IMPLEMENTASI DAN ANALISA KINERJA ALGORITMA *ANT SYSTEM* (AS) DALAM PENYELESAIAN *MULTIPLE TRAVELLING SALESMAN PROBLEM* (MTSP)

Boko Susilo, Rusdi Efendi, Siti Maulinda
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu
E-mail :adiluhung05@yahoo.com, r_efendi@yahoo.com, lindsay.maulinda@gmail.com

ABSTRAK
Penelitian ini bertujuan untuk membangun dan menganalisa kinerja suatu sistem algoritma Ant System (AS) untuk penyelesaian Multiple Travelling Salesman Problem (MTSP). MTSP adalah permasalahan distribusi yang membutuhkan lebih dari satu salesman untuk mengunjungi sejumlah titik dan kembali ke titik awal. Sistem dibangun dengan menggunakan pemrograman Delphi 7 dan database MySQL. Hasil dari keseluruhan proses pada sistem ditampilkan dalam bentuk teks maupun visualisasi yang menunjukkan rute perjalanan dari setiap salesman. Sistem ini cukup efektif dalam penentuan rute dan jarak minimum untuk permasalahan MTSP tersebut. Pengujian dengan kasus-kasus yang berbeda menunjukkan adanya pengaruh jumlah titik, jumlah salesman dan nilai parameter (α, β dan ρ) terhadap performa algoritma.
Kata kunci: *Ant System (AS), MTSP, rute, jarak*

1. PENDAHULUAN

Multiple Travelling Salesman Problem (MTSP) adalah pengembangan dari permasalahan TSP. MTSP adalah permasalahan distribusi yang membutuhkan lebih dari satu salesman untuk mengunjungi sejumlah titik dan kembali ke titik awal.

Penelitian yang telah dilakukan Leksono (2009) menemukan bahwa salah satu algoritma *Ant Colony Optimization* (ACO) yaitu pendekatan *Ant System* (AS) merupakan pendekatan yang bagus untuk mencari solusi yang mendekati minimum untuk permasalahan *Travelling Salesman Problem* (TSP). Oleh karena algoritma AS telah terbukti cukup mampu dalam pemecahan masalah TSP, maka tujuan penelitian ini adalah membangun suatu aplikasi algoritma AS untuk penyelesaian kasus TSP yang lebih kompleks yaitu *Multiple Travelling Salesman Problem* (MTSP). *Ant System* (AS) adalah algoritma yang mengadaptasi perilaku/aktivitas semut dalam mencari jalur terpendek ke suatu sumber dan dapat digunakan untuk menyelesaikan kasus seperti MTSP. Bagian yang akan dibahas juga mencakup pengaruh nilai parameter-parameter yang digunakan algoritma AS dalam proses pembangunan solusi yang minimum.

Dalam pengembangan sistem, batasan masalah yang digunakan melingkupi hal-hal sebagai berikut :

- Jenis graf yang digunakan adalah graf lengkap (*complete graph*) tak-berarah dan memiliki bobot yang merepresentasikan jarak antartitik.
- Jenis permasalahan MTSP yang dibahas adalah kasus depot tunggal. Pada penelitian ini, Titik ke-1 ditetapkan sebagai titik yang dijadikan depot.
- Jumlah salesman yang diperbolehkan tidak boleh melebihi aturan yang telah disesuaikan dengan jumlah titik.
- Visualisasi graf berupa simulasi tanpa skala dengan jarak sebenarnya.

Dengan demikian, penelitian ini diharapkan dapat menjadi salah satu referensi dalam pengembangan penyelesaian kasus MTSP selanjutnya.

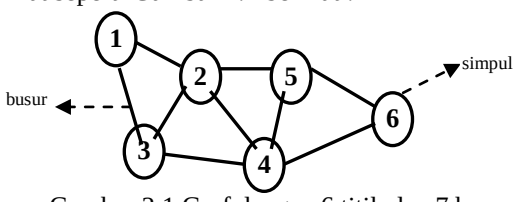
2. LANDASAN TEORI

2.1 Graf

Menurut Pradhana (2006) "Graf adalah himpunan simpul yang dihubungkan dengan busur-busur. Setiap busur diasosiasikan dengan tepat dua simpul".

Munir (2006) menyatakan graf *G* secara matematis sebagai pasangan himpunan (*V,E*) dimana *V* = himpunan tidak kosong dari simpul – simpul v_i : $\{v_1,v_2,...,v_n\}$, *E* = himpunan busur yang menghubungkan sepasang simpul v_j : $\{e_1,e_2,...,e_n\}$. Busur $e_i = (v_i,v_j)$ adalah pasangan simpul dengan $v_i,v_j \subset V$ dan nilai $i,j = 1,2,3,\ldots$

Berdasarkan pengertian diatas, dapat disimpulan bahwa sebuah graf *G* didefinisikan sebagai pasangan (*V,E*), dimana *V* adalah sekumpulan titik dan *E* adalah relasi biner pada *V* yang artinya *E* adalah kumpulan busur yang dapat menghubungkan titik-titik *V*. Contoh graf dapat dilihat seperti Gambar 2.1 berikut :

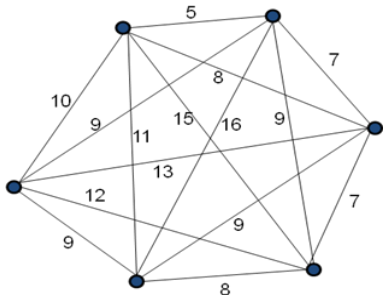


Gambar 2.1 Graf dengan 6 titik dan 7 busur

Pemodelan graf pada kasus MTSP yang akan dibahas pada penelitian ini terbatas pada graf yang termasuk jenis graf lengkap (*complete graph*) tak-berarah dan memiliki bobot yang merepresentasikan

 ISSN: 1907-5022

jarak. Graf lengkap adalah graf sederhana yang setiap titiknya mempunyai busur ke semua titik lainnya. Graf tak-berarah adalah graf yang busurnya tidak memiliki orientasi arah. Sedangkan Graf berbobot (*weighted graph*) adalah graf yang setiap busurnya diberi harga (bobot) yang dalam penelitian ini nilai bobot adalah panjang jarak. Gambar 2.2 menunjukkan contoh graf lengkap yang tak-berarah dan berbobot.



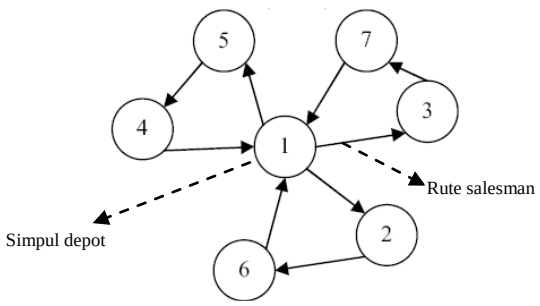
Gambar 2.2 Contoh graf lengkap yang tak-berarah dan berbobot

2.2 Multiple Travelling Salesman Problem

Dalam teorema graf, MTSP dapat digambarkan sebagai berikut : Bila diketahui suatu graf berbobot penuh dan lengkap (dimana tiap titiknya merepresentasikan titik-titik, tiap busur merepresentasikan tiap jalan dari satu titik ke titik lainnya, dan tiap bobot busur merepresentasikan jarak atau biaya yang dibutuhkan untuk menempuh jalan dari satu titik ke titik lainnya), maka tujuan penyelesaian MTSP adalah mencari sirkuit-sirkuit Hamilton dari setiap perjalanan salesman yang bobot totalnya paling kecil. Sirkuit Hamilton adalah suatu lintasan pada graf yang melalui setiap titik tepat satu kali dan akan kembali ke titik awal, tetapi tidak perlu melalui melalui semua sisi yang ada pada graf. Sirkuit hamilton setiap salesman selanjutnya akan disebut rute salesman.

Contoh solusi MTSP dapat dilihat pada Gambar 2.3. Gambar tersebut adalah contoh solusi yang dikerjakan oleh 3 orang salesman. Depot (agen pusat) diberi tanda dengan titik 1. Sedangkan rute-rute yang berhasil dibentuk ketiga salesman sebagai berikut :

- Salesman 1 : 1 → 3 → 7 → 1
- Salesman 2 : 1 → 2 →6 → 1
- Salesman 3 : 1 → 5 → 4 → 1



Gambar 2.3 Contoh solusi MTSP (Junjie dan Dingwei, 2007)

Menurut Setiyono (2002) secara matematis persoalan MTSP dapat diformulasikan sebagai berikut:

$$Z = \min\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}\right\} \quad (2.1)$$

Kendalanya adalah :

$$\sum_{i=1}^{n} x_{ij} = 1 \; untuk \; j = 1,2,3,\ldots,n-1 \quad (2.2)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \; untuk \; i = 1,2,3,\ldots,n-1 \quad (2.3)$$

$$\sum_{i=1}^{n} x_{i1} = m \quad (2.4)$$

$$\sum_{j=1}^{n} x_{1j} = m \quad (2.5)$$

Dengan
Z : total nilai minimum
x_{ij} = 1, jika ada perjalanan salesman dari titik *i* menuju titik *j*
x_{ij} = 0, jika tidak ada perjalanan salesman dari titik *i* menuju titik *j*
c_{ij} : menyatakan jarak antara titik i menuju titik j dengan *i* dan *j* = 1,2,3…

Persamaan (2.1) merupakan fungsi objektif yang akan dicari. Persamaan (2.2) dan (2.3) menjamin bahwa setiap titik hanya dikunjungi sekali, sedangkan persamaan (2.4) dan (2.5) menjamin bahwa sejumlah *m* salesman melakukan rute perjalanan.

2.3 Ant System (AS)

Menurut Dorigo dan Gambardella (dalam Leksono, 2009), *Ant Colony Optimization (ACO)* diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut. Sedangkan *Ant System* (AS) sendiri adalah sebuah algoritma dalam kelompok ACO yang diaplikasikan untuk masalah TSP. Beberapa jenis pendekatan ACO yang lain yaitu *Ant System* (AS), *Elitist Ant System* (EAS), *Rank-based Ant System* (ASRank), dan *Max-min Ant System* (*MMAS*).

Secara informal, khususnya untuk kasus TSP murni, AS bekerja sebagai berikut : Setiap semut memulai rutenya melalui sebuah titik yang dipilih secara acak (setiap semut memiliki titik awal yang berbeda). Secara berulang kali, satu per satu titik yang ada dikunjungi oleh semut dengan tujuan untuk menghasilkan sebuah rute. Pemilihan titik-titik yang akan dilaluinya didasarkan pada suatu fungsi

probabilitas, dinamai aturan transisi status (*state transition rule*), dengan mempertimbangkan *visibility* (invers dari jarak) titik tersebut dan jumlah feromon yang terdapat pada ruas yang menghubungkan titik tersebut.

Perbedaan mendasar sistem kerja AS pada kasus TSP dan MTSP terletak pada inisialisasi awal semut. Seperti dijelaskan diatas, pada kasus TSP, inisialisasi awal dilakukan dengan meletakkan semut-semut pada titik awal secara acak, dimana setiap semut akan memiliki titik awal yang berbeda. Kemudian semut-semut tersebut harus mengunjungi semua titik yang belum dikunjungi untuk menghasilkan rute-nya masing-masing. Sedangkan untuk kasus MTSP, inisialisasi awal dilakukan dengan meletakkan semut-semut pada suatu titik awal (yang akan dianggap sebagai depot tunggal). Kemudian semut pertama terlebih dahulu menempuh sejumlah titik yang belum dikunjungi. Misalkan jumlah titik yang harus dikunjungi semut ke-*i* ditetapkan dengan notasi tn_i, maka ketentuan nilainya harus mengikuti persamaan berikut (Junjie dan Dingwei, 2007):

$$2 \le tn_i \le l_i \tag{2.6}$$

$$\sum_{i}^{m} tn_i = n - 1 \tag{2.7}$$

Dengan
l_i : jumlah maksimal titik yang dapat ditempuh semut *i*
m : jumlah salesman
n : jumlah titik dengan *i* = 1,2,3,...

Saat semut *i* sedang membangun rute perjalanan, jika jumlah titik yang dikunjungi semut *i* sama dengan tn_i, maka hal ini berarti bahwa semut selanjutnya siap untuk memulai perjalanannya dari depot. Semut selanjutnya harus mengunjungi titik-titik yang belum dikunjungi semut sebelumnya. Titik-titik yang telah dikunjungi akan disimpan dalam *tabulist*. Setelah semua titik telah berhasil dikunjungi, maka perhitungan jarak perjalanan setiap semut akan dijumlahkan.

Aturan-aturan AS yang berlaku pada kasus TSP juga digunakan pada penyelesaian kasus MTSP, namun dengan penyesuaian lebih lanjut. Aturan-aturan tersebut adalah sebagai berikut :

2.3.1 Aturan Transisi Status (*State Transition Rule*)

Transisi status dapat disamakan dengan perpindahan satu titik ke titik lain. Anggap bahwa probabilitas semut *k* bergerak dari titik *i* ke titik *j* adalah P_{ij}^{k}, menggunakan rumus probabilitas berikut (Junjie dan Dingwei, 2007) :

$$P_{ij}^{k} \begin{cases} \dfrac{[\tau_{ij}(t)^{\alpha}][\eta_{ij}{}^{\beta}]}{\sum_{j \in A_k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} & \text{Jika } j \in A_k \\ 0 & \text{Jika sebaliknya} \end{cases} \tag{2.8}$$

Dengan
i dan *j* : indeks titik dengan *i* dan *j* = 1,2,3,…,*n-1* dengan *n* adalah banyak titik
τ_{ij} : intensitas feromon pada busur (*i*,*j*) yang akan selalu diperbaharui pada setiap siklus
d_{ij} : jarak dari titik *i* ke titik *j*. Jika jarak antartitik hanya diketahui koordinatnya saja maka $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y}$.
η_{ij} : fungsi bobot pilihan (visibilitas antartitik) atau invers jarak. Nilai $\eta_{ij} = (d_{ij})^{-1}$ atau

A_k : himpunan titik-titik yang semut *k* belum kunjungi. A_k = *n-tabu*$_k$ *(t)* dengan *tabu*$_k$ *(t)* menunjukkan titik-titik yang semut *k* telah kunjungi dan *n* menunjukkan banyak titik.
α : tetapan pengendali intensitas feromon semut dengan nilai (α) > 0
β : tetapan pengendali visibilitas dengan nilai (β) > 0.

2.3.2 Aturan Pembaruan Feromon (*Update Pheromone Trail*)

Pembaruan feromon adalah elemen kunci untuk mengadaptasi teknik pembelajaran ACO. Peranan dari aturan pembaruan feromon ini adalah untuk mengacak arah rute-rute yang sedang dibangun, sehingga titik-titik yang telah dilewati sebelumnya oleh rute seekor semut mungkin akan dilewati kemudian oleh rute semut yang lain. Dengan kata lain, pengaruh dari pembaruan ini adalah untuk membuat tingkat ketertarikan ruas-ruas yang ada berubah secara dinamis: setiap kali seekor semut menggunakan sebuah ruas maka ruas ini dengan segera akan berkurang tingkat ketertarikannya (karena ruas tersebut kehilangan sejumlah feromonnya), secara tidak langsung semut yang lain akan memilih ruas-ruas lain yang belum dikunjungi. Konsekuensinya, semut tidak akan memiliki kecenderungan untuk berkumpul pada jalur yang sama, sehingga terdapat kemungkinan yang lebih tinggi dimana salah satu dari mereka akan menemukan solusi yang lebih baik daripada jika mereka semua berkumpul dalam rute yang sama. Hal ini dilakukan dengan persamaan harga feromon berikut :

$$\tau_{ij} = \rho\tau_{ij} + \Delta\tau_{ij} \tag{2.9}$$

Dengan
ρ : parameter kecepatan penguapan feromon dengan nilai $0<\rho\le1$
$\Delta\tau_{ij}$: perubahan harga intensitas jejak feromon dalam busur yang telah ditempuh pada rute semut *k*.
Harga perubahan $\Delta\tau_{ij}$ ini dapat ditentukan menggunakan persamaan berikut :

 ISSN: 1907-5022

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (2.10)$$

Dengan
k : indeks semut dengan nilai k = 1,2,3,…, *m-1*
m : jumlah semut
$\Delta\tau_{ij}^{k}$ adalah perubahan harga intensitas jejak feromon antartitik setiap semut yang dihitung berdasarkan persamaan berikut :

$$\Delta\tau_{ij}^{k} \begin{cases} {}^{Q}/_{L_k}, & \text{jika menggunakan busur (i,j)} \\ 0, & \text{jika sebaliknya} \end{cases} \qquad (2.11)$$

Dengan
Q : tetapan siklus semut yang bernilai konstan
L_k : jarak yang telah ditempuh semut-*k*
L_k dapat dihitung dengan menggunakan persamaan berikut :

$$L_k = \sum_{s=1}^{n-1} d_{tabu_k(s),tabu_k(s+1)} + d_{tabu_k(n),tabu_k(1)} \qquad (2.12)$$

Dengan
s : indeks titik dengan nilai = 1,2,3,…, *n-2*
$d_{tabu_k(s),tabu_k(s}$: jarak busur dari titik *s* sampai ke titik *s+1* pada *tabulist* yang ditempati semut *k*
$d_{tabu_k(n),tabu_}$: jarak antara titik *n* (akhir) dan titik pertama (awal) pada *tabulist* yang ditempati semut *k*.

Dalam hal ini, nilai L_k sama artinya dengan panjang sirkuit hamilton bagi setiap semut (salesman). Perhitungan total jarak yang ditempuh semua semut dapat dihitung dengan persamaan berikut :

$$Total = \sum_{k=1}^{m} L_k \qquad (2.13)$$

Dengan demikian, model penyelesaian MTSP dengan menggunakan algoritma AS diformulasikan menjadi persamaan berikut :

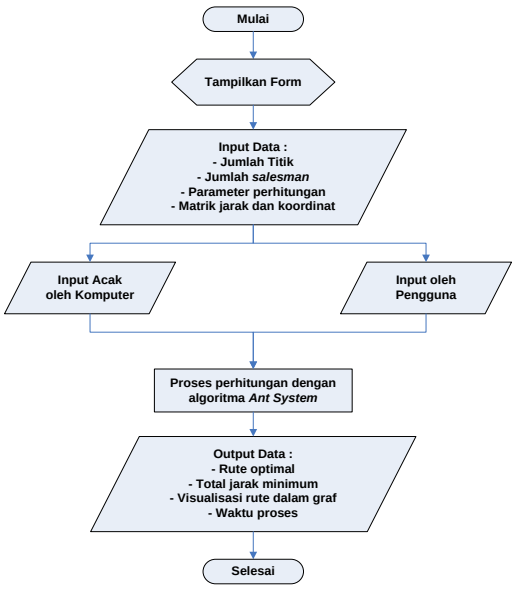
$$Z = min \sum_{k=1}^{m} L_k \qquad (2.14)$$

Dengan
m : jumlah semut (salesman)
Z : nilai minimum yang dicari
k : indeks semut dengan nilai k = 1,2,3,…, *m-1*

3. Analisis dan Perancangan Sistem

3.1 Flowchart Sistem

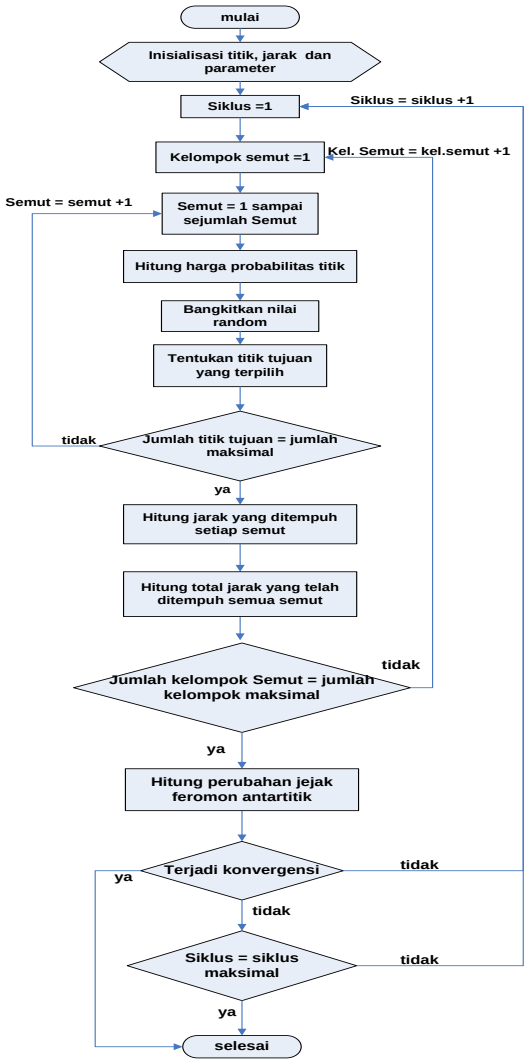
F*lowchart* sistem secara umum yang menggambarkan alur kerja sistem dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Flowchart* Sistem Aplikasi Secara Umum

3.2 Flowchart Algoritma AS

F*lowchart* algoritma *Ant System* (AS) yang menggambarkan alur tahapan algoritma dapat dilihat pada Gambar 3.5.



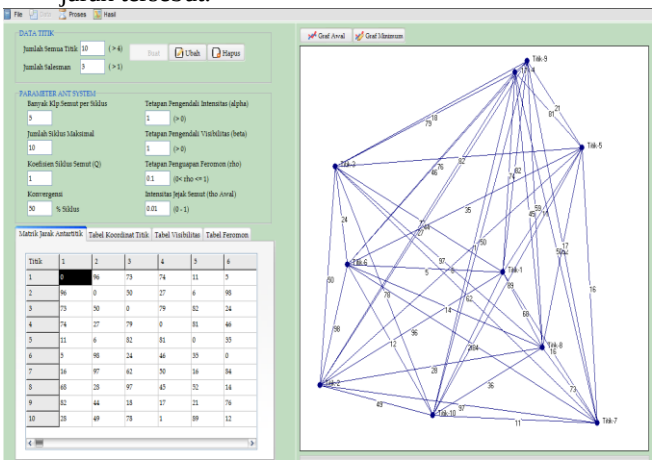
Gambar 3.5 *Flowchart* Algoritma *Ant System*

 ISSN: 1907-5022

3.3 Implementasi Antarmuka

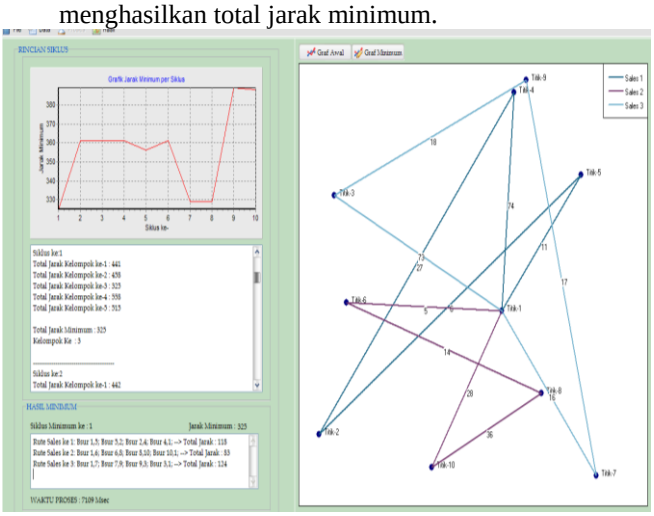
Implementasi antarmuka untuk pengguna dapat dilihat pada Gambar 3.6 dan Gambar 3.7.

Antarmuka Input terdiri dari Panel Isian Jumlah Titik, Panel Isian Parameter, dan Tabel-tabel Bantuan yang melingkupi Matriks Jarak Antartitik, Tabel Koordinat, Tabel Visibilitas dan Tabel Feromon yang menampilkan intensitas feromon pada setiap busur. Setelah semua input dimasukkan, maka sistem akan menampilkan jarak secara acak, kemudian Panel Graf Awal akan menampilkan bentuk graf lengkap yang sesuai dengan panjang jarak tersebut.



Gambar 3.6 Antarmuka Input

Antarmuka Output terdiri dari grafik minimum per Siklus, rincian per siklus, serta hasil minimum yang didapatkan dari siklus-siklus tersebut. Sedangkan Panel Graf Minimum akan menampilkan bentuk graf lintasan setiap salesman yang menghasilkan total jarak minimum.



Gambar 3.7 Antarmuka Output

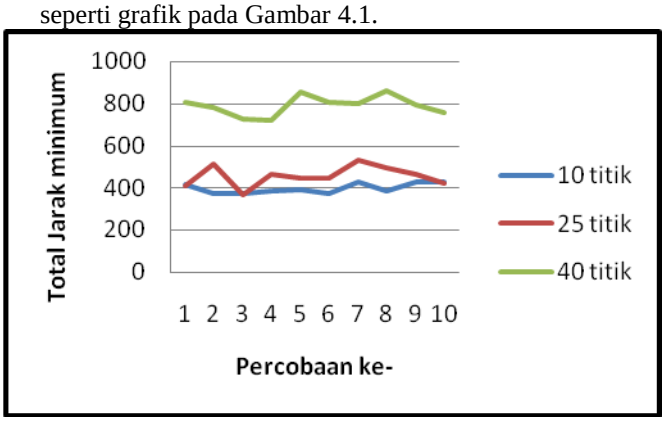
4. HASIL DAN PEMBAHASAN

Implementasi untuk menganalisis kinerja sistem dibutuhkan untuk mengetahui kehandalan sistem dan keoptimalan hasil yang didapatkan. Analisis yang dilakukan melingkupi perubahan masukan jumlah titik, jumlah salesman, dan jumlah salesman.

4.1 Analisis Berdasarkan Jumlah Titik

Analisis berdasarkan jumlah titik ini dilakukan dengan memberikan masukan yang menurut spesifikasi awal dan pengetahuan yang diijinkan. Pengujian dilakukan dengan masukan data kecil (10 titik dengan 2 salesman), data menengah (25 titik dengan 5 salesman), dan data besar (40 titik dengan 8 salesman) yang masing-masing dilakukan sebanyak 10 kali dengan data yang sama.

Hasil pengujian tersebut dapat ditampilkan seperti grafik pada Gambar 4.1.



Gambar 4.1 Hasil Pengujian Berdasarkan Jumlah Titik

Dari hasil pengujian tersebut maka dapat diambil beberapa kesimpulan sebagai berikut :

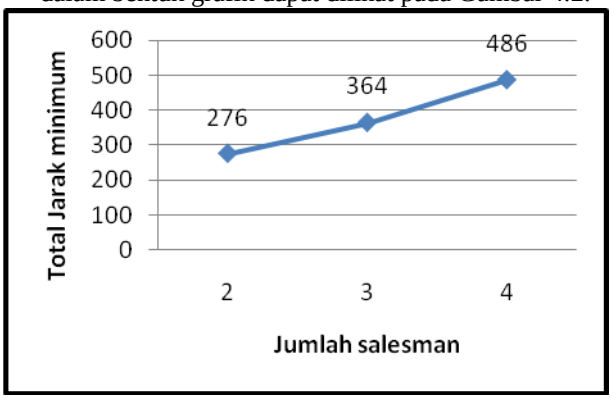
1. Semakin banyak jumlah titik, maka semakin banyak waktu proses yang dibutuhkan untuk mendapatkan jarak minimum.
2. Semakin banyak jumlah titik, maka semakin banyak variasi hasil jarak minimum yang didapatkan. Hal ini disebabkan peningkatan jumlah probabilitas titik yang akan dipilih selanjutnya.
3. Semakin sedikit jumlah titik, semakin stabil jarak minimum yang dihasilkan pada setiap siklus. Dengan kata lain, semakin sedikit jumlah titik, maka akan semakin dekat dengan nilai paling minimal. Hal ini terlihat pada grafik yang menunjukkan bahwa kurva grafik pada pengujian data kecil lebih terlihat stabil karena menghasilkan jarak minimum dengan selisih yang relatif kecil. Sedangkan pengujian data sedang dan data besar relatif kurang stabil karena menghasilkan jarak-jarak dengan perbedaan selisih yang besar.
4. Semakin sedikit jumlah titik, semakin cepat terjadinya konvergensi. Hal ini terlihat bahwa proses pada pengujian data sedang dan data besar selalu berhenti pada siklus maksimal yang telah ditentukan yaitu 10 siklus, sedangkan pengujian data kecil terdapat 5 hasil pengujian yang prosesnya telah berhenti sebelum mencapai siklus maksimal.

4.2 Analisis Berdasarkan Jumlah Salesman

Analisis terhadap jumlah salesman diperlukan untuk mengetahui pengaruh jumlah salesman

 ISSN: 1907-5022

terhadap jarak minimum yang dihasilkan. Pada pengujian ini akan digunakan 10 titik. Untuk pengujian dengan titik, maka jumlah salesman yang dibolehkan yaitu : 2, 3, dan 4. Setiap pengujian menggunakan jarak antartitik yang sama dan setiap salesman dilakukan sebanyak 5 kali. Hasil pengujian dalam bentuk grafik dapat dilihat pada Gambar 4.2.



Gambar 4.2 Grafik Hasil Minimum Berdasarkan Jumlah Salesman

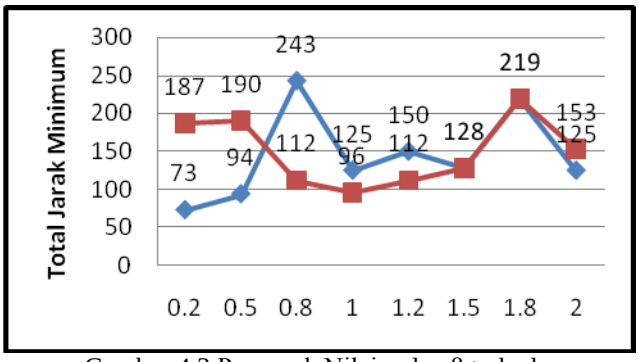
Dari hasil pengujian tersebut dapat disimpulkan bahwa :

1. Semakin banyak jumlah salesman maka semakin panjang jarak yang didapatkan dan semakin lama waktu proses yang dibutuhkan. Dengan kata lain, peningkatan jumlah salesman pada jumlah titik yang sama akan memperpanjang jarak jarak. Hal ini dikarenakan semakin banyak tambahan jarak busur untuk menutup rute masing-masing salesman kembali ke titik awal (depot).
2. Grafik peningkatan jumlah salesman untuk jumlah titik yang sama membentuk grafik yang mendekati linier.

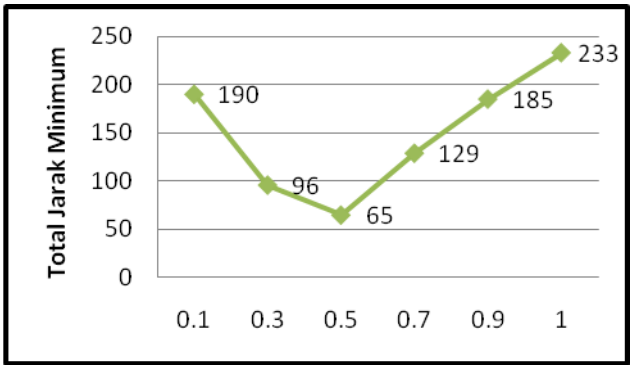
4.3 Analisis Berdasarkan Nilai Parameter

Parameter – parameter α, β, dan ρ mempunyai pengaruh terhadap performa hasil yang diperoleh pada algoritma *Ant System*. Pada pengujian ini akan dibahas nilai – nilai parameter tersebut terhadap akurasi hasil perhitungan minimum untuk penyelesaian masalah MTSP. Nilai parameter-parameter tersebut akan diuji dengan nilai $\alpha \in \{0.2; 0.5; 0.8; 1; 1.2; 1.5; 1.8; 2\}$, $\beta \in \{0.2; 0.5; 0.8; 1; 1.2; 1.5; 1.8; 2\}$, dan $\rho \in \{0.1; 0.3; 0.5; 0.7; 0.9; 1\}$. Jika salah satu parameter diganti nilainya, maka nilai parameter lain akan dibuat tetap Pengujian dilakukan dengan membandingkan pengaruh nilai parameter untuk 15 titik dengan 3 salesman. Nilai jarak antartitik pada setiap percobaan akan dibuat sama.

Setelah melakukan eksperimen, maka pengaruh nilai parameter α dan β dapat dilihat pada Gambar 4.3, dan pengaruh nilai parameter ρ dapat dilihat pada Gambar 4.4.



Gambar 4.3 Pengaruh Nilai α dan β terhadap Performa Algoritma



Gambar 4.4 Pengaruh Nilai ρ terhadap Performa Algoritma

Dari hasil pengujian tersebut maka dapat disimpulkan bahwa :

1. Pengaruh nilai α, β dan ρ menghasilkan jarak yang bervariasi dan tidak mengikuti pola tertentu. Hal ini terlihat pada kurva grafik yang selalu dinamis.
2. Dari ketiga parameter tersebut didapatkan kombinasi yang paling bagus untuk memperoleh hasil yang terbaik yaitu $\alpha = 0.2$, $\beta = 1$ dan $\rho = 0,5$. Meskipun demikian, tidak ada pemilihan nilai parameter yang pasti untuk memperoleh total jarak yang paling minimum, karena nilai parameter dan jumlah titik serta jarak yang dihasilkan tidak memiliki pola tertentu. Hal ini juga tergantung pada besarnya masalah yang diselesaikan.

5. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

1. Algoritma *Ant System* (AS) telah berhasil dibangun dan diimplementasikan dengan cukup efektif digunakan dalam penyelesaian MTSP.
2. Hasil percobaan menujukkan bahwa semakin banyak jumlah titik, maka :
   - semakin banyak variasi hasil jarak minimum yang didapatkan
   - semakin tidak stabil jarak minimum yang dihasilkan pada setiap siklus
   - semakin lama terjadinya konvergensi.

3. Hasil percobaan menujukkan bahwa peningkatan jumlah salesman pada jumlah titik yang sama akan memperpanjang jarak yang didapatkan.
4. Hasil percobaan menunjukkan pengaruh parameter sebagai berikut :
   - Pengaruh nilai α, β dan ρ menghasilkan jarak yang bervariasi dan dinamis.
   - Kombinasi yang paling bagus untuk memperoleh hasil yang terbaik yaitu $\alpha = 0.2$, $\beta = 1$ dan $\rho = 0{,}5$. Meskipun demikian, tidak ada pemilihan nilai parameter yang pasti untuk memperoleh total jarak yang paling minimum, karena nilai parameter dan jumlah titik serta jarak yang dihasilkan tidak memiliki pola tertentu.

DAFTAR PUSTAKA

Amin R, dkk. 2006. Travelling Salesman Problem. [Online] Tersedia : *www.informatika.org/~rinaldi/Stmik/Makalah/MakalahStmik30.pdf [*21 Maret 2010*]*

Bektas, Tolga. 2004. "The multiple traveling salesman problem: an overview of formulations and solution procedures". Omega 34 (2006) 209 – 219. [Online] Tersedia : *han-4.tem.nctu.edu.tw/Thesis/... /The multiple traveling salesman problem-an overview of formulations and solution procedures.pdf* [2 Juni 2010]

Deny. 2000. *Struktur Data dan Algoritma*. Fakultas Ilmu Komputer. Universitas Indonesia.

Efendi, Rusdi. 2003. *Penerapan Algoritma Semut Untuk Pemecahan Masalah Spanning Tree pada kasus pemasangan Jaringan Kabel Telepon*. Fakultas Teknologi Industri. Universitas Islam Indonesia : Skripsi Tidak Diterbitkan.

Feryanto, Aris. 2009. Ant Colony System untuk Penyelesaian Masalah Travelling Salesman Problem. [Online] Tersedia : *www.informatika.org/~rinaldi/Stmik/2009.../MakalahIF3051-2009-021.pdf [*21 Maret 2010]

Hasibuan, Zainal. 2007. *Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi.* Fakultas Ilmu Komputer. Universitas Indonesia.

Junjie, P dan Dingwei, W. 2006. An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem. Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06). [Online] Tersedia : *doi.ieeecomputersociety.org/10.1109/ICICIC.2006.40 [*2 Juni 2010]

Kusumadewi, S dan Purnomo, H. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik.* Yogyakarta : Graha Ilmu

Leksono, Agus. 2009. Algoritma Ant Colony Optimization (ACO) Untuk Menyelesaikan Traveling Salesman Problem (TSP). Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Diponegoro. [Online] Tersedia : *eprints.undip.ac.id/7314/1/Tugas_Akhir_(full).pdf* [2 Juni 2010]

Liu, C.L. 1985. *Elements Of The Discrete Mathematics (Second Edition).* Department of Computer Science. University of Illionis at Urbana-Champaign.

Pradhana, Bayu. 2006. Studi Dan Implementasi Persoalan Lintasan Terpendek Suatu Graf Dengan Algoritma Dijkstra Dan Algoritma Bellman-Ford. [Online] Tersedia : *www.informatika.org/~rinaldi/Matdis/20Jurusan 06.../Makalah0607-26.pdf [*21 Februari 2010]

Rachmah, N. 200. Aplikasi Algoritma Dijkstra dalam Pencarian Lintasan Terpendek Graf. [Online] Tersedia : *www.informatika.org/~rinaldi/.../MakalahIF2153-0708-113.pdf [*21 Maret 2010]

Refianti R, dan Mutiara A. 2005. Solusi Optimal Travelling Salesman Problem Dengan Ant Colony System (ACS). Jurusan Teknik Informatika. Universitas Gunadarma. [Online] Tersedia : *paper.abmutiara.info/Ant%20Colony%20System/paper_J_GND_2005_3doc.pdf [*22 Maret 2010]

Saleh,Chairul. 2000. *Algoritma Genetik untuk Pemecahan Traveling Salesman Problem*. Jurnal TEKNOIN. Yogyakarta

Maria A, dkk. 2008. Penyelesaian Masalah Travelling Salesman Problem Menggunakan Ant Colony Optimization (ACO). [Online] Tersedia : *www.informatika.org/~rinaldi/Stmik/Makalah/MakalahStmik29.pdf [*22 Maret 2010]

Mario, T. 2006. *Implementasi Perbandingan Algoritma Ant Colony System Dengan Algoritma Subset Dynamic Programming Pada Kasus Travelling Salesman Problem.* Seminar Nasional Aplikasi Teknologi Informasi 2006 (SNATI 2006).

Munir, Rinaldi. 2006. Diktat Kuliah IF2153 Matematika Diskrit Edisi Keempat. Departemen Teknik Informatika, Institut Teknologi Bandung.

Muttakhiroh, I'ing. 2007. Menentukan Jalur Terpendek Menggunakan Algoritma Semut. Fakultas Teknologi Industri. Universitas Islam Indonesia. [Online] Tersedia : *ienx.wordpress.com/2008/01/03/pencarian-jalur-terpendek-menggunakan-algoritma-semut/* [29 November 2010]

Vitra, I. 2004. *Perbandingan metode-metode dalam algoritma genetika untuk Travelling Salesman Problem.* Proceedings Seminar Nasional Aplikasi Teknologi Informasi 2004