

IMPLEMENTASI LOCAL PHEROMONE UPDATING RULE PADA ALGORITMA ANT COLONY UNTUK MEMBANTU MENCARI PENYELESAIAN TRAVELING SALESMAN PROBLEM

Susana Limanto - Melissa Angga*

* *Fakultas Teknik, Jurusan Teknik Informatika, Universitas Surabaya*
susana@ubaya.ac.id – melissa@ubaya.ac.id

Abstract

Ant Colony Algorithm was developed based on the ant's behaviour on the real world. On the real world, the ants grouped as blind animals, however they have the ability to discover the shortest path from their nest to the food resources and return. As a media of communication along the path and to decide the preference path, the ants used a substance called pheromone. While they strolling, the ants marked their preference path by putting some pheromone (*local pheromone updating rules*). Thus the other ants while they moving would prefer more on the path with more pheromone on it which has put by the previous ants rather than the others.

Ant Colony Algorithm is an heuristic algorithm which is suitable for solving the combinatory problem. One example of a combinatory problem is the Travelling Salesman Problem. Travelling Salesman Problem (TSP) would solve the problem to find the cheapest cost to travelling each town once and then return to the origin town. This research is conducted to implement the ant colony algorithm to solve the travelling salesman problem.

Keyword : *Ant Colony algorithm, Travelling Salesman Problem, local pheromone updating rules.*

1. Pendahuluan

Ant colony algorithm yang dikembangkan oleh para ilmuwan untuk mencari penyelesaian dari Traveling Salesman Problem (TSP), selama ini dilakukan dengan menempatkan seekor *artificial ant* di setiap vertek (kota), kemudian menjalankan *artificial ant* ini melewati semua vertek yang lainnya sampai kemudian tiba kembali di vertek semula. Setelah semua *artificial ant* kembali ke vertek semula, maka jumlah *pheromone* yang terletak pada setiap jalur yang menghubungkan dua buah vertek akan mengalami penyesuaian karena setiap *artificial ant* akan menjatuhkan *pheromone* di setiap jalur yang dilaluinya. Penyesuaian jumlah *pheromone* setelah semua *artificial ant* kembali ke tempat semula ini disebut dengan *global pheromone updating rule* (Dorigo dan Gambardella, 1996).

Pada kenyataannya, *global pheromone updating rule* ini kurang realistis karena biasanya semut akan menjatuhkan *pheromone* pada saat dia melintasi suatu jalur, bukan setelah dia kembali ke tempat semula. *Pheromone* yang dijatuhkan pada saat semut melintasi suatu jalur mungkin akan mempengaruhi semut yang lain untuk mengambil jalur tersebut tepat pada saat itu juga (tidak menunggu kembali ke tempat semula dulu). Dengan mempertimbangkan kasus di atas, maka penelitian ini akan mencoba mengembangkan *ant colony algorithm* dengan memakai *local pheromone updating rule*.

2. Traveling Salesman Problem

Sebuah Graf yang dinyatakan dengan notasi $G(N, E)$ adalah suatu pasangan terurut antara himpunan vertek $N = \{n_1, n_2, n_3, \dots, n_m\}$ dan himpunan busur $E = \{e_1, e_2, e_3, \dots, e_n\}$. E adalah himpunan busur yang menghubungkan vertek-vertik dalam N (Liu 1995). Setiap vertek biasanya digambarkan sebagai titik dan setiap busur digambarkan sebagai garis. Sebuah graf yang tidak memiliki arah tertentu disebut sebagai graf tak berarah. Pada graf tak berarah busur (A, B) dianggap sama dengan busur (B, A) . Suatu graf tidak berarah disebut graf lengkap (undirected

complete graph) jika setiap vertek terhubung ke semua vertek lainnya yang menyusun graf tersebut.

Busur-busur yang menyusun sebuah graf dapat memiliki bobot yang mewakili suatu besaran tertentu seperti panjang busur. Jika setiap bobot dari busur yang menghubungkan vertek A dengan vertek B pada suatu graf berbobot sama dengan bobot dari busur yang menghubungkan vertek B dengan vertek A maka graf tersebut disebut graf simetris (Cook dan Cvatal 2005). Salah satu permasalahan yang dapat digambarkan dengan graf berbobot adalah TSP.

TSP dapat digambarkan sebagai permasalahan untuk mencari jalur tertutup terpendek yang dapat ditempuh jika seseorang harus mengunjungi semua vertek pada sebuah graf G tepat satu kali diawali dari vertek awal hingga kembali ke vertek awal semula. Graf G dapat diperhitungkan sebagai sebuah peta yang mempunyai n vertek yang dapat dianggap sebagai n kota. Busur yang menyusun graf dianggap sebagai jalur yang dapat dilalui sedangkan bobot mewakili jarak antara vertek-vertek yang menghubungkan busur tersebut.

3. Hasil dan Pembahasan

Untuk menyelesaikan TSP, *ant colony algorithm* mengawali dengan menempatkan seekor *artificial ant* pada setiap vertek penyusun graf. Setelah seluruh *artificial ant* diletakkan pada setiap kota, maka pada waktu $t = 0$ semua *artificial ant* tersebut akan mulai bergerak. Untuk setiap interval waktu antara t dan $t + 1$, setiap *artificial ant* akan bergerak dari satu vertek ke vertek berikutnya yang belum pernah dikunjungi. Pada saat seekor *artificial ant* k berada pada vertek i , ia akan memilih vertek j sebagai vertek tujuan berikutnya jika vertek j memiliki nilai probabilitas terbesar. Probabilitas yang digunakan merupakan fungsi dari jumlah *pheromone* dan jarak, seperti terlihat pada Persamaan 1.

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{z \in \{kota\ yang\ belumdikunjungi\}} [\tau_{iz}(t)]^\alpha [\eta_{iz}]^\beta} & \text{if } j \in \{kota\ yang\ belum\ dikunjungi\} \\ 0 & \text{yang lainnya} \end{cases} \dots\dots\dots 1$$

Di mana :

- α menyatakan derajat kepentingan dari *pheromone*, $\alpha \geq 0$.
- β menyatakan derajat kepentingan dari jarak, $\beta \geq 0$.
- η_{ij} adalah suatu fungsi yang nilainya diperoleh dari $1/(\text{jarak antara vertek } i \text{ dan } j)$.
- $\tau_{ij}(t)$ adalah intensitas *pheromone* pada busur yang menghubungkan vertek i dengan vertek j pada waktu ke t .

Penyesuaian atau perhitungan ulang nilai *pheromone* pada setiap busur yang menghubungkan dua buah vertek dilakukan setiap kali seekor *artificial ant* melewati sebuah busur tertentu. Proses perubahan seperti ini dinamakan *Local pheromone updating rule*. Ketika seekor semut melalui sebuah jalur, jumlah *pheromone* pada jalur tersebut telah berkurang karena sebagian telah menguap yaitu sebesar persentase penguapan *pheromone* (ρ) yang telah ditentukan sebelumnya. Pada saat yang sama jumlah *pheromone* juga bertambah sebesar jumlah total intensitas *pheromone* ($\Delta\tau$) yang dijatuhkan seekor semut. Jumlah total intensitas *pheromone* ($\Delta\tau$) pada suatu busur dapat diperoleh dengan membagi jumlah *pheromone* yang dijatuhkan (Q) dengan panjang busur penghubung vertek i dan vertek j (d_{ij}). Untuk menghitung nilai intensitas *pheromone* (τ) pada suatu busur setelah seekor semut melakukan perpindahan digunakan Persamaan 2.

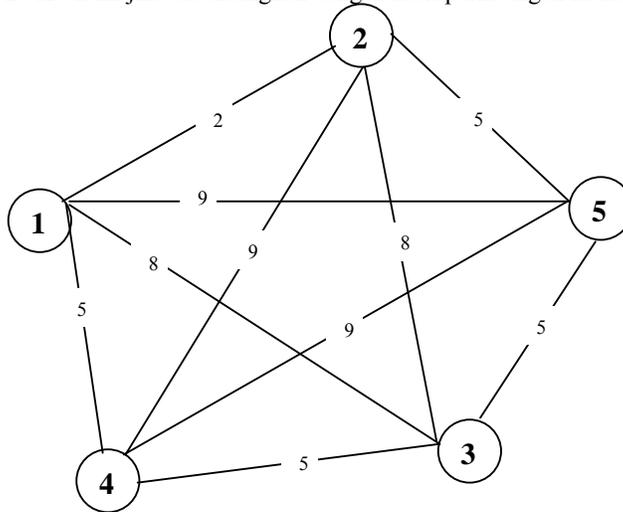
$$\tau_{ij} = [(1 - \rho) * \tau_{ij}] + \left[\frac{Q}{d_{ij}} \right] \dots\dots\dots 2$$

Di mana :

- ρ menyatakan penguapan *pheromone* antara waktu t dan $t + n$, dimana $0 < \rho < 1$. Nilai ini bermanfaat agar tidak terjadi penumpukan *pheromone* secara tidak terbatas mengingat jumlah *pheromone* akan terus bertambah setiap kali iterasi,
- Q merupakan suatu parameter konstan yang menyatakan jumlah *pheromone* yang diletakkan seekor *artificial ant* saat berjalan,
- d_{ij} menyatakan jarak antara node i dengan node j .

Pemilihan jalur terpendek didasarkan pada jalur terpendek yang sudah didapat sebelumnya dibandingkan dengan jalur terpendek yang baru saja diperoleh oleh seekor *artificial ant*. Proses pencarian jalur terpendek akan terus diulang sampai tercapai kondisi stagnan. Kondisi ini tercapai saat jalur yang terbentuk oleh semua semut adalah sama seperti iterasi sebelumnya.

Uji coba dilakukan dengan menggunakan graf yang tersusun atas lima buah vertek seperti pada Gambar 1. Uji coba dilakukan dengan memberikan nilai parameter $\alpha = 1$, $\beta = 1$, $\rho = 0,1$, $Q = 1$, $c = 0,1$, dan $N_{max} = 10$, serta $n = 5$. Hasil yang diberikan oleh program aplikasi dapat dilihat pada Gambar 2. Hasil ini jika dibandingkan dengan hasil perhitungan manual adalah sama.



Gambar 1. Graf dengan lima vertek penyusunnya.

```

ITER : 1
1 4 3 5 2 1 D1:77
2 5 3 4 1 2 D1:22
3 5 2 1 4 3 D1:22
4 1 2 5 3 4 D1:22
5 2 1 4 3 5 D1:22

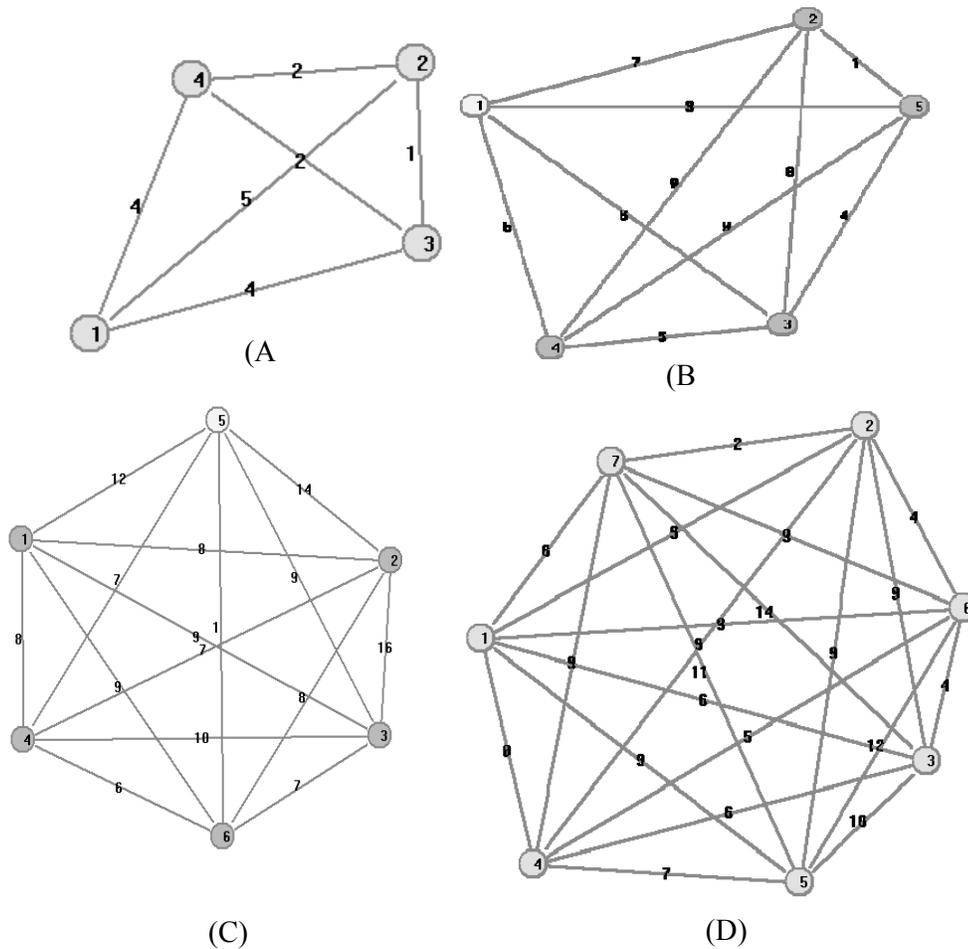
ITER : 2
1 4 3 5 2 1 D1:22
2 5 3 4 1 2 D1:22
3 5 2 1 4 3 D1:22
4 1 2 5 3 4 D1:22
5 2 1 4 3 5 D1:22
  
```

Gambar 2. Hasil program.

Setelah uji coba hasil perhitungan selesai dilakukan, dilanjutkan dengan uji coba parameter. Uji coba ini dilakukan untuk mencari nilai parameter yang terbaik untuk dipakai dalam program agar dapat memberikan hasil yang optimal. Beberapa parameter yang dianggap berpengaruh terhadap hasil perhitungan *ant colony algorithm* dicoba dalam berbagai variasi nilai. Berbagai variasi nilai yang dipakai untuk setiap parameter adalah $\alpha \in \{0, 1, 2, 3, 4, 5, 10\}$, $\beta \in \{0, 1, 2, 3, 4, 5, 10\}$, $\rho \in \{0,1; 0,2; 0,5; 0,9\}$, c (nilai *pheromone* tiap jalur mula-mula) $\in \{0; 0,1; 0,2; 0,5; 0,9\}$, dan $Q \in \{1, 2, 5, 50\}$. Pengujian kali ini dilakukan pada empat graf dengan jumlah vertek yang berbeda-beda (Gambar 3).

Hasil program terhadap graf dengan empat vertek menunjukkan bahwa *ant colony algorithm* memberikan hasil yang terbaik untuk nilai $\rho = 0,1$ dan $\rho = 0,2$ selama nilai β tidak sama dengan 0, nilai c tidak sama dengan 0 dan nilai $Q = 50$. Hasil terbaik disini berarti kondisi stagnan dicapai dengan jumlah iterasi yang sedikit serta panjang perjalanan yang diperoleh minimum. Nilai $\alpha = 0$ dan β tidak sama dengan 0 juga dapat memberikan hasil terbaik. Sedangkan jika nilai $\alpha = \beta$ untuk nilai α dan nilai β tidak sama dengan 0 dan nilai c tidak sama dengan 0, panjang perjalanan minimum dapat dicapai tetapi dengan jumlah iterasi yang lebih banyak. Demikian juga pada saat nilai $\beta = 0$, *ant colony algorithm* selalu memberikan hasil yang buruk, yaitu untuk mencapai kondisi stagnan diperlukan jumlah iterasi yang lebih banyak atau panjang perjalanan yang dihasilkan tidak optimum.

Hasil program terhadap graf dengan lima vertek menunjukkan bahwa *ant colony algorithm* memberikan hasil terbaik pada saat nilai β tidak sama dengan 0 dan nilai c tidak sama dengan 0. Demikian juga untuk nilai $\alpha = 0$ dan nilai β tidak sama dengan 0 dengan nilai $c = 0$. Sedangkan untuk $c = 0$, hasil yang diperoleh tidak baik, kecuali untuk nilai $\rho = 0,1$ dan $\rho = 0,2$ dimana jumlah iterasi yang digunakan untuk mencapai kondisi stagnan lebih sedikit atau nilai panjang perjalanan yang diperoleh lebih minimum dari pada nilai ρ yang lebih besar ($\rho = 0,5$ dan $\rho = 0,9$). Nilai $\beta = 0$ pada contoh kasus ini juga selalu memberikan hasil yang jelek.



Gambar 3. Gambar graf yang digunakan untuk ujicoba.

Hasil pengujian pada graf dengan enam vertek menunjukkan bahwa jika nilai β sama dengan 0, maka hasil yang diperoleh tidak pernah optimum, yaitu jumlah iterasi yang dibutuhkan lebih banyak untuk mencari panjang perjalanan minimum, atau panjang perjalanan tidak pernah minimum dan tetap membutuhkan jumlah iterasi yang lebih banyak. Demikian juga nilai $c = 0$ selalu memberikan hasil yang jelek.

Hasil uji coba pada graf tujuh vertek menunjukkan bahwa *ant colony algorithm* memberikan hasil terbaik untuk nilai $\alpha = 0$ dan β tidak sama dengan 0. Hasil terbaik juga diperoleh saat nilai β lebih besar dari pada nilai α . Sedangkan jika nilai $\beta = 0$, atau $c = 0$, atau $\alpha = \beta$ maka hasil yang diperoleh tidak baik.

4. Kesimpulan Dan Saran

Dari penelitian yang telah dibuat dapat ditarik beberapa kesimpulan, yaitu:

- *Ant colony algorithm* selalu dapat mencari solusi optimum dari kasus TSP dengan kombinasi parameter tertentu.
- Hasil uji coba dengan *ant colony algorithm* pada kasus *traveling salesman problem* menunjukkan bahwa dengan nilai $c = 0$, hasil yang diperoleh cenderung tidak optimum atau jumlah iterasi yang dibutuhkan lebih banyak.
- Hasil uji coba dengan *ant colony algorithm* pada kasus *traveling salesman problem* menunjukkan bahwa hasil yang dicapai cenderung optimum atau jumlah iterasi yang digunakan lebih sedikit jika parameter ρ diisi dengan nilai yang kecil yaitu 0,1 dan 0,2 dan parameter β diisi dengan nilai tidak sama dengan 0 (nol). Parameter β yang diisi dengan nilai sama dengan 0 (nol) menyebabkan solusi yang dicapai tidak optimum. Tidak optimum disini berarti suatu kondisi dimana panjang perjalanan yang dicapai tidak minimum atau jumlah iterasi yang digunakan lebih banyak.

Untuk pengembangan lebih lanjut disarankan untuk melakukan penyesuaian nilai *pheromone* dengan memperhitungkan waktu. Program yang dibuat dalam penelitian ini belum memperhitungkan waktu sehingga perubahan *pheromone* dilakukan setelah setiap semut melakukan satu kali perpindahan padahal jarak tempuh untuk setiap semut belum tentu sama. Semut dengan jarak tempuh lebih pendek, seharusnya lebih cepat melakukan perpindahan sehingga jumlah *pheromone* pada jalur tersebut seharusnya lebih cepat berubah sebelum semut lain tiba di tujuan.

5. Daftar Pustaka

- Bonabeau, E. and Botee, H. M. (1998). Evolving Ant Colony Optimization. *Adv. Complex Systems*, [online]. **1**, pp 149-159. Available from: <http://www.santafe.edu/sfi/publications/Working-Papers/99-01-009.pdf> [Accessed 11 February 2006].
- Chvatal, V. (2005). *The Traveling Salesman Problem*. [online]. Available from: <http://www.cs.concordia.ca/~chvatal/tsp/tsp.html> [Accessed 25 Maret 2006].
- Chvatal, V. and Cook, R. (2005). *Traveling Salesman Problem*. [online]. Available from: <http://www.tsp.gatech.edu/> [Accessed 25 Maret 2006].
- Colomi, A., Dorigo, M. and Maniezzo, V. (1996). The Ant System: By A Colony Of Cooperation Agents. *IEEE Transactions on System, Man, and Cybernetics*. **26**, 1, pp 1-13.
- Croes, G.A. (1958). A Method For Solving Traveling Salesman Problems. *Operations Research*, [online]. **6**, pp 791-812. Available from: <http://www.idsia.ch/~luca/icec96-acsc.pdf> [Accessed 11 February 2006].
- Dakin R. (2002). *The Traveling Salesman Problem*. [online]. Available from: <http://www.pcug.org.au/~dakin/tsp.html> [Accessed 25 Maret 2006].
- DiCaro, G., Dorigo, M. and Gambardella, L. M. (1999). Ant Algorithms for Discrete Optimization. *Artificial Life*, [online]. Available from: http://www.idsia.ch/~luca/ij_23-alife99.pdf [Accessed 11 February 2006].
- Dorigo, M. (1997). Ant Colony Optimization [online] Available from: URL:<http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html> [Accessed 5 February 2006].

- Dorigo, M. and Gambardella, L. M. (1996). Solving Symmetric and Asymmetric TSPs by Ant Colonies. *IEEE Conference on Evolutionary Computation*, [online]. Available from: <http://www.idsia.ch/~luca/icec96-acs.pdf> [Accessed 11 February 2006].
- Dorigo, M. and Gambardella, L. M. (1996). Ant Colony System: A Cooperation Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, [online]. **1**, 1. Available from: <http://www.idsia.ch/~luca/acs-ec97.pdf> [Accessed 11 February 2006].
- Dorigo, M. and Gambardella, L. M. (1997). Ant Colony System: A Cooperation Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, [online]. **1**, 1. Available from: <http://www.idsia.ch/~luca/acs-ec97.pdf> [Accessed: 11 February 2006].
- Limanto, S., Angga, M. (2007), Implementasi Local Pheromone Updating Rule Pada Algoritma Ant Colony Untuk Membantu Mencari Penyelesaian Traveling Salesman Problem, Hasil Penelitian, Ubaya.
- Liu C.L. (1995). *Dasar-dasar Matematika Diskret*. 2nd edn. Gramedia Pustaka Utama. Jakarta