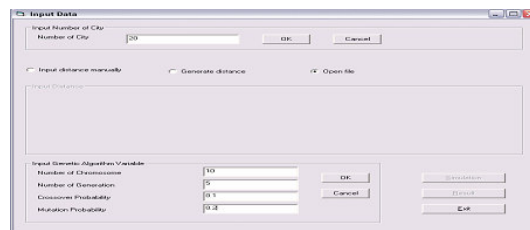


Gambar 1. Flowchart Rancangan Sistem

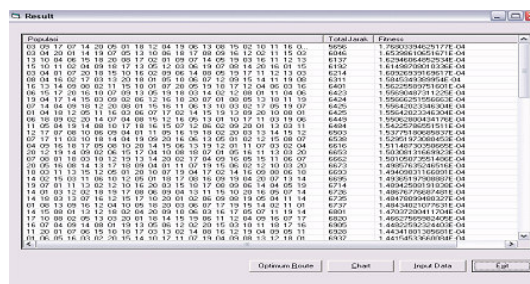
### 3. Rancangan Sistem

Pada sistem ini, kromosom terdiri dari sejumlah gen dimana masing-masing gen tersebut merepresentasikan setiap kota yang dilewati *salesman*. Jumlah gen yang ada pada suatu kromosom merupakan jumlah kota ditambah satu dimana kota terakhir yang dilalui *salesman* sama dengan kota awal yang dilalui *salesman* sehingga dapat dikatakan bahwa kromosom merupakan rute yang mungkin dilalui oleh *salesman*.

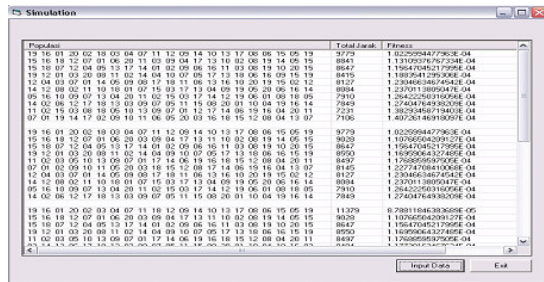
Sistem ini dibagi menjadi tiga bagian *input*, yaitu bagian *input* jumlah kota, bagian *input* jarak antar kota dan bagian *input* variabel algoritma genetika. Sedangkan bagian *output* pada sistem ini dibagi menjadi dua bagian utama, yaitu menu hasil dan menu simulasi. *Flowchart* rancangan sistem dapat dilihat pada gambar 1 sedangkan rancangan menu *input*, menu hasil serta menu simulasi dapat dilihat pada gambar 2, gambar 3 dan gambar 4.



Gambar 2. Menu Input



Gambar 3. Menu Hasil



Gambar 4. Menu Simulasi

#### 4. Hasil Percobaan dan Batasan

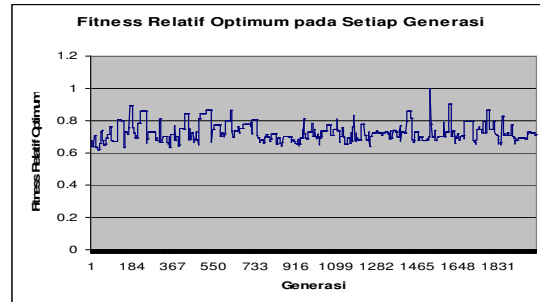
Percobaan pada sistem ini dilakukan untuk menguji keandalan sistem dan bagaimana pengaruh parameter algoritma genetik terhadap masalah TSP. Untuk itu dibuatkan matrik jarak simetris antar 20 kota yang diperlihatkan pada gambar 5 berikut.

Gambar 5. Matriks yang berisi jarak antar kota yang digunakan dalam percobaan

##### 4.1 Mekanisme Sistem

Percobaan ini untuk melihat bagaimana keandalan sistem untuk mencari jawab TSP atas 20 kota dengan parameter jumlah kromosom untuk satu generasi adalah 2000 dan berlangsung selama 2000 generasi dengan probabilitas rekombinasi dan mutasinya masing-masing sebesar 0.1. Hasil percobaan ini diperlihatkan pada gambar 6 yang merupakan grafik *fitness* relative optimum setiap generasi.

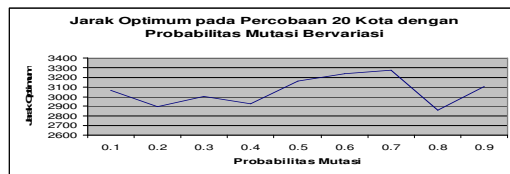
Dari percobaan terlihat bahwa fitness optimum setiap generasi berbeda. Terlihat bahwa fitness relative optimum maksimum yaitu sama dengan satu terdapat pada generasi ke 1524 dengan total jarak minimum sebesar 3066. Dari percobaan ini sudah terlihat bahwa sistem bekerja dengan baik. Akan tetapi apakah ini solusi terbaik masih memerlukan percobaan lanjutan sebagai berikut.



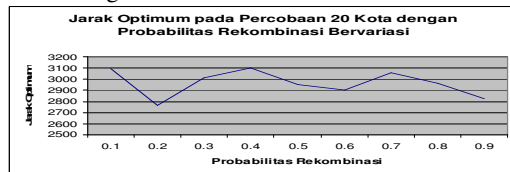
Gambar 6. Grafik *Fitness* Relatif Optimum pada Percobaan Pertama

##### 4.2 Pengaruh Rekombinasi dan Mutasi

Pada percobaan ini ingin didapat jawab terbaik dari masalah TSP diatas serta pengaruh dari probabilitas rekombinasi dan probabilitas mutasi. Ada dua percobaan yang dilakukan. Percobaan pertama, mengubah variabel probabilitas rekombinasi dari 0.1 hingga 0.9 dengan step 0.1 dengan variabel lainnya dipegang tetap. Setelah mendapat nilai probabilitas rekombinasi terbaik yang optimum maka variabel probabilitas mutasi diubah dari 0.1 sampai dengan 0.9 dengan step 0.1 akan tetapi variabel lainnya dipegang tetap. Dari percobaan ini didapat bahwa Probabilitas rekombinasi terbaik adalah 0.8 dengan probabilitas mutasinya 0.1 menghasilkan solusi terbaik yaitu dengan total jarak minimumnya adalah 2381 pada generasi ke 872. Percobaan kedua, seperti percobaan pertama tetapi dengan urutan perubahan yang berbeda. Pada percobaan kedua ini, maka variabel bebas pertamanya adalah probabilitas mutasi yang kemudian diikuti dengan variabel bebas keduanya yaitu probabilitas rekombinasi. Hasil dari percobaan kedua ini didapat probabilitas rekombinasi terbaik adalah 0.2 dengan probabilitas mutasinya 0.8 menghasilkan solusi terbaik yaitu dengan total jarak minimumnya adalah 2766 pada generasi ke 1833. Terlihat bahwa hasilnya kurang baik dibandingkan dengan percobaan pertama. Hal ini disebabkan dengan makin tingginya probabilitas mutasi maka kromosom baru yang terbentuk relative menjadi makin tak terkendalikan sehingga kromosom terbaik dari suatu generasi mungkin saja berubah secara tak terduga. Gambar 7 memperlihatkan grafik hasil percobaan kedua.



Grafik Jarak Optimum vs Probabilitas Mutasi dengan Probabilitas rekombinasi = 0.1



Grafik Jarak Optimum vs Probabilitas Rekombinasi dengan Probabilitas Mutasi = 0.8

**Gambar 7.** Pengaruh Perubahan Probabilitas Mutasi Kemudian Probabilitas Rekombinasi .

### 4.3 Pengaruh Generasi dan Kromosom

Percobaan ini dilakukan dengan mengubah jumlah generasi mulai dari 1000 hingga 4000 dengan step 500 dan probabilitas rekombinasi dan mutasinya dipegang tetap yaitu 0.8 dengan jumlah kromosom 2000. Maka hasil terbaiknya dengan total jarak minimumnya adalah 2564 pada generasi ke 1374 dari 1500 generasi. Apabila jumlah kromosom diubah dari 1000 hingga 4000 dengan step 500 dan probabilitas rekombinasi maupun mutasi dipegang tetap 0.8 dan jumlah generasi adalah 1500 maka didapat total jarak minimumnya adalah 2666 pada generasi ke 1205 dengan jumlah kromosom sebanyak 2500 pada setiap generasi.

Terlihat bahwa pengaruh jumlah kromosom dan jumlah generasi terlihat kurang berarti dalam sistem. Padahal secara teoritis bahwa makin besar hasil perkalian jumlah kromosom dengan jumlah generasi maka peluang terciptanya kromosom terbaik menjadi semakin besar. Tetapi pada percobaan terlihat tidaklah demikian ini dikarenakan probabilitas mutasi yang besar. Apabila percobaan dilakukan dengan konfigurasi parameter jumlah Kromosom = 2000, jumlah Generasi = 4000, probabilitas rekombinasi = 0.7 dan probabilitas mutasi = 0.1 maka total jarak minimumnya 2201 dihasilkan pada generasi ke 3088. Ini adalah jawab terbaik.

## 5. Kesimpulan

Berdasarkan penelitian yang dilakukan penulis dengan cara melakukan percobaan yang dilakukan secara berulang-ulang maka penulis dapat mengambil beberapa kesimpulan dari hasil percobaan yang didapat, yaitu:

- Perkalian antara variabel jumlah kromosom dan jumlah generasi harus menghasilkan suatu bilangan yang besar.

- Penentuan probabilitas rekombinasi terlebih dahulu yang diikuti dengan penentuan probabilitas mutasi akan menghasilkan hasil yang lebih optimum dibandingkan dengan penentuan probabilitas mutasi terlebih dahulu yang diikuti dengan penentuan probabilitas rekombinasi.
- Hasil yang optimum belum tentu didapat pada akhir generasi pada setiap proses algoritma genetika.

## Daftar Pustaka

- Encoding, <http://cs.felk.cvut.cz/~xobitko/ga/encoding.html>
- Selection, <http://cs.felk.cvut.cz/~xobitko/ga/selection.html>
- Fox, Roland dan Steven Tseng. "Traveling Salesman Problem by Genetic Algorithm and Simulated Annealing", 1997. <http://www.cs.nthu.edu.tw/~jang/courses/cs5611/project/2/#sec#1>.
- Parameters of GA, <http://cs.felk.cvut.cz/~xobitko/ga/params.html>
- P, Irving Vitra. *Perbandingan Metode-Metode dalam Algoritma Genetika untuk Traveling Salesman Problem*. Yogyakarta. Seminar Nasional Aplikasi Teknologi Informasi. 2004