

# Penerapan Algoritma *Ant System* dalam Menemukan Jalur Optimal pada *Traveling Salesman Problem* (TSP) dengan Kekangan Kondisi Jalan

Andhi Akhmad Ismail<sup>1</sup>, Samiadji Herdjunanto<sup>2</sup>, Priyatmadi<sup>3</sup>

**Abstract**—The completion of Traveling Salesman Problem (TSP) is to find the shortest path to visit all of the cities. With the shortest path, it is expected that the travel time will also be shorter. In fact, when a salesman visits all of the cities in his list, he will find obstacles such as poor road conditions, congestion, damaged roads, or other constraints. Therefore, although the shortest path has been established, if there is an obstacle the travel time to all cities will be longer. One way to solve the TSP is by ant algorithm. The modifications were made to the Ant System by providing constraint pheromone to each road which could not be passed and also gave a long distance to the roads that should not be passed. The results of this study indicate that the ants never pass constrained sections, for square grid data also two data from TSPLIB95. This occurs because the segments were given constraints, the pheromone were weighted 0 and given the longest distance

**Intisari**— Penyelesaian *Traveling Salesman Problem* (TSP) adalah mencari lintasan terpendek kunjungan ke semua kota. Dengan lintasan terpendek tersebut maka waktu tempuh diharapkan juga akan menjadi lebih cepat. Pada kenyataannya seorang *salesman* ketika mengunjungi semua kota dalam daftar kunjungannya banyak terkendala dengan kondisi jalan seperti kemacetan, jalan yang rusak, atau kendala-kendala lain. Sehingga walaupun lintasan terpendek sudah didapatkan tetapi jika pada lintasan tersebut terdapat ruas jalan yang mengalami gangguan akan menyebabkan waktu tempuh ke semua kota menjadi lebih panjang. Salah satu cara menyelesaikan permasalahan TSP adalah menggunakan algoritma semut. Penulis melakukan modifikasi pada *Ant System* dengan memberi kekangan feromon pada tiap ruas jalan yang tidak bisa dilewati dan memberi jarak panjang pada ruas yang tidak boleh dilewati tersebut. Dengan modifikasi ini diharapkan semut tidak pernah melewati ruas jalan yang mengalami kekangan. Sehingga semut akan mencari jalur lain dalam mengunjungi semua kota yang ada. Hasil penelitian ini menunjukkan semut tidak pernah melewati ruas berkekangan, baik untuk data *square grid* maupun dua data dari TSPLIB95.

**Kata Kunci**— TSP, algoritma semut, *Ant System*, kekangan kondisi jalan, kemacetan, feromon.

## I. PENDAHULUAN

ACO (*Ant Colony Optimization*) mengambil inspirasi dari perilaku semut dalam mencari makanan dari sarangnya. Semut-semut ini menyimpan feromon di permukaan tanah yang digunakan untuk menandai jalur yang akan diikuti oleh semut-semut lain dalam koloninya. ACO menerapkan mekanisme ini untuk memecahkan masalah optimasi. Salah satu permasalahan optimasi adalah TSP (*Traveling Salesman Problem*), penyelesaian TSP adalah mencari lintasan terpendek kota-kota yang akan dikunjungi oleh seorang *salesman*, dengan kekangan kota-kota tersebut hanya dikunjungi sekali sebelum akhirnya kembali ke kota asal.

Seorang *salesman* pada kenyataannya harus menghadapi beberapa kendala dalam mengunjungi kota-kota yang ada dalam daftar kunjungannya. Salah satu kendala adalah adanya kemacetan lalu lintas atau ruas jalan yang tidak boleh dilewati karena jalan tersebut rusak.

Pada tesis ini akan dilihat kemampuan *Ant System* dalam mencari jalur terpendek pada TSP jika ada ruas jalan yang tidak boleh dilewati.

## II. TSP (*TRAVELING SALESMAN PROBLEM*)

TSP menurut A.J Hoffman dan P. Wolfe dalam bukunya berjudul *The Traveling Salesman Problem A Guided Tour of Combinatorial Optimization*, adalah permasalahan yang dihadapi oleh seorang *salesman* yang berangkat dari kota asalnya, berusaha menemukan lintasan paling pendek yang dapat ditempuh ke semua kota pelanggan dan kemudian kembali ke kota asal, dengan syarat semua kota pelanggan hanya boleh dikunjungi sekali dalam tiap *tour*nya.

Pada TSP hasil yang ingin didapatkan adalah menemukan jalur terpendek yang saling terhubung dari  $n$  kota. Tiap kota hanya boleh dikunjungi sekali. Jarak antara kota  $i$  dan kota  $j$  didefinisikan dengan persamaan *Euclidian* dengan  $d_{ij}$ ,  $d_{ij} = \left[ (x_i - x_j)^2 + (y_i - y_j)^2 \right]^{1/2}$ ,  $x_i$  dan  $x_j$  adalah koordinat kota  $i$  [4].

TSP dapat direpresentasikan ke dalam permasalahan graf. Tiap kota diwakili oleh *node* dalam koordinat kartesian. Tiap *node* mempunyai koordinat  $(x,y)$ , dan semua *node* tersebut saling terhubung satu dengan lainnya dengan jarak masing-masing *node* dapat dihitung dari koordinat-koordinat *node*nya. Jika ada *node* yang tidak terhubung, maka *node* tersebut dihubungkan dan diberi nilai jarak yang nantinya tidak bisa digunakan sebagai solusi optimal [4]. Pada Gbr. 1. ditunjukkan

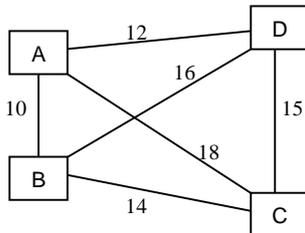
<sup>1</sup> Jurusan Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jln. Grafika 2 Yogyakarta 55281 INDONESIA

<sup>2,3</sup> Jurusan Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jln. Grafika 2 Yogyakarta 55281 INDONESIA

contoh permasalahan TSP empat kota, dengan jarak masing-masing kota diketahui.

Terdapat beberapa kemungkinan lintasan dan panjang lintasan yang dilalui oleh seorang *salesman* dari kota A kaembali ke kota A lagi setelah mengunjungi seluruh kota-kota lainnya. Kemungkinan lintasan adalah sebagai berikut:

1. Lintasan pertama :  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$ , panjang lintasan:  $(10+14+15+12) = 51$
2. Lintasan kedua :  $A \rightarrow D \rightarrow C \rightarrow B \rightarrow A$ , panjang lintasan:  $(12+15+14+10) = 51$



Gbr. 1 Contoh permasalahan empat kota

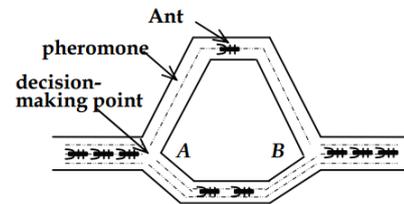
3. Lintasan ketiga :  $A \rightarrow D \rightarrow B \rightarrow C \rightarrow A$ , panjang lintasan:  $(12+16+14+18) = 60$
4. Lintasan keempat :  $A \rightarrow C \rightarrow B \rightarrow D \rightarrow A$ , panjang lintasan:  $(18+14+16+12) = 60$
5. Lintasan kelima :  $A \rightarrow B \rightarrow D \rightarrow C \rightarrow A$ , panjang lintasan:  $(10+16+15+18) = 59$
6. Lintasan keenam :  $A \rightarrow C \rightarrow D \rightarrow B \rightarrow A$ , panjang lintasan:  $(18+15+16+10) = 59$

Jika *node* yang akan diselesaikan pada permasalahan TSP berjumlah banyak, maka kemungkinan penyelesaian akan membutuhkan perhitungan yang banyak pula, karena penyelesaian yang mungkin dilakukan adalah sebanyak  $(n-1)!$ , dengan  $n$  adalah jumlah *node*. Beberapa metode digunakan untuk penyelesaian permasalahan TSP, salah satunya adalah dengan algoritma semut *Ant System*.

### III. ACO (ANT COLONY OPTIMIZATION)

#### A. Prinsip Dasar ACO

Prinsip dari ACO adalah semut selalu meninggalkan suatu zat kimia khusus (feromon) pada jalur yang dilewatinya selama melakukan perjalanan. Feromon yang ditinggalkan pada tempat yang dilewati oleh semut ini menjadi pemandu bagi semut-semut lain dalam melakukan perjalanan. Semakin banyak semut yang melewati jalur tersebut maka jumlah feromon juga akan semakin banyak, sehingga kemungkinan semut-semut lain mengikuti jalur tersebut akan semakin besar. Selanjutnya feromon yang ditinggalkan oleh semut pada suatu jalur akan mengalami penguapan seiring dengan berjalannya waktu.



Gbr. 2 Proses pemilihan jalur oleh semut [11]

Gbr. 2 menunjukkan proses pemilihan suatu jalur oleh semut. Pada saat semut harus memilih jalur mana yang akan diikuti pada titik A, beberapa semut memilih jalur bawah dan beberapa yang lain memilih jalur atas secara acak. Semut-semut tadi berjalan dengan kecepatan yang sama. Semut-semut yang memilih jalur bawah akan sampai pada titik B lebih cepat jika dibandingkan dengan semut-semut yang memilih jalur atas yang lebih panjang. Semut-semut yang memilih jalur bawah akan lebih dulu sampai pada sumber makanan. Jika semut-semut yang sudah sampai dari sumber makanan tersebut kembali ke sarang, maka semut-semut tadi akan mengikuti jalur yang ada feromon yang ditinggalkan sebelumnya. Hal ini menyebabkan jumlah feromon pada jalur bawah akan semakin banyak, dan jumlah semut yang melewati jalur tersebut juga akan semakin banyak.

Semut buatan mengadopsi perilaku semut sebenarnya saat mencari penyelesaian permasalahan optimasi. Feromon merupakan kunci dasar semut-semut saat membuat keputusan [11]

#### B. TSP Menggunakan Ant System [1]

Tiap semut pada awal pencarian jalur terpendek masing-masing menempatkan diri pada tiap kota awal secara acak. Kemudian tiap semut mengunjungi kota-kota lain yang belum pernah dikunjunginya sampai semua kota dikunjungi. Tiap semut akan mempunyai daftar kunjungan kota-kota yang pernah dilewatinya, daftar kunjungan ini disebut *tabu list*.

Pemilihan kota-kota yang belum pernah dikunjungi didasarkan pada suatu aturan yang disebut aturan transisi status (*state transition rule*). Aturan ini mempertimbangkan visibilitas (invers jarak satu kota ke kota lainnya) dan jumlah feromon yang terdapat pada tiap ruas yang menghubungkan satu kota dengan kota lain. Semut pada AS akan cenderung memilih kota berikutnya yang paling dekat dengan kota asal (mempunyai visibilitas besar) dan jumlah feromon terbanyak pada suatu ruas.

*Tabu list* yang dimiliki oleh masing-masing semut berfungsi untuk melarang semut mengunjungi kota yang sudah pernah dikunjunginya. Ketika sebuah *tour* selesai dilakukan, *tabu list* ini berfungsi untuk menghitung panjang lintasan yang sudah dilalui oleh semut pada *tour*nya.

Setelah semut menyelesaikan *tour*nya, *tabu list* akan penuh. Panjang *tour* yang dilakukan oleh tiap semut dihitung berdasarkan *tabu list* tersebut. Langkah selanjutnya adalah melakukan proses pembaharuan feromon pada tiap ruas yang dilalui oleh semut. Suatu aturan pembaharuan feromon global (*global pheromone updating rule*) diberlakukan pada tiap ruas. Semakin pendek sebuah *tour* yang dihasilkan oleh semut, jumlah feromon yang ditinggalkan pada ruas-ruas yang dilalui

akan semakin besar. Hal ini akan menyebabkan ruas-ruas yang diberi feromon lebih banyak akan semakin diminati semut pada *tour* selanjutnya. Sebaliknya pada ruas-ruas dengan feromon sedikit semakin kurang diminati oleh semut, sehingga pada *tour* selanjutnya akan semakin jarang dilewati, sampai akhirnya ruas tersebut tidak pernah dilewati lagi. Pada pembaharuan feromon global dilakukan proses penguapan feromon. Penguapan feromon bertujuan agar tidak terjadi stagnasi, yaitu suatu peristiwa ketika semua semut berakhir dengan melakukan *tour* yang sama.

Proses diulang sampai *tour* yang dilakukan mencapai jumlah maksimal atau sistem ini menghasilkan perilaku stagnasi, yaitu perilaku ketika sistem ini berhenti mencari solusi alternatif. Selanjutnya *tour* terpendek yang ditemukan oleh semut disimpan dan *tabu list* dikosongkan kembali.

1) *Aturan Transisi Status*: Suatu aturan transisi adalah probabilitas semut  $k$  untuk berkunjung dari kota awal  $i$  menuju kota berikutnya  $j$  selama membangun suatu solusi ke- $t$ . Aturan ini disebut dengan *random proportional rule*. Probabilitas transisi dari kota  $i$  ke kota  $j$  oleh semut  $k$  pada AS didefinisikan sebagai berikut [1]:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta}, & \text{if } j \in N_i^k \\ 0, & \text{selainnya} \end{cases} \quad (1)$$

$\tau_{ij}$  adalah jumlah feromon yang ada pada tiap ruas antara *node*  $i$  dan *node*  $j$ ,  $\eta_{ij}$  adalah invers jarak antara *node*  $i$  dan *node*  $j$  ( $1/d_{ij}$ ), nilai  $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ .  $\alpha$  adalah suatu parameter yang mengendalikan bobot feromon,  $\beta$  adalah parameter pengendali jarak, dan  $N_i^k$  adalah himpunan *node-node* yang belum dikunjungi oleh semut  $k$ .

2) *Update Pheromone Trail*: Setelah semua semut selesai membangun sebuah *tour*, jejak feromon yang ada pada tiap ruas diperbaharui nilainya. Pembaharuan nilai feromon ini dilakukan dengan lebih dulu mengurangi (menguapkan) feromon yang ada pada ruas dengan suatu nilai penguapan konstan, kemudian menambahkannya dengan feromon baru. *Update* feromon dilakukan sebagai berikut[3]:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{if } (i, j) \in \text{tour yang ada pada } \text{tabu}_k \\ 0, & \text{selainnya} \end{cases} \quad (4)$$

$\rho$  adalah parameter penguapan feromon,  $m$  adalah jumlah semua semut, *tour described by*  $\text{tabu}_k$  adalah *tour* yang dilakukan semut  $k$  berdasarkan  $\text{tabu}_k$ , dan  $L_k$  adalah panjang *tour* yang dilakukan semut  $k$ ,  $Q$  adalah tetapan jumlah feromon untuk disimpan.

3) *Urutan Penyelesaian TSP dengan Ant System*: Urutan penyelesaian TSP menggunakan *Ant System* dapat disederhanakan sebagai berikut:

**Pertama**, Menentukan parameter-parameter awal *Ant System* yaitu  $\alpha, \beta, \rho, \tau_0 = \tau_{ij}, Q, NCmax$ .

**Kedua**, Menghitung jarak antar *node* dan menghitung visibilitas antar *node*.

**Ketiga**, Menempatkan setiap semut pada tiap *node* secara acak. Posisi awal semut pada *node* pertama ini merupakan posisi pertama *tabu list* tiap semut.

**Keempat**, Mengunjungi *node* lain yang belum ada pada *tabu list* berdasarkan aturan *random proportional rule*. *Node* dengan probabilitas terbesar merupakan *node* yang akan dikunjungi. Kemudian mengisikan kunjungan tersebut pada *tabu list*.

**Kelima**, Setelah semua semut mengunjungi semua kota dan *tabu list* masing-masing semut penuh, panjang lintasan masing-masing semut dalam mengunjungi semua kota tersebut dihitung berdasarkan *tabu list* masing-masing semut.

**Keenam**, Melakukan proses pembaharuan feromon pada tiap ruas dan mengosongkan *tabu list* masing-masing semut.

**Ketujuh**, Mengulang proses ketiga sampai proses keenam dengan menggunakan feromon hasil pembaharuan sebagai feromon awal tiap ruas. Atau menghentikan proses jika  $NCmax$  sudah terpenuhi.

#### IV. CARA PENELITIAN

##### A. Bahan Penelitian

Bahan penelitian berupa koordinat *node-node* dalam bentuk *square grid* dengan ukuran 3x3, 4x4, 5x5, 6x6 *node* dengan jarak sama pada arah sumbu x dan sumbu y. Dari ukuran *node* dan jarak pada tiap *node* satu satuan panjang dapat diketahui panjang lintasan tertutup *node-node* tersebut secara pasti. Selain itu digunakan juga dua data TSP dari <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>, yaitu *ulyse16* dan *ulysses22*, yang berisi data koordinat-koordinat *node* pada penelitian TSP sebelumnya, serta hasil optimalnya.

TABEL I  
DATA PENELITIAN PERMASALAHAN SQUARE GRID

Param	Nilai	Keterangan
$\alpha$	1	parameter bobot untuk feromon tiap lintasan
$\beta$	4	parameter bobot untuk visibilitas tiap lintasan
$\rho$	0,5	parameter penguapan feromon
$Q$	1	konstanta kuantitas jejak yang diletakkan semut
$NCmax_x$	3500	jumlah iterasi maksimal

TABEL II  
DATA PENELITIAN TSPLIB95

Proble m	Dimensi (n)	Solusi Terbaik
3x3	9	9,4142
4x4	16	16
5x5	25	25,4142
6x6	36	36

**B. Alat Penelitian**

Alat penelitian yang digunakan berupa perangkat keras dan perangkat lunak. Alat yang digunakan adalah:

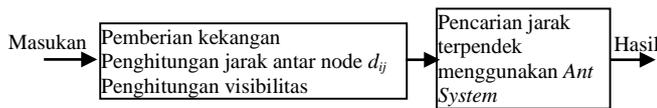
- a. Perangkat Keras:
  - 1. *Notebook* : Intel(R) Pentium(R) Dual CPU T3200 @ 2.00GHz  
Chace Memory 1.024 kB  
RAM 2.00 GB  
HDD Serial ATA 250GB
- b. Perangkat Lunak:
  - 1. Menggunakan sistem operasi MS Windows XP
  - 2. Perangkat lunak aplikasi sebagai alat bantu, yaitu program untuk mengimplementasikan sistem yang dibuat dengan menggunakan Matlab

**C. Jalan Penelitian**

Jalan penelitian yang dilakukan digambarkan dalam diagram blok seperti pada Gambar 3. Pengujian untuk kondisi tidak berkekangan maupun kondisi berkekangan mempunyai parameter sama. Parameter-parameter tersebut ditunjukkan pada Tabel III berikut:

TABEL III  
PARAMETER PERCOBAAN ANT SYSTEM

Problem	Dimensi (n)	Urutan Node
ulysses16	16	1-14-13-12-7-6-15-5-11-9-10-16-3-2-4-8-1
ulysses22	22	1-14-13-12-7-6-15-5-11-9-10-19-20-21-16-3-2-17-22-4-18-8-1



Gbr. 3 Diagram blok penelitian

Masukan berupa data seperti pada Tabel I dan Tabel II, dari data tersebut dicari panjang *tour* minimal tanpa kekangan menggunakan *Ant System*. Selanjutnya adalah mencari panjang *tour* minimal jika satu atau beberapa ruas diberi kekangan.

Kekangan diberikan pada feromon awal (proses pertama), dan saat *update* feromon (proses keenam). Kekangan berupa matrik dengan ukuran  $n \times n$ , diagonal matrik kekangan adalah 0, ruas yang tidak boleh dilewati diberi bobot 0, sedangkan ruas yang boleh dilewati diberi bobot 1.

Hasil percobaan berupa urutan kunjungan kota, dan panjang lintasan kunjungan kota-kota dalam mencari solusi terbaik. Pada pembahasan selanjutnya akan ditunjukkan hasil percobaan dan pembahasannya untuk data *square grid* 4x4.

**V. PERCOBAAN DAN PEMBAHASAN**

Percobaan dilakukan dengan menerapkan modifikasi pada *Ant System*. Data percobaan yang digunakan adalah *square grid* 4x4. Tiap percobaan yang dilakukan adalah mencari penyelesaian permasalahan tanpa adanya kekangan dan jika diberi kekangan. Kekangan diberikan awalnya pada satu ruas kemudian dilanjutkan pada beberapa ruas, sampai pada pemberian kekangan yang ekstrim. Percobaan pada

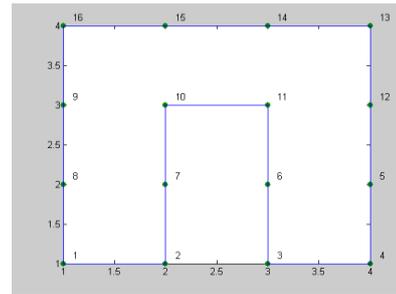
permasalahan berkekangan dilakukan sebanyak 10 kali dan hasilnya ditabelkan.

Semua parameter percobaan dibuat sama, baik pada permasalahan tanpa kekangan maupun pada permasalahan dengan kekangan. Parameter tersebut adalah  $\alpha = 1$ ,  $\beta = 4$ ,  $\rho = 0,5$ , dan  $NC_{max} = 3500$

**A. Percobaan Tanpa Kekangan**

Percobaan untuk data *square grid* 4x4 menggunakan *Ant System* diperoleh hasil seperti berikut:

Panjang *tour* terbaik : 16;  
Urutan kunjungan :  
3→6→11→10→7→2→1→8→9→16→15→14→13→12→5→4→3



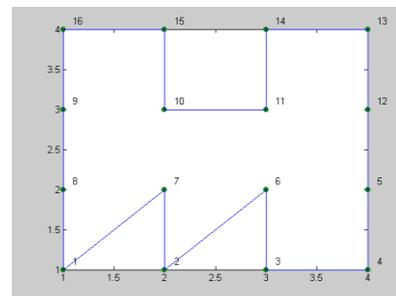
Gbr. 4 Lintasan terpendek *square grid* 4x4 tanpa kekangan

**B. Percobaan dengan Kekangan**

1) *Ruas Antara Node 1-2 Tidak Boleh Dilewati:*

Percobaan yang dilakukan didapatkan hasil sebagai berikut

Panjang *tour* terbaik : 16,8284;  
Urutan kunjungan :  
12→13→14→11→10→15→16→9→8→1→7→2→6→3→4→5→12



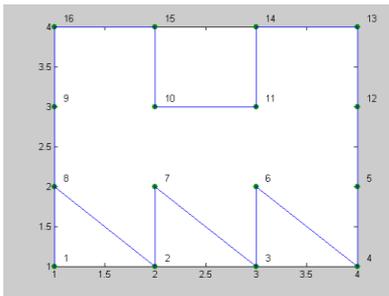
Gbr. 5 Lintasan terpendek *square grid* 4x4 node 1-2 berkekangan

2) *Ruas Antara Node 1-2-3-4 Tidak Boleh Dilewati:*

Percobaan yang dilakukan didapatkan hasil sebagai berikut

Panjang *tour* terbaik : 18,2426;

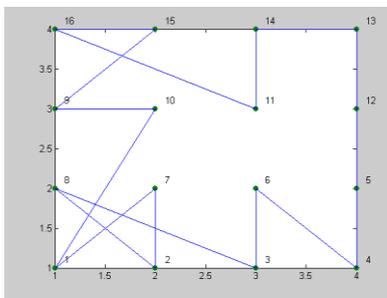
Urutan kunjungan :  
 5→4→6→3→7→2→8→1→9→16→15→10→11→14→13→12→5



Gbr. 6 Lintasan terpendek square grid 4x4 node 1-2-3-4 berkekangan

3) Ruas Antara Node 1-2-3-4, dan Antara Node 1-8-9-16 Tidak Boleh Dilewati: Percobaan yang dilakukan didapatkan hasil sebagai berikut

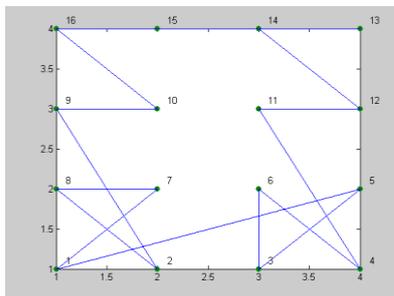
Panjang tour terbaik : 21,3651;  
 Urutan kunjungan :  
 7→1→10→9→15→16→11→14→13→12→5→4→6→3→8→2→7



Gbr. 7 Lintasan terpendek square grid 4x4 node 1-2-3-4 dan node 1-8-9-16 berkekangan

4) Ruas Antara Node 1-2-3-4, Antara Node 1-8-9-16, dan Antara Node 4-5-12-13 Tidak Boleh Dilewati: Percobaan yang dilakukan didapatkan hasil sebagai berikut

Panjang tour terbaik : 24,1197;  
 Urutan kunjungan :  
 8→7→1→5→3→6→4→11→12→14→13→15→16→10→9→2→8

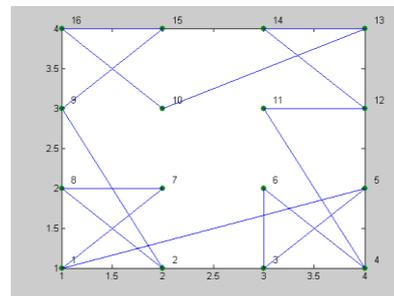


Gbr. 8 Lintasan terpendek square grid 4x4 node 1-2-3-4, node 1-8-9-16, dan node 4-5-12-13 berkekangan

5) Ruas Antara Node 1-2-3-4, Antara Node 1-8-9-16, Antara Node 4-5-12-13, Antara Node 7-6-5, dan Antara Node

9-10-11 Tidak Boleh Dilewati: Percobaan yang dilakukan didapatkan hasil sebagai berikut

Panjang tour terbaik : 24,7700;  
 Urutan kunjungan :  
 2→8→7→1→5→3→6→4→11→12→14→13→10→16→15→9→2



Gbr. 9 Lintasan terpendek square grid 4x4 node 1-2-3-4, node 1-8-9-16, node 4-5-12-13, node 7-6-5, dan node 9-10-11 berkekangan

Dari percobaan yang dilakukan untuk square grid 4x4 dapat dilihat bahwa semakin banyak jumlah kekangan yang diberikan akan menghasilkan tour yang semakin panjang, hal ini terjadi karena semakin banyak lintasan yang tidak boleh dilewati menyebabkan semut mencari jalur lain yang lebih panjang dari lintasan tanpa kekangan

### VI. KESIMPULAN

Pada percobaan dari data square grid maupun TSPLIB95 tidak ada kekangan yang dilewati oleh semut. Hal ini terjadi karena ruas berkekangan diberi feromon dengan bobot 0 serta diberi jarak paling besar, sehingga probabilitas semut memilih ruas berkekangan selalu paling kecil.

### REFERENSI

- [1] M. Dorigo and T. Stützle, *Ant Colony Optimization*, The MIT Press, Cambridge, Massachusetts London, England, 2004.
- [2] M. Dorigo, V. Maniezzo, and A. Colomi, *Positive feedback as a search strategy*, Technical Report 91-016, 1991.
- [3] M. Dorigo, V. Maniezzo, and A. Colomi, *The Ant System: Optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol.26, No.1, 1996, pp.1-13, 1996.
- [4] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence From Natural to Artificial Systems*, New York Oxford, Oxford University Press, 1999.
- [5] I. Mutakhiroh, F. Saptono, N. Hasanah, R. Wiryadinata, *Pemanfaatan Metode Heuristik Dalam Pencarian Jalur Terpendek Dengan Algoritma Semut dan Algoritma Genetik*, Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007), ISSN: 1907-5022, 2007.
- [6] I. Mutakhiroh, Indrato, T. Hidayat, *Pencarian Jalur Terpendek Menggunakan Algoritma Semut*, Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007), ISSN: 1907-5022, 2007.
- [7] A. Leksono *Algoritma Ant Colony Optimization (ACO) Untuk Menyelesaikan Traveling Salesman Problem (TSP)*, Skripsi, F. MIPA UNDIP, 2009.
- [8] B. Yuwono, AS. Aribowo, BW. Siswanto, *Implementasi Algoritma Koloni Semut Pada Proses Pencarian Jalan Protokol di Kota Yogyakarta*, Seminar Nasional Informatika (semnasIF), ISSN: 1979-2328, 2009.
- [9] CJ. Eyckelhof, N. Snoek, *Ant Systems for a Dynamic TSP Ants caught in a traffic jam*, Dept. of Computer Science, University of Twente, 2001.
- [10] M. Guntsch, M. Middendorf and H. Schmeck, *An ant colony optimization approach to dynamic TSP*. In Lee Spector et al., editor, *Proceedings of the Genetic and Evolutionary Computation*

- Conference (GECCO-2001), pages 860–867, San Francisco, California, USA, 7-11 July 2001.
- [11] E. Chen and X. Liu, *Multi-Colony Ant Algorithm*. In *Ant Colony Optimization-Methods and Applications*. Edited by Avi Ostfeld., InTech, 2011