

# Ant Colony Optimization

Karjono<sup>#1</sup>, Moedjiono<sup>#2</sup>, Denni Kurniawan<sup>#3</sup>

<sup>#</sup>Program Studi Magister Ilmu Komputer, Program Pascasarjana, Universitas Budi Luhur

Jalan Ciledug Raya, Petukangan Utara, Jakarta Selatan, DKI Jakarta 12260 (021) 5853753

<sup>1</sup>aryo.y2k@gmail.com

<sup>2</sup>moedjiono@budiluhur.ac.id

<sup>3</sup>denni.kurniawan@budiluhur.ac.id

**Abstraksi** — Metode Optimasi Abstrak lebih dikenal salah satunya adalah metode pendekatan. Metode pendekatan tidak menjamin solusi optimal. Namun, meskipun metode ini tidak menjamin hasil yang optimal, tetapi hasil penyelesaian umumnya cukup baik. Metaheuristik sebenarnya adalah pendekatan yang didasarkan pada metode heuristik. Jadi tidak mengherankan bahwa metode heuristik sering terintegrasi dalam metode metaheuristik. *Ant Colony Optimization* (ACO) adalah salah satu contoh dari metode metaheuristik yang dapat digunakan untuk masalah *Traveling Salesman Problem*.

**Kata kunci** : *Optimization, Ant Colony Optimization, Metaheuristik*

**Abstract** — Optimization methods are often known one of them is the method of approach. Approximation method does not ensure that the solution is optimal. However, although this method does not guarantee optimal results, but generally pretty good settlement outcomes. Metaheuristic in fact is the approach that is based on heuristic methods. So no wonder that the heuristic methods are often integrated within metaheuristic method. *Ant Colony Optimization* (ACO) is one example of a metaheuristic method that can be used for problem *Traveling Salesman Problem*.

**Keywords** — *Optimization, Ant Colony Optimization, Metaheuristik*.

## I. PENDAHULUAN

*Ant-Colony Optimization* termasuk dalam kelompok *Swarm Intelligence*, yang merupakan salah satu jenis pengembangan paradigma yang digunakan untuk menyelesaikan masalah optimasi dimana inspirasi yang digunakan untuk memecahkan masalah tersebut berasal dari perilaku kumpulan atau kawanan (*swarm*) serangga. ACO biasanya digunakan untuk menyelesaikan *discrete optimization problems* dan persoalan yang kompleks dimana terdapat banyak variabel. Hasil yang diperoleh dengan menggunakan ACO, walaupun tidak optimal namun mendekati optimal. ACO sudah diterapkan di berbagai masalah seperti VRP, penjadwalan proyek dengan sumberdaya terbatas, data mining, penjadwalan pekerjaan (*job scheduling*)

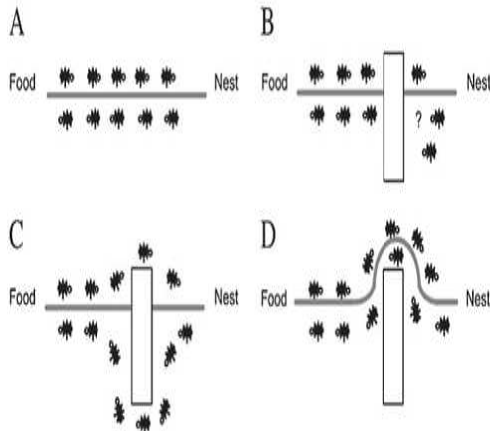
dan beberapa masalah kombinatorial yang lain. Dalam ACO, setiap semut dalam kawanan yang berjalan akan meninggalkan *pheromone* (semacam zat kimia) pada jalur yang dilaluinya. Pheromone ini menjadi semacam sinyal bagi sesama semut. Jalur yang pendek akan menyisakan sinyal yang lebih kuat. Semut berikutnya, pada saat memutuskan jalur mana yang harus dipilih, biasanya akan cenderung memilih untuk mengikuti jalur dengan sinyal yang paling kuat, sehingga jalur terpendek akan ditemui karena lebih banyak semut yang akan melewati jalur tersebut. Semakin banyak semut yang lewat suatu jalur, semakin kuat sinyal di jalur itu. Hal ini dapat dilihat pada Gambar 1, dimana ditunjukkan bagaimana jalur yang dilalui oleh semut pada saat mencari makan dari sarang sampai ke sumber makanan, penjadwalan merupakan suatu proses pengorganisasian waktu untuk mendapatkan waktu yang efektif dan optimal. Sebuah jadwal merupakan sekumpulan dari pertemuan pada waktu tertentu. Sebuah pertemuan adalah kombinasi dari sumber daya (ruangan, orang, dan lainnya), dimana beberapa diantaranya ditentukan oleh masalah dan beberapa mungkin dialokasikan sebagai bagian dari pemecahan [1]. Algoritma *Ant* merupakan salah satu dari teknik yang paling sukses dalam hal penjadwalan menurut [2] dan [3], terutama diaplikasikan dalam TSP (*travelling salesman problem*) Generasi pertama program masalah penjadwalan dengan komputer dikembangkan pada awal tahun 1960 yang berusaha mengurangi pekerjaan administratif [4], [5].

## II. DASAR TEORI

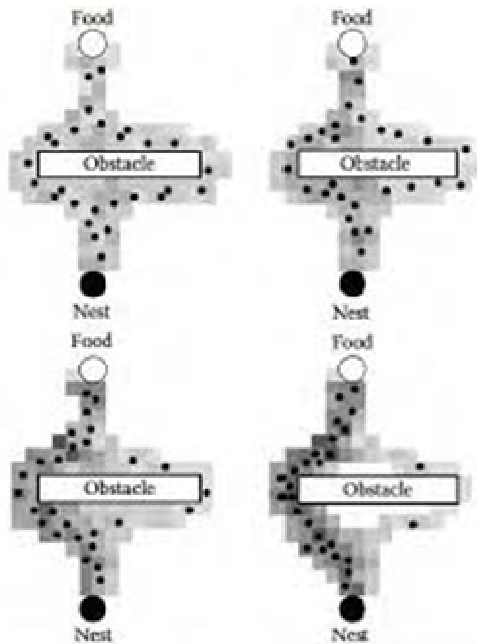
### A. Algoritma Ant Colony

Algoritma semut diperkenalkan oleh Moysen dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo. Algoritma semut adalah *bioinspired metaheuristic*, mempunyai sekelompok khusus yang berusaha menyamai karakteristik kelakuan dari serangga sosial, yaitu koloni semut. Kelakuan dari tiap pelaku dalam meniru kelakuan dari semut hidup dan bagaimana mereka berinteraksi satu dengan lainnya agar dapat menemukan sumber makanan dan membawanya ke koloni mereka dengan efisien. Selama berjalan tiap semut mengeluarkan feromon, dimana semut lainnya sensitif dengan feromon tersebut sehingga memberikan harapan untuk mengikuti jejaknya. Lebih atau kurang intensitasnya tergantung pada konsentrasi dari feromon. Setelah beberapa waktu, jalur terpendek akan lebih sering diikuti dan feromonnya menjadi

jenuhsemut diperkenalkan oleh Moyson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo.



Gbr. 1 Contoh klasik dari pembangunan jejak feromon dalam mencari jalan yang lebih pendek.



Gbr. 2 Perubahan konsentrasi feromon

Jadi cara kerja Algoritma *Ant* adalah sebagai berikut:

1. Pada awalnya, semut berkeliling secara acak.
2. Ketika semut-semut menemukan jalur yang berbeda misalnya sampai pada persimpangan, mereka akan mulai menentukan arah jalan secara acak
3. Sebagian semut memilih berjalan ke atas dan sebagian lagi akan memilih berjalan ke bawah
4. Ketika menemukan makanan mereka kembali ke koloninya sambil memberikan tanda dengan jejak feromon.
5. Karena jalur yang ditempuh lewat jalur bawah lebih pendek, maka semut yang bawah akan tiba lebih dulu dengan asumsi kecepatan semua semut adalah sama

6. Feromon yang ditinggalkan oleh semut di jalur yang lebih pendek aromanya akan lebih kuat dibandingkan feromon di jalur yang lebih panjang
7. Semut-semut lain akan lebih tertarik mengikuti jalur bawah karena aroma feromon lebih kuat [6].

Proses dalam ACO bisa dijelaskan sebagai berikut. Misalkan ada  $N$  semut dalam satu koloni. Semut-semut itu akan memulai gerakan dari sarang mereka menuju tujuan akhir melalui beberapa simpul dan berakhir pada simpul tujuan di akhir setiap siklus atau iterasi. Kalau semua semut sudah menyelesaikan lintasannya, jumlah pheromone pada lintasan terbaik secara global akan diperbarui. Lintasan global terbaik artinya terbaik diantara semua semut. Pada awal proses, yaitu pada iterasi 1, semua ruas dari simpul awal akan diberi jumlah pheromone yang sama. Sehingga pada iterasi 1, semua semut akan mulai dari simpul awal dan berakhir pada simpul tujuan dengan cara memilih simpul-simpul antara secara random. Proses optimasi berakhir jika jumlah iterasi maksimum sudah tercapai atau tidak ada lagi solusi yang lebih baik yang bisa didapat dalam beberapa iterasi yang berturutan.

Semut menggunakan lingkungannya sebagai media komunikasi. Mereka bertukar informasi secara tidak langsung melalui pheromone secara mendetail seperti status kerja, dll. Informasi yang ditukar memiliki ruang lingkup lokal, dimana hanya seekor semut yang terletak di tempat pheromone itu berada. Sistem ini disebut "*Stigmergy*" dan terjadi di banyak hewan yang hidup bersosial masyarakat (hal itu telah dipelajari dalam kasus pembangunan pilar dalam sarang rayap). Mekanisme untuk menyelesaikan masalah yang kompleks untuk ditangani oleh satu semut adalah contoh yang baik dari suatu sistem organisme. Sistem ini didasarkan pada feedback positif (menarik feromon semut lain yang akan memperkuat sendiri) dan negatif (disipasi dari rute oleh sistem mencegah penguapan dari labrakan). Secara teori, jika jumlah feromon tetap sama dari waktu ke waktu pada semua sisi, tidak ada rute yang akan dipilih. Namun, karena feedback, sedikit variasi pada sisi akan diperkuat dan dengan demikian memungkinkan pilihan sisi tersebut.

Algoritma akan bergerak dari keadaan yang tidak stabil di mana tidak ada sisi yang lebih kuat daripada yang lain, untuk ke yang lebih stabil di mana jalur terdiri dari sisi paling kuat. Algoritma Ant Colony Optimization merupakan teknik probabilistik untuk menjawab masalah komputasi yang bisa dikurangi dengan menemukan jalur yang baik dengan graf. ACO pertama kali dikembangkan oleh Marco Dorigo pada tahun 1991.

Sesuai dengan nama algoritmanya ACO di inspirasi oleh koloni semut karena tingkah laku semut yang menarik ketika mencari makanan. Semut-semut menemukan jarak terpendek antara sarang semut dan sumber makanannya. Ketika berjalan dari sumber makanan menuju sarang mereka, semut memberikan tanda dengan zat feromon sehingga akan tercipta jalur feromon. Feromon adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis, individu lain kelompok, dan untuk membantu proses reproduksi.

Berbeda dengan hormon, feromon menyebar ke luar tubuh dan hanya dapat mempengaruhi dan dikenali oleh individu lain yang sejenis, proses peninggalan feromon ini dikenal sebagai stigmergy. Semut dapat mencium feromon dan ketika mereka memilih jalur mereka, mereka cenderung memilih jalur yang ditandai oleh feromon dengan konsentrasi yang tinggi. Apabila semut telah menemukan jalur yang terpendek maka semut-semut akan terus melalui jalur tersebut. Jalur lain yang ditandai oleh feromon lama akan memudar atau menguap, seiring berjalannya waktu. Jalur-jalur yang pendek akan mempunyai ketebalan feromon dengan probabilitik yang tinggi dan membuat jalur tersebut akan dipilih dan jalur yang panjang akan ditinggalkan. Jalur feromon membuat semut dapat menemukan jalan kembali ke sumber makanan atau sarang mereka.

Formula :

- **Temporary** (i,j)  

$$v = \max\{[\tau(i,j)] \cdot [\eta(i,j)^\beta]\} \dots\dots\dots(1)$$

- **Pemilihan Jalur :**

Seekor semut akan berjalan dari simpul i menuju simpul j dengan probabilitas .....(2)

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

dimana  $\tau_{i,j}$  = jumlah pheromone pada sisi  $i,j$   
 $\alpha$  = parameter pengontrol pengaruh  $\tau_{i,j}$   
 $\eta_{i,j}$  = desirability sisi  $i,j$  (biasanya  $1 / d_{i,j}$ , dimana d adalah jarak)  
 $\beta$  = parameter pengontrol pengaruh  $\eta_{i,j}$

- **Penambahan dan Penguapan Pheromone :**

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j} \dots\dots\dots(3)$$

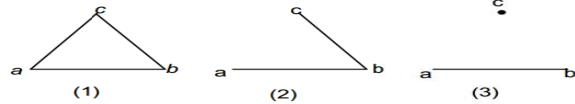
dimana  $\tau_{i,j}$  = jumlah pheromone pada sisi  $i,j$   $\rho$  = tingkat penguapan pheromone dan  $\Delta\tau_{i,j}$  = jumlah pheromone dihasilkan

- **Update Pheromone**

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

**B. Graf Hamilton**

Lintasan Hamilton ialah lintasan yang melalui tiap titik di dalam graf tepat satu kali. Sirkuit Hamilton ialah sirkuit yang melalui tiap titik di dalam graf tepat satu kali, kecuali titik asal (sekaligus titik akhir) yang dilalui dua kali. Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi - Hamilton.



Gbr. 3 Graf Hamilton(1), Graf Semi-Hamilton(2), Graf Bukan Hamilton

**C. Penyelesaian TSP Menggunakan Algoritma Semut**

TSP adalah salah satu teka-teki optimisasi yang cukup terkenal di kalangan peneliti dan pecinta matematika selama bertahun-tahun. Mereka berlomba untuk mencari penyelesaian kasus TSP dengan tekniknya masing-masing. Teknik yang cukup terkenal adalah simulated annealing, genetic algorithm, and ant colony optimization (algoritma semut). Algoritma semut atau Ant Colony Optimization telah digunakan untuk mencari lintasan optimal pada Travelling Salesman Problem (TSP). Pada simulasi algoritma semut, diperlukan tiga tabel besar (dengan dimensi  $n \times n$  dimana  $n$  adalah banyaknya kota) untuk mencari lintasan optimal.

Tabel pertama adalah tabel jarak (distance array), untuk menghitung seluruh jarak dari kota yang satu ke kota lainnya. Tabel kedua adalah tabel pheromon (pheromone array), untuk menyimpan kadar pheromon pada jalur antara seluruh kota. Tabel ketiga adalah tabel delta pheromon (delta pheromone array), untuk menyimpan sementara pheromon untuk ditambahkan ke tabel pheromon pada akhir iterasi. Tabel delta pheromon digunakan agar semua semut mengetahui hasil dari iterasi sebelumnya.

Gbr. 4 Ilustrasi TSP

**III.METODOLOGI PENELITIAN**

Pemilihan penelitian dengan menggunakan metode membangun sistem model pendekatan RAD (Rapid Application Development) atau rapid prototyping karena melihat aplikasi yang dikembangkan adalah aplikasi sederhana dan tidak membutuhkan waktu yang lama.

1. Rencana Kebutuhan (*Requirement Planning*)

Pada tahap ini, pengguna dan analis melakukan semacam pertemuan untuk melakukan identifikasi tujuan dari aplikasi atau sistem dan melakukan identifikasi kebutuhan informasi untuk mencapai tujuan. Pada tahap ini hal terpenting adalah adanya keterlibatan dari kedua belah pihak, bukan hanya sekedar persetujuan akan proposal yang sudah dibuat. Untuk lebih jauh lagi, keterlibatan pengguna bukan hanya dari satu tingkatan pada suatu organisasi, melainkan beberapa tingkatan organisasi sehingga informasi yang dibutuhkan untuk masing-masing pengguna dapat terpenuhi dengan baik.

2. Proses Desain (*Design Workshop*)

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara pengguna dan analis. Untuk tahap ini maka keaktifan pengguna yang terlibat sangat menentukan untuk mencapai tujuan, karena pengguna bisa langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain. Biasanya pengguna dan analis berkumpul menjadi satu dan duduk di meja melingkar dimana masing-masing orang bisa melihat satu dengan yang lain tanpa ada halangan. Apabila memungkinkan, maka masing-masing pengguna diberikan satu komputer yang terhubung satu dengan yang lain, sehingga masing-masing bisa melihat desain yang dibuat dan langsung memberikan komentar. Pada selang waktu tersebut, pengguna bisa memberikan tanggapan akan sistem yang sudah dikembangkan untuk selanjutnya dilakukan perbaikan-perbaikan. Dengan demikian proses pengembangan suatu sistem membutuhkan waktu yang cepat.

3. Implementasi (*Implementation*)

Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh pengguna dan analis, maka pada tahap ini programmer mengembangkan desain menjadi suatu program. Setelah program selesai baik itu sebagian maupun secara keseluruhan, maka dilakukan proses pengujian terhadap program tersebut apakah terdapat kesalahan atau tidak sebelum diaplikasikan pada suatu organisasi. Pada saat ini maka pengguna bisa memberikan tanggapan akan sistem yang sudah dibuat serta persetujuan mengenai sistem tersebut. Adapun hal terpenting adalah bahwa keterlibatan pengguna sangat diperlukan supaya system yang di kembangkan dapat memberikan kepuasan kepada pengguna dan di samping itu sistem yang lama tidak perlu dijalankan secara paralel dengan sistem yang baru.

4. Tahapan keseluruhan

Dengan berdasarkan pada tahapan di atas maka proses utama pengembangan suatu sistem dengan menggunakan metode RAD adalah sebagai berikut:

- 1) Pengembang membuat prototype berdasarkan kebutuhan-kebutuhan yang sudah didefinisikan sebelumnya.
- 2) Desainer melakukan penilaian terhadap prototype.
- 3) Pengguna melakukan uji coba pada prototype dan memberikan masukan mengenai kebutuhan-kebutuhan yang kurang.
- 4) Pengguna dan pengembang melakukan pertemuan untuk memberikan penilaian terhadap produk secara bersama-

sama, menyesuaikan kebutuhan serta memberikan komentar apabila diperlukan perubahan.

- 5) Semua kebutuhan akan sistem dan perubahan-perubahan yang terjadi dilakukan proses "timeboxed" dengan mempunyai 2 kemungkinan :

- Perubahan yang tidak dapat ditampung seperti yang sudah direncanakan harus dihilangkan.
- Jika diperlukan, kebutuhan-kebutuhanyang bersifat sekunder ditiadakan [7].

b. HASIL DAN PEMBAHASAN

A. Perhitungan Jarak Rute Pengiriman Barang

Data penjualan dari PT XYZ berisi secara umum nama-nama barang yang dipesan oleh para konsumen, lokasi dari tiap-tiap konsumen tersebut serta kuantitas barang yang dipesan. Data ini merupakan data yang digunakan sebagai acuan dalam melakukan aktivitas delivery order. Data penjualan untuk semua barang dapat dilihat pada tabel I dibawah ini.

TABEL I  
HASIL PENJUALAN

NO PO	KODE PELANGGA	TGL ORD	PART NUMBEI	PART NAME	LOCATION	ORD
PO0001	1711	15012015	PN - 1411-1010	Extreme Puseda CCTV Kamera 3001 - Putih	JL RADIO DALAM	5
PO0001	1711	15012015	PN - 1411-7540	Hardlenz Kamera CCTV Outdoor CMOS 800TVL 36 IR HZ-635 C	JL RADIO DALAM	7
PO0001	1711	15012015	PN - 1411-1510	Extreme Puseda CCTV Kamera 3139	JL BINTARO RAYA	4
PO0001	1711	15012015	PN - 1411-1012	Extreme Puseda CCTV Kamera 9002 - Abu-abu	JLSWADARMA RAY	3
PO0001	1711	15012015	PN - 1411-2435	IPeKAM DVR KIT GOLD 4CH - CCTV - Kamera Keamanan - ALL	JLSWADARMA RAY	3
PO0001	1711	15012015	PN - 1411-6170	IPeKAM DVR KIT SILVER 4CH - CCTV - Kamera Keamanan - ALL	JLSWADARMA RAY	3
PO0001	1711	15012015	PN - 1411-1021	Hardlenz Dome Kamera Indoor CCTV SONY 1000TVL 720P - Pu	JL YUSUF	6
PO0001	1711	15012015	PN - 1411-2514	Hardlenz Kamera CCTV Outdoor SONY 1000TVL 72 IR HZ721000	JL YUSUF	4
PO0001	1711	15012015	PN - 1411-1017	Hardlenz HZ800CD Kamera CCTV Dome CMOS 800TVL 24 IR - P	PLAZA SENAYAN	4
PO0001	1711	15012015	PN - 1411-3531	Hardlenz Kamera CCTV Outdoor SONY 700TVL 36 IR HZ-635 SO	PLAZA SENAYAN	3
PO0001	1711	15012015	PN - 1411-1030	Hardlenz J663 Dome Kamera Indoor CCTV SuperHAD CCD II SQ	JL PROF DR SATRIO	2
PO0001	1711	15012015	PN - 1411-6130	Hardlenz Kamera CCTV Outdoor CMOS 800TVL 36 IR HZ800CC	JL PROF DR SATRIO	10
PO0001	1711	15012015	PN - 1411-1718	Hardlenz Kamera CCTV Dome Glass SONY 700TVL 18 IR J773	JL CASABLANCA	3
PO0001	1711	15012015	PN - 1411-2134	Hardlenz Kamera CCTV Dome SONY 1000TVL 36 IR HZ1385HR3	JL RASAMALA RAYA	5
PO0001	1711	15012015	PN - 1411-3540	Hardlenz Kamera CCTV Dome SONY 700TVL 24 IR LS-DBSHE	JL BANGKA RAYA	5
PO0001	1711	15012015	PN - 1411-6540	Hardlenz Kamera CCTV Outdoor SONY 700TVL 24 IR LS-24SHE	JL BANGKA RAYA	6

B. Jarak Antar Lokasi

Dalam pengaplikasian *software* pencarian rute terpendek yang dibuat berdasarkan algoritma ACO maka diperlukan suatu data jarak terpendek antar lokasi yang terdapat pada gudang sebagai acuan utama dalam pencarian rute terpendek tersebut. Data jarak untuk semua titik dapat dilihat pada tabel II dibawah ini.

TABEL II  
JARAK ANTAR LOKASI DALAM SATUAN KILO METER

	GUDANG	LOCA	LOCB	LOCC	LOCD	LOCE	LOCF	LOGG	LOCH	LOCI
GUDANG	0	19.8	22.0	13.6	26.1	19.3	25.6	25.8	28.7	26.2
LOCA	19.8	0.00	2.20	6.20	6.30	0.50	5.80	6.00	8.90	6.40
LOCB	22.0	2.20	0.00	8.40	4.10	2.70	3.60	3.80	6.70	4.20
LOCC	13.6	6.20	8.40	0.00	12.50	5.70	12.00	12.20	15.10	12.80
LOCD	26.1	6.30	4.10	12.50	0.00	6.80	0.50	0.30	2.60	0.10
LOCE	19.3	0.50	2.70	5.70	6.80	0.00	6.30	6.50	9.40	6.90
LOCF	25.6	5.80	3.60	12.00	0.50	6.30	0.00	0.20	3.10	0.60
LOGG	25.8	6.00	3.80	12.20	0.30	6.50	0.20	0.00	2.80	0.40
LOCH	28.7	8.90	6.70	15.10	2.60	9.40	3.10	2.90	0.00	2.50
LOCI	26.2	6.40	4.20	12.80	0.10	6.90	0.60	0.40	2.50	0.00

C. Perhitungan Jarak Rute Pengiriman Barang

Sebelum memasuki perhitungan pada tahap satu dalam perhitungan algoritma ACO maka terlebih dahulu dilakukan perhitungan awal untuk menghitung invers jarak. Nilai invers untuk semua titik dapat dilihat pada tabel III dibawah ini.

TABEL III  
INVERS JARAK

	GUDANG	LOCA	LOCB	LOCC	LOCD	LOCE	LOCF	LOGG	LOCH	LOCI
GUDANG	0	0.05151	0.04545	0.07353	0.03831	0.05181	0.03916	0.03876	0.03494	0.03817
LOCA	0.05151	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
LOCB	0.04545	0.45455	0.00000	0.11985	0.24399	0.37037	0.27778	0.26516	0.14925	0.23810
LOCC	0.07353	0.16129	0.11985	0.00000	0.08000	0.17544	0.08333	0.08157	0.06623	0.07937
LOCD	0.03831	0.15873	0.24399	0.08000	0.00000	0.14706	2.00000	3.33333	0.38462	10.00000
LOCE	0.05181	2.00000	0.37037	0.17544	0.14706	0.00000	0.15873	0.15385	0.10638	0.14493
LOCF	0.03916	0.17441	0.27778	0.08333	2.00000	0.15873	0.00000	5.00000	0.32258	1.66667
LOGG	0.03876	0.16667	0.26516	0.08157	3.33333	0.15385	5.00000	0.00000	0.34493	2.50000
LOCH	0.03494	0.11236	0.14925	0.06623	0.38462	0.10638	0.32258	0.34493	0.00000	0.40000
LOCI	0.03817	0.15625	0.23810	0.07937	10.00000	0.14493	1.66667	2.50000	0.40000	0.00000

Nilai dari semua pheromone ( $\tau$ ) pada awal perhitungan ditetapkan dengan angka awal yang sangat kecil. Pada contoh perhitungan penelitian ini nilai pheromone  $\tau$  awal menggunakan nilai awal sebesar 0,0001. Penetapan nilai pheromone awal dimaksudkan agar tiap-tiap ruas memiliki nilai ketertarikan untuk dikunjungi oleh tiap-tiap semut. Nilai pheromone untuk semua titik dapat dilihat pada tabel IV dibawah ini.

TABEL IV  
PHEROMONE AWAL

	GUDANG	LOCA	LOCB	LOCC	LOCD	LOCE	LOCF	LOGG	LOCH	LOCI
GUDANG	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCA	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCB	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCC	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCD	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCE	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCF	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOGG	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCH	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
LOCI	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

Dalam pemilihan titik selanjutnya yang dituju, pertama-tama dilakukan penetapan dari nilai  $\beta \geq 0$  adalah parameter perhitungan untuk mendapatkan nilai yang optimal dalam ACO, untuk mempermudah perhitungan diambil nilai  $\beta = 2$ . Selanjutnya dilakukan perhitungan untuk mendapatkan nilai *temporary* (i,j) berdasarkan persamaan (1) serta nilai probabilitas berdasarkan persamaan (2) dari titik awal DEPOT (i) ke titik selanjutnya yang belum dilalui (j). Nilai *temporary* digunakan untuk menentukan titik-titik yang akan dituju selanjutnya. Hasil perhitungan *temporary* dan probabilitas dari titik awal GUDANG dapat dilihat pada tabel V dibawah ini.

TABEL V

HASIL PERHITUNGAN TEMPORARY DAN PROBABILITAS DARI TITIK AWAL GUDANG

	GUDANG	LOCA	LOCB	LOCC	LOCD	LOCE	LOCF	LOGG	LOCH	LOCI
nilai peronom	0	0.0285	0.0287	0.0541	0.0147	0.0288	0.0163	0.0150	0.0121	0.0146
probabilitas	0	0.1283	0.1040	0.2720	0.0739	0.1351	0.0768	0.0756	0.0611	0.0733
akumulatif probabilitas	0	0.1283	0.2283	0.5043	0.5782	0.7133	0.7900	0.8656	0.9267	1.0000

Untuk memilih persamaan yang tepat sebagai acuan dalam pemilihan lokasi selanjutnya maka perlu dibangkitkan suatu bilangan random (q) antara 0 sampai 1 serta menetapkan suatu bilangan pembatas (q0) antara 0 sampai 1. Pada perhitungan ini ditetapkan nilai q0 sebesar 0,9 serta bilangan random yang dibangkitkan memiliki nilai q sebesar 0,1 yang artinya semut melakukan proses eksploitasi dengan probabilitas 90% dan proses  $q \leq q0$  eksplorasi 10%. Karena , maka penentuan lokasi yang akan dituju berdasarkan persamaan (1), yaitu dengan melihat hasil *temporary* yang paling besar. Sehingga lokasi yang terpilih adalah lokasi **LOCC**.

D. Tahap pembaharuan pheromone Lokal

Setelah pengirim berpindah menuju lokasi selanjutnya maka tahap  $\tau$  selanjutnya adalah melakukan pembaharuan pheromone ( $\tau$ ) secara lokal dengan menggunakan persamaan (3).Persamaan dari pembaharuan  $\tau$  pheromone ( $\tau$ ) lokal, hasil keseluruhan pembaharuan dapat dilihat pada tabel VI dibawah ini.

TABEL VI

NILAI PHEROMONE SETELAH TAHAP MENGALAMI PEMBAHARUAN LOKAL

	GUDANG	LOCA	LOCB	LOCC	LOCD	LOCE	LOCF	LOGG	LOCH	LOCI
GUDANG	0.00010	0.00010	0.00010	0.00083	0.00010	0.00010	0.00010	0.00010	0.00044	0.00010
LOCA	0.00010	0.00010	0.00009	0.00010	0.00010	0.02039	0.00010	0.00010	0.00010	0.00010
LOCB	0.00010	0.00009	0.00010	0.00010	0.00010	0.00010	0.00287	0.00010	0.00010	0.00010
LOCC	0.00083	0.00010	0.00010	0.00010	0.00010	0.01184	0.00010	0.00010	0.00010	0.00010
LOCD	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.03342	0.00010
LOCE	0.00010	0.02039	0.00010	0.01184	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010
LOCF	0.00010	0.00010	0.00287	0.00010	0.00010	0.00010	0.00010	0.00010	0.05309	0.00010
LOGG	0.00010	0.00010	0.00010	0.00010	0.03342	0.00010	0.05309	0.00010	0.00010	0.00010
LOCH	0.00044	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00403
LOCI	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00403	0.00010

E. Tahap pembaharuan pheromone Global

Setelah tahap 1 dan 2 telah selesai untuk mendapatkan satu rute dan setiap  $\tau$  lokasi yang dikunjungi telah mengalami pembaharuan pheromone ( $\tau$ ) secara lokal, maka tahap selanjutnya adalah untuk membarui pheromone  $\tau$  ( $\tau$ ) secara



global berdasarkan persamaan (4) namun hanya lokasi yang menghasilkan rute dengan jarak terpendek. Persamaan dari pembaharuan  $\tau$  pheromone ( $\tau$ ) global, hasil keseluruhan pembaharuan dapat dilihat pada tabel VII dibawah ini.

TABEL VII  
NILAI PHEROMONE SETELAH TAHAP MENGALAMI PEMBAHARUAN GLOBAL

	GUDANG	LOCA	LOCB	LOCC	LOCD	LOCE	LOCF	LOGG	LOCH	LOCI
GUDANG	0.0030	0.0030	0.0030	0.0248	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
LOCA	0.0030	0.0030	0.0182	0.0030	0.0030	0.0030	0.0482	0.0030	0.0030	0.0030
LOCB	0.0030	0.0182	0.0030	0.0030	0.0030	0.0030	0.0482	0.0030	0.0030	0.0030
LOCC	0.0248	0.0030	0.0030	0.0030	0.0030	0.0248	0.0030	0.0030	0.0030	0.0030
LOCD	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0182	0.0030	0.0182
LOCE	0.0030	0.0182	0.0030	0.0248	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
LOCF	0.0030	0.0030	0.0482	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030
LOGG	0.0030	0.0030	0.0030	0.0030	0.0030	0.0182	0.0030	0.0030	0.0030	0.0030
LOCH	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0030	0.0182
LOCI	0.0030	0.0030	0.0030	0.0030	0.0182	0.0030	0.0030	0.0030	0.0030	0.0030

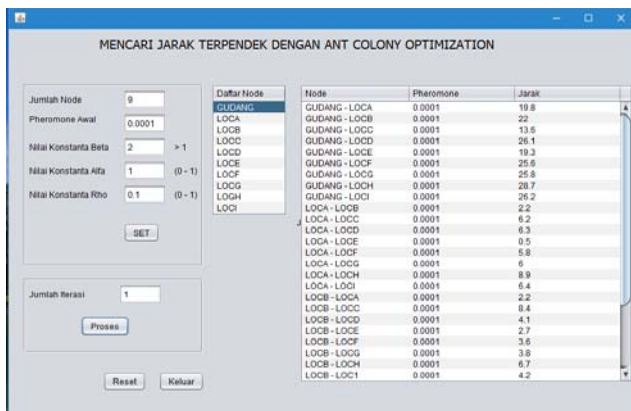
F. Interface

Desain *interface software* yang dibuat memiliki 2 bentuk form yang berfungsi sebagai input maupun output dari *software* ini, yaitu :

Form Input Form input yang merupakan form utama dari seluruh aktivitas pencarian rute terpendek dengan menggunakan algoritma *Ant Colony Optimization*. Form ini memiliki 2 bagian sebagai input dari *software* ini, yaitu :

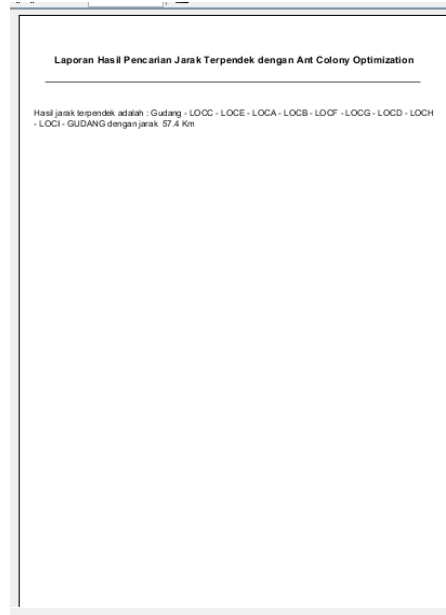
1). Input jumlah lokasi dan nama lokasi Pada bagian ini user mengisi jumlah lokasi yang akan dikunjungi untuk mengisi nama lokasinya user langsung memilih dengan cara dengan cara mengetikkannya pada label yang telah disediakan dan meng-klik nama lokasi yang telah tercantum pada list box yang telah disediakan.

2). Input parameter perhitungan Pada bagian ini user mengisi nilai-nilai parameter yang digunakan pada perhitungan dengan menggunakan algoritma *Ant Colony Optimization* dengan cara mengetikkannya pada label-label yang telah disediakan. Nilai-nilai parameter yang harus diisi oleh user adalah nilai pheromone awal, nilai q0, nilai beta, nilai rho, nilai alpha, dan jumlah iterasi perhitungan yang diinginkan. Form input ditunjukkan pada gambar berikut.



Gbr. 5 Input jarak terdekat

Form Output Form output berisi laporan dari hasil pengolahan dengan menggunakan *software* ini yang akan digunakan oleh para picker sebagai panduan sewaktu melakukan pengambilan barang digudang. Laporan ini memuat urutan lokasi yang harus dikunjungi oleh para picker beserta panduan arah untuk membantu para picker menuju lokasi yang ditetapkan. Form output ditunjukkan pada gambar berikut ini



Gbr. 6 Output

Berdasarkan hasil perhitungan jarak tempuh dengan algoritma ACS terlihat bahwa jarak tempuh terpendek didapat melalui perhitungan dengan menggunakan algoritma ACO yaitu 57,4 Km dimulai dari GUDANG - LOCC - LOCE - LOCA - LOCB - LOCF - LOGG - LOCD - LOCH - LOCI - GUDANG.

IV. PENUTUP

A. Kesimpulan

Berikut merupakan kesimpulan berdasarkan aplikasi yang telah di buat : Algoritma *ant colony optimization* menggunakan fungsi heuristik untuk mendapatkan hasil yang optimal sehingga kekurangan dari algoritma *ant colony optimization* ini adalah waktu proses dalam mendapatkan hasil yang paling optimal sangat tergantung dari jumlah iterasi perhitungan yang digunakan.

B. Saran

Berikut merupakan saran berdasarkan aplikasi yang telah di buat :

1. Diharapkan dalam pengembangan ditambahkan pengaturan lokasi lebih jelas
2. Diharapkan untuk mendapatkan hasil travel sales person yang lebih baik dilakukan pembelajaran teknik-

teknik algoritma semut yang lebih spesifikasi dan mendalam

#### REFERENSI

- [1] Jain Ashish, Jain Dr. Suresh, and Chande Dr. P.K., "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable", International Journal of Innovation, Management and Technology, Vol. 1, No. 3, pp. 248-251, August 2010.
- [2] Karl F.Doerner, Daniel Merkle, and Thomas Stϑzle, "Special Issue on Ant Colony Optimization", Swarm Intell (2009) 3: 1-2, DOI 10.1007/s11721-008-0025-1.
- [3] Pei Hua Chen and Hua Hua Cheng, "IRT-based Automated Test Assembly: A Sampling and Stratification Perspective", The University of Texas at Austin, August 2005.
- [4] Cole A. J., "The Preparation of Examination Time-tables Using A Small-Store Computer", Computer Journal, 7: 117-121, 1964.
- [5] Welsh D.J.A. and Powell M. B., "An Upper Bound for The Chromatic Number of A Graph and Its Application to Timetabling Problems", Computer Journal, 10(1): PP. 85-86, 1967.
- [6] Marco Dorigo and Alberto Colorni, "The Ant Sytem: Optimization by A Colony of Cooperating Agents", IEEE Transaction on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, pp. 1-13,1996.
- [7] Noertjahya dan Agustinus, "Studi Analisis Rapid Aplication Development sebagai salah satu alternative metode pengembangan perangkat lunak", Universitas Kristen Petra, Jakarta, Indonesia.2002.