

# Algoritma Koloni Semut dan Manfaatnya untuk Menentukan Jalur Pengumpulan Sampah

Muhtar Hartopo 13513068  
Program Sarjana Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
muhtar\_hartopo@s.itb.ac.id

**Abstract**— Pada proses pengangkutan sampah dari tempat sampah menuju ke tempat pembuangan akhir sampah, seringkali petugas salah mengambil rute pengumpulan sampah, akibatnya proses pengambilan sampah di tempat sampah menjadi lambat. Untuk mengatasi permasalahan tersebut maka dilakukan pemodelan lokasi tempat sampah dan jalurnya dengan menggunakan graf. Petugas harus mendatangi seluruh tempat sampah yang ada pada suatu kawasan (kota, desa, kompleks perumahan, dll) untuk mengumpulkan sampah. Dalam representasi graf, hal ini sama dengan mencari sirkuit Hamilton berbobot minimum dalam graf. Permasalahan ini merupakan contoh permasalahan *Traveling Salesman Problem (TSP)*. TSP dikenal sebagai permasalahan yang sulit karena dibutuhkan waktu lama untuk menyelesaikannya. Oleh karena itu diperkenalkan sebuah algoritma optimasi yang didasarkan pada perilaku hidup semut yaitu *Ant Colony Optimization* salah satu algoritmanya adalah *Ant Colony System (ACS)*.

**Keywords**— Graf, TSP, ACS, semut.

## I. PENDAHULUAN

Sampah adalah salah satu permasalahan yang menjadi perhatian saat ini. Semakin rusak kondisi lingkungan akibat sampah membuat semakin banyak pergerakan untuk menyelamatkan lingkungan dan gerakan peduli terhadap sampah. Salah satu contoh kebijakan pemerintah kota Bandung untuk mengurangi permasalahan sampah itu adalah menyediakan banyak tempat sampah di berbagai tempat di penjuru kota Bandung.

Hal ini terlihat mampu mengurangi kecenderungan masyarakat untuk membuang sampah sembarangan. Masyarakat sudah mulai belajar untuk membuang sampah di tempat sampah. Namun ada satu permasalahan lagi yang muncul yaitu pengangkutan sampah dari tempat sampah ke tempat pembuangan akhir dinilai lambat dan tidak teratur. Akibatnya tempat sampah mengalami kelebihan kapasitas dan akhirnya sampah kembali berserakan di sekitar tempat sampah.

Keadaan ini dapat dimodelkan dengan tempat sampah sebagai simpul, jalur dari suatu tempat sampah ke tempat sampah lainnya sebagai sisi dengan nilai waktu tempuhnya dan pengumpul sampah mendatangi setiap tempat sampah. Agar sampah dapat diangkut dengan

cepat maka pengumpul sampah harus memilih jalur dengan waktu minimum. Persoalan ini merupakan salah satu persoalan *The Salesman Problem (TSP)*.

Salah satu algoritma untuk menyelesaikan permasalahan TSP adalah algoritma *Ant Colony System* (sistem koloni semut). Algoritma ini diperoleh dengan mengamati cara koloni semut untuk menentukan jalur terpendek dari sarang menuju sumber makanan mereka.

## II. GRAF

### A. Definisi graf

Graf didefinisikan sebagai sebuah struktur diskrit yang terdiri dari pasangan simpul dan sisi. Secara matematis graf dapat dituliskan sebagai

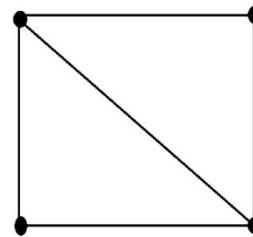
$$G = (V, E)$$

$$G = \text{Graf}$$

$$V = \text{himpunan tidak kosong dari simpul (vertices)} \\ = \{v_1, v_2, v_3, \dots, v_n\}$$

$$E = \text{himpunan sisi (edges) yang menghubungkan} \\ \text{sepasang simpul} \\ = \{e_1, e_2, e_3, \dots, e_n\}$$

Graf yang hanya memiliki satu buah simpul tanpa memiliki sisi disebut graf virtual. [2]



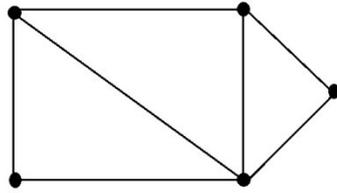
Gambar 1. Contoh sebuah graf dengan 4 simpul

### B. Jenis-jenis graf

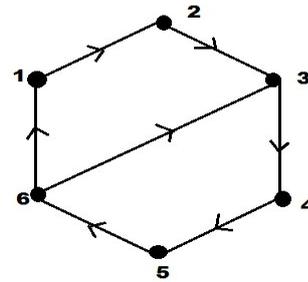
1. Berdasarkan ada atau tidaknya gelang atau sisi ganda pada graf, maka graf dibagi atas dua jenis yaitu :

#### a. Graf sederhana

Graf sederhana adalah graf yang tidak memiliki sisi ganda ataupun sisi gelang. Sisi ganda terbentuk jika terdapat lebih dari satu sisi yang menghubungkan dua buah simpul yang sama. Sisi gelang terbentuk jika sebuah sisi menghubungkan dua buah simpul yang sama. [2]



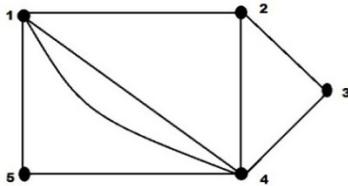
Gambar 2. Contoh graf sederhana



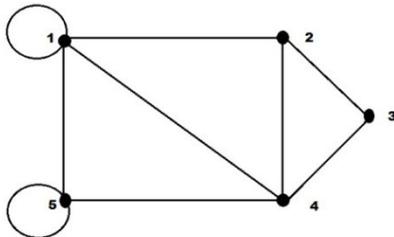
Gambar 6. Contoh graf berarah

b. Graf tidak sederhana

Graf tak sederhana adalah graf yang memiliki gelang atau sisi ganda didalamnya. Terdapat dua jenis graf tak sederhana yaitu graf ganda dan graf semu. Graf ganda (*multiple graph*) adalah graf yang memiliki sisi ganda. Graf semu adalah graf (*pseudo graph*) yang memiliki gelang.



Gambar 3. Contoh graf ganda

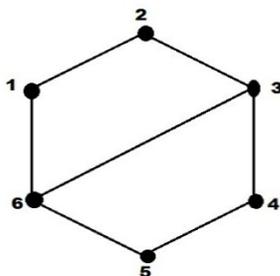


Gambar 4. Contoh graf semu

2. Berdasarkan orientasi arah pada sisi, graf dibagi menjadi dua jenis yaitu :

a. Graf tak berarah

Graf berarah adalah graf sisinya tidak memiliki orientasi arah



Gambar 5. Contoh graf tak berarah

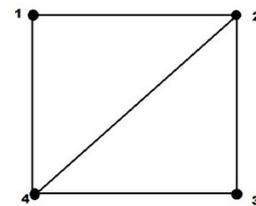
b. Graf berarah

Graf berarah adalah graf setiap sisinya memiliki orientasi arah

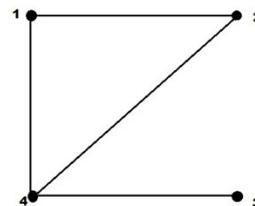
C. Lintasan dan Sirkuit Hamilton

Lintasan Hamilton merupakan lintasan yang melalui seluruh simpul pada sebuah graf tepat sebanyak satu kali. Jika lintasan Hamilton ini kembali ke simpul asal sehingga membentuk sebuah lintasan tertutup (membentuk sebuah sirkuit) maka sirkuit ini dinamakan sirkuit Hamilton. Dengan kata lain, sirkuit Hamilton adalah sirkuit yang melalui setiap simpul pada graf tepat sebanyak satu kali kecuali simpul awal (sekaligus juga sebagai simpul akhir.) yang dilalui sebanyak dua kali.

Graf yang memiliki lintasan Hamilton disebut sebagai graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi-hamilton. Setiap graf lengkap adalah graf Hamilton [2].



Gambar 7. Contoh graf Hamilton



Gambar 8. Contoh graf semi-Hamilton

III. TRAVELING SALESMAN PROBLEM (TSP)

*Traveling Salesman Problem (TSP)* adalah sebuah persoalan klasik yang banyak dipelajari oleh orang-orang di bidang informatika/ilmu komputer, matematika, optimasi kombinatorial dll. Persoalan TSP telah dikaji dari beberapa dekade yang lalu tapi belum ditemukan algoritma paling optimal untuk menyelesaikan permasalahan tersebut. [3]

Definisi permasalahan TSP pada waktu itu adalah terdapat sebanyak N buah kota, kemudian seorang penjajak (*salesman*) berangkat dari satu kota kemudian ia harus mendatangi semua kota kemudian kembali ke kota

asal dengan biaya yang paling sedikit. Biaya pada persoalan ini yaitu jarak, energi, ongkos dan sebagainya. Persoalan TSP ini telah dikaji sejak tahun 1950-an. Persoalan TSP sangat mudah untuk dideskripsikan namun sangat sulit untuk diselesaikan.

Dalam teori graf TSP dinyatakan sebagai graf  $G_{TSP} = (N,A)$ . Dengan N sebagai himpunan semua simpul pada graf, A sebagai himpunan semua sisi yang menghubungkan sisi pada graf tersebut. Setiap sisi memiliki bobot masing-masing. Dengan kata lain TSP merupakan persoalan untuk mencari sirkuit Hamilton dengan bobot minimum pada suatu graf terhubung.

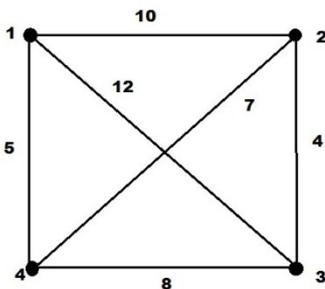
Untuk menghitung banyaknya kemungkinan sirkuit Hamilton pada suatu graf lengkap, maka kita misalkan terdapat n buah simpul. Kita mulai bergerak pada suatu simpul, sehingga terdapat n-1 buah pilihan. Kemudian di simpul ke dua kita ingin bergerak ke simpul ke 3 terdapat n-2 pilihan. Dari simpul ke 3 kita ingin bergerak ke simpul ke empat maka terdapat n-3 pilihan. Sehingga totalnya adalah

$$(n-1)(n-2)(n-3)\dots(1) = n!$$

Karena sirkuit hamiltonnya terhitung sebanyak dua kali maka jumlah kemungkinan sirkuit Hamilton pada suatu graf lengkap dengan n buah simpul adalah

$$\frac{(n-1)!}{2}$$

Sebagai contoh lihat gambar dibawah ini



Gambar 9. Graf lengkap dengan 4 simpul

Pada pada graf diatas (gambar 9) terdapat n = 4 buah simpul. Sehingga terapat sirkuit Hamilton sebanyak

$$\frac{(n-1)!}{2} = \frac{(4-1)!}{2} = \frac{3!}{2} = 3 \text{ buah.}$$

Seacara teoretis TSP dapat diselesaikan dengan melakukan *exhausted search* terhadap seluruh sirkuit Hamilton yaitu sebanyak  $\frac{(n-1)!}{2}$ , menghitung bobot sirkuit lalu mencari sirkuit dengan nilai paling minimum. Namun hal ini sangat sulit untuk dilakukan untuk n yang sangat besar. Contohnya untuk n = 14 saja, maka akan terdapat 3.113.510.400 (lebih dari 3 milyar) buah sirkuit Hamilton. Sangat susah untuk melalukan menyelesaikannya dengan cara yang biasa. Waktu yang dibutuhkan meningkat secara eksponensial seiring dengan bertambahnya jumlah

simpul. TSP memiliki kompleksitas algoritma yang sebanding dengan  $O(n) = n!$ .

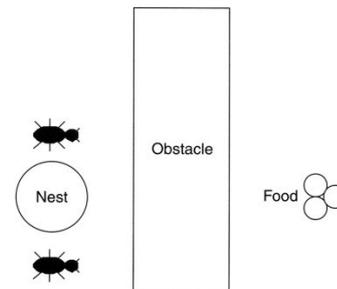
#### IV. ALGORITMA SISTEM KOLONI SEMUT

Pada tahun 1996, Moysen dan Manderic memperkenalkan sebuah algoritma optimasi yang diambil dari perilaku semut mencari jalur tercepat dari sarang menuju sumber makanannya.

##### A. Cara Kerja Algoritma Semut Mencari Jalur Optimal

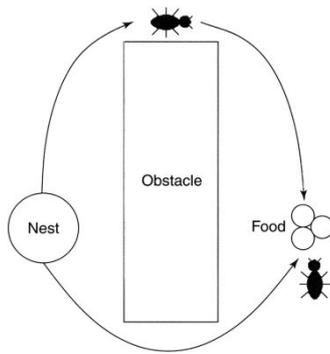
Walaupun semut itu tidak bisa melihat, semut dapat mengenali lingkungannya dengan menggunakan zat kimia yang diekskresikannya kelingkuangan. Zat ini bernama feromon. Proses pengeluaran feromon ke lingkungan disebut *stigmergy*. Saat semut berjalan dari sarangnya menuju ke lokasi makanan maka semut akan mengeluarkan zat feromon tersebut. Feromon yang dikeluarkannya digunakan oleh semut untuk mengingat jalan pulang kembali ke sarangnya. Tak hanya itu zat feromon juga digunakan oleh semut sebagai sarana berkomunikasi semut dengan teman-teman koloninya [1]. Seiring berjalannya waktu, feromon yang ditinggalkan oleh semut akan menguap, sehingga jika semut lama kembali ke sarangnya maka semakin banyak feromon yang menguap dan konsentrasi feromonnya akan semakin berkurang.

Berikut ini ilustrasi perjalanan semut mencari jalur tercepat menuju makanan mereka.



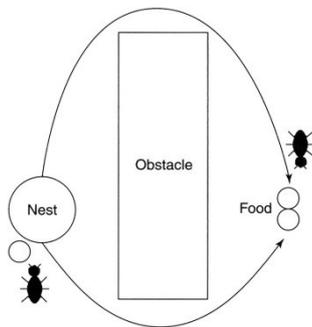
Gambar 10. Ilustrasi dua ekor semut bersiap  
Sumber : AI Application Programming

Pada keadaan awal terdpat dua ekor semut yang dihalangi oleh sebuah penghalang untuk menuju ke tempat makanannya berada.



Gambar 11. Kedua semut bergerak menuju makanan  
 Sumber : AI Application Programming

Pada keadaan awal, peluang terpilihnya kedua lintasan sama. Sehingga satu semut bergerak melalui lintasan bagian atas (misal semut 1) dan satunya lagi melalui lintasan bagian bawah (misal semut 2). Pada perjalanannya kedua semut meninggalkan feromon. Jarak tempuh pada lintasan bawah lebih pendek dari pada lintasan bagian atas sehingga semut yang melalui lintasan bawah lebih cepat sampai. Dengan kata lain jalur yang ditempuh oleh semut 2 adalah jalur optimal.



Gambar 12. Semut yang melewati lintasan bawah kembali  
 Sumber : AI Application Programming

Setelah semut 2 sampai kemudian semut 2 kembali ke sarangnya dengan membawa makanan kemudian kembali ke melakukan perjalanan seperti sebelumnya, sedangkan semut 1 baru sampai di lokasi makanannya. Karena waktu yang dibutuhkan oleh semut 1 ke makanannya lebih lama dari semut 2, maka konsentrasi feromon yang semut 1 yang tersisa di lingkungan lebih sedikit dari konsentrasi feromon semut 2, maka setelah kembali ke sarangnya, semut 1 akan mengikuti jalur yang ditempuh oleh semut 2 menuju ke lokasi makanannya.

B. Sistem Koloni Semut

Sistem koloni semut atau *Ant Colony System* (ACS) adalah sebuah metodologi yang dihasilkan berdasarkan pengamatan dari perilaku semut. Metodologi ini merupakan pengembangan dari

metodologi yang ada sebelumnya yaitu *Ant System* (AS). Kompleksitas waktu algoritma koloni semut ini sebanding dengan  $O(n) = n^3$  [8]. Terdapat 3 bagian utama pada algoritma ACS yaitu :

1. Aturan transisi status

Aturan transisi status yang berlaku pada ACS adalah sebagai berikut: seekor semut yang ditempatkan pada titik  $t$  memilih untuk menuju ke titik  $v$ , kemudian diberikan bilangan pecahan acak  $q$  dimana  $0 \leq q \leq 1$ ,  $q_0$  adalah sebuah parameter yaitu probabilitas semut melakukan eksplorasi pada setiap tahapan, dimana ( $0 \leq q_0 \leq 1$ ) dan  $P_k(t,v)$  adalah probabilitas dimana semut  $k$  memilih untuk bergerak dari titik  $t$  ke titik  $v$ . [6]  
 Jika  $q \leq q_0$  maka pemilihan titik yang akan dituju menerapkan aturan yang ditunjukkan oleh persamaan berikut

$$temporary(t,u) = [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta, i = 1,2,3,\dots,n$$

$$v = \max \{ [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta \}$$

Dengan  $v$  adalah titik yang dituju.  
 Jika  $q \geq q_0$  maka

$$v = p_k(t,v) = \frac{[\tau(t,v)] \cdot [\eta(t,v)]^\beta}{\sum_{i=1}^n [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta}$$

dengan

$$\eta(t,u_i) = \frac{1}{jarak(t,u_i)}$$

Dimana  $\tau(t,u)$  adalah jejak feromon pada titik  $(t,u)$ ,  $\eta(t,u)$  adalah fungsi heuristik dan  $\beta$  adalah parameter heuristik.

2. Aturan perubahan feromon lokal

Ketika melakukan tur TSP, maka semut akan mengubah kadar TSP dengan mengikuti aturan berikut

$$\tau(t,v) \leftarrow (1 - \rho) \cdot \tau(t,v) + \rho \cdot \Delta\tau(t,v)$$

$$\Delta\tau(t,v) = \frac{1}{L_{nn} \cdot c}$$

Dengan  $L_{nn}$  adalah panjang jalur tur,  $c$  adalah jumlah lokasi,  $\rho$  parameter evaporasi feromon dengan rentang 0 sampai 1 dan  $\Delta\tau$  adalah perubahan feromon.

3. Aturan perubahan feromon global

Pada sistem ini perubahan feromon secara global hanya dilakukan oleh semut yang membuat tur terpendek dalam percobaan, perubahan ini dilakukan pada akhir iterasi setelah semua semut menyelesaikan turnya. Perubahan feromon ini memenuhi persamaan dibawah ini

$$\tau(t,v) \leftarrow (1-\alpha) \cdot \tau(t,v) + \alpha \cdot \Delta\tau(t,v)$$

$$\Delta\tau(t,v) = \begin{cases} L_{gb}^{-1} & \text{jika } (t,v) \in \text{tur\_terbaik} \\ 0 & \text{lainnya} \end{cases}$$

Dengan  $\tau(t,v)$  adalah feromon akhir setelah pembaharuan feromon lokal,  $L_{gb}$  adalah panjang lintasan minimum,  $\alpha$  adalah tingkat kepentingan relative feromon dan  $\Delta\tau$  adalah perubahan feromon.

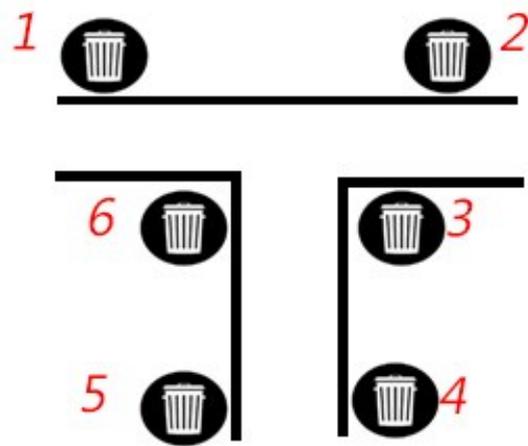
Algoritma ini terdiri dari beberapa tahapan yaitu :

1. Menentukan pheromone awal masing- masing semut. Tapi sebelum itu tentukan terlebih dahulu banyaknya semut dalam proses tersebut, setelah itu tentukan titik awal masing-masing semut.
2. Setelah itu tentukan titik selanjutnya yang akan dituju, ulangi proses sampai semua titik terlewati.
3. Apabila sudah ada titik yang dituju maka dilakukan perubahan feromon lokal
4. Hitung panjang lintasan masing-masing semut
5. Lakukan perubahan feromon global
6. Setelah seluruh proses selesai maka akan didapatkan panjang lintasan minimum.

## V. PEMODELAN JALUR PENGUMPULAN SAMPAH

Untuk mengumpulkan seluruh sampah di suatu kawasan, maka pengumpul sampah harus mendatangi seluruh tempat sampah yang ada pada kawasan tersebut satu persatu. Kemudian agar sampah tidak bertumpuk karena lamanya waktu pengambilan sampah, maka petugas pengumpul sampah haruslah memilih rute tercepat untuk mendatangi seluruh tempat sampah yang ada pada kawasan tersebut. Bila ditelaah maka kita dapat melihat masalah ini sebagai sebuah permasalahan *Traveling Salesman Problem* (TSP). Seperti yang telah dibahas sebelumnya bahwa mencari solusi TSP dengan searching biasa akan memakan waktu yang sangat lama. Oleh karena itu akan digunakan algoritma koloni semut untuk menyelesaikan permasalahan tersebut.

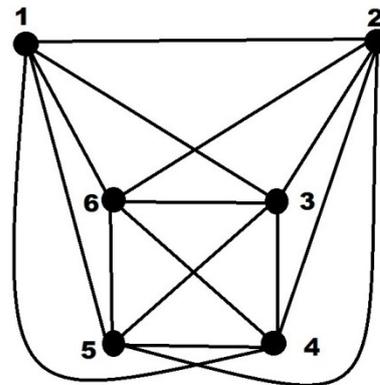
Pertama-tama hal yang harus dilakukan adalah memetakan tempat sampah-tempat sampah yang ada pada suatu daerah. Berikut ini contohnya



Gambar 13. Peta lokasi tempat sampah

Kemudian dari peta lokasi tempat sampah tersebut kita memebentuk sebuah graf dengan memandang tempat sampah sebagai simpul dan jalur yang dilalui dari suatu tempat sampah ke tempat sampah lainnya sebagai sisi yang menghubungkan simpul dan waktu tempuh dari suatu tempat sampah ke tempat sampah lainnya sebagai nilai/bobot dari sisi graf.

Berikut ini contoh graf yang dibentuk dari Gambar 13



Gambar 14. Graf representasi dari gambar 13

Setelah graf terbentuk, selanjutnya adalah menerapkan algoritma sistem koloni semut untuk mencari sirkuit Hamilton dengan bobot minimum. Penjelasan mengenai algoritma sistem koloni semut dapat dilihat pada bagian IV. Pencarian sirkuit Hamilton minimum dapat dilakukan dengan mengimplementasikan algoritma sistem koloni semut kedalam sebuah program.

## VI. KESIMPULAN

Kompleksitas waktu algoritma koloni semut lebih baik dari pada algoritma *exhausted search* dalam menemukan sirkuit Hamilton dengan bobot minimum, sehingga lebih baik menggunakan algoritma koloni semut untuk mencari jalur pengangkutan sampah.

## VII. UCAPAN TERIMAKASIH

Pertama-tama penulis kepada Allah SWT, karena atas rahmat ilmu, kesehatan dan kesempatan yang diberikannya lah penulis dapat menyelesaikan makalah ini. Ucapan terimakasih juga kepada Dr. Ir. Rinaldi Munir atas bimbingan dan jasa beliau yang selama ini telah mengajar dan memberikan ilmu bagi penulis, sehingga penulis mampu membuat tulisan ini. Penulis juga sangat berterimakasih kepada para penulis lain yang karyanya dijadikan referensi oleh penulis. Dan ucapan terimakasih kepada orang tua, dan rekan-rekan yang selalu memberikan motivasi bagi penulis.

## REFERENSI

- [1] Jones M.Tim, "AI Application Programming", Charles River Media INC, Massachussets : 2003
- [2] Munir, Rinaldi, "Matematika Diskrit", Informatika, Bandung: 2010
- [3] Marco Dorigo, Gianni Di Caro, Luca M Garbandella (1999) "Ant Algorith for Discrete Optimation".IRIDIA, Universit' e Libre de Bruxelles Brussels, Belgium
- [4] Ivan Brezina Jr. Zuzana Čičková, "Solving the Travelling Salesman Problem Using the Ant Colony Optimization"(2011), Management Information Systems, Vol. 6 (2011), No. 4, pp. 010-014
- [5] Zar Chi Su Su Hlaing and May Aye Khine (2011)"Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm". Jurnal of Information and Education Technology
- [6] Eka Mindaputra (2009)"PENGUNAAN ALGORITMA ANT COLONY SYSTEM DALAM TRAVELING SALESMAN PROBLEM (TSP) PADA PT. EKA JAYA MOTOR".Universitas Diponegoro.
- [7] Marco Dorigo, Luca Maria Gambardella (1996) "Ant colonies for the traveling salesman problem", IRIDIA Université Libre de Bruxelles Belgium.
- [8] Frank Neumann , Dirk Sudhol, and Carsten Witt "Computational Complexity of Ant Colony Optimization and Its Hybridization"  
[http://homepage.univie.ac.at/Walter.Gutjahr/papers/Rom\\_compose\\_d\\_10.pdf](http://homepage.univie.ac.at/Walter.Gutjahr/papers/Rom_compose_d_10.pdf)  
diunduh pada 11 Desember 2014.pukul 11.13

## 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, 11 Desember 2014



Muhtar Hartopo  
13513068