

SKRIPSI

**PENYELESAIAN MULTIPLE TRAVELING SALESMAN
PROBLEM DENGAN MENGGUNAKAN HYBRID
GRAVITATIONAL ALGORITHM**



Adithia Theo Buditama

NPM: 2013730040

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2020**

UNDERGRADUATE THESIS

**SOLVING MULTIPLE TRAVELING SALESMAN PROBLEMS
USING HYBRID GRAVITATIONAL ALGORITHMS**



Adithia Theo Buditama

NPM: 2013730040

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2020**

PERNYATAAN

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

PENYELESAIAN MULTIPLE TRAVELING SALESMAN PROBLEM DENGAN MENGGUNAKAN HYBRID GRAVITATIONAL ALGORITHM

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 8 Agustus 2020



Adithia Theo Buditama
NPM: 2013730040

LEMBAR PENGESAHAN

PENYELESAIAN MULTIPLE TRAVELING SALESMAN PROBLEM DENGAN MENGGUNAKAN HYBRID GRAVITATIONAL ALGORITHM

Adithia Theo Buditama

NPM: 2013730040

Bandung, 8 Agustus 2020

Menyetujui,

Pembimbing

Dr.rer.nat. Cecilia Esti Nugraheni

Ketua Tim Penguji

Anggota Tim Penguji

Luciana Abednego, M.T.

Lionov, Ph.D.

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 MULTIPLE TRAVELING SALESMAN PROBLEM DENGAN MENGGUNAKAN HYBRID GRAVITATIONAL ALGORITHM

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 8 Agustus 2020



Adithia Theo Buditama
NPM: 2013730040

ABSTRAK

Travelling Salesman Problem (TSP) adalah masalah pencarian rute terpendek atau rute dengan total jarak minimum oleh seorang *salesman* dari suatu kota asal ke sejumlah kota lain tepat satu kali dan kembali ke kota asal. *Multiple Travelling Salesman Problem* (MTSP) dapat dipandang sebagai pengembangan atau bentuk umum dari TSP, dimana banyaknya *salesman* dan kota asal bisa lebih dari satu.

Permasalahan *Multiple Travelling Salesman Problem* (MTSP) merupakan permasalahan optimisasi kombinatorial, yaitu suatu cara yang digunakan untuk mencari semua kemungkinan nilai *real* dari suatu fungsi objektif. Dengan kata lain, optimisasi kombinatorial mencari jarak minimum yang ditempuh *salesmen*. Algoritma dari optimalisasi kombinatorial digunakan untuk menyelesaikan masalah yang cukup rumit dan memiliki ruang lingkup yang cukup besar seperti *Multiple Travelling Salesman Problem* (MTSP). Beberapa algoritma yang dapat digunakan untuk permasalahan optimisasi kombinatorial adalah algoritma genetik dan algoritma pencarian gravitasional.

Algoritma genetik yang mengadopsi proses evolusi makhluk hidup, pada penerapannya mengikuti langkah pembentukan populasi, menentukan nilai *fitness*, melakukan proses seleksi, melakukan pindah silang dan mutasi, dan membentuk individu baru. Populasi yang dibentuk berisi individu-individu yang berisi urutan kota dan urutan *salesmen* yang menempuh kota tersebut. Proses pindah silang dan mutasi akan membuat urutan kota dan urutan *salesmen* berubah sehingga menghasilkan individu baru.

Algoritma pencarian gravitasional merupakan algoritma pencari kumpulan partikel yang berinteraksi satu sama lain. Algoritma ini terinspirasi dari fenomena alam yaitu hukum gravitasi dan gerak Newton dimana setiap partikel saling tarik menarik dan bergerak menuju partikel yang terbaik. Implementasi dari algoritma pencarian gravitasional yaitu memilih satu partikel terbaik yang akan dijadikan acuan terhadap partikel-partikel lain yang kurang baik sehingga partikel-partikel yang kurang baik tersebut bergerak mendekati partikel terbaik.

Hasil dari penelitian dalam skripsi ini adalah algoritma genetik dan algoritma pencarian gravitasional dapat diimplementasikan untuk mencari solusi MTSP, yaitu rute minimum yang dapat ditempuh oleh seluruh *salesman*. Perangkat lunak sudah diuji fungsionalitasnya dengan parameter *input* yang berbeda dan uji eksperimental perangkat lunak menghasilkan solusi yang mendekati optimal untuk kasus yang dicoba di dalam eksperimen.

Kata-kata kunci: *multiple traveling salesman problem*, *salesman*, rute minimum, algoritma genetik, algoritma pencarian gravitasional

ABSTRACT

Traveling Salesman Problem (TSP) is the problem of finding the shortest route or a route with a minimum total distance by an individual from one city of origin to a number of other cities exactly one time and returning to the city of origin. Multiple Traveling Salesman Problems (MTSP) can be seen as a development or general form of TSP, wherein there are so many people and cities of origin that can be more than one.

The Problem of Multiple Traveling Salesman Problems (MTSP) is a combinatorial optimization problem, which is a method used to find all possible real values of an objective function. In other words, combinatorial optimization looks for the minimum distance traveled by salesmen. Algorithms from combinatorial optimization are used to solve problems that are quite complex and have a large enough scope such as Multi Traveling Salesman Problem (MTSP). Some algorithms that can be used for combinatorial optimization problems are genetic algorithms and gravitational search algorithms.

Genetic algorithms that adopt the evolutionary process of living things, in their application follow the steps of population formation, determine the value of witnesses, carry out the selection process, make cross-moves and mutations, and form new individuals. Population formed contains individuals who contain the order of the city and the sequence of the people who take the city. The process of crossing and mutation will make the order of the city and sequence change to produce a new individual.

Gravitational search algorithm is a search algorithm for a collection of particles that interact with each other. This algorithm is inspired by natural phenomena namely Newton's laws of gravity and motion in which each particle attracts and moves towards the best particles. The implementation of gravitational search algorithm is to choose one of the best particles that will be used as a reference for other particles that are not good so that the particles that are not moving well approach the best particles.

The results of the research in this thesis are genetic algorithms and gravity search algorithms can be implemented to find MTSP solutions, which is the minimum route that can be taken by all salesmen. The software has been tested for its functionality with different input parameters and the experimental test of the software produces a near optimal solution to the cases that were tried in the experiment.

Keywords: multiple traveling salesman problem, minimum route, genetic algorithms, gravitational search algorithms

Your life is as good as your mindset

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus atas penyertaan-Nya selama proses penulisan skripsi ini sehingga skripsi ini dapat selesai dengan baik. Saya juga ingin berterima kasih kepada beberapa pihak yang telah membantu dan mendukung saya selama proses penulisan skripsi ini:

1. Papi, Mami, adik laki-lakiku Oka, dan adik perempuanku Trixy yang selalu memberi motivasi positif dan selalu percaya bahwa saya bisa menyelesaikan skripsi ini.
2. Ibu Cecilia Esti Nugraheni sebagai pembimbing skripsi saya. Terima kasih atas semua waktu dan dukungan yang telah Ibu berikan kepada saya sehingga saya dapat menyelesaikan skripsi ini. Terima kasih sudah sabar dalam membimbing saya. Saya akan selalu ingat semua hal-hal positif yang telah Ibu berikan kepada saya.
3. Ibu Luciana Abednego sebagai penguji skripsi saya. Terima kasih sudah memberikan masukan yang baik untuk skripsi saya sehingga saya dapat menyelesaikan skripsi ini dengan baik.
4. Pak Lionov sebagai penguji skripsi saya. Terima kasih sudah memberikan masukan yang baik untuk skripsi saya sehingga saya dapat menyelesaikan skripsi ini dengan baik.
5. Teman-teman Informatika Universitas Katolik Parahyangan angkatan 2013 baik yang sudah lulus, tidak lulus, ataupun mengundurkan diri, tetap semangat untuk segala sesuatu yang sedang kalian rintis atau kerjakan. Semoga menjadi buah yang baik untuk kalian semua.
6. Teman-teman SMA Penabur Cirebon angkatan 2013. Tetap semangat untuk segala sesuatu yang sedang kalian rintis atau kerjakan. Semoga menjadi buah yang baik untuk kalian semua.
7. Siapapun yang menjadikan skripsi ini sebagai referensi. Semoga skripsi yang saya tulis ini dapat membantu dan memberikan manfaat positif untuk kalian.

Bandung, Agustus 2020

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	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Metodologi	4
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 Graf[1]	5
2.1.1 Jenis-JenisGraf	6
2.1.2 Terminologi Graf	7
2.1.3 Lintasan dan Sirkuit Hamilton	10
2.2 Multiple Traveling Salesman Problem	11
2.3 Proses Optimasi [2]	12
2.4 Heuristik dan Metaheuristik [3]	12
2.5 Algoritma Genetik	13
2.5.1 Istilah-istilah dalam Algoritma Genetik	13
2.5.2 Parameter dalam Algoritma Genetik	14
2.5.3 Proses Pengkodean (<i>Encoding</i>)	15
2.5.4 Proses Pembangkitan Populasi (Inisialisasi)	15
2.5.5 Proses Seleksi	16
2.5.6 Proses <i>Crossover</i>	17
2.5.7 Proses Mutasi	21
2.5.8 Penentuan Parameter	21
2.5.9 Terminasi	21
2.6 Gravitational Search Algorithm	22
2.7 <i>Hybrid Gravitational Algorithm</i>	26
3 ANALISIS	27
3.1 Analisis Cara Kerja HGA	27
3.2 Pemodelan Graf	28
3.3 Pemodelan Jarak Antar Kota	28
3.4 Algoritma Genetik	29
3.4.1 Pemodelan Kromosom	29

3.4.2	Perhitungan Nilai <i>Fitness</i>	29
3.4.3	Seleksi, <i>Crossover</i> , dan Mutasi	29
3.5	Algoritma Pencarian Gravitasi	33
3.6	Deskripsi Perangkat Lunak	39
3.7	Diagram <i>Use Case</i>	40
4	PERANCANGAN	47
4.1	Perancangan Masukan	47
4.2	Perancangan Keluaran	48
4.3	Perancangan Antarmuka	48
4.4	Diagram Kelas	50
4.4.1	Chromosome	51
4.4.2	GA	52
4.4.3	GSA	55
4.4.4	GaGsaInterface	57
4.4.5	MTSP	58
4.4.6	Problem	59
5	IMPLEMENTASI DAN PENGUJIAN	61
5.1	Lingkungan Perangkat Keras	61
5.2	Lingkungan Perangkat Lunak	61
5.3	Implementasi	61
5.4	Pengujian Fungsional	62
5.5	Pengujian Eksperimental	73
5.5.1	Parameter Eksperimen	73
5.5.2	Hasil Eksperimen	74
6	KESIMPULAN DAN SARAN	77
6.1	Kesimpulan	77
6.2	Saran	77
	DAFTAR REFERENSI	79
	A KODE PROGRAM	81
	B HASIL EKSPERIMEN	97

DAFTAR GAMBAR

1.1	Representasi Graf Untuk Contoh Masalah TSP	2
1.2	Representasi Graf Untuk Contoh Masalah MTSP	3
2.1	Ilustrasi Graf 1	5
2.2	Ilustrasi Graf 2	5
2.3	Ilustrasi Graf 3	6
2.4	Ilustrasi Graf Berarah	7
2.5	Matriks Ketetanggaan Untuk Graf Tidak Berarah	8
2.6	Matriks Ketetanggaan Untuk Graf Berarah	9
2.7	Matriks Ketetanggaan Untuk Graf Berbobot	9
2.8	Ilustrasi Graf Tak Berarah	9
2.9	Ilustrasi Graf Berbobot	10
2.10	Representasi <i>inked list</i> untuk graf berbobot	10
2.11	Contoh Graf Semi-Hamilton	11
2.12	Contoh Graf Hamilton	11
2.13	Visualisasi Gen, Allele, Kromosom, dan Populasi	14
2.14	Diagram Pie Metode Roulette Wheel Selection	16
2.15	Diagram Pie sebelum Ranking (Diagram berdasarkan <i>Fitness</i>)	17
2.16	Diagram Pie sesudah Ranking (Diagram berdasarkan Urutan)	17
2.17	Single-Point Crossover	18
2.18	Multi-Point Crossover	19
2.19	Order Crossover	19
2.20	HAG Crossover	20
2.21	Pemodelan Jarak Euclidian	20
2.22	Diagram Alur Algoritma Genetik	22
2.23	Ilustrasi Gaya Tarik Menarik Antar Partikel	23
2.24	Diagram Alur Algoritma Pencarian Gravitasi	25
2.25	Diagram Alur <i>Hybrid Gravitational Algorithm</i>	26
3.1	Diagram Alur GA - GSA	27
3.2	Contoh Pemodelan Graf MTSP	28
3.3	Pemodelan Jarak Antar Kota	28
3.4	Contoh Kromosom	29
3.5	Contoh Kromosom dan Nilai <i>fitness</i>	30
3.6	Contoh Probabilitas Setiap Kromosom	31
3.7	Contoh <i>Order Crossover</i>	31
3.8	Contoh Mutasi	32
3.9	Populasi Hasil Dari Algoritma Genetik	34
3.10	Perubahan Posisi Partikel	35
3.11	Contoh Pendekatan Partikel j Ke Partikel i	36
3.12	Kota Ke k Pada PB	37
3.13	Mencari kota setelah kota k pada PA dan PB	37
3.14	Hapus kota ke k pada PA dan PB	37

3.15	Memasukkan k ke partikel baru (Langkah Pertama)	38
3.16	Memasukkan k ke partikel baru (Langkah Kedua)	38
3.17	Memasukkan k ke partikel baru (Langkah Ketiga)	38
3.18	Memasukkan k ke partikel baru (Langkah Keempat)	38
3.19	Perubahan Partikel	39
3.20	Diagram <i>Use Case</i>	41
4.1	<i>File</i> Graf Peta	47
4.2	<i>Output</i> Perangkat Lunak	48
4.3	Rancangan Antarmuka Perangkat Lunak	49
4.4	Diagram Kelas	50
4.5	Kelas Chromosome	51
4.6	Kelas GA	53
4.7	Kelas GSA	55
4.8	Kelas Particle	56
4.9	Kelas GaGsaInterface	57
4.10	Kelas MTSP	59
4.11	KelasProblem	60
5.1	Menekan Tombol Cari	62
5.2	Memasukkan Jumlah <i>Salesman</i>	63
5.3	Memasukkan Nomor Kota Asal	64
5.4	Memasukkan Ukuran Populasi	65
5.5	Memasukkan Banyak Iterasi	66
5.6	Memasukkan Probabilitas <i>Crossover</i>	67
5.7	Memasukkan Probabilitas Mutasi	68
5.8	Menekan Tombol Proses	69
5.9	Memasukkan <i>salesman</i> Sama Dengan Jumlah Kota	70
5.10	Menekan tombol proses tanpa ada <i>file input</i> dan <i>input</i>	71
5.11	Memasukkan Ukuran Populasi Bukan Dengan Angka	72
5.12	Memasukkan <i>file Input</i> Yang Salah	73
5.13	MTSP <i>benchmark</i> dengan <i>Optimal Cost</i>	75
B.1	Koordinat Kota Nomor 1 sampai Kota Nomor 25	97
B.2	Koordinat Kota Nomor 26 sampai Kota Nomor 51	98
B.3	Matriks Ketetanggaan 51 Kota	99

DAFTAR TABEL

2.1	<i>Adjacency list</i> untuk graf pada Gambar	10
2.2	Permutation Encoding pada TSP	15
3.1	Skenario <i>Use Case</i> Memasukkan Graf Permasalahan MTSP	42
3.2	Skenario <i>Use Case</i> Memasukkan Banyaknya <i>Salesman</i>	43
3.3	Skenario <i>Use Case</i> Memasukkan Nomor Kota Asal	43
3.4	Skenario <i>Use Case</i> Memasukkan Ukuran Populasi	44
3.5	Skenario <i>Use Case</i> Memasukkan Maksimal Iterasi	44
3.6	Skenario <i>Use Case</i> Memasukkan Probabilitas <i>Crossover</i>	45
3.7	Skenario <i>Use Case</i> Memasukkan Probabilitas Mutasi	45
3.8	Skenario <i>Use Case</i> Pencarian Solusi	46
3.9	Skenario <i>Use Case</i> Mengakhiri Program	46
5.1	Lingkungan perangkat keras	61
5.2	Lingkungan perangkat lunak	61
5.3	Tabel Hasil Pengujian Eksperimental Terbaik Menggunakan Algoritma Genetik . .	74
5.4	Tabel Hasil Pengujian Eksperimental Terbaik Menggunakan Gabungan Algoritma Genetik Dan Algoritma Pencarian Gravitasiional	74
5.5	Tabel Hasil Pengujian Eksperimental Terburuk Menggunakan Algoritma Genetik .	74
5.6	Tabel Hasil Pengujian Eksperimental Terburuk Menggunakan Gabungan Algoritma Genetik Dan Algoritma Pencarian Gravitasiional	75
B.1	Tabel Parameter Pertama MTSP Untuk Kasus 1	100
B.2	Tabel Parameter Kedua MTSP Untuk Kasus 1	101
B.3	Tabel Parameter Ketiga MTSP Untuk Kasus 1	102
B.4	Tabel Parameter Keempat MTSP Untuk Kasus 1	103
B.5	Tabel Pengujian Eksperimental Pertama Untuk Kasus 1 Menggunakan Algoritma Genetik	104
B.6	Tabel Pengujian Eksperimental Pertama Untuk Kasus 1 Menggunakan Algoritma Genetik Dan Algoritma Pencarian Gravitasiional	105
B.7	Tabel Pengujian Eksperimental Kedua Untuk Kasus 1 Menggunakan Algoritma Genetik	106
B.8	Tabel Pengujian Eksperimental Kedua Untuk Kasus 1 Menggunakan Algoritma Genetik Dan Algoritma Pencarian Gravitasiional	107
B.9	Tabel Pengujian Eksperimental Ketiga Untuk Kasus 1 Menggunakan Algoritma Genetik	108
B.10	Tabel Pengujian Eksperimental Ketiga Untuk Kasus 1 Menggunakan Algoritma Genetik Dan Algoritma Pencarian Gravitasiional	109
B.11	Tabel Pengujian Eksperimental Keempat Untuk Kasus 1 Menggunakan Algoritma Genetik	110
B.12	Tabel Pengujian Eksperimental Keempat Untuk Kasus 1 Menggunakan Algoritma Genetik Dan Algoritma Pencarian Gravitasiional	111

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Traveling Salesman Problem (TSP) merupakan sebuah permasalahan optimasi yang 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 rute yang optimum yaitu jarak minimum terbaik.

Multiple Traveling Salesman Problem (MTSP) adalah pengembangan dari TSP atau bentuk umum dari TSP, dimana banyaknya *salesman* lebih dari satu. Setiap *Salesman* dapat mengunjungi sejumlah kota dari kota asal yang sama atau dari kota asal yang berbeda. Permasalahan TSP dan MTSP sering dimodelkan dengan graf berbobot dengan simpul merepresentasikan kota dan sisi merepresentasikan jalan yang menghubungkan dua buah kota. Graf dapat dimodelkan sebagai graf terhubung lengkap berbobot positif dan juga graf terhubung tidak lengkap berbobot positif.

Algoritma metaheuristik digunakan untuk menyelesaikan permasalahan *Multiple Traveling Salesman Problem* (MTSP). Optimasi metaheuristik dilakukan dengan memperbaiki kandidat penyelesaian secara iteratif sesuai dengan fungsi objektifnya. Metode ini mampu menghasilkan penyelesaian yang baik dalam waktu yang cepat, tetapi tidak menjamin bahwa penyelesaian yang dihasilkan merupakan penyelesaian terbaik (optimal). Metode metaheuristik banyak dipakai dalam optimisasi stokastik (optimisasi stokastik merupakan optimisasi yang memiliki derajat ketidakpastian atau random).

Kelebihan algoritma metaheuristik jika dibandingkan dengan algoritma deterministik adalah algoritma metaheuristik berusaha menemukan solusi yang baik tanpa melebihi batasan waktu yang telah ditentukan. Solusi yang baik belum tentu yang paling optimal, namun sudah dapat diterima oleh *user*. Sedangkan optimasi deterministik selalu menghasilkan solusi yang tetap untuk suatu input yang diberikan. Algoritma ini biasanya digunakan untuk masalah yang ruang solusinya tidak terlalu besar, sedangkan optimasi probabalistik digunakan untuk menyelesaikan ruang masalah dengan ruang solusi yang sangat besar, bahkan tak terbatas.

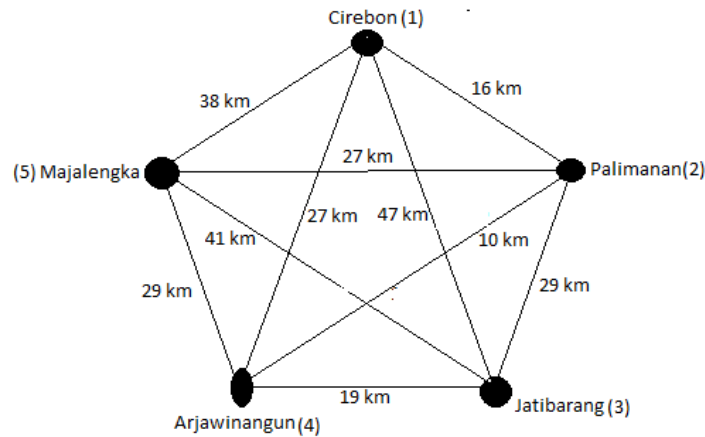
Beberapa contoh metaheuristik adalah algoritma genetik (GA - *Genetics Algorithm*), algoritma koloni semut (ACO - *Ant Colony Optimization*), simulated annealing (SA - *Simulated Annearling*), algoritma pencarian gravitasional (GSA - *Gravitational Search Algorithm*), tabu search (TS - *Tabu Search*), algoritma *firefly*, dan sebagainya.

Algoritma genetik dapat digunakan untuk menentukan rute minimal optimum pada TSP dan MTSP. Penentuan rute minimal optimum dengan langkah mendefinisikan populasi, menentukan nilai *fitness*, melakukan proses seleksi, melakukan *crossover*, melakukan mutasi, dan memperoleh individu baru yang menuju ke penyelesaian optimal.[4]

Algoritma pencarian gravitasional dapat digunakan untuk menentukan rute minimal optimum pada TSP dan MTSP. Algoritma ini terinspirasi dari fenomena alam yaitu hukum gravitasi dan gerak Newton. Menurut hukum gaya gravitasi Newton, setiap partikel akan menarik partikel lain

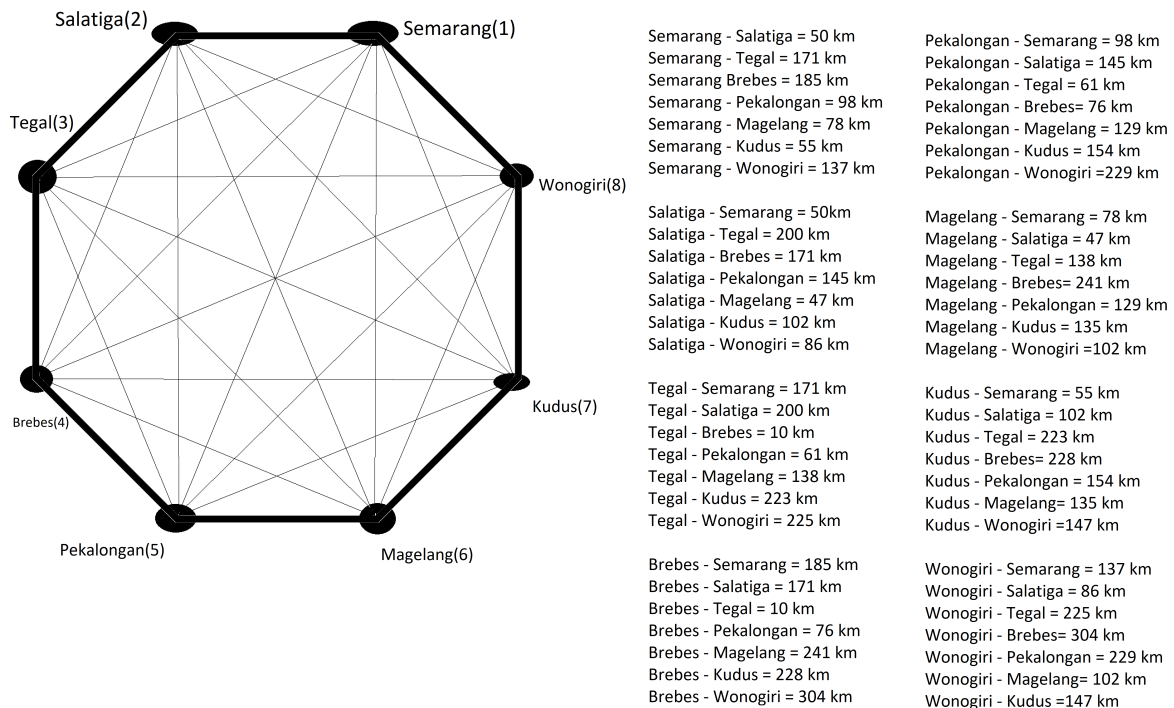
dengan gaya gravitasi sebanding dengan massanya dan berbanding terbalik dengan kuadrat jarak diantara keduanya. Sedangkan menurut hukum gerak Newton, percepatan partikel dapat ditentukan dengan membandingkan gaya interaksi antar partikel dan massa partikel. Pada algoritma pencarian gravitasional, massa atau benda dianggap sebagai sebuah partikel yang berinteraksi satu sama lain untuk mencari solusi optimal sesuai dengan hukum gravitasi dan gerak Newton.[4]

Penyelesaian TSP dan MTSP dalam skripsi ini menggunakan *Hybird Gravitational Algorithm* (HGA). Algoritma ini merupakan penggabungan dari 2 algoritma yaitu algoritma genetik dan algoritma pencarian gravitasional. Kemampuan dari algoritma genetik yaitu mampu mencapai *global optimum* dan algoritma pencarian gravitasional untuk mencegah terjadinya permasalahan *local optimum*. Diharapkan dengan menggunakan 2 kemampuan algoritma ini, masalah MTSP dapat diselesaikan.[4]



Gambar 1.1: Representasi Graf Untuk Contoh Masalah TSP

Gambar 1.1 merupakan representasi graf dari contoh masalah TSP dimana terdapat 5 buah simpul yang merepresentasikan kota Cirebon, Palimanan, Jatibarang, Arjawinangun, dan Majalengka. Sisi antar simpul tersebut yang merepresentasikan jarak antar kota. Untuk permasalahan TSP dimana hanya terdapat satu *salesman* yang harus mengunjungi tiap kota satu kali dan diasumsikan simpul 1 adalah kota awal, maka rute minumum terbaik yang dapat dilalui oleh salesman tersebut adalah Cirebon - Palimanan - Arjawinangun - Jatibarang - Majalengka - Cirebon dengan total jarak yang ditempuh salesman tersebut adalah 124 km.



Gambar 1.2: Representasi Graf Untuk Contoh Masalah MTSP

Gambar 1.2 merupakan representasi graf dari contoh masalah MTSP dimana dimana terdapat 8 buah simpul yang merepresentasikan kota Semarang, Salatiga, Tegal, Brebes, Pekalongan, Magelang, Kudus, Wonogiri. Untuk permasalahan MTSP dimana diasumsikan terdapat 2 *salesman* yang harus mengunjungi tiap kota satu kali dan diasumsikan simpul 1 sebagai kota awal, maka rute minimum terbaik yang dapat dilalui oleh salesman adalah masing-masing sebagai berikut:

- Rute *salesman*1 adalah Semarang - Salatiga - Wonogiri - Magelang - Tegal - Brebes - Pekalongan - Semarang dengan total jarak 560km.
- Rute *salesman*2 adalah Semarang - Kudus - Semarang dengan total jarak 110km.

Total jarak yang ditempuh kedua salesman tersebut adalah 670 km.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka permasalahan yang diangkat pada skripsi ini dapat dirumuskan sebagai berikut:

1. Bagaimana cara kerja GA dan GSA?
2. Bagaimana menggabungkan GA dan GSA untuk menyelesaikan permasalahan MTSP?
3. Bagaimana membuat perangkat lunak untuk menyelesaikan permasalahan MTSP dengan menggunakan gabungan GA dan GSA?

1.3 Tujuan

1. Membuat perangkat lunak untuk mencari solusi MTSP dengan menggunakan gabungan antara algoritma genetik dan algoritma pencarian gravitasi *Hybrid Gravitational Algorithm*.

2. Membandingkan solusi yang dihasilkan perangkat lunak hanya dengan menggunakan algoritma genetik dan gabungan antara algoritma genetik dan algoritma pencarian gravitasional *Hybrid Gravitational Algorithm*

1.4 Batasan Masalah

Batasan masalah yang dibahas pada skripsi ini dapat dirumuskan sebagai berikut:

1. Multiple traveling salesman problem (MTSP) pada skripsi ini menggunakan sistem satu kota asal (setiap *salesman* akan berangkat dan kembali pada simpul kota yang sama).
2. Perangkat lunak hanya mampu mengeluarkan solusi MTSP apabila graf yang hendak dicari solusinya merupakan graf terhubung lengkap berbobot positif.
3. Banyaknya kota selalu lebih besar daripada banyaknya *salesman*.

1.5 Metodologi

Metodologi yang dikerjakan pada skripsi ini dapat dirumuskan sebagai berikut:

1. Studi literatur yang meliputi MTSP, GA, GSA, Hybrid Gravitational Algorithm(GA - GSA) untuk penyelesaian MTSP.
2. Pengembangan perangkat lunak yang meliputi analisis, desain, implementasi, dan pengujian.
3. Uji eksperimental menggunakan MTSP *benchmark*
4. Pengambilan kesimpulan.

1.6 Sistematika Pembahasan

1. BAB 1 - berisi tentang latar belakang masalah, rumusan masalah, tujuan penulisan skripsi, deskripsi perangkat lunak, batasan masalah, metodologi penelitian, dan sistematika pembahasan.
2. BAB 2 - berisi tentang landasan teori yang digunakan untuk mendukung dasar-dasar analisis dan perancangan perangkat lunak, yaitu teori graf, *multiple traveling salesman problem*, algoritma genetik, algoritma pencarian gravitasional, dan *Hybrid Gravitational Algorithm*.
3. BAB 3 - berisi tentang analisis penyelesaian MTSP dengan Hybrid Gravitational Algorithm(GA - GSA) dan diagram use case.
4. BAB 4 - berisi tentang perancangan masukan, perancangan keluaran, perancangan antarmuka, dan diagram kelas perangkat lunak.
5. BAB 5 - berisi tentang lingkungan perangkat keras, lingkungan perangkat lunak, implementasi perangkat lunak, pengujian fungsional, dan pengujian eksperimental.
6. BAB 6 - berisi tentang kesimpulan dari hasil pengujian perangkat lunak dan saran-saran untuk perangkat lunak yang dibangun.