SKRIPSI

PENCARIAN SOLUSI FAMILY TRAVELING SALESMAN PROBLEM DENGAN ALGORITMA GENETIK

KRISOGONUS FERDIE RENDRAGRAHA

NPM: 2014730022

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2019

UNDERGRADUATE THESIS

SEARCHING FOR SOLUTIONS FOR FAMILY TRAVELING SALESMAN PROBLEMS WITH GENETIC ALGORITHMS

KRISOGONUS FERDIE RENDRAGRAHA

NPM: 2014730022

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2019

LEMBAR PENGESAHAN

PENCARIAN SOLUSI FAMILY TRAVELING SALESMAN PROBLEM DENGAN ALGORITMA GENETIK

KRISOGONUS FERDIE RENDRAGRAHA

NPM: 2014730022

Bandung, 11 Desember 2019

Menyetujui,

Pembimbing Tunggal

Kristopher David Harjono M.T.

Ketua Tim Penguji	Anggota Tim Penguji
Luciana Abednego, M.T.	**Elisati Hulu, M.T.**

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENCARIAN SOLUSI FAMILY TRAVELING SALESMAN PROBLEM DENGAN ALGORITMA GENETIK

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 11 Desember 2019

Meterai

Krisogonus Ferdie Rendragraha
NPM: 2014730022

ABSTRAK

Algoritma genetik adalah algoritma berbasis populasi dengan melakukan pendekatan secara biologis. Strategi evolusi dan pemrograman ini menggunakan operasi mutasi sebagai cara untuk mencari solusi. Solusi potensial (atau bisa disebut dengan kromosom) dapat direpresentasikan sebagai rangkaian bit. Selain mutasi, algoritma ini juga menggunakan operasi crossover (persilangan) untuk mencari solusi baru (kromosom baru).

Family Traveling Salesman Problem (FTSP) dimotivasi dari aplikasi pengambilan pesanan di gudang. Meski pekerjaan dilakukan dengan desain dan kontrol gudang, pemesanan biasanya dimodelkan menggunakan *Travelling Salesman Problem*, di mana produk serupa dapat disimpan bersama. Karenanya, produk serupa dapat disimpan secara terpisah.

Pada skripsi ini, sebuah perangkat lunak dibangun untuk mengimplementasikan secara bruteforce dan algoritma genetik. Input yang dipakai pada perangkat lunak ini adalah plaintext dan input yang diberikan oleh pengguna dengan *keyboard*. Setelah perangkat lunak dibangun, pengujian dilakukan terhadap implementasi secara bruteforce dan algoritma genetik.

Perangkat lunak diuji dengan 2 metode pengujian, yaitu pengujian fungsional dan eksperimental. Berdasarkan pengujian fungsional, dapat disimpulkan bahwa perangkat lunak yang dibangun mampu mengimplementasikan proses bruteforce dan algoritma genetik dengan baik dan juga mampu menghasilkan kromosom yang sesuai dengan ketentuan pengambilan berapa banyak kota dalam setiap *family*. Berdasarkan pengujian eksperimental, dapat disimpulkan bahwa parameter input yang diberikan oleh pengguna perangkat lunak yaitu banyaknya kromosom yang dibentuk dalam 1 populasi, *mutation rate*, iterasi, dan *treshold* dapat menentukan dan mempengaruhi hasil pencarian kromosom dengan jarak tempuh minimum.

Kata-kata kunci: Algoritma genetik, *Family Traveling Salesman Problem*, bruteforce, crossover, mutasi, selection

ABSTRACT

Genetic algorithm is an algorithm based on participation by conducting biological evaluations. This evolutionary and programming strategy uses mutation operations as a way to find solutions. Potential solutions (called chromosomes) can be represented as a series of bits. In addition to mutations, this algorithm also uses crossover operations (crosses) to find new solutions (new chromosomes).

Family Traveling Salesman Problem (FTSP) is motivated from the application of taking orders at the warehouse. Although the work is done with warehouse design and control, orders are usually modeled using the Traveling Salesman Problem, where similar products can be stored together. Therefore, similar products can be stored separately.

In this thesis, a software is built to implement bruteforce and genetic algorithm. Inputs used in this software are plaintext and input provided by the user with the keyboard. After the software is built, testing is carried out on the implementation of bruteforce and genetic algorithms.

The software is tested with 2 testing methods, namely functional and experimental testing. Based on functional testing, it can be concluded that the software built is able to implement bruteforce processes and genetic algorithms well and is also able to produce chromosomes that are in accordance with the provisions on how many cities in each family. Based on experimental testing, it can be concluded that the input parameters given by software users are the number of chromosomes formed in 1 population, mutation rate, iteration, and treshold can determine and influence the results of chromosome searches with minimum mileage.

Keywords: Genetic algorithm, Family Traveling Salesman Problem, bruteforce, crossover, mutation, selection

Dipersembahkan untuk keluarga, diri sendiri, pembimbing, para sahabat dan semua orang yang telah berperan dalam pembuatan skripsi ini.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya penulis berhasil menyelesaikan penyusunan skripsi yang berjudul "Menyelesaikan Family Traveling Salesman Problem dengan Algoritma Genetik". Penulis menyadari bahwa penyelesaian penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

- Kedua orang tua penulis yang selalu memberikan movitasi, dukungan, serta kepercayaan kepada penulis.
- Ibu Natalia S.Si., M.Si. selaku dosen pembimbing atas bimbingan, dukungan, dan kesabarannya selama proses penyusunan skripsi ini.
- Seluruh rekan seperjuangan dari Jurusan Teknik Informatika UNPAR yang telah menjadi teman dan sahabat penulis selama masa perkuliahan.
- Pihak lain yang tidak dapat disebutkan satu-persatu yang telah memberikan bantuan dan dukungan dalam proses penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan memohon maaf apabila terdapat kesalahan dan kekurangan pada skripsi ini. Semoga skripsi ini bermanfaat bagi pembaca yang sedang meneliti atau mempelajari topik yang berkaitan dengan skripsi ini.

Bandung, Desember 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Travelling Salesman Problem (TSP) diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota[1][2]. Seorang pedagang akan diberikan informasi mengenai sejumlah kota dan jarak antar kota. Pedagang harus menentukan rute terpendek yang harus dilalui untuk mengunjungi seluruh kota. Pedagang akan berangkat dari sebuah kota asal dan kembali lagi ke kota asal keberangkatan. Pedagang akan menyinggahi setiap kota tepat satu kali.

Family Traveling Salesman Problem (FTSP) merupakan pengembangan dari *Traveling Salesman Problem* (TSP). FTSP dimotivasi dari aplikasi pengambilan pesanan yang barada di gudang [3]. Pemesanan biasanya dimodelkan dengan menggunakan *Travelling Salesman Problem* (TSP). Dalam hal ini, setiap gudang menyimpan produk yang serupa dan berada di beberapa kota yang berbeda. Jika setiap produk adalah *family* dan jumlah anggota *family* yang ingin dikunjungi adalah permintaan dari produk yang terkait dengan *family* tersebut, maka pengambilan pesanan masalah di gudang dapat dilihat sebagai FTSP.

Perbedaan FTSP dengan TSP adalah kota akan dibagi menjadi beberapa kelompok kota. Setiap kelompok kota ini disebut *family*. Setiap kota dalam *family* tidak akan dikunjungi seluruhnya tetapi akan dipilih kota mana saja yang akan dikunjungi. Sebagai contoh, terdapat sebuah kota asal dan n buah kota yang akan dikunjungi. Sekumpulan kota yang akan dikunjungi dibagi menjadi beberapa kelompok kota yang disebut *family*. Kemudian akan ditentukan rute yang memenuhi jumlah kota yang ingin dikunjungi dalam masing - masing *family*. Dari rute tersebut, akan dicari jarak tempuh yang seminimal mungkin.

Untuk mencari rute dengan jarak tempuh minimum, dapat dilakukan dengan membuat semua kemungkinan solusi yang ada menggunakan algoritma *bruteforce*. Dalam algoritma *bruteforce*, masalah diselesaikan dengan cara yang paling mudah berdasarkan pernyataan masalah dan definisi istilah dalam FTSP seperti definisi kota, jarak antar kota, dan *family*. Algoritma *bruteforce* dirancang untuk memecahkan masalah tanpa memperhatikan sumber daya komputasi yang diperlukan. Sebagai contoh, dalam konteks FTSP, solusi untuk permasalahan FTSP dapat ditemukan dengan membuat setiap solusi yang mungkin menggunakan kombinasi dan permutasi dengan mengikuti ketentuan dan aturan dari *family*. Dari semua solusi yang terbentuk, akan dicari solusi terbaik dari seluruh solusi yang ada.

Algoritma *bruteforce* dapat menemukan solusi ketika inputnya tidak terlalu besar. Tetapi ketika inputnya terlalu besar maka algoritma *bruteforce* tidak dapat dilakukan karena membutuhkan sum-

ber daya komputasi yang besar dan waktu proses yang lama. Oleh karena itu, diperlukan algoritma heuristik yang dapat menghasilkan solusi dalam waktu proses yang singkat. Solusi ini mungkin bukan yang terbaik dari semua solusi yang ada atau hanya perkiraan solusi yang tepat. Tetapi hal itu masih dapat digunakan karena tidak memerlukan waktu yang lama dan dapat menerima input yang besar.

Salah satu dari algoritma heuristik adalah algoritma genetik. Algoritma genetik adalah kelas khusus dari algoritma evolusioner dengan menggunakan teknik yang terinspirasi oleh prinsip biologi[1][2]. Prinsip ini didasarkan pada semua makhluk hidup yang terdiri dari gen. Gen - gen ini akan menjadi pembentuk kromosom. Kromosom ini yang dijadikan landasan untuk algoritma genetik. Solusi yang potensial didefinisikan ke dalam kromosom dan elemen individu dari solusinya adalah gen.

Dalam konteks FTSP, setiap kota didefinisikan sebagai gen. Setiap kromosom mengkodekan solusi untuk masalah yaitu rute perjalanan. Nilai fitnes kromosom terkait dengan panjang perjalanan, tergantung pada tujuan kota yang akan dikunjungi. Terdapat beberapa operasi yang dilakukan dalam algoritma genetik agar dapat menemukan kromosom dengan nilai fitnes minimum. Pertama, operasi selection. Operasi ini mengambil dua kromosom untuk dijadikan sebagai *parent*. Kedua kromosom ini akan dipakai untuk membentuk kromosom baru. Kedua, operasi persilangan (*crossover*) dan mutasi. Dari kedua kromosom yang dipilih, akan dilakukan operasi persilangan dan mutasi untuk menghasilkan keturunan baru (kromosom baru) yang akan mewarisi gen - gen dari *parent*. Ketiga, operasi *generation replacement*. Setiap kromosom baru yang terbentuk akan dimasukkan ke dalam populasi. Kromosom ini akan menggantikan kromosom yang memiliki nilai fitnes yang kurang baik.

Untuk dapat menemukan solusi potensial dari permasalahan FTSP ini, dibuat sebuah perangkat lunak dengan menerapkan algoritma genetik. Perangkat lunak ini membutuhkan input *plaintext*. Perangkat lunak ini juga membutuhkan input dari pengguna perangkat lunak, yaitu jumlah kromosom dalam sebuah populasi, jumlah iterasi, *mutation rate*, dan *treshold*. Untuk memastikan perangkat lunak dapat berjalan dengan baik, maka dilakukan pengujian perangkat lunak. Pengujian dilakukan dengan menggunakan dua metode, yaitu pengujian fungsional dan pengujian eksperimental. Pengujian fungsional dilakukan dengan cara membandingkan hasil proses yang dilakukan antara perangkat lunak dengan implementasi manual. Sedangkan, pengujian eksperimental dilakukan terhadap perangkat lunak yang mengimplementasikan algoritma genetik.

1.2 Rumusah Masalah

Berdasarkan latar belakang permasalahan maka dapat dirumuskan bahwa :

1. Bagaimana memodelkan permasalahan *Family Traveling Salesman Problem* dengan menggunakan algoritma genetik?

2. Bagaimana cara untuk menemukan solusi potensial dari permasalahan *Family Traveling Salesman Problem* dengan menerapkan algoritma genetik?

3. Bagaimana kinerja algoritma genetik dalam pencarian solusi permasalahan *Family Traveling Salesman Problem*?

1.3 Tujuan

Berdasarkan rumusan masalah yang ada, maka tujuan dari penelitian ini adalah :

1. Memodelkan permasalahan *Family Traveling Salesman Problem* dalam algoritma genetik.
2. Membuat sebuah perangkat lunak yang dapat menemukan solusi potensial dari permasalahan *Family Traveling Salesman Problem* yang menerapkan algoritma genetik dengan menggunakan bahasa pemrograman *Java*.
3. Mengukur kinerja pemodelan algoritma genetik dalam mencari rute terpendek dari permasalahan *Family Traveling Salesman Problem*.

1.4 Metodologi Penelitian

Langkah - langkah yang akan dilakukan dalam melakukan penelitian ini adalah :

1. Melakukan studi literatur mengenai teori - teori graf dan Traveling Salesman Problem.
2. Melakukan studi literatur mengenai *Family Traveling Salesman Problem.*
3. Melakukan studi literatur mengenai algoritma genetik.
4. Memodelkan permasalahan *Family Traveling Salesman Problem* ke dalam algoritma genetik.
5. Mengimplementasikan algoritma genetik dengan menggunakan kode program *Java*.
6. Melakukan pengujian perangkat lunak dengan melakukan dua metode pengujian, yaitu pengujian fungsional dan pengujian eksperimental.

1.5 Sistematika Pembahasan

Sistematika dalam penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dijalankan. Sistematika penelitian ini adalah sebagai berikut :

1. Bab Pendahuluan
   Bab ini berisi latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian, yang kemudian diikuti dengan deskripsi perangkat lunak, asumsi, serta sistematika pembahasan.
2. Bab Dasar Teori
   Bab ini berisi teori - teori dasar tentang graf dan Traveling Salesman Problem serta teori - teori dasar mengenai Algoritma Genetik.
3. Bab Analisis
   Bab ini berisi analisis masalah, analisis perangkat lunak, dan analisis data yang dibutuhkan oleh perangkat lunak.

4. Bab Perancangan dan Implementasi
   Bab ini perisi perancangan perangkat lunak yang akan dibangun yang meliputi kebutuhan masukan dan keluaran perangkat lunak, perancangan antarmuka, diagram kelas, implementasi antarmuka perangkat lunak, pengujian fungsional perangkat lunak, dan pengujian eksperimental perangkat lunak.

5. Bab Kesimpulan dan Saran
   Bab ini berisi kesimpulan berdasarkan penelitian yang telah dilakukan serta saran bagi pengembangan selanjutnya.