

PENERAPAN ALGORITMA SEMUT UNTUK OPTIMISASI RUTE PENJEMPUTAN BARANG PADA TEMPAT JASA PENITIPAN SEMENTARA LION EXPRESS

Studi Kasus : Konsolidator Lion Express Tanjungpinang

IkhsanJaelani

Mahasiswa Informatika, FT UMRAH, ikhsanjaelani@gmail.com

ABSTRAK

Lion Express didukung oleh bisnis unit Lion Group, sehingga memiliki jaringan layanan ke seluruh wilayah kota di Indonesia, terutama di daerah terpencil. Mulai dari konsolidator yang mewakili disetiap daerah hingga POS (tempat penyimpanan sementara). Adapun penyebab dan permasalahan pada konsolidator daerah Kota Tanjungpinang yang dihadapi dalam bidang jasa pengiriman Lion Express ini yaitu terjadinya keterlambatan waktu penjemputan barang dari konsolidator menuju setiap POS. Lion Express mempunyai 1 angkutan, dengan waktu bersamaan angkutan Lion Express melakukan penjemputan barang kesetiap POS dan juga melakukan pengantaran barang ke alamat penerima, agar tidak terjadi penumpukan barang didalam angkutan. Dalam hal tersebut dapat mempengaruhi rute penjemputan barang kesetiap POS menjadi tidak optimal. Untuk itu dibutuhkan aplikasi yang dapat menganalisa permasalahan yang ada di konsolidator Lion Express sehingga didapatkan rute optimal.

Dari hasil penelitian ini diselesaikan dengan algoritma semut menggunakan parameter alpha (α) dan beta (β) = 1,0 kemudian nilai rho (ρ) = 0,1 dengan siklus optimum 900, maka didapat hasil rute dari POS 1 (PT. Semangat Persada Tour & Travel) menuju POS 3 (PT. Granindo Duta Selaras) menghasilkan jarak 27,624 Km & waktu 1,289 Jam. Didapat kesimpulan bahwa semakin banyak siklus yang dilakukan, maka semakin akurat semut menentukan rute kunjungannya.

Kata Kunci : Rute Terpendek, meta-heuristics, algoritma semut

ABSTRACT

Lion Express is supported by a business unit of Lion Group, so it has a network of services to all areas of the city in Indonesia, especially in remote areas. Starting from consolidators who represent each area to POS (temporary storage) in various points in order to support progress towards community services. As for the causes and problems at regional consolidator Tanjungpinang encountered in shipping Lion Express this is the time delay pick up goods from consolidators towards each POS. Lion Express has one transport, at the same time transport Lion Express did pick up goods to every POS and also delivers the goods to the recipient's address, in order to avoid the accumulation of goods in transit. In it can affect a shuttle service to any POS items are not optimal. That requires an application that can analyze the problem as it exists in the consolidator Lion Express to obtain the optimal route.

From the results of this study completed by ant algorithms using parameter alpha (α) and beta (β) = 1.0 then the value of rho (ρ) = 0.1 with optimum cycle 900, then the results obtained from POS 1 (PT. Semangat Persada Tour & Travel) to POS 3 (PT. Granindo Duta Selaras) produces 27.624 Km distance and time of 1,289 hours. Concluded that a growing number of cycles performed, the more accurate the ant determines the visit route.

Keywords : Shortest route , meta - heuristics , algorithms ant

I. PENDAHULUAN

A. Latar Belakang

Kemajuan dalam dunia usaha turut berperan penting dalam meningkatkan perekonomian disuatu daerah khususnya di Tanjungpinang dengan menelusuri salah satu jenis usaha yang bergerak pada jasa pengiriman. Bidang jasa pengiriman Lion Express ini salah satunya, didirikan pada tanggal 14 Februari 2013 dan bergerak di bidang jasa pengiriman barang yang melayani wilayah domestik, didukung oleh infrastruktur jaringan Lion Group dan juga sebagai salah satu perusahaan penerbangan terbesar. Sebagai bagian dari Lion Group, misi Lion Express adalah untuk mengembangkan industri logistik dengan filosofi untuk membantu mempercepat pertumbuhan ekonomi di semua wilayah melalui konsep "*Just In Time Air Distribution*".

Adapun penyebab dan permasalahan pada konsolidator daerah Kota Tanjungpinang yang dihadapi dalam bidang jasa pengiriman Lion Express ini yaitu terjadinya keterlambatan waktu penjemputan barang dari konsolidator menuju setiap POS. Lion Express mempunyai 1 angkutan, dengan waktu bersamaan angkutan Lion Express melakukan penjemputan barang kesetiap POS dan juga melakukan pengantaran barang ke alamat penerima, agar tidak terjadi penumpukan barang didalam angkutan. Dalam hal tersebut dapat mempengaruhi rute penjemputan barang kesetiap POS menjadi tidak

optimal. Untuk itu dibutuhkan aplikasi yang dapat menganalisa permasalahan yang ada di konsolidator Lion Express sehingga didapatkan rute optimal penjemputan barang kesetiap POS yang nantinya berdampak pada jarak dan waktu tempuh antar POS (tempat penyimpanan sementara Lion Express). Dimana konsolidator ditugaskan untuk melakukan penjemputan barang kesetiap POS dan pengantaran barang ke alamat penerima.

Terkait uraian diatas, saat ini pada konsolidator Lion Express belum menemukan rute, jarak dan waktu tempuh penjemputan barang yang optimal. Maka peneliti bermaksud untuk membuat sebuah penerapan algoritma semut untuk optimisasi rute penjemputan barang pada tempat jasa penitipan sementara Lion Express, sehingga fokus penelitian ini pada konsolidator Lion Express di Tanjungpinang. Pada penelitian yang dilakukan Mutakhiroh dkk. (2007) dalam hal pencarian jalur terpendek menggunakan algoritma semut, koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilewati. Semakin banyak semut yang melewati suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Algoritma semut sangat tepat digunakan untuk diterapkan dalam penyelesaian masalah optimasi, salah satunya adalah untuk menentukan jalur terpendek.

II. TINJAUAN PUSTAKA

A. Tinjauan Penelitian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini akan dicantumkan beberapa hasil penelitian terdahulu antara lain:

Budi Triandi (2012) dalam penelitian yang berjudul “Penemuan Jalur Terpendek dengan Algoritma Ant Colony” menyimpulkan bahwa algoritma ant colony dapat melakukan optimisasi / pengefisienan waktu dalam penemuan jalur terpendek.

Bambang Yuwono, Agus Sasmito Aribowo, Siswanto Budi Wardoyo (2009) dalam penelitian ini yang berjudul “Implementasi Algoritma Koloni Semut Pada Proses Pencarian Jalur Terpendek Jalan Protokol di Kota Yogyakarta” menyimpulkan bahwa (1) Algoritma Koloni Semut dapat digunakan untuk melakukan pencarian jalur terpendek berdasarkan jarak jalan. (2) Keberhasilan pencarian jalur terpendek bergantung pada jumlah semut. Semakin besar jumlah semut, semakin besar pula kemungkinan keberhasilan pencarian jalur terpendeknya dan hasilnya pun semakin akurat. (3) Pengacakan urutan simpang sebagai dasar pencarian dapat dilakukan dengan mengurutkan simpang secara ascending dan descending pada perulangan siklusnya.

Himmawati Puji Lestari, Eminugroho Ratna Sari (2013) dalam penelitian ini yang berjudul

“Penerapan Algoritma Koloni Semut Untuk Optimisasi Rute Distribusi Pengangkutan Sampah di Kota Yogyakarta” menyimpulkan bahwa menggunakan algoritma koloni semut, pengambilan sampah oleh badan Lingkungan Hidup Kota Yogyakarta menjadi lebih efektif. Pengambilan yang biasanya dilakukan seminggu dua sampai tiga kali untuk masing-masing TPS, dapat dilakukan setiap hari. Selain lebih efektif dilihat dari jarak yang ditempuh, hal ini akan berakibat efektif dari segi biaya. Oleh karena sampah juga terambil setiap hari, maka keluhan masyarakat akan menumpuknya sampah dapat diminimalisir.

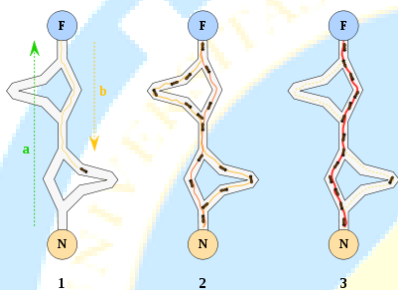
B. Teori Dasar

1. Algoritma Koloni Semut

Ant Colony Optimization (ACO) atau Algoritma Koloni Semut adalah sebuah probabilitas komputasi teknik untuk memecahkan masalah yang dapat dikurangi untuk menemukan jalur yang baik melalui grafik. Algoritma ini adalah anggota dari keluarga algoritma koloni semut, pada intelijen segerombolan metode, dan hal itu merupakan beberapa metaheuristic optimasi. Awalnya diusulkan oleh Marco Dorigo tahun 1992 di gelar PhD tesis, algoritma pertama yang bertujuan untuk mencari jalan yang optimal dalam grafik, berdasarkan perilaku semut mencari jalan antara koloni dan sumber makanan. Ide ini telah diversifikasi untuk menyelesaikan kelas yang

lebih luas dari masalah numerik, dan sebagai hasilnya, beberapa masalah telah muncul, menggambar tentang berbagai aspek perilaku semut.

Gagasan awalnya berasal dari mengamati makanan eksploitasi sumber daya di antara semut, di mana semut secara individual memiliki kemampuan kognitif terbatas secara kolektif mampu menemukan jalur terpendek antara sumber makanan dan sarang.



Gambar 2.1. Ilustrasi Rute Yang Dibentuk Semut dan Koloninya

Keterangan :

1. Semut pertama menemukan sumber makanan (F), melalui cara apapun (a), kemudian kembali ke sarang (N), meninggalkan jejak feromon (b)
2. Semut tanpa pandang bulu cara mengikuti empat kemungkinan, tapi penguatan landasan membuatnya lebih menarik sebagai rute terpendek.
3. Semut mengambil rute terpendek, panjang bagian-bagian dari cara-cara lain kehilangan jejak feromon.

Demikian juga dengan jalan atas, semakin sedikit semut yang melalui

jalan atas, maka feromon yang ditinggalkan semakin berkurang bahkan hilang. Dari sinilah kemudian terpilihilah jalur terpendek antara sarang dan sumber makanan. Dalam algoritma semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek, yaitu:

Berdasarkan penelitian ini, dapat dijabarkan langkah Algoritma Semut sebagai berikut :

Langkah 1 :

- a) Inisialisasi parameter-parameter algoritma. Parameter-parameter yang diinisialisasikan adalah :
 - 1) Alpha (α) adalah pengendali intensitas jejak kaki semut, dengan batasan nilai $\alpha \geq 0$.
 - 2) Beta (β) adalah pengendali intensitas jarak visibilitas pada semut, dengan batasan nilai $\beta \geq 0$.
 - 3) Rho (ρ) adalah tetapan nilai laju penguapan *Pheromone*, dengan batasan nilai $0 < \rho < 1$.
 - 4) Tetapan siklus semut (Q), digunakan apabila terdapat jalur yang dikunjungi oleh semut, dimana (Q) dibagi dengan total jarak (L_k).
 - 5) Jejak *Pheromon* awal (T_{ij}) antar POS dengan berubahannya setiap siklus. Artinya jejak *Pheromone* POS asal (i) ke POS tujuan (j).
 - 6) Nilai jarak antar POS

$$(d_{ij}) = \sqrt{(JP_{ij})^2 + (WT_{ij})^2},$$

dimana i adalah POS asal, j adalah POS tujuan dan

sebaliknya (i) ke (j) = (j) ke (i). Bila dinyatakan :

- JP = Jarak antar POS (Km)
- WT = Waktu tempuh antar POS (Jam)
- 7) Visibilitas (η_{ij}) jarak antar POS asal (i) ke POS tujuan (j) $= \frac{1}{d_{ij}}$
- 8) Banyak Semut (m)
- 9) Banyak POS (n)
- 10) Jumlah siklus maksimum (NC_{max}) bersifat tetap selama algoritma dijalankan, sedangkan T_{ij} (jejak *Pheromone* antar POS) akan selalu diperbaharui nilainya pada setiap siklus algoritma mulai dari siklus pertama (NC_1) sampai tercapai jumlah siklus maksimum ($NC = NC_{max}$) atau sampai terjadi konvergensi.
- 11) POS(k) = data POS yang dipilih, k adalah indeks urutan POS dari 1 sampai dengan n.

- a) Inisialisasi kunjungan POS pertama pada setiap jalur semut. Setelah inisialisasi τ_{ij} (jejak *Pheromon* awal) dilakukan, kemudian semut (m) ditempatkan pada jalur semut pertama tertentu secara acak.

Langkah 2 :

Setelah dilakukan pengacakan kunjungan POS pertama pada jumlah semut setiap jalurnya, maka jalur semut pertama diisi kedalam *Tabu List* sesuai dengan jumlah kunjungan semutnya. Hasil dari pengisian ke dalam *Tabu List* diberi indeks sesuai jumlah jalur

semut, dengan semut pertama berarti bahwa $tabu_k(1)$ yaitu total *tabu* POS pertama dan bisa berisi indeks POS antara 1 sampai n sebagaimana hasil inisialisasi pada langkah 2 ini.

Bila dinyatakan :

- $tabu_k$ adalah rute POS yang berada didalam *tabu list*
- S sebagai indeks POS didalam *tabu list* sebagai contoh $tabu_k(1)$

Langkah 3 :

Menghitung nilai probabilitas, dimana proses ini untuk menentukan rute kunjungan semut selanjutnya. Dimulai dari POS awal ke rute selanjutnya yang terdapat didalam *tabu list* dengan cara mengunjunginya satu persatu hingga POS tujuan akhir tercapai. Jika berada di POS kunjungan kedua maka rute yang dikunjungi selanjutnya yang tidak terdapat pada indeks *tabu list* yang telah dikunjungi. Bila POS lainnya atau POS yang belum dikunjungi dinyatakan dengan $\{N-tabu_k\}$. Untuk menentukan POS tujuan digunakan persamaan probabilitas sebagai berikut :

$$p_{ij}^k = \frac{[T_{ij}]^\alpha \cdot [n_{ij}]^\beta}{\sum_{k' \in [N-tabu_k]} [T_{ik'}]^\alpha \cdot [n_{ik'}]^\beta}$$

nilai $[n_{ij}]$ untuk $j \in \{N - tabu_k\}$ (3.1)

dan Untuk j pada visibilitas $[n_{ij}]$ yang sudah dikunjungi diberi nilai = 0.....(3.2)

Maka p_{ij}^k probabilitas POS asal (i) ke POS tujuan (j) = 0

Keterangan :

- p_{ij}^k = Indeks Probabilitas dimana POS asal (i) ke POS tujuan (j)
- T_{ik} = Nilai *Pheromone* POS asal ke indeks POS tujuan n dan perubahan
- n_{ik} = Nilai visibilitas POS asal ke POS tujuan n
- T_{ij} = Nilai *Pheromone* POS asal ke POS tujuan dan perubahannya
- n_{ij} = Nilai visibilitas antar POS asal ke POS tujuan

Setelah nilai probabilitas didapat, selanjutnya menghitung nilai kumulatif dengan persamaan :

$$kumulatif_k = p_{ij}^k + kumulatif_{k-1} \dots \dots \dots (3.3)$$

Keterangan :

- $kumulatif_k$ = nilai kumulatif indeks POS 1 sampai dengan n
- p_{ij}^k = nilai probabilitas POS asal ke POS tujuan n
- $kumulatif_{k-1}$ = hasil nilai kumulatif indeks POS sebelumnya

Setelah nilai kumulatif didapatkan dan selanjutnya adalah bangkitkan nilai random dan dipilih dari $kumulatif_1$ sampai dengan $kumulatif_n$ untuk mencari POS tujuan yang akan dikunjungi selanjutnya oleh semut dimana nilai random $\leq kumulatif_k$.

Langkah 4 :

- a) Perhitungan panjang rute setiap jalur semut.
Pada proses ini dilakukan setelah satu siklus diselesaikan dimana perhitungan ini dilakukan berdasarkan $tabu_k$ dihitung panjang rute setiap

jalur semut. Dengan persamaan sebagai berikut :

$$L_k = d_{tabu_k(n),tabu_k(i)} + \sum_{s=1}^{n-1} d_{tabu_k(s),tabu_k(s+1)} \dots \dots (3.4)$$

Jarak pada *tabu* indeks POS asal ditambah dengan jumlah jarak *tabu* indeks POS kunjungan ke dua dan indeks POS tujuan selanjutnya sampai dengan *tabu* indeks POS tujuan tercapai.

- b) Pencarian rute terpendek jalur semut.

Setelah L_k setiap semut dihitung, akan didapat nilai minimal panjang rute tertutup setiap siklus atau L_{minNC} dan nilai minimal panjang rute tertutup secara keseluruhan (L_{min}).

- c) Pada proses ini dapat mengalami perubahan nilai *Pheromone* antar POS. Dimana semut akan meninggalkan jejak – jejak kaki pada lintasan antar POS yang dilaluinya. Terjadinya penguapan pada jejak kaki semut, maka persamaan perubahan ini adalah :

$$T_{ij} = \sum_{k=1}^m T_{ij}^k \dots \dots \dots (3.5)$$

Pada $\Delta\tau_{ij}^k$ ini digunakan apabila terdapat jalur semut dari posisi awal dan tujuan merupakan salah satu jalur yang terdapat pada jalur semut ke-n, dengan persamaan :

$$\Delta T_{ij}^k = \frac{Q}{L_k}, \text{ untuk } (i, j) \in \text{POS asal dan POS tujuan dalam } tabu_k \dots \dots \dots (3.6)$$

Dan persamaan ini apabila bukan merupakan \in POS asal dan

POS tujuan dalam $tabu_k$.
 Berikut persamaannya :
 $\Delta T_{ij}^k = 0$, untuk POS (i) ke (j)
 sama dengan nol.....(3.7)

Langkah 5 :

a) Hitung *Update Pheromone* atau nilai intensitas jejak kaki semut. Dimana proses ini untuk melanjutkan perhitungan siklus selanjutnya dihitung dengan persamaan :
 $T_{ij}(\text{baru}) = (1 - \rho) T_{ij} + \Delta T_{ij}$
(3.8)

Keterangan :

ρ = Parameter penguapan *Pheromone* dengan batasan $0 < \rho < 1$.

T_{ij} = Perubahan jumlah nilai *Pheromon* setiap rute kunjungan berdasarkan persamaan (3.5)

ΔT_{ij} = Faktor pembesar mempengaruhi nilai *Pheromone* berdasarkan persamaan (3.6) dan (3.7)

b) Atur ulang nilai perubahan *Pheromone* baru.
 Untuk siklus selanjutnya perubahan nilai intensitas jejak semut antar POS perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6 :

Menampilkan hasil rute optimal dalam bentuk *Google Map* dan Mengosongkan *tabu list*, dan ulangi langkah 2 jika diperlukan, *Tabu list* perlu dikosongkan untuk diisi lagi dengan urutan POS yang baru pada siklus selanjutnya, apabila jumlah siklus maksimum belum tercapai,

Algoritma diulang lagi dari langkah 3 dengan nilai *Pheromone* antar POS yang sudah diperbaharui.

1. Ukuran Kedekatan

a. Jarak *Euclidean* mengukur jumlah kuadrat perbedaan nilai pada masing-masing variabel (Laeli, 2014).

$$d_{ij} = \sqrt{\sum_{k=1}^p (X_{ik} - X_{jk})^2}$$

Dimana :

d_{ij} = jarak antara obyek ke-i dan obyek ke-j

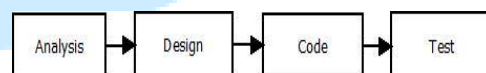
p = jumlah variabel *cluster*

X_{ik} = data dari subjek ke-i pada variabel ke-k

X_{jk} = data dari subjek ke-j pada variabel ke-k

III. METODE PENELITIAN

Penelitian yang akan dilakukan ini menggunakan model pengembangan *Waterfall*. Proses pengembangannya dilakukan melalui beberapa tahap yaitu : *Analisa kebutuhan, Design, Coding, Pengujian* dan *Pemeliharaan*. Pada metodologi pengembangan ini hanya sampai pada tahap pengujian (testing) saja.



Gambar 3.1. Model Pengembangan

1. Berikut adalah penjelasan bagaimana metode pengembangan sistem yang digunakan *Analisis*

Pada tahap ini menguraikan kebutuhan sistem yang utuh menjadi komponen-komponen sistem untuk mengetahui bagaimana sistem dibangun dan untuk mengetahui kelemahan – kelemahan sistem yang sudah ada sehingga dapat dijadikan masukan dan pertimbangan dalam penyusunan sistem yang baru.

2. *Design*

Pada tahap ini merupakan tahap perancangan sistem. Tahap *design* ini menggunakan flowchart berfungsi untuk menyatakan aliran metode atau proses sehingga memberi solusi dalam penyelesaian masalah yang ada di dalam proses atau algoritma tersebut. Sementara Entity Relationship Diagram (ERD) digunakan untuk membantu menggambarkan diagram sistem yang akan dibangun.

3. *Code*

Pada tahap ini adalah penerjemahan rancangan dalam tahap desain ke dalam bahasa pemrograman.

4. *Test*

kasus Konsolidator Lion Express di Tanjungpinang.

Tabel 4.1. Tabel Data Proses

Variable	Nilai
Jumlah Angkutan	1
Jumlah POS	5
POS	POS 1
POS	POS 3
POS Yang Dikunjungi	POS 2 – POS 4 – POS 5

Tabel 4.2. Tabel Data Jarak POS

Nama POS	1	2	3	4	5
1		1,300 KM 443 dtk	1,390 KM 336 dtk	4,270 KM 733 dtk	1,970 KM 296 dtk
2	1,300 KM 443 dtk		1,150 KM 320 dtk	3,680 KM 480 dtk	3,270 KM 739 dtk
3	1,390 KM 336 dtk	1,150 KM 320 dtk		4,440 KM 643 dtk	3,360 KM 632 dtk
4	4,270 KM 733 dtk	3,680 KM 480 dtk	4,440 KM 643 dtk		4,630 KM 527 dtk
5	1,970 KM 296 dtk	3,270 KM 739 dtk	3,360 KM 632 dtk	4,630 KM 527 dtk	

Tabel 4.3. Tabel Titik koordinat POS

Kode POS	Nama POS	Titik Koordinat	
		Latitude	Longitude
POS 1	PT. SP	0,9170027	104,5036490
POS 2	CV. Jaya Bersama	0,9187490	104,5135959
POS 3	PT. GDS	0,9124121	104,5120630
POS 4	PT. Rainbow T&T	0,9446121	104,5135312
POS 5	Teuku Imam .M	0,918397	104,489639

IV. PEMBAHASAN

Pada bab ini akan dijelaskan mengenai analisa pembahasan hasil penelitian pada system optimasi penjemputan barang pada tempat penitipan sementara Lion Express dengan menggunakan Algoritma Semut dengan data studi

A. Optimasi Rute Penjemputan Barang Pada Penitipan Sementara Lion Express Menggunakan Algoritma Semut.

1. Menentukan parameter

Alpha : 1.0

Beta : 1.0

Rho : 0.5

Q : 1

Feromon Awal : 0.01

Jumlah Siklus (NCmax) : 1

Tabel 4.4. Tabel T_{ij} (Feromon Awal)

Nama POS	POS 1	POS 2	POS 3	POS 4	POS 5
POS 1	0	0.01	0.01	0.01	0.01
POS 2	0.01	0	0.01	0.01	0.01
POS 3	0.01	0.01	0	0.01	0.01
POS 4	0.01	0.01	0.01	0	0.01
POS 5	0.01	0.01	0.01	0.01	0

Selanjutnya akan dilakukan perhitungan nilai jarak antar POS (d_{ij}) dengan memperhatikan data jarak antar POS dan waktu tempuh antar POS, berikut perhitungannya:

$$d_{ij} = \sqrt{(P_{ij})^2 + (WT_{ij})^2}$$

$$d_{12} = \sqrt{(1,300)^2 + \left(\frac{443}{3600}\right)^2} \quad d_{12} = \sqrt{1,6900000000000000 + (0,1230555555555560)^2}$$

$$d_{12} = \sqrt{1,6900000000000000 + 0,0151426697530864}$$

$$d_{12} = \sqrt{1,7051426697530900}$$

$$d_{12} = 1,30581111156492298$$

Dengan cara yang sama, dihitung sampai semua POS kunjungan. Maka didapat hasil perhitungan jarak antar POS (d_{ij}). Maka didapat tabel jarak antar POS sebagai berikut :

Tabel 4.5. Tabel Jarak Antar POS

Nama POS	POS 1	POS 2	POS 3	POS 4	POS 5
POS 1	0	1,30581	1,39312	4,27485	1,97171
POS 2	1,30581	0	1,15343	3,68241	3,27643
POS 3	1,39312	1,15343	0	4,44359	3,36458
POS 4	4,27485	3,68241	4,44359	0	4,6323
POS 5	1,97171	3,27643	3,36458	4,63231	0

Tabel 4.6. Tabel Visibilitas Antar POS

Nama POS	POS 1	POS 2	POS 3	POS 4	POS 5
POS 1	0	0,765	0,717	0,233	0,507
POS 2	0,765	0	0,866	0,271	0,305
POS 3	0,717	0,866	0	0,225	0,297
POS 4	0,233	0,271	0,225	0	0,215
POS 5	0,507	0,305	0,297	0,215	0

Untuk urutan POS sama seperti urutan yang ada pada tabel 4.3

2. Pengisian POS Pertama Kedalam Tabu List

Pada tahap pengisian semut pertama ke dalam tabu list. Hasil inialisasi POS pertama setiap semut dalam langkah ini harus diisikan sebagai elemen pertama *tabu list*. Berikut pengisian semut pertama dalam tabu list :

Tabel 4.7. Tabel *Tabu List* Pada Jalur Semut

Jalur semut dalam Tabu list	POS Asal $tabu_k(i)$	POS Kunjungan 1 $tabu_k(1)$	POS Kunjungan 2 $tabu_k(2)$	POS Kunjungan 3 $tabu_k(3)$	POS Tujuan $tabu_k(j)$
Jalur semut pertama	POS 1	POS 4	-	-	POS 3
Jalur semut kedua	POS 1	POS 5	-	-	POS 3
Jalur semut ketiga	POS 1	POS 2	-	-	POS 3

3. Menghitung Nilai Probabilitas

Penyusunan rute kunjungan setiap semut ke setiap POS. Koloni semut sudah terdistribusi ke sejumlah atau setiap POS, akan mulai melakukan perjalanan dari POS pertama masing – masing sebagai POS asal dan salah satu POS – POS lainnya sebagai POS tujuan. menggunakan persamaan (3.1)

- Jalur semut pertama mengunjungi POS kunjungan 2

$$\sum_{k' \in [N-tabu_k]} [T_{ik'}]^\alpha \cdot [n_{ik'}]^\beta = 0,009471391145130233$$

$$p_{ij}^k = \frac{[T_{ij}]^\alpha \cdot [n_{ij}]^\beta}{\sum_{k' \in [N-tabu_k]} [T_{ik'}]^\alpha \cdot [n_{ik'}]^\beta}$$

- $p_{41}^1 = 0,0$
- $p_{42}^2 = 0,28693939106942506$
- $p_{43}^3 = 0,0$
- $p_{44}^4 = 0,0$
- $p_{45}^5 = 0,2280998013272018$

- Hitung nilai komulatif menggunakan persamaan (2.3).

- $kumulatif_1 = p_{41}^1 + kumulatif_{1-1} = 0,0$
- $kumulatif_2 = p_{42}^2 + kumulatif_{2-1} = 0,28693939106942506$
- $kumulatif_3 = p_{43}^3 + kumulatif_{3-1} = 0,28693939106942506$

- $kumulatif_4 = p_{44}^4 + kumulatif_{4-1} = 0,28693939106942506$
- $kumulatif_5 = p_{45}^5 + kumulatif_{5-1} = 0,5150391923966269$

- Nilai random = 0,156835819298238

Keterangan : dimana nilai random $\leq kumulatif_2$

Dipilih jalur POS pertama kunjungan 2 adalah POS 2

- Jalur semut pertama mengunjungi POS kunjungan 3

$$\sum_{k' \in [N-tabu_k]} [T_{ik'}]^\alpha \cdot [n_{ik'}]^\beta = 0,022095572871269963$$

$$p_{ij}^k = \frac{[T_{ij}]^\alpha \cdot [n_{ij}]^\beta}{\sum_{k' \in [N-tabu_k]} [T_{ik'}]^\alpha \cdot [n_{ik'}]^\beta}$$

- $p_{21}^1 = 0,0$
- $p_{22}^2 = 0,0$
- $p_{23}^3 = 0,0$
- $p_{24}^4 = 0,0$
- $p_{25}^5 = 0,13813156154166906$

- Hitung nilai komulatif menggunakan persamaan (2.3).

- $kumulatif_1 = p_{21}^1 + kumulatif_{1-1} = 0,0$
- $kumulatif_2 = p_{22}^2 + kumulatif_{2-1} = 0,0$
- $kumulatif_3 = p_{23}^3 + kumulatif_{3-1} = 0,0$
- $kumulatif_4 = p_{24}^4 + kumulatif_{4-1} = 0,0$
- $kumulatif_5 = p_{25}^5 + kumulatif_{5-1} = 0,13813156154166906$

Nilai random = 0,05746576804113842

Keterangan : dimana nilai random $\leq kumulatif_5$

Dipilih jalur POS pertama kunjungan 3 adalah POS 5

Sampai dengan jalur semut kedua dan ketiga dihitung dan semua semut mengunjungi semua POS, langkah berikutnya adalah mengisi jalur tersebut kedalam *tabu list* siklus 1. Berikut adalah *tabu list* siklus 1.

Tabel 4.8. Tabel *Tabu List* Siklus 1

Jalur semut dalam <i>Tabu list</i>	POS Asal $tabu_k(i)$	POS Kunjungan 1 $tabu_k(1)$	POS Kunjungan 2 $tabu_k(2)$	POS Kunjungan 3 $tabu_k(3)$	POS Tujuan $tabu_k(j)$
Jalur semut pertama	POS 1	POS 4	POS 2	POS 5	POS 3
Jalur semut kedua	POS 1	POS 5	POS 4	POS 2	POS 3
Jalur semut ketiga	POS 1	POS 2	POS 4	POS 5	POS 3

4. Perhitungan Panjang Rute dan Waktu Tempuh Setiap Jalur Semut

Perhitungan panjang rute dan waktu tempuh ini berdasarkan Tabu List yang dilakukan setelah satu siklus diselesaikan dimana perhitungan ini masing – masing menggunakan persamaan (3.4). Jarak pada tabu indeks POS asal ditambah dengan jumlah tabu indeks POS kunjungan ke dua dan hasilnya ditambah dengan indeks POS selanjutnya sampai dengan tabu indeks POS tujuan tercapai dan sebaliknya pada perhitungan waktu tempuh. Setelah hitung panjang rute dilakukan pencarian rute terpendek setiap jalur semut L_k , akan didapat nilai minimal panjang rute tertutup setiap siklus (L_{minNC}) dan nilai minimal panjang rute tertutup secara keseluruhan (L_{min}).

- Jalur Semut pertama Angkutan 1, dengan rute POS :

POS 1 => POS 4 => POS 2 => POS 5 => POS 3

$$L_k = d_{tabu_k(n),tabu_k(i)} + \sum_{s=1}^{n-1} d_{tabu_k(s),tabu_k(s+1)}$$

$$L_k = tabu_k(1) , tabu_k(i) + (tabu_k(1) , tabu_k(2) , tabu_k(2) , tabu_k(3) + (tabu_k(3) , tabu_k(j))$$

$$L_k = POS 4 => POS 1 + (POS 4 => POS 2 + POS 2 => POS 5 + POS 5 => POS 3)$$

$$L_k = 4,270 \text{ Km} + (3,680 \text{ Km} + 3,270 \text{ Km} + 3,360 \text{ Km}) = 14.579999999999998 \text{ Km}$$

Dengan cara yang sama perhitungan jarak antar POS dan waktu tempuh dihitung, sampai dengan semua dihitung yaitu jalur kedua dan ketiga pada semut.

Pada perhitungan *update pheromon* atau perubahan nilai intensitas jejak kaki semut ini dengan menggunakan persamaan (3.8), jika $\Delta\tau_{ij}$ dengan L_k yang digunakan adalah jumlah rata-rata dari panjang setiap jalur semut. Dimana batas inisialisasi *Pheromone* awal adalah $0,01 \geq T_{ij} \leq 1,00$, pada perhitungan nilai intensitas jejak semut (τ_{ij}) adalah nilai *Pheromone* dan perubahannya setiap kunjungan antar POS dengan persamaan (3.5) kemudian pada (ΔT_{ij}) jalur yang merupakan dari POS asal (i) ke POS tujuan (j) maka akan menggunakan persamaan (3.6) dan sebaliknya, jika bukan jalur dari POS asal (i) ke POS tujuan (j) menggunakan persamaan (3.7) atau sama dengan nol. Berikut perhitungannya :

- Jalur Semut pertama Angkutan 1, dengan rute POS : POS 1 => POS 4 => POS 2 => POS 5 => POS 3

Dengan total jarak 14,579999999999998 Km

$$T_{ij}(\text{baru}) = (1 - \rho) T_{ij} + \Delta T_{ij}$$

$$T_{12} = (1 - 0,5) 0.01 + 0 = 0,005$$

$$T_{13} = (1 - 0,5) 0.01 + 0 = 0,005$$

$$T_{14} = (1 - 0,5) 0.01 + 1/14,579999999999998 = 0,685871056241$$

$$T_{15} = (1 - 0,5) 0,01 + 0 = 0,005$$

Dengan cara yang sama dilakukan perhitungan *Update Pheromone* pada jalur semut kedua dan jalur semut ketiga, maka didapat hasil *update pheromone* baru jika diperlukan dan siklus ditambah dengan 1.

5. Pengosongan Tabu List

Kosongkan semua *tabu list* dan ulangi ke langkah 4.1.2. sampai dengan iterasi = NC_{max} baru mencari jalur terbaik dari setiap iterasi.

6. Tampilan Rute

Pada pembahasan ini akan menampilkan hasil rute optimal dari POS 1 Menuju POS 3 dengan POS yang dikunjungi meliputi POS 2 – POS 4 – POS 5. Tampil rute sebagai berikut



Gambar 4.1. Tampilan Hasil Rute di *Google Map*
Hasil rute yang didapat adalah :

POS awal pada posisi POS 1 (PT. Semangat Persada Tour & Travel) mengunjungi POS kedua yaitu POS 5 (CV. Jaya Bersama) mengunjungi POS ketiga yaitu POS 4 (PT. Rainbow Tour & Travel) mengunjungi POS keempat POS 2 (CV. Jaya Bersama) setelah semua dikunjungi rute tujuan akhir di kunjungi yaitu POS 3 (PT. Granindo Duta Selaras) dimana POS C tersebut adalah POS kunjungan terakhir.

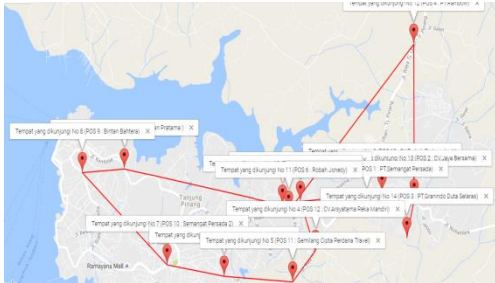
V. HASIL UJI COBA

Berdasarkan uji coba yang telah dilakukan oleh peneliti, maka peneliti menemukan parameter optimum yang dapat menentukan hasil rute, jarak serta waktu tempuh yang optimal dengan nilai alpha (α) dan beta (β) adalah 1,0 dan nilai rho (ρ) adalah 0.1 dan uji coba saat ini akan dilakukan menggunakan jumlah siklus optimum dengan jumlah siklus 200, 500, 800 dan 900

Tabel 5.1. Tabel Hasil Kesimpulan Parameter dan Siklus Optimum

No	(α)	(β)	(ρ)	(NC_{max}) Siklus	Rute POS	Jarak (Km)	Waktu Tempuh (Jam)
1	1,0	1,0	0,1	200	1-13-5-12-11-14-10-9-8-7-6-4-2-3	27,824	1,289
2	1,0	1,0	0,1	500	1-13-5-12-11-14-10-9-8-7-6-4-2-3	27,824	1,289
3	1,0	1,0	0,1	800	1-13-6-7-8-9-10-14-11-12-5-4-2-3	27,824	1,289
4	1,0	1,0	0,1	900	1-13-6-7-8-9-10-14-11-12-5-4-2-3	27,824	1,289

Maka tampil hasil rute optimal dalam bentuk *Google Map*.



Gambar 5.1. Hasil Rute Optimal

VI. KESIMPULAN DAN SARAN

Berdasarkan tujuan dan manfaat dari penelitian ini adalah dengan menerapkan algoritma semut ini terlebih dahulu peneliti akan mencari nilai parameter optimum dengan alpha (α) dan beta (β) = 1,0 kemudian nilai rho (ρ) = 0,1 dengan jumlah siklus optimum didapat 200, 500, 800 dan 900, dari hasil percobaan yang dilakukan peneliti mendapatkan rute optimal dengan kunjungan asal dari POS 1 menuju kunjungan

akhir POS 3 dengan hasil rute optimal POS 1 => POS 13 => POS 6 => POS 7 => POS 8 => POS 9 => POS 10 => POS 14 => POS 11 => POS 12 => POS 5 => Pos 4 => POS 2 => POS 3 dengan jarak 27,624 Km dan waktu tempuh 1,289 Jam.

Saran yang diharapkan oleh peneliti adalah dengan membandingkan dua metode optimasi rute dan menambahkan data-data parameter pendukung sehingga informasi yang diberikan oleh sistem lebih rinci terhadap user / penggunanya.

DAFTAR PUSTAKA

- Fernandez, A., 2012, Pembangunan Aplikasi Penyusunan Jadwal Kuliah Menggunakan Algoritma Semut, *Jurusan Teknik Elektro*, Universitas Diponegoro, Semarang.
- Fitriyani, D. R., 2015, Analisa Pencarian Jalur Terpendek Ke Penginapan Di Kota Batam Dengan Menggunakan Algoritma Semut, *Jurusan Teknik Informatika*, Universitas Maritim Raja Ali Haji, Kepulauan Riau.
- Laeli, S., 2014, *Analisis Cluster dengan Average Linkage Method dan Ward's Method untuk Data Responden Nasabah Asuransi Jiwa Unit Link*, *Jurusan Pendidikan Matematika*, Universitas Negeri Yogyakarta, Yogyakarta.
- Lestari, H. P., Sari, E. R., 2013, Penerapan Algoritma Koloni Semut Untuk Optimasi Rute Distribusi Pengangkutan Sampah Di Kota Yogyakarta, *Jurusan Pendidikan Matematika*, Universitas Negeri Yogyakarta, Yogyakarta.
- Mursids, (2009), Algoritma Koloni Semut (ACO), <http://mursids.blogspot.co.id/2009/12/algoritma-koloni-semut-aco.html>, 28 Desember 2009
- Mutakhirah, I., Indrato, dan Hidayat, T. "Pencarian Jalur Terpendek Menggunakan Algoritma Semut" *Seminar Nasional Aplikasi Teknologi Informasi*, Laboratorium Pemrograman dan Informatika Teori – UII Yogyakarta, 2007.
- Triandi, B., 2012, Algoritma Penemuan Jalur Terpendek Dengan Algoritma Ant Colony, *Jurusan Teknik Informatika*, STMIK Potensi Utama, Medan.
- Yuwono, B., Aribowo, A. S., Wardoyo, S. B., 2009, Implementasi Algoritma Koloni Semut Pada Proses Pencarian Jalur Terpendek Jalan Protokol Di Kota Yogyakarta, *Jurusan Teknik Informatika*, UPN Yogyakarta, Yogyakarta.