TUGAS AKHIR

PENYELESAIAN *TRAVELING SALESMAN PROBLEM* MENGGUNAKAN ALGORITMA *FRUITFLY OPTIMIZATION*

Darren Lee

NPM: 6181801009

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2024

FINAL PROJECT

SOLVING TRAVELING SALESMAN PROBLEM USING FRUITFLY OPTIMIZATION ALGORITHM

Darren Lee

NPM: 6181801009

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2024

LEMBAR PENGESAHAN

PENYELESAIAN *TRAVELING SALESMAN PROBLEM* MENGGUNAKAN ALGORITMA *FRUITFLY OPTIMIZATION*

Darren Lee

NPM: 6181801009

Bandung, 11 Juli 2024

Menyetujui,

Pembimbing

Digitally signed by Husnul Hakim

Husnul Hakim, M.T.

Ketua Tim Penguji

Digitally signed by Cecilia Esti Nugraheni

Dr.rer.nat. Cecilia Esti Nugraheni

Anggota Tim Penguji

Digitally signed by Natalia

Natalia, M.Si.

Mengetahui,

Ketua Program Studi

Digitally signed by Lionov

Lionov, Ph.D.

PERNYATAAN

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

PENYELESAIAN *TRAVELING SALESMAN PROBLEM* MENGGUNAKAN ALGORITMA *FRUITFLY 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 11 Juli 2024

Darren Lee
NPM: 6181801009

ABSTRAK

Traveling Salesman Problem (TSP) merupakan sebuah permasalahan di mana seseorang ingin bepergian ke setiap kota yang terdapat dalam peta masing-masing satu kali, dan kembali ke kota asalnya dengan bobot serendah mungkin. Untuk mencari rute yang memiliki bobot yang paling rendah ini tentunya dapat dilakukan dengan menghitung segala jenis kemungkinan rute yang ada, namun semakin banyak kota dalam peta tersebut, maka semakin lama waktu yang dibutuhkan untuk mencari rute terbaik tersebut dan semakin banyak kemungkinan rute yang menjadi pertimbangan untuk rute yang memiliki bobot paling rendah.

Maka dari itu, pendekatan optimisasi akan dilakukan untuk mencari rute dengan bobot paling optimal dengan menggunakan waktu yang relatif lebih sedikit. Optimal yang dimaksud di sini adalah sebuah keadaan di mana hasil yang didapatkan paling mendekati hasil yang terbaik. Dengan demikian, tidak perlu diperhitungkan segala kemungkinan yang ada. *Fruitfly Optimization Algorithm* (FOA) merupakan sebuah algoritma optimisasi yang meniru pergerakan dari lalat buah untuk mencari makanan. Lalat buah ini akan mencari makanan dalam kawanan. Lalat-lalat buah yang terdapat dalam suatu area akan berusaha berkumpul dengan kawanannya sebelum mencari makanan dengan *smell concentration value* yang paling tinggi di setiap kawanan lalat buah tersebut. Calon solusi / rute paling optimal dalam TSP dapat dimodelkan menjadi lalat buah dalam FOA. Titik atau kota dalam peta dalam TSP dapat dimodelkan dengan arah pergerakan kawanan lalat buah pada FOA. Bobot dari perjalanan dalam TSP ini dapat dimodelkan dengan *smell concentration value*. Semakin sedikt bobotnya maka semakin tinggi *smell concentration value* tersebut.

Dalam penelitian ini akan dibuat sebuah perangkat lunak berbasis *web* yang menggunakan *web framework* Django. *Web framework* merupakan sebuah kumpulan dari peralatan yang dapat meringankan kesulitan untuk penyelesaian persoalan dalam pengembangan jaringan. Perangkat lunak ini akan menerima masukan berupa variabel-variabel seperti iterasi pengerjaan FOA, banyaknya calon solusi, konstanta Vr, banyaknya iterasi operasi mutasi *bit*, jenis *sensitive vision function*, dataset beserta jenis datasetnya. Setelah menerima masukan, perangkat lunak akan memproses masukan dan mengeluarkan pemetaan titik-titiknya dalam sebuah graf, dan akan ditampilkan hasil dari rute yang paling optimal beserta visualisasi grafnya dan total bobot yang diperlukan untuk perjalanan tersebut.

Kinerja FOA dalam menyelesaikan TSP akan diuji dengan memasukkan berbagai variasi pada variabel-variabel masukan yang telah disebut sebelumnya, yaitu, banyaknya titik pada dataset yang akan diuji, *sensitive vision function* yang digunakan, konstanta Vr yang digunakan, banyaknya iterasi operasi mutasi *bit*, banyaknya calon solusi yang paling optimal, dan banyaknya iterasi penggunaan FOA. Kinerja FOA dalam menyelesaikan TSP lebih baik jika jumlah titik pada dataset sedikit, penggunaan *sensitive vision function* tertentu, semakin dekat konstanta Vr ke 50%, banyaknya populasi, dan iterasi penggunaan FOA yang cukup banyak.

Kata-kata kunci: *Traveling Salesman Problem*, *Fruitfly Optimization Algorithm*, *framework*, Django, *smell concentration value*, *sensitive vision function*

ABSTRACT

Traveling Salesman Problem is a problem in which someone wanted to travel within a set of cities in a particular map, visiting once for every city, and return to his/her starting point with the lowest possible cost. To find the lowest cost, it is possible to calculate all possible routes, but the time required to find the best solution will significantly increase and the the amount of solutions to be put into considerations increases as well.

Therefore, optimization approach is done to find the most optimal route using relatively shorter time. Optimal in this circumstance is defined to obtain a result which is relatively close to the best solution. Thus, there will be no need to consider every possibility of results. Fruitfly Optimization Algorithm (FOA) is an optimization algorithm which inspired from the movements of fruitflies. These fruitflies will look for food in swarms. The fruitflies in a particular area will try to gather with the swarm before going for the food in the area with the highest smell concentration value amongst the swarms. The optimal solution candidates for TSP can be modelled by the fruitflies in FOA. The points/cities in the map in TSP can be modelled by the swarm of fruitflies in FOA. The cost for the journey in TSP can be modelled by the smell concentration value. The lower the cost, the higher the smell concentration value will be.

For this research, a web-based software will be made using the Django web framework. A web framework is a collection of tools to help reduce the difficulty in network development and solving its problems. The software will receive input in form of variables such as FOA iteration, the amount of solution candidates, Vr constants, bit mutation operation iterations, variant of sensitive vision function, the datasets with their types. After receiving the input, the software will process the inputs and will generate the initial mapping for the points and the mapping result, along with the most optimal route and its cost needed for the journey.

The performance of FOA to solve TSP will be examined with variations of the inputs stated previously, which are the points that will be calculated in the datasets, sensitive vision function used, Vr constant used, the amouunt of bit mutation operation iterations, the amount of most optimal solution candidates, and the amount of FOA iterations. The performance of FOA to solve TSP will be better if the amount of points that will be calculated in the datasets is relatively small, certain type of sensitive vision function, the closer the Vr constant to 50%, population size, and FOA iterations.

Keywords: Traveling Salesman Problem, Fruitfly Optimization Algorithm, framework, Django, smell concentration value, sensitive vision function

Untuk keluarga tercinta

KATA PENGANTAR

Puji syukur penulis panjatkan untuk Tuhan Yang Maha Esa hanya karena kehendakNya dan karuniaNya penulis dapat diizinkan untuk menyelesaikan tugas akhir ini yang merupakan syarat kelulusan untuk memperoleh gelar sarjana. Berbagai hambatan dan tantangan tentu penulis hadapi dalam pengerjaan tugas akhir ini, tetapi pada akhirnya tantangan-tantangan ini tetap dapat dilalui dan dapat diselesaikan dengan baik. Maka dari itu, penulis hendak berterima kasih secara khusus kepada:

1. Tuhan Yang Maha Esa
2. Keluarga yang selalu memberi dukungan selama penulis mengerjakan tugas akhir
3. Bapak Husnul Hakim selaku dosen pembimbing yang telah membimbing penulis mengerjakan tugas akhir ini
4. Maichel Yunarto sebagai teman yang sudah bersedia menyediakan waktunya dan tanpa pamrih membimbing dan mengajarkan beberapa materi yang diperlukan untuk menyelesaikan tugas akhir ini
5. Kelompok teman dekat dari Bandung-BSD, yaitu William Horeb, Posma Salomo Joy Nielsen, Christopher Nathanael, Patrick, Ivan Chris, Mark Tito Randa, dan Kiki Yohanes
6. Jonathan Axelrod, Kelvin Hidayat, dan Gihond Uloski sebagai teman dari gereja yang sudah menemani penulis ketika penulis mengalami kesulitan dalam masa pengerjaan tugas akhir ini
7. Teman kampus, yaitu M.Kenny Rizky, Chris Ardiansyah, Fransiskus Putra dan Christopher William yang sudah bersedia menjadi teman mengobrol penulis.
8. Pemilik dari Fin's COffee, yaitu Ivan Limosi, Nicholas Khrisna, dan Clement Rivandi yang sudah menyediakan tempat untuk penulis mengerjakan tugas akhir
9. Teman-teman lain komunitas gereja dari GII Setrasari Arrows dan Flare yang tidak dapat penulis sebutkan satu-satu namanya tetapi yang sudah membantu penulis dalam dukungan doa
10. Komunitas *trading card game* dari UNPAR dan Mishraworkshop.
11. Orang-orang lain yang tidak dapat disebutkan namanya di sini

Penulis berharap hasil dari tugas akhir ini dapat berguna bagi orang lain di kemudian hari. Selain itu, penulis juga memohon maaf apabila terdapat kesalahan dalam tugas akhir ini, ataupun ada kata-kata yang menyinggung dari pihak manapun. Penulis juga sangat terbuka untuk kritik dan saran yang membangun. Penulis juga berharap agar karya ini dapat menjadi inspirasi dan referensi bagi orang lain sehingga dapat menciptakan karya yang lebih baik dari penulis.

Bandung, Juli 2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE PROGRAM

BAB 1

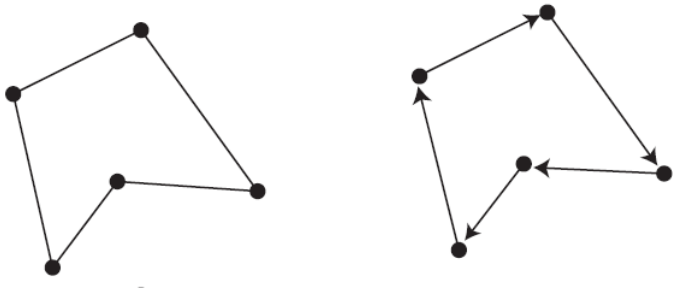
PENDAHULUAN

1.1 Latar Belakang

Menurut Paluch (2009), [4] *Traveling Salesman Problem* (TSP) termasuk sebagai permasalahan optimasi kombinatorial. Permasalahan optimasi kombinatorial merupakan sebuah permasalahan di mana penyelesaiannya adalah mencari sebuah objek optimum dalam sekumpulan objek. Biasanya objek ini direpresentasikan secara singkat, tetapi jelas melalui sebuah graf, di mana jumlah objek ini besar sehingga dengan mencari objek ini satu per satu tidak mungkin dilakukan [5]. Objek yang dimaksud di sini adalah solusi yang dihasilkan dari penyelesaian permasalahan tersebut. TSP merupakan sebuah permasalahan di mana seseorang ingin bepergian ke berbagai kota dan ingin mencari rute yang paling optimal untuk mendatangi kota-kota tersebut dengan jalan termudah dan termurah tepat satu kali [6] [4]. Berdasarkan penelitian yang dilakukan oleh Gilbert Laporte, permasalahan ini pertama kali ditemukan oleh Euler pada tahun 1759, lalu diformulasikan pada abad ke-19 oleh seorang ahli matematika asal Irlandia bernama William Rowan Hamilton dan ahli matematika asal Inggris bernama Thomas Kirkman [7].

TSP secara umum dapat dimodelkan dalam graf. Graf ini akan berisi titik-titik yang akan dihubungkan dengan titik-titik lainnya, yang kemudian akan dicari rute yang optimal untuk seorang *salesman* bepergian dari satu titik, mendatangi setiap titik tepat satu kali, dan kembali ke titik awal. Pertama akan ditentukan titik yang menjadi titik awal. Kemudian, titik awal ini akan dihubungkan dengan titik-titik lainnya dalam graf, melewati setiap titik dalam graf tersebut tepat satu kali, dan kembali ke titik awal tersebut dengan biaya terendah. Bentuk dari biaya ini adalah bobot dari sisi yang menghubungkan sebuah titik dengan titik lain. Bobot ini bisa dalam berbagai bentuk. Beberapa contohnya adalah antara lain jarak dari antartitik atau kota, jumlah uang yang perlu dikeluarkan untuk perjalanan antarkota, waktu yang ditempuh untuk bepergian antarkota, dan sebagainya. TSP dibagi menjadi dua jenis, yaitu simetris dan asimetris [8]. Gambar untuk dua jenis TSP ini terdapat pada Gambar 1.1 [7]. TSP simetris adalah sebuah permasalahan di mana sisi di antara dua titik tidak memiliki arah khusus. Struktur dasar pada graf untuk menggambarkan TSP ini adalah graf tidak berarah. Pada TSP asimetris, jarak pada dua titik dapat berbeda untuk kedua arah, dan ada kemungkinan hanya terdapat satu arah untuk ke titik tertentu saja. Struktur dari graf ini adalah berarah. Contoh dari TSP asimetris ini adalah pada bagian daerah tertentu di dalam graf peta terdapat beberapa ruas jalan yang hanya dapat dilalui satu arah.

Berdasarkan penelitian yang dilakukan oleh Lan Huang, Gui-chao Wang, Tian Bai, dan Zhe Wang, [9], ada 2 cara untuk menyelesaikan TSP. Cara pertama adalah dengan metode eksak untuk mendapatkan hasil yang optimal, seperti metode *branch-and-bound*(Lawler dan Wood, 1966) [10]. Namun karena seiring dengan meningkatnya banyak kota, menurut John D. C. Little, Katta G. Murty, Dura W. Sweeney dan Caroline Kare (1963) [11], maka kompleksitas waktu untuk penyelesaian ini akan menaik secara eksponensial sehingga metode *branch-and-bound* hanya cocok dilakukan untuk kasus skala kecil. Metode *branch-and-bound* adalah sebuah metode pencarian di mana sekumpulan objek yang akan dicari akan dipisahkan secara bertahap dengan sebuah batasan untuk mendapatkan objek yang dicari, dan mengeliminasi objek-objek yang melebihi batasan tersebut. [10] Dalam pencarian hasil ini akan ditentukan sebuah batasan yang merupakan sebuah



Gambar 1.1: Contoh TSP
(Kiri) TSP simetris (Kanan) TSP asimetris

biaya untuk mempersempit kemungkinan solusi yang didapat dengan dilakukan partisi untuk setiap pencarian yang dilakukan. Partisi yang dimaksud di sini adalah pemisahan di antara objek-objek yang akan dicari dan yang akan dieliminasi dari pencarian karena sudah melebihi batasan biaya. Setelah setiap partisi, objek-objek yang melebihi dari batas biaya akan dieliminasi. Pencarian akan dilakukan terus dan berhenti sampai sebuah hasil ditemukan di mana biayanya tidak akan lebih besar dari batasan.

Cara kedua yaitu menggunakan metode pendekatan yang dapat mengembalikan hasil yang tidak pasti paling optimal tetapi akan menggunakan waktu yang lebih sedikit. Optimal di sini merupakan sebuah keadaan di mana hasil yang didapatkan yang paling mendekati dengan hasil yang terbaik, sehingga tidak perlu diperhitungkan segala kemungkinan yang ada untuk mendapat hasil terbaik sehingga meningkatkan efisiensi waktu. Contoh dari penggunaan metode pendekatan ini adalah dengan penggunaan algoritma metaheuristik. Algoritma metaheuristik merupakan sebuah algoritma yang menggunakan kerangka algoritma sederhana, yang pada umumnya terinsipirasi dari alam, yang dirancang untuk mencari solusi dan menyelesaikan permasalahan optimisasi yang kompleks [12]. Algoritma metaheuristik cocok untuk membangun algoritma untuk menemukan solusi yang optimal dengan waktu eksekusi yang terjangkau. Contoh untuk penggunaan dari algoritma metaheuristik yang paling populer adalah metode-metode yang terinspirasi dari alam, terutama yang berdasarkan dari kecerdasan kawanan (*swarm intelligence*). Algoritma yang menggunakan kecerdasan kawanan ini disebut algoritma *swarm intelligence*. Algoritma *swarm intelligence* ini merupakan sebuah algoritma yang meniru fenomena dari alam secara fisik atau biologis yang digunakan untuk menyelesaikan permasalahan optimisasi. Pembuatan keputusan untuk menyelesaikan masalah ini dibuat oleh pelaku yang merupakan makhluk hidup seperti semut, lebah, dan makhluk hidup lainnya yang berkawanan. Sebuah contoh dari algoritma *swarm intelligence* adalah algoritma *fruitfly optimization* (FOA). FOA termasuk dalam sejenis algoritma *swarm intelligence* karena lalat buah termasuk makhluk hidup yang hidup dalam kawanan untuk menyelesaikan permasalahannya, salah satu contohnya adalah mencari makan.

FOA merupakan sebuah algoritma optimisasi yang terinspirasi dari perilaku lalat buah. Menurut Takeshi Shimada dari Universitas Tokyo, lalat buah merupakan makhluk hidup yang dapat dilihat sebagai makhluk hidup kedua dari yang paling sederhana dari segi sistem saraf karena lalat buah memiliki indra yang cukup terbatas karena hanya memiliki ratusan neuron dan tidak mempunyai otak [13]. Mereka terangsang pada buah-buahan yang matang pada musim panas. Lalat-lalat tersebut dapat mencium makanan sejauh 40km. Lalu, dengan menggunakan organ *osphresis*nya, mereka akan mencari keberadaan makanan, lalu akan mencoba mencari kawanan lainnya, setelah itu terbang ke arah makanan [14] [15].

TSP merupakan sebuah permasalahan NP-*hard*(*non-deterministic polynomial time - hard*), artinya sebuah permasalahan di mana algoritma yang digunakan untuk menyelesaikan permasalahan tersebut membutuhkan waktu polinomial, tetapi tidak dapat menjamin dapat menemukan solusi terbaik. Jika jumlah objek yang menjadi perbandingan untuk mencari solusi terbaik (jika dalam

kasus TSP objek tersebut merupakan titik-titik yang perlu dikunjungi) sedikit, maka akan mudah untuk mencari solusi terbaik dengan metode eksak. Namun jika jumlah titik-titik ini banyak, maka akan sulit untuk mendapatkan solusi terbaik dengan metode eksak karena dengan melakukan metode eksak ini setiap kemungkinan solusi yang ada akan dihitung, maka efisiensi dari pencarian solusi terbaik ini akan berkurang karena waktu yang dibutuhkan lebih banyak dengan menggunakan metode eksak. Maka dari itu, penggunaan metode pendekatan seperti penggunaan algoritma metaheuristik lebih baik digunakan karena meskipun algoritma metaheuristik tidak dapat dipastikan mendapatkan solusi terbaik, tetapi masih bisa mendapatkan solusi yang mendekati solusi terbaik. Solusi pendekatan ini didapat dengan menambah efisiensi pencarian solusi karena kompleksitas waktu untuk metode eksak adalah eksponensial, sedangkan kompleksitas waktu untuk metode pendekatan adalah polinomial. Karena TSP merupakan sebuah permasalahan NP-*hard*, salah satu algoritma yang cocok digunakan untuk menyelesaikan permasalahan dalam tugas akhir ini adalah FOA.

Graf yang akan digunakan dalam TSP bersifat *complete*, yang berarti setiap titik akan bertetangga dan terhubung dengan sisi. Selain itu, graf yang akan digunakan dalam TSP juga bersifat sederhana (sebuah graf yang mana 2 titik yang berbeda hanya dihubungkan dengan paling banyak satu sisi), terhubung (sebuah graf yang mana semua titik dalam graf tersebut terhubung setidaknya dengan sebuah sisi yang dapat membuat sebuah lintasan), dan berbobot (setiap sisi dari graf tersebut memiliki sebuah bilangan positif yang menjadi bobot dari graf tersebut) [3].

Pada tugas akhir ini dibangun sebuah perangkat lunak untuk menyelesaikan TSP dengan menggunakan FOA. Perangkat lunak menerima masukan berupa dataset yang berisi koordinat x dan y dari setiap titik atau matriks ketetanggaan jarak dari setiap titik, yang kemudian akan dihasilkan sebuah graf yang merepresentasikan TSP secara acak oleh perangkat lunak sebagai calon-calon solusi paling optimal. Dataset ini dapat diperoleh dari https://people.sc.fsu.edu/ jburkardt/datasets/tsp/tsp.html dan http://www.math.uwaterloo.ca/tsp/data/. Keluaran yang dihasilkan oleh perangkat lunak adalah rute yang optimal dan bobot dari rute yang paling optimal. Pengujian dari perangkat lunak ini melibatkan pencarian solusi paling optimal dengan mencari bobot yang paling rendah di antara kemungkinan-kemungkinan calon solusi paling optimal yang telah dihasilkan oleh perangkat lunak secara acak. Perangkat lunak yang dibuat menggunakan *web framework* Django [2].

1.2 Rumusan Masalah

Berdasarkan deskripsi di atas, terdapat rumusan masalah sebagai berikut.

- Bagaimana FOA dapat membantu menyelesaikan TSP?
- Bagaimana cara membangun sebuah perangkat lunak untuk menyelesaikan TSP menggunakan FOA?
- Bagaimana kinerja FOA untuk menyelesaikan TSP?

1.3 Tujuan

- Mempelajari bagaimana cara FOA membantu menyelesaikan TSP.
- Membangun sebuah perangkat lunak untuk menyelesaikan TSP menggunakan FOA.
- Melakukan pengujian untuk menilai kinerja FOA untuk menyelesaikan TSP.

1.4 Batasan Masalah

Bobot yang menjadi pertimbangan dalam penyelesaian TSP dalam penelitan ini adalah jarak Euclidean antarkota. Peta dan rute hanya dapat ditampilkan jika jenis data yang menjadi masukannya memiliki titik beserta koordinatnya. Untuk jenis data yang merupakan matriks ketetanggaan

dari jarak antartitik tidak ditampilkan visualisasi grafnya karena proses penghasilan posisi titik secara acak berdasarkan matriks ketetanggaan melibatkan perhitungan yang rumit sehingga tidak dapat dikerjakan.

1.5 Metodologi

Metodologi yang dilakukan dalam tugas akhir ini adalah sebagai berikut.

- Melakukan studi literatur tentang TSP
- Melakukan studi literatur tentang FOA
- Melakukan studi literatur tentang penggunaan FOA dalam TSP
- Melakukan analisis permasalahan dan kebutuhan perangkat lunak
- Melakukan perancangan perangkat lunak untuk penyelesaian TSP menggunakan FOA
- Membangun perangkat lunak untuk menyelesaikan TSP dengan FOA
- Melakukan pengujian perangkat lunak untuk menilai kinerja FOA untuk menyelesaikan TSP
- Menulis dokumen tugas akhir

1.6 Sistematika Pembahasan

Dokumen tugas akhir ini terdiri dari 6 bab. Bab pertama berisi tentang latar belakang masalah. Bab 2 berisi tentang landasan teori untuk tugas akhir ini. Bab 3 berisi tentang analisis permasalahan dalam tugas akhir ini dan sebuah contoh kasus untuk menyelesaikan permasalahan dalam tugas akhir ini. Bab 4 berisi tentang perancangan dan penjelasan dari perangkat lunak yang dibangun untuk menyelesaikan masalah dalam tugas akhir ini. Bab 5 berisi penjelasan untuk pengujian dan implementasi dari perangkat lunak yang telah dibuat untuk menyelesaikan permasalahan dalam tugas akhir ini. Bab 6 berisi kesimpulan dari tugas akhir ini. Kode dari perangkat lunak juga diikutsertakan dalam dokumen tugas akhir ini pada bagian Lampiran.