

SKRIPSI

**PENYELESAIAN *FAMILY TRAVELING SALESMAN*
PROBLEM DENGAN ALGORITMA *BEE COLONY*
*OPTIMIZATION***



Yonathan Kristian Purnama

NPM: 2015730016

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

UNDERGRADUATE THESIS

**SOLVING FAMILY TRAVELING SALESMAN PROBLEM
USING BEE COLONY OPTIMIZATION ALGORITHM**



Yonathan Kristian Purnama

NPM: 2015730016

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

LEMBAR PENGESAHAN

**PENYELESAIAN *FAMILY TRAVELING SALESMAN*
PROBLEM DENGAN ALGORITMA *BEE COLONY*
*OPTIMIZATION***

Yonathan Kristian Purnama

NPM: 2015730016

Bandung, 22 Mei 2019

Menyetujui,

Pembimbing

Elisati Hulu, M.T.

Ketua Tim Penguji

Anggota Tim Penguji

Mariskha Tri Adithia, P.D.Eng

Raymond Chandra Putra, 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:

PENYELESAIAN *FAMILY TRAVELING SALESMAN PROBLEM* DENGAN ALGORITMA *BEE COLONY OPTIMIZATION*

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 22 Mei 2019

Meterai Rp. 6000

Yonathan Kristian Purnama
NPM: 2015730016

ABSTRAK

Pada penulisan karya ilmiah ini akan dibahas mengenai penyelesaian *Family Traveling Salesman Problem*. *Family Traveling Salesman Problem* adalah permasalahan yang melibatkan n buah node yang terbagi dalam f buah *family*. Dalam tiap *family* terdapat v buah node yang harus dikunjungi. Hasil akhir dari permasalahan ini adalah sebuah jalur yang mengunjungi seluruh v node yang ada dalam tiap f yang memiliki panjang jalur terkecil. Algoritma *bee colony optimization* digunakan untuk menyelesaikan permasalahan tersebut.

Algoritma *Bee Colony Optimization* digunakan dalam penelitian ini karena algoritma ini cocok untuk menyelesaikan permasalahan pencarian jalur. Algoritma ini juga mengikuti cara kerja dari koloni lebah. Cara kerjanya adalah lebah akan mengikuti lebah lain yang memiliki informasi dari jarak, dan kualitas makanan paling baik. Pada algoritma ini akan digunakan nilai preferensi untuk menentukan apakah suatu tujuan akan dipilih. Untuk beberapa kasus dengan masukan parameter yang tepat algoritma ini akan menghasilkan jalur yang dianggap optimal, sedangkan untuk kasus-kasus yang cukup rumit algoritma ini akan menghasilkan jalur yang kurang baik tetapi masih memiliki nilai error yang cukup kecil. Eksperimen yang akan dilakukan adalah dengan mengganti nilai parameter yang ada pada perangkat lunak.

Berdasarkan hasil dari pengujian yang telah dilakukan dengan mengganti nilai-nilai parameter yang ada. Dari hasil pengujian eksperimental untuk menguji apakah nilai probabilitas sebuah *node* akan dikunjungi berpengaruh pada panjang jalur yang dihasilkan. Didapatkan hasil bahwa jika nilai probabilitas pemilihan *node* digunakan, maka dihasilkan jalur yang lebih pendek. Sedangkan untuk pengaruh nilai jarak dari satu *node* ke *node* lain didapatkan jika besar nilai jarak yang digunakan sebesar 80% sampai 100% akan berpengaruh pada hasil panjang jalur yang didapatkan. Untuk pengujian terhadap nilai probabilitas terbesar yang digunakan didapatkan hasil jika besar nilai probabilitas yang dimasukkan berkisar antara 60% sampai 80% akan menghasilkan jalur dengan panjang yang lebih kecil, lalu untuk pengujian nilai error didapatkan hasil untuk kasus dengan banyak node 10 perangkat lunak memiliki nilai error rata-rata sebesar 22%. Sedangkan untuk kasus dengan banyak node 20 didapatkan rata-rata nilai error sebesar 19%, untuk kasus dengan banyak node 50 sampai 200 nilai error yang didapat berkisar antara 4% sampai 3,5%.

Kata-kata kunci: Algoritma *Bee Colony Optimization*, *Family Traveling Salesman Problem*, Pencarian jalur terdekat

ABSTRACT

This thesis will be discussed about solving Family Traveling Salesman Problem. Family Traveling Salesman Problems are problems that involve a node that is divided into several families in each family that has a node that must be visited. The end result of this problem is a path that visits all the methods in each where the length of the path produced is the path with the smallest path length. Bee Colony Optimization algorithm will be used to solve these problems.

Bee Colony Optimization Algorithm will be used in this study because the algorithm is a local search algorithm. The workings of this algorithm are to talk about the work of a bee colony, where bees will follow other bees that have information from distance, and the best quality of food. In this algorithm you will use the value of arc fitness to determine whether a destination will be selected.

Based on the results of the tests that have been carried out by replacing existing parameter values. From the results of experimental testing to test whether the probability value of a method visited will affect the length of the path produced. The results obtained are that if the probability value of the selection is used, a shorter path will be generated. While the effect of the distance value from the model model is obtained if the distance value used at 80% to 100% will affect the result of the length of the path obtained. To test the greatest probability values, it is obtained that if the probability values range from 60% to 80% will result in paths with smaller lengths, then for testing error values for cases with many nodes 10 software has an average error value of 22% . Whereas for cases with many node 20 the average error value is 19%, for cases with many nodes 50 to 200 the error values range from 4% to 3.5%.

Keywords: Bee Colony Optimization Algorithm, Family Traveling Salesman Problem, Shortest path problem

*Dipersembahkan untuk Tuhan Yang Maha Esa, Keluarga, dan
Dosen pembimbing*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat, rahmat, dan perlindungan-Nya, penulis dapat menyelesaikan penulisan skripsi ini yang berjudul "*Menyelesaikan Family Traveling Salesman Problem dengan Algoritma Bee Colony Optimization*". Penulis ingin berterima kasih kepada:

- Orang tua dan kerabat yang telah memberikan dukungan kepada penulis baik secara moral dan financial selama penulis mengerjakan skripsi.
- Ibu Natalia.M.Si. selaku dosen yang telah memberikan bantuan kepada penulis selama proses pengerjaan skripsi.
- Dosen pembimbing, Bapak Elisati Hulu, M.T. yang memberikan bimbingan, masukan, dan tambahan wawasan selama proses pembuatan skripsi ini.
- Rekan-rekan Vincent Joel Sinatra, Raymond Naga Wijaya, Yosua Halim, Tedi Tri, Aditya Putra, Kevin Tjoe, Staf Pusat Pengembangan Karir UNPAR, yang telah membantu penulis mengembangkan keahlian penulis di bidang IT dan memberikan dukungan kepada penulis untuk menyelesaikan skripsi ini.
- Staf tata usaha FTIS yang sudah membantu proses perkuliahan penulis selama menimba ilmu di jurusan teknik informatika UNPAR.

Semoga seluruh pihak yang membantu dalam penyusunan skripsi ini mendapat berkat dari Tuhan Yang Maha Esa. Akhir kata, penulis memohon maaf apabila terdapat kesalahan dan kekurangan pada skripsi ini. Semoga skripsi ini dapat berguna bagi semua pihak yang membutuhkan.

Bandung, Mei 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Graf	5
2.1.1 Definisi Graf	5
2.1.2 Terminologi Graf	6
2.1.3 Jenis-jenis Graf	7
2.1.4 Representasi Graf	10
2.2 <i>Traveling Salesman Problem</i>	11
2.3 <i>Family Traveling Salesman Problem</i>	12
2.4 <i>Bee Colony Optimization</i>	12
2.4.1 <i>Bee Colony</i>	12
2.4.2 Model BCO	13
2.4.3 BCO untuk TSP	13
3 ANALISIS MASALAH	17
3.1 Analisis Penyelesaian Masalah	17
3.2 Studi Kasus	19
3.2.1 Pemodelan Algoritma penyelesaian FTSP	21
3.3 Analisis Kebutuhan Perangkat Lunak	24
3.3.1 Pemodelan Kelas	25
3.3.2 Cara Kerja Perangkat Lunak	28
4 PERANCANGAN PERANGKAT LUNAK	29
4.1 Perancangan Tampilan Antarmuka	29
4.1.1 Halaman Antarmuka	29
4.2 Diagram Kelas Perangkat Lunak	30
4.2.1 Kelas FXMLDocumentControler	31
4.2.2 Kelas <i>Bee Colony</i>	32

4.2.3	<i>Bee</i>	35
4.2.4	Graf	36
4.2.5	<i>Family</i>	38
4.2.6	Node	39
5	IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	41
5.1	Implementasi Perangkat Lunak	41
5.1.1	Implementasi	41
5.2	Pengujian Fungsional dan Eksperimental	43
5.2.1	Pengujian Fungsional	44
5.2.2	Pengujian Eksperimental	45
6	KESIMPULAN	55
6.1	Kesimpulan	55
6.2	Saran	55
	DAFTAR REFERENSI	57
	A KODE PROGRAM	59
	B HASIL PENGUJIAN EKSPERIMEN NILAI α	71
	C HASIL PENGUJIAN EKSPERIMEN NILAI β	73
	D HASIL PENGUJIAN EKSPERIMEN NILAI λ	75
	E TEST CASE DALAM PENGUJIAN YANG DILAKUKAN	77

DAFTAR GAMBAR

2.1	Graf sederhana dengan enam buah simpul dan delapan buah sisi	5
2.2	graf ketetanggaan	6
2.3	Graf berbobot	7
2.4	Contoh graf sisi ganda	7
2.5	Contoh graf sisi gelang	8
2.6	Contoh graf berarah	9
2.7	Contoh graf berarah berisi dan bersisi ganda	9
2.8	Contoh graf berarah dan memiliki sisi gelang	9
2.9	Contoh macam-macam graf lengkap	10
2.10	Contoh graf adjacency	10
2.11	Gambar graf-4	10
2.12	Gambar graf TSP	11
2.13	Gambar solusi FTSP	12
3.1	contoh <i>flowchart-1</i>	18
3.2	Contoh permasalahan FTSP	20
3.3	Graf permasalahan TSP	22
3.4	Graf permasalahan FTSP dengan 1 depot 2 keluarga, $keluarga_1 = \{2, 3\}$, $keluarga_2 = \{4, 5, 6\}$ dengan $v_1 = 1$ dan $v_2 = 2$	22
3.5	Contoh jalur pertama permasalahan FTSP	23
3.6	Contoh isi <i>file.txt</i>	25
4.1	Contoh antarmuka sistem	29
4.2	Diagram kelas rinci dari perangkat lunak	30
4.3	Diagram kelas FXMLDocumentControler	31
4.4	Diagram Kelas <i>Bee Colony</i>	32
4.5	Diagram Kelas <i>Bee</i>	35
4.6	Diagram Kelas Graf	36
4.7	Diagram Kelas <i>Family</i>	38
4.8	Diagram Kelas Node	39
5.1	Halaman Awal dari Perangkat Lunak	41
5.2	Halaman saat mengklik <i>button choose</i>	42
5.3	Halaman saat berhasil memilih file	42
5.4	Halaman saat perogram selesai melakukan pencarian rute	42
5.5	Contoh file test yang diuji untuk pengujian fungsional	43
5.6	Diagram batang hasil Tabel 5.14, pengujian BCO dibandingkan dengan <i>Brute force</i> untuk test10 (10 node)	49
5.7	Diagram batang hasil Tabel 5.15, pengujian BCO dibandingkan dengan <i>Brute force</i> untuk test20 (20 node)	50
5.8	Diagram batang hasil Tabel 5.16 pengujian BCO dibandingkan dengan hasil terkecil untuk pengujian test50 (50 node)	51

5.9	Diagram batang hasil Tabel 5.17, pengujian BCO dibandingkan dengan jalur terkecil yang ditemukan untuk test100 (100 node)	52
5.10	Diagram batang hasil Tabel 5.18, pengujian BCO dibandingkan dengan hasil terkecil yang ditemukan untuk test200 (200 node)	53

DAFTAR TABEL

2.1	<i>adjacency list table</i>	10
2.2	Table probabilitas P_{follow}	15
3.1	Contoh jalur dan jarak yang dihasilkan dari Gambar 3.2	20
3.2	Table probabilitas P_{follow}	24
4.1	Tabel penjelasan kegunaan <i>item</i> pada <i>mockup</i>	30
5.1	Tabel penjelasan dari file yang diuji	43
5.2	Hasil pengujian fungsional untuk 7 node	44
5.3	Hasil pengujian fungsional untuk 10 node	44
5.4	Hasil pengujian fungsional untuk 15 node	45
5.5	Hasil pengujian fungsional untuk 20 node	45
5.6	Table probabilitas P_{follow} pertama	46
5.7	Table probabilitas P_{follow} kedua	46
5.8	Table probabilitas P_{follow} ketiga	46
5.9	Table probabilitas P_{follow} terakhir	47
5.10	Hasil pengujian dengan menggunakan nilai P_{follow}	47
5.11	Rata-rata hasil pengujian untuk nilai α , terhadap setiap test pengujian	47
5.12	Rata-rata hasil pengujian untuk nilai β , terhadap setiap test pengujian	48
5.13	Rata-rata hasil pengujian untuk nilai λ , terhadap setiap test pengujian	48
5.14	Hasil pengujian 10 node, dengan membiarkan program mencari jalur yang dalam 1000000 iterasi tidak berubah dan hasil pengujian <i>brute force</i>	49
5.15	Hasil pengujian 20 node, dengan membiarkan program mencari jalur yang dalam 1000000 iterasi tidak berubah dan hasil pengujian <i>brute force</i>	50
5.16	Hasil pengujian 50 node, dengan membiarkan program mencari jalur yang dalam 1000000 iterasi tidak berubah dan hasil pengujian terkecil	51
5.17	Hasil pengujian 100 node, dengan membiarkan program mencari jalur yang dalam 1000000 iterasi tidak berubah dan hasil pengujian terkecil	52
5.18	Hasil pengujian 200 node, dengan membiarkan program mencari jalur yang dalam 1000000 iterasi tidak berubah dan hasil pengujian terkecil	53
B.1	Hasil Pengujian Nilai α , dengan 200 node yang terbagi kedalam 14 <i>family</i>	71
B.2	Hasil Pengujian Nilai α , dengan 100 node yang terbagi kedalam 10 <i>family</i>	71
B.3	Hasil Pengujian Nilai α , dengan 50 node yang terbagi kedalam 7 <i>family</i>	72
B.4	Hasil Pengujian Nilai α , dengan 20 node yang terbagi kedalam 4 <i>family</i>	72
B.5	Hasil Pengujian Nilai α , dengan 10 node yang terbagi kedalam 3 <i>family</i>	72
C.1	Hasil Pengujian Nilai β , dengan 200 node yang terbagi kedalam 14 <i>family</i>	73
C.2	Hasil Pengujian Nilai β , dengan 100 node yang terbagi kedalam 10 <i>family</i>	73
C.3	Hasil Pengujian Nilai β , dengan 50 node yang terbagi kedalam 7 <i>family</i>	73
C.4	Hasil Pengujian Nilai β , dengan 20 node yang terbagi kedalam 4 <i>family</i>	73
C.5	Hasil Pengujian Nilai β , dengan 10 node yang terbagi kedalam 3 <i>family</i>	73

D.1	Hasil Pengujian Nilai λ , dengan 200 node yang terbagi kedalam 14 <i>family</i>	75
D.2	Hasil Pengujian Nilai λ , dengan 100 node yang terbagi kedalam 10 <i>family</i>	75
D.3	Hasil Pengujian Nilai λ , dengan 50 node yang terbagi kedalam 7 <i>family</i>	75
D.4	Hasil Pengujian Nilai λ , dengan 20 node yang terbagi kedalam 4 <i>family</i>	75
D.5	Hasil Pengujian Nilai λ , dengan 10 node yang terbagi kedalam 3 <i>family</i>	75

BAB 1

PENDAHULUAN

Bab ini berisi tentang pendahuluan dalam penelitian, yang berisi latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan. Latar belakang masalah akan berisi alasan kenapa topik ini dipilih, rumusan masalah akan berisi masalah-masalah yang akan diteliti, tujuan penelitian adalah untuk menjawab rumusan masalah, pembatasan masalah dilakukan agar penelitian menjadi lebih fokus ke hal yang akan dituju, metodologi akan berisi landasan teori yang akan membantu dalam penelitian, sistematika pembahasan akan berisi detail dari bab-bab yang akan ditulis.

1.1 Latar Belakang

Pencarian jalur terdekat adalah suatu permasalahan di mana tujuan dari permasalahan ini adalah mendapatkan rute dengan jarak tempuh seminimal mungkin. *Family Traveling Salesman Problem* (FTSP) merupakan salah satu permasalahan pencarian jalur terdekat[1]. Salah satu permasalahan yang dapat dimodelkan dengan FTSP adalah pencarian barang dalam gudang. Dalam permasalahan ini akan dicari beberapa gudang yang terbagi kedalam kelompok-kelompok gudang lalu dalam kelompok gudang tersebut terdapat jumlah gudang yang harus dikunjungi untuk mendapatkan barang yang diinginkan oleh pelanggan. Hasil dari permasalahan ini akan berupa jalur terpendek yang akan mengunjungi gudang-gudang yang dibutuhkan untuk memenuhi pesanan pelanggan. Untuk mencari jalur tersebut akan digunakan algoritma *Bee Colony Optimization*.

Solusi dari FTSP dapat diselesaikan dengan algoritma *brute force*, yaitu dengan menelusuri setiap kemungkinan solusi yang ada. Tetapi karena dalam pencarian menggunakan algoritma *brute force* memakan waktu yang cukup lama, sehingga hanya dapat diimplementasikan untuk graf yang berukuran kecil. Oleh karena itu dikembangkan algoritma heuristik untuk mencari jalur sekecil mungkin tetapi membutuhkan waktu pengerjaan yang lebih cepat yaitu algoritma *bee colony optimization*.

Algoritma *Bee Colony Optimization* (BCO) merupakan salah satu algoritma pencarian. BCO dapat menghasilkan solusi optimal dari suatu permasalahan kombinatorial yang kompleks. BCO merupakan algoritma yang terinspirasi oleh perilaku lebah dalam proses pengumpulan sari pati bunga yang akan digunakan sebagai bahan dasar madu [2]. Hasil dari algoritma ini adalah lokasi-lokasi yang memiliki jumlah dan hasil makanan yang baik untuk koloni lebah. Akan dibuat perangkat lunak yang mengimplementasikan algoritma BCO yang dapat menyelesaikan permasalahan pencarian barang dalam gudang yang dimodelkan dalam FTSP. Dalam perangkat lunak ini dibutuhkan pemodelan permasalahan dalam bentuk graf FTSP. Isi dari graf tersebut adalah koordinat (x, y) dari letak gudang.

Hasil dari perangkat lunak yang akan dibuat berupa jalur dan panjang dari jalur yang ditemukan. Hal yang dilakukan untuk menguji kualitas hasil solusi yang diperoleh algoritma BCO adalah dengan membandingkan solusi yang dihasilkan BCO dengan solusi yang diperoleh oleh algoritma *brute force*.

1.2 Rumusan Masalah

Seperti yang telah dijabarkan sebelumnya, permasalahan FTSP adalah pencarian jalur terdekat untuk mengunjungi beberapa buah node yang terbagi dalam sejumlah kelompok. Sebelum mencari jalur terdekat akan dilakukan pemodelan masalah *Family Traveling Salesman Problem*, harus mengetahui cara kerja dari algoritma *bee colony optimization*, dan menyelesaikan permasalahan FTSP menggunakan algoritma *bee colony optimization*. Dalam skripsi ini ada dua rumusan masalah yang akan diselesaikan yaitu:

1. Bagaimana membangun perangkat lunak yang mengimplementasikan algoritma BCO untuk menyelesaikan permasalahan FTSP?
2. Bagaimana hasil kinerja dari perangkat lunak yang dibuat jika dibandingkan dengan algoritma *brute force*?

1.3 Tujuan

Dengan mengacu pada rumusan masalah, skripsi ini akan membahas tentang permasalahan tersebut beserta analisis solusinya. Tujuan yang hendak dicapai dari penelitian ini adalah:

1. Membangun perangkat lunak yang mengimplementasikan algoritma BCO yang dapat digunakan untuk mencari jarak minimal dari permasalahan FTSP.
2. Mengukur hasil dari kerja perangkat lunak setelah dibandingkan dengan algoritma *brute force*.

1.4 Batasan Masalah

Pada penelitian ini, permasalahan akan dibatasi sebagai berikut:

1. Input dari perangkat lunak yang akan dibangun merupakan kasus FTSP simetris dimana setiap simpul-simpulnya saling bertetangga.

1.5 Metodologi

Metodologi yang dipakai dalam penulisan tugas akhir ini adalah:

1. Melakukan studi pustaka tentang graf, *family traveling salesman problem*, dan *bee colony optimization*.
2. Melakukan analisis algoritma *bee colony optimization*. Analisis ini dilakukan untuk lebih mengetahui cara kerja dari algoritma *bee colony optimization*.
3. Melakukan pemodelan BCO untuk menyelesaikan FTSP.
4. Melakukan perancangan model perangkat lunak. Perancangan yang dilakukan adalah merancang diagram kelas yang akan digunakan, dan perancangan antarmuka.
5. Melakukan implementasi rancangan perangkat lunak. Implementasi akan dilakukan menggunakan bahasa *Java* dengan bantuan aplikasi pemograman Netbean.
6. Melakukan pengujian perangkat lunak. Pengujian ini bertujuan untuk mengetahui apakah perangkat lunak dapat berjalan dengan baik. Pengujian yang dilakukan adalah pengujian fungsional dan pengujian eksperimental.
7. Melakukan pengambilan kesimpulan dari hasil pengujian.
8. Membuat penyusunan dokumentasi seluruh kegiatan yang dilakukan kedalam laporan dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematika yang akan dilakukan pada penulisan karya tulis ini adalah:

- Bab satu berisi latar belakang tentang topik penyelesaian *family traveling salesman problem* dengan algoritma *bee colony*, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.
- Bab dua berisi pembahasan tentang graf, *Traveling Salesman Problem*, *Family Traveling Salesman Problem*, dan *Bee Colony Optimization*.
- Bab tiga berisi analisis masalah, dan analisis perangkat lunak.
- Bab empat berisi perancangan tampilan antarmuka, implementasi perangkat lunak, dan diagram kelas rinci dari perangkat lunak.
- Bab lima berisi pengujian perangkat lunak di mana pengujian yang dilakukan adalah pengujian fungsional dan pengujian eksperimental.
- Bab enam berisi kesimpulan dan saran.