

# IMPLEMENTASI ALGORITMA LEBAH UNTUK PENCARIAN JALUR TERPENDEK DENGAN MEMPERTIMBANGKAN HEURISTIK

Dian Nurdiana

## ABSTRAK

*Rekomendasi jalur yang optimum sangatlah dibutuhkan oleh para pemudik. Hal ini disebabkan oleh banyaknya permasalahan yang dihadapi pada saat melakukan perjalanan mudik. Ada asumsi bahwa pengambilan rute yang tepat dapat mengurangi waktu dan biaya yang dibutuhkan selama perjalanan mudik. Oleh karena itu, dibutuhkan suatu perhitungan yang dapat merekomendasikan rute yang efisien pada jalur mudik. Salah satu metode yang dapat menyelesaikan permasalahan jalur terpendek adalah algoritma lebah. algoritma lebah itu sendiri terinspirasi dari perilaku sosial koloni lebah dimana seekor lebah dapat menjangkau sumber makanan dengan rute terdekat. Setelah mereka menemukan makanan lebah-lebah akan kembali kesarang dan menginformasikan sumber makan yang dia temukan kepada teman-temannya dengan menggunakan waggle dance. Dalam penelitian ini pencarian jalur terpendek yang dilakukan lebah tidak hanya mempertimbangkan jarak saja, tetapi mempertimbangkan heuristik lainnya seperti kemacetan, lampu jalan, jalan tol, rawan bencana dan keamanan. Sehingga rute yang dihasilkan merupakan rute yang optimum. Hasil yang didapat dari mengimplementasikan algoritma lebah untuk pencarian jalur terpendek dengan mempertimbangkan heuristik adalah rute jalur optimum yang bisa dilalui dari kota awal ke kota tujuan beserta panjang jalur yang dapat ditempuh.*

**Kata Kunci :** Pencarian Jalur Terpendek, Algoritma Lebah (Algorithm Bee Colony), Jalur Mudik.

## 1. PENDAHULUAN

Mudik merupakan salah satu kegiatan tahunan yang terjadi di Indonesia. Hampir seluruh masyarakat di Indonesia melakukan kegiatan ini, terutama masyarakat yang berada di pulau Jawa. Pada saat musim mudik berlangsung banyak sekali permasalahan yang harus dihadapi oleh para pemudik, permasalahan tersebut misalnya kurangnya kelengkapan jalan seperti lampu jalan, kondisi jalan yang rusak, ruas jalan yang berkapasitas sedikit, dll. Sehingga pemilihan rute jalan yang tepat sangat diperlukan pada saat melakukan perjalanan mudik.

Pentingnya pemilihan rute jalan yang tepat bisa menghindari permasalahan yang dihadapi pada saat mudik. Misalnya ketika sebuah jalan yang dilewati sering terjadi

kemacetan maka para pemudik alangkah lebih baiknya mencari jalur yang lain untuk dilalui walaupun jarak yang ditempuh cukup panjang. Hal seperti itu masih jarang dilakukan oleh para pemudik, para pemudik sering kali memaksakan melewati rute yang mempunyai jarak pendek walaupun kondisi jalan yang dilalui sering terjadi kemacetan, sehingga mengakibatkan menumpuknya kendaraan pada ruas jalan tersebut. Hal ini dikarenakan kurangnya informasi mengenai rute jalan yang efisien. Oleh karena itu sangat diperlukan media yang bisa memberikan rekomendasi jalur optimum seperti aplikasi pencarian jalur terpendek dengan mempertimbangkan heuristik, sehingga diharapkan perjalanan para pemudik bisa semakin nyaman karena bisa mengurangi waktu dan biaya.

Applikasi pencarian jalur terpendek yang dibangun ini memanfaatkan teorema graf dalam representasi datanya. Sedangkan dalam penentuan jalur terpendeknya menggunakan algoritma lebah. Algoritma lebah terinspirasi dari perilaku sosial koloni lebah dimana seekor lebah dapat menjangkau sumber makanan dengan rute terdekat. Setelah mereka menemukan sumber makanan kemudian mereka akan kembali ke sarang dan melakukan waggle dance, dengan menggunakan waggle dance semua koloni saling berkomunikasi tentang sumber makanan yang mereka temukan, sehingga lebah-lebah yang lain akan mengetahui letak dari sumber makanan yang paling dekat dari sarang.

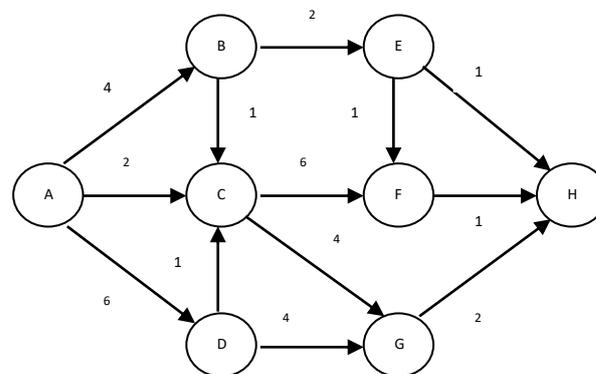
Proses penggabungan informasi dengan pendekatan teorema graf dengan algoritma lebah ini tidak hanya menentukan sebuah jalur terpendek saja, tetapi dalam pencarian jalur terpendeknya mempertimbangkan heuristik. heuristik yang dipertimbangkan yaitu faktor kemacetan, faktor lampu jalan, faktor jalan tol, faktor rawan kecelakaan, faktor keamanan. Sehingga rute jalur yang direkomendasikan merupakan jalur yang optimum untuk dilalui sesuai dengan heuristik yang dipilih.

## 2. Graf

### 2.1 Pengertian Graf

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan dalam kehidupan sehari-hari sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Banyak persoalan pada dunia nyata yang sebenarnya merupakan representasi visual dari graf. Contoh salah satu representasi visual dari graf adalah peta. Banyak hal yang dapat digali dari representasi tersebut, diantaranya adalah menentukan jalur terpendek dari satu tempat ke tempat lain, menggambarkan 2 kota yang bertetangga dengan warna yang berbeda

pada peta, menentukan tata letak jalur transportasi, pengaturan jaringan komunikasi atau jaringan internet dan masih banyak lagi. Selain peta, masih banyak hal lain dalam dunia nyata yang merupakan representasi visual dari graf.



**Gambar 2.1** Graf

Secara matematis, graf didefinisikan sebagai berikut [1] :

Graf  $G$  didefinisikan sebagai pasangan himpunan  $(V,E)$  yang dalam hal ini :

$V$  = himpunan tidak kosong dari simpul - simpul (*vertices* atau *node*):

$$\{v_1, v_2, \dots, v_n\}$$

$E$  = himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul:  $\{e_1, e_2, \dots, e_n\}$

atau dapat ditulis singkat notasi  $G = (V,E)$ . Definisi tersebut menyatakan bahwa  $V$  tidak boleh kosong sedangkan  $E$  boleh kosong. Artinya, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpul harus ada, minimal satu (Munir, 2003: 291).

## 3. Representasi Graf

### 3.1 Matrik Kedekatan (Adjacency Matrik)

Untuk suatu graf dengan jumlah simpul sebanyak  $n$ , maka matrik 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 ABCDEFGH dapat dilihat pada Tabel 2.1 :

**Table 3.1** matriks kedekatan untuk graf ABCDEFGH

	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	0	0
B	1	0	1	0	1	0	0	0
C	1	1	0	1	0	1	1	0
D	1	0	1	0	0	0	1	0
E	0	1	0	0	0	1	0	1
F	0	0	1	0	1	0	0	1
G	0	0	1	1	0	0	0	1
H	0	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).

## 4. Algoritma Lebah

### 4.1 Swarm Intelligence

Perkembangan AI dari tahun ke tahun semakin pesat dan semakin banyak cabangnya. Menurut (ilmukuilmumu, 2009) Algoritma Optimasi terbagi menjadi dua jenis, yaitu algoritma optimasi dengan pendekatan berbasis *deterministic* dan algoritma optimasi dengan pendekatan berbasis *probabilistic*. yang termasuk kedalam algoritma berbasis *deterministic* diantaranya *State Space Search*, *Dynamic Programming*, dan *Branch and Bound*. Sedangkan algoritma optimasi yang termasuk kedalam algoritma yang berbasis pendekatan probabilistic adalah algoritma Monte Carlo dengan berbagai macam turunannya.

*Evolutionary Computation* merupakan salah satu algoritma yang termasuk kedalam algoritma optimasi berbasis probabilistic. definisi dari algoritma *Evolutionary Computation* adalah abstraksi dari teori evolusi biologis yang digunakan untuk membuat prosedur atau metodologi optimasi, biasanya diterapkan

pada komputer, yang digunakan untuk memecahkan masalah. Algoritma ini memiliki ide dasar dari bagaimana proses evolusi yang terjadi pada makhluk hidup. Yang menganggap bahwa hasil setiap evolusi itu akan membawa sesuatu menjadi lebih baik dan optimal.

Beberapa algoritma yang termasuk kedalam algoritma *Evolutionary Computation* diantaranya adalah *swarm intelligent* dimana algoritma ini didasarkan dari kecerdasan berkelompok. Dengan semakin banyaknya anggota kelompok dan terkumpulnya kecerdasan-kecerdasan individual maka akan menyebabkan terkumpulnya kecerdasan kelompok yang luar biasa. Beberapa yang termasuk kedalam algoritma *swarm intelligent* diantaranya *Particle Swarm Optimization*, *Ant Colony Optimization*, *Artificial Bee Colony Optimization*.

*Swarm intelligent* merupakan sebuah metode penyelesaian masalah yang memanfaatkan perilaku sekumpulan agen yang saling bekerja sama. Khususnya *swarm intelligence* pada algoritma lebah (*Algorithm Bee Colony*) terinspirasi dari perilaku sosial koloni lebah dimana seekor lebah bisa menjangkau sumber makanan sekaligus mengingat letak, arah dan jarak dari sumber makanan. Sekembalinya dari pencarian sumber makanan maka seekor lebah akan melakukan *waggle dance*. *Waggle dance* merupakan alat komunikasi yang dilakukan oleh koloni lebah.

### 4.2. Algoritma Lebah

#### 4.2.1. Koloni Lebah

Lebah merupakan serangga sosial yang sangat terorganisir. Kelangsungan hidup sebuah koloni tergantung dari setiap individu yang lain. 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.

Prilaku yang dilakukan oleh koloni lebah masih menjadi misteri selama bertahun-tahun sampai Von Frisch menterjemahkan bahasa isyarat yang berada pada tarian lebah. Tarian lebah ini digunakan sebagai alat komunikasi yang digunakan oleh koloni. 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.

#### 4.2.2. Pembangunan Jalur Oleh Lebah

Menurut (Wong) model yang diusulkan kami, lebah diperbolehkan untuk mengeksplorasi dan mencari jalan tur sampai selesai. Sebelum meninggalkan sarang lebah akan mengamati tarian yang dilakukan oleh lebah lainnya. Kemudian lebah akan diset dengan pengetahuan yang didapatkan dari tarian. Set pergerakan, yaitu "*preferred path*" yang dinotasikan dengan  $\theta$ , maka akan berfungsi sebagai panduan dalam proses mencari makanan.  $\theta$  berisi tur lengkap yang telah di eksplorasi sebelumnya oleh lebah yang akan pergi ke tempat tujuan.

Selama proses pencarian makanan, lebah akan melakukan perjalanan dari satu kota ke kota yang lain sampai mencapai tujuan. 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. *Arc fitness* yang lebih tinggi ditugaskan untuk tepi yang ditugas 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.

#### 4.2.3. Algoritma Lebah Dengan 2-opt untuk TSP

Tingkah laku lebah pada akhirnya menginspirasi seeley (1955) untuk menjadikan 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 grup. Sunil nakrani dari *Oxford University* dan Craig Tovey dari *Georgia Institute Of Technology* menerapkan cara penyelesaian masalah oleh lebah madu tersebut pada permasalahan pada *Internet Host*

Langkah – langkah penyesuaian algoritma lebah sebagai berikut :

##### 1. Forage

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.

$$P_{ij,n} = \frac{[\rho_{ij}]^\alpha \cdot \left[\frac{1}{d_{ij}}\right]^\beta}{\sum_{j \in A_{i,n}} \left([\rho_{ij}]^\alpha \cdot \left[\frac{1}{d_{ij}}\right]^\beta\right)} \quad (4.1)$$

Dimana :

$\rho_{ij}$  = 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.  $\alpha$  merupakan variabel yang berperan sebagai pembobot untuk *arc fitness*, sedangkan  $\beta$  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 akan diungkapkan melalui persamaan di bawah ini.

$$\rho_{ij} = \begin{cases} \lambda & \text{if } i \in [A_{i,n} \cap F_{i,n}] \\ 1 - \lambda & \text{if } i \in [A_{i,n} - F_{i,n}] \\ 1 & \text{if } i \in [A_{i,n} - F_{i,n}] \end{cases} \quad (4.2)$$

$\lambda$  menunjukkan probabilitas dari sebuah kota yang diikuti pada  $\theta$ .  $F_{i,n}$  adalah kumpulan yang berisi satu kota yang mana lebah lebih suka berpindah dari  $i$  pada  $n$  transisi, seperti yang direkomendasikan oleh  $\theta$ . Mari  $\theta(m)$  melambangkan elemen ke- $m$  di  $\theta$ . Jika seekor lebah baru saja memulai eksplorasi dari sarang,  $F_{H,0} = \{\theta(1)\}$ . Jika arus mengunjungi kota  $i$  adalah pada posisi ke- $m$  di jalur yang dipilih setelah transisi  $n$ , maka  $F_{\theta(m),n} = \{\theta(m+1)\}$ .  $F_{i,n}$  hanya berisi satu elemen  $\theta(m+1)$  (kota depan yang bersebelahan dengan  $i$  di  $\theta$ ) hal ini dipertimbangkan.  $F_{i,n}$  dapat memiliki dua kota jika  $\theta(m-1)$  (kota yang berdekatan dibelakang) juga dipertimbangkan

## 2. Waggle Dance

Jika seekor lebah menari, tarian lebah akan berlangsung selama beberapa durasi. Durasi tarian  $D_i$  dari lebah  $i$  ditentukan oleh linear fungsi seperti yang diberikan pada Persamaan. 2.3. Durasi diukur selama iterasi pada algoritma dieksekusi. Menurut fungsi linear, jika lebah  $i$  memiliki  $Pf_i$  yang lebih tinggi, maka akan diberikan kesempatan untuk menari lagi (dance

muncul dalam iterasi lebih). Jika tidak, itu tarian untuk jangka pendek.  $Pf_i$  melambangkan skor profitabilitas lebah  $i$  sebagaimana didefinisikan pada Persamaan. 2.4.  $Pf_{colony}$  menunjukkan koloni lebah dengan rata-rata profitabilitas seperti dalam Persamaan. 2.5 dan diperbarui setelah masing-masing lebah menyelesaikan tur.  $K$  didefinisikan sebagai skala faktor yang mengendalikan besarnya durasi

$$D_i = K \cdot \frac{Pf_i}{Pf_{colony}} \quad (4.3)$$

$$Pf_i = \frac{1}{L_i} \quad L_i = \text{tour length} \quad (4.4)$$

$$Pf_{colony} = \frac{1}{N_{Bee}} \sum_{i=1}^{N_{Bee}} Pf_i \quad (4.5)$$

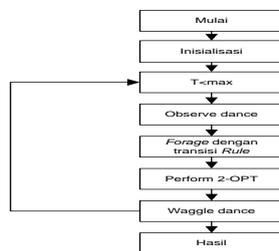
$Pf_i$  dapat ditafsirkan sebagai kuantitas nektar yang dikumpulkan oleh lebah  $i$ . kuantitas yang lebih tinggi dari nektar akan dikumpulkan jika lebah melakukan perjalanan yang panjang dengan rute lebih pendek. Dengan demikian,  $Pf_i$  didefinisikan berbanding terbalik dengan panjang tur. Kemungkinan  $r_i$  mengikuti *path* yang biasa menurut *profitability rating* dari lebah dan koloni pada dasarnya seperti yang diperlihatkan di tabel 1 (diambil dari Nakrani 26 dan Tovey 2004). Terutama, lebah lebih menyukai mengobservasi secara random dan mengikuti *waggle dance* dalam *dance floor* jika *profitability rating* rendah sebagai perbandingan terhadap koloni.

**Tabel 4.1** Penyesuaian Penentuan Mengikuti Waggle Dance

Profitability Scores	$P_{follow} / R_i$
$Pf_i < 0.95 Pf_{colony}$	0.60
$0.95 Pf_{colony} \leq Pf_i < 0.975 Pf_{colony}$	0.20
$0.975 Pf_{colony} \leq Pf_i < 0.99 Pf_{colony}$	0.02
$0.99 Pf_{colony} \leq Pf_i$	0.00

Dalam kasus ekstrim, dimana ri adalah 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.

Penggunaan kebijakan memori yang ditentukan dan tampilan tabel bertujuan untuk menghindari optima lokal. Kedua mekanisme mendorong eksplorasi sehingga lebah lebah dapat mencari solusi-solusi terbaik dari solusi-solusi yang lebih efisien.



**Gambar 4.1** Flow Chart Algoritma Lebah

### 5. Pengujian

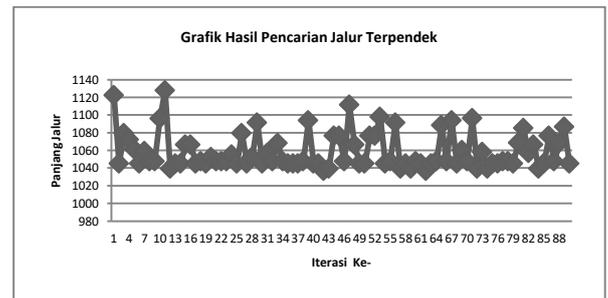
Parameter inputan yang digunakan pada pencarian jalur terpendek algoritma lebah dengan memperimbangan heuristik yaitu :

1. Kota awal yang digunakan dalam pengujian adalah kota Merak.
2. Kota tujuan yang digunakan dalam pengujian adalah kota Banyuwangi.
3. Perbandingan yang digunakan untuk bobot jarak dan bobot faktor adalah 1 : 3.
4. Nilai Apha yang digunakan bernilai 1 sesuai dengan penelitian sebelumnya yang telah dilakukan oleh Dorigo dan Gambaedella (1997).
5. Nilai beta yang digunakan bernilai 2 sesuai dengan penelitian sebelumnya yang telah dilakukan

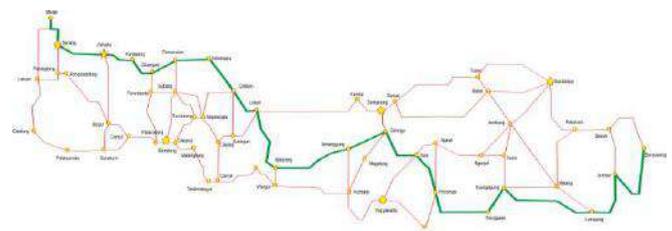
oleh Dorigo dan Gambaedella (1997).

6. Lamda yang digunakan dalam pengujian bernilai 0,7.
7. Jumlah iterasi yang dilakukan pada penelitian sebanyak 100 kali.
8. Jumlah lebah yang dilibatkan dalam pengujian sebanyak 100.

Pencarian jalur terpendek yang dilakukan untuk pengujian sebanyak 100 kali percobaan. Dari 100 kali percobaan tersebut, 90 kali percobaan menghasilkan rute dan panjang jalur terpendek, sedangkan 10 kali percobaan lainnya tidak menghasilkan rute dan panjang jalur terpendek. Data data hasil percobaan dapat di lihat pada lampiran 4.8. Grafik Hasil pencarian jalur terpendek algoritma lebah dengan mempertimbangkan heuristik dapat dilihat pada gambar 4.4.



**Gambar 5.1** Grafik Hasil Pencarian Jalur Terpendek



**Gambar 5.2** Rute Jalur Terpendek dengan Nilai Terendah

Nilai modus pencarian jalur terpendek hasil simulasi adalah 1044,9907 Km. Dengan kemunculan sebesar 26 kali. Jalur rutenya adalah 1=merak - 2=serang - 11=jakarta - 12=karawang - 13=cikampek - 20=pamanukan - 21=indramayu - 22=cirebon - 29=losari - 31=ajibarang -

34=temanggung - 36=salatiga - 37=solo -  
39=ponorogo - 48=trenggarek -  
49=tulungagung - 55=lumajang -  
56=jember - 57=banyuwangi. Nilai modus ini menunjukkan rute dan panjang jalur optimum yang direkomendasikan untuk hasil penelitian. Penentuan hasil optimasi berdasarkan nilai modus dikarenakan pencarian jalur terpendek yang dilakukan algoritma lebah tidak hanya mempertimbangkan jarak saja, tetapi mempertimbangkan heuristik lainnya juga.

Dari hasil simulasi dihasilkan nilai rata-rata sebesar 1058,395 Km dengan standar deviasi sebesar 20,33801 Km. Standar deviasi menunjukkan kisaran yang didapatkan dari hasil simulasi. Kisaran nilai terbesar yang dihasilkan dari simulasi sebesar 1078,7330 Km sedangkan nilai terkecil yang dihasilkan dari percobaan sebesar 1038,0569 Km. Besarnya kisaran yang dihasilkan merupakan panjang jalur yang dapat ditempuh dari kota awal ke kota tujuan.

## 6. Pembahasan

Dari hasil penelitian dengan 100 kali simulasi untuk mencari panjang jalur terpendek, didapat 90 kali simulasi menghasilkan rute dan panjang jalur sedangkan 10 simulasi lainnya tidak berhasil menemukan rute dan panjang jalur. Hal ini dapat dijelaskan sebagai berikut:

Setiap simulasi menggunakan 100 lebah. Setiap lebah akan berusaha untuk mencari rute dan panjang jalur terpendek. Dalam melakukan pencarian rute dan jalur terpendek tersebut, lebah dikatakan berhasil jika:

- a. Lebah memiliki titik awal dan titik akhir yang berbeda. Hal ini sesuai dengan konsep atau teori jalur terpendek, yaitu titik awal dan titik akhir berbeda.
- b. Lebah hanya boleh mengunjungi satu titik satu kali. Hal ini

memenuhi teori atau konsep *Travelling Salesman Problem* (TSP). Konsep TSP digunakan karena sebenarnya TSP juga merupakan permasalahan untuk mencari jalur terpendek dengan titik awal dan titik akhir yang sama.

Jika lebah tidak mampu memenuhi kedua syarat tersebut maka lebah dikatakan tidak berhasil atau gagal menemukan rute dan jalur terpendek. Inilah alasan mengapa dari 100 kali simulasi yang dilakukan, 90 kali simulasi mendapatkan rute sedangkan panjang jalur sedangkan 10 simulasi lainnya tidak berhasil menemukan rute dan panjang jalur.

Hal lain yang menarik adalah bahwa setiap simulasi menghasilkan rute dan panjang jalur yang berbeda. Perbedaan ini dapat dilihat pada grafik di sub bab 4.4 sebelumnya yang tidak membentuk garis lurus dari simulasi pertama hingga simulasi ke seratus. Perbedaan rute dan panjang jalur yang dihasilkan pada setiap simulasi menunjukkan bahwa permasalahan jalur terpendek merupakan masalah optimasi yang menggunakan probabilitas. Penggunaan probabilitas pada algoritma ABC ini dapat dilihat pada proses transition rule (pemilihan kota yang akan dikunjungi berikutnya), yaitu :

$$P_{ij,n} = \begin{cases} \frac{[\rho_{ij,n}] \cdot \left( \frac{a}{d_{ij}} \right)^{\alpha+b} \left( \frac{f_1+f_2+f_3+f_4+f_5}{\sum(f)*m} \right)^{\beta}}{a+b} \\ \sum_{j \in A_{ij,n}} \left[ [\rho_{ij,n}] \cdot \frac{\left( \frac{a}{d_{ij}} \right)^{\alpha+b} \left( \frac{f_1+f_2+f_3+f_4+f_5}{\sum(f)*m} \right)^{\beta}}{a+b} \right] \cdot .1 \end{cases} \quad (1)$$

$j_i^k = 0$ , jika  $j \notin j_i^k$   $m =$  nilai maksimum dari  $f$ .

Dengan adanya rumus tersebut, setiap lebah akan memilih kota yang akan dikunjungi berikutnya secara acak. Pemilihan secara acak inilah yang menyebabkan setiap lebah (simulasi) akan

menghasilkan rute dan panjang jalur yang berbeda.

## 7. Kesimpulan

Penelitian ini telah berhasil mengimplementasikan algoritma lebah untuk menyelesaikan pencarian jalur terpendek secara optimal. Perangkat lunak yang dibangun berhasil merekomendasikan jalur mudik yang paling optimum dari kota pemberangkatan ke kota tujuan kepada pengguna.

Pencarian jalur terpendek yang dilakukan untuk pengujian sebanyak 100 kali percobaan. Dari 100 kali percobaan tersebut, 90 kali percobaan menghasilkan rute dan panjang jalur terpendek, sedangkan 10 kali percobaan lainnya tidak menghasilkan rute dan panjang jalur terpendek. Modus pencarian jalur terpendek hasil simulasi adalah 1044,9907 Km. Dengan kemunculan sebesar 26 kali. Nilai rata – rata sebesar 1058,395 Km dengan standar deviasi sebesar 20,33801 Km. Dari standar deviasi menunjukkan hasil simulasi yang didapat mempunyai nilai terbesar 1078,7330 Km dan nilai terkecil 1038,0569 Km.

## 8. Saran

Agar perangkat lunak pencarian jalur terpendek ini semakin baik maka disarankan untuk penelitian selanjutnya adalah :

1. Menggunakan nilai yang valid untuk mengisi nilai heuristik yang dipertimbangkan, sehingga hasil rekomendasi pencarian jalur terpendek yang dihasilkan semakin baik hasilnya.
2. *User* bisa memasukkan koordinat dan kota-kota yang terlibat dalam pencarian jalur terpendek secara manual.
3. Perangkat lunak yang dibangun bersifat *mobile system*, sehingga mudah untuk dibawa.

## 9. Daftar Pustaka

- Anonim, (2009).”Teknik Optimasi”. [online]. Tersedia : <http://ilmukuilmumu.wordpress.com/2009/11/12/teknik-optimasi/>. (31 Oktober 2010).
- Anugraha, R. (2009). “Jarak di Permukaan Bumi”. [online]. Tersedia: <http://www.eramuslim.com/syariah/ilmu-hisab/cetak/jarak-di-permukaan-bumi>. [6 Juli 2010].
- Bonabeau, E., Dorigo, M. dan Theraulaz, G. (1999). “Swarm Intelligence From Natural to Artificial Systems”. New York: Oxford University Press.
- Budianto, A.P. “Penerapan Graf Untuk Struktur Data Himpunan Saling Lepas”. [online] tersedia : <http://www.informatika.org/~rinaldi/Matdis/2006-2007/Makalah/Makalah0607-79.pdf>. (25 Oktober 2010).
- Chong, C. S. , Sivakumar, A.I , Low, M. Y. H. And Gay, K.L. (2006). “A Bee Colony Optimization Algorithm To Job Shop Scheduling”. [online]. Tersedia : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.109.3897&rep=rep1&type=pdf>. (27 Maret 2010).
- Chong, C. S. , Sivakumar, A.I , Low, M. Y. H. And Gay, K.L. “Using A Bee Colony Algorithm For Neighborhood Search In Job Shop Scheduling Problems”. [online]. Tersedia : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.9011&rep=rep1&type=pdf>. (27 Maret 2010).
- Fredivianus, N.(2009)” Organic Computing: Rahasia Aturan Sederhana”. [online]. Tersedia : [http://www.forkom-german.org/index.php?option=com\\_c](http://www.forkom-german.org/index.php?option=com_c)

- [ontent&view=article&id=130:organizational-computing-rahasia-aturan-sederhana&catid=34:tausiyah&Itemid=67](#). (31 Oktober 2010).
- Kusumadewi.S, Artificial Intelligence (Teknik dan Aplikasinya), Edisi 2, Penerbit Graha Ilmu, 2002.
- Munir, R. (2003). Matematika Diskrit Edisi Kedua. Bandung. Penerbit informatika.
- Nakrani, S. and Tovey, C.. (2004) "On honey bees and dynamic server allocation in Internet hosting centers," Adaptive Behavior, vol. 12, no. 3-4, pp.223-240, 2004.[online]. Tersedia : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.8534&rep=rep1&type=pdf#page=117>. [23 Juni 2010].
- Pham D.T., dkk. (2005). "The Bees Algorithm – A Novel Tool for Complex Optimisation Problems". [online]. Tersedia : <http://www.bees-algorithm.com/modules/2/4.pdf>. (27 Maret 2010).
- Pressman, R.S. (2001). Software Engineering A Practitioner Approach Fifth Edition. New York:McGraw-Hill.
- Saidah, N. H. (2010)." Implementasi Algoritma Optimasi Bee Colony Untuk Penjadwalan Job Shop". [online]. Tersedia : <http://digilib.its.ac.id/public/ITS-Undergraduate-9833-Paper.pdf>. (19 Mei 2010).
- Utami, N.(2010)."Implementasi Algoritma Max-Min Ant System Dalam Pencarian Jalur Terpendek (Studi Kasus Pada Pencarian Jalur Terpendek Pipa Transmisi Gas)". Sekripsi tidak terpublikasikan, Bandung : Universitas Pendidikan Indonesia.
- Wismabahasa.(2007),"Fenomena Mudik Lebaran".[online]. Tersedia : <http://wismabahasa.wordpress.com/2007/10/16/fenomena-mudik-lebaran/>. (6 Juli 2010)
- Wong L. P., Low M. Y. H., and Chong C. S.. (2008). "A bee colony optimization algorithm for traveling salesman problem," in Proceedings of Second Asia International Conference on Modelling & Simulation (AMS 2008), 2008. pp. 818-823.[online]. Tersedia : <http://web.mysites.ntu.edu.sg/yhlow/public/Shared%20Documents/papers/tsp-indin08.pdf>. [6 Juli 2010]
- Wong, L. P, Low, M. Y. H. and Chong, C. S. "Bee Colony Optimization with Local Search for Traveling Salesman Problem". [online]. Tersedia : <http://web.mysites.ntu.edu.sg/yhlow/public/Shared%20Documents/papers/tsp-indin08.pdf>. (27 Maret 2010).
- Wong, L. P, Low, M. Y. H. and Chong, C. S. "A bee colony optimization algorithm with the fragmentation state transition rule for traveling salesman problem". [online]. Tersedia : <http://web.mysites.ntu.edu.sg/yhlow/public/Shared%20Documents/papers/iproms09-bco.pdf>. (27 Maret 2010).
- Wong, L. P., Chong C. S. "An Efficient Bee Colony Optimization Algorithm for Traveling Salesman Problem using Frequency-based Pruning". [online]. Tersedia : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.7883&rep=rep1&type=pdf>. (27 Maret 2010)



