ALGORITMA SEMUT UNTUK PENYELESAIAN *TRAVELLING SALESMAN PROBLEM*

Octariani, Darius Antoni & Emigawaty
Mahasiswa dan Dosen Universitas Bina Darma

Abstract: *The growth of computer science has been applied in problem solving to make easy human work. One of them is in scheduling work and to determine the route of salesman transportation. The activity of salesman transportation is solved heavy problem to determine which route will be passed or before hand passed and has been passed. Often times salesman is consusing to start there transportation. To solve that problem, writer want make ant algorithm applying system to solve the travelling salesman problem with microsoft visual basic 6.0 to solved salesman transportation problem which is developed by waterfall model method.*
Keyword: Ant Algorithm, Stigmery, Travelling Salesman Problem

1. PENDAHULUAN

Perkembangan ilmu komputer dapat diterapkan untuk memecahkan permasalahan yang rumit. Hal ini juga dapat digunakan pada masalah *Traveling Salesman Problem* (TSP). *Traveling Salesman Problem* merupakan sebuah permasalahan yang dapat diterapkan pada berbagai kegiatan seperti *routing* dan penjadwalan produksi. *Traveling Salesman Problem* adalah sebuah masalah optimasi yang terkenal dan telah menjadi standar untuk mencoba algoritma komputasional.

Dalam pemecahan *travelling salesman problem* ada beberapa algoritma yang digunakan, diantaranya algoritma *steepest ascent hill climbing* dalam algoritma ini solusi optimum diibaratkan sebagai puncak gunung tetinggi yang pernah didaki, algoritma genetika merupakan algoritma pencarian yang dikembangkan berdasarkan seleksi alami dan genetika alami, algoritma greddy adalah algoritma yang membuat sebuah *list* yang berisi semua *edge* yang ada pada *graph* dan kemudian mengurutkannya dari yang paling besar ke yang paling kecil.

Permasalahan *Traveling Salesman Problem* pada dasarnya adalah menemukan rute paling pendek pada kota-kota yang harus dikunjungi oleh seorang *traveling salesman*, dari kota awal *salesman* harus mengunjungi setiap kota satu kali dan kembali lagi ke kota awal. Banyaknya kombinasi rute yang mungkin adalah faktorial dari banyak kota.

Dalam pembahasan ini dirumuskan permsalahannya adalah bagaimana menggunakan algoritma semut dengan perancangan suatu simulasi untuk menyelesaikan masalah penugasan *Traveling Salesman Problem* dengan menggunakan bahasa pemrograman *Visual Basic 6.0*.

Yang menjadi ruang lingkup atau pembatasan masalah yaitu :

a. Simulasi dirancang dengan menggunakan bahasa pemrograman Visual Basic 6.0
b. *Travelling salesman problem* yang akan diselesaikan diasumsikan adalah *symmetric travelling salesman problem* dimana jarak kota a ke kota b adalah sama dengan jarak kota b ke kota a dan jumlah kota maksimum 30 kota.

Tujuan dari perancangan simulasi ini adalah untuk memberikan hasil gambaran atau simulasi dalam penerapan algoritma semut untuk pemecahan masalah *travelling salesman problem*.

Manfaat dari perancangan simulasi ini diharapkan dapat memberikan kemudahan bagi para *salesman* untuk melakukan perjalanan yang efektif dan efisien.

2. Tinjauan Pustaka

2.1. Algoritma Semut

Algoritma semut adalah sebuah algoritma yang digunakan untuk menemukan perkiraan solusi untuk masalah yang sulit dipecahkan, dimana algoritma semut merupakan algoritma sistem cerdas yang dikembangkan berdasarkan kehidupan koloni semut yang diterapkan pada ilmu komputer.

Dalam algoritma ini solusi rute tertutup terpendek diadopsi dari rute terpendek yang ditemukan oleh koloni semut di antara sumber - sumber makanan dan sarang. Setiap semut dalam koloni semut secara acak memilih sebuah kota sebagai kota pertama. Kemudian setiap semut akan memilih kota lainnya sebagai kota kedua. Dalam setiap pemilihan kota selanjutnya, kota yang pernah dikunjungi tidak dapat dipilih lagi. Demikian untuk seterusnya sampai semua kota dikunjungi sekali. Setiap semut akan mempunyai panjang rute tertutup masing – masing, yang merupakan jumlah jarak kota pertama ke kota kedua, jarak kota kedua ke kota ketiga dan seterusnya sampai kota terakhir ditambah dengan jarak kota terakhir ke kota pertama. Solusi yang ditemukan adalah panjang rute tertutup paling pendek di antara panjang rute yang ditempuh semua semut.
http://dsp.jpl.nasa.gov/ members/payman/swarm/dorigo96-itsmc.pdf

2.2. Stigmergy

Operasi dalam koloni semut memiliki beberapa tugas yang berbeda. Tugas dalam koloni semut meliputi :

1. Reproduksi : merupakan tugas dari ratu semut.
2. Penjagaan : merupakan tugas dari semut prajurit.

3. Pencari makanan : merupakan tugas dari semut pekerja.
4. Pelindung : merupakan tugas dari semut pekerja.
5. Pembuatan sarang : merupakan tugas dari semut pekerja.
6. Perawatan sarang : merupakan tugas dari semut pekerja.

Jika proses distribusi dan eksekusi terjadi secara magis, tanpa ada suatu pusat yang berfungsi sebagai pengatur seluruh perintah yang ada. Tugas distribusi dan eksekusi tersebut dibedakan berdasarkan perbedaan anatomi dan *stigmergy*. Perbedaan anatomi seperti ukuran dan struktur, sebagai contoh yang membedakan antara semut prajurit dengan semut pekerja pencari makanan.

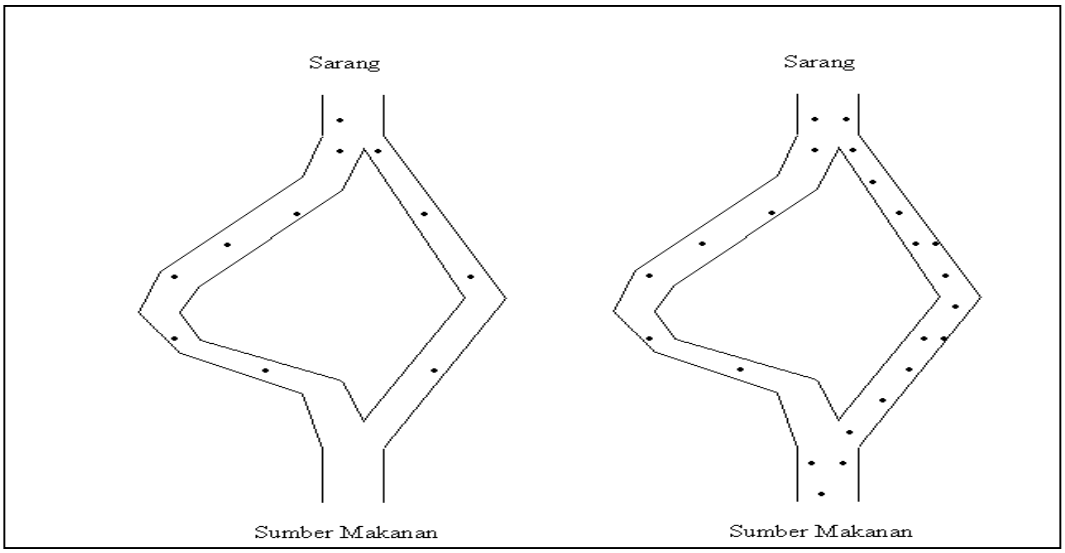
Yang membagi – bagi kelakuan yang ada dalam koloni inilah yang disebut dengan *stigmergy*. *Stigmergy* dikelompokkan oleh Dorigo 1999 sebagai berikut :

1. Kekurangan dari koordinat pusat.
2. Komunikasi antara individu dalam koloni berdasarkan pada perubahan dari lingkungan sekitarnya.
3. Pengaruh arus balik positif, yang dimana merupakan penguatan dari tindakan yang telah diperbuat.

Bentuk algoritma dari koloni semut berdasarkan konsep *stigmergy* tiruan, didefinisikan oleh Dorigo dan Di Caro sebagai "komunikasi tak langsung yang menengahi perubahan numerik yang berhubungan dengan lingkungan dimana hanya ditempat itu dapat diperoleh dengan saling berkomunikasi" [Dorigo dan Di Caro 1999]. Inti dari bentuk koloni semut, atau dari aspek operasi pada koloni semut yaitu untuk mencari bentuk matematis yang mana secara akurat mengambarkan karakteristik *stigmergetic* pada individu semut yang berhubungan.

2.3. *Pheromone*

Semut memiliki kemampuan untuk selalu menemukan jalan yang paling pendek di antara sarang dan sumber makanan. Beberapa percobaan telah dilakukan untuk mempelajari sifat atau kemampuan tersebut. Percobaaan meliputi pembuatan jalan yang memiliki panjang yang berbeda di antara sarang dan sumber makanan, dan memperhatikan jumlah semut yang melewati setiap jalannya. Pemilihan jalan dilakukan secara acak, dan dari hasil yang didapat dapat dilihat bahwa banyak semut akan lebih memilih jalan yang lebih pendek.



Sumber : http://dsp.jpl.nasa.gov/ members/payman/swarm/dorigo96-itsmc.pdf

Gambar 1 *Pheromone* jejak semut

Sifat ini hanya bisa dijelaskan bila setiap semut meletakkan *pheromone*. Pada waktu mencari makanan, dan perjalanan kembali dari sumber makanan menuju ke sarang, setiap semut meletakkan *pheromone*. Untuk memilih jalan yang akan dilewati, semut akan memilih jalan yang memiliki konsentrasi *pheromone* yang besar. Jalan yang lebih pendek memiliki *pheromone* yang kuat, sejak semut kembali lebih cepat dari sumber makanan melewati jalan itu (semut meletakkan banyak *pheromone* pada jalan kembali ke sarang) dari pada jalan yang lebih panjang dan lebih lama. Dan lama kelamaan konsentrasi *pheromone* pada jalan yang lebih panjang tersebut mengalami penurunan lebih cepat dibandingkan dengan jalan yang lebih pendek.

2.4. Langkah – langkah membangun algoritma semut

Langkah – langkah menyelesaikan *Traveling Salesman Problem* dengan algoritma semut adalah sebagai berikut: 1) Tentukan jumlah kota (M) dan semut (N) yang akan melakukan simulasi perjalanan TSP, 2) Masukkan semua jarak dari satu kota ke kota lainnya yang dilambangkan dengan d_{ij}, 3) Lakukan perhitungan visibilitas antara kota i dan j (η_{ij}), 4) *Generate* semua nilai $\tau_{ij}(0)$ yang merupakan intensitas jejak semut dari kota ke kota j pada keadaan awal secara sembarang. Jadi nilai $\tau_{ij}(0)$ ada sebanyak C_2^n, *5)* Tentukan probabilitas setiap lintasan dari kota i ke kota j untuk setiap semut dengan rumus berikut dimana k adalah semut ke k, 6) Penyusunan rute kunjungan setiap semut ke setiap kota berdasarkan Φ_{ij}, 7) Tentukan jarak yang telah ditempuh setiap semut. Jarak yang telah

ditempuh semut ke k pada waktu ke t dilambangkan dengan $L_k(t)$. $L_k(t)$ dihitung dengan menjumlahkan seluruh jarak yang telah ditempuh semut, 8) Tentukan nilai Q yang merupakan nilai dari $L_k(t)$ terkeciL, 9) Hitung perubahan harga intensitas jejak semut antar kota, 10) Hitung harga intensitas jejak semut antar kota untuk siklus berikutnya, 11) Ulangi langkah 5 sampai langkah 10 untuk siklus berikutnya.

2.5. Algoritma Menentukan Rute Semut

Langkah berikutnya yaitu menentukan rute semut berdasarkan probabilitas kunjungan kota untuk setiap semut. Langkah - langkah menentukan rute semut berdasarkan Φ_{ij} yaitu: 1) Tentukan jumlah semut (N), 2) Tentukan dan pilih kota yang akan menjadi kota pertama sesuai dengan rute perjalanan yang telah ditetapkan. Misalkan rute perjalanan yang telah ditetapkan yaitu 1 2 3 4 5 1, maka kota pertama pastilah kota ke 1. Kota awal = I, 3) Tentukan kota ke I + 1 yang dilalui dan jumlah semutnya dengan cara sebagai berikut: a) Lakukan perhitungan total Φ_{ij} untuk setiap j yang mungkin dilalui dilambangkan dengan $T\Phi_{ij}$, b) Pilih kota berikutnya dari kota ke I nilai Φ_{ij} yang terbesar berikutnya yang rutenya belum pernah dilalui, c) Tentukan jumlah semut yang melalui kota I - (I + 1) sebanyak n1 = (Φ_{ij} * n) / T, d) Ulangi langkah a dan b untuk n2, n3 dan seterusnya hingga n1 + n2 + n3 +... = n. 4) Lakukan langkah dua kembali untuk menentukan jumlah semut ke kota berikutnya pada setiap jalurnya hingga kembali ke kota awal (I = 1). I berubah sesuai dengan langkah sebelumnya, 5) Lakukan pengecekan rute perjalanan kalau ada yang sama maka lakukan perbaikan rute menurut langkah 6 dan seterusnya jika tidak maka selesai, 6) Lakukan pertukaran kota sesuai dengan

3. METODOLOGI PENELITIAN

3.1 Objek Peneliitian

Dalam penelitian ini, yang menjadi objek penelitian adalah *Traveling Salesman Problem (TSP)* yang merupakan salah satu permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan, salah satunya yaitu pada *routing, Traveling Salesman Problem* dapat diselesaikan dengan menggunakan algoritma semut, dengan perancangan suatu simulasi. Simulasi dalam penelitian ini dirancang dengan menggunakan bahasa pemrograman *Visual Basic 6.0.*

3.2 Metode Pengumpulan Data

Metode yang digunakan dalam proses pengumpulan data adalah metode studi pustaka yaitu dengan cara mencari bahan yang mendukung

dalam pendefinisian permasalahan melalui buku-buku, internet, yang erat kaitannya dengan objek penelitian.

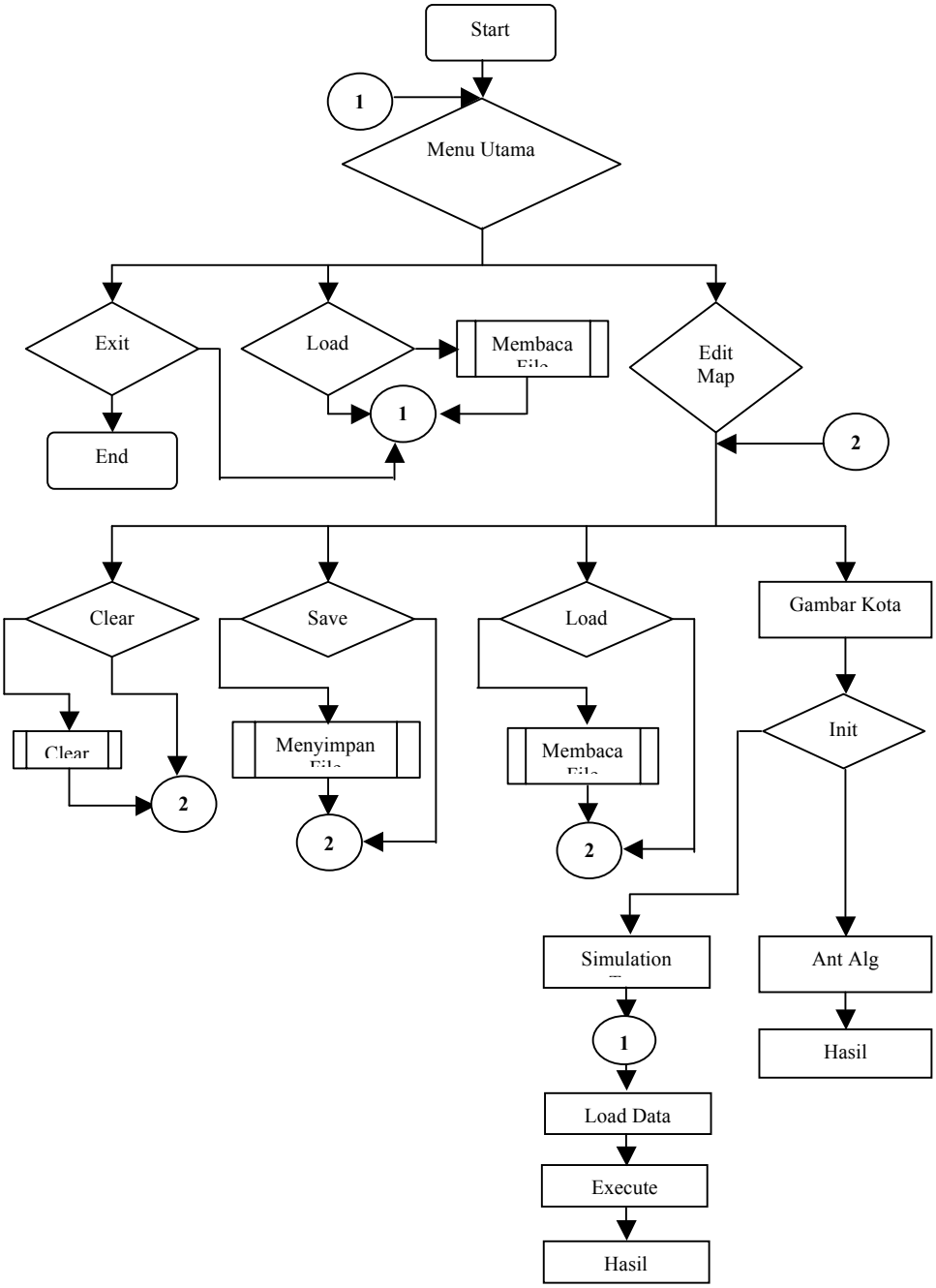
3.3 Tahapan Pengembangan Sistem

Metodologi penelitian yang digunakan dalam penelitian ini adalah metodologi *linier sequential* atau sering disebut *waterfall model.* Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai sistem itu diterapkan, dioperasikan dan dipelihara. Menurut Pressman (2002:38) ada 6 fase proyek pengembangan sistem melingkupi sebagai berikut: 1) Rekayasa sistem dan analisis (*system engineering and analysis*), 2) Analisis kebutuhan perangkat lunak (*software requirements analisys*), 4) Perancangan (*design*), 5) Pembuatan kode (*coding*), 6) Penterjemahan perancangan ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman, 7) Pengujian (*testing*), 8) Pemeliharaan (*maintenance*)

3.1 Perancangan

3.1.1 Rancangan Alur Kerja Simulasi

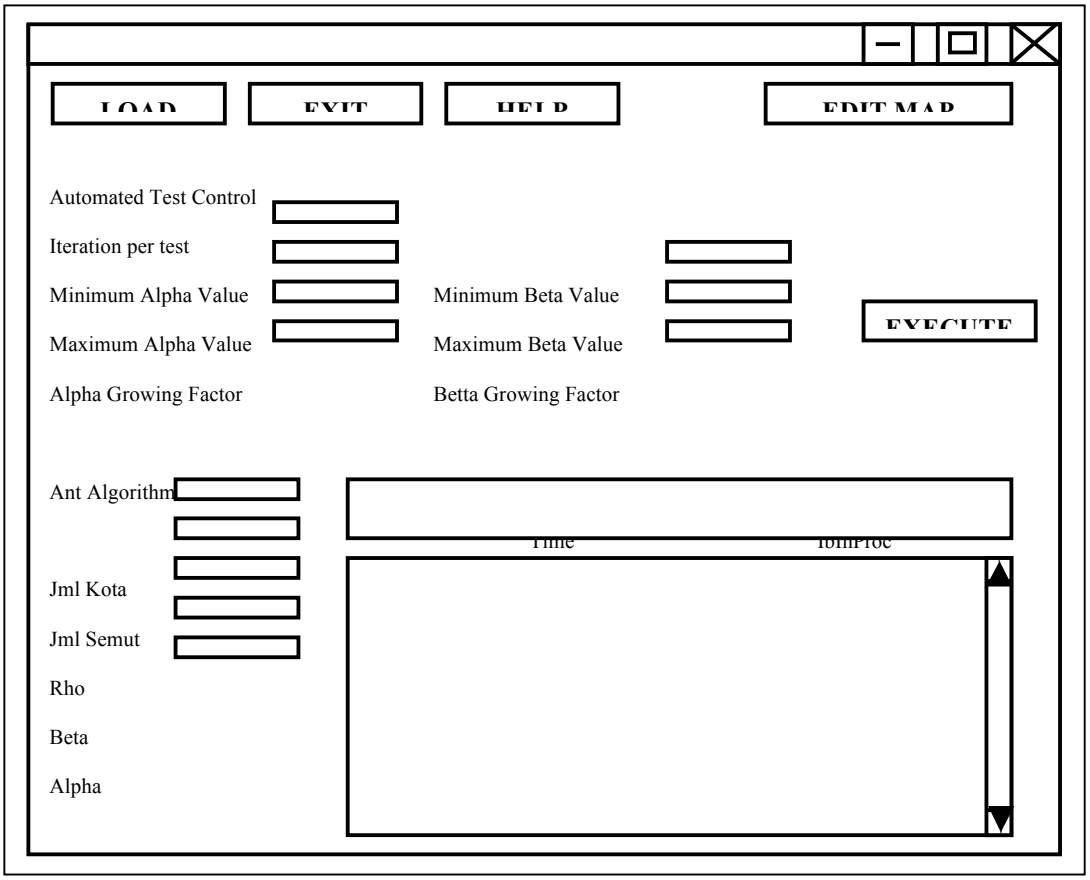
Alur kerja dari simulasi yang dirancang adalah seperti yang ditunjukkan pada gambar flowchart berikut:



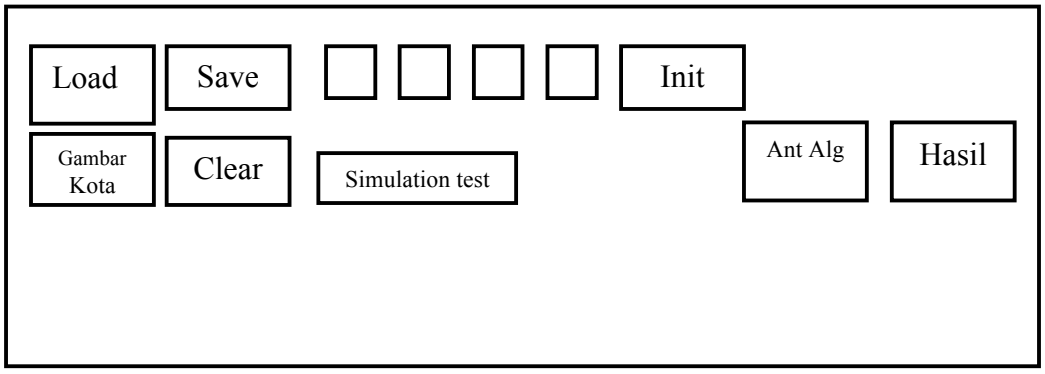
Gambar 2. Gambar Flowchart Alur Kerja Simulasi

3.1. Rancangan Menu Utama

Pada menu utama terdiri dari 4 tombol yang meliputi : tombol *load*, tombol *save*, tombol *edit map*, dan tombol *execute*. Berikut rancangan dan penjelasan dari tombol-tombol yang terdapat pada *form* ini.



Gambar 3.a. Objek-objek pada menu utama



Gambar 3.b. Objek-objek yang *Form Edit Map.*

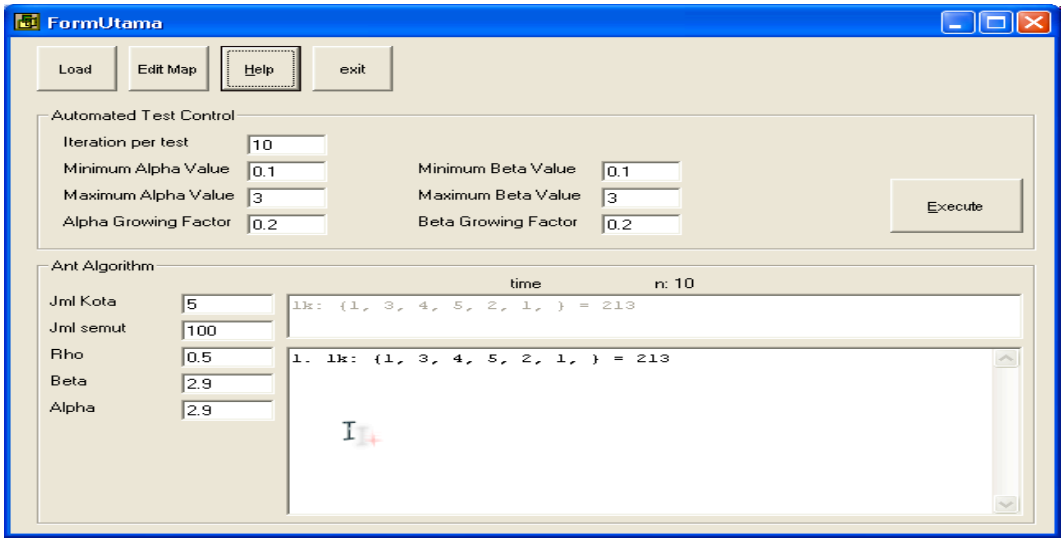
Objek yang terdapat pada Form Edit Map terdiri dari: *Obyek cmdEditing, Obyek cmdInit. Obyek cmdAnt, Obyek cmdClear, Obyek cmdAuto, Obyek txtRho, Obyek txtAlpha, Obyek txtBeta, Obyek txtJmlSemut, Obyek txtBest, Obyek txtStat, Obyek kota, Obyek lineSelected, Obyek Line1, Obyek CommonDialog1*.

Pada form diatas terdapat beberapa tombol meliputi : **1) Load:** Tombol load merupakan obyek commandButton yang di gunakan untuk memilih data yang akan di buka, 2)**Save:** Tombol save merupakan obyek commandButton yang di gunakan untuk menyimpan data apabila telah menggambar kota pada *form* editmap**, 3) Clear:** Tombol ini berguna untuk melakukan penghapusan atau melakukan perhitungan kembali, ketika tombol ini ditekan maka akan melakukan *unload form* utama kemudian memanggil kembali, 4) **init:** Tujuan dari di buatnya tombol init ini adalah untuk membantu dalam simulasi, 5)**Ant Alg:** Tombol ini di perlukan untuk memulai simulasi, ketika tombol ini di klik maka akan memulai perhitungan untuk mencari solusi masalah penugasan dengan algoritma semut, 6) **Gambar Kota:** Tombol gambar kota ini di butuhkan untuk menggambar kota, ketika tombol di klik maka akan memulai untuk menempatkan posisi kota 1, kota 2 dan seterusnya sebanyak yang diinginkan, 7) **Simulation Test:** Tujuan dari di buatnya tombol simulation test ini adalah untuk melakukan percobaan simulasi dan juga merupakan link untuk kembali ke *form* utama.

4. PEMBAHASAN

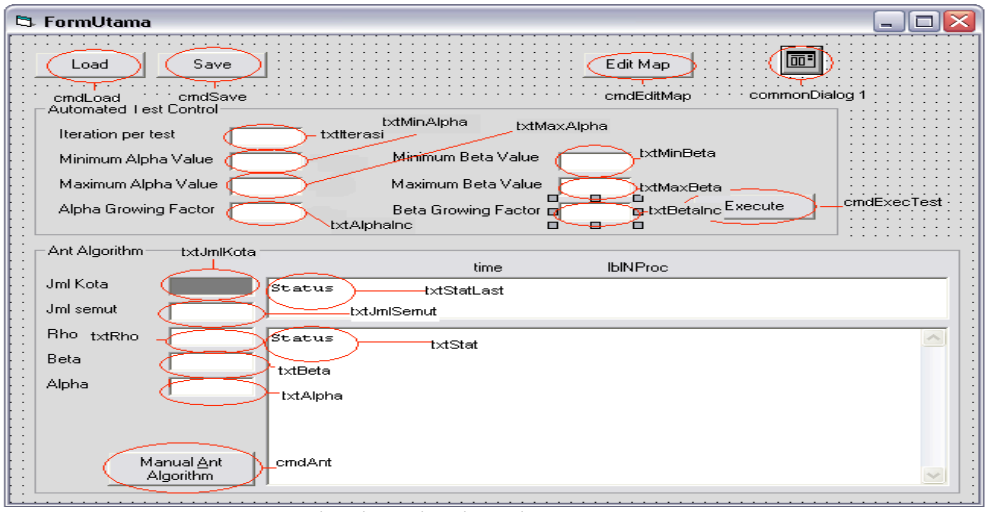
4.1. Menu Utama

Pertama kali simulasi dijalankan, maka akan memanggil *form* ini. Sesuai dengan rancangan tampilan simulasi, maka *form* ini seperti yang ditunjukkan oleh Gambar 4.



Gambar 4 Tampilan menu utama

Gambar 4 juga menunjukkan obyek – obyek yang terdapat pada *form* ini. Kemudian pada Gambar 5 akan menunjukkan program simulasi yang telah selesai melakukan perhitungan.

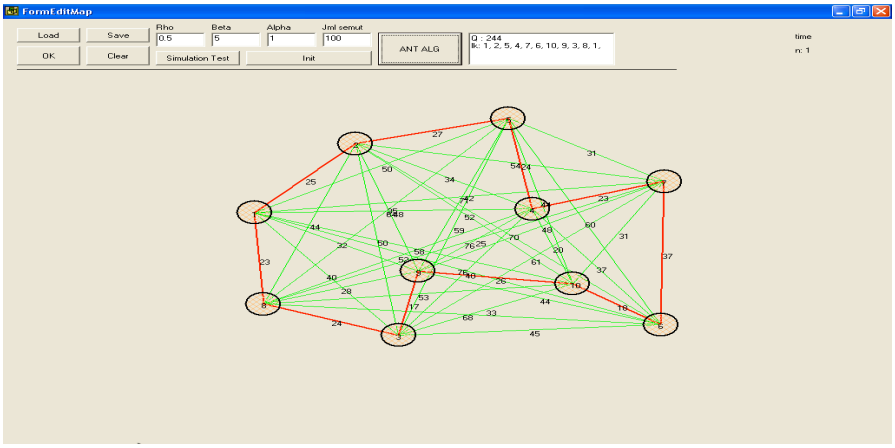


Gambar 5 Obyek – obyek pada menu utama

4.2. Form EditMap

Form ini dibangun untuk mendukung *form* utama, *form* ini akan dipanggil jika obyek cmdEditMap pada *form* utama diklik. Secara sederhana kerja dari *form* ini yaitu, ketika dipanggil maka akan dapat digunakan untuk menggambar peta kota. Jika ingin mengganti gambar peta kota maka dapat langsung diganti dari *form* ini. Setelah melakukan

pembuatan gambar kota untuk memulai simulasi percobaan dapat dipilih obyek cmdSim pada *form* EditMap.



Gambar 6 Tampilan *EditMap*
k pada Tampilan EditMap

4.3 Percobaan Simulasi

Percobaan ini dilakukan dengan tujuan untuk menguji program simulasi. yang telah selesai dibangun. Percobaan ini akan menggunakan nilai input yang cukup besar untuk mengetahui bahwa program simulasi ini dapat dilakukan untuk nilai input yang besar.

Dengan percobaan ini, diharapkan akan memperlihatkan pengaruh penggunaan kombinasi parameter α (koefisien kendali intensitas jejak semut) dan β (koefisien kendali visibilitas) yang berbeda pada kelakuan siklus semut.

Untuk percobaan simulasi ini akan dilakukan dengan langkah – langkah sebagai berikut :1) Jumlah kota (M) yang akan digunakan adalah sebanyak 10 kota, 20 kota dan 30 kota, 2) Koefisien kendali intensitas jejak semut adalah α, α = {0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9}. Koefisien kendali visibilitas (informasi lokal) adalah β, β = {0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9}. Tetapan penguapan jejak semut adalah ρ, ρ = {0.5}. Jumlah semut adalah N, N = 4 semut.

Adapun langkah-langkah yang dilakukan berdasarkan ketentuan tersebut, sebagai berikut: 1) Untuk percobaan mengetahui pengaruh kombinasi α dan β pada siklus semut yaitu melakukan perhitungan α = {0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9} dan β = {0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9}. 2) Pertama – tama dilakukan percobaan dengan menggunakan 10 kota untuk perhitungan α = 0.1, β = {0.1,

0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9}, kemudian percobaan dilanjutkan dengan melakukan perhitungan yang sama untuk α = {0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9} dengan menggunakan parameter β = {0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7, 2.9}. (3) Dilakukan percobaan yang sama untuk 20 kota dan 30 kota, dimulai dari langkah 1 hingga langkah 3.

Untuk setiap percobaan, dicatat hasil percobaannya yaitu berupa jumlah iterasi atau siklus (n), lama waktu (ms) yang dibutuhkan untuk melakukan siklus tersebut, dan jarak yang diperlukan untuk mencapai kondisi rute terbaik.

5. SIMPULAN

1. Dari semua hasil percobaan yang telah dilakukan, dapat diketahui bahwa rute pencarian yang dilakukan dengan algoritma semut jauh lebih kecil daripada semua kemungkinan kombinasi yang harus diperiksa secara manual (n! kombinasi). Ini berarti bahwa algoritma semut dapat mengurangi beban komputasi yang sangat besar.
2. Pemanfaatan teknologi komputer, seperti dengan adanya program untuk melakukan pencarian rute terpendek khususnya algoritma semut dapat menghemat waktu perjalanan dengan jarak lintasan yang lebih pendek.

DAFTAR RUJUKAN

Dorigo,Marco.; Maniezzo, Vittorio.; Colorni, Alberto., *The Ant System Optimization by a colony of cooperating agents*. http://dsp.jpl.nasa.gov/ members/payman/swarm/dorigo96-itsmc.pdf 4 Mei 2006.

Http :// cyber.im.fsu.edu.tw/course/13141/ant %20 system.PPT 25 juli 2006

Http :// www.derrick babb.com/academics/research/aco/aco.PPT 25 juli 2006.

Kusumo, Ario Suryo. 2000.*Buku Latihan Microsoft Visual Basic 6.0.* Jakarta : PT. Elex Media Komputindo.

Munir, Rinaldi, 1999. *Buku Teks Ilmu Komputer Algoritma dan Pemrograman Buku Pertama.* Penerbit Informatika, Bandung.

Presman, Roger S, 2002, *Rekayasa Perangkat Lunak,* Penerbit Andi Offset, Yogyakarta.

Reinelt, G., *The Traveling Salesman Problem* : *Computational Solutions for TSP Applications*, 1994.