Penyelesaian Traveling Salesperson Problem dengan Menggunakan Algoritma Semut

Irfan Afif (13507099)
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika,
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung, email: if17099@students.if.itb.ac.id

Abstract *– Traveling Salesperson Problem adalah salah satu permasalahan yang sangat terkenal di dalam teori graf. Permasalahannya adalah diberikan beberapa kota yang saling berhubungan dan jarak dari suatu kota ke kota lainnya. Berapakah jarak minimum yang dibutuhkan untuk mengunjungi setiap kota masing-masing sekali dan kembali lagi ke kota awal. Permasalahan ini dapat diselesaikan dengan menggunakan algortima semut. Algoritma semut adalah algoritma yang didasarkan pada perilaku semut dalam mencari makan. Dengan algoritma semut ini akan dicari penyelesaian dari Traveling Salesperson Problem.*

Kata Kunci*: Traveling Salesperson Problem*, algoritma semut, graf.

1. PENDAHULUAN

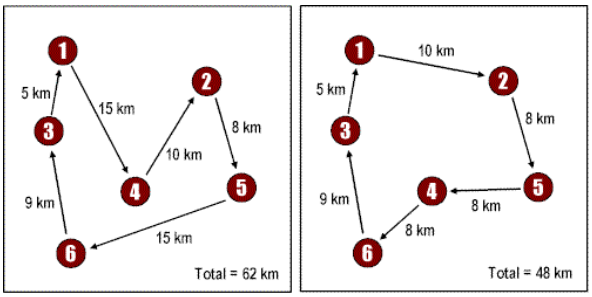
Semut adalah makhluk Tuhan yang unik. Semut mampu mencari makan dengan cara yang efektif. Mereka mampu menemukan jarak terpendek antara sumber makanan dengan sarang mereka tanpa menggunakan indra penglihatan. Pada awalnya mereka berjalan mencari makan secara acak. Sepanjang perjalanan mereka meninggalkan jejak dengan menggunakan feromon. Ketika mereka menemukan makanan, mereka akan kembali ke sarang dengan mengikuti jejak feromon yang mereka tinggalkan. Semut yang lain akan mengikuti jejak feromon ini untuk membantu mengambil makanan tersebut. Feromon ini bisa menguap. Semakin jauh jarak suatu jalan yang dibentuk feromon, maka akan semakin lama semut melintasi jalan tersebut sehingga feromon yang berada di jalan tersebut akan lebih cepat berkurang karena menguap dibandingkan dengan jalan feromon yang memiliki jarak lebih pendek. Karena feromon yang berjarak lebih jauh lebih sedikit daripada feromon yang berjarak lebih pendek, maka semut-semut yang lain tidak terlalu tertarik untuk mengikuti jalan tersebut, sehingga semut-semut akan cenderung mengambil jalan yang paling pendek.

Dengan mengikuti perilaku semut tersebut dalam mencari makan, dibuat suatu algoritma yang dinamakan algoritma semut (*ant colony algorithm*). Fokus dari makalah ini adalah Pemecahan *Traveling Salesman Problem* dengan menggunakan algoritma semut.

2. TRAVELING SALESPERSON PROBLEM

Traveling Salesperson Problem (TSP) merupakan masalah klasik dalam bidang teori graf. Nama persoalan ini diilhami oleh seorang pedagang yang berkeliling mengunjungi sebuah kota. Permasalahannya adalah: diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari kota awal dan menyinggahi semua kota tepat satu kali dan kembali lagi ke kota awal keberangkatan.

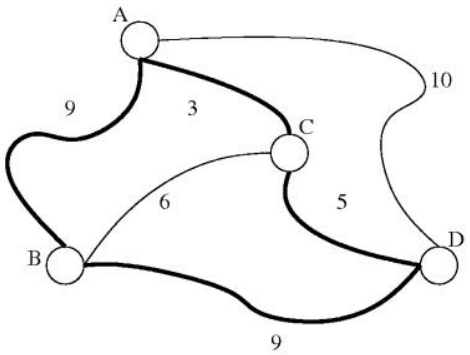
Masalah TSP ini dapat direpresentasikan dengan menggunakan graf berbobot. Setiap kota dilambangkan dengan simpul dan sisi yang berjarak menyatakan hubungan antar kota dan jaraknya. Jawaban dari masalah ini adalah sirkuit Hamilton yang memiliki jumlah bobot paling kecil.



Gambar 1

Gambar 1 merupakan contoh permasalahan TSP. Kedua gambar tersebut adalah sirkuit Hamilton, tetapi gambar disebelah kiri memiliki jumlah jarak yang lebih banyak daripada gambar disebelah kanan.

Pada sembarang graf lengkap dengan n buah simpul (n>2), jumlah sirkuit Hamilton yang berbeda adalah sebanyak (n-)!/2. Rumus ini dihasilkan dari kenyataan bahwa dimulai dari sembarang simpul kita mempunyai n-1 buah sisi untuk dipilih dari simpul pertama, n-2 buah sisi untuk dipilih dari simpul kedua, n-3 buah sisi untuk dipilih dari simpul ke tiga, dan seterusnya. Pada akhirnya kita memiliki (n-1)! Pilihan. Jumlah itu harus dibagi 2, karena setiap sirkuit Hamilton terhitung dua kali sehingga jumlah sirkuit Hamilton yang terbentuk adalah (n-1)!/2.



Gambar 2
Contoh Traveling Salesman Problem

Gambar 1.1 merupakan salah satu contoh dari TSP. Jumlah simpul dari graf tersebut adalah 4, oleh karena itu berdasarkan rumus (n-1)!/2, didapat jumlah sirkuit Hamilton pada graf tersebut adalah 3. Sirkuit Hamilton tersebut adalah A-C-B-D-A, A-B-C-D-A, dan A-B-D-C-A. Lintasan A-C-B-D-A memiliki jumlah bobot 28, lintasan A-B-C-D-A memiliki jumlah bobot 30 dan lintasan A-B-D-C-A memiliki jumlah bobot 26. Oleh karena itu lintasan Hamilton terpendek adalah lintasan A-B-D-C-A.

Penyelesaian masalah TSP diatas adalah dengan menggunakan cara enumerasi semua lintasan Hamilton. Untuk jumlah simpul yang sedikit, cara eumerasi merupakan cara yang paling efektif. Tetapi semakin banyak simpulnya akan semakin sulit untuk melakukan enumerasi karena jumlah sirkuit Hamilton akan bertambah secara faktorial.

Jumlah Simpul	Jumlah sirkuit Hamilton
1	0
2	1
3	1
4	3
5	12
6	60
7	360
8	2520
9	20160
10	181440
11	1814400
12	19958400
13	239500800
…	…
20	6 x 10^16

Tabel 1
Perbandingan antara jumlah simpul dengan banyaknya sirkuit Hamilton pada graf lengkap

Sampai saat ini belum ada algoritma yang mangkus untuk menyelesaikan TSP. Dengan cara mengenumerasim, atau dengan *Brute Force* (mencoba semua kemungkinan), semua sirkuit Hamilton dicari, lalu dibandingkan mana yang memiliki jumlah bobot paling kecil. Dengan cara ini pasti didapat solusi yang pasti, tetapi dari segi komputasi sulit untuk dilakukan. Misalkan ada 20 kota, maka n=20. Jumlah sirkuit Hamilton berdasarkan rumus (n-1)!/2dengan n=20 adalah $6x10^{16}$ sirkuit Hamilton. Dengan jumlah sebanyak itu, dibutuhkan waktu yang lama untuk melakukan komputasi.

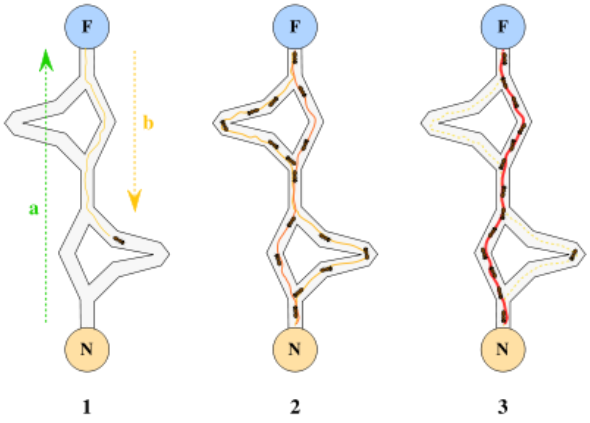
3. Algoritma Semut (*Ant Colony Algorithm*)

Seperti yang sudah dsebutkan di atas, algoritma semut adalah sebuah algoritma yang meniru perilaku semut dalam mencari makan. Algoritma ini biasanya digunakan untuk mencari jarak minimum dari suatu tempat ke tempat lain.

Perilaku semut dapat dimodelkan sebagai berikut:

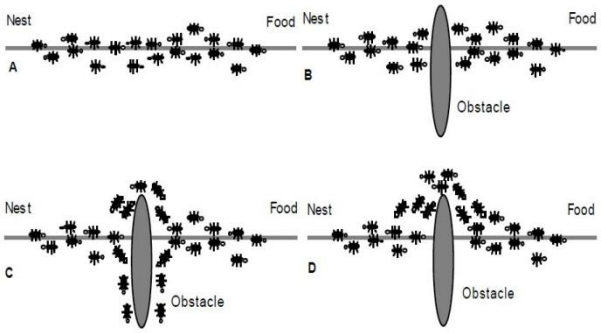
1. Semut berjalan secara acak mencari makanan di sekitar lingkungannya.
2. Ketika mereka menemukan makanan, maka mereka akan kembali ke sarang mereka dan meninggalkan jejak feromon.
3. Feromon ini menarik unuk semut yang lain sehingga semut yag lain akan cenderung mengikuti jejak feromon.
4. Ketika semut-semut yang mengambil makanan kembali, mereka akan memperkuat feromon jalan tersebut.
5. Jika ada lebih dari satu jalan yang terbentuk menuju makanan tersebut, maka pada jalan yang jaraknya lebih jauh akan semakin lama semut bolak-balik melewati jalan tersebut.
6. Akibatnya jalan yang jaraknya lebih jauh akan nehilangan feromon karena feromonnya menguap.
7. Sedangkan jalan yang paling pendek adalah jalan yang paling menarik karena memiliki feromon paling banyak.
8. Pada akhirnya, semua semut akan mengikuti jalan yang paling pendek untuk mendapatkan makanan.

Semut menggunakan lingkungannya sebagai media komunikasi. Mereka bertukar informasi dengan secara tidak langsung dengan menggunakan feromon. Cara mereka bertukar informasi ini hanya berpengaruh semut-semut yang berada pada jangkauan pendeteksian feromon tersebut.



Gambar 3
Lintasan terpendek yang dipakai oleh semut dalam mencari makan

Sistem semut mencari makanan ini menggunakan umpan balik positif, yaitu ketika feromon yang dilepas semut menarik semut lainnya, dan semut tersebut juga mengeluarkan feromon sehingga makin memperkuat feromon di jalur tersebut. Sistem ini juga menggunakan umpan balik negatif, yaitu ketika suatu jalur tidak dipilih karena lebih jauh, maka feromon pada jalur tersebut akan mengilang karena menguap. Jika jumlh feromon tidak berubah mengikuti waktu, maka kecenderungan semut memilih jalan akan sama setiap waktunya. Tetapi karena sistem ini memiliki umpan balik, maka dalam setiap waktu keadaan akan terus berubah dan pilihan jalur yang paling pendek akan menjadi semakin kuat karena jalurnya memiliki jumlah feromon yang paling banyak. Pada awalnya pilihan semut dalam mencari jalur memang tidak stabil, tetapi seiring berjalannya waktu dan akibat dari umpan balik positif dan negatif dari sistem ini, semut-semut akan cenderung memilih jalan yang lebih pendek sehingga jalur yang terbentuk akan menjadi stabil.



Gambar 4
Percobaan pada semut

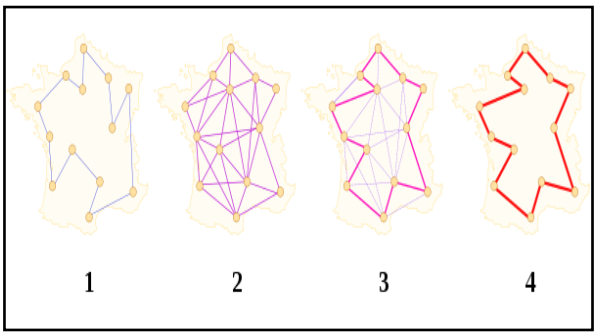
1. Semut sudah menemukan rute antara sarang dan makanan.
2. Rute dihalangi sehingga semut memiliki 2 pilihan, melewati jalan yang panjang atau pendek
3. Pada awalnya kedua jalan dilewati semut. Karena rute bagian bawah lebih panjang, maka intensitas semut melewati jalur bawah semakin sedikit.
4. Pada akhirnya semua semut melewati jalur yang paling pendek

3.1 Semut buatan

Dalam masalah TSP, semut buatan merupakan agen yang bergerak dari satu kota-ke kota lain dalam *Traveling Salesperson Problem*. Semut buatan ini adalah pengandaian dari satu semut yang mencari makan di dekat sarang. Pada awalnya, semut-semut diletakkan pada beberapa kota secara acak. Selanjutnya mereka akan bergerak dari kota-ke kota untuk membentuk sirkuit Hamilton. Dalam setiap perjalanannya semut-semut akan meninggalkan jejak feromon di setiap sisi yang dilewati, dengan kata lain menambah feromon ke sisi. Proses penambahan feromon ke sisi ketika satu semut berjalan melewati sisi disebut sebagai *local updating.* Ketika semua semut telah menyelesaikan semua perjalanannya dan kembali ke kota awal, maka rute terpendek yang dilakukan oleh salah satu semut dicatat. Lalu semua sisi pada rute ini akan ditambahkan feromon. Jumlah feromon yang ditambahkan berbanding terbalik dengan jarak rute tersebut. Proses ini disebut dengan *Global updating.*

Semut ini memilih kota selanjutnya berdasarkan beberapa aturan. Aturan-aturannya adalah:

1. Semut-semut ini harus mengunjungi semua kota dalam graf tepat satu kali. Hal ini sesuai dengan aturan dari sirkuit Hamilton.
2. Kota yang lebih jauh memiliki kemungkinan lebih kecil untuk dipilih sebagai kota selanjutnya ketika semut-semut berjalan dari kota ke kota..
3. Makin banyak feromon yang berada pada suatu sisi diantara dua kota, makin besar kemungkinan semut untuk memilih melewati sisi tersebut.
4. Setiap kali iterasi dilakukan, feromon yang terletak pada sisi akan berkurang.



Gambar 5
Perilaku semut dalam menemukan lintasan terpendek

1. Contoh permasalahan TSP
2. Rute yang diambil semut setelah semut disebar
3. Rute yang paling banyak dilewati semut memiliki jumlah feromon yang paling banyak

Pada penggunaannya, semut buatan dalam program untuk menyelesaikan TSP memiliki kelebihan daripada semut asli. Semut buatan ini harus mampu mengingat berapa jarak yang dia tempuh. Semut ini juga harus memiliki ingatan untuk mengingat kota mana saja yang telah di kunjungi. Ingatan ini terus diperbaharui ketika semut berjalan dari kota ke kot. Ingatan ini diperlukan agar lintasan yang dilewati semut merupakan sirkuit Hamilton.

Semut akan bergerak dari satu kota ke kota lain dengan menggunakan probabilitas. Probabilitas ini berdasarkan jarak kota dan jumlah feromon suatu sisi. Probabilitas tersebut dirumuskan dengan:

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}$$

$\tau_{i,j}$ adalah banyaknya feromon yang terdapat pada sisi antara simpul i dan simpul j

α adalah parameter yang menentukan seberapa besar pengaruh feromon ($\tau_{i,j}$) dalam pengambilan keputusan semut.

$\eta_{i,j}$ adalah nilai dari sisi yang menjadi pertimbangan semut untuk menentukan tujuan, dalam hal ini berbanding terbalik dengan jarak. Semakin kecil jarak sisi semakin besar nilai η

β adalah parameter yang menentukan seberapa besar pengaruh $\eta_{i,j}$ terhadap pengambilan keputusan semut.

$p_{i,j}$ adalah peluang semut yang berada di i untuk menuju simpul j.

Jejak feromon diubah secara global dan lokal. Penambahan feromon yang dilakukan secara global dimaksudkan untuk memberikan penghargaan kepada sirkuit terpendek. Ketika semua semut dalam satu kelompok telah melakukan perjalanan, semut terbaik yang mendapatkan jarak terpendek menambahkan feromon ke sisi yang telah dia lewati. Feromon yang ditambahkan berbaanding terbalik dengan jumlah jarak pada sirkuit yang dilewati. Penambahan feromon secara lokal berguna agar tidak ada sisi yang memiliki feromon sangat banyak sehingga semua semut akan mengambil jalan yang sama.

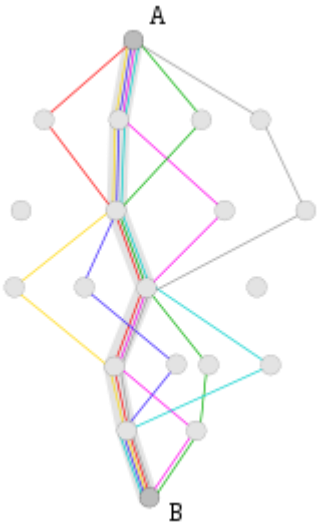
Contoh *pseudocode* dari algoritma semut adalah:

```
procedure Algoritma_Semut
{
     Inisialisasi();
     while(not kondisi_berhenti)
     {
          NextSimpul();
          HitungJarak();
          UpdateMemory();
          UpdateFeromonLokal();
     }
     UpdateFeromonGlobal();
}
```

3.2 Contoh Penggunaan Algoritma Semut

Algoritma semut dapat digunakan untuk menentukan jarak terpendek dari satu simpul ke simpul yang lain. Cara kerja algoritma semut dalam mencari jarak terpendek pada suatu graf adalah:

- beberapa semut dilepas dari simpul awal dan menuju berjalan menuju simpul tujuan
- Sisi yang kemungkinannya paling besar untuk dilewati semut adalah sisi yang paling pendek dan memiliki feromon paling banyak. Tetapi tidak menutup kemungkinan semut mengambil jalan yang lain.
- Semakin lama, jalan yang paling banyak memiliki feromon adalah jalan yang akan dilewati semua semut.
- Jalan itulah yang merupakan jalan terpendek karena kemungkinan sisi yang memiliki feromon terbanyak adalah sisi yang paling pendek.



Gambar 6
Simulasi algoritma semut dalam mencari lintasan terpendek

Pada gambar 6, warna-warna yang berbeda adalah semut-semut yang dilepas untuk mencari jalan. Secara individu, semut-semut tersebut berjalan tanpa melalui lintasan terpendek. Tetapi dengan sistem yang mereka miliki, suma semut selanjutnya akan melewati lintasan terpendek. Lintasan terpendek pada gambar 6 adalah lintasan yang memiliki jumlah warna terbanyak.

4. HASIL DAN PEMBAHASAN

Penyelesaian masalah TSP dengan menggunakan algoritma semut memiliki langkah yang mirip seperti

saat menyelesaikan masalah lintasan terpendek. Perbedaan antara TSP dengan pencarian lintasan terpendek adalah jika pada pencarian solusi lintasan terpendek semut diprogram untuk mencapai satu tujuan dan semut hanya berasal dari satu simpul, sedangkan TSP semut diprogram untuk membuat lintasan Hamilton dan semut ditempatkan pada simpul secara acak.

Berdasarkan percobaan yang dilakukan oleh Dorigo dan Gambardella, didapatkan solusi TSP dengan menggunakan algoritma semut. Untuk mendapatkan bukti kemangkusan algoritma semut, hasil dari algoritma semut akan dibandingkan dengan algoritma-algortima lain untuk permasalahan yang sama. Tabel berikut adalah perbandingan solusi algoritma semut yang dilakukan oleh Dorigo dan Gambardella dengan algoritma lainnya berdasarkan buku (Durbin and Willshaw.1987; Potvin. 1993):

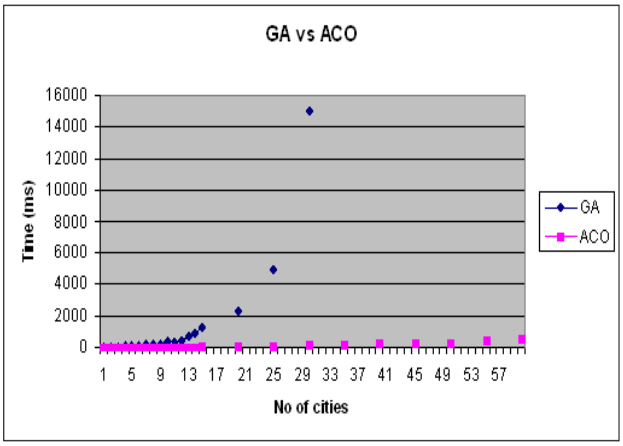
Permasalahan	ACA	SA	EN	SOM	FI
City set 1	5**.86**	5.88	5.98	6.06	6.03
City set 2	6.05	**6.01**	6.03	6.25	6.28
City set 3	**5.57**	5.65	5.70	5.83	5.85
City set 4	**5.70**	5.81	5.86	5.87	5.96
City set 5	**6.17**	6.33	6.49	6.70	6.71

Tabel 2
Perbandingan solusi algoritma semut untuk memecahkan TSP dengan algoritma-algoritma lainnya

Keterangan:
ACA : Ant Colony Algorithm
SA : Simulated Annealing
EN : Elastic Net
SOM : Self Organizing Map
FI : Farthest Insertion

Berdasarkan data pada tabel 2, dari 5 set percobaan, algoritma semut berhasil mendapatkan jarak terpendek sebanyak 4 percobaan. Walaupun begitu tetap saja algoritma semut tidak mendapat lintasan terpendek untuk percobaan ke 2.

Peter Kohout melakukan percobaan untuk membandingkan kemampuan algoritma semut dengan algoritma genetik. Permasalahan yang dipakai dalam percobaan yang dilakukan Kohout adalah *Traveling Salesperson Problem*. Hasil yang didapat adalah waktu komputasi algoritma semut lebih kecil dibandingkan dengan algoritma genetik. Perbedaan waktu ini semakin membesar jika jumlah kota semakin banyak.



Grafik 1
Perbandingan antara algoritma genetik dengan algoritma semut. Keterangan:
Warna merah: algoritma semut
Warna biru : Algoritma Genetik
Sumbu x: Banyak kota
Sumbu y: waktu (ms)

Berdasarkan grafik diatas, waktu yang dibutuhkan algoritma semut untuk melakukan komputasi lebih sedikit daripada algoritma genetik. Untuk jumlah kota yang sedikit, waktu komputasi algoritma semut dan algoritma genetika tidak terlalu berbeda. Tetapi ketika jumlah kota meningkat, maka waktu komputasi algoritma genetika meningkat secara eksponensial. Algoritma semut ketika jumlah kota meningkat, waktu komputasinya naik secara *linier*.

5. KESIMPULAN

Traveling Salesperson Problem adalah suatu permasalahan yang terkenal sulit dalam teori graf. Belum ada algoritma yang mangkus untuk menyelesaikan masalah tersebut. Salah satu algoritma yang digunakan untuk menyelesaikan masalah TSP ini adalah algoritma semut. Algoritma semut adalah sebuah algoritma yang meniru perilaku semut dalam menentukan lintasan dalam pencarian makanan. Algoritma semut ini biasanya digunakan untuk mencari lintasan minimum dari suatu graf berbobot.

Solusi yang dihasilkan algoritma semut untuk menyelesaikan TSP cukup baik. Penyelesaian yang dihasilkan algoritma semut dapat mendekati nilai optimum dengan jumlah iterasi yang lebih sedikit dibanding dengan algoritma lain. Algoritma semut ini mampu menghasilkan solusi yang baik untuk jumlah simpul yang relatif banyak dalam waktu yang singkat relatif dengan algoritma lain.

DAFTAR REFERENSI

[1]Munir,Rinaldi 2008. *Diktat Kuliah IFF2153 Struktur Diskrit Diskrit*. Program Studi Teknik Informatika, Institut Teknologi Bandung; Bandung

[2]http://www.codeproject.com/KB/recipes/GeneticandAntAlgorithms.aspx. Tanggal 3 Januari 2009.
[3]http://mathworld.wolfram.com/AntColonyAlgorithm.html, diakses tanggal 2 Januari 2009
[4]Potvin, J-Y.1993. The Traveling Salesman Problem: A Neural Network Perspektive. ORSA *Journal of Computing.*