

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada penelitian ini, algoritma lebah digunakan sebagai metode untuk memecahkan permasalahan TSP (*Travelling Salesman Problem*), yang mana dengan metode tersebut diharapkan didapatkan rute atau urutan yang paling optimal dengan meminimasi tiap-tiap rute yang ada sehingga didapatkan total jarak yang terpendek.

Penelitian sebelumnya yaitu berjudul “*Implementasi Algoritma Lebah Untuk Pencarian Jalur Terpendek Dengan Mempertimbangkan Heuristik*” (Nurdiana: 2010). Pada penelitian tersebut, pencarian jalur terpendek dibangun dengan memanfaatkan teorema graf dalam representasi datanya, kemudian dalam pencarian jalur terpendeknya juga mempertimbangkan heuristik, yang mana meliputi faktor kemacetan, faktor lampu jalan, faktor jalan tol, faktor rawan kecelakaan, dan faktor keamanan. Sehingga rute jalur yang direkomendasikan merupakan jalur yang optimum untuk dilalui sesuai dengan heuristik yang dipilih.

Penelitian tentang algoritma lebah juga pernah dilakukan yakni berjudul “*Implementasi Algoritma Optimasi Bee Colony Untuk Penjadwalan Job Shop*” (Nafiuna: 2010). Pada penelitian ini algoritma lebah digunakan sebagai suatu metode dalam manajemen perusahaan, agar perusahaan tersebut dapat berjalan dengan efektif dan efisien. Fungsi tujuan dari penyelesaian permasalahan penjadwalan *job shop* adalah meminimalkan *makespan*. Jika semakin minimum

waktu mulai dari operasi yang paling akhir, maka semakin minimum juga waktu selesai dari keseluruhan operasi yang ada (*makespan*).

2.2 Travelling Salesman Problem (TSP)

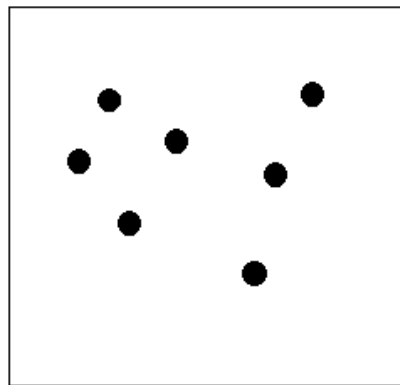
Travelling salesman problem (TSP) adalah permasalahan yang sangat terkenal dalam Teori Graf. Permasalahan pada TSP adalah mencari sirkuit terpendek pada suatu graf tidak berarah yang berasal dari suatu simpul dengan melewati seluruh simpul dan kembali ke simpul asal. Sirkuit ini disebut juga sirkuit Hamilton.

Travelling Salesman Problem (TSP) pertama kali diperkenalkan oleh RAND pada tahun 1984. Perusahaan itulah yang memperkenalkan metode TSP menjadi persoalan yang populer dan saat ini menjadi salah satu bagian dalam ilmu riset operasi (*linear programming*) dan dijadikan dasar dalam penyelesaian persoalan kombinasi (*combinational problem*). TSP secara sederhana dapat didefinisikan sebagai proses pencarian lintasan terefisien dan terpendek dari beberapa kota yang dipresentasikan. *Travelling Salesman Problem* megibaratkan pelibatan seorang *travelling salesman* yang harus melakukan kunjungan ke sejumlah kota dalam menjajakan produknya. Rangkaian kota-kota yang dikunjungi harus membentuk suatu jalur sedemikian sehingga kota-kota tersebut hanya boleh dilewati tepat satu kali dan kemudian kembali lagi ke kota awal. Persoalan yang dihadapi TSP adalah bagaimana merencanakan total jarak minimum. Untuk menyelesaikan persoalan tersebut, tidak mudah dilakukan karena terdapat ruang pencarian dari sekumpulan

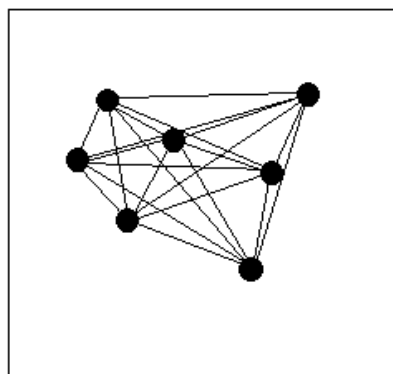
permutasi sejumlah kota. Maka TSP kemudian dikenal dengan persoalan Non Polinomial.

Penyelesaian eksak terhadap masalah TSP mengharuskan untuk melakukan perhitungan terhadap semua kemungkinan rute yang dapat diperoleh, kemudian memilih salah satu rute yang terpendek. Untuk itu jika terdapat n kota yang harus dikunjungi, maka diperlukan proses pencarian sebanyak $n!/2n$ rute.

Gambaran sederhana tentang definisi dan pengertian TSP adalah sebagai berikut,



Gambar 2.1 Posisi Kota Yang Ditentukan

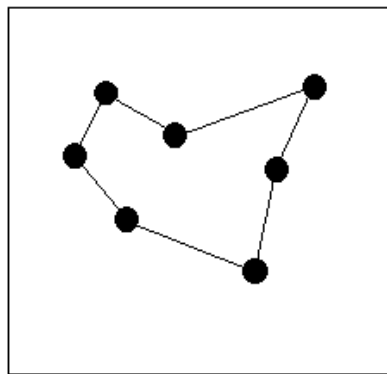


Gambar 2.2 Rute Yang Mungkin Dilewati

Kota-kota pada gambar diatas masing-masing mempunyai koordinat (x,y), sehingga jarak antar kedua kota dapat dihitung dengan rumus:

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.1)$$

Setelah jarak-jarak yang menghubungkan tiap kota diketahui, maka dicari rute terpendek dari jalur yang akan dilewati untuk kembali ke kota awal.



Gambar 2.3 Posisi Kota Yang Akan Dilewati

2.3 Teori Graf

2.3.1 Definisi Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges*). Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E.

- a. Verteks (simpul) :V = himpunan simpul yang terbatas dan tidak kosong
- b. Edge (sisi/busur):E = himpunan busur yang menghubungkan sepasang simpul.

Simpul-simpul pada graf dapat merupakan obyek sembarang seperti kota, atom-atom suatu zat, nama anak, jenis buah, komponen alat elektronik dan sebagainya. Busur dapat menunjukkan hubungan (relasi) sembarang seperti rute penerbangan,

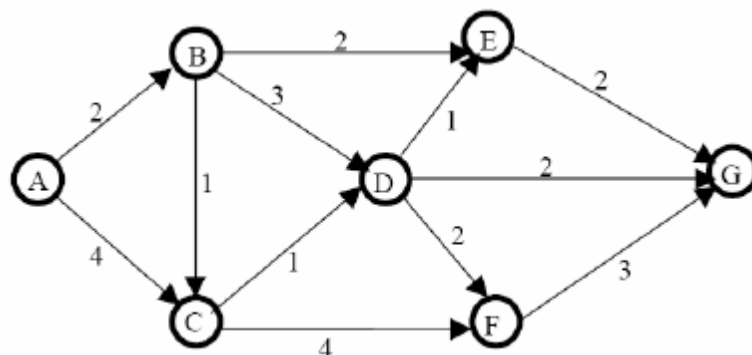
jalan raya, sambungan telepon, ikatan kimia, dan lain-lain. Notasi graf: $G(V,E)$ artinya graf G memiliki V simpul dan E busur.

2.3.2 Macam-macam Graf

Menurut arah dan bobotnya, graf dibagi menjadi empat bagian, yaitu :

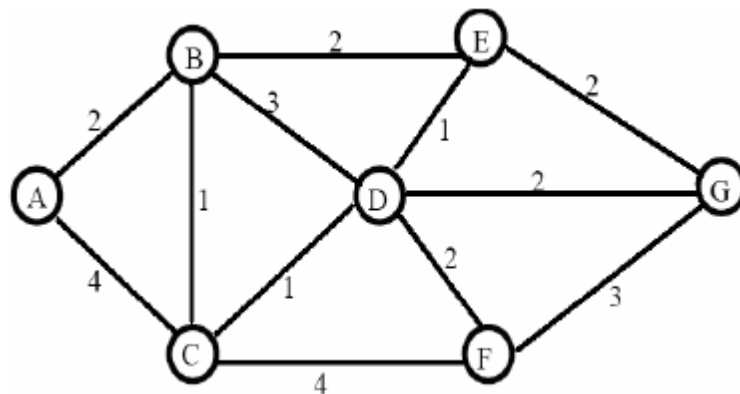
1. Graf berarah dan berbobot : tiap busur mempunyai anak panah dan bobot.

Gambar 2.4 menunjukkan graf berarah dan berbobot yang terdiri dari tujuh titik yaitu titik A,B,C,D,E,F,G. Titik menunjukkan arah ke titik B dan titik C, titik B menunjukkan arah ke titik D dan titik C, dan seterusnya. Bobot antar titik A dan titik B pun telah di ketahui.



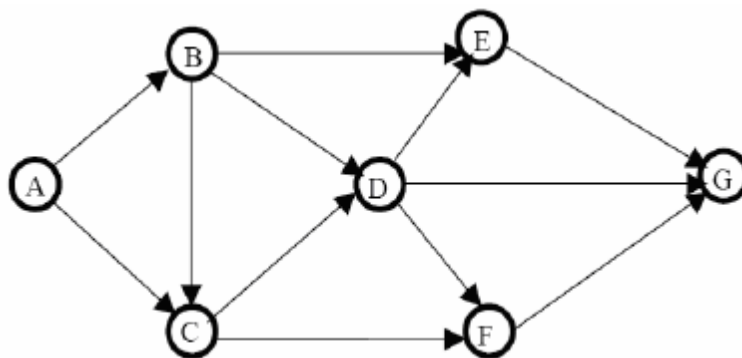
Gambar 2.4 Graf berarah dan berbobot

2. Graf tidak berarah dan berbobot : tiap busur tidak mempunyai anak panah tetapi mempunyai bobot. Gambar 2.5 menunjukkan graf tidak berarah dan berbobot. Graf terdiri dari tujuh titik yaitu titik A,B,C,D,E,F,G. Titik A tidak menunjukkan arah ke titik B atau C, namun bobot antara titik A dan titik B telah diketahui. Begitu juga dengan titik yang lain



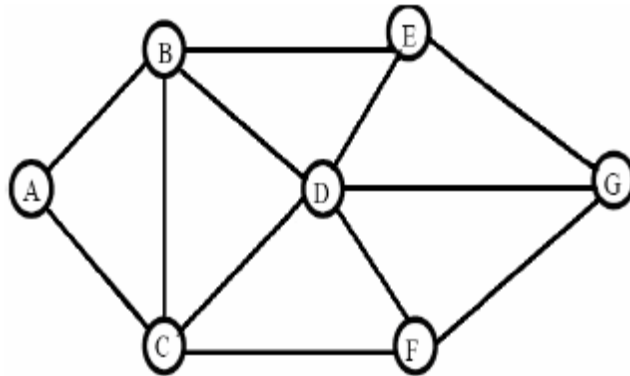
Gambar 2.5 Graf tidak berarah dan berbobot

3. Graf berarah dan tidak berbobot: tiap busur mempunyai anak panah yang tidak berbobot. Gambar 2.6 menunjukkan graf berarah dan tidak berbobot.



Gambar 2.6 Graf berarah dan tidak berbobot

4. Graf tidak berarah dan tidak berbobot: tiap busur tidak mempunyai anak panah dan tidak berbobot.



Gambar 2.7 Graf tidak berarah dan tidak berbobot

2.3.3 Representasi Graf

1. Matriks Kedekatan (*Adjacency Matrix*)

Untuk suatu graf dengan jumlah simpul sebanyak n , maka matriks kedekatan mempunyai ukuran $n \times n$ (n baris dan n kolom). Jika antara dua buah simpul terhubung maka elemen matriks bernilai 1, dan sebaliknya bernilai 0 jika tidak terhubung. Tabel matriks kedekatan untuk graf ABCDEFG dapat dilihat pada tabel berikut,

Tabel 2.1 Matriks kedekatan graf ABCDEFG

	A	B	C	D	E	F	G
A	0	1	1	0	0	0	0
B	1	0	1	1	1	0	0
C	1	1	0	1	0	1	0
D	0	1	1	0	1	1	1
E	0	1	0	1	0	0	1
F	0	0	1	1	0	0	1
G	0	0	0	1	1	1	0

Pada tabel diatas, elemen matriks kedekatan bernilai 0 untuk diagonal dan elemen yang tidak terhubung dengan simpul lain (elemen matriks bernilai 0 jika simpul tidak terhubung dengan simpul lainnya).

Ruang (memori) yang diperlukan untuk matriks kedekatan adalah sebagai berikut :

$$n \times n = (N^2) \quad (2.2)$$

Untuk graf yang tidak berarah yakni sebagai berikut:

- Ruang yang diperlukan $= \frac{1}{2} (N^2) - \frac{1}{2} (N)$
- Derajat simpul $i = \sum_{j=1}^n A[i, j]$

Untuk graf berarah ialah:

- Derajat luar (*out degree*) = jumlah dalam 1 baris (matriks) atau banyaknya busur dengan simpul V sebagai kepala.
- Derajat dalam (*in degree*) = jumlah dalam 1 kolom (matriks) atau banyaknya busur dengan simpul V sebagai ekor.

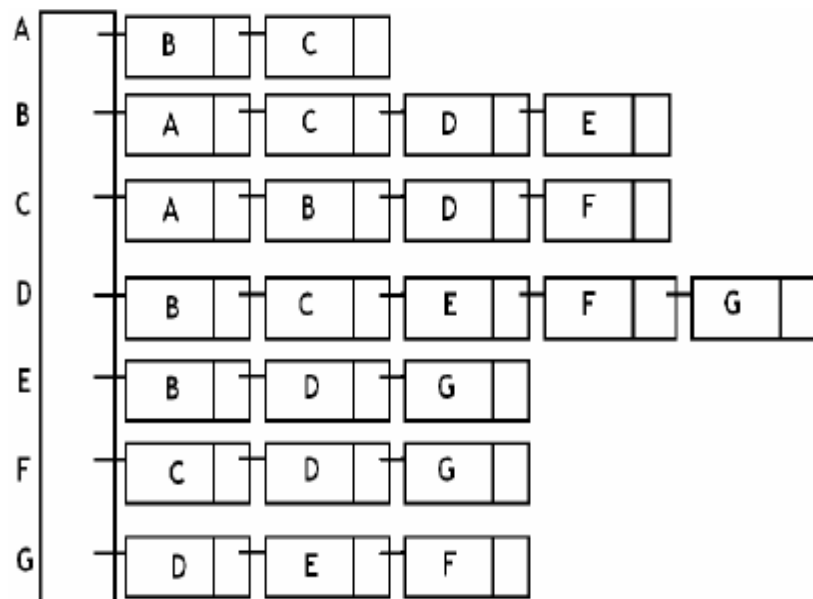
2. Senarai Kedekatan

Pada simpul x dapat dianggap sebagai suatu senarai yang terdiri dari simpul pada graf yang berdekatan dengan x. Representasi senarai kedekatan mempunyai kesamaan fleksibilitas dengan matriks kedekatan, akan tetapi representasi ini lebih tersusun rapi. Ruang (memori) yang

diperlukan untuk n simpul dan e sisi pada graf tidak berarah = n head node
+ $2e$ node list.

Senarai kedekatan untuk graf ABCDEFG dapat dilihat pada gambar

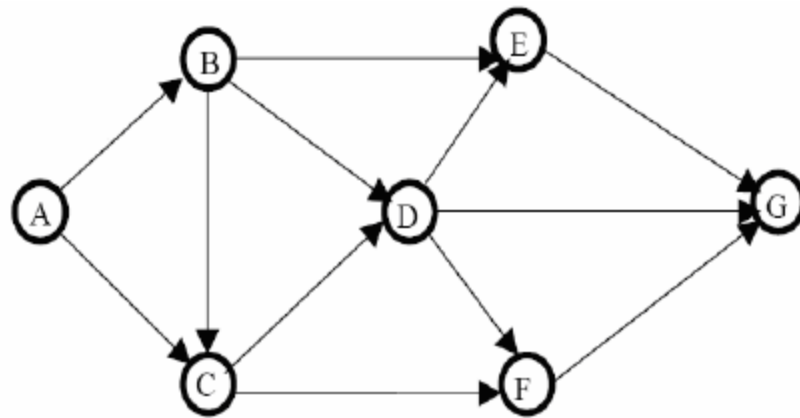
2.8 :



Gambar 2.8 Senarai kedekatan graf ABCDEFG

2.4 Masalah Lintasan Terpendek

Persoalan lintasan terpendek pada dasarnya adalah merupakan suatu persoalan untuk mencari jarak terpendek yang setidaknya melibatkan dua buah titik yang membentuk suatu simpul pada graf berbobot, yang mana bobot tersebut dapat digunakan nilai jarak antar kota, atau jarak dari tiap node yang terhubung. Dimisalkan terdapat graf sebagai berikut,



Gambar 2.9 Graf ABCDEFG

Dari gambar diatas maka akan dapat dihitung jarak antar titik yakni satu ke titik lainnya. Jarak antar titik yang belum diketahui tersebut dapat dihitung berdasarkan koordinat yang dimiliki oleh tiap-tiap titik yang ada. Dengan demikian maka jalur terpendek akan diketahui dengan melihat bobot jarak total yang dihasilkan.

2.5 *Swarm Intelligence*

Swarm Intelligence adalah salah satu teknik kecerdasan buatan yang berlandaskan kepada perilaku kolektif (*collective behaviour*) pada sistem yang terdesentralisasi dan dapat mengatur dirinya sendiri (*self-organizing*).

Sistem yang memanfaatkan *Swarm Intelligence* biasanya merupakan sebuah populasi yang terdiri atas anggota berupa agen-agen yang sederhana, yang berinteraksi secara lokal dengan sesama anggota, dan juga berinteraksi dengan lingkungan. Walaupun pada umumnya tidak ada struktur kendali secara terpusat (*centralized*) yang mendikte bagaimana masing-masing individu bertindak, namun

interaksi secara lokal (di antara anggota) seringkali menuju pada pembentukan (*emergence*) perilaku global. Contoh dari sistem seperti ini dapat ditemukan pada alam, misalnya koloni semut, burung-burung, ikan, dan juga termasuk koloni lebah. Secara implisit, hal ini menyatakan bahwa sistem koloni tidak bersifat linear, ataupun kumulatif. Dengan kata lain, suatu kawanan dapat dipandang sebagai sebuah organisme sendiri. Suatu kawanan dapat memiliki sifat-sifat yang lebih dari sekedar penjumlahan linier dari sifat-sifat yang dimiliki oleh masing-masing anggotanya.

2.6 Algoritma Lebah

BCO (*Bee Colony Optimization*) Adalah suatu metode pencarian nilai optimal yang terinspirasi oleh kegiatan berupa pola dari kawanan lebah dalam mencari makanan. Dalam metode tersebut, perilaku cerdas tertentu dari sekawanan lebah madu berupa perilaku mencari makan ditinjau, dan sebuah algoritma baru dari koloni lebah buatan yang mensimulasikan perilaku lebah madu tersebut dijelaskan untuk memecahkan permasalahan optimisasi multidimensi dan multimodel.

Berdasarkan sifatnya, Lebah menggunakan segregasi sistematis untuk memastikan kelanjutan keberadaan dari koloninya. Mereka juga melakukan berbagai macam tugas seperti mencari makanan, reproduksi, mengurus lebah yang masih muda, patroli dan pembangunan sarang. Dari jumlah lebah yang banyak dalam sebuah koloni, mencari makan merupakan kegiatan utama yang dilakukan oleh koloni lebah untuk menjamin pasokan sumber makanan bagi koloni lainnya.

Lebah memiliki bahasa isyarat yang menjadi ciri utamanya yakni berupa tarian lebah, atau yang disebut dengan *waggle dance*. Tarian lebah (*waggle dance*) ini digunakan sebagai alat komunikasi yang digunakan oleh koloni lebah. Misalnya setelah lebah kembali dari sarangnya, ia akan melakukan tarian yang disebut dengan *waggle dance*. Dengan menggunakan tarian ini lebah yang lain menerima informasi mengenai sumber makanan. Informasi yang diberikan pada saat *waggle dance* adalah jarak dan arah untuk menuju sumber makanan yang telah ditemukan.

2.6.1 Pembangunan Jalur Lebah

Selama proses pencarian makanan, lebah akan melakukan perjalanan dari satu kota ke kota yang lain sampai mencapai tujuan. Menurut Wong, Low dan Chong (2008), lebah diperbolehkan untuk mengeksplorasi dan mencari jalan *tour* sampai selesai. Sebelum meninggalkan sarang lebah akan mengamati tarian yang dilakukan oleh lebah lainnya disarang. Kemudian lebah akan diset dengan pengetahuan yang didapatkan dari tarian tersebut. Set pergerakan yang dimaksud, yaitu "*preferred path*" yang dinotasikan dengan θ , maka akan berfungsi sebagai panduan dalam proses mencari makanan. θ berisi tur lengkap yang telah di eksplorasi sebelumnya oleh lebah yang akan pergi ke tempat tujuan. Dalam hal ini, aturan heuristik digunakan untuk bantuan lebah dalam pengambilan keputusan untuk kota yang akan dikunjungi berikutnya. Aturan ini terdiri dari dua faktor : *arc fitness* dan jarak. *Arc fitness* dihitung untuk semua path yang mungkin untuk kota kota yang bisa dikunjungi. *Acr fitness* yang lebih tinggi ditugaskan untuk tepi

yang merupakan pilihan jalur. Dengan melakukan ini, lebah cenderung memilih kota berdasarkan jalan pilihan. Disisi lain, dibawah pengaruh jarak heuristik lebah cenderung memilih kota berikutnya yang terdekat dengan kota saat ini.

2.6.2 Algoritma Lebah Untuk TSP

Tingkah laku lebah tersebut pada akhirnya dijadikan sebuah model untuk sistem belajar fungsional organisasi di tingkat grup karena sifat interaksinya dengan lingkungan sebagai suatu keseluruhan kohoren dan memiliki sejumlah adaptasi yang berfungsi untuk *group*. Langkah-langkah penyelesaian algoritma lebah adalah sebagai berikut:

1. *Forage*

Forage pada dasarnya adalah suatu perilaku lebah dalam mencari makanannya. Tahapan ini akan diberikan pada setiap lebah yang akan mengunjungi sumber makanan. Aturan ini diberlakukan ketika lebah dihadapkan pada beberapa pilihan *node*. Berikut ini adalah fungsi dari waktu proses operasi dan *arc fitness* yang ditampilkan pada *edge* yang terhubung.

$$P_{ij,n} = \frac{[\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}} \right]^{\beta}}{\sum_{j \in A_{i,n}} \left([\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}} \right]^{\beta} \right)} \quad (2.3)$$

Dimana:

ρ_{ij} = rata-rata sisi antara node i dan j

d_{ij} = jarak heuristik antara node i dan j

P_{ij} = kemungkinan pecabangan dari i dan j

P_{ij} mempunyai nilai berbanding terbalik dengan d_{ij} . Dengan kata lain, dibawah pengaruh jarak, lebah akan mengunjungi *node* dengan waktu proses yang lebih cepat. α merupakan variabel yang berperan sebagai pembobot untuk *arc fitness*, sedangkan β berfungsi untuk mengontrol signifikan level untuk *heuristic distance*-nya. Untuk pembobotan pada *arc fitness* pun terdapat aturan bahwa untuk *node* pilihan yang ada ternyata tersedia pada *preferred path*, maka *node* tersebut akan diberikan bobot yang paling tinggi. Sedangkan *node-node* pilihan yang lain akan diberi bobot dari rata-rata sisa pembobotan. Jumlah keseluruhan pembobotan adalah 1 untuk semua *node* yang ada dalam pilihan. Hal ini dapat dilihat persamaan di bawah ini,

$$P_{ij} = \begin{cases} \frac{\lambda}{1 - \lambda[Ai,n \cap Fi,n]} \\ \frac{1}{[Ai,n - Fi,n]} \end{cases} \quad (2.4)$$

λ menunjukan probabilitas dari sebuah kota yang diikuti pada θ . Fi, n adalah kumpulan yang berisi satu kota yang mana lebah lebih suka berpindah dari i pada n transisi, seperti yang direkomendasikan oleh θ . Pada $\theta(m)$ melambangkan elemen ke- m di θ .

2. Waggle Dance

Jika seekor lebah menari, tarian lebah akan berlangsung selama beberapa durasi. Durasi tarian Di dari lebah i ditentukan oleh linear fungsi yakni sebagai berikut,

$$Di = K \cdot \frac{Pfi}{Pf_{colony}} \quad (2.5)$$

Dari rumus diatas, durasi diukur selama iterasi pada algoritma yang dieksekusi. K didefinisikan sebagai skala faktor yang mengendalikan besarnya durasi. Menurut fungsi linear, jika lebah i memiliki Pfi yang lebih tinggi, maka akan diberikan kesempatan untuk menari lagi. Pfi melambangkan skor profitabilitas lebah i sebagaimana didefinisikan pada persamaan berikut,

$$Pfi = \frac{1}{Li} \quad Li = \text{tour length} \quad (2.6)$$

Pfi dapat ditafsirkan sebagai kuantitas nektar yang dikumpulkan oleh lebah i . Kuantitas yang lebih tinggi dari nektar akan dikumpulkan jika lebah melakukann perjalanan yang panjang dengan rute lebih pendek. Dengan demikian, Pfi didefinisikan berbanding terbalik dengan panjang tur.

Sedangkan Pf_{colony} menunjukkan koloni lebah dengan rata-rata profitabilitas dan diperbarui setelah masing-masing lebah menyelesaikan tur. Sebagaimana persamaan berikut ini,

$$Pf_{colony} = \frac{1}{N_{Bee}} \sum_{i=1}^{N_{bee}} Pfi \quad (2.7)$$

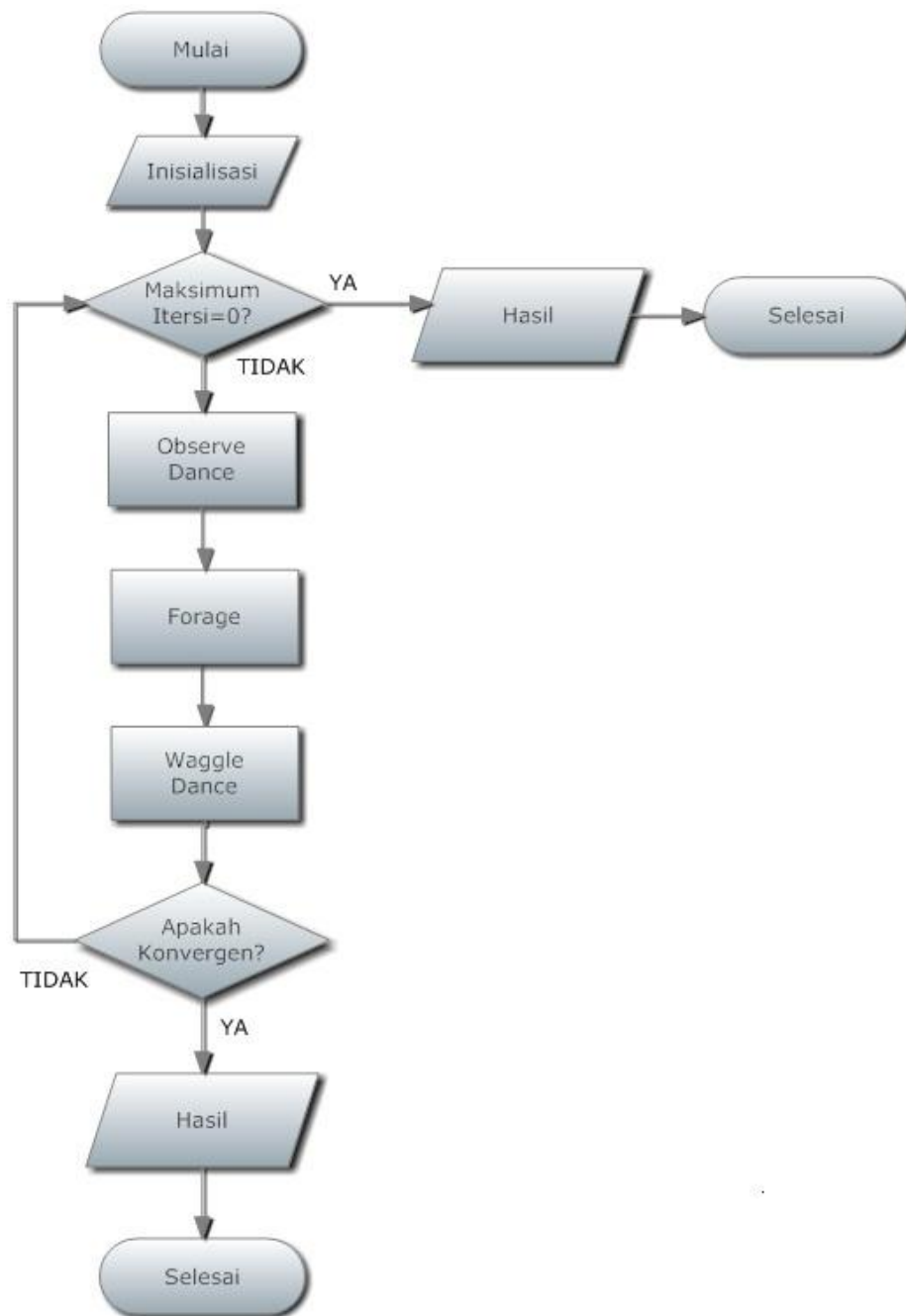
Pada dasarnya lebah lebih menyukai mengobservasi secara random dan mengikuti *waggle dance* dalam *dance floor* jika *profitability rating* rendah sebagai perbandingan terhadap koloni. *profitability rating* dari lebah dan koloni pada dasarnya seperti yang diperlihatkan di berikut ini, (diambil dari Nakrani 26 dan Tovey 2004).

Tabel 2.2 *Profitability rating*

Profitability Rating	Ri
$P_{fi} < 0.9P_{fcolony}$	0.60
$0.9P_{fcolony} \leq P_{fi} < 0.95P_{fcolony}$	0.20
$0.95P_{fcolony} \leq P_{fi} < 1.15P_{fcolony}$	0.02
$1.15P_{fcolony} \leq P_{fi}$	0.00

Dalam kasus tertentu, bilamana R_i bernilai nol, maka lebah akan memelihara *path*-nya sendiri. Lebah lebih suka melakukan pencarian secara random dan mengikuti *waggle dance* jika *profitability rating*-nya rendah ketika dibandingkan dengan rata-rata *profitability* koloninya.

Secara garis besar, diagram alir dari algoritma lebah yaitu sebagai berikut,



Gambar 2.10 Diagram Alir Algoritma lebah

2.7 Pemrograman Java

2.7.1 Definisi Pemrograman Java

Java pada dasarnya adalah bahasa pemrograman yang mendukung konsep OOP (Object Oriented Programming) secara total. Maksudnya adalah bahwa sejak awal mula Java diciptakan, Java memang dibuat untuk mendukung konsep OOP tersebut. Oleh karena itu, struktur bahasa pemrograman Java harus memiliki sebuah class utama dan sebuah method utama. Struktur dasar pemrograman java antara lain ialah sebagai berikut:

- **Komentar**

Komentar adalah baris program yang tidak ikut diproses atau dikerjakan sebagai suatu perintah oleh kompiler dan interpreter. Baris komentar hanya berfungsi sebagai tag atau tanda atau keterangan tentang baris atau blok (kumpulan) perintah dibawahnya.

- **Kata Kunci**

Kata kunci adalah kata-kata yang sudah memiliki arti tersendiri atau khusus bagi interpreter dan kompiler Java untuk diterjemahkan menjadi perintah kepada komputer untuk mengerjakan sesuatu. Misalnya bila komputer menemukan kata class, maka otomatis kompiler akan menerjemahkan bahwa itu adalah awal dari class dan nama class-nya ada disebelah kanannya. Contoh lain dari kata kunci adalah public, static, void.

- *Modifier*

Java menggunakan beberapa kata khusus tertentu yang disebut *modifiers* yang berfungsi untuk menetapkan properti dari data, method, atau class dan bagaimana cara menggunakannya. Contoh dari kata khusus yang berupa modifiers adalah public, static, private, final, abstract, protected. Sebuah variabel atau class atau method yang memiliki modifier public berarti dapat diakses oleh class lainnya. Sedangkan bila modifier private, maka tidak dapat diakses oleh class lainnya, hanya class yang mendefinisikannya.

- *Statement*

Statement pada dasarnya mempresentasikan sebuah aksi atau sebuah urutan aksi.

- Blok

Tanda *brace* (kurung kurawal) didalam program yang membentuk sekelompok perintah atau komponen lain dalam program yang digunakan untuk membentuk sebuah struktur pada program seperti class atau method.

- *Class*

Setiap program Java setidaknya harus memiliki sebuah class karena pada Java class adalah struktur program yang paling mendasar. Class harus memiliki setidaknya sebuah method. Class utama dalam program Java harus memiliki method utama.

- *Method*

Method pada dasarnya haruslah ada didalam class dan tidak dapat berdiri sendiri seperti sebuah fungsi diluar class.

- *Main Method*

Seperti yang telah disebutkan diatas bahwa tiap class harus memiliki method, dan setiap class utama dalam program juga harus memiliki method utama yang disebut juga main method. Main method ini fungsinya mengontrol seluruh alur dari program sewaktu menjalankan tugasnya.

2.7.2 J2SE (*Java 2 Standart Edition*) dan JDK (*Java Development Kit*)

Java Platform Standart Edition (Java SE, sebelumnya dikenal sebagai J2SE) menyediakan lingkungan kerja yang mendukung untuk pengembangan aplikasi dan penyebaran program pada desktop dan server. Java SE pada dasarnya kini menuntut untuk *embedded* dan *real time environments*.

JDK adalah development environment untuk aplikasi development, applet, dan komponen lainnya dengan menggunakan bahasa pemrograman Java dan termasuk tools yang sangat berguna untuk mengembangkan dan menguji program yang di tulis dengan bahasa pemrograman Java.

Perangkat lunak Java 2 Platform, Standard Edition (J2SE) adalah pilihan platform utama yang dapat dengan cepat mengembangkan suatu project dan dapat digunakan dengan sangat aman. Aplikasi Java 2 PlatformStandard Edition (J2SE)

Development Kit (JDK) ini berjalan secara portable pada server dan sistem desktop.

BAB III

PERANCANGAN SISTEM

Pada perancangan aplikasi yang akan dibuat ini, yaitu aplikasi untuk memecahkan masalah TSP (*Travelling Salesman Problem*) dengan menggunakan pemrograman java, yang mana diperlukan suatu algoritma untuk memecahkan permasalahan tersebut. Usaha pemecahan permasalahan tersebut dapat dilakukan melalui berbagai algoritma tertentu sebagai metodenya, yang dalam hal ini dipilih sebuah metode berupa algoritma lebah.

Aplikasi dirancang untuk memecahkan masalah penentuan urutan rute yang hendak dilalui dari tiap-tiap kota dengan memperhatikan tingkat efisiensi jarak yang menghubungkan tiap-tiap kota tersebut, sehingga dapat ditempuh dengan jarak yang lebih pendek.

3.1 Alat dan Bahan Penelitian

3.1.1 Alat Penelitian

Alat-alat yang digunakan sebagai sarana untuk mendukung penelitian ini antara lain yaitu:

1. Perangkat keras berupa personal komputer dengan spesifikasi sebagai berikut,
 - Intel Processor Dual Core 1.6 GHz
 - RAM 2 GB
 - Harddisk dengan ruang minimal 1GB

- Monitor 32bit *Color*
2. Perangkat lunak yakni meliputi,
 - Netbeans 6.9.1
 - JDK 1.6

3.1.2 Bahan Penelitian

Pada penelitian ini, bahan yang digunakan untuk penelitian antara lain sebagai berikut,

1. Daftar titik apotik berupa koordinat, data apotek mengambil sampel dari nama-nama apotek yang berada dikawasan sekitar Daerah Istimewa Yogyakarta.
2. Data jalan berupa jalan lokasi apotek dan juga jalan yang hendak dilalui dari satu titik apotek ke titik apotek lainnya.

3.2 Langkah Penelitian

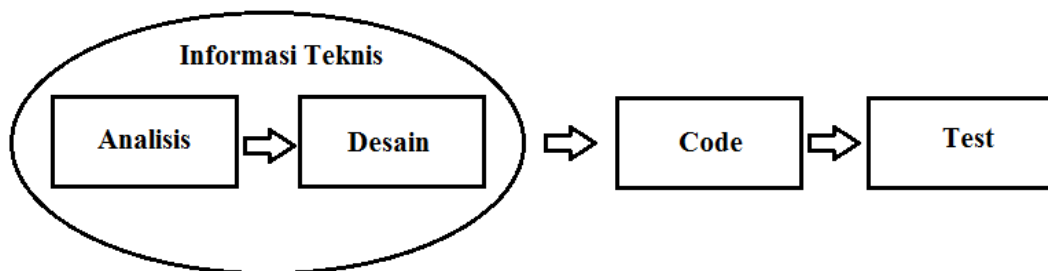
Terdapat beberapa langkah yang dilakukan guna melakukan penelitian terkait pemecahan masalah *travelling salesman problem* dengan metode algoritma lebah, langkah tersebut antara lain,

1. Studi literatur, merupakan proses mempelajari dan memahami konsep-konsep serta teori yang berhubungan dengan pembuatan atau pengembangan perangkat lunak, teori algoritma lebah, dan juga teori mengenai *travelling salesman problem*.

2. Modifikasi algoritma lebah yang dilakukan guna menyesuaikan metode dan penyelesaian masalah yang hendak dicapai, dalam hal ini adalah pemecahan masalah mengenai *travelling salesman problem*.
3. Pengumpulan data berupa apotek-apotek yang wilayah persebarannya meliputi kawasan yang ada di Daerah Istimewa Yogyakarta. Pengumpulan data dilakukan dengan cara observasi langsung ke lapangan.
4. Pembuatan data koordinat lokasi apotek yang diinput kedalam perangkat lunak yang hendak dibangun.
5. Graf terkait matriks ketetangaan, dan bobot merupakan bahan representasi utama untuk pengembangan perangkat lunak.
6. Dokumentasi baik berupa dokumen teknis maupun berupa tulisan yang dimuat dalam bentuk skripsi, yang mana didalamnya memuat tentang hasil penelitian yang telah dilakukan.

3.3 Model Penelitian

Dalam penelitian ini digunakan suatu model yang dikenal dengan nama waterfall pressman, dalam hal ini model tersebut dijadikan acuan dalam pembangunan dan pengembangan perangkat lunak.



Gambar 3.1 Model *Waterfall Pressman*

Pada model diatas yakni model *waterfall pressman*, informasi teknis pada hakikatnya harus didapatkan terlebih dahulu, informasi teknis yang dimaksud adalah berupa analisis dan desain terkait perangkat lunak yang hendak dibangun. Analisis merupakan tahapan paling awal dalam model ini, tahapan ini digunakan untuk mengetahui informasi dan spesifikasi sistem yang digunakan, baik itu kebutuhan fungsional maupun nonfungsional dalam pembangunan perangkat lunak terkait pencarian urutan rute yang paling optimal dengan metode algoritma lebah.

Dalam penelitian ini, data-data yang dibutuhkan antara lain berupa data koordinat, data ketetanggaan, data jalan dan data jarak yang menghubungkan antara titik satu dengan titik yang lainnya. Nantinya tahapan pertama yang dilakukan adalah menentukan titik (memasukkan data koordinat), dari data koordinat tersebut maka akan didapatkan jarak atau bobot antar titik, setelah itu akan dimunculkan data ketetanggaan, data ketetanggaan ini berisi tentang semua kemungkinan urutan yang mungkin terjadi. Bilamana semua data sudah tersedia dan terpenuhi, maka yang selanjutnya adalah memulai pencarian urutan yang paling optimal dengan menggunakan algoritma lebah.

Tahapan yang berikutnya adalah desain perangkat lunak, pada tahapan ini bertujuan untuk membuat desain dari hasil analisis yang telah dilakukan pada tahapan pertama. Informasi dan spesifikasi yang didapat dari hasil analisis yang diubah menjadi sebuah desain sistem nantinya akan akan dikodekan. Pada tahap pengkodean, kode-kode yang dibuat ialah berupa kode desain yang telah dibuat sebelumnya, yakni dikodekan kedalam bahasa pemrograman yang dapat dibaca

oleh komputer dan diubah menjadi suatu tampilan yang dapat digunakan guna menunjang penelitian. Pada tahapan pengkodean ini mula-mula adalah perancangan data, data yang digunakan antara lain adalah matriks koordinat, matriks ketetangaan, dan matriks bobot jarak. Tahapan pengkodean yang kedua adalah perancangan antar muka yang antara lain meliputi menu utama, panel jalur, dan lain sebagainya.

Tahapan yang paling akhir dari model *waterfall pressman* adalah tahapan pengujian, tahapan ini merupakan tahapan dimana *software* atau perangkat lunak yang telah dibuat diujikan apakah sudah sesuai dengan kebutuhan ataukah belum. Pengujian yang dilakukan ialah berupa input data hingga muncul hasil akhir sehingga diketahui apakah perangkat lunak yang dibangun sudah sesuai atau belum.

3.4 Analisis Kebutuhan Perangkat Lunak

Dari data yang diperoleh berdasarkan observasi lapangan yakni berupa observasi lokasi-lokasi apotik di Daerah Istimewa Yogyakarta. Maka kebutuhan proses, masukan dan keluaran yakni sebagai berikut,

- Kebutuhan Proses

Kebutuhan proses dalam penentuan urutan rute paling optimal yaitu meliputi,

1. Proses penentuan data apotek dan rute pada peta
2. Proses penentuan titik apotek selanjutnya

3. Proses perhitungan jarak dari tiap-tiap lebah
4. Proses penyeleksian jarak terpendek

- Kebutuhan masukan

Input atau masukan dari aplikasi pencarian urutan rute terbaik berupa parameter-parameter yang dibutuhkan antara lain,

1. Data graph berupa banyaknya apotek (n) yang diinput disertai dengan data koordinat (x,y), n merupakan banyaknya apotek yang masuk kedalam tabulist. Banyaknya n diinputkan oleh pengguna.
2. Parameter lain yang dibutuhkan adalah berupa jumlah iterasi, dan banyaknya lebah, namun banyaknya lebah sudah disesuaikan berdasarkan banyaknya titik apotek yang dimasukkan.

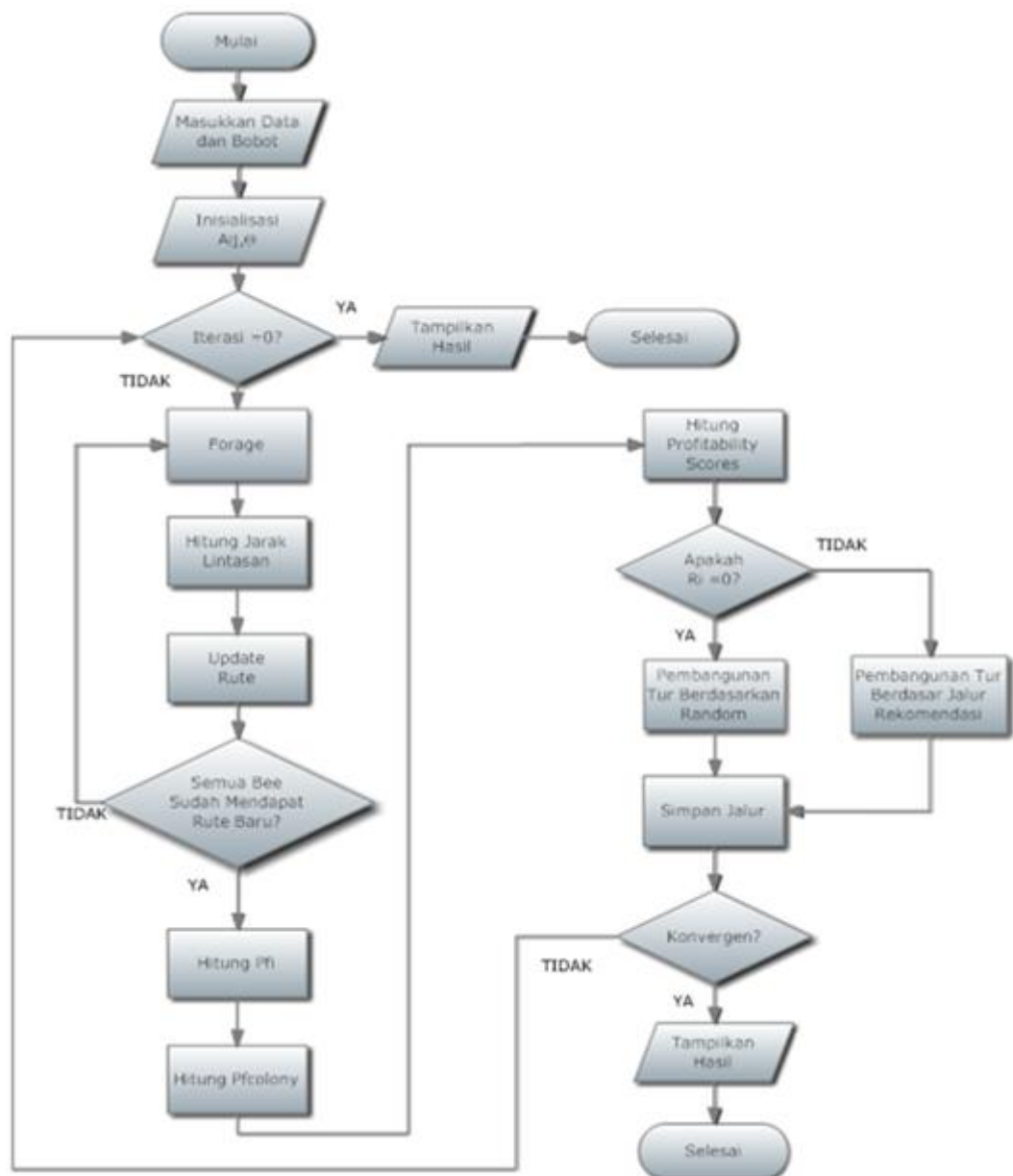
- Kebutuhan keluaran

Data keluaran dari proses aplikasi ini adalah didapatkannya urutan rute terbaik dari seluruh kemungkinan rute yang telah ada dan telah ditampilkan sebelumnya.

3.5 Diagram Alir Algoritma Lebah Pada *Travelling Salesman Problem*

Untuk menyelesaikan permasalahan penentuan urutan rute yang paling optimal dengan metode algoritma lebah, maka algoritma harus disesuaikan dengan

kebutuhan sistem yang hendak digunakan dalam penelitian yang mana dalam hal ini adalah terkait pemecahan masalah travelling salesman problem. Adapun diagram alirnya adalah sebagai berikut,



Gambar 3.2 Diagram Alir Algoritma Lebah Untuk *Travelling Salesman Problem*

3.6 Rancangan GUI (*Graphical User Interface*) J2SE Sebagai Media Interaksi

Graphical User Interface (GUI) yang diterapkan pada aplikasi ini bertujuan sebagai media interaksi dengan pengguna yang menggunakan aplikasi tersebut. GUI yang ada pada aplikasi tersebut didalamnya termuat *source code* atau sintaks program yang membangun GUI itu sendiri sehingga menghasilkan sebuah tampilan antar muka. GUI yang dirancang tersebut menampilkan sejumlah menu yang dianggap diperlukan. Perancangan GUI guna aplikasi ini dibuat dengan menggunakan bahasa pemrograman Java untuk versi *Java 2 Standart Edition* (J2SE). Pembuatan GUI aplikasi ini ialah dengan menggunakan Netbeans 6.9.1, yang mana software tersebut memang ditujukan dan digunakan untuk pemrograman Java. Rancangan jendela aplikasi yang hendak dibuat adalah sebagai berikut,

Travelling Salesman Problem	
File Edit Run View Help	
	Jumlah Iterasi
	<input type="text"/>
	Melintas Pada Pukul:
	<input type="text"/>
	Kemungkinan Rute Yang Dapat Dilewati:
	<input type="text"/>
	Proses Iterasi
	<input type="text"/>
	Rute Yang Paling Optimal:
	<input type="text"/>
	Total Jarak Tempuh
	<input type="text"/>

Gambar 3.3 Rancangan Antarmuka Aplikasi

Pada gambar diatas terdapat 5 menu utama aplikasi, yang antara lain terdiri dari menu *File*, *Edit*, *Run*, *View* dan *Help*. Menu tersebut memiliki fungsi masing-masing, dalam menu *file* terdapat sub-menu yang terdiri dari menu *exit* dan menu *print*. Menu *exit* digunakan untuk melakukan perintah untuk keluar dari aplikasi sedangkan pada menu *print* difungsikan untuk mencetak halaman yang ada di panel aplikasi.

Menu yang selanjutnya adalah menu *Edit*. Pada menu ini juga terdiri dari sub-menu lainnya antara lain yaitu terdiri dari menu *undo*, *redo*. Pada menu *undo* berfungsi untuk mengembalikan ke langkah sebelumnya. Tiap dilakukannya pemilihan tombol *undo* berarti akan kembali pada 1 langkah sebelumnya atau 1 langkah kebelakang. Kemudian pada tombol *redo* adalah kebalikan dari tombol *undo*, yang mana pada tombol ini akan berfungsi untuk mengembalikan ke langkah setelahnya. Sehingga akan kembali pada langkah yang telah dilalui, namun pada 1 langkah kedepannya.

Menu berikutnya adalah menu *View*. Dalam menu ini juga terdiri dari beberapa sub-menu antara lain adalah menu antara lain adalah menu *show*, *empty*, dan *data*. Pada menu *show* masih terdiri dari beberapa sub-menu lagi yakni *without line* dan *show all possible lines*. Menu *without line* apabila digunakan maka akan menghilangkan tampilan semua garis yang mungkin dilalui, sehingga yang akan muncul hanya titik-titik yang diibaratkan sebagai titik kota yang akan dilewati. Sedangkan menu *show all possible line* akan mengerjakan perintah untuk menampilkan semua kemungkinan rute yang dapat dilalui. Selanjutnya pada menu *empty* digunakan sebagai perintah untuk mengosongkan semua isi atau gambar

yang ada pada panel inti, sehingga panel tersebut tidak ditampilkan baik titik maupun garisnya. Dan pada menu *data* merupakan menu yang memuat tentang informasi koordinat titik ataupun keterangan bobot yang ada pada panel inti.

Menu yang terakhir adalah menu *Help*. Menu tersebut digunakan untuk memberikan pertolongan yakni berupa informasi-informasi penting yang berkaitan dengan aplikasi.

Pada menu utama aplikasi juga terdapat tombol *Run*. Menu ini berfungsi menjalankan algoritma lebah, sehingga didapatkan urutan rute yang paling optimal dari semua kemungkinan rute yang ada. Dengan dijalankannya menu ini maka hanya akan didapat urutan rute yang paling optimal yang disertai munculnya total panjang rute keseluruhan. Hasil optimasi dengan algoritma lebah akan ditampilkan pada kotak Rute Tercepat Yang Dapat Dilalui, hal yang ditampilkan pada kotak tersebut meliputi informasi nama jalan yang seharusnya dilewati, yang mana jalan tersebut adalah rute terpendek dan yang tercepat.

Sedangkan pada panel utama meliputi gambar peta yang terdiri dari lokasi apotek yang dapat ditandai dengan sebuah titik sebagai acuan lokasi yang nantinya akan dilewati dan dicari rute optimalnya. Pada bagian sebelah kiri terdapat *text area* yang terdiri dari jumlah iterasi yang diinput, kemungkinan rute yang dapat dilewati, proses iterasi, rute yang paling optimal, dan total jarak tempuh. Bagian *text area* yang berupa jumlah iterasi merupakan kolom yang disediakan oleh pengguna untuk memasukkan banyaknya jumlah iterasi yang dapat diproses dalam aplikasi tersebut. Sehingga angka yang dimasukkan dalam

kolom jumlah iterasi tersebut nantinya akan diproses iterasinya sebagaimana jumlah angka yang dimasukkan oleh pengguna. Kolom lainnya adalah kemungkinan rute yang dapat dilewati, kolom ini berisi tentang data-data seluruh alamat jalan yang dapat dilalui, pada kolom ini seluruh kemungkinan yang ada pada panel yakni berupa hubungan antar titik yang terhubung secara penuh. Sehingga semua kemungkinan jalan yang tiap titik-titiknya terhubung maka alamat jalannya akan ditampilkan secara keseluruhan.

Kolom *text area* yang berikutnya adalah proses iterasi, pada kolom ini akan menampilkan proses yang terjadi selama iterasi berlangsung, banyaknya data hasil iterasi adalah sejumlah data yang diinput oleh *user* pada kolom jumlah iterasi. Di kolom ini proses selama berjalannya iterasi akan direkam sehingga dapat dilihat hasil dari setiap langkah proses iterasinya yakni dari awal proses iterasi hingga akhir iterasi. Data hasil iterasi ini akan muncul bilamana tombol *run* telah dijalankan. Kolom yang berikutnya yaitu rute yang paling optimal, kolom *text area* ini berisikan data alamat jalan yang mana data yang ditampilkan adalah hanya berupa alamat jalan yang diambil dari urutan rute yang paling optimal berupa hasil dari proses optimasi. Kolom bagian terakhir adalah kolom *text area* yang berisikan tentang data total jarak tempuh yang harus dilalui, yang mana data total jarak merupakan jarak sebenarnya, jarak pada kolom ini menggunakan satuan meter. Data pada kolom bagian ini juga akan dimunculkan setelah program selesai di *run*.

BAB IV

PENGUJIAN, ANALISIS DAN PEMBAHASAN

4.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan guna mengetahui apakah perangkat lunak yang dibangun sudah dapat bekerja dan berfungsi dengan baik sebagaimana yang sasaran diinginkan. Pengujian dilakukan terhadap perangkat lunak yang meliputi sistem-sistem utama yang mendukung berfungsinya perangkat lunak yang dimaksud.

4.1.1 Pengujian Data Tabulist

Pengujian input perlu dilakukan guna mengetahui apakah input yang diberikan dapat diproses di langkah selanjutnya. Input pada penelitian ini adalah berupa titik-titik koordinat yang mewakili posisi dari tiap-tiap apotek yang ada pada peta. Titik koordinat tersebut digambarkan dalam bentuk koordinat x,y yang mana titik x,y tersebut merepresentasikan posisi dari masing-masing apotik.

Tabel 4.1 Tabel Pengujian Data Koordinat

ID	Nama	X	Y
1	Apotik Kentungan(1)	500	214
2	Apotik Maranatha(2)	170	845
3	Apotek Wisnu(3)	418	484
4	Apotek Artha Farma(4)	165	490
5	Apotek Annisa(5)	404	407
6	Apotek Orimedic(6)	815	475
7	Apotek K24(7)	677	285
8	Apotek Kimia Farma ...	396	605
9	Apotek Hayam Wur...	492	630
10	Apotek Afina(10)	566	588
11	Apotek Dian Farma(11)	73	700
12	Apotek Pendowo(12)	557	818
13	Apotek Shinta(13)	652	867
14	Apotek Ratna(14)	426	947

Dari tabel diatas dapat dilihat bahwa tiap-tiap apotek memiliki data koordinat masing-masing sesuai dengan letak atau lokasi yang ada pada peta, dari data koordinat diatas nantinya dapat dicari jarak dari satu titik apotek ke titik apotek lainnya. Berikut adalah data jarak pada tabulist,

Tabel 4.2 Data Jarak Pada Tabulist

Nomor dan Nama Apotik	x	y		x	y	Jarak Euclidean	Jarak Sebenarnya (m)
Titik 1 (Apotek Kentungan)	500	214	Titik2 (Apotek Maranatha)	170	845	712,0822	10200
			Titik3 (Apotek Wisnu)	418	484	282,1772	3910
			Titik4 (Apotek Artha Farma)	165	490	434,0518	6970
			Titik5 (Apotek Annisa)	404	407	215,5574	3740
			Titik6 (Apotek Orimedic)	815	475	409,0795	6630
			Titik7 (Apotek K24)	677	285	190,7092	3570
			Titik8 (Apotek Kimia Farma)	396	605	404,5949	5950
			Titik9 (Apotek Hayam Wuruk)	492	630	416,0769	5270
			Titik10 (Apotek Afina)	566	588	379,7789	5100
			Titik11 (Apotek Dian Farma)	73	700	646,9351	9180
			Titik12 (Apotek Pendowo)	557	818	606,6836	7480
			Titik13 (Apotek Shinta)	652	867	670,4573	7820
			Titik14 (Apotek Ratna)	426	947	736,7259	9010
Titik2 (Apotek Maranatha)	170	845	Titik 1 (Apotek Kentungan)	500	214	712,0822	10200
			Titik3 (Apotek Wisnu)	418	484	437,9783	4250
			Titik4 (Apotek Artha Farma)	165	490	355,0352	3230
			Titik5 (Apotek Annisa)	404	407	496,5884	5780
			Titik6 (Apotek Orimedic)	815	475	743,5893	8140
			Titik7 (Apotek K24)	677	285	755,4131	9010
			Titik8 (Apotek Kimia Farma)	396	605	329,6604	5780
			Titik9 (Apotek Hayam Wuruk)	492	630	387,1808	4080

Nomor dan Nama Apotik	x	y		x	y	Jarak Euclidean	Jarak Sebenarnya (m)
Titik2 (Apotek Maranatha)	170	845	Titik10 (Apotek Afina)	566	588	472,0858	4240
			Titik11 (Apotek Dian Farma)	73	700	174,4534	1190
			Titik12 (Apotek Pendowo)	557	818	387,9407	4120
			Titik13 (Apotek Shinta)	652	867	482,5018	4250
			Titik14 (Apotek Ratna)	426	947	275,5721	3400

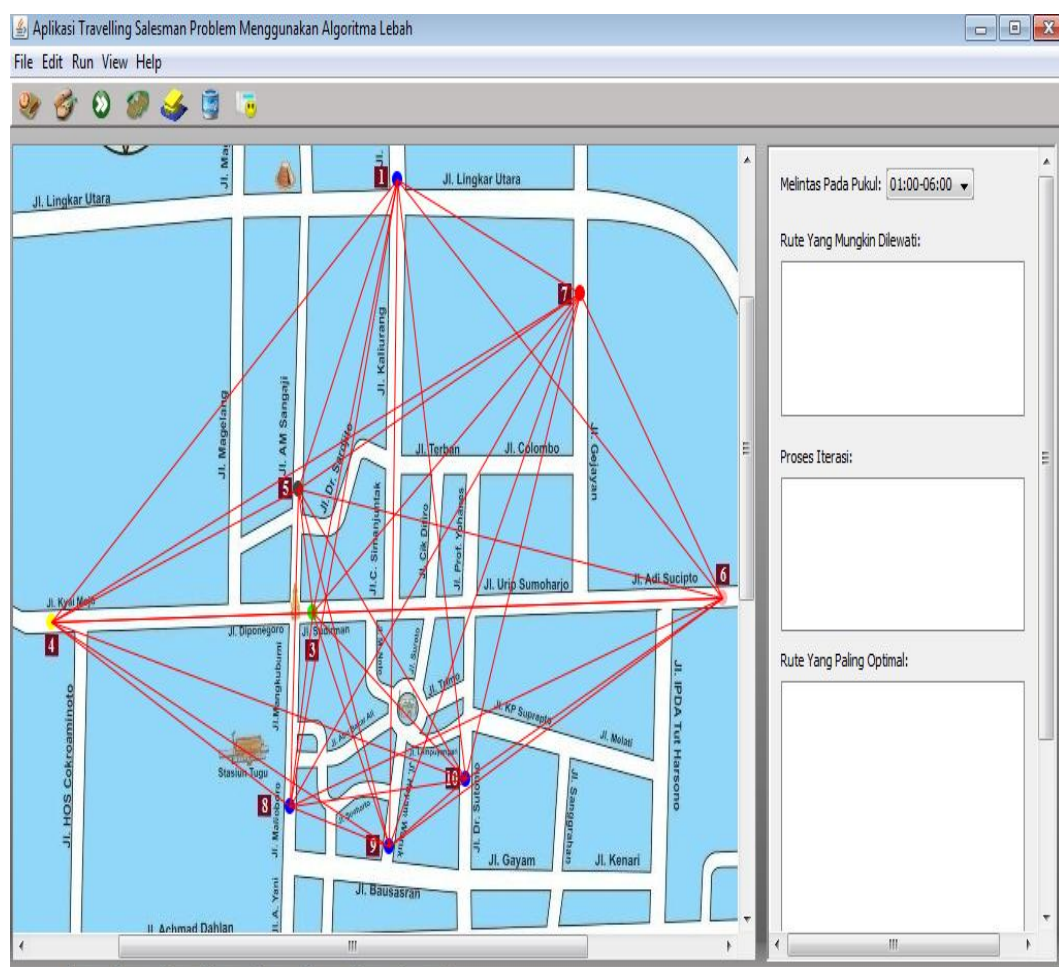
Untuk lebih lengkapnya mengenai data tabulist jarak antar titik yang terhubung, dapat dilihat pada lampiran data tabulist.

4.1.2 Pengujian Data Graf

Graf dalam hal ini merupakan suatu simpul yang menghubungkan antara satu titik apotek dengan apotek lain. Simpul dalam penelitian ini digambarkan melalui sebuah garis yang menghubungkan antar titiknya. Sehingga titik-titik apotek yang diinput maka akan muncul simpul yang menghubungkannya.

Simpul atau graf dalam hal ini tergolong dalam jenis graf berbobot tak berarah, yang mana simpul yang menghubungkan tiap titik pada dasarnya memiliki nilai tertentu yakni berupa jarak yang telah dihitung melalui rumus euclidean, namun tidak memiliki arah tertentu. Pada pengujian graf, mula-mula *user* atau pengguna memasukkan titik-titik apotek yang ada kemudian dari titik-titik tersebut dapat ditemukan jarak yang menghubungkan antar titiknya, jarak itulah yang disebut dengan bobot. Dalam penelitian ini, bobot pada dasarnya secara tidak langsung termuat dalam tabulist, sehingga nantinya tabulist

tersebutlah yang menjadi acuan perhitungan keseluruhan jarak sebelum akhirnya ditentukan jarak mana yang dianggap paling optimal berdasarkan perhitungan jarak dengan rumus euclidean, pada dasarnya hasil yang dianggap optimal tentunya berimplikasi terhadap urutan yang paling optimal pula. Berikut adalah tampilan pengujian simpul atau graf,



Gambar 4.1 Tampilan Pengujian Graf

Pada gambar diatas terlihat bahwa terdapat sembilan titik sample dari empat belas titik keseluruhan, dari kesembilan titik tersebut graf terhubung ke seluruh

titik secara penuh, yang artinya satu titik maka akan terhubung keseluruhan titik yang ada, dalam hal ini satu titik terhubung ke delapan titik lainnya.

4.2 Pengujian Algoritma Lebah Pada Permasalahan TSP

Pengujian algoritma lebah dalam hal ini digunakan untuk pemecahan masalah *travelling salesman problem* dengan studi kasus berupa persebaran apotek-apotek yang berada di kawasan Daerah Istimewa Yogyakarta. Perangkat lunak yang akan diujikan adalah berupa penyelesaian kasus dengan menggunakan algoritma lebah yang dibuat dengan menggunakan simulasi pemrograman JAVA.

Pengujian ini dilakukan guna memeriksa apakah algoritma lebah yang digunakan sudah sesuai atau belum, kemudian juga untuk mengetahui sejauh mana tingkat keoptimalan dari algoritma lebah tersebut terhadap pemecahan masalah berkenaan dengan *travelling salesman problem*.

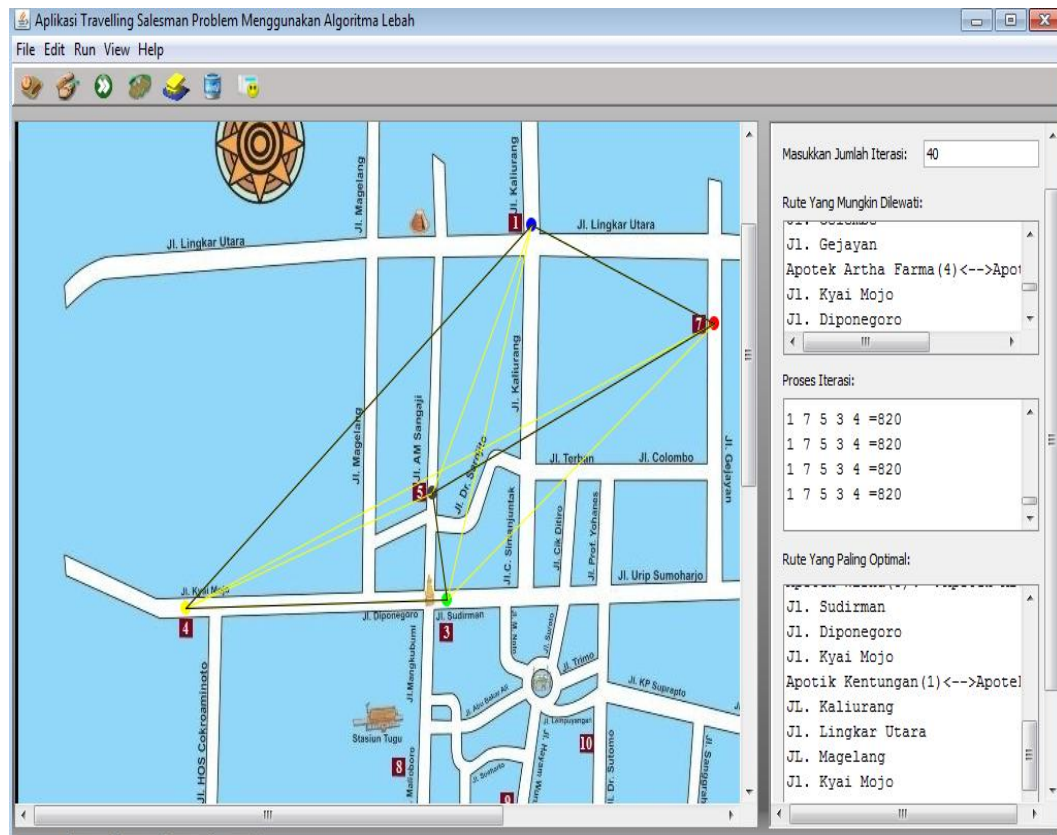
4.2.1 Pengujian Dengan Menggunakan 5 Titik Apotek

Pengujian yang pertama dilakukan dengan pengambilan sampel sebanyak 5 titik. Titik yang diambil dimisalkan diambil secara acak oleh *user*. Sehingga user memilih 5 titik secara acak kemudian nantinya dicari urutan rute terbaiknya dengan menggunakan algoritma lebah sebagaimana yang sudah tertanam dalam program.

Tabel 4.3 Lima Titik Yang Dipilh Acak Oleh *User*

ID	Nama	X	Y
1	Apotik Kentungan(1)	500	214
3	Apotek Wisnu(3)	418	484
4	Apotek Artha Farma(4)	165	490
5	Apotek Annisa(5)	404	407
7	Apotek K24(7)	677	285

Dari data diatas dapat dilihat bahwa data yang dipilih oleh *user* adalah ID titik bernomor 1,3,4,5,dan 7. Yang berturut-turut adalah Apotek Kentungan, Apotek Wisnu, Apotek Artha Farma, Apotek Annisa, dan Apotek K24. Sebagai mana yang terlihat pada tabel, Apotek kentungan berada di koordinat $x= 500$ dan koordinat $y= 214$, kemudian Pada apotek Wisnu berada pada koordinat $x= 418$ dan koordinat $y= 484$, untuk Apotek Artha Farma koordinatnya beradapada $x= 165$ dan $y= 490$, sedangkan Apotek Annisa koordinatnya adalah $x= 404$ dan $y= 407$, dan yang terakhir adalah Apotek K24 yang mana memiliki titik koordinat $x= 677$ dan $y= 285$. Setelah data-data diatas dipilih oleh user, maka selanjutnya program dijalankan sehingga didapatkan hasil sebagai berikut,



Gambar 4.2 Hasil Pengujian Dengan Data 5 Titik Apotek

Pada hasil pengujian diatas, didapatkan urutan yang dianggap paling optimal yang mana urutan tersebut berturut-turut antara lain adalah 1,4,3,5,7 kemudian kembali lagi ke 1, atau sama dengan 1,7,5,3,4 dan kembali lagi ke 1. Atau dengan kata lain misal mula-mula user berangkat dari titik 1 yakni Apotik Kentungan, maka yang selanjutnya dia harus menuju ke titik 4 yaitu Apotek Artha Farma, dan dilanjutkan menuju ke titik 3 yaitu Apotek Wisnu, selanjutnya menuju ke Apotek Annisa yang dan terakhir adalah menuju ke Apotek K24, kemudian setelah itu barulah kembali ke titik awal yaitu Apotek Kentungan.

Adapun jalan-jalan yang harus dilewati untuk menempuh apotek-apotek tersebut antara lain yaitu dari apotek kentungan menuju apotek Artha Farma

adalah Jalan Kaliurang – Jalan Lingkar Utara – Jalan Magelang – Jalan Kyai Mojo. Kemudian selanjutnya diteruskan dari Apotek Artha Farma menuju Apotek Wisnu dengan melewati jalan yaitu Jalan Kyai Mojo – Jalan Diponegoro – Jalan Sudirman. Dari Apotek Wisnu Menuju Apotek Annisa yakni melewati Jalan Sudirman – Jalan AM. Sangaji. Setelah itu menuju ke Apotek K24, dengan jalan yang dilewati adalah Jalan AM. Sangaji – Jalan Dr. Sardjito – Jalan Terban – Jalan Kolombo – Jalan Gejayan. Dan terakhir kembali lagi ketitik semula yaitu Apotek Kentungan, jalan yang dilewati yakni Jalan Gejayan – Jalan Lingkar Utara – Jalan Kaliurang. Dari seluruh jalan yang telah dilewati tersebut, yang mana jalan tersebut merupakan jalan yang digunakan untuk menempuh seluruh apotek tujuan dengan total jarak tempuhnya yaitu sejauh 17.680 meter atau samadengan 17,68 kilometer.

4.2.2 Pengujian Dengan Menggunakan 10 Titik Apotek

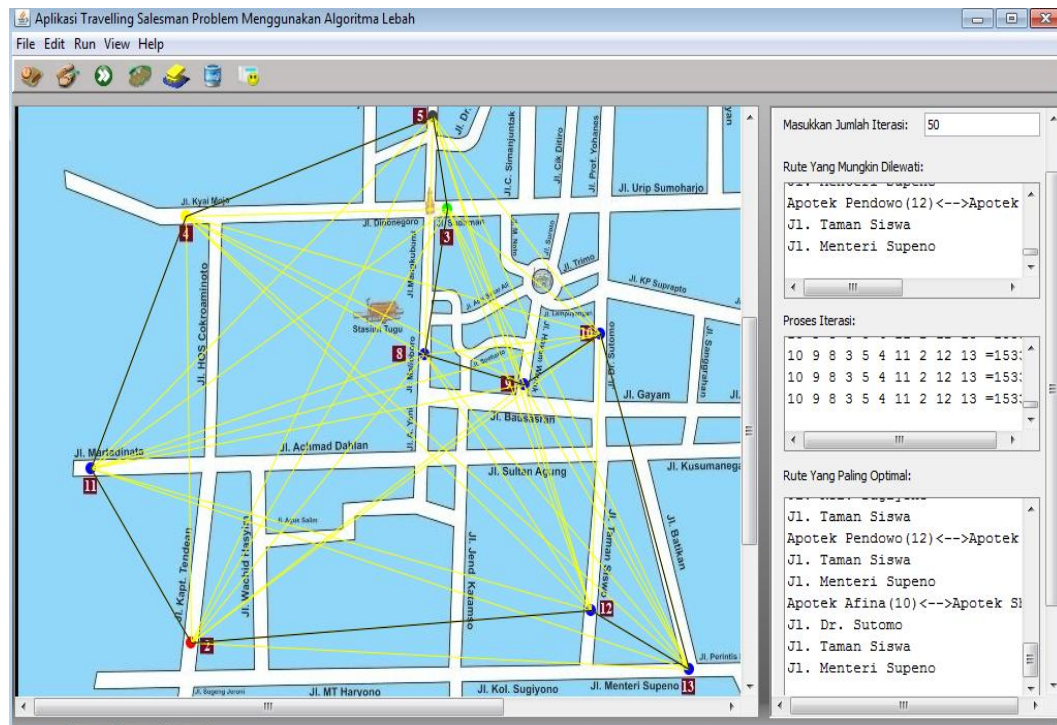
Pengujian yang selanjutnya adalah pengujian perangkat lunak dengan menggunakan sample sebanyak 10 titik apotek dari 14 titik yang ada. Sebagaimana seperti halnya pada percobaan dengan menggunakan 5 titik, titik-titik tersebut dapat diinput oleh user secara acak yang kemudian nantinya dengan menggunakan metode algoritma lebah urutan yang paling optimalnya dapat ditemukan.

Tabel 4.4 Sepuluh Titik Yang Dipilih Acak Oleh User

ID	Nama	X	Y
5	Apotek Annisa(5)	404	407
4	Apotek Artha Farma(4)	165	490
3	Apotek Wisnu(3)	418	484
8	Apotek Kimia Farma Malioboro(8)	396	605
9	Apotek Hayam Wuruk(9)	492	630
10	Apotek Afina(10)	566	588
12	Apotek Pendowo(12)	557	818
2	Apotik Maranatha(2)	170	845
11	Apotek Dian Farma(11)	73	700
13	Apotek Shinta(13)	652	867

Pada tabel diatas dapat dilihat bahwa apotek yang digunakan sebagai sample dalam percobaan ini adalah sebanyak 10 titik, yang mana titik-titik yang diambil adalah titik 5 yaitu Apotek Annisa, titik 4 yakni Apotek Artha Farma, titik 3 yaitu Apotek Wisnu, kemudian titik dengan ID 8 yaitu Apotek Kimia Farma, Apotek Hayam Wuruk dengan ID 9, Apotek Afina yang memiliki ID nomor 10, Apotek Pendowo dengan nomor ID titik 12, titik nomor 2 yakni Apotek Maranatha, titik nomo 11 Apotek Dian Farma dan yang terakhir adalaj ID titik 13 yaitu Apotek Shinta.

Sebagaimana yang terlihat pada tabel, tiap-tiap titik apotek yaitu kesepuluh apotek yang dijadikan sample, seluruhnya memiliki koordinatnya masing-masing yang tentunya memiliki koordinat yang berbeda-beda antara titik satu dengan titik lainnya. Karena digunakannya 10 titik dari 14 titik yang disediakan, maka kemungkinan solusi yang dapat dimunculkan seluruhnya adalah berjumlah 181.440 solusi.



Gambar 4.3 Hasil Pengujian Dengan Data 10 Titik Apotek

Dari hasil pengujian diatas diasumsikan mula-mula titik awalnya adalah titik 5 yaitu Apotek Annisa, yang mana terletak di Jl. AM. Sangaji. Berdasarkan dari hasil percobaan tersebut urutan yang dihasilkan adalah 5 – 3 – 8 – 9 – 10 – 13 – 12 – 2 – 11 – 4 – 5. Untuk menemukan urutan yang dianggap optimal tersebut dibutuhkan iterasi sebanyak 11 kali iterasi sehingga akhirnya ditemukan hasil berupa konvergenya seluruh urutan yang dihasilkan. Adapun alamat jalan yang harus dilalui untuk menempuh dari apotek ke apotek yaitu mula-mula dari Apotek Annisa(5) menuju Apotek Wisnu(3) jalan yang harus dilewati adalah JL. AM Sangaji – Jl. Sudirman. Kemudian dari Apotek Wisnu(3) ke Apotek Kimia Farma Malioboro(8) jalan dilewati adalah Jl. Sudirman – Jl. Mangkubumi – Jl.

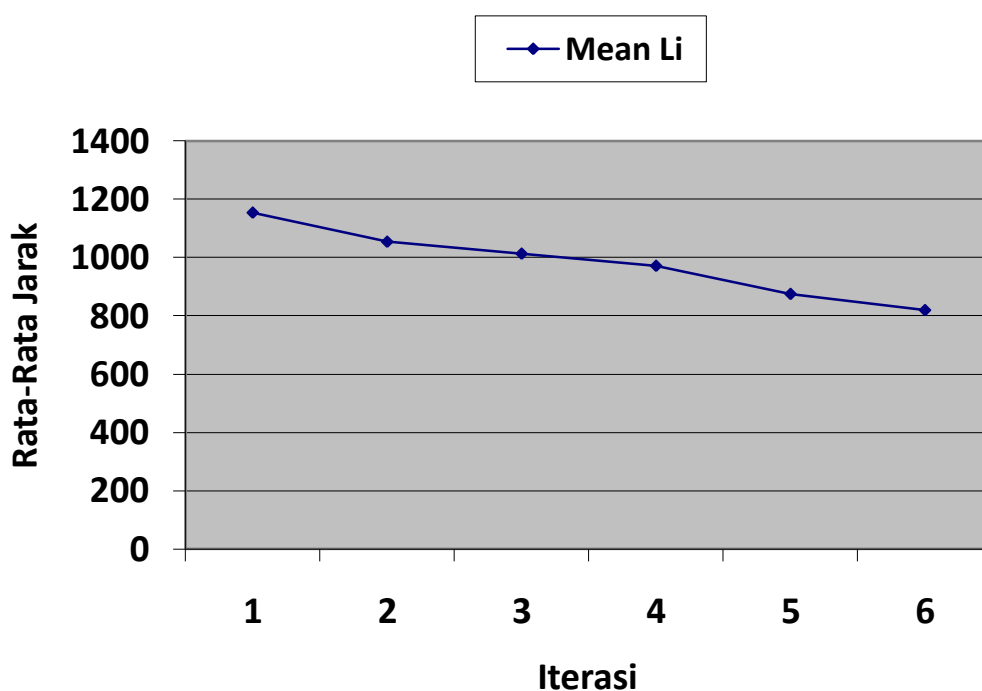
Malioboro, selanjutnya adalah menuju ke Apotek Hayam Wuruk(9), untuk menempuh apotek tersebut maka jalur yang perlu dilewati adalah yakni Jl. Malioboro – Jl. Bausasran – Jl. Hayam Wuruk. Kemudian berdasarkan urutan yang harus dilalui maka selanjutnya adalah menuju ke titik yang bernomor 10 yakni Apotek Afina, rute yang harus dilewati adalah Jl. Bausasran – Jl. Dr. Sutomo. Urutan selanjutnya adalah dari Apotek Afina(10) menuju ke Apotek Shinta(13) dengan rute yang dilewati adalah Jl. Dr. Sutomo – Jl. Kusmanegara – Jl. Batikan – Jl. Menteri Supeno. Dari Apotek Shinta(13) menuju ke Apotek Pendowo(12) dengan jalan yang dilewati yaitu Jl. Menteri Supeno – Jl. Taman Siswa. Sedangkan untuk menuju titik apotek dari Apotek Pendowo(12) ke Apotek Maranatha(2) adalah melewati alamat jalan yakni Jl. Taman Siswa – Jl. Kolonel Sugiyono – Jl. MT Haryono – Jl. Sugeng Jeroni – Jl. Kapt. Tendean. Ke tujuan yang berikutnya adalah menuju ke Apotek Dian Farma(11) yang mana jalan yang harus dilalui adalah melewati Jl. Kapt. Tendean – Jl. Martadinata. Dan apotek tujuan yang terakhir yakni Apotek Artha Farma (4), untuk menuju ke Apotek Artha Farma maka jalan yang harus dilalui adalah Jl. Martadinata - Jl. HOS Cokroaminoto – Jl. Kyai Mojo. Setelah semua apotek berhasil dikunjungi maka untuk kembali ke titik semula yaitu menuju ke Apotek Annisa jalan yang bisa dilewati antara lain yaitu melalui Jl. Kyai Mojo - Jl. Diponegoro - Jl. AM. Sangaji. Untuk mengunjungi seluruh titik apotek yang ada, dalam percobaan ini total jarak tempuh yang harus dilalui adalah sejauh 22.130 meter atau setara dengan 22,13 kilometer,

4.3 Pengujian Proses Iterasi

Pada tahap pengujian ini akan diuji yaitu sejauh mana algoritma lebah yang dibuat hasilnya bisa konvergen terhadap satu lebah dengan lebah lain, yang artinya saat diberikan sejumlah iterasi maka lebah pada akhirnya berkerumun dan mengikuti lintasan yang sama, yang mana hal itu menandakan jika jalur itulah yang dianggap paling optimal untuk menuju makanannya. Dalam algoritma lebah ini, mula-mula lebah disebar secara acak untuk mencari rute mana yang paling optimal untuk menuju makanannya. Tiap-tiap lebah tersebut akan mengunjungi secara acak satu per satu makanannya hingga keseluruhan makanannya dikunjungi kemudian lebah tersebut kembali kesarang untuk menginformasikan rute yang dilewati kepada koloninya. Jika terdapat informasi terbaru yang lebih optimal, maka lebah akan menggunakan informasi tersebut sebagai informasi barunya, sedangkan apabila informasi yang baru dianggap lebih buruk dari yang sebelumnya maka informasi tersebut tidak digunakan melainkan hanya digunakan informasi yang sebelumnya yang lebih dahulu didapatkan pada saat *foraging*, atau dengan kata lain digunakan jalur rekomendasi yang sebelumnya. Jika semua lebah telah menginformasikan rute yang sama terhadap lebah lainnya, maka lebah-lebah tersebut akan terbang dengan rute yang sama (berkoloni) dalam menuju makanan-makanannya. Dalam pengujian ini akan diambil beberapa titik sebagai sample untuk diujikan sejauh mana algoritma lebah tersebut dapat konvergen hasilnya antara lebah satu dengan lebah lainnya dalam menentukan urutan titik yang harus dikunjungi.

4.3.1 Pengujian Proses Iterasi Terhadap 5 Titik

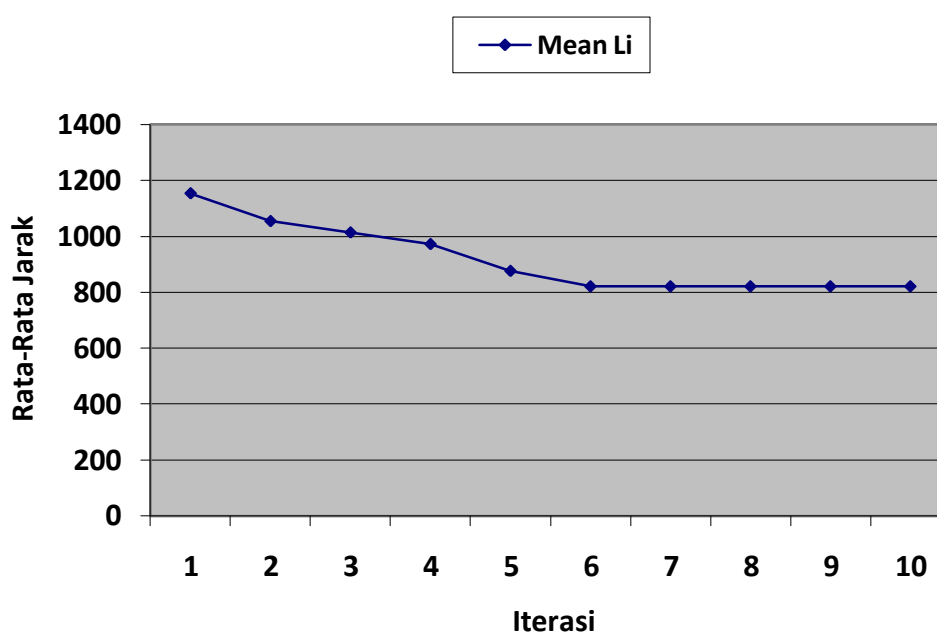
Berdasarkan 5 titik yang diujikan, yang mana kelima titik tersebut adalah ID titik bernomor 1,3,4,5,dan 7. Yang berturut-turut adalah Apotek Kentungan, Apotek Wisnu, Apotek Artha Farma, Apotek Annisa, dan Apotek K24 pada saat proses iterasinya dapat digambarkan sebagaimana grafik berikut,



Gambar 4.4 Grafik Proses Iterasi Dengan Sample 5 Titik Yang Berhenti Ketika Konvergen

Sebagaimana terlihat pada grafik diatas, mula-mula nilai rata-rata Li yaitu rata-rata jarak keseluruhan menunjukkan angka rata-rata yang cukup besar pada iterasi yang pertama yakni berada pada nilai 1153,6 yang kemudian nilai rata-ratanya menurun pada iterasi yang kedua yaitu pada nilai rata-rata 1054, demikian pula pada iterasi ketiga nilai rata-ratanya menurun lagi menjadi 1013,2, pada

iterasi keempat menjadi 971,6 dan pada iterasi yang kelima rata-ratanya menjadi 875,4. Rata-rata terkecil didapatkan ketika sampai pada iterasi yang keenam, yang mana nilai rata-ratanya adalah bernilai 820. Pada iterasi yang keenam ini pulalah didapatkan hasil yang konvergen dari seluruh lebah yang awalnya disebar secara acak. Apabila siklus diteruskan hingga mencapai nilai iterasi maksimum dimisalkan 10 iterasi, maka grafik dapat digambarkan sebagai berikut,

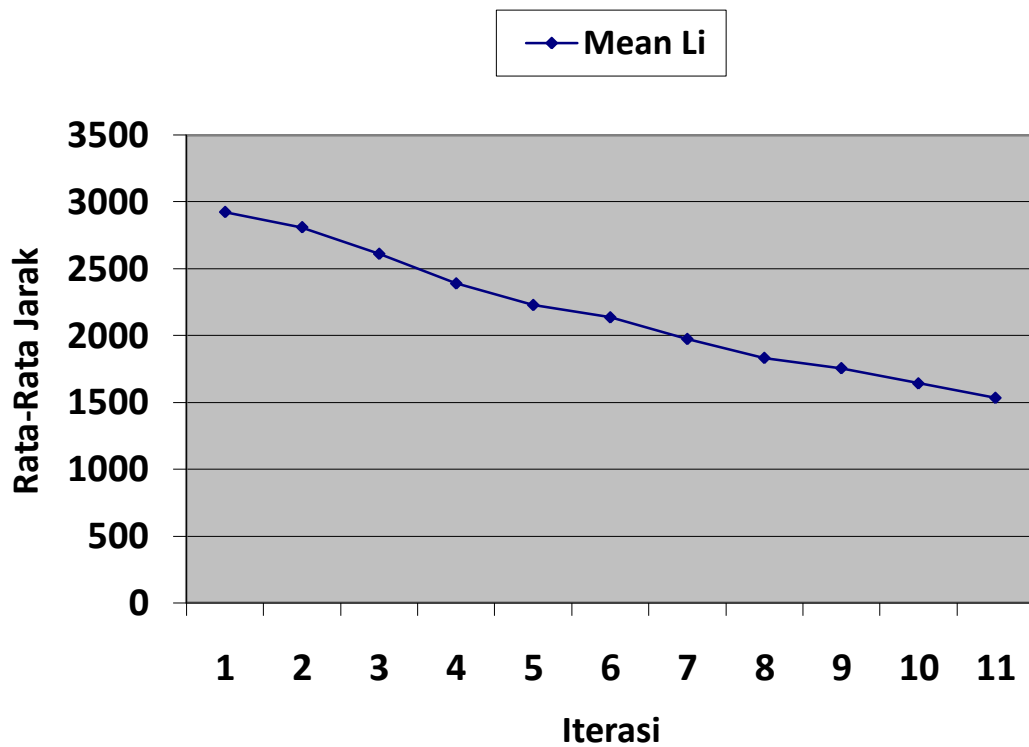


Gambar 4.5 Grafik Proses Iterasi Dengan Sample 5 Titik Yang Berhenti Ketika Iterasi Maksimum Tercapai

4.3.2 Pengujian Proses Iterasi Terhadap 10 Titik

Pengujian yang selanjutnya adalah dengan pengambilan sample sebanyak 10 titik apotek, sama seperti pengujian sebelumnya, titik-titik apotek tersebut diinputkan oleh *user*. Sample yang diambil antara lain adalah titik 5, 4, 3, 8, 9, 10, 12, 2, 11 dan 13. Yang mana apotek tersebut antara lain adalah Apotek Annisa, Apotek Artha Farma, Apotek Wisnu, Apotek Kimia Farma Malioboro, Apotek

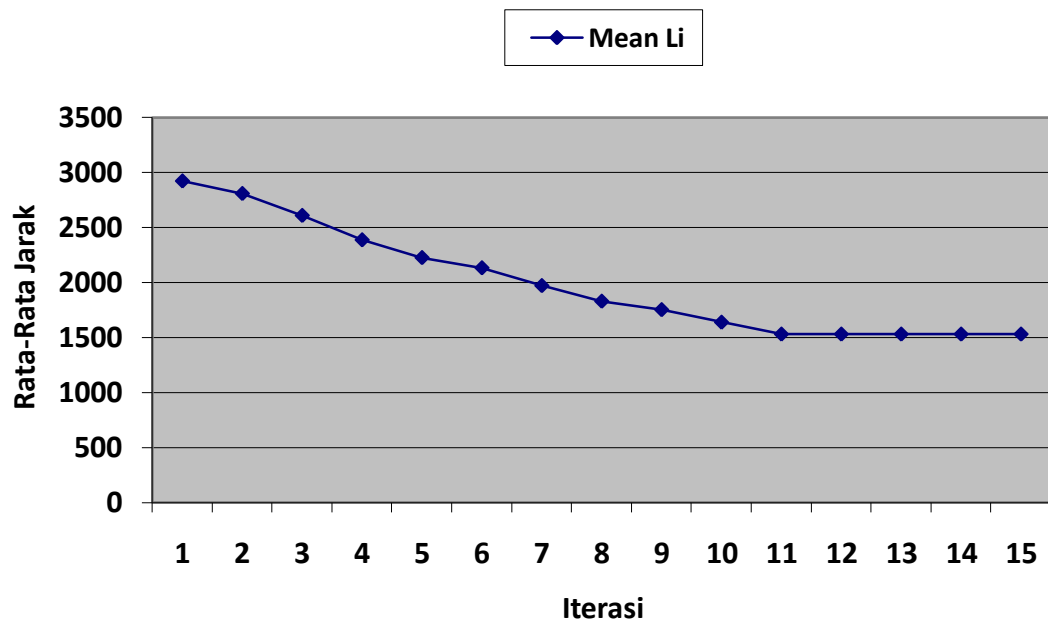
Hayam Wuruk, Apotek Afina, Apotek Pendowo, Apotek Maranatha, Apotek Dian Farma dan terakhir adalah Apotek Shinta. Dari kesepuluh titik apotek tersebut kemudian dicari urutan terbaiknya dengan menggunakan algoritma lebah yakni dengan grafik proses iterasi sebagai berikut,



Gambar 4.6 Grafik Proses Iterasi Dengan Sample 10 Titik Yang Berhenti Ketika Konvergen

Pada gambar diatas terlihat bahwa pada iterasi yang pertama, rata-rata jarak keseluruhan dari urutan-urutan rute yang dihasilkan dari tiap-tiap lebah adalah bernilai 2925, kemudian pada iterasi yang kedua menurun menjadi 2811, penurunan nilai rata-rata ini disebabkan karena adanya urutan rute baru yang ditemukan oleh lebah, yang mana rute tersebut memiliki nilai jarak yang lebih

kecil dibandingkan sebelumnya. Pada iterasi yang ketiga juga terjadi penurunan nilai rata-ratanya, yakni menjadi 2611,7 karena pada iterasi yang ketiga ini terdapat 2 lebah yang mendapatkan urutan jarak yang lebih pendek. Iterasi yang berikutnya adalah iterasi keempat, sama halnya dengan iterasi yang sebelumnya, pada iterasi yang keempat ini nilai rata-rata total jaraknya juga menurun yakni menjadi 2389,9. Untuk iterasi yang kelima rata-ratanya terus menurun menjadi 2228,4 demikian pula pada iterasi yang keenam yang mana nilai rata-rata jaraknya adalah 2135,7 dan terus menurun pada iterasi ketujuh sehingga pada iterasi yang ketujuh nilai rata-rata jaraknya menjadi 1973,9. Demikian pula halnya pada iterasi yang kedelapan, di iterasi yang kedelapan ini nilai rata-rata jaraknya terus menyusut sehingga bernilai 1832 kemudian diikuti oleh iterasi yang kesembilan bernilai 1755,2 dan iterasi kesepuluh yaitu bernilai 1641. Sedangkan pada iterasi yang kesebelas pada akhirnya iterasinya terhenti dikarenakan seluruh lebah telah menunjukkan hasil yang konvergen. Nilai rata-rata pada iterasi kesebelas adalah 1533. Keseluruhan lebah telah mendapatkan urutan rute yang sama sehingga nilai jarak yang didapatkan oleh tiap-tiap lebah nilainya sama dengan lebah lainnya yang sebelumnya juga melakukan observasi ke tiap-tiap titik yang menjadi sasaran makanannya. Adapun jika siklus iterasi dilanjutkan hingga maksimum iterasi tercapai, yakni dalam hal ini adalah 15kali iterasi, maka grafik proses iterasinya yaitu sebagai berikut,



Gambar 4.7 Grafik Proses Iterasi Dengan Sample 10 Titik Yang Berhenti Ketika Iterasi Maksimum Tercapai

4.4 Pengujian Dengan Titik *Start* Yang Berbeda

Pada algoritma lebah, posisi *start* atau posisi awal dapat menentukan urutan terbaik dari suatu urutan dalam suatu graf. Hal dikarenakan pada algoritma lebah terdapat kecenderungan lebah akan mengunjungi titik makanan atau nektar yang posisinya lebih dekat dari posisinya saat ini. Pada pengujian dengan menggunakan menggunakan keseluruhan titiknya yaitu 14 titik, posisi awal yang dimulai dari titik nomor 5 (Apotek Annisa) akan menghasilkan urutan graf yang lebih baik. Jika posisi awal dimulai dari titik nomor 5, maka urutan yang dihasilkan dari 14 titik yang digunakan akan menjadi 5 – 3 – 8 – 9 – 10 – 12 – 13 – 14 – 2 – 11 – 4 – 1 – 7 – 6 – 5. Dengan menggunakan urutan diatas maka total jarak tempuh yang

dihasilkan adalah sebesar 37770 meter. Pengujian lainnya adalah dengan menggunakan 14 titik namun dengan posisi start yang berbeda yaitu pada titik nomor 1 (Apotek Kentungan). Dengan menggunakan titik nomor 1 sebagai acuan titik awalnya, maka untuk mengunjungi keseluruhan titiknya dihasilkan urutan sebagai berikut 1 – 7 – 6 – 10 – 9 – 8 – 3 – 5 – 4 – 11 – 2 – 14 – 12 – 13 – 1. Dengan total jarak yang dihasilkan adalah 39470 meter, yakni memiliki selisih sebesar 1700 meter jika dibandingkan dengan posisi start pada titik nomor 5. Pengujian berikutnya dilakukan terhadap titik start yang dimulai dari titik nomor 12 (Apotek Pendowo), dalam pengujian ini dihasilkan urutan yakni 12 – 13 – 14 – 2 – 11 – 4 – 5 – 3 – 8 – 9 – 10 – 6 – 7 – 1 – 12. Maka total jarak tempuh yang dihasilkan dari urutan rute tersebut adalah 38960 meter, atau selisih 1190 meter jika dibandingkan dengan posisi start di titik nomor 5.

4.5 Analisis Proses Perhitungan Pada Algoritma Lebah Dengan Sample 4 Titik

Pada algoritma lebah untuk penelitian ini yaitu terkait TSP (*Travelling Salesman Problem*), proses perhitungan mula-mula diawali dengan menghitung forage dengan menggunakan data jarak yang ada pada tabulist. Kemudian setelah itu dilakukan perhitungan *waggle dance* dengan mencari nilai P_{fi} dan $P_{fcolony}$ yang nantinya digunakan untuk dibandingkan pada tabel *profitability scores*. Proses perhitungan ini diambil 4 sample titik antara lain adalah ID titik 1,3,4,5.

Yaitu Apotek Kentungan, Apotek Wisnu, Apotek Artha Farma, dan Apotek Annisa. Proses Perhitungan secara lengkap adalah sebagai berikut,

Tabel 4.5 Jarak Antar Titik 4 Sampel

ID Titik	1	3	4	5
1	0,0000	282,1772	434,0518	215,5574
3	282,1772	0,0000	253,0711	78,2624
4	434,0581	253,0711	0,0000	253,0020
5	215,5574	78,2624	253,0020	0,0000

Parameter yang ditentukan:

Alpha (α)= 1.00

Betha (β)= 1.00

Rho (ρ)= 1.00

Lambda (λ)= 0.7

Banyaknya lebah= 4

Dari jarak kota yang telah diketahui kemudian dapat dihitung visibilitas antar kota (η_{ij}) = $1/d_{ij}$, sehingga menjadi,

Tabel 4.6 Jarak Antar Titik 4 Sampel (1/Jarak)

ID Titik	1	3	4	5
1	0,0000	0,0035	0,0023	0,0046
3	0,0035	0,0000	0,0039	0,0128
4	0,0023	0,0039	0,0000	0,0040
5	0,0046	0,0128	0,0040	0,0000

$$\rho_{ij} = \{1, 3, 4, 5\}$$

$$\Theta = \{1, 3, 5, 4\} \quad \{4, 1, 5, 3\}$$

$$\{3, 5, 4, 1\} \quad \{5, 3, 1, 4\}$$

❖ Lebih ke-1

$$\rho_{ij} = \{1, 3, 4, 5\}$$

$$\Theta = \{1, 3, 5, 4\}$$

Nilai dari $P_{ij,n}$ lebih ke-1 dapat dicari dengan menggunakan rumus yaitu sebagai berikut,

$$\bullet \quad \text{Transisi} = 0 \quad P_{ij,n} = \frac{[\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}}{\sum_{j \in A_{i,n}} \left([\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}\right)}$$

$$\rho_{ij} = \{3, 4, 5\} \quad F_{1,0} = 1$$

$$= \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$$

$$= \{0.33, 0.33, 0.33, 0.33\}$$

$$P_{1,3} = \frac{(0.33)(0.0035)}{(0.33)(0.0035) + (0.33)(0.0023) + (0.33)(0.0046)}$$

$$= 0,3365$$

$$P_{1,4} = \frac{(0.33)(0.0023)}{(0.33)(0.0035) + (0.33)(0.0023) + (0.33)(0.0046)}$$

$$= 0,2212$$

$$P_{1,5} = \frac{(0.33)(0.0046)}{(0.33)(0.0035) + (0.33)(0.0023) + (0.33)(0.0046)}$$

$$= 0,4423$$

Maka yang terpilih adalah $P_{1,5}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 1 – 5.

- Transisi = 1

$$\rho_{ij} = \{3, 4\} \quad F_{4,1} = 4$$

$$= \{1-\lambda, \lambda\}$$

$$= \{0.3, 0.7\}$$

$$P_{5,3} = \frac{(0.3)(0.0128)}{(0.3)(0.0046) + (0.3)(0.0128) + (0.3)(0.0040)}$$

$$= 0,5981$$

$$P_{5,4} = \frac{(0.7)(0.0040)}{(0.7)(0.0046) + (0.7)(0.0128) + (0.7)(0.0040)}$$

$$= 0,1869$$

Maka yang terpilih adalah $P_{5,3}$. Sehingga urutan yang dihasilkan pada transisi 1 ini adalah 1 – 5 – 3.

- Transisi=2

$$\rho_{ij} = \{4\} \quad F_{2,2} = 3$$

$$= \left\{ \frac{1}{1} \right\}$$

$$= 1$$

$$P_{3,4} = \frac{(1)(0.0039)}{(1)(0.0035) + (1)(0.0039) + (1)(0.0128)}$$

$$= 0,1931$$

Maka yang terpilih adalah $P_{3,4}$. Sehingga urutan yang dihasilkan pada transisi 2 ini adalah 1 – 5 – 3 – 4.

❖ Lebah ke-2

$$\rho_{ij} = \{1, 3, 4, 5\}$$

$$\Theta = \{3, 5, 4, 1\}$$

Nilai dari $P_{ij,n}$ lebah ke-2 dapat dicari dengan menggunakan rumus yaitu sebagai berikut,

$$P_{ij,n} = \frac{[\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}}{\sum_{j \in A_{i,n}} \left([\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}\right)}$$

• Transisi =0

$$\rho_{ij} = \{1, 4, 5\} \quad F_{2,0} = 5$$

$$= \left\{ \frac{1-\lambda}{2}, \frac{1-\lambda}{2}, \lambda \right\}$$

$$= \{0.15, 0.15, 0.7\}$$

$$P_{3,1} = \frac{(0.15)(0.0035)}{(0.15)(0.0035) + (0.15)(0.0039) + (0.15)(0.0128)}$$

$$= 0,1733$$

$$P_{3,4} = \frac{(0.15)(0.0039)}{(0.15)(0.0035) + (0.15)(0.0039) + (0.15)(0.0128)}$$

$$= 0,1931$$

$$P_{3,5} = \frac{(0.7)(0.0128)}{(0.7)(0.0035) + (0.7)(0.0039) + (0.7)(0.0128)}$$

$$= 0,6337$$

Maka yang terpilih adalah $P_{3,5}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 3 – 5.

- Transisi= 1

$$\rho_{ij} = \{1, 4\} \quad F_{4,1} = 1$$

$$= \{\lambda, 1-\lambda\}$$

$$= \{0.7, 0.3\}$$

$$P_{5,1} = \frac{(0.7)(0.0046)}{(0.7)(0.0046) + (0.7)(0.0128) + (0.7)(0.0040)}$$

$$= 0,2150$$

$$P_{5,4} = \frac{(0.3)(0.0040)}{(0.3)(0.0046) + (0.3)(0.0128) + (0.3)(0.0040)}$$

$$= 0,1869$$

Maka yang terpilih adalah $P_{3,5}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 3 – 5 – 1.

- Transisi =2

$$\rho_{ij} = \{4\} \quad F_{1,2} = 3$$

$$= \left\{ \frac{1}{1} \right\}$$

$$= 1$$

$$P_{1,4} = \frac{(1)(0.0023)}{(1)(0.0035) + (1)(0.0023) + (1)(0.0046)}$$

$$= 0,2212$$

Maka yang terpilih adalah $P_{3,5}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah $3 - 5 - 1 - 4$.

❖ Lebah ke-3

$$\rho_{ij} = \{1, 3, 4, 5\}$$

$$\Theta = \{4, 1, 5, 3\}$$

Nilai dari $P_{ij,n}$ lebah ke-3 dapat dicari dengan menggunakan rumus yaitu sebagai berikut,

$$P_{ij,n} = \frac{[\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}}{\sum_{j \in A_{i,n}} \left([\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}\right)}$$

• Transisi =0

$$\rho_{ij} = \{1, 3, 5\} \quad F_{3,0} = 5$$

$$= \left\{ \frac{1-\lambda}{2}, \frac{1-\lambda}{2}, \lambda \right\}$$

$$= \{0.15, 0.15, 0.7\}$$

$$P_{4,1} = \frac{(0.15)(0.0023)}{(0.15)(0.0023) + (0.15)(0.0039) + (0.15)(0.0040)}$$

$$= 0,2255$$

$$P_{4,3} = \frac{(0.15)(0.0039)}{(0.15)(0.0023) + (0.15)(0.0039) + (0.15)(0.0040)}$$

$$= 0,3824$$

$$P_{4,5} = \frac{(0.7)(0.0040)}{(0.7)(0.0023) + (0.7)(0.0039) + (0.7)(0.0040)}$$

$$= 0,3922$$

Maka yang terpilih adalah $P_{4,5}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 4 – 5.

- Transisi= 1

$$\rho_{ij} = \{1, 3\} \quad F_{4,1} = 3$$

$$= \{\lambda-1, \lambda\}$$

$$= \{0.3, 0.7\}$$

$$P_{5,1} = \frac{(0.3)(0.0046)}{(0.3)(0.0046) + (0.3)(0.0128) + (0.3)(0.0040)}$$

$$= 0,2150$$

$$P_{5,3} = \frac{(0.7)(0.0128)}{(0.7)(0.0046) + (0.7)(0.0128) + (0.7)(0.0040)}$$

$$= 0,5981$$

Maka yang terpilih adalah $P_{5,3}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 4 – 5 – 3.

- Transisi =2

$$\rho_{ij} = \{1\} \quad F_{2,2} = 1$$

$$= \{\lambda\}$$

$$= 0.7$$

$$P_{3,1} = \frac{(0.7)(0.0035)}{(0.7)(0.0035) + (0.7)(0.0039) + (0.7)(0.0128)}$$

$$= 0,1733$$

Maka yang terpilih adalah $P_{3,1}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah $4 - 5 - 3 - 1$.

❖ Lebah ke-4

$$\rho_{ij} = \{1, 3, 4, 5\}$$

$$\Theta = \{5, 3, 1, 4\}$$

Nilai dari $P_{ij,n}$ lebah ke-4 dapat dicari dengan menggunakan rumus yaitu sebagai berikut,

$$P_{ij,n} = \frac{[\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}}{\sum_{j \in A_{i,n}} \left([\rho_{ij}]^{\alpha} \cdot \left[\frac{1}{d_{ij}}\right]^{\beta}\right)}$$

• Transisi =0

$$\rho_{ij} = \{1, 3, 4\} \quad F_{4,0} = 4$$

$$= \left\{ \frac{1-\lambda}{2}, \frac{1-\lambda}{2}, \lambda \right\}$$

$$= \{0.15, 0.15, 0.7\}$$

$$P_{5,1} = \frac{(0.15)(0.0046)}{(0.15)(0.0046) + (0.15)(0.0128) + (0.15)(0.0040)}$$

$$= 0,2150$$

$$P_{5,3} = \frac{(0.15)(0.0128)}{(0.15)(0.0046) + (0.15)(0.0128) + (0.15)(0.0040)}$$

$$= 0,5981$$

$$P_{5,4} = \frac{(0.7)(0.0040)}{(0.7)(0.0046) + (0.7)(0.0128) + (0.7)(0.0040)}$$

$$= 0,1869$$

Maka yang terpilih adalah $P_{5,3}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 5 – 3.

- Transisi= 1

$$\rho_{ij} = \{1, 4\} \quad F_{2,1} = 3$$

$$= \left\{ \frac{1}{2}, \frac{1}{2} \right\}$$

$$= \{0.5, 0.5\}$$

$$P_{3,1} = \frac{(0.5)(0.0035)}{(0.5)(0.0035) + (0.5)(0.0039) + (0.5)(0.0128)}$$

$$= 0,1733$$

$$P_{3,4} = \frac{(0.5)(0.0039)}{(0.5)(0.0035) + (0.5)(0.0039) + (0.5)(0.0128)}$$

$$= 0,1931$$

Maka yang terpilih adalah $P_{3,4}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 5 – 3 – 4 .

- Transisi =2

$$\rho_{ij} = \{1\} \quad F_{3,2} = 1$$

$$= \{ \lambda \}$$

$$= 0.7$$

$$P_{4,1} = \frac{(0.7)(0.0023)}{(0.7)(0.0023) + (0.7)(0.0039) + (0.7)(0.0040)}$$

$$= 0,2255$$

Maka yang terpilih adalah $P_{3,1}$. Sehingga urutan yang dihasilkan pada transisi 0 ini adalah 5 – 3 – 4 - 1.

Proses selanjutnya adalah mencari nilai Pfi, yang mana rumus dari Pfi yakni sebagai berikut,

$$Pfi = \frac{1}{Li}$$

Sehingga urutan rute yang didapat lebah selama *foraging* memiliki nilai Pfi masing-masing yakni,

- Lebah ke-1 = 1 – 5 – 3 – 4

$$\begin{aligned} Pfi &= \frac{1}{Li} \\ &= \frac{1}{546,8909} \\ &= 0,0018 \end{aligned}$$

- Lebah ke-2 = 3 – 5 – 1 – 4

$$\begin{aligned} Pfi &= \frac{1}{Li} \\ &= \frac{1}{727,8716} \\ &= 0,0014 \end{aligned}$$

- Lebah ke-3 = 4 – 5 – 3 – 1

$$\begin{aligned} Pfi &= \frac{1}{Li} \\ &= \frac{1}{613,4416} \end{aligned}$$

$$= 0.0016$$

- Lebah ke-4= 5 – 3 – 4 – 1

$$\begin{aligned} Pfi &= \frac{1}{Li} \\ &= \frac{1}{765,3853} \\ &= 0.0013 \end{aligned}$$

Nilai dari Pfcology dapat dicari dengan menggunakan rumus yaitu sebagai berikut,

$$Pfcology = \frac{1}{Nbee} \sum Pfi$$

Sehingga nilai dari Pfcology sebagaimana dengan menggunakan rumus diatas yakni,

$$\begin{aligned} Pfcology &= \frac{1}{Nbee} \sum Pfi \\ &= \frac{1}{4} \times 0,0061 \\ &= 0,001525 \end{aligned}$$

Nilai berupa Pfi dan Pfcology diatas nantinya digunakan untuk menentukan *profitability rating*. Pada *profitability rating* (diambil dari Nakrani 26 dan Tovey 2004) terdapat empat syarat yaitu,

1. Jika $Pfi < 0.9 Pfcology$, maka Ri bernilai 0.60.
2. Jika $0.9 Pfcology \leq Pfi < 0.95 Pfcology$, maka Ri bernilai 0.20.

3. Jika $0.95 P_{fcolony} \leq P_{fi} < 1.15 P_{fcolony}$, maka R_i bernilai 0.02.

4. Jika $1.15 P_{fcolony} \leq P_{fi}$, maka R_i bernilai 0.

Sehingga dari urutan rute yang telah didapatkan dari tiap-tiap lebah tersebut memiliki nilai R_i yakni,

1. Lebah ke-1, karena memenuhi persyaratan $1.15 P_{fcolony} \leq P_{fi}$, maka R_i bernilai 0
2. Lebah ke-2, karena memenuhi persyaratan $0.9 P_{fcolony} \leq P_{fi} < 0.95 P_{fcolony}$, maka R_i bernilai 0.20.
3. Lebah ke-3, karena memenuhi persyaratan $0.95 P_{fcolony} \leq P_{fi} < 1.15 P_{fcolony}$, maka R_i bernilai 0.02
4. Lebah ke-4, karena memenuhi persyaratan $P_{fi} < 0.9 P_{fcolony}$, maka R_i bernilai 0.60.

Dengan demikian maka pada perhitungan ini dipilih lebah ke-1, dikarenakan bilamana R_i bernilai nol, maka jalur tersebutlah yang dipilih karena dianggap sebagai hasil *foraging* dengan jarak yang paling pendek berdasarkan *random*. Kemudian jalur yang dihasilkan oleh lebah ke-1 tersebut diikuti kembali pada iterasi yang berikutnya hingga ditemukan jarak yang lebih pendek lagi dan akan berhenti iterasinya hingga hasilnya konvergen atau nilai maksimum iterasinya telah tercapai. Sedangkan bilamana nilai R_i tidak nol atau solusi yang baru tidak lebih baik dari solusi sebelumnya, maka digunakan *path* yang lama, atau menggunakan *path* sebelumnya sebagaimana yang telah diinformasikan melalui *waggle dance*.