

## PENCARIAN JALUR TERPENDEK MENGGUNAKAN ALGORITMA ANT COLONY OPTIMIZATION

Diana.Fallo

Program Studi Teknik Informatika STKIP Citra Bina Nusantara Kupang

Email : [dianayani25@gmail.com](mailto:dianayani25@gmail.com)<sup>1</sup>

### ABSTRAK

*Ant Colony Optimization (ACO)* merupakan algoritma heuristic yang telah terbukti diterapkan ke sejumlah masalah *Travelling Salesman Problem (TSP)* dan mampu menemukan jalur terpendek dengan baik. Dari hasil uji coba yang dilakukan di temukan bahwa hasil jalur terpendek menggunakan ACO di temukan total waktu tempuh lebih cepat. ACO bekerja dengan sangat cepat dalam menemukan rute terpendek karena ACO tidak tergantung pada iterasi yang diberikan. Jika hasil nya telah di ketahui maka secara otomatis akan di tampilkan hasilnya.

Kata Kunci : *Pencarian Jalur Terpendek, Algoritma ACO*

### 1. Pendahuluan

Swarm intelligence adalah sebuah pendekatan relative baru untuk pemecahan masalah yang mengambil inspirasi dari social perilaku serangga dan hewan lainnya. Upaya dalam penelitian teknologi computer adalah mengembangkan algoritma yang terinspirasi oleh perilaku serangga untuk memecahkan masalah optimiasi. Ant Colony Optimization (ACO) adalah salah satu algoritma yang paling sukses dalam bidang swarm intellenge.(Hlaing & Khine , 2011).

Penyelesaian *Travelling Salesman Problem (TSP)* adalah mencari lintasan terpendek kunjungan ke semua kota. Dengan lintasan terpendek tersebut maka waktu tempuh diharapkan juga kan menjadi lebih cepat. Pada kenyataannyaseorang *salesman* ketika mengunjungi semua kota dalam daftar kunjungan sulit mengetahui jarak terpendek sehingga menyebabkan perjalanan menjadi lebih panjang (Ismail et al., 2012)

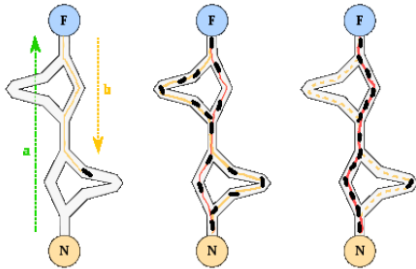
ACO menerapkan mekanisme ini untuk memecahkan masalah optimasi. Salah satu permasalahan optimasi adalah TSP (*Travelling Salesman Problem*), penyelesaian TSP adalah mencari lintasan terpendek kota-kota yang akan di kunjungi oleh seorang *salesman*, dengan kekakangan kita-kota tersebut hanya

dikunjungi sekali sebelum akhirnya kembali ke kota asal (Ismail et al., 2012). Tujuan dari penelitian ini adalah untuk menemukan jalur terpendek menggunakan ACO (*Ant Colony Optimization*)

Menurut penelitian (Tyas & Prijodiprojo, 2013) algoritma koloni semut (ACO) diadopsi dari perilaku koloni semut yang dikenal sebagai system semut. Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang menuju ke sumber makanan dan kembali lagi, pada saat semut berjalan, semut meninggalkan sebuah informasi yang disebut *pheromone*, di tempat yang dilaluinya dan menandai rute tersebut. Penelitian yang dilakukan oleh (Mindaputra, 2009) menemukan bahwa algoritma koloni semut sering digunakan untuk penyelesaian masalah pencarian jalur terpendek salah satunya *travelling salesman problem (TSP)* dan mendekati hasil yang optimal.

Algoritma semut (ACO) telah di perkenalkan oleh Macro Dorigo di awal tahun 1990an. Macro terinspirasi oleh perlakuan seut dalam pencarian makanan. Bagaimana semut dapat menemukan jalur terpendek antara sumber makanan dan sarang mereka. Saat mencari makanan, semut awalnya menjelajahi daerah sekitaryadengan cara acak, (Alhanjouri & Alfarra, 2012).

Penelitian mengenai algoritma semut telah menyumbangkan pengetahuan yang sangat besar terhadap bidang bioinformatika, yang merupakan observasi bagaimana suatu kejadian yang ada di alam dapat di gunakan untuk mengatasi permasalahan yang ada pada dunia industri. Berikut ini merupakan gambaran rute perjalanan semut dalam menemukan makanan.



Gambar 1. Perjalanan semut menemukan sumber makanan (Dorigo,1996)

## 2. METODE PENELITIAN

Menurut penelitian ( Hlaing & Khine, 2011, Sutoyo,2014, Mutakhirah 2007) Ant Colony memerlukan beberapa variabel sebelum melakukan pencarian jalur terpendek yaitu :

### Langkah I

Inisialisasi harga parameter.

- $(\tau_{ij})$  adalah intensitas jejak *pheromone* semut
- (n) adalah banyak kota termasuk koordinat (x,y) atau jarak antar kota ( $d_{ij}$ ) Rumus untuk menentukan jarak adalah  $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  (1)
- Q adalah tetapan siklus semut.
- $\alpha$  adalah tetapan pengendali intensitas jejak semut, nilai  $\alpha > 0$
- $\beta$  adalah tetapan pengendali visibilitas, nilai  $\beta > 0$
- Visibilitas antar kota ( $\eta_{ij} = 1/d_{ij}$ )
- m adalah jumlah semut.
- $\rho$  adalah tetapan penguapan jejak semut, dimana  $0 \leq \rho \leq 1$
- NCmax adalah jumlah siklus maksimum.

### Langkah 2

Pengisian *node* pertama ke dalam *tabu list*.

*Tabu list* digunakan untuk menyimpan daftar urutan *node-node* yang sudah di kunjungi setiap semut. Setiap kali semut berkunjung ke suatu kota maka elemen *tabu list* akan bertambah satu, seterusnya sampai *tabu list* penuh.

### Langkah 3

Hitung persamaan probabilitas untuk menentukan kota tujuan

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in J_r^k} [\tau_{iu}]^\alpha [\eta_{iu}]^\beta} \quad (1)$$

### Langkah 4

Tahap ini dilakukan pembaruan *pheromone* (t) lokal

$$\tau(t, v) = (1 - \rho) \cdot \tau(t, v) + \rho \cdot \Delta\tau(t, v) \quad (2)$$

$$\Delta\tau(t, v) = \frac{1}{L_{min} \cdot C} \quad (3)$$

Dimana

$L_{min}$  = Panjang tur yang diperoleh

C = Jumlah h lokasi

$\rho$  = Parameter dengan nilai 0 sampai 1

$\Delta\tau$  = Perubahan *pheromone*

### Langkah 5

Tahap ini di lakukan pembaruan *pheromone* global

$$\tau(t, v) \leftarrow (1 - \alpha) \cdot \tau(t, v) + \alpha \cdot \Delta\tau(t, v) \quad (4)$$

$$\Delta\tau(t, v) = \begin{cases} L_{gb}^{-1} & \text{jika } (t, v) \text{ etur\_terbaik} \\ 0 & \text{lainnya} \end{cases} \quad (5)$$

Dimana

$\tau(t, v)$  = nilai *pheromone* setelah mengalami pembaruan lokal

$L_{gb}$  = Panjang tur terpende pada akhir siklus

$\alpha$  = parameter dengan nilai antara 0 sampai 1

$\Delta\tau$  = Perubahan *pheromone*

Berikut ini merupakan *pseudocode* menurut (Fenwa et al., 2014 dan Chawda & Sureja, 2012) tentang algoritma koloni semut yaitu :

```

Begin
  Initialize the pheromone
  trails and parameter;
  Generate of m solutions
  (ants);
  For each individual ant k to
  m;
  Calculate probabilitas;
  For each and determine its
  best position;
  Determine the best global
  ant;
  Update the pheromone trail;
  Check if distance true;
End
    
```

### 3. HASIL DAN PEMBAHASAN

Sebuah contoh kasus perusahaan Unilever akan mendistribusikan pada toko-toko di sekitar perusahaanya. Perusahaan tersebut ingin mengetahui jalur terpendek untuk mendistribusikan barang tersebut, dengan ketentuan bahwa :

- Tidak ada toko yang akan terlewatkan untuk di kunjungi
- Perusahaan tersebut tidak/belum mengetahui jarak antara toko dengan toko lainnya.

Jarak antara toko tersebut seperti terlihat pada tabel di bawah ini:

**Tabel 1. Koordinat x dan y**

Toko Ke	Koordinat x	Koordinat y
1	20	25
2	40	30
3	30	45
4	30	35
5	45	55

Adapun langkah-langkah yang digunakan dalam algoritma semut ialah Inisialisasi harga parameter

$m = 5$

$n = 5$

$\alpha = 0.1$

$\beta = 1$

$\rho = 0.5$

$Q = 1$

$\tau_{ij} = 0.01$

$NC_{max} = 2$

Setelah melakukan inisialisasi maka adapun langkah-langkah untuk menyelesaikan masalah optimasi menggunakan algoritma semut ialah : Hitung jarak antar kota dengan kota lain menggunakan persamaan (1) dan dapat dilihat pada tabel dibawah ini

**Tabel 2. Jarak antar kota ( $d_{ij}$ )**

Jarak	1	2	3	4	5
1	0	20	22	14	39
2	45	0	18	11	25
3	22	18	0	10	18
4	14	11	10	0	25
5	39	25	18	25	0

Setelah jarak antar kota di ketahui dapat di hitung visibilitas antar kota yang tunjukkan pada tabel 3.

**Tabel 3. Nilai visibilitas antar kota**

( $\eta_{ij} = 1 / (d_{ij})$ )

Kota	1	2	3	4	5
1	0	0.05	0.0454	0.0714	0.0256
2	0.0222	0	0.0555	0.0909	0.04
3	0.0454	0.055	0	0.1	0.0555
4	0.0714	0.090	0.1	0	0.04
5	0.0256	0.04	0.0555	0.04	0

Kemudian hitung probabilitas tiap kota yang dituju dengan menggunakan persamaan (2). Seluruh titik telah dikunjungi, selanjutnya lakukan pembaruan *pheremone* lokal dengan menggunakan persamaan (3) dan (4) yang ditunjukkan pada tabel 4.

**Tabel 4. Nilai pembaruan *pheremone* lokal**

( $\tau(t, v)$ )

Semut	1	2	3	4	5
S1	0	1,055	0.969	0.782	0.666
S2	1,055	0	0.011	0.019	0.009
S3	0.969	0.011	0	0.014	0.009
S4	0.782	0.019	0.014	0	0.009
S5	0.666	0.009	0.009	0.009	0

Maka rute yang ditempuh serta panjang rute diperoleh dapat di lihat pada tabel 5.

**Tabel 5. Nilai pembaruan *pheromone* lokal ( $\Delta\tau(t, v)$ )**

Semut	Rute	Panjang	$\Delta\tau(t, v)$
S1	[1 3 4 5 2]	0.037	30.86
S2	[2 3 1 4 5]	0.049	25.70
S3	[3 5 4 2 1]	0.045	32.05
S4	[4 5 1 2 3]	0.055	18.00
S5	[1 2 3 5 4]	0.035	25.95

Diketahui bahwa rute terbaik pada iterasi ke lima adalah 0.035 merupakan jalur terpendek. Selanjutnya di lakukan perhitungan pembaharuan *pheremone* global dengan persamaan (5) dan (6).

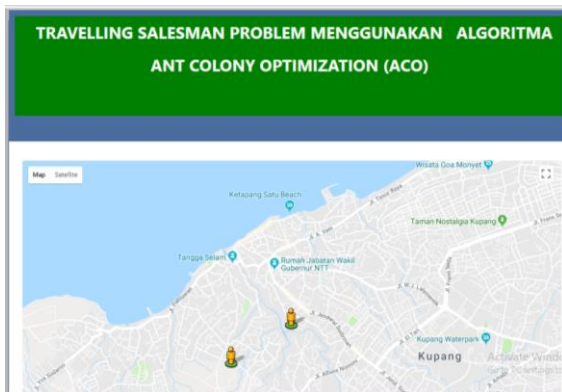
Maka di peroleh  

$$\tau_{21} = \tau_{12} = ((1 - 0.1) * (0.010) + (0.1 * 25.95)) = 2.6031$$
 Jika pada iterasi berikutnya tidak ditemukan rute yang terbaik maka pembaharuan *pheremone* global tetap dilakukan sampai di temukan nilai yang terbaik.

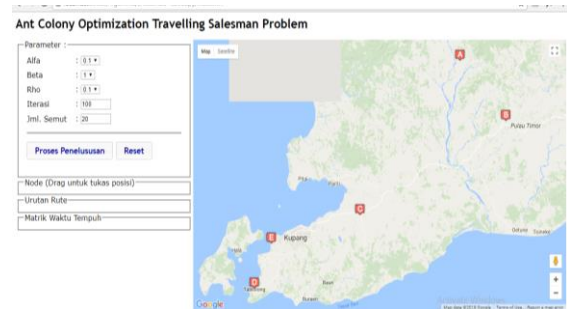
**Pengujian**

Pengujian ini menggunakan aplikasi berbasis *Open Source* Visual Studio 2013 yang di dalamnya terdapat *JavaScript* dan *ASP.Net* sedangkan untuk peta menggunakan *Google Map API*.

**Gambar 2. Tampilan Awal**



Gambar diatas merupakan tampilan awal dari penelitian yang dibuat.



**Gambar 3. Hasil Pengujian**

Pada gambar ini terlihat bahwa nilai alfa, beta, rho, iterasi dan jumlah semut dapat di tentukan oleh *user*. Setelah itu *user* dapat menginputkan titik lokasi dalam hal ini (*latitude dan longitudo*) yang sering dilewati oleh *Travelling Salesman Problem (TSP)*.

**4. KESIMPULAN**

Melalui pengujian yang dilakukan maka dapat disimpulkan

1. Pengujian pada titik lokasi yang sama untuk pencarian jalur terpendek terbukti algoritma ACO mampu bekerja dengan baik dalam penyelesaian masalah TSP.
2. Selanjutnya semakin banyak titik digunakan dan semakin banyak prorese iterasi yang dilakukan maka hasilnya pun akan semakin berbeda.

**5. SARAN**

Penelitian ini menggunakan algoritma ACO dan diharapkan adanya penelitian lebih lanjut dengan menggunakan perbandingan banyak algoritma untuk dapat membedakan manakah algoritma yang lebih baik dalam menentukan rute terpendek.

**Daftar Pustaka**

[1] Alhanjouri, M. & Alferra, B., 2012. Ant Colony Versus Genetic Algorithm Based on Travelling Salesman Problem. *International Journal Comp.Tech.Appl.*, 2 (3), pp.570-78.

[2] Tyas, Y.S. & Prijodiprojo, W., 2013. Aplikasi Pencarian Rute

- Terbaik dengan Metode Ant Colony Optimazation (ACO). *IJCCS*, 7, pp.55-64.
- [3] Muh.Nurtanzis Sutoyo Ant Colony Optimization (ACO). [mr.iyes@yahoo.co.id](mailto:mr.iyes@yahoo.co.id). Artikel. 2014
- [4] Zar Chi Su Su Hlaing and May Aye Khine, *International Journal of Information and Education Technology* Vol 1. No.5 December 2011.
- [5] Mindaputra, E. (2009). *Algoritma Ant Colony Sistem (ACS) Untuk Menyelesaikan Travelling Salesman Problem (TSP)*. Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro, Semarang.
- [6] Murakhiroh, I., Saptono, F., Hasanah, N. & Wiryadinata, R., (2007). Pemanfaatan Metode Heuristik dalam pencarian Jalur terpendek dengan algortima semut dan algoritma genetika. Seminar nasional aplikasi teknologi informasi
- [7] Dorigo, M., 1996. *The Ant Colont Optimization Metaheuristics: Algorithms, Applications and Advances*. Universitas Libre de Bruxelle, IRIDIA.
- [8] Andhi Akhmad Ismail, Samiadji Herdjunto, Priyatmadi, Penerapan Algoritma Ant System dalam menentukan jarak Optimal pada Travelling Salesman Problem (TSP) dengan kekangan kondisi jalan, *JNTETI*, Vol 1 No.3 November 2012.