

# Penerapan Algoritma Semut untuk Pencarian Jalur Terpendek

Fabrian Oktavino Hardjasutanto (13510053)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia

[Fabrian.oktavino@students.itb.ac.id](mailto:Fabrian.oktavino@students.itb.ac.id)

## Abstract

Penggunaan jalur terpendek pada suatu perjalanan merupakan hal yang penting dalam dunia pengiriman barang karena dengan menggunakan jalur terpendek, kita dapat menghemat waktu, tenaga serta biaya. Salah satu cara untuk menemukan jalur terpendek ialah menggunakan teori algoritma semut.

**Kata Kunci**—Algoritma semut, jalur terpendek

## I. PENDAHULUAN

Dalam hidup ini, kita sering melakukan perjalanan dari satu tempat ke tempat yang lain. Tentu saja perjalanan yang kita lakukan tidak tanpa pertimbangan terlebih dahulu. Pertimbangan yang dilakukan tentu berdasarkan beberapa faktor seperti : biaya, waktu, dan efisiensi. Sehingga dalam perjalanan kita perlu menentukan jalur terpendek dari tempat kita menuju ke tempat tujuan.

Secara umum, pencarian jalur terpendek dapat terbagi menjadi dua metode, metode konvensional dan metode heuristik. Metode konvensional cenderung lebih mudah dipahami daripada metode heuristik, yaitu hanya membandingkan jarak masing-masing node dan kemudian mencari jarak yang terpendek. Tetapi, bila dibandingkan hasilnya, hasil yang diperoleh dari metode heuristik lebih variatif, hasil yang didapatkan lebih akurat, tingkat kesalahan yang dihasilkan pada perhitungan lebih kecil, dan waktu perhitungan yang diperlukan lebih singkat. Metode heuristik terdiri dari beberapa macam algoritma, salah satunya adalah algoritma semut (*Ant Colony*, Antco). Antco diambil dari perilaku koloni semut dalam pencarian jalur terpendek antara sarang dan sumber makanan.

Algoritma semut diperkenalkan oleh Moyson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo. Algoritma semut merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik melalui grafik.

Pada dunia nyata, semut berkeliling secara acak, dan ketika menemukan makanan mereka kembali ke koloninya sambil memberikan tanda dengan jejak feromon. Jika semut-semut lain menemukan jalur tersebut, mereka tidak akan berpersion dengan acak lagi, melainkan akan mengikuti jejak tersebut, kembali dan menguatkannya jika pada akhirnya merekapun

menemukan makanan.

Seiring waktu, bagaimanapun juga jejak feromon akan menguap dan akan mengurangi kekuatan daya tariknya. Lebih lama seekor semut pulang pergi melalui jalur tersebut, lebih lama jugalah feromon menguap. Sebagai perbandingan, sebuah jalur yang pendek akan berbaris lebih cepat, dan dengan demikian kerapatan feromon akan tetap tinggi karena terletak pada jalur secepat penguapannya. Penguapan feromon juga mempunyai keuntungan untuk mencegah konvergensi pada penyelesaian optimal secara lokal. Jika tidak ada penguapan sama sekali, jalur yang dipilih semut pertama akan cenderung menarik secara berlebihan terhadap semut-semut yang mengikutinya. Pada kasus yang demikian, eksplorasi ruang penyelesaian akan terbatas.

Oleh karena itu, ketika seekor semut menemukan jalur yang bagus (jalur yang pendek) dari koloni ke sumber makanan, semut lainnya akan mengikuti jalur tersebut, dan akhirnya semua semut akan mengikuti sebuah jalur tunggal. Ide algoritma koloni semut adalah untuk meniru perilaku ini melalui 'semut tiruan' berjalan seputar grafik yang menunjukkan masalah yang harus diselesaikan. Mengingat prinsip algoritma yang didasarkan pada perilaku koloni semut dalam menemukan jarak perjalanan paling pendek tersebut, Antco sangat tepat digunakan untuk diterapkan dalam penyelesaian masalah menentukan jalur terpendek.

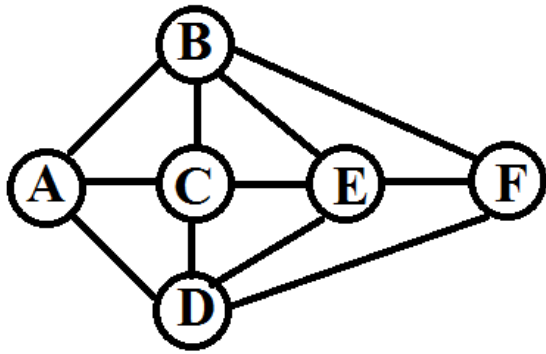
## II. GRAF

Graf adalah kumpulan simpul (*nodes*) yang terhubung satu sama lain melalui sisi (*edges*). Suatu graf  $G$  terdiri dari dua himpunan  $V$  dan  $E$ . Verteks (simpul) :  $V =$  himpunan simpul yang terbatas dan tidak kosong. Edge (sisi) :  $E =$  himpunan sisi yang menghubungkan sepasang simpul. Simpul – simpul pada graf dapat merupakan obyek sembarang seperti : kota, atom – atom suatu zat, jenis buah, dan sebagainya. Sisi dapat menunjukkan relasi sembarang seperti rute penerbangan, jalan raya, dan lain-lain. Notasi graf  $G(V,E)$  menyatakan bahwa graf  $G$  memiliki  $V$  simpul dan  $E$  sisi.

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis :

1. Graf tak-berarah (*undirected graph*)

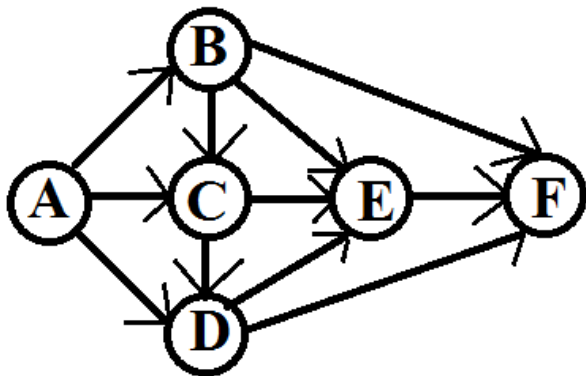
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.



Gambar 2.1 Graf tak-berarah

2. Graf berarah (*directed graph* atau *digraph*)

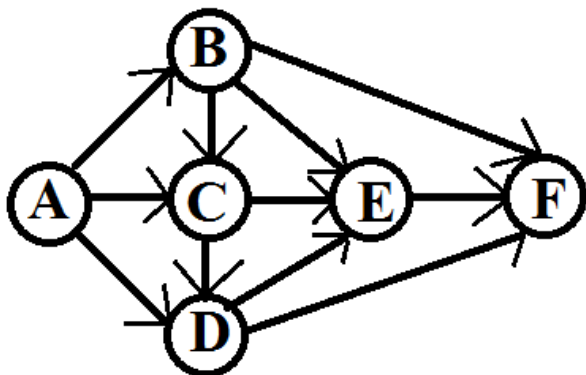
Graf yang setiap sisinya diberikan orientasi arah disebut graf berarah.



Gambar 2.2 Graf berarah

III. JALUR TERPENDEK

Jalur terpendek adalah suatu jaringan pengarah perjalanan dimana seorang pengarah ingin mendapatkan jalur terpendek antara dua kota berdasarkan jalur - jalur alternatif yang tersedia, dimana titik tujuan hanya satu. Gambar 3.1 menunjukkan graf ABCDEF yang berarah.



Gambar 3.1 Graf ABCDEF

Pada gambar diatas, dimisalkan kita dari kota A ingin menuju ke kota F, terdapat beberapa jalur yang tersedia :

- A → B → F
- A → B → E → F
- A → B → C → E → F
- A → B → C → D → F
- A → B → C → D → E → F

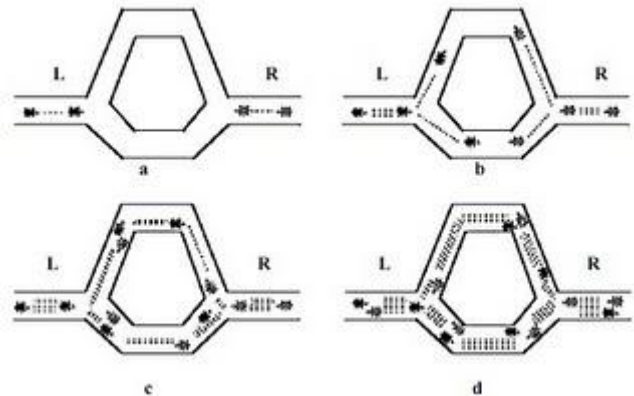
- A → C → E → F
- A → C → D → F
- A → C → D → E → F
- A → D → F
- A → D → E → F

Berdasarkan data diatas, dapat dihitung jalur terpendek dengan mencari jarak antara jalur - jalur tersebut. Apabila jarak antar jalur belum diketahui, jarak dapat dihitung berdasarkan jarak antara koordinat kota - kota tersebut, lalu dihitung total jaraknya.

IV. ALGORITMA SEMUT

Algoritma Semut diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996). Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut.

Gambar 4.1 menunjukkan perjalanan semut dalam menemukan jalur terpendek dari sarang ke sumber makanan.



Gambar 4.1 Perjalanan semut menemukan sumber makanan.

Gambar 4.1.a di atas menunjukkan ada dua kelompok semut yang akan melakukan perjalanan. Satu kelompok bernama L yaitu kepompok yang berangkat dari arah kiri yang merupakan sarang semut dan kelompok lain yang bernama kelompok R yang berangkat dari kanan yang merupakan sumber makanan. Kedua kelompok semut dari titik berangkat sedang dalam posisi pengambilan keputusan jalan sebelah mana yang akan diambil. Kelompok semut L membagi dua kelompok lagi. Sebagian melalui jalan atas dan sebagian melalui jalan bawah. Hal ini juga berlaku pada kelompok semut R. Gambar b dan gambar c menunjukkan bahwa kelompok

semut berjalan pada kecepatan yang sama dengan meninggalkan feromon atau jejak kaki di jalan yang telah dilalui. Feromon yang ditinggalkan oleh kumpulan semut yang melalui jalan atas telah mengalami banyak penguapan karena semut yang melalui jalan atas berjumlah lebih sedikit dari pada jalan yang di bawah. Hal ini dikarenakan jarak yang ditempuh lebih panjang daripada jalan bawah. Sedangkan feromon yang berada di jalan bawah, penguapannya cenderung lebih lama. Karena semut yang melalui jalan bawah lebih banyak daripada semut yang melalui jalan atas. Gambar d menunjukkan bahwa semut-semut yang lain pada akhirnya memutuskan untuk melewati jalan bawah karena feromon yang ditinggalkan masih banyak. Sedangkan feromon pada jalan atas sudah banyak menguap sehingga semut-semut tidak memilih jalan atas tersebut. Semakin banyak semut yang melalui jalan bawah maka semakin banyak semut yang mengikutinya.

Demikian juga dengan jalan atas, semakin sedikit semut yang melalui jalan atas, maka feromon yang ditinggalkan semakin berkurang bahkan hilang. Dari sinilah kemudian terpilihlah jalur terpendek antara sarang dan sumber makanan.

#### V. ALGORITMA SEMUT DALAM PENCARIAN JALUR TERPENDEK

Dalam algoritma semut, diperlukan beberapa variabel dan langkah – langkah untuk menentukan jalur terpendek, yaitu :

Langkah 1 :

a. Inisialisasi nilai parameter – parameter logika yang terdiri atas :

1. Intensitas jejak semut antar kota dan perubahannya ( $\tau_{ij}$ )
2. Banyak tempat ( $n$ ) termasuk koordinat ( $x,y$ ) atau jarak antar kota ( $d_{ij}$ )
3. Tempat berangkat dan tempat tujuan
4. Tetapan siklus-semut ( $Q$ )
5. Tetapan pengendali intensitas jejak semut ( $\alpha$ ), nilai  $\alpha \geq 0$
6. Tetapan pengendali visibilitas ( $\beta$ ), nilai  $\beta \geq 0$
7. Visibilitas antar kota =  $1/d_{ij}$  ( $\eta_{ij}$ )
8. Banyak semut ( $m$ )
9. Tetapan penguapan jejak semut ( $\rho$ ), nilai  $\rho$  harus  $> 0$  dan  $< 1$  untuk mencegah jejak feromon yang tak terhingga.
10. Jumlah siklus maksimum ( $NC_{max}$ ) bersifat tetap selama algoritma dijalankan ,sedangkan  $\tau_{ij}$  akan selalu diperbaharui harganya pada setiap siklus algoritma mulai dari siklus pertama ( $NC=1$ ) sampai tercapai jumlah siklus maksimum ( $NC=NC_{max}$ ) atau sampai terjadi konvergensi.

b. Inisialisasi kota pertama setiap semut. Setelah inisialisasi  $\tau_{ij}$  dilakukan, kemudian  $m$  semut ditempatkan pada kota pertama tentu secara acak.

Langkah 2 :

Pengisian kota pertama ke dalam tabu list. Hasil inisialisasi kota pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen pertama tabu list. Hasil dari langkah ke 2 ini adalah terisinya elemen pertama tabu list setiap semut dengan indeks kota tertentu. Yang berarti bahwa setiap  $tabu_k(1)$  bisa berisi indeks kota antara 1 sampai  $n$  sebagaimana hasil inisialisasi pada langkah 1.

Langkah 3 :

Penyusunan rute kunjungan setiap semut ke setiap kota. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap kota, akan mulai melakukan perjalanan dari kota pertama masing – masing sebagai kota asal dan salah satu kota lainnya sebagai kota tujuan. Kemudian dari kota kedua masing – masing, koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari kota – kota yang tidak terdapat pada  $tabu_k$  sebagai kota tujuan selanjutnya. Begitu selanjutnya terus menerus sampai semua kota satu persatu dikunjungi atau telah menempati  $tabu_k$ . Jika  $s$  menyatakan indeks urutan kunjungan, kota asal dinyatakan sebagai  $tabu_k(s)$  dan kota – kota lainnya dinyatakan sebagai  $\{N - tabu_k\}$ , maka untuk menentukan kota tujuan digunakan persamaan probabilitas kota untuk dikunjungi sebagai berikut :

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in \{N - tabu_k\}} [\tau_{ik}]^\alpha \cdot [\eta_{ik}]^\beta} \text{ untuk } j \in \{N - tabu_k\} \dots\dots\dots (1)$$

dan

$$p_{ij}^k = 0, \text{ untuk } j \text{ lainnya} \dots\dots\dots (2)$$

dengan  $i$  sebagai indeks kota asal dan  $j$  sebagai indeks kota tujuan.

Langkah 4 :

a. Perhitungan panjang rute tiap semut.

Perhitungan panjang rute tertutup (*length closed tour*) atau  $L_k$  setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini didasarkan atas  $tabu_k$  masing – masing sesuai persamaan berikut :

$$L_k = d_{tabu_k(n), tabu_k(1)} + \sum_{s=1}^{n-1} d_{tabu_k(s), tabu_k(s+1)} \dots\dots\dots (3)$$

dengan  $d_{ij}$  adalah jarak antar kota  $i$  dan  $j$  yang dihitung berdasarkan persamaan :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \dots\dots\dots (4)$$

b. Pencarian rute terpendek.

Setelah  $L_k$  setiap semut dihitung, akan didapat harga minimal panjang rute tertutup setiap siklus atau  $L_{minNC}$  dan harga minimal panjang rute tertutup secara keseluruhan adalah  $L_{min}$ .

c. Perhitungan perubahan harga intensitas jejak kaki

semut antar kota.

Koloni semut akan meninggalkan jejak – jejak kaki pada lintasan antar kota yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar kota. Persamaan perubahan ini adalah :

$$\Delta\tau_{ij}^k = \sum_{k=1}^m \Delta\tau_{ij}^k \dots\dots\dots(5)$$

$\Delta\tau_{ij}^k$  adalah perubahan harga intensitas jejak kaki semut antar kota setiap semut yang dihitung berdasarkan persamaan :

$$\Delta\tau_{ij}^k = \frac{Q}{L_k}, \text{ untuk } (i,j) \in \text{kota asal dan kota tujuan dalam } \textit{tabu}_k \dots\dots\dots(6)$$

$$\Delta\tau_{ij}^k = 0, \text{ untuk } (i,j) \text{ lainnya } \dots\dots\dots(7)$$

Langkah 5 :

- a. Perhitungan harga intensitas jejak kaki semut antar kota untuk siklus selanjutnya.

Harga intensitas jejak kaki semut antar kota pada semua lintasan antar kota ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewatinya. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antar kota untuk siklus selanjutnya dihitung dengan persamaan :

- b. Atur ulang harga perubahan intensitas jejak kaki semut antar kota.

Untuk siklus selanjutnya, perubahan harga intensitas jejak semut antar kota perlu diatur kembali agar memiliki nilai sama dengan nol.

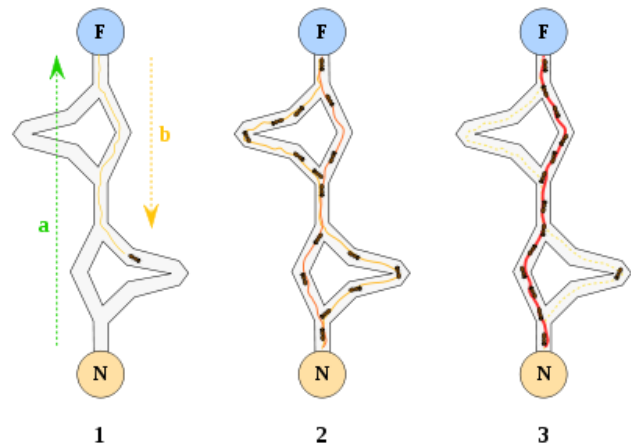
Langkah 6 :

Pengosongan tabu list, dan ulangi langkah 2 jika diperlukan. Tabu list perlu dikosongkan untuk diisi lagi dengan urutan kota yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi. Algoritma diulang lagi dari langkah 2 dengan harga parameter intensitas jejak kaki semut antar kota yang sudah diperbaharui.

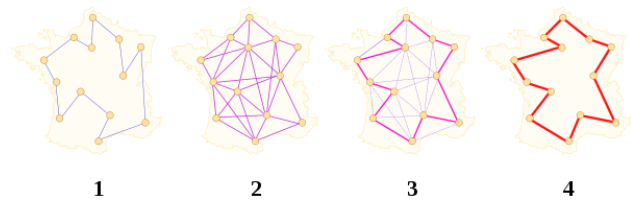
## V. KESIMPULAN

Dari hasil ulasan di atas dapat dilihat bahwa penggunaan algoritma semut dalam pencarian jalur terpendek memberikan hasil yang akurat. Namun dari ulasan diatas juga dapat dilihat bahwa penghitungan menggunakan algoritma semut tidak mudah dan memerlukan waktu yang lama apabila kita menghitung secara manual. Namun hal ini dapat diatasi dengan membuat program untuk penghitungan tersebut pada komputer, sehingga pengguna hanya perlu memasukkan variabel yang dibutuhkan, dan komputer akan mengolah data yang didapatkan untuk mendapatkan hasilnya. Tentu waktu yang dibutuhkan menjadi lebih sedikit apabila dibandingkan dengan penghitungan manual.

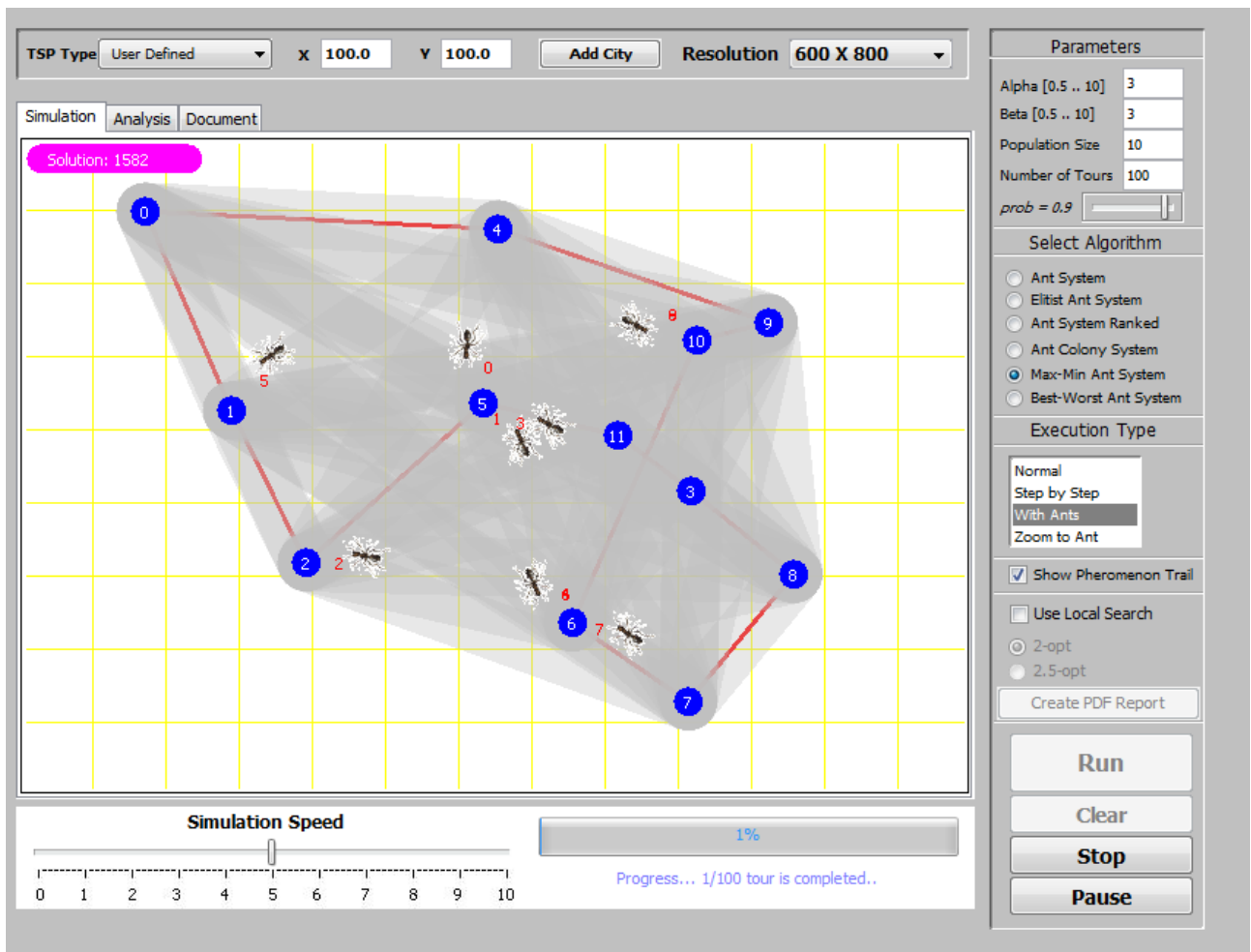
## VI. APPENDIX



Gambar 6.1 Ilustrasi jejak semut



Gambar 6.2 Ilustrasi algoritma semut



Gambar 6.2 Contoh program simulasi Algoritma Semut yang sudah ada

## REFERENCES

- [1] <http://www.anneahira.com/algoritma-semut.htm>
- [2] <http://edvinramadhan.blogspot.com/2010/10/sekilas-tentang-algoritma-semut-antnet.html>
- [3] [http://id.wikipedia.org/wiki/Algoritma\\_semut](http://id.wikipedia.org/wiki/Algoritma_semut)
- [4] Rinaldi Munir, Matematika Diskrit edisi ketiga, Informatika, 2005.
- [5] [http://en.wikipedia.org/wiki/Ant\\_colony\\_optimization\\_algorithms](http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms)
- [6] <http://kovarik.felk.cvut.cz/ant-algorithms/materials-applets.php>

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

ttd

Fabrian Oktavino H  
13510053