

**PENGGUNAAN ALGORITMA ANT COLONY SYSTEM DALAM
TRAVELING SALESMAN PROBLEM (TSP) PADA PT. EKA JAYA
MOTOR**

Eka Mindaputra

J2A 003 021

Skripsi

Diajukan sebagai syarat untuk memperoleh gelar Sarjana Sains / Sarjana Komputer

pada

Program Studi Matematika

**PROGRAM STUDI MATEMATIKA JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS DIPONEGORO**

SEMARANG

2009

HALAMAN PENGESAHAN

Judul : Penggunaan Algoritma Ant Colony System Dalam Traveling
Salesman Problem (TSP) Pada PT. Eka Jaya Motor

Nama : Eka Mindaputra

NIM : J2A 003 021

Telah diujikan pada sidang Tugas Akhir tanggal 19 Juni 2009

dan dinyatakan **lulus** pada tanggal Juni 2009

Semarang, Juni 2009
Panitia Penguji Tugas Akhir
Ketua,

Bambang Irawanto, S.Si, M.Si
NIP. 132 102 826

Mengetahui,
Ketua Jurusan Matematika
FMIPA UNDIP

Mengetahui,
Ketua Program Studi Matematika
Jurusan Matematika FMIPA UNDIP

Dr. Widowati, M.Si
NIP. 132 090 819

Bambang Irawanto, S.Si, M.Si
NIP. 132 102 826

HALAMAN PENGESAHAN

Judul : Penggunaan Algoritma Ant Colony System Dalam Traveling
Salesman Problem (TSP) Pada PT. Eka Jaya Motor

Nama : Eka Mindaputra

NIM : J2A 003 021

Telah diujikan pada sidang Tugas Akhir tanggal 19 Juni 2009

Pembimbing Utama

Semarang, Juni 2009
Pembimbing Anggota

Bambang Irawanto, S.Si, M.Si
NIP. 132 102 826

Lucia Ratnasari, S.Si M.Si
NIP. 132 204 997

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyusun tugas akhir ini. Shalawat dan salam penulis sampaikan kepada Rasulullah SAW beserta keluarganya, sahabatnya, dan orang-orang yang tetap setia mengikuti sunnahnya.

Tugas Akhir ini berjudul **“Penggunaan Algoritma Ant Colony System Dalam Traveling Salesman Problem (TSP) Pada PT. Eka Jaya Motor”** disusun sebagai salah satu syarat untuk memperoleh gelar sarjana strata satu pada Fakultas Matematika dan Ilmu Pengetahuan Alam di Universitas Diponegoro Semarang.

Pada kesempatan ini penulis mengucapkan terimakasih kepada:

1. Dr. Widowati ,S.Si, M.Si selaku Ketua Jurusan Matematika FMIPA UNDIP dan juga dosen wali penulis yang telah mengarahkan penulis dari awal perkuliah hingga selesainya Tugas Akhir ini.
2. Bambang Irawanto, S.Si, M.Si selaku dosen Pembimbing I yang dengan sabar membimbing dan mengarahkan penulis hingga selesainya Tugas Akhir ini.
3. Lucia Ratnasari, S.Si, M.Si selaku dosen Pembimbing II yang telah membimbing dan mengarahkan penulis hingga selesainya Tugas Akhir ini.
4. Bapak dan Ibu dosen Jurusan Matematika FMIPA UNDIP di mana penulis mendapatkan pengalaman dan ilmu pengetahuan.

5. Bapak dan Ibu di rumah atas segala sesuatunya yang telah diberikan kepada penulis sampai saat ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dalam penyelesaian Tugas Akhir ini. Semoga Allah SWT membalas segala kebaikan yang telah Anda berikan kepada penulis, Amiiin.

Semoga tulisan ini bermanfaat bagi siapa saja yang berkepentingan dengan ilmu Matematika.

Semarang, Juni 2009

Penulis

ABSTRAK

Ant Colony System (ACS) adalah sebuah metodologi yang dihasilkan melalui pengamatan terhadap semut. Pada algoritma ACS, semut berfungsi sebagai agen yang ditugaskan untuk mencari solusi terhadap suatu masalah optimisasi. ACS telah diterapkan dalam berbagai bidang, salah satunya adalah untuk mencari solusi optimal pada Traveling Salesman Problem (TSP).

Tugas akhir ini memberikan usulan penggunaan algoritma Ant Colony System dalam aktivitas order picking pada PT. Eka Jaya Motor untuk mendapatkan rute yang paling pendek serta pengaplikasian strategi tersebut dengan membangun sebuah sistem informasi pencarian rute yang dapat membantu dalam aktivitas order picking tersebut.

Dengan menggunakan strategi S-Shape yang sekarang digunakan oleh PT. Eka Jaya Motor, picker harus menempuh jarak sejauh 70,03 meter dengan waktu berjalan selama 84,036 detik sedangkan dengan menggunakan algoritma Ant Colony System picker harus menempuh jarak sejauh 52,53 meter dengan waktu berjalan selama 63,036 detik.

Kata kunci: picker, order picking, rute, ant colony system, strategi s-shape, optimisasi, traveling salesman problem.

ABSTRACT

Ant Colony System (ACS) is a method that is produced through an observation to ants. In ACS algorithm, the ants functioned as the agent to find solution regarding an optimization. ACS has been used in many sectors, one of them is to search optimal solution in Taveling Salesman Problem (TSP)

This final project gives a proposal of using Ant Colony System algorithm in order picking activity at PT. Eka Jaya Motor to get the shortest route and aplicating this algorithm with build the finder of shortest rute information system that will help the activity of order picking.

By using s-shape strategy that now used by the company, picker must walk for 70,03 meter and with walking time for 84,036 second. By using the Ant Colony System algorithm, picker must walk for 52,53 meter and with walking time for 63,036 second.

Keywords: picker, order picking, rute, ant colony system, s-shape strategy, optimization, traveling salesman problem.

DAFTAR ISI

BAB I	PENDAHULUAN	1
1.1	Latar Belakang	1
1.3	Perumusan Masalah	3
1.4	Tujuan	3
1.5	Pembatasan Masalah	3
1.5	Sistematika Penulisan	3
BAB II	TEORI PENUNJANG	5
2.1	Graf	5
2.1.1	Definisi Graf	5
2.1.2	Graf Hamilton	8
2.2	Optimasi	9
2.2.1	Pengertian Optimasi	9
2.2.2	Pengertian Nilai Optimal	9
2.2.3	Macam-macam Persoalan Optimasi	10
2.3	Traveling Salesman Problem (TSP)	10
2.3.1	Penerapan Algoritma Semut	10
2.3.2	Contoh Kasus	11
2.3.3	Penyelesaian TSP Menggunakan Algoritma Semut	13
BAB III	PEMBAHASAN	14
3.1	Algoritma Semut	14
3.1.1	Sejarah Algoritma Semut	14

3.1.2 Cara Kerja Semut Mencari Jalur Optimal	14
3.1.3 <i>Ant Colony System</i>	17
3.1.4 Karakteristik <i>Ant Colony System</i> (ACS).....	18
3.2 Algoritma <i>Ant Colony System</i> (ACS)	23
3.3 Analisis Algoritma Semut untuk Mencari Nilai Optimal Menggunakan Graf.....	29
3.4 Penyelesaian Masalah dengan <i>Ant Colony System</i>	34
3.5 Perhitungan Jarak Rute Pengambilan Part.....	40
3.6 Desain Program	52
3.6.1 Proses <i>Software</i>	52
3.6.2 Diagram Konteks.....	53
3.6.3 DFD level 0	54
3.6.4 Data Base.....	56
3.6.4.1 <i>Entity Relationship Diagram</i>	56
3.6.4.2 Transformasi Model Data ke Basis Data Fisik	58
3.6.5 <i>Interface</i>	62
BAB IV KESIMPULAN DAN SARAN	68
4.1 Kesimpulan.....	68
4.2 Saran	69

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2. 1	Contoh Graf.....	5
Gambar 2. 2	Walk Dalam Graf	6
Gambar 2. 3	Graf Berarah dan Graf Tak Berarah.....	7
Gambar 2. 4	Graf Terhubung dan Tak Terhubung	8
Gambar 2. 5	Graf Hamilton, Graf Semi-Hamilton, Graf Bukan Hamilton	9
Gambar 2. 6	Graf Lengkap	12
Gambar 2.7	Sirkuit Hamilton.....	12
Gambar 3. 1	Lintasan Awal Semut Menuju Tempat Makanan.....	16
Gambar 3.2	Lintasan Optimal Semut Menuju Tempat Makanan.....	16
Gambar 3.3	Algoritma ACS.....	26
Gambar 3. 4	Lintasan Awal Semut Menuju Tempat Makanan.....	30
Gambar 3. 5	Lintasan Semut Menuju Sarang	31
Gambar 3. 6	Lintasan Awal Semut Menuju Makanan pada Iterasi ke-2	32
Gambar 3. 7	Lintasan Awal Semut Menuju Sarang pada Iterasi ke-2.....	33
Gambar 3. 8	Lintasan Optimal Semut Menuju Tempat Makanan	33
Gambar 3. 9	Denah Lokasi Gudang.....	39
Gambar 3. 10	Jalur tempuh dengan menggunakan strategi S-Shape	41
Gambar 3. 11	Diagram konteks sistem pencarian rute	54
Gambar 3. 12	DFD level 0 sistem pencarian rute	57
Gambar 3. 13	ERD Database Sistem	57

Gambar 3. 14	Input	64
Gambar 3. 15	Output.....	65
Gambar 3. 16	Jalur tempuh dengan menggunakan algoritma ACS.....	67

DAFTAR TABEL

Tabel 3. 1	Data Penjualan.....	36
Tabel 3. 2	Jarak antar lokasi dalam satuan meter (Algoritma ACS).....	37
Tabel 3. 3	Invers jarak ($\eta(r, s)$)	44
Tabel 3. 4	Pheromone (τ) awal	45
Tabel 3. 5	Hasil perhitungan <i>temporary</i> dan probabilitas dari titik awal DEPOT	46
Tabel 3. 6	Nilai pheromone (τ) setelah mengalami pembaharuan lokal untuk $\tau(\text{depot}, B01 - A)$	48
Tabel 3. 7	Nilai pheromone (τ) setelah tahap mengalami pembaharuan pheromone lokal dari semua picker.....	49
Tabel 3. 8	Nilai pheromone (τ) setelah mengalami pembaharuan global.....	52
Tabel 3. 9	Tabel TGridJarak.....	58
Tabel 3. 10	Tabel TGridProbabilitas	59
Tabel 3. 11	Tabel THasil	59
Tabel 3. 12	Tabel THasilText.....	60
Tabel 3. 13	Tabel THasilUrut.....	60
Tabel 3. 14	Tabel TJarakNode	61
Tabel 3. 15	Tabel TNodeAwal	61
Tabel 3. 16	Tabel TNode.....	62
Tabel 3. 17	Tabel Hasil Perhitungan Software APS	65
Tabel 3. 18	Perbandingan Jarak Tempuh antara strategi S-Shape dengan ACS	66

DAFTAR SIMBOL

- q = bilangan pecahan acak
- q_0 = Probabilitas semut melakukan eksplorasi pada setiap tahapan
- $\tau(t,u)$ = nilai dari jejak *pheromone* pada titik (t,u)
- $\eta(t,u)$ = invers jarak antara titik t dan u
- β = parameter yang mempertimbangkan kepentingan relatif dari informasi *heuristic*
- L_{nn} = panjang tur yang diperoleh
- c = jumlah lokasi
- ρ = koefisien penguapan *pheromon*
- $\Delta\tau$ = perubahan *pheromone*
- L_{gb} = panjang jalur terpendek pada akhir siklus
- α = tingkat kepentingan relatif dari *pheromone*
- \leftarrow = menuju ke-

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang yang digunakan dalam penulisan Tugas Akhir, permasalahan, tujuan dari penulisan, perumusan masalah, batasan masalah, serta sistematika penulisan Tugas Akhir sebagai syarat mendapatkan gelar Sarjana Strata 1 (S1).

1.1 Latar Belakang

Secara umum suatu gudang membutuhkan produk *handling* (basis operasi yang mengikutsertakan manusia dan mesin dalam pengoprasian gudang) yang sangat besar dan itu sangat membutuhkan waktu yang besar. Berdasarkan penelitian yang dilakukan oleh M. Shouman (2005) gudang ataupun *distribution center* pada suatu perusahaan memiliki tiga kategori utama dalam menangani produk *handling* yaitu pendesainan layout dari gudang dan alokasi produknya, *order batching*, serta *order picking* atau pemilihan rute pengambilan barang.

Dari ketiga kategori tersebut, pembenahan pada *order picking* atau rute pengambilan barang merupakan hal yang sangat mempengaruhi waktu pelayanan terhadap konsumen serta menghabiskan 65% dari total biaya operasi gudang (Petersen, 1999).

Strategi S-Shape merupakan salah satu strategi rute pengambilan barang dalam aktivitas *order picking* yang saat ini digunakan PT Eka Jaya Motor, dimana *picker* masuk dari ujung *aisle* yang satu dan keluar dari ujung yang lain pada *aisle*

yang sama. Strategi ini sangat mudah untuk digunakan namun sangat tidak efisien dalam mengurangi jarak tempuh dari aktivitas *order picking* tersebut.

Permasalahan rute pada aktivitas *order picking* dalam mengurangi jarak tempuh dapat dikategorikan sebagai *Travelling Salesman Problem* (TSP) dimana pada aktivitas tersebut *picker* harus menuju ke semua lokasi barang yang akan diambil dan kembali lagi ke lokasi awal dimana *picker* tersebut berangkat.

Berdasarkan hasil penelitian yang dilakukan oleh M.Dorigo dan L. M Gambardella (1997) dalam penyelesaian kasus TSP, terbukti bahwa algoritma Ant Colony System (ACS) mampu mendapatkan hasil tur terbaik dibandingkan dengan algoritma genetik (GA), *evolutionary programming* (EP), *simulated annealing* (SA), dan *annealing-genetic algorithm* (AG).

Untuk itu penelitian tugas akhir ini menerapkan algoritma Ant Colony System sebagai sistem usulan dalam pemilihan rute untuk mendapatkan rute terpendek pada aktivitas *order picking*.

1.2 Perumusan Masalah

Sebuah perusahaan yang bekerja sebagai penyuplai komponen-komponen dalam perakitan mobil mendapat sedikit kendala dalam memenuhi permintaan konsumennya, salah satunya adalah proses pemindahan barang atau pengambilan barang (*order picking*) dari penyimpanan untuk dikirimkan kepada konsumen. Pada saat ini PT. Eka Jaya Motor dalam proses *order picking* menggunakan strategi *S-Shape*, yaitu dengan menyisir seluruh gudang penyimpanan untuk mengambil barang yang telah dipesan oleh konsumen, strategi ini dirasa kurang

efisien dan memakan banyak waktu, sehingga konsumen yang telah memesan tidak dapat dilayani dengan cepat.

Akan dibandingkan penyelesaian masalah TSP dalam proses order picking PT. Eka Jaya Motor yang menggunakan strategi S-Shape dengan Ant Colony System (ACS).

1.3 Tujuan

Adapun tujuan dari penulisan Tugas Akhir ini adalah pengaplikasian algoritma Ant Colony System dalam Traveling Salesman Problem (TSP) PT. Eka Jaya Motor.

1.4 Pembatasan Masalah

Pembatasan masalah dalam penulisan Tugas Sarjana ini hanya difokuskan pada aktivitas *order picking* di PT Eka Jaya Motor.

1.5 Sistematika Penulisan

Sistematika penulisan Tugas Sarjana ini adalah sebagai berikut :

BAB I PENDAHULUAN

Berisi tentang latar belakang permasalahan, perumusan masalah yang ada, tujuan pemecahan masalah, batasan masalah dan sistematika penulisan.

BAB II TEORI PENUNJANG

Bab ini berisi dasar-dasar teori dan metode yang digunakan sebagai dasar dan alat untuk memecahkan masalah.

BAB III PEMBAHASAN

Berisi data-data yang akan digunakan dalam analisis atau perhitungan maupun data penunjang yang telah disiapkan atau diolah untuk pemecahan masalah serta desain dari program yang digunakan.

BAB IV PENUTUP

Berisi tentang kesimpulan dari hasil pembahasan yang telah dilakukan, serta saran bagi penulis pada khususnya dan pembaca pada umumnya.

BAB II

TEORI PENUNJANG

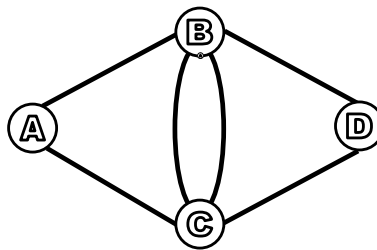
2.1 Graf

2.1.1 Definisi Graf

Definisi 2.1 (3)

Suatu graf G didefinisikan sebagai diagram yang terdiri dari titik-titik yang disebut *vertices* dinyatakan dengan $V(G)$ dan dihubungkan oleh sisi-sisi yang disebut *edges* dinyatakan dengan $E(G)$, serta setiap sisi menghubungkan tepat dua titik. Suatu graf G dapat dinotasikan sebagai $G = (V,E)$, dengan $V(G) =$ tidak kosong .

Contoh 2.1 :



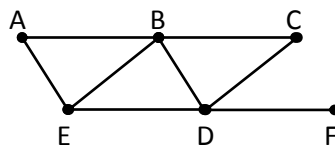
Gambar 2.1 Contoh Graf

Pada graf gambar 2.1, $E(G) = \{(A,B), (B,C), (B,D), (C,B), (C,A), (D,C)\}$ dan $V(G) = \{A,B,C,D\}$.

Definisi 2.2 (3)

Suatu jalan (*walk*) dalam graf G adalah barisan titik-titik dan sisi-sisi yang dimulai dan diakhiri oleh suatu titik. Panjang suatu *walk* dihitung berdasarkan jumlah sisi dalam *walk* tersebut.

Contoh 2.2 :



Gambar 2.2 Walk dalam graf

Salah satu *walk* pada graf pada gambar 2.2 adalah $A, (A,B), B, (B,C), C, (C,D), D, (D,E), E, (E,A), A$ dengan panjang 5.

Definisi 2.3 (3)

Jika semua sisi suatu *walk* berbeda, maka *walk* disebut *trail*. Jika semua titiknya juga berbeda, maka *trail* disebut *path* (*lintasan*).

Contoh 2.3 :

Pada gambar 2.2 *walk* $A, (A,B), B, (B,C), C, (C,D), D, (D,E), E, (E,A), A$ adalah suatu *trail* tetapi bukan suatu *path*, sedangkan $A, (A,B), B, (B,C), C, (C,D), D, (D,F), F$ merupakan *path*.

Definisi 2.4 (3)

Suatu *walk* tertutup dalam graf G yang titik awal sama dengan titik akhirnya dan semua titik-titik didalamnya berbeda disebut *cycle*(*sirkuit*).

Contoh 2.4 :

Pada gambar 2.2 *walk* $A, (A,B), B, (B,C), C, (C,D), D, (D,B), B, (B,E), E, (E,A), A$ merupakan *walk* tertutup tetapi bukan suatu *cycle*, sedangkan $A, (A,B), B, (B,C), C, (C,D), D, (D,E), E, (E,A), A$ merupakan suatu *cycle* (*sirkuit*).

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

1. Graf tak-berarah (*undirected graph*) Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.
2. Graf berarah (*directed graph* atau *digraph*) Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.

Contoh :

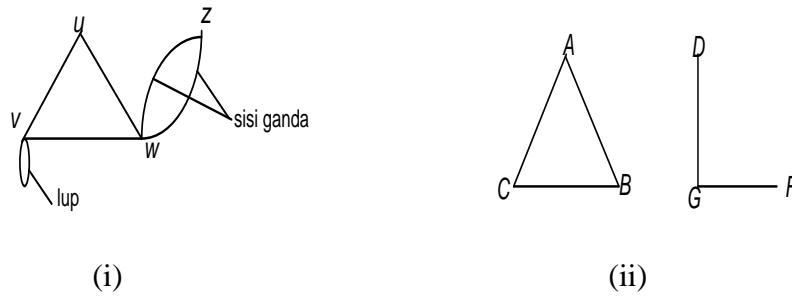


Gambar 2.3. Graf Berarah (a) dan Graf tidak Berarah (b)

Dua buah titik v_1 dan titik v_2 disebut terhubung jika terdapat sisi dari v_1 ke v_2 . Graf G disebut graf terhubung (*connected graph*) jika untuk setiap pasang

titik v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j . Jika tidak, maka graf G disebut **graf tak-terhubung** (*disconnected graph*).

Contoh :



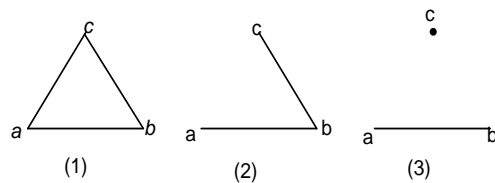
Gambar 2.4. Graf G terhubung (i) dan tak terhubung (ii)

Graf berarah G dikatakan terhubung jika graf tidak berarahnya terhubung (graf tidak berarah dari G diperoleh dengan menghilangkan arahnya).

2.1.2 Graf Hamilton

Definisi 2.5 (2)

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**.



Gambar 2.5. Graf Hamilton(1), Graf Semi-Hamilton(2), Graf Bukan Hamilton

Optimisasi

2.2.1 Pengertian Optimisasi (2)

Optimisasi ialah suatu proses untuk mencapai hasil yang ideal atau **optimal** (nilai efektif yang dapat dicapai). Dalam disiplin matematika optimisasi merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maksimal dari suatu fungsi nyata. Untuk dapat mencapai nilai *optimal* baik minimal atau maksimal tersebut, secara sistematis dilakukan pemilihan nilai variabel integer atau nyata yang akan memberikan solusi optimal.

2.2.2 Pengertian Nilai Optimal (2)

Nilai optimal adalah nilai yang didapat dengan melalui suatu proses dan dianggap menjadi suatu solusi jawaban yang paling baik dari semua solusi yang ada. Nilai optimal dapat dicari dengan dua cara, yaitu:

1. Cara konvensional, yaitu mencoba semua kemungkinan yang ada dengan mencatat nilai yang didapat cara ini kurang efektif, karena optimasi akan berjalan secara lambat.
2. Cara kedua adalah dengan menggunakan suatu rumus sehingga nilai optimal dapat diperkirakan dengan cepat dan tepat.

2.2.3 Macam-macam Persoalan Optimisasi

Persoalan yang berkaitan dengan optimisasi sangat kompleks dalam kehidupan sehari-hari. Nilai optimal yang didapat dalam optimisasi dapat berupa besaran panjang, waktu, jarak dan lain-lain. Berikut ini adalah beberapa persoalan yang memerlukan optimisasi: Menentukan lintasan terpendek dari suatu tempat ke tempat yang lain, menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal, mengatur jalur kendaraan umum agar semua lokasi dapat dijangkau, mengatur *routing* jaringan kabel telepon agar biaya pemasangan kabel tidak terlalu besar.

2.3 Traveling Salesman Problem (TSP)

2.3.1 Penerapan Algoritma semut

Algoritma optimisasi koloni semut telah digunakan untuk menghasilkan penyelesaian yang mendekati optimal. Aplikasi algoritma semut dalam kehidupan sehari-hari mencakup beberapa persoalan, yaitu:

1. *Traveling Salesman Problem (TSP)*, yaitu mencari jalur terpendek dalam sebuah graf menggunakan sirkuit Hamilton.
2. *Quadratic Assignment Problem (QAP)* yang berusaha meng-*assign* sejumlah n *resources* untuk ditempatkan pada sejumlah m lokasi dengan meminimalisir biaya *assignment*.

3. *Job-shop Scheduling Problem* (JSP) juga salah satu contoh aplikasi algoritma semut untuk menjadwalkan sejumlah j pekerjaan menggunakan sejumlah m mesin demikian sehingga seluruh pekerjaan diselesaikan dalam waktu yang minimal.

4. pengaturan jalur kendaraan.

5. pewarnaan graf.

6. network routing, dll.

2.3.2 Contoh Kasus

Travelling Salesman Problem (TSP) adalah suatu masalah yang ditemukan oleh pedagang yang harus bepergian dan singgah di beberapa kota hingga kembali ke kota semula. Dalam kehidupan sehari-hari, kasus TSP ini dapat diaplikasikan untuk menyelesaikan kasus lain, yaitu:

1. Pak Pos mengambil surat di kotak pos yang tersebar pada n buah lokasi di berbagai sudut kota.

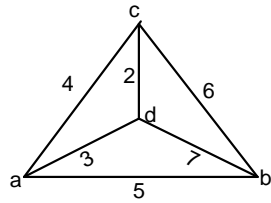
2. Lengan robot mengencangkan n buah mur pada beberapa buah peralatan mesin dalam sebuah jalur perakitan.

3. Produksi n komoditi berbeda dalam sebuah siklus.

Seperti yang diketahui, bahwa untuk mencari jumlah sirkuit Hamilton di dalam graf lengkap

dengan n simpul adalah: $(n - 1)!/2$.

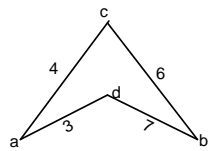
Contoh:



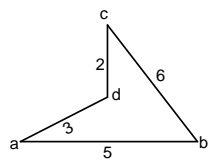
Gambar 2.6. Graf Lengkap

Graf di atas memiliki $(4 - 1)!/2 = 3$ sirkuit Hamilton (Gambar 2.6), yaitu:

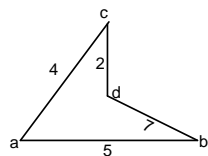
- $I_1 = (a, d, b, c, a)$ atau (a, c, b, d, a) dengan panjang
 $= 4 + 6 + 7 + 3 = 20$
- $I_2 = (a, b, c, d, a)$ atau (a, d, c, b, a) dengan panjang
 $= 5 + 6 + 2 + 3 = 16$
- $I_3 = (a, c, d, b, a)$ atau (a, b, d, c, a) dengan panjang
 $= 4 + 2 + 7 + 5 = 18$



I_1



I_2



I_3

Gambar 2.7. Sirkuit Hamilton

Jadi, sirkuit Hamilton terpendek adalah $I_2 = (a, b, c, d, a)$ atau (a, d, c, b, a)

dengan panjang sirkuit $= 5 + 6 + 2 + 3 = 16$

Jika jumlah simpul $n = 20$ akan terdapat $(19!)/2$ sirkuit Hamilton atau sekitar 6×10^{16} penyelesaian.

2.3.3 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). Dalam tugas akhir ini kita akan membahas teknik yang terakhir, yaitu 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.

BAB III

PEMBAHASAN

3.1 Algoritma Semut

3.1.1 Sejarah Algoritma Semut

Pada tahun 1996, dunia ilmu pengetahuan pun ikut belajar dari semut dengan diperkenalkannya algoritma semut, atau *Ant Colony Optimization*, sebagai sebuah simulasi multi agen yang menggunakan metafora alami semut untuk menyelesaikan *problem* ruang fisik. Algoritma semut diperkenalkan oleh **Moyson** dan **Manderick** dan secara meluas dikembangkan oleh **Marco Dorigo**, merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik. Algoritma ini diambil dengan analogi oleh perilaku semut dalam menemukan jalur dari koloninya menuju makanan.

3.1.2 Cara Kerja Algoritma Semut Mencari Jalur Optimal

Semut mampu mengindra lingkungannya yang kompleks untuk mencari makanan dan kemudian kembali ke sarangnya dengan meninggalkan zat *pheromon* pada jalur-jalur yang mereka lalui. **Pheromon** 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, *pheromon* menyebar ke luar tubuh dapat mempengaruhi dan dikenali oleh individu lain yang sejenis (satu spesies). Proses peninggalan *pheromon* ini dikenal sebagai *stigmergy*, sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan pulang ke sarang,

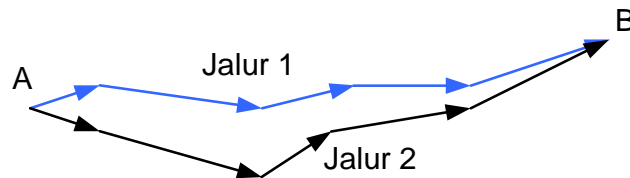
tetapi juga memungkinkan para semut berkomunikasi dengan koloninya. Seiring waktu, bagaimanapun juga jejak *pheromon* akan menguap dan akan mengurangi kekuatan daya tariknya. Lebih lama seekor semut pulang pergi melalui jalur tersebut, lebih lama jugalah *pheromon* menguap. Agar semut mendapatkan jalur optimal, diperlukan beberapa proses:

1. Pada awalnya, semut berkeliling secara acak, hingga menemukan makanan.

Lihat Gambar 3.1.

2. Ketika menemukan makanan mereka kembali ke koloninya sambil memberikan tanda dengan jejak *pheromon*.
3. Jika semut-semut lain menemukan jalur tersebut, mereka tidak akan bepergian dengan acak lagi, melainkan akan mengikuti jejak tersebut.
4. Kembali dan menguatkannya jika pada akhirnya mereka pun menemukan makanan.
5. Seekor semut yang secara tidak sengaja menemukan jalur optimal akan menempuh jalur ini lebih cepat dari rekan-rekannya, melakukan *round-trip* lebih sering, dan dengan sendirinya meninggalkan *pheromon* lebih banyak dari jalur-jalur yang lebih lambat ditempuh.
6. *Pheromon* yang berkonsentrasi tinggi pada akhirnya akan menarik semut – semut lain untuk berpindah jalur, menuju jalur paling optimal, sedangkan jalur lainnya akan ditinggalkan.

7. Pada akhirnya semua semut yang tadinya menempuh jalur yang berbeda - beda akan beralih ke sebuah jalur tunggal yang ternyata paling optimal dari sarang menuju ke tempat makanan. Lihat Gambar 3.2.



Gambar 3.1. Lintasan Awal Semut Menuju Tempat Makanan

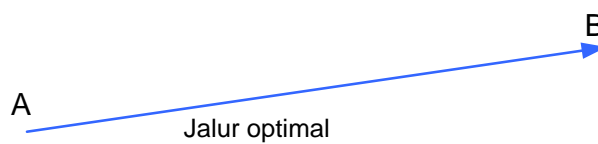
Keterangan Gambar 3.1:

A : Tempat awal koloni (sarang)

B : Tujuan koloni semut (makanan)

Jalur 1 (biru): Lintasan yang ditempuh oleh semut 1

Jalur 2 (hitam): Lintasan yang ditempuh oleh semut 2



Gambar 3.2. Lintasan Optimal Semut Menuju Tempat Makanan

Keterangan Gambar 3.2:

A : Tempat awal koloni (sarang)

B : Tujuan koloni semut (makanan)

Jalur Optimal : Jalur yang dilewati semut setelah beberapa iterasi.

Seluruh proses ini menunjukkan berlangsungnya optimisasi alami kaum semut yang bisa kita tiru dalam kehidupan sehari-hari.

3.1.3 *Ant Colony System*

Ant Colony System (ACS) adalah sebuah metodologi yang dihasilkan melalui pengamatan terhadap semut. Pada algoritma ACS, semut berfungsi sebagai agen yang ditugaskan untuk mencari solusi terhadap suatu masalah optimisasi. ACS telah diterapkan dalam berbagai bidang, salah satunya adalah untuk mencari solusi optimal pada Traveling Salesman Problem (TSP). Dengan memberikan sejumlah n titik, TSP dapat didefinisikan sebagai suatu permasalahan dalam menemukan jalur terpendek dengan mengunjungi setiap titik yang ada hanya sekali.

Secara informal, ACS bekerja sebagai berikut: pertama kali, sejumlah m semut ditempatkan pada sejumlah n titik berdasarkan beberapa aturan inisialisasi (misalnya, secara acak). Setiap semut membuat sebuah tur (yaitu, sebuah solusi TSP yang mungkin) dengan menerapkan sebuah aturan transisi status secara berulang kali. Selagi membangun turnya,

seekor semut juga memodifikasi jumlah *pheromone* (sejumlah informasi yang ditinggalkan oleh semut di tempat yang dilalui dan menandai jalur tersebut) pada ruas-ruas yang dikunjunginya dengan menerapkan aturan pembaruan *pheromone* lokal. Setelah semua semut mengakhiri tur mereka, jumlah *pheromone* yang ada pada ruas-ruas dimodifikasi kembali (dengan menerapkan aturan pembaruan *pheromone* global). Dalam membuat tur, semut 'dipandu' oleh informasi heuristik (mereka lebih memilih ruas-ruas yang pendek) dan oleh informasi *pheromone*. Sebuah ruas dengan jumlah *pheromone* yang tinggi merupakan pilihan yang sangat diinginkan. Kedua aturan pembaruan *pheromone* itu dirancang agar semut cenderung untuk memberi lebih banyak *pheromone* pada ruas-ruas yang harus mereka lewati.

3.1.4 Karakteristik *Ants Colony System* (ACS)

Terdapat tiga karakteristik utama dari ACS, yaitu : Aturan transisi status, Aturan pembaruan *pheromone* lokal, Aturan pembaruan *pheromone* global.

1. Aturan transisi status

Aturan transisi status yang berlaku pada ACS adalah sebagai berikut: seekor semut yang ditempatkan pada titik t memilih untuk menuju ke titik v , kemudian diberikan bilangan pecahan acak q dimana $0 \leq q \leq 1$, q_0 adalah sebuah parameter yaitu Probabilitas semut melakukan eksplorasi pada setiap tahapan, dimana ($0 \leq q_0 \leq 1$) dan $p_k(t,v)$ adalah probabilitas dimana semut k memilih untuk bergerak dari titik t ke titik v .

Jika $q \leq q_0$ maka pemilihan titik yang akan dituju menerapkan aturan yang ditunjukkan oleh persamaan (1)

$$\begin{aligned}
 \text{temporary}(t,u) &= [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta, i = 1,2,3,\dots,n \\
 v &= \max \{ [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta \} \dots\dots\dots(1)
 \end{aligned}$$

Dengan v = titik yang akan dituju

sedangkan jika $q > q_0$ digunakan persamaan (2)

$$v = p_k(t,v) = \frac{[\tau(t,v)] \cdot [\eta(t,v)]^\beta}{\sum_{i=1}^n [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta} \dots\dots\dots(2)$$

dengan

$$\eta(t,u_i) = \frac{1}{\text{jarak}(t,u_i)}$$

dimana $\tau(t,u)$ adalah nilai dari jejak *pheromone* pada titik (t,u) , $\eta(t,u)$ adalah fungsi heuristik dimana dipilih sebagai invers jarak antara titik t dan u , β merupakan sebuah parameter yang mempertimbangkan kepentingan relatif dari informasi *heuristic*, yaitu besarnya bobot yang diberikan terhadap parameter informasi heuristik, sehingga solusi yang dihasilkan cenderung berdasarkan nilai fungsi matematis. Nilai untuk parameter β adalah ≥ 0 . **Pheromon** 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, *pheromon* menyebar ke luar tubuh dapat mempengaruhi dan dikenali oleh individu lain yang sejenis (satu spesies). Proses peninggalan *pheromon* ini dikenal sebagai *stigmergy*, sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan pulang ke sarang, tetapi juga memungkinkan para semut berkomunikasi dengan koloninya. Seiring waktu, bagaimanapun juga jejak *pheromon* akan menguap dan akan mengurangi kekuatan daya tariknya, sehingga jejak *pheromon* harus diperbaharui. Pada ACS pembaruan *pheromon* dibagi menjadi 2, yaitu: Aturan pembaruan *pheromon* lokal, Aturan pembaruan *pheromon* global.

2. Aturan pembaruan *pheromon* lokal

Selagi melakukan tur untuk mencari solusi dari TSP, *semut* mengunjungi ruas-ruas dan mengubah tingkat *pheromon* pada ruas-ruas tersebut dengan menerapkan aturan pembaruan *pheromon* lokal yang ditunjukkan oleh persamaan (3)

$$\tau(t, v) \leftarrow (1 - \rho) \cdot \tau(t, v) + \rho \cdot \Delta\tau(t, v) \dots\dots\dots (3)$$

$$\Delta\tau(t, v) = \frac{1}{L_{nn} \cdot c}$$

dimana :

L_{nn} = panjang tur yang diperoleh

c = jumlah lokasi

ρ = parameter dengan nilai 0 sampai 1

$\Delta\tau$ = perubahan *pheromon*

ρ adalah sebuah parameter (koefisien evaporasi), yaitu besarnya koefisien penguapan *pheromon*. Adanya penguapan *pheromone* menyebabkan tidak semua semut mengikuti jalur yang sama dengan semut sebelumnya. Hal ini memungkinkan dihasilkan solusi alternatif yang lebih banyak. Peranan dari aturan pembaruan *pheromone* lokal ini adalah untuk mengacak arah lintasan yang sedang dibangun, sehingga titik-titik yang telah dilewati sebelumnya oleh tur seekor semut mungkin akan dilewati kemudian oleh tur *semut* yang lain. Dengan kata lain, pengaruh dari pembaruan lokal ini adalah untuk membuat tingkat ketertarikan ruas-ruas yang ada berubah secara dinamis: setiap kali seekor semut menggunakan sebuah ruas maka ruas ini dengan segera akan berkurang tingkat ketertarikannya (karena ruas tersebut kehilangan sejumlah *pheromon*-nya), secara tidak langsung *semut* yang lain akan memilih ruas-ruas lain yang belum dikunjungi. Konsekuensinya, *semut* tidak akan memiliki kecenderungan untuk berkumpul pada jalur yang sama. Fakta ini, yang telah diamati dengan melakukan percobaan [Dorigo dan Gambardella, 1997]. Merupakan sifat yang diharapkan bahwa jika *semut* membuat tur-tur yang berbeda maka akan terdapat kemungkinan yang lebih tinggi dimana salah satu dari mereka akan menemukan solusi yang lebih baik daripada mereka semua berkumpul dalam tur yang sama. Dengan cara ini, *semut* akan membuat penggunaan informasi *pheromon* menjadi lebih baik tanpa pembaruan

lokal, semua *semut* akan mencari pada lingkungan yang sempit dari tur terbaik yang telah ditemukan sebelumnya.

3. Aturan pembaruan *pheromon* global

Pada sistem ini, pembaruan *pheromon* secara global hanya dilakukan oleh semut yang membuat tur terpendek sejak permulaan percobaan. Pada akhir sebuah iterasi, setelah semua *semut* menyelesaikan tur mereka, sejumlah *pheromon* ditaruh pada ruas-ruas yang dilewati oleh seekor semut yang telah menemukan tur terbaik (ruas-ruas yang lain tidak diubah). Tingkat *pheromon* itu diperbarui dengan menerapkan aturan pembaruan *pheromon* global yang ditunjukkan oleh persamaan (4).

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

$$\Delta\tau(t,v) = \begin{cases} L_{gb}^{-1} & \text{jika } (t,v) \in \text{tur_terbaik} \\ 0 & \end{cases}$$

Dimana :

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

L_{gb} = panjang jalur terpendek pada akhir siklus

α = parameter dengan nilai antara 0 sampai 1

$\Delta\tau$ = perubahan *pheromone*

$\Delta\tau(t,v)$ bernilai $\frac{1}{L_{gb}}$ jika ruas (t,v) merupakan bagian dari rute terbaik namun jika sebaliknya $\Delta\tau(t,v)=0$. α adalah tingkat kepentingan relatif dari *pheromon* atau besarnya bobot yang diberikan terhadap *pheromon*, sehingga solusi yang dihasilkan cenderung mengikuti sejarah masa lalu dari semut dari perjalanan sebelumnya, dimana nilai parameter α adalah ≥ 0 , dan L_{gb} adalah panjang dari tur terbaik secara global sejak permulaan percobaan. Pembaruan *pheromon* global dimaksudkan untuk memberikan *pheromon* yang lebih banyak pada tur-tur yang lebih pendek. Persamaan (3) menjelaskan bahwa hanya ruas-ruas yang merupakan bagian dari tur terbaik secara global yang akan menerima penambahan *pheromone*.

3.2 Algoritma *Ants Colony System* (ACS)

Sama halnya dengan cara kerja semut dalam mencari jalur yang optimal, untuk mencari jalur terpendek dalam penyelesaian masalah *Traveling Salesman Problem* (TSP) diperlukan beberapa langkah untuk mendapatkan jalur yang optimal, antara lain :

1. Menentukan *pheromone* awal masing- masing semut. Tapi sebelum itu tentukan terlebih dahulu banyaknya semut dalam proses tersebut, setelah itu tentukan titik awal masing-masing semut.
2. Setelah itu tentukan titik selanjutnya yang akan dituju, ulangi proses sampai semua titik terlewati. Dengan menggunakan persamaan 1 atau 2 dapat ditentukan titik mana yang akan dituju, yaitu dengan :

Jika $q \leq q_0$ maka pemilihan titik yang akan dituju menerapkan aturan yang ditunjukkan oleh persamaan (1)

$$temporary(t,u) = [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta \quad i = 1,2,3,\dots,n$$

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

Dimana v = titik yang akan dituju

sedangkan jika $q > q_0$ digunakan persamaan (2)

$$v = p_i(t,v) = \frac{[\tau(t,v)] \cdot [\eta(t,v)]^\beta}{\sum_{i=1}^n [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta} \dots\dots\dots(2)$$

dengan

$$\eta(t,u_i) = \frac{1}{jarak(t,u_i)}$$

jika titik yang dimaksud bukanlah titik yang akan dilalui, maka kembali ke titik sebelumnya.

3. Apabila telah mendapatkan titik yang dituju, *pheromone* masing-masing pada titik tersebut diubah dengan menggunakan persamaan 3, yaitu :

$$\tau(t,v) \leftarrow (1 - \rho) \cdot \tau(t,v) + \rho \cdot \Delta\tau(t,v) \dots\dots\dots(3)$$

$$\Delta\tau(t,v) = \frac{1}{L_{nn} \cdot c}$$

dimana :

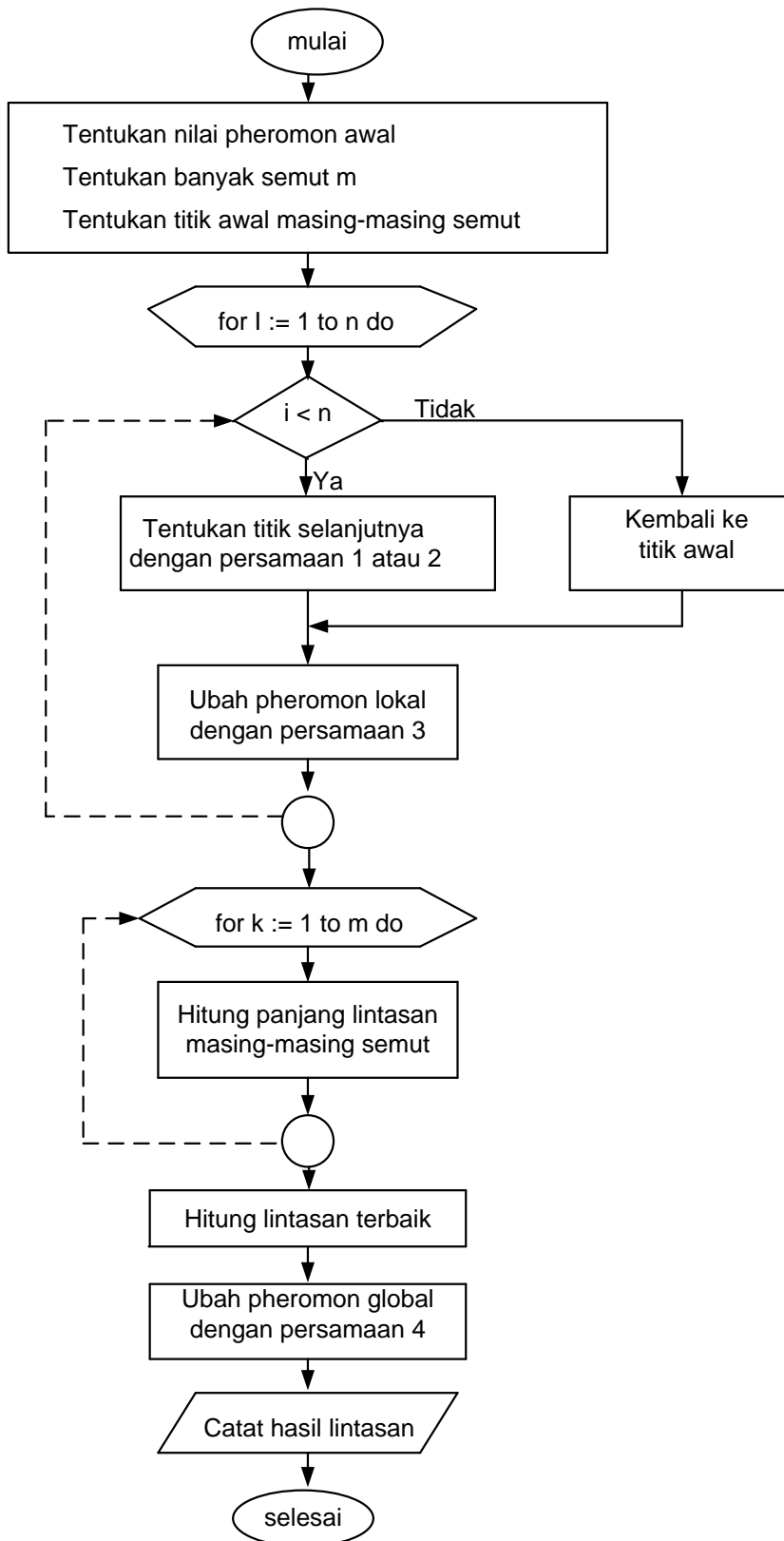
- L_{nn} = panjang tur yang diperoleh
- c = jumlah lokasi
- ρ = parameter dengan nilai 0 sampai 1

$\Delta\tau$ = perubahan *pheromone*

Perubahan *pheromon* tersebut dinamakan perubahan *pheromon* lokal.

4. Setelah proses diatas selesai, hitung panjang lintasan masing-masing semut.
5. Kemudian akan didapatkan panjang lintasan yang minimal.
6. Ubah *pheromone* pada titik-titik yang termuat dalam lintasan tersebut.
7. Setelah semua proses telah dilalui, maka akan didapatkan lintasan dengan panjang lintasan yang minimal.

Berikut adalah algoritma ACS



Gambar 3.3. Algoritma ACS

1. */* Initialization phase */*

For each pair (t,v) $\tau(t,v) := \tau_0$

End-for

For k:=1 to m do

Let t_{k1} be the starting city for ant k

$J_k(t_{k1}) := \{1, \dots, n\} - r_{k1}$

/ $J_k(t_{k1})$ is the set of yet to be visited cities for ant k in city t_{k1} */*

$t_k := t_{k1}$

/ t_k is the city where ant k is located */*

End-for

2. */* This is the phase in which ants build their tours. The tour of ant k is stored in $Tour_k$. */*

For i:=1 to n do

If $i < n$ then

For k:=1 to m do

Choose the next city v_k according to Eq.(1) and Eq.(2)

$J_k(s_k) := J_k(t_k) - v_k$

$Tour_k(i) := (t_k, v_k)$


```

End-for

Else

For k:=1 to m do

    /* In this cycle all the ants go back to the initial city  $t_{k1}$  */

     $v_k := t_{k1}$ 

     $Tour_k(i) := (t_k, v_k)$ 

End-for

End-if

/* In this phase local updating occurs and pheromone is updated
using Eq. (4)*/

For k:=1 to m do

 $\tau(t_k, v_k) := (1-\rho)\tau(t_k, v_k) + \rho\tau_0$ 

 $t_k := v_k$  /* New city for ant k */

End-for

End-for

3. /* In this phase global updating occurs and pheromone is updated */

For k:=1 to m do

    Compute  $L_k$ 

```

/ L_k is the length of the tour done by ant k */*

End-for

Compute L_{best}

/ Update edges belonging to L_{best} using Eq. (3) */*

For each edge (t,v)

$$\tau(t_k, v_k) := (1 - \alpha)\tau(t_k, v_k) + \alpha (L_{best})^{-1}$$

End-for

4. If (End_condition = True) then Print shortest of L_k

Else

goto Phase 2

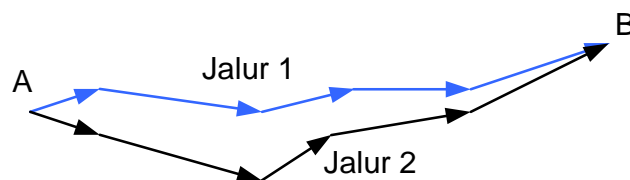
3.3 Analisis Algoritma Semut untuk Mencari Nilai Optimal Menggunakan Graf

Untuk mendiskusikan algoritma semut, lingkungan yang akan digunakan adalah sebuah graf yang *fully connected* (setiap *node* memiliki busur ke *node* yang lain) dan *bidirectional* (setiap jalur bisa ditempuh bolak-balik dua arah). Setiap busur memiliki bobot yang menunjukkan jarak antara dua buah *nodes* yang dihubungkan oleh busur tersebut. Algoritma ini menggunakan sistem multi agen, yang berarti kita akan mengerahkan seluruh koloni semut yang masing-masingnya bergerak sebagai agen tunggal. Setiap semut menyimpan daftar yang memuat *nodes* yang sudah pernah ia lalui, dimana ia tidak diijinkan untuk melalui *node*

yang sama dua kali dalam satu kali perjalanan (daftar ini disebut juga sebagai jalur Hamilton, yaitu jalur pada graf dimana setiap *node* hanya dikunjungi satu kali). Sebuah koloni semut diciptakan, dan setiap semut ditempatkan pada masing-masing *node* secara merata untuk menjamin bahwa tiap *node* memiliki peluang untuk menjadi titik awal dari jalur optimal yang dicari. Setiap semut selanjutnya harus melakukan tur semut, yaitu perjalanan mengunjungi semua *nodes* pada graf tersebut.

Berikut adalah tahapan-tahapan algoritma semut menggunakan graf:

1. Dari sarang, semut berkeliling secara acak mencari makanan kemudian dicatat jarak antara *node* yang semut lalui.
2. Ketika sampai ke makanan, Total jarak dari tiap *node* yang semut tempuh dijumlahkan untuk mendapatkan jarak dari sarang ke makanan.



Gambar 3.4. Lintasan Awal Semut Menuju Tempat Makanan

Keterangan Gambar 3.4:

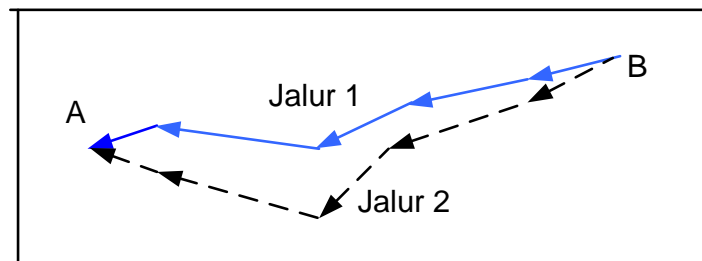
A : Tempat awal koloni (sarang)

B : Tujuan koloni semut (makanan)

Jalur 1 (biru): Lintasan yang ditempuh oleh semut 1

Jalur 2 (hitam): Lintasan yang ditempuh oleh semut 2

3. Ketika kembali ke sarang, sejumlah konsentrasi *pheromon* ditambahkan pada jalur yang telah ditempuh berdasarkan total jarak jalur tersebut. Makin kecil total jarak (atau makin optimal), maka makin banyak kadar *pheromon* yang dibubuhkan pada masing-masing busur pada jalur tersebut.



Gambar 3.5. Lintasan Semut Menuju Sarang

Keterangan Gambar 3.5:

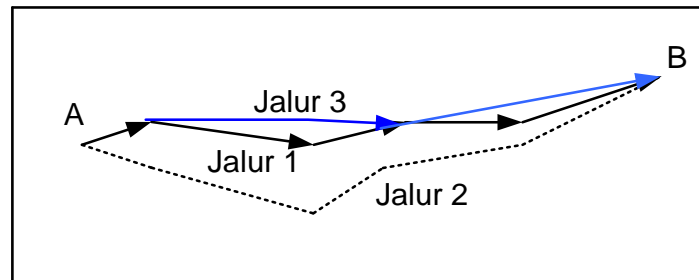
A : Sarang semut

B : Tempat ditemukannya makanan

Jalur 1 (biru) : Jalur yang ditempuh oleh semut 1 dengan pemberian kadar *pheromon* yang tinggi

Jalur 2 (hitam) : Jalur yang ditempuh oleh semut 2 dengan pemberian kadar *pheromon* yang rendah

4. Untuk memilih busur mana yang harus dilalui berikutnya, digunakan sebuah rumus yang pada intinya menerapkan suatu fungsi *heuristic* untuk menghitung intensitas *pheromon* yang ditinggalkan pada suatu busur.



Gambar 3.6. Lintasan Semut Menuju Makanan pada Iterasi ke-2

Keterangan Gambar 3.6:

A : Sarang semut

B : Tempat ditemukannya makanan

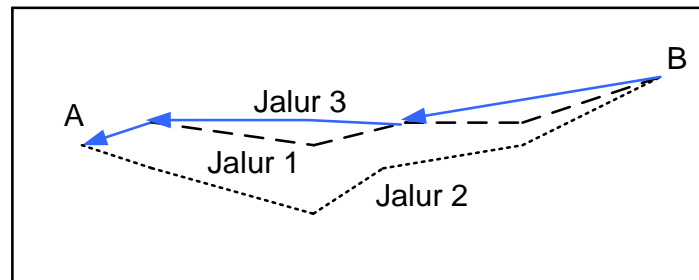
Jalur 1 : Jalur yang ditempuh oleh semut 1 karena kadar *pheromon* yang tinggi

Jalur 2 : Jalur yang tidak ditempuh oleh semut karena kadar *pheromon* yang rendah

Jalur 3 : Jalur yang ditemukan oleh semut 2

5. Pada iterasi berikutnya, busur-busur yang mengandung *pheromon* lebih tinggi ini akan cenderung dipilih sebagai busur yang harus ditempuh berikutnya berdasarkan rumus pemilihan busur. Akibatnya, lama-kelamaan akan terlihat **jalur optimal** pada graf, yaitu jalur yang dibentuk oleh busur-busur dengan kadar

pheromon yang tinggi, yang pada akhirnya akan dipilih oleh semua multi agen semut.



Gambar 3.7. Lintasan Semut Menuju Sarang pada Iterasi ke-2

Keterangan Gambar 3.7:

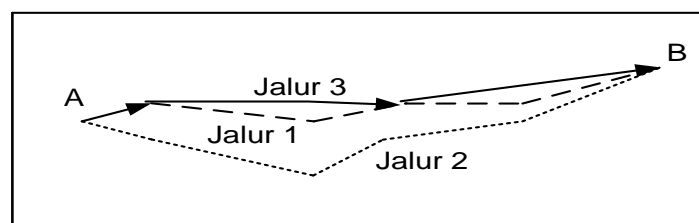
A : Sarang semut

B : Tempat ditemukannya makanan

Jalur 1 (hitam) : Jalur yang ditempuh oleh semut 2 dengan pemberian kadar *pheromon* yang rendah

Jalur 2 : Jalur yang tidak ditempuh

Jalur 3 (biru) : Jalur yang ditempuh oleh semut 2 dengan pemberian kadar *pheromon* yang tinggi.



Gambar 3.8. Lintasan Optimal Semut Menuju Tempat Makanan

Keterangan Gambar 3.8:

A : Sarang semut

B : Tempat ditemukannya makanan

Jalur 1 : : Jalur yang tidak ditempuh karena kadar feromon yang rendah

Jalur 2 : Jalur yang tidak ditempuh karena kadar feromon yang sangat rendah

Jalur 3 : Jalur optimal yang ditempuh oleh semut karena kadar feromon yang tinggi

3.4 Penyelesaian Masalah dengan *Ant Colony System*

PT Eka Jaya Motor adalah perusahaan yang bergerak dibidang pendistribusian kendaraan bermotor dan suku cadang kendaraan merk TOYOTA. Salah satu divisi yang terdapat pada perusahaan ini adalah Part Division. Divisi ini menangani segala aktivitas berkaitan dengan persiapan pemesanan kendaraan suku cadang kendaraan Toyota yang berasal dari PT Toyota Astra Motor (TAM) sampai dengan Dealer Nasmoco Grup.

PT. Eka Jaya Motor bekerja sebagai penyuplai komponen – komponen dalam perakitan mobil mendapat sedikit kendala dalam memenuhi permintaan konsumennya, salah satunya adalah proses pemindahan barang atau pengambilan barang (*order picking*) dari penyimpanan untuk dikirimkan kepada konsumen. Pada saat ini perusahaan dalam proses *order picking* menggunakan strategi S-

Shape, yaitu dengan meyisir seluruh gudang penyimpanan untuk mengambil barang yang telah dipesan oleh konsumen, strategi ini dirasa kurang efisien dan memakan banyak waktu, sehingga konsumen yang telah memesan tidak dapat dilayani dengan cepat.

Dalam pembahasan ini akan dibandingkan penyelesaian masalah perusahaan dalam menentukan rute terpendek (proses *order picking*) dengan solusi yang dimiliki oleh perusahaan (S-Shape) dan Ant Colony System (ACS). Untuk pencarian rute terpendek dalam proses *order picking* dengan menggunakan ACS diperlukan 3 data utama dalam penyelesaiannya, yaitu :

1. Data Penjualan

Data penjualan berisi secara umum nama-nama barang yang dipesan oleh para konsumen, lokasi dari tiap-tiap part tersebut serta kuantitas barang yang dipesan. Data ini merupakan data yang digunakan sebagai acuan dalam melakukan aktivitas *order picking*. Data yang digunakan pada penelitian ini merupakan data penjualan dari tahun 2006 dan 2007. Contoh data penjualan dapat dilihat pada tabel 3.1

Tabel 3. 1 Data penjualan

NOORDER	CUSTOMER	TGLTRX	PARTNUMBER	PARTNAME	LOCATION	ORDER
RB131J	2423	20070731	85214-0A010	RUBBER WIPER 7K2L	A01A-402	2
RB131J	2423	20070731	90430-12031	GASKET 7K	A01A-503	10
RB131J	2423	20070731	90916-03083	THERMOSTAT 7K	A01A-601	1
RB131J	2423	20070731	11213-54050	GASKET CYLINDER HEAD	A01D-501	1
RB131J	2423	20070731	23303-56031	ELEMENT FUEL BJ40	A01G-305	1
RB131J	2423	20070731	55670-0B040	REGIST A/S INST PNL	A02H-303	1
RB131J	2423	20070731	63273-95701	RETAINER KF4#,5#	A03L-601	1
RB131J	2423	20070731	15601-BZ010	ELEMENT S/A, OIL FIL	B01A-102	4
RB131J	2423	20070731	90915-TB001	FILTER OIL	B01A-103	4
RB131J	2423	20070731	90919-T1004	PLUG, SPARK	B01A-203	12
RB131J	2423	20070731	17801-05040	ELEMENT S/A AIR CLEA	B01B-101	1
RB131J	2423	20070731	90919-01059- 8N	PLUG W16EX- U	B01B-202	8
RB131J	2423	20070731	04465-BZ010	PAD KIT, DISC	B01B-203	1
RB131J	2423	20070731	90915-10003	FILTER,OIL SOLUNA	B01E-101	2
RB131J	2423	20070731	90915-20003	FILTER, OIL	B01E-203	1

				KF EFI		
RB131J	2423	20070731	17801-0C010	ELMN SA AIR CLEN FLR	B01G-202	1

2. Jarak Antar Lokasi (Part)

Dalam pengaplikasian *software* pencarian rute terpendek yang dibuat berdasarkan algoritma ACS maka diperlukan suatu data jarak terpendek antar lokasi yang terdapat pada gudang sebagai acuan utama dalam pencarian rute terpendek tersebut. Pengukuran jarak antar lokasi part dihitung dengan bantuan software AutoCad 2005 berdasarkan gambar denah lokasi gudang yang diperlihatkan pada gambar 3.9 . Sebagai contoh aplikasi dari *software* yang dibuat jarak antar lokasi yang diukur pada sub-bab ini dilakukan berdasarkan data penjualan pada sub-bab sebelumnya. Data jarak antar lokasi dapat dilihat pada tabel 3.2

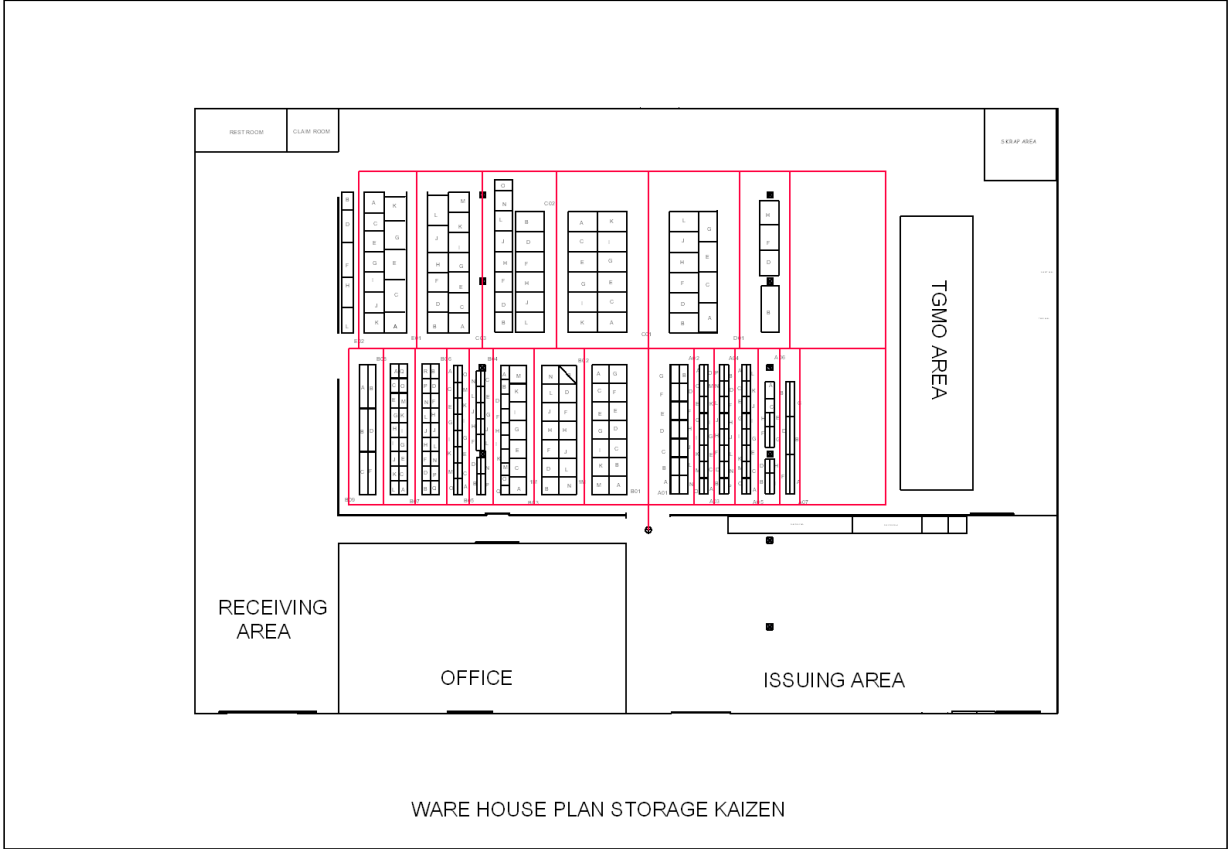
Tabel 3. 2 Jarak antar lokasi dalam satuan meter (Algoritma ACS)

Jarak	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
Depot	0.00	3.91	8.73	13.54	12.69	16.59	3.86	5.48	10.34	13.54
A01-A	3.91	0.00	4.81	9.63	12.21	16.11	0.05	1.55	6.43	9.63
A01-D	8.73	4.81	0.00	4.81	17.03	17.60	4.88	3.26	1.61	4.81
A01-G	13.54	9.63	4.81	0.00	13.19	12.79	9.69	8.08	3.20	0.0125
A02-H	12.69	12.21	17.03	13.19	0.00	13.65	12.15	13.76	16.39	13.19
A03-L	16.59	16.11	17.60	12.79	13.65	0.00	16.05	17.66	15.99	12.79
B01-A	3.86	0.05	4.88	9.69	12.15	16.05	0.00	1.61	6.48	9.68
B01-B	5.48	1.55	3.26	8.08	13.76	17.66	1.61	0.00	4.86	8.06
B01-E	10.34	6.43	1.61	3.20	16.39	15.99	6.48	4.86	0.00	3.06
B01-G	13.54	9.63	4.81	0.0125	13.19	12.79	9.68	8.06	3.06	0.00

3. Denah Lokasi

Pada penelitian ini denah lokasi berfungsi sebagai alat bantu dalam menghitung jarak antar lokasi yang terdapat di gudang. Denah lokasi dari tiap-tiap lokasi yang ada di gudang dapat dilihat pada gambar 3.9 berikut ini.

Gambar 3. 9 Denah Lokasi Gudang

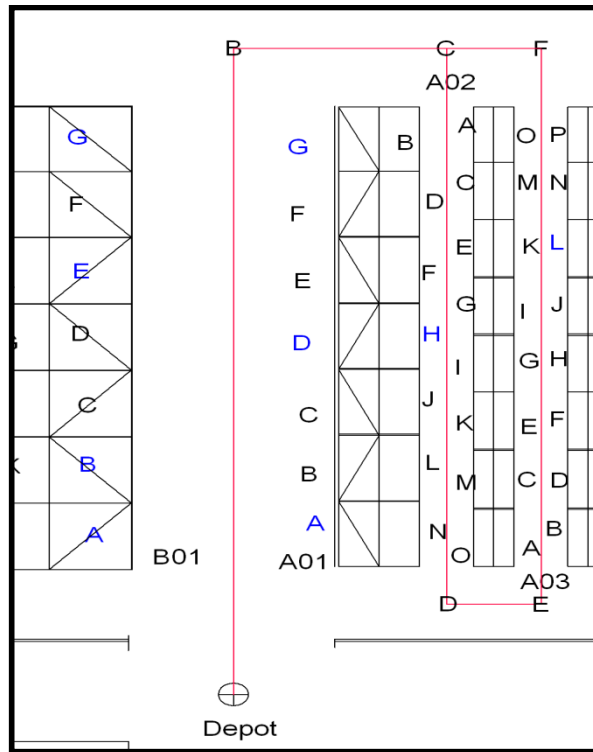


3.5 Perhitungan Jarak Rute Pengambilan Part

Pada sub-bab ini perhitungan yang dilakukan adalah perhitungan jarak rute pengambilan part dengan menggunakan strategi S-Shape dan perhitungan jarak rute pengambilan part dengan menggunakan algoritma ACS.

1. Perhitungan jarak rute pengambilan part dengan menggunakan strategi S-Shape

Strategi yang digunakan oleh perusahaan saat ini dalam melakukan aktivitas *order picking* adalah strategi S-Shape. Untuk mengukur jarak tempuh dalam melakukan aktivitas *order picking* dengan menggunakan strategi ini digunakan gambar denah lokasi gudang yang diperlihatkan pada gambar 3.2 dengan bantuan software AutoCad 2005 serta lokasi yang didatangi berdasarkan data penjualan yang ditunjukkan pada tabel 3.1 Berdasarkan gambar 3.2, jarak depot ke B = 15,75 ; B ke C = 3,95 ; C ke D = 1,75 ; D ke E = 13,56 ; E ke F = 5,7 dengan B, C, D, E, F adalah titik-titik untuk mempermudah pengukuran jarak.



Gambar 3.10 Jalur tempuh dengan menggunakan strategi S-Shape

Sehingga rute yang dipilih dengan menggunakan strategi S-Shape menempuh jarak sebagai berikut :

Jarak tempuh = (Depot, B) + (B, C) + (C, D) + (D, E) + (E, F) + (F, B) + (B, Depot)

$$= 15,75 + 3,95 + 13,56 + 1,75 + 13,56 + 5,7 + 15,75$$

$$= 70,03 \text{ meter}$$

Jika diasumsikan waktu tempuh berjalan sejauh 1 meter memakan waktu 1,2 detik maka waktu untuk berjalan menempuh jarak 70,03 meter diluar waktu pengambilan barang adalah :

$$\begin{aligned}
\text{Waktu berjalan} &= 70,03 \text{ meter} \times 1,2 \text{ detik/meter} \\
&= 84,036 \text{ detik} \\
&= 1,4 \text{ menit}
\end{aligned}$$

2. Perhitungan jarak rute pengambilan part dengan menggunakan algoritma Ant Colony System (ACS)

Sebagai contoh dalam aplikasi dari *software* Ant Picking System (APS) yang dibuat peneliti dalam menunjang pengaplikasian dari algoritma ACS, perhitungan jarak rute dari aktivitas *order picking* dengan menggunakan algoritma ACS menggunakan lokasi yang sama dengan lokasi yang dituju pada perhitungan dengan menggunakan strategi S-Shape serta jarak antar lokasi yang akan dituju berdasarkan jarak yang telah diukur yang diperlihatkan pada tabel 3.2

Pada software APS terdapat tiga tahapan dalam menghitung jarak rute terpendek dengan menggunakan algoritma Ant Colony system, yaitu:

I. Tahap pemilihan titik yang akan dituju

Pada tahap ini seorang picker yang ditempatkan pada titik r memilih untuk menuju ke titik s dengan menerapkan aturan yang ditunjukkan oleh persamaan (1) dan persamaan (2).

$$\text{temporary}(t,u) = [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta \quad i = 1,2,3,\dots,n$$

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

$$v = p_i(t, v) = \frac{[\tau(t, v)] \cdot [\eta(t, v)^\beta]}{\sum_{i=1}^n [\tau(t, u_i)] \cdot [\eta(t, u_i)^\beta]} \dots\dots\dots(2)$$

$$\eta(t, u_i) = \frac{1}{\text{jarak}(t, u_i)}$$

Contoh perhitungan :

Pada contoh perhitungan ini, titik awal lokasi picker 1 untuk menjalani turnya berawal dari lokasi DEPOT.

- a. Sebelum memasuki perhitungan pada tahap satu dalam perhitungan algoritma ACS maka terlebih dahulu dilakukan perhitungan awal untuk menghitung invers jarak

($\eta(t, v)$) antar tiap titik berdasarkan tabel 3.2 sebagai berikut :

$$\eta(t, v) = \frac{1}{\text{jarak}(t, v)}$$

Contoh perhitungan $\eta(t, v)$ pada titik $\eta(\text{depot}, A02 - H)$:

$$\eta(\text{depot}, A02 - H) = \frac{1}{\text{jarak}(\text{depot}, A02 - H)} = \frac{1}{12,69} = 0,07880$$

Hasil keseluruhan dari invers jarak ($\eta(t, v)$) dapat dilihat pada tabel 3.3 dibawah ini.

Tabel 3. 3 Invers jarak ($\eta(t, v)$)

Jarak	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
Depot	0.00000	0.25575	0.11455	0.07386	0.07880	0.06028	0.25907	0.18248	0.09671	0.07386
A01-A	0.25575	0.00000	0.20790	0.10384	0.08190	0.06207	20.00000	0.64516	0.15552	0.10384
A01-D	0.11455	0.20790	0.00000	0.20790	0.05872	0.05682	0.20492	0.30675	0.62112	0.20790
A01-G	0.07386	0.10384	0.20790	0.00000	0.07582	0.07819	0.10320	0.12376	0.31250	100.00000
A02-H	0.07880	0.08190	0.05872	0.07582	0.00000	0.07326	0.08230	0.07267	0.06101	0.07582
A03-L	0.06028	0.06207	0.05682	0.07819	0.07326	0.00000	0.06231	0.05663	0.06254	0.07819
B01-A	0.25907	20.00000	0.20492	0.10320	0.08230	0.06231	0.00000	0.62112	0.15432	0.10331
B01-B	0.18248	0.64516	0.30675	0.12376	0.07267	0.05663	0.62112	0.00000	0.20576	0.12407
B01-E	0.09671	0.15552	0.62112	0.31250	0.06101	0.06254	0.15432	0.20576	0.00000	0.32680
B01-G	0.07386	0.10384	0.20790	100.00000	0.07582	0.07819	0.10331	0.12407	0.32680	0.00000

Nilai dari semua pheromone (τ) pada awal perhitungan ditetapkan dengan angka awal yang sangat kecil. Pada contoh perhitungan penelitian ini nilai pheromone awal menggunakan nilai τ 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 3.4 dibawah ini.

Tabel 3.4 pheromone (τ) awal

τ	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
Depot	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
A01-A	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
A01-D	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
A01-G	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
A02-H	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
A03-L	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
B01-A	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
B01-B	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
B01-E	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001
B01-G	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001	0,0001

b. Tahap pemilihan titik yang akan dituju

Dalam pemilihan titik selanjutnya yang dituju, pertama-tama dilakukan penetapan dari nilai $\beta \geq 0$ adalah parameter perhitungan untuk mendapatkan nilai yang optimal dalam ACS, untuk mempermudah perhitungan diambil nilai $\beta = 2$. Selanjutnya dilakukan perhitungan untuk mendapatkan nilai *temporary* (t,u) berdasarkan persamaan (1) serta nilai probabilitas berdasarkan persamaan (2) dari titik awal DEPOT (t) ke titik selanjutnya yang belum dilalui (u). Nilai *temporary* digunakan untuk menentukan titik-titik yang akan dituju selanjutnya. Contoh perhitungan serta hasil perhitungan nilai *temporary* dan nilai probabilitas dapat dilihat sebagai berikut :

$$\text{temporary (t,u)} = [\tau(t, u_i)] \cdot [\eta(t, u_i)]^\beta \quad i = 1, 2, 3, \dots, n$$

$$\begin{aligned} \text{temporary (depot, B01-G)} &= [\tau(\text{depot}, B01G)] \cdot [\eta(\text{depot}, B01-G)]^2 \\ &= [0,0001] \cdot [0.07386]^2 \\ &= 0,0545 \times 10^{-5} \end{aligned}$$

$$\text{Probabilitas (r,u)} = \frac{[\tau(t, v)] \cdot [\eta(t, v)]^\beta}{\sum_{i=1}^n [\tau(t, u_i)] \cdot [\eta(t, u_i)]^\beta}$$

$$\begin{aligned} \text{Probabilitas (depot, B01-G)} &= \frac{0,0545 \times 10^{-5}}{2,09 \times 10^{-5}} \\ &= 0.0261 \end{aligned}$$

Hasil perhitungan *temporary* dan probabilitas dari titik awal DEPOT dapat dilihat pada tabel 3.5 dibawah ini.

Tabel 3.5 Hasil perhitungan *temporary* dan probabilitas dari titik awal DEPOT

(depot)	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
<i>Temporary</i> (x 10 ⁻⁵)	0	0.6541	0.1312	0.0545	0.0621	0.0363	0.6712	0.333	0.0935	0.0545
Probabilitas	0	0.3129	0.0628	0.0261	0.0749	0.0174	0.3210	0.1593	0.0447	0.0261
Probabilitas akumulatif	0	0.3129	0.3757	0.401	0.4314	0.4488	0.7698	0.9291	0.9739	1

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 (q_0) antara 0 sampai 1. Pada perhitungan ini ditetapkan nilai q_0 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 eksplorasi 10% (Bauer,n.d). Karena $q \leq q_0$, maka penentuan lokasi yang akan dituju berdasarkan persamaan (1), yaitu dengan melihat hasil temporary yang paling besar. Sehingga lokasi yang terpilih adalah lokasi **B01-A**.

II. Tahap pembaharuan *pheromone* (τ) lokal

Setelah picker berpindah menuju lokasi selanjutnya maka tahap selanjutnya adalah melakukan pembaharuan *pheromone* (τ) secara lokal dengan menggunakan persamaan (3). Persamaan dari pembaharuan *pheromone* (τ) lokal, contoh perhitungan serta hasil perhitungan dapat dilihat sebagai berikut :

$$\tau(t, v) \leftarrow (1 - \rho) \cdot \tau(t, v) + \rho \cdot \Delta\tau(t, v)$$

$$\Delta\tau(t, v) = \frac{1}{L_{nn} \cdot c}$$

dimana :

L_{nn} = panjang tur yang diperoleh

c = jumlah lokasi

ρ = parameter dengan nilai 0 sampai 1

$\Delta\tau$ = perubahan *pheromone*

Contoh perhitungan :

Dalam memperbaharui pheromone secara lokal dibutuhkan suatu parameter (ρ) yang memiliki nilai antara 0 sampai 1. Pada perhitungan ini nilai ρ ditetapkan dengan nilai sebesar 0,1. Contoh perhitungan serta hasil perhitungan dapat dilihat sebagai berikut :

$$\Delta\tau(\text{depot}, B01-A) = \frac{1}{3,86 \cdot 10}$$

$$= 0,0259$$

$$\tau(\text{depot}, B01-A) \leftarrow (1-\rho) \cdot \tau(\text{depot}, B01-A) + \rho \cdot \Delta\tau(\text{depot}, B01-A)$$

$$\tau(\text{depot}, B01-A) \leftarrow (1-0,1) \cdot 0,0001 + 0,1 \cdot \Delta\tau(\text{depot}, B01-A)$$

$$\tau(\text{depot}, B01-A) \leftarrow 0,00268$$

Hasil pembaharuan pheromone (τ) lokal untuk $\tau(\text{depot}, B01-A)$ dapat dilihat pada tabel 3.6 dibawah ini dengan tulisan yang dicetak miring.

τ	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
Depot	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	<i>0,00268</i>	0,00010	0,00010	0,00010
A01-A	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
A01-D	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
A01-G	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
A02-H	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
A03-L	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
B01-A	<i>0,00268</i>	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
B01-B	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
B01-E	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010
B01-G	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010	0,00010

Tabel 3. 6 Nilai pheromone (τ) setelah mengalami pembaharuan lokal

untuk $\tau(\text{depot}, B01-A)$

Dengan proses yang sama hasil keseluruhan dari pembaharuan pheromone lokal dari semua picker dapat dilihat pada tabel 3.7 dibawah ini.

Tabel 3. 7 Nilai pheromone (τ) setelah tahap mengalami pembaharuan pheromone lokal dari semua picker

τ	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
Depot	0.00010	0.00010	0.00010	0.00010	0.00088	0.00010	0.00268	0.00010	0.00010	0.00010
A01-A	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.20009	0.00654	0.00010	0.00010
A01-D	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00316	0.00010	0.00010
A01-G	0.00010	0.00010	0.00010	0.00010	0.00010	0.00087	0.00010	0.00010	0.00630	1.00009
A02-H	0.00088	0.00010	0.00010	0.00010	0.00010	0.00082	0.00010	0.00010	0.00010	0.00010
A03-L	0.00010	0.00010	0.00010	0.00087	0.00082	0.00010	0.00010	0.00010	0.00010	0.00010
B01-A	0.00268	0.20009	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010
B01-B	0.00010	0.00654	0.00316	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010	0.00010
B01-E	0.00010	0.00010	0.00010	0.00630	0.00010	0.00010	0.00010	0.00010	0.00010	0.00336
B01-G	0.00010	0.00010	0.00010	1.00009	0.00010	0.00010	0.00010	0.00010	0.00336	0.00010

III. Tahap pembaharuan *pheromone* (τ) global

Setelah tahap 1 dan 2 telah selesai untuk mendapatkan satu rute dan setiap lokasi yang dikunjungi telah mengalami pembaharuan pheromone (τ) secara lokal, maka tahap selanjutnya adalah untuk membaharui pheromone (τ) secara global berdasarkan persamaan (4) namun hanya lokasi yang menghasilkan rute dengan jarak terpendek. Persamaan dari pembaharuan pheromone (τ) global, contoh perhitungan serta hasil perhitungan dapat dilihat sebagai berikut :

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

$$\Delta\tau(t,v) = \begin{cases} L_{gb}^{-1} & \text{jika}(t,v) \in \text{tur_terbaik} \\ 0 & \end{cases}$$

Dimana :

$\tau(t,v)$ = nilai pheromone akhir setelah mengalami pembaharuan lokal

L_{gb} = panjang jalur terpendek pada akhir siklus

α = parameter dengan nilai antara 0 sampai 1

$\Delta\tau$ = perubahan *pheromone*

Contoh perhitungan:

Setelah *picker* 1 pada iterasi 1 telah melewati tahap I dan tahap II, maka rute yang dihasilkan adalah **DEPOT, B01-A, A01-A, B01-B, A01-D, B01-E, B01-G, A01-G, A03-L, A02-H** dan kembali ke lokasi **DEPOT**. Dari rute tersebut didapat panjang jalur sebesar 52,53 m dan merupakan panjang jalur terpendek pada iterasi pertama. Maka pembaharuan pheromone-nya adalah sebagai berikut.

$$\alpha = 0,1$$

$$L_{gb} = 52,53$$

Nilai pheromone akhir =

- Untuk (t,v) bagian dari rute terpendek

$$\Delta\tau(t,v) = L_{gb}^{-1} = (52,53)^{-1}$$

$$= 0,019$$

Sebagai contoh digunakan pembaharuan pheromone global untuk pheromone $\tau(\text{depot}, B01-A)$:

$$\tau(\text{depot}, B01-A) \leftarrow (1-\alpha) \cdot \tau(\text{depot}, B01-A) + \alpha \cdot \Delta\tau$$

$$\tau(\text{depot}, B01-A) \leftarrow (1-0,1) \cdot (0,00268) + (0,1 \cdot 0,019)$$

$$\tau(\text{depot}, B01-A) \leftarrow 0,00431$$

- Untuk (t,v) bagian dari rute terpendek

$$\Delta\tau(t,v) = 0$$

Sebagai contoh digunakan pembaharuan pheromone global untuk pheromone $\tau(\text{Depot}, B01-B)$:

$$\tau(\text{Depot}, B01-B) \leftarrow (1-\alpha) \cdot \tau(\text{Depot}, B01-B) + \alpha \cdot \Delta\tau$$

$$\tau(\text{Depot}, B01-B) \leftarrow (1-0,1) \cdot (0,0001) + (0,1 \times 0)$$

$$\tau(\text{Depot}, B01-B) \leftarrow 0,00009$$

Hasil pembaharuan pheromone (τ) global dapat dilihat pada tabel 3.8 berikut ini.

Tabel 3.8 Nilai pheromone (τ) setelah mengalami pembaharuan global

τ	Depot	A01-A	A01-D	A01-G	A02-H	A03-L	B01-A	B01-B	B01-E	B01-G
Depot	0.00009	0.00009	0.00009	0.00009	0.00256	0.00009	0.00431	0.00009	0.00009	0.00009
A01-A	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.18185	0.00766	0.00009	0.00009
A01-D	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00461	0.00744	0.00009
A01-G	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00467	0.90185
A02-H	0.00256	0.00009	0.00009	0.00009	0.00009	0.00251	0.00009	0.00009	0.00009	0.00009
A03-L	0.00009	0.00009	0.00009	0.00009	0.00251	0.00009	0.00009	0.00009	0.00009	0.00255
B01-A	0.00418	0.18185	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009
B01-B	0.00009	0.00766	0.00461	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009
B01-E	0.00009	0.00009	0.00744	0.00467	0.00009	0.00009	0.00009	0.00009	0.00009	0.00009
B01-G	0.00009	0.00009	0.00009	0.90185	0.00009	0.00255	0.00009	0.00009	0.00009	0.00009

3.6 Desain Program

Desain yang dilakukan pada penelitian ini meliputi desain pada proses *software*, desain database yang digunakan oleh *software* serta interface dari *software* yang dibuat dalam menunjang pengaplikasian dari algoritma ACS.

3.6.1 Proses Software

Pada tugas akhir ini digunakan *Data Flow Diagram* (DFD) untuk membantu dalam mengidentifikasi dan menganalisis proses dalam sistem baik secara fisik maupun logikanya.

DFD adalah suatu alat bantu yang digunakan untuk menggambarkan tata laksana suatu sistem dimana tata laksana yang digambarkan dapat berupa suatu sistem baru atau sistem lama yang akan dikembangkan. Adapun kegunaan dari aliran data ini adalah:

- Sebagai alat analisa data
- Sebagai alat komunikasi antara sistem analisa dengan pemakai
- Sebagai alat dokumentasi

Terdapat 2 tipe DFD, antara lain:

1. Context Diagram, merupakan diagram tingkat atas, yaitu diagram paling tidak detail dari sebuah sistem informasi yang menggambarkan aliran-aliran kedalam atau keluar entitas eksternal.
2. DFD Level 0, yaitu system pencarian rute (hasil

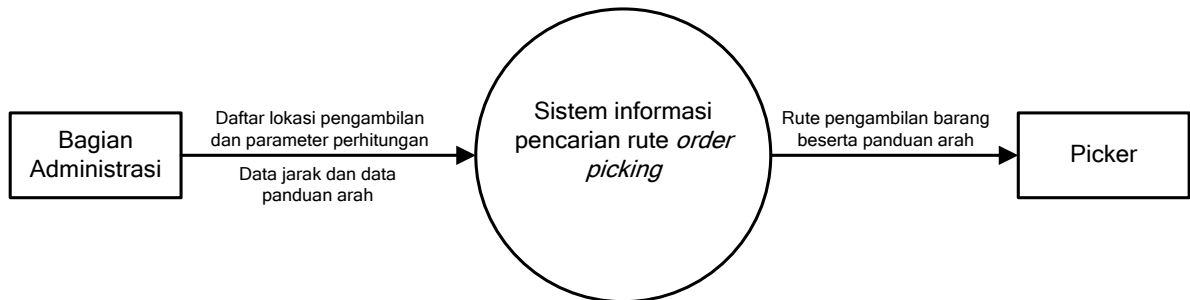
DFD Level 0 dibagi menjadi 2, yaitu :

- a. DFD Fisik, representasi grafik dari sebuah sistem yang menunjukkan entitas-entitas eksternal dan internal dari sistem tersebut dan aliran-aliran data ke dalam atau keluar dari entitas tersebut.
- b. DFD Logis, representasi grafik dari sebuah sistem yang menunjukkan proses-proses dalam sistem tersebut dan aliran-aliran data kedalam atau keluar.

3.6.2 Diagram Konteks

Dalam sistem pencarian rute pada aktivitas *order picking*, aktivitas utamanya adalah pencarian rute terpendek dimana sistem ini memiliki input berupa daftar lokasi, data jarak, data panduan arah serta parameter-parameter yang digunakan dalam perhitungan pencarian rute. Output dari sistem ini berupa laporan urutan pengambilan barang beserta panduan arahnya yang akan digunakan

oleh picker dalam melakukan aktivitasnya. Diagram konteks sistem pencarian rute dapat dilihat pada gambar 3.11 dibawah ini.



Gambar 3.11 Diagram konteks sistem pencarian rute

3.6.3 DFD level 0

Pada gambar 3.12 dibawah terlihat Pada DFD level 0 terdapat tujuh proses yang menunjang sistem pencarian rute. Proses yang pertama adalah proses update data lokasi. Proses ini memiliki input berupa lokasi-lokasi baru dari part-part yang mengalami perubahan maupun terdapat part baru yang sebelumnya belum mempunyai lokasi. Output dari proses ini adalah daftar lokasi baru dimana output tersebut akan disimpan dalam basis data yang bernama data lokasi.

Proses yang kedua adalah proses update data arah. Proses ini memiliki input berupa panduan arah untuk menuju dari satu lokasi ke lokasi yang lain. Proses ini dilakukan jika tata letak pada gudang mengalami perubahan sehingga posisi lokasi yang lama berkemungkinan mengalami perubahan lokasi. Output dari proses ini yang berupa panduan arah yang baru disimpan dalam basis data yang bernama basis data arah.

Proses yang ketiga adalah proses update data jarak. Proses ini memiliki input berupa jarak dari satu lokasi ke lokasi yang lain. Proses ini dilakukan jika tata letak pada gudang mengalami perubahan sehingga posisi lokasi yang lama berkemungkinan mengalami perubahan lokasi. Output dari proses ini yang berupa data jarak yang baru disimpan dalam basis data yang bernama basis data jarak.

Proses yang keempat adalah proses input lokasi. Aktivitas dari proses ini dilakukan penentuan lokasi yang akan dituju oleh para picker. Input dari proses ini berupa daftar lokasi yang akan dituju oleh picker sesuai dengan order dari konsumen. Output dari proses ini adalah lokasi yang akan dituju disimpan dalam basis data yang bernama basis data perhitungan.

Proses yang kelima adalah proses input parameter perhitungan. Input dari proses ini berupa parameter-parameter perhitungan. Output dari proses ini adalah berupa parameter-parameter perhitungan yang akan digunakan dalam proses perhitungan pencarian rute. Output dari proses ini akan disimpan dalam basis data yang bernama basis data perhitungan.

Proses yang keenam adalah proses perhitungan pencarian rute. Aktivitas dari proses ini merupakan aktivitas yang paling utama dalam sistem informasi ini. Input dari proses ini didapat dari basis data jarak dan dari basis data perhitungan. Output dari proses ini adalah berupa rute pengambilan dan jarak yang ditempuh, dimana output dari proses ini akan digunakan sebagai input pada proses selanjutnya.

Proses yang ketujuh adalah proses pembuatan laporan pengambilan. Input dari proses ini diambil dari proses sebelumnya yaitu proses perhitungan pencarian rute dan basis data arah. Output dari proses ini berupa laporan yang berisi urutan pengambilan barang serta dilengkapi panduan arah yang digunakan oleh para picker dalam membantu dari aktivitas picker tersebut.

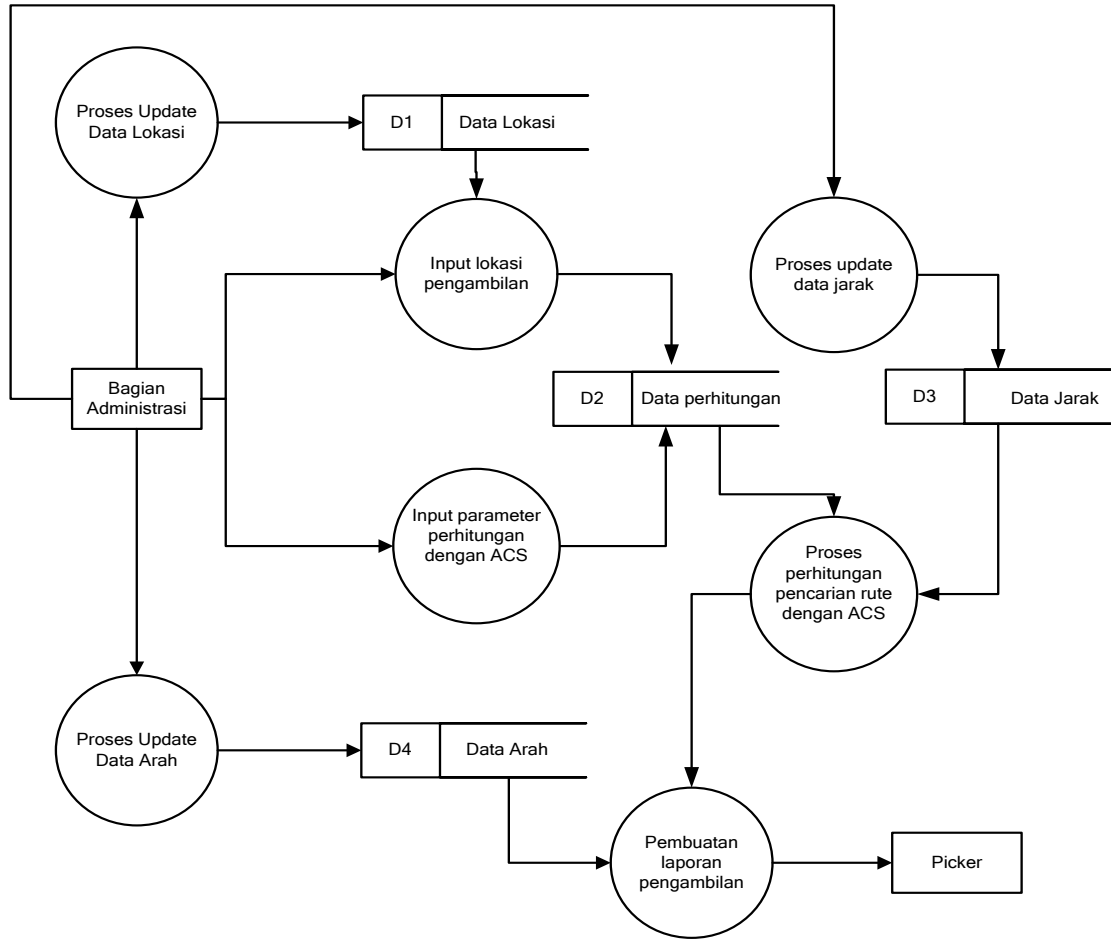
3.6.4 Data Base

3.6.4.1 Entity Relationship Diagram

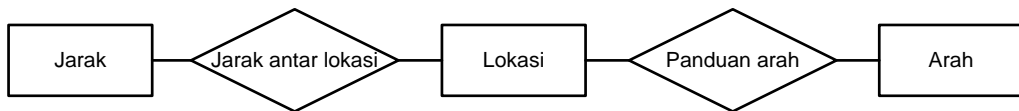
Sebelum melakukan penyusunan basis data perlu dibuat *Entity Relationship Diagram* (ERD) dari entitas yang terlibat dalam sistem pencarian rute ini. *Entity – Relationship* adalah suatu hubungan antara dua file atau lebih yang saling berkaitan. Entitas yang saling berhubungan antara satu dengan yang lain akan membentuk suatu relasi dan isi masing-masing entitas tersebut saling melengkapi.

ERD adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DFD). ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilaksanakan. ERD menggunakan notasi dan sumber untuk menggambarkan struktur dan hubungan antar data.

ERD dapat digambarkan pada gambar 3.13 berikut.



Gambar 3.12 DFD level 0 sistem pencarian rute



Gambar 3.13 ERD Database System

3.6.4.2 Transformasi Model Data ke Basis Data Fisik

Berikut ini adalah transformasi model data yang dinyatakan dalam ERD ke dalam basis data fisik. ERD yang berupa himpunan entitas dan relasi ditransformasikan menjadi tabel-tabel yang merupakan komponen utama pembentuk basis data. Selanjutnya, atribut-atribut yang melekat pada masing-masing himpunan entitas dan relasi dinyatakan sebagai *field-field* dari tabel yang sesuai. Tabel-tabel tersebut dapat dilihat sebagai berikut.

1. Tabel Grid Jarak

Tabel grid jarak yang memiliki nama file TGridJarak digunakan dalam proses perhitungan rute. Fungsi dari tabel ini adalah sebagai tabel bantuan dalam mencatat pembaharuan pheromone. Pada tabel 3.11 berikut diperlihatkan field –field dari tabel grid jarak.

Tabel 3.9 Tabel TGridJarak

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Node	50	Text	Nama lokasi
2	Pheromone	Single	Number	Nilai pheromone
3	Jarak	Single	Number	Jarak antar lokasi

2. Tabel Grid Probabilitas

Tabel grid probabilitas yang memiliki nama file TGridProbabilitas yang dalam proses perhitungan rute. Fungsi dari tabel ini adalah sebagai tabel

bantuan dalam mencatat perhitungan probabilitas dan akumulasi dari probabilitas yang berguna untuk memilih lokasi selanjutnya yang akan dituju. Pada tabel 3.10 berikut diperlihatkan field –field dari tabel grid probabilitas.

Tabel 3.10 Tabel TGridProbabilitas

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Node	50	Text	Nama lokasi
2	Probabilitas	Single	Number	
3	Probabilitas akumulatif	Single	Number	

3. Tabel Hasil

Tabel hasil yang memiliki nama file THasil digunakan dalam proses perhitungan rute. Fungsi dari tabel ini adalah sebagai tabel bantuan dalam mencatat hasil perhitungan berupa urutan lokasi yang harus dituju. Pada tabel 3.11 berikut diperlihatkan field –field dari tabel hasil

Tabel 3.11 Tabel THasil

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Nomor	50	Text	
2	Node	50	Text	Nama lokasi

4. Tabel Hasil Text

Tabel hasil text yang memiliki nama file THasilText digunakan dalam pembuatan laporan pengembalian barang. Fungsi dari tabel ini adalah sebagai tabel bantuan dalam mengkonversi hasil perhitungan berupa urutan lokasi yang harus dituju yang berupa tabel menjadi dalam bentuk teks. Pada tabel 3.12 berikut diperlihatkan field –field dari tabel hasil text.

Tabel 3.12 Tabel THasilText

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Hasil	50	Text	Hasil rute yang terpilih

5. Tabel Hasil Urut

Tabel hasil urut yang memiliki nama file THasilUrut digunakan dalam pembuatan laporan pengembalian barang. Fungsi dari tabel ini adalah sebagai tabel yang berisi hasil perhitungan berupa urutan lokasi yang harus dituju yang telah diurutkan agar berawal dari lokasi depot. Pada tabel 3.13 berikut diperlihatkan field –field dari tabel hasil urut.

Tabel 3. 13Tabel THasilUrut

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Nomor	50	Text	
2	Node	50	Text	Nama lokasi

6. Tabel Jarak Node

Tabel jarak node yang memiliki nama file TJarakNode digunakan dalam perhitungan pengambilan barang. Fungsi dari tabel ini adalah sebagai pemberi informasi mengenai jarak dari satu lokasi ke lokasi yang lain. Pada tabel 3.14 berikut diperlihatkan field –field dari tabel jarak node.

Tabel 3. 14 Tabel TJarakNode

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Node	50	Text	Nama lokasi
2	Jarak	Single	Number	Jarak antar lokasi

7. Tabel Node Awal

Tabel node awal yang memiliki nama file TNodeAwal digunakan dalam perhitungan pengambilan barang. Fungsi dari tabel ini adalah sebagai pemberi informasi mengenai daftar nama-nama lokasi yang terdapat di gudang. Pada tabel 3.15 berikut diperlihatkan field –field dari tabel node awal.

Tabel 3.15 Tabel TNodeAwal

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Node	50	Text	Nama lokasi

8. Tabel Node

Tabel node yang memiliki nama file TNode digunakan dalam pembuatan laporan pengembalian barang. Fungsi dari tabel ini adalah sebagai tabel bantuan dalam mengurutkan hasil perhitungan berupa urutan lokasi yang harus dituju agar berawal dari lokasi depot. Pada tabel 3.16 berikut diperlihatkan field –field dari tabel node.

Tabel 3.16 Tabel TNode

No.	Nama Field	Ukuran Field	Tipe Data	Keterangan
1	Node	50	Text	Nama lokasi

3.6.5 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 System*. 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 System* dengan cara mengetikkannya pada label-label yang telah disediakan. Nilai-nilai parameter yang harus diisi oleh user adalah nilai pheromone awal, nilai q_0 , nilai beta, nilai rho, nilai alpha, dan jumlah iterasi perhitungan yang diinginkan.

Form input ditunjukkan pada gambar 3.14 berikut.

Gambar 3.14 Input

MENCARI JARAK TERPENDEK DENGAN ALGORITMA SEMUT

TENTUKAN JUMLAH NODE:

JUMLAH KOMBINASI NODE:

PILIH NODE YG DIINGINKAN:

DAFTAR NODE:

Node
Depot
A01-A
A01-D
A01-G
A02-H
A03-L

DAFTAR JARAK ANTAR NODE:

Node	Pheromone	Jarak
Depot-B01-G	0.0001	13.54
Depot-B01-E	0.0001	10.34
Depot-B01-B	0.0001	5.48
Depot-B01-A	6.913302E-03	3.86
Depot-A03-L	0.0001	16.59
Depot-A02-H	2.836086E-03	12.69

NILAI PHEROMONE AWAL:

NILAI PEMBANDING RANDOM (q_0): (0 - 1)

NILAI KONSTANTA BETA: > 1

NILAI KONSTANTA RHO: (0 - 1)

NILAI KONSTANTA ALPHA: (0 - 1)

SET

JUMLAH ITERASI:

PROSES

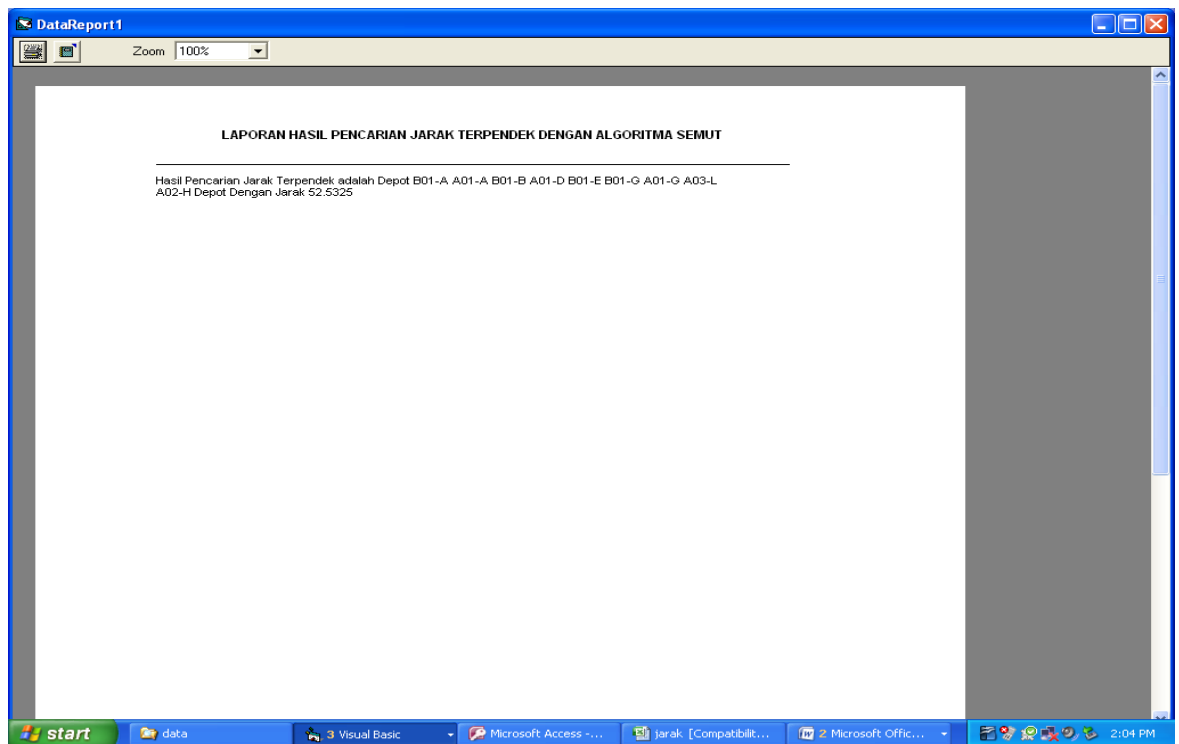
RESET PADUKAN KELUAR

- 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 3.15 berikut ini

Gambar 3.15 Output



Pada tugas akhir ini perhitungan dengan menggunakan Software APS(*software* Ant Picking System) dilakukan dengan menggunakan jumlah iterasi sebanyak 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, dan 25 dimana tiap jumlah iterasi dilakukan pengulangan perhitungan sebanyak 10 kali. Hasil perhitungan secara lengkap dapat dilihat pada tabel 3.17 berikut.

Tabel 3.17 Tabel Hasil Perhitungan software APS

Iterasi Run	1	3	5	7	10	13	15	18	22	25
1	52.5325	52.5325	52.5325	52.6725	52.5325	52.5325	52.5325	52.6725	52.5325	52.5325
2	52.6725	52.6725	52.5325	52.6725	52.6725	52.6725	52.6725	52.5325	52.5325	52.5325
3	52.532	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325
4	52.532	52.532	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325
5	52.6725	52.6725	52.5325	52.6725	52.6725	52.6725	52.5325	52.5325	52.6725	52.6725
6	52.6725	52.6725	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325
7	52.6725	52.6725	52.6725	52.6725	52.6725	52.5325	52.6725	52.6725	52.5325	52.5325
8	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325
9	52.5325	52.5325	52.5325	52.5325	52.5325	52.6725	52.6725	52.5325	52.5325	52.5325
10	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325	52.5325
rata-rata	52.5884	52.58845	52.5465	52.5885	52.5745	52.5745	52.5745	52.5605	52.5465	52.5465

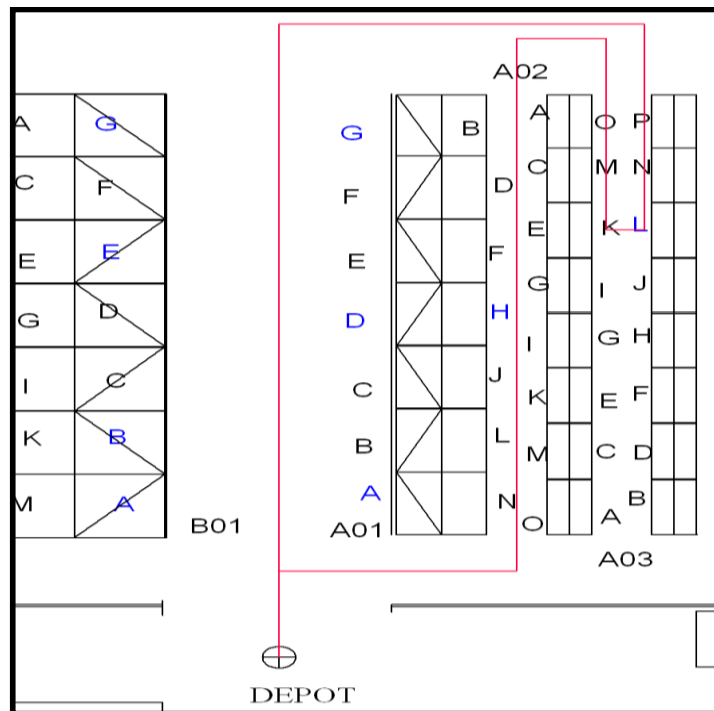
Berdasarkan hasil perhitungan pada tabel 3.17 didapat hasil rute dengan jarak terpendek sejauh 52,5325 m dengan jalur tempuh yang diperlihatkan pada gambar 3.16. Jika diasumsikan waktu tempuh berjalan sejauh 1 meter memakan waktu 1,1 detik maka waktu untuk berjalan menempuh jarak 52,5325 meter diluar waktu pengambilan barang adalah :

$$\text{Waktu berjalan} = 52,5325 \text{ meter} \times 1,2 \text{ detik/meter}$$

$$= 63,036 \text{ detik}$$

$$= 1,05 \text{ menit}$$

Gambar jalur yang ditempuh dapat dilihat pada gambar 3.16 berikut ini.



Gambar 3.16 Jalur tempuh dengan menggunakan algoritma ACS

Berdasarkan hasil perhitungan jarak tempuh antara strategi S-Shape dengan algoritma ACS terlihat bahwa jarak tempuh terpendek didapat melalui perhitungan dengan menggunakan algoritma ACS yaitu dapat memangkas jarak tempuh sejauh 17,5 meter seta memangkas waktu berjalan selama 21 detik dibandingkan dengan jarak tempuh dengan strategi S-Shape.

Tabel 3.18 Perbandingan Jarak Tempuh antara strategi S-Shape dengan ACS

Aspek Perbandingan	S-Shape	ACS
Jarak Tempuh (meter)	70.03	52,53
Waktu Berjalan (detik)	84,036	63,036

Hasil rute dengan menggunakan software APS didapat jarak tempuh yang optimal sejauh 52.53 meter. Dalam penggunaan jumlah iterasi kurang dari 10 rata-rata hasil jarak yang didapatkan sebesar 52,57 meter namun untuk penggunaan

jumlah iterasi 10 dan lebih dari 10 didapat rata-rata hasil jarak sebesar 52,54 meter. Untuk itu dalam mendapatkan hasil yang optimal dibutuhkan jumlah iterasi dengan batas minimal sebanyak 10 iterasi.

BAB IV

KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan tentang kesimpulan yang didapat berdasarkan hasil pengolahan data dan analisis dari bab sebelumnya serta saran-saran yang diberikan oleh peneliti untuk penelitian selanjutnya.

4.1 Kesimpulan

Penelitian dalam tugas akhir ini memberikan beberapa kesimpulan sebagai berikut:

1. Penggunaan strategi S-Shape yang oleh PT Eka Jaya Motor sekarang kurang efisien dalam mengurangi jarak tempuh aktivitas *order picking* dibandingkan strategi usulan dengan menggunakan algoritman Ant Colony System. Pemilihan rute dengan menggunakan algoritma Ant Colony System menghasilkan rute dengan jarak tempuh sejauh 52,53 meter dengan waktu berjalan selama 63,036 detik sedangkan pemilihan rute dengan Strategi S-Shape dihasilkan rute dengan jarak tempuh sejauh 70,03 meter dengan waktu berjalan selama 84,36 detik.
2. Algoritma ant colony system menggunakan fungsi heuristik untuk mendapatkan hasil yang optimal sehingga kekurangan dari algoritma ant colony system ini adalah waktu proses dalam mendapatkan hasil yang paling optimal sangat tergantung dari jumlah iterasi perhitungan yang digunakan.

3. Penggunaan software Ant Picker System dalam pencarian rute pada aktivitas *order picking* memiliki fungsi utama dalam pemberian petunjuk urutan lokasi yang harus dikunjungi oleh picker tersebut sehingga para picker tidak perlu lagi berfikir untuk mendapatkan rute yang terpendek setiap mereka melakukan aktivitas *order picking*.
4. Kekurangan dari penggunaan software Ant Picker System adalah belum tersedianya fasilitas untuk meng-*update* data nama lokasi, panduan arah, serta jarak antar lokasi secara langsung namun harus menggunakan bantuan software lain yaitu microsoft access untuk meng-*update* data-data tersebut.

4.2 Saran

Berikut ini adalah beberapa saran perbaikan dan pengembangan bagi perusahaan dan penelitian selanjutnya:

1. Pengembangan sistem informasi agar dapat terhubung dengan sistem informasi penerimaan order sehingga input lokasi tidak perlu dilakukan secara manual.
2. Pengembangan sistem informasi pencarian rute agar dapat digunakan tidak hanya sebagai pencarian rute pada aktivitas *order picking* namun pada pencarian rute lainnya seperti rute pendistribusian barang dan sebagainya.

Daftar Pustaka

1. Frazelle, Edward (2002), *World-Class Warehousing and Material Handling*, McGraw Hill, Inc, Singapore.
2. Ibnu Sina Wardy, *Penggunaan Graf dalam Algoritma Semut untuk Melakukan Optimisasi*, ITB Bandung.
3. J. Wilson, Robin and J. Watkins, John (1992), *GRAPH*. University Press IKIP Surabaya.
4. M. Dorigo dan L. M. Gambardella (1997), *Ant Colonies for the Traveling Salesman Problem*, Cambridge, Massachusetts. London, England
5. Meyers, Fred E (1993), *Plant Layout and Material Handling*, Regests/ Prentice Hall.
6. Mulcahi, David E (1994), *Warehouse Distribution and Operations Handbook*, McGraw Hill, Inc, Singapore.
7. Petersen, Charles G II and Schmenner, Roger W (1999), *An Evaluation of Routing and Volume-Based Storage Policies In An Order picking Operation*, Decision Sciense.
8. R, Jacques and R, Angel (2007), *Improving Product Location and Order picking Activities In A Distribution Center*.
9. R, Rina dan M, A. Beny. *Solusi Optimal Travelling Salesman Problem dengan Menggunakan Ant Colony System (ACS)*. ITB Bandung
10. Shouman, M A et.al (2005), *Comprehensive Survey And Classification Scheme of Warehousing System*, Proceedings of the 2005 International Conference on Simulating and Modelling.
11. Wignjosoebroto, Sritomo (2000), *Tata Letak Pabrik dan Pemandahan Bahan*, Guna Widya, Surabaya.