THE APPLICATION OF SEARCHING CLOSEST DISTANCE (TRAVELLING SALESMEN PROBLEM) USING ANT COLONY OPTIMIZATITON ALGORITHM (ACO) (A CASE STUDY: PT. NUSANTARA ABADI)

Panji Novantara[1], Muhamad Usep Sapari Septiana[2]
[1,2]Universitas Kuningan
Jl.Cut Nyak Dhien no.36A kuningan
panji@uniku.ac.id[1]

ABSTRACT

*Traveling from one place or city to another by considering the efficiency of time and costs requires accuracy in determining the path between the nearby cities. The results need speed and accuracy by using computers. Additionally the service application system is very supportive towards the mobility of users. This condition also happens in the distribution of mineral water. PT. Abadi Nusantara produces several sizes Nalala drinking water circulating in the community, including 19 liter gallon of bottled water, bottled water 500 milliliter, and bottled water 250 milliliter glass. The courier distributors that send the goods, have to consider the time and cost efficiencies when they conduct distribution activities, then sought the shortest distance from their distribution journey. There are various theories of algorithms that can be used to solve the problems of goods distribution or it is often called Travelling Salesmen Problem (TSP), such as brute force algorithm, greedy algorithm, ant colony algorithm, Djikstra, A *, etc. In this thesis, the author chooses one of the algorithms, it is ant colony algorithm to solve the problem of distribution.*

Keywords : Searching Closest Distance, Traveling Salesman Problem, Ant Colony Optimization Algorithm.

1. PENDAHULUAN

1.1 Latar Belakang

PT. Nusantara Abadi adalah produk lokal yang bergerak pada sektor air mineral. Produknya mudah kita jumpai di toko-toko baik itu toko besar maupun toko yang kecil. PT. Nusantara Abadi ini mempunyai resort di Kabupaten Kuningan sebagai produksi air mineral, khusus untuk menangani distribusi produk-produknya. Dengan cakupan yang luas para distributor di PT. Nusantara Abadi ini mendistribusikan produk-nya, tetapi sayangnya belum ada implementasi metode untuk pencarian jarak terdekat yang dapat mengoptimasi jarak tempuh para kurir untuk mendistribusikan produknya.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang ada, maka penulis merumuskan masalah dalam penelitian ini sebagai berikut :

1. Bagaimana membangun aplikasi pencarian jarak terdekat pendistribusian air mineral PT. Nusantara Abadi ?
2. Bagaimana mengimplementasikan metode *Ant Colony Optimization* pada aplikasi pencarian jarak terdekat pendistribusian air mineral PT. Nusantara Abadi ?

1.3 Batasan Masalah

Adapun batasan permasalahan dalam penelitian ini adalah sebagai berikut :

1. Media yang digunakan untuk menggambarkan lokasi titik dan peta menggunakan *drawing canvas* sebagai pengganti peta, dan hasil berupa graf.
2. Aplikasi dibatasi hanya untuk mencari jarak terpendek antara distributor ke titik/toko sebagai *customer* dan pencarian jarak terpendek hanya 30 (tiga puluh) titik/toko.
3. Hasil yang didapat adalah hasil pencarian dan perhitungan jarak terpendek menggunakan algoritma *ant colony*.

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan Penelitian

Adapun tujuan dari pembuatan aplikasi ini adalah :

1. Untuk mensimulasikan metode heuristik melalui algoritma koloni semut pada objek distribusi barang.
2. Untuk membantu distributor mencari jarak terpendek agar perjalanan distribusi efektif dan efisien.

1.4.2 Manfaat Penelitian

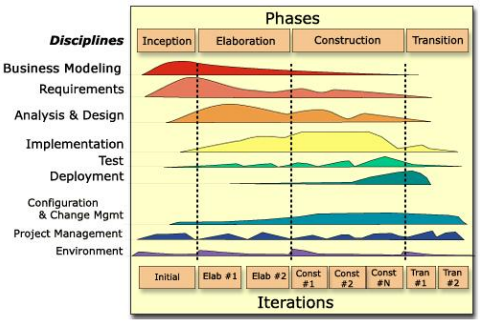
Adapun manfaat dari pembuatan aplikasi ini adalah :

1. Membantu PT. Nusantara Abadi khususnya karyawan bagian distribusi terutama yang berhubungan langsung dengan transportasi, dalam menentukan jalur terpendek saat mereka melaksanakan perjalanan mereka sehingga dapat menghemat waktu, tenaga dan biaya.
2. Menambah pembendaharaan mengenai penentuan jalur terpendek.

1.5 Metodologi Penelitian dan Teknik Pengumpulan Data

1.5.1 Metode Pengembangan Sistem

RUP adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iteratve*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). (Rosa A.S. dan M. Shalahuddin, 2011: 105)



Gambar 1.1 Tahap RUP
(Sumber : Rosa A.S. dan M. Shalahuddin, 2011: 105)

Adapun tahap-tahap dari pelaksanaan pengembangan RUP adalah sebagai berikut :

1. Permulaan (*Inception*)
   Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. Perluasan/Perencanaan (*Elaboration*)
   Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih analisis dan desain sistem serta implementasi sistem yang fokus pada *prototype.*
3. Konstruksi (*Construction*)
   Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak secara lengkap yang siap diserahkan kepada pemakai.
4. Transisi (*transition*)
   Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user.* Aktifitas pada tahap ini termasuk pada pelatihan *user,* pemeliharaan dan pengujian sistem apakah sistem sudah memenuhi harapan *user.*

1.5.2 Teknik Pengumpuan Data

1. Observasi
   Mengadakan observasi dan mengajukan pertanyaan kepada narasumber yang mengetahui tentang hal yang berhubungan dengan topik.
2. Wawancara
   Tahap ini dilakukan dengan cara mengajukan pertanyaan secara langsung oleh pewawancara kepada responden dan jawaban-jawaban responden dicatat atau direkam dengan alat perekam.
3. Studi Literatur
   Teknik studi literatur yaitu mengumpulkan data melalui studi literatur dari buku-buku referensi dan jurnal yang terkait dengan algoritma *ant colony optimization*.

2. LANDASAN TEORI

2.1 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan Aplikasi berasal dari kata *application* yang artinya penerapan; lamaran; penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju.

2.2 Teori Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satusama lain melalui sisi/busur (edges) (Zakaria, 2006). Suatu Graf G terdiri dari dua himpunan yaitu himpunan V dan himpunan E.

a. Verteks (simpul) : V = himpunan simpul yang terbatas dan tidak kosong
b. Edge (sisi/busur) : E = himpunan busur yang menghubungkan sepasang simpul.

Simpul-simpul pada graf dapat merupakan obyek sembarang seperti titik, atom-atom suatu zat, nama anak, jenis buah, komponen alat elektronik. Busur dapat menunjukkan hubungan (relasi) sembarang seperti rute penerbangan, jalan raya, sambungan telepon, ikatan kimia, dan lain-lain. Notasi graf: G(V,E) artinya graf G memiliki V simpul dan E busur.

2.3 *Traveling Salesman Problem*

Traveling Salesman Problem adalah permasalahan yang sudah cukup tua di dunia optimasi. Pada permasalahan ini, ada sebuah kota awal dan sejumlah *n* kota untuk

dikunjungi. Seorang salesman dituntut memulai perjalanan dari kota awal ke seluruh kota yang harus dikunjungi tepat satu kali.

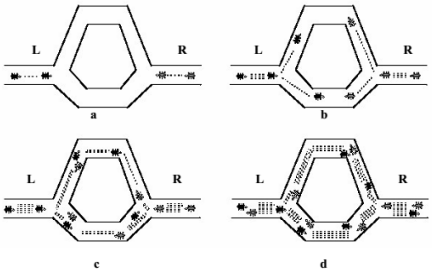
Secara ringkas, berikut adalah karakterisktik dari permasalahan TSP

1. Perjalanan berawal dan berakhir dari dan ke kota awal
2. Ada sejumlah kota yang semuanya harus dikunjungi tepat satu kali
3. Perjalanan tidak boleh kembali ke kota awal sebelum semua kota tujuan dikunjungi
4. Tujuan dari permasalahan ini adalah meminimumkan total jarak yang ditempuh salesman dengan mengatur urutan kota yang harus dikunjungi.

2.4 Algoritma Semut

Algoritma Semut diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996) Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut.

Gambar 2.6 menujukkan perjalanan semut dalam menemukan jalur terpendek dari sarang ke sumber makanan.



Gambar 2.6. Perjalanan semut menemukan makanan

Gambar 2.6.a di atas menunjukkan ada dua kelompok semut yang akan melakukan perjalanan. Satu kelompok bernama L yaitu kelompok yang berangkat dari arah kiri yang merupakan sarang semut dan kelompok lain yang bernama kelompok R yang berangkat dari kanan yang merupakan sumber makanan. Kedua kelompok semut dari titik berangkat sedang dalam posisi pengambilan keputusan jalan sebelah mana yang akan diambil. Kelompok semut L membagi dua kelompok lagi.Sebagian melalui jalan atas dan sebagian melalui jalan bawah. Hal ini juga berlaku pada kelompok semut R. Gambar 2.6.b dan gambar 2.6.c menunjukkan bahwa kelompok semut berjalan pada kecepatan yang sama dengan meninggalkan feromon atau jejak kaki di jalan yang telah dilalui. Feromon yang ditinggalkan oleh kumpulan semut yang melalui jalan atas telah mengalami banyak penguapan karena semut yang melalui jalan atas berjumlah lebih sedikit dari pada jalan yang di bawah. Hal ini dikarenakan jarak yang ditempuh lebih panjang daripada jalan bawah. Sedangkan feromon yang berada di jalan bawah, penguapannya cenderung lebih lama. Karena semut yang melalui jalan bawah lebih banyak daripada semut

yang melalui jalan atas. Gambar 2.7.d menunjukkan bahwa semut-semut yang lain pada akhirnya memutuskan untuk melewati jalan bawah karena feromon yang ditinggalkan masih banyak. Sedangkan feromon pada jalan atas sudah banyak menguap sehingga semut-semut tidak memilih jalan atas tersebut. Semakin banyak semut yang melalui jalan bawah maka semakin banyak semut yang mengikutinya. Demikian juga dengan jalan atas, semakin sedikit semut yang melalui jalan atas, maka feromon yang ditinggalkan semakin berkurang bahkan hilang. Dari sinilah kemudian terpilihlah jalur terpendek antara sarang dan sumber makanan.

3. ANALISA DAN PERANCANGAN

3.1 Analisis Permasalahan

Permasalahan yang akan dibahas dalam penyusunan skripsi ini adalah bagaimana cara membangun sebuah aplikasi yang dapat menghitung dan menunjukan jalur distribusi. Dengan dibuatnya aplikasi ini diharapkan bisa membantu masyarakat khususnya para kurir distributor untuk menggunakan aplikasi tersebut.

Dipilihnya algoritma semut *(ant colony)* dikarenakan algoritma ini menggunakan simulasi daya jelajah koloni semut yang selalu dapat menemukan jalur terpendek dengan efektif, sehingga terjadi *matching* dengan permasalahan yang penyusun temukan.

3.2 Analisis Kebutuhan Sistem

Dalam pembangunan aplikasi ini, aplikasi ditekankan untuk bisa berinteraksi dengan pengguna. Sistem harus mengetahui karakteristik pengguna sehingga aplikasi yang dibuat mudah digunakan oleh pengguna (*user friendly*) dan sesuai dengan fungsi dan tujuan aplikasi.

Dari aplikasi yang telah dibangun haruslah dapat memberikan kesenangan dan kepuasan bagi pengguna sehingga tidak menimbulkan kejenuhan lagi. Oleh karena itu perancangan aplikasi ini di fokuskan pada *interface* aplikasi agar dapat digunakan dengan mudah (*user friendly*), sehingga dapat menarik perhatian dari pengguna.

Kebutuhan proses dalam sistem penentuan jalur terpendek antara lain

1. Proses pembuatan rute/jalur pada bidang peta
2. Proses penggambaran graf/node simpul jalur
3. Proses perhitungan jarak tempuh masing-masing siklus
4. Proses penyeleksian jarak distribusi terpendek

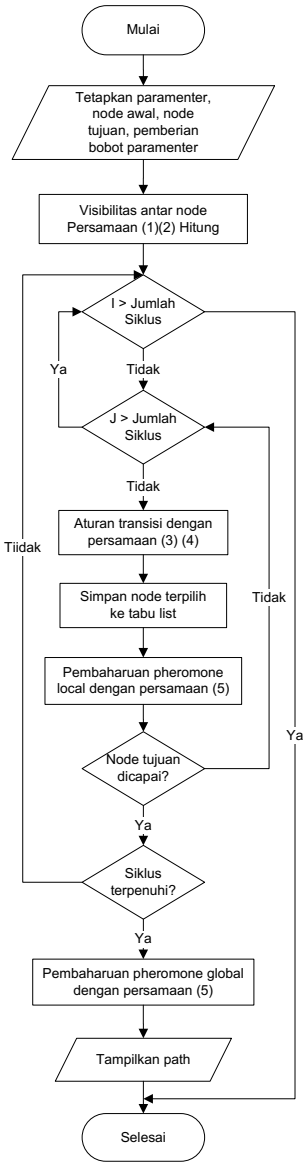
Input atau masukan dari aplikasi penentuan jalur terpendek ini, berupa parameter yang diperlukan dalam algoritma semut yaitu :

a. Data graph berupa banyak titik (n) termasuk koordinat (x,y). n merupakan banyak kecamatan yang masuk ke dalam tabu list. n ditentukan oleh pengguna. Pada penentuan koordinat dilakukan dengan cara sebagai berikut :
b. Parameter yang diperlukan dalam perhitungan algoritma semut,yaitu
   1) Intensitas jejak semut antar kecamatan (τij) dan pergerakannya ($\Delta\tau ij$). τij harus diinisialisai sebelum memulai siklus. τij digunakan dalam persamaan probabilitas kecamatan yang akan dikunjungi. $\Delta\tau ij$ diinisialisasi setelah selesai satu siklus $\Delta\tau ij$ digunakan untuk menentukan τij siklus selanjutnya.
   2) Tetapan siklus semut (Q). Q merupakan konstanta yang

digunakan dalam persamaan untuk menentukan Δτij. Nilai Q ditentukan oleh pengguna.

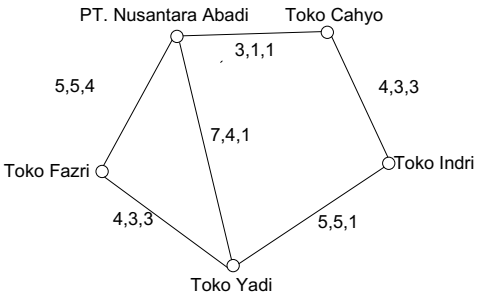
3) Tetapan pengendali intensitas jejak semut (α). α digunakan dalam persamaan probabilitas kota yang akan dikunjungi, berfungsi sebagai pengendali intensitas jejak semut. Nilai α ditentukan oleh pengguna.
4) Tetapan pengendali visibilitas (β). β digunakan dalam persamaan probabilitas kota yang akan dikunjungi yang berfungsi sebagai pengendali visibilitas. Nilai β ditentukan oleh pengguna.
5) Visibilitas antar kota (ηij). ηij digunakan dalam persamaan probabilitas kota yang akan dikunjungi. Nilai ηij merupakan hasil dari 1/dij (jarak kecamatan).
6) Banyak semut (m). m merupakan banyak semut yang akan melakukan siklus dalam algoritma semut. Nilai m ditentukan oleh pengguna.
7) Tetapan penguapan/buffer jejak semut (ρ). ρ digunakan untuk menentukan τij untuk siklus selanjutnya. Nilai ρ ditentukan oleh pengguna.
8) Jumlah siklus maksimum (NCmax). NCmax adalah jumlah maksimum siklus yang akan berlangsung. Siklus akan berhenti sesuai dengan NCmax yang telah ditentukan atau telah konvergen. Nilai NCmax ditentukan oleh pengguna.

Penjelasan input parameter dan proses pada perancagan ant colony dapat dilihat pada flowchart berikut:



Gambar 3.1 *Flowchart* Algoritma *Ant Colony*

Misalkan terdapat graf G yang berbobot dengan lima *node* dan enam *edge*, yaitu sebagai berikut:



Gambar 3.2 Contoh Graf Yang Berbobot

1. Menginisialisasi harga parameter-parameter ACO :
   - τij = 0,01
   - q0 = 0,1
   - α = 0,1
   - m = 4
   - ρ=0,5
   - Ncmax =1
   - W = { 0,5 0,2 0,2 0,45 }

   Terdapat aturan dalam menentukan nilai parameter dalam algorima semut yaitu nilai α dan β harus lebih besar atau sama dengan 0 sementara ρ harus diantara 0 dan 1. Dari graf yang diberikan diperoleh adjacency matrix jarak, lubang, tikungan, kepadatan antar node (dij).
   Contoh tabel matrik nilai dari jarak (dij).

Tabel 3.1 Matrik nilai dari dij (jarak)

	PT. NA	Toko Fazri	Toko Yadi	Toko Cahyo	Toko Indri
PT. NA	**0**	**5**	**7**	**3**	**0**
Toko Fazri	**5**	**0**	**4**	**0**	**0**
Toko Yadi	**7**	**4**	**0**	**0**	**5**
Toko Cahyo	**3**	**0**	**0**	**0**	**4**
Toko Indri	**0**	**0**	**5**	**4**	**0**

2. Menghitung visibilitas antar node sebagai media informasi kualitas suatu edges, langkah pertama dengan melakukan normalisasi tabel.

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{\max_i x_{ij}} & \text{Jika j adalah atribut keuntungan} \\ \dfrac{\min_i x_{ij}}{x_{ij}} & \text{Jika j adalah atribut biaya} \end{cases}$$

Dengan rij adalah rating kinerja ternomalisasi dari alternatif PT. Nusantara Abadi i pada atribut Toko Yadi j, i=1,2,…., m dan j=1,2,….,n.

xij merupakan rating kinerja alternatif ke i terhadap atribut ke j.
Contoh normalisasi tabel kemungkinan node a ke node lain :
PT. Nusantara Abadi ke Toko Fazri = 3/5 5/5 4/4 1/5 = [0,6 1 1 0,2]
PT. Nusantara Abadi ke Toko Yadi = 3/7 4/5 1/4 5/5 = [0,43 0,8 0,8 0,25]
PT. Nusantara Abadi ke Toko Cahyo = 3/3 1/5 1/4 1/5 = [1 0,2 0,25 0,2]

Langkah kedua mencari nilai preferensi untuk setiap atribut (Vj).
Perhitungannya adalah:

a. PT. Nusantara Abadi ke Toko Fazri = (0,15 x 0,6) + (0,2 x 1) + (0,2 x 1) + (0,45 x 0,2) = 0,09 + 0,2 + 0,2 + 0,09 = 0,58
b. PT. Nusantara Abadi ke Toko Yadi = (0,15 x 0,43) + (0,2 x 0,8) + (0,2 x 0,8) + (0,2 x 0,25) = 0,06 + 0,16 + 0,16 + 0,05 = 0,43
c. PT. Nusantara Abadi ke Toko Cahyo = (0,15 x 1) + (0,2 x 0,2) + (0,2 x 0,25) + (0,45 x 0,2) = 0,15 + 0,04 + 0,16 + 0,09 = 0,44

Hasil lengkap visibilitas antar node dapat dilihat pada Tabel 3.2

Tabel 3.2 Visibilitas (ηij)

	PT. NA	Toko Fazri	Toko Yadi	Toko Cahyo	Toko indri
PT. NA	**0**	**0,58**	**0,43**	**0,44**	**0**
Toko Fazri	**0,58**	**0**	**0,93**	**0**	**0**
Toko Yadi	**0,43**	**0,87**	**0**	**0**	**0,84**
Toko Cahyo	**0,44**	**0**	**0**	**0**	**0,58**
Toko indri	**0**	**0**	**0,836**	**0,96**	**0**

- Mencari node tujuan berikutnya Siklus I, Semut 1 :

Probabilitas dari node PT. Nusantara Abadi ke node tujuan :
Nilai bilangan yang dibang-kitkan (q), q=0,565; q ≥ q0 maka propabilitas dihitung dengan :

τ(r,u) : Jumlah pheromone pada sisi dari node r ke node u.
η(r,u) : Panjang edges dari node r ke node u
β : Parameter perbandingan jumlah pheromone relatif terhadap jarak (merupakan parame-ter yang telah ditentukan sebelumnya)
Jk(r) : Himpunan yang berisi node yang akan dikunjungi oleh ants
U : node yang berada dalam Jk(r)
q : Bilangan random
q0 : Parameter perbandingan eksploitasi terhadap eksplorasi
s : node berikutnya yang dipilih berdasarkan persamaan (4).
Eksploitasi : semut memilih node yang paling pendek dan jumlah phero-mone yang tinggi.
Eksplorasi : semut mengeksplo-rasi node yang Jika pernah dikunjungi sebelumnya.
Pk(r,s) : probabilitas semut k yang memilih berada di node r memilih node s untuk tujuan selanjutnya.
τ(r,s) : jumlah pheromone pada sisi dari node r ke node s.
η(r,s) : panjang edges dari node r ke node s.
B : parameter yang dimenen-tukan besarnya pengaruh jarak terhadap jumlah pheromone.
Jk(r) : himpunan yang berisi node yang akan dikunjungi oleh semut.
U : node yang berada dalam Jk(r)

visibility measure, η (r,s). dapat dihitung dengan
(0,01 * 0) + (0,01 * 0,58) + (0,01 *0,43) + (0,01 * 0,44) + (0,01 * 0) = 0 + 0,0058 + 0,0043 + 0,0044 + 0 = 0,0145
Hitung probabilitas dari node A menuju ke setiap node
Node PT. Nusantara Abadi = 0
Node Toko Fazri = (0,01) (0,58) / 0,0145 = 0,0058/0,0145 = 0,4
Node Toko Yadi = (0,01)(0,43) / 0,0145 = 0,0043/0,0145 = 0,296
Node Toko Cahyo = (0,01)(0,44) / 0,0145 = 0,0044 / 0,0145 = 0,303
Node Toko Indri = 0

- Probabilitas Kumulatif = 0 0,354 0,799 1 1
- Bilangan random yang dibangkit-kan r = 0,825
- Isi daftar kota = PT. Nusantara Abadi, Toko Cahyo
- Melakukan pembaharuan phero-mone local.

Dimana :
ρ = tetapan penguapan pheromone. Nilai ini bermanfaat agar tidak terjadi penumpukan pheromone secara tidak terbatas mengingat jumlah pheromone akan terus bertambah setiap kali iterasi.
τ0 = (η.Lm) dimana Lm adalah jarak antara node r ke s
η = jumlah node

Perhitungannya adalah :
τ(PT. Nusantara Abadi, Toko Cahyo) = (1-0,05).(0,01) + (0,066)
τ(PT. Nusantara Abadi, Toko Cahyo) = 0,0755
dan diperoleh hasil penyusunan path oleh ants seperti pada tabel berikut :

Tabel 3.3 Hasil pencarian dari siklus pertama

Semut Ke	*Rute*	Pnjng *Rute*	Lubang	Tikungan	Kepadatan
1	PT. NA, Tk Cahyo, Tk Indri	7	4	4	2
2	PT. NA, Tk Fazri, Tk Yadi, Tk Indri	14	13	8	11
3	PT. NA, Tk Yadi, Tk Fazri	-	-	-	-
4	PT. NA, Tk Yadi, Tk Indri	12	9	2	10

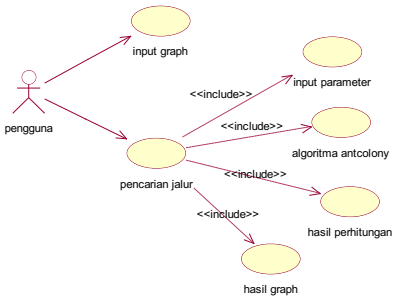
- Proses perangkingan dengan menggunakan bobot yang telah diberikan menggunakan persamaan (1) untuk normalisasi tabel dan persamaan (2) untuk menghitung nilai dari preferensi.
  Perhitungannya :
  W = [0,15 0,20 0,20 0,45]
  V1 = (0,15)(1) + (0,2)(0,308) + (0,2)(0,5) + (0.45)(0,182) = 0,3935
  V2 = (0,15)(0,5) + (0,2)(1) + (0,2)(1) + (0.45)(1) = 0,925
  V3 = (0,15)(0,583) + (0,2)(0,692) + (0,2)(0,25) + (0.45)(0,909) = 0,684

Nilai terbesar adalah V3, sehingga rute terbaik adalah *rute* yang dibangun oleh semut 2 dan rute alternatif adalah rute yang dibangun oleh semut 3 yaitu jalur PT. Nusantara Abadi, Toko Fazri, Toko Yadi, Toko Indri

3.2 Perancangan Perangkat Lunak

3.2.1 UseCase Diagram

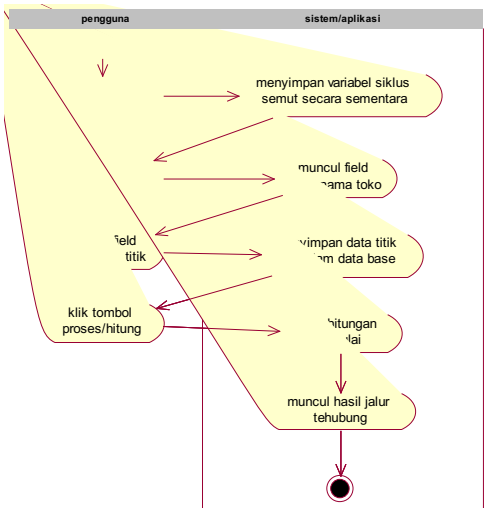
Usease diagram menggambarkan fungsionalitas yang diharapkan dari sebuah system yang ditekankan adalah "Apa" yang diperbuat system, dan bukan "Bagaimana". Sebuah Usecase merepresentasikan sebuah interaksi antara aktor dengan system, berikut adalah use case diagram dari proses yang terjadi dalam aplikasi yang dikembangkan dalam penulisan Skripsi. Di bawah ini merupakan rancangan usecase dari aplikasi distribusi ini.



Gambar 3.3 Use Case Diagram

3.2.2 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing aliran berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah Activity Diagram dari aplikasi pencarian jalur terpendek menggunakan algoritma semut

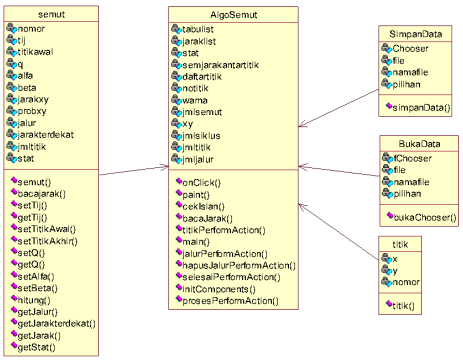


Gambar 3.4 *Activity Diagram*
Aplikasi Pencarian Jalur Distribusi

3.2.3 *Class Diagram*

Class diagram digunakan untuk melakukan visualisasi struktur kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak digunakan. *Class diagram* juga dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (*logical view*) dari suatu sistem. Selam proses desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

Berikut ini adalah akan dijelas-kan class diagram yang digunakan untuk melakukan visualisasi struktur kelas-kelas yang terdapat dalam aplikasi solusi TSP menggunakan algoritma *ant colony*.

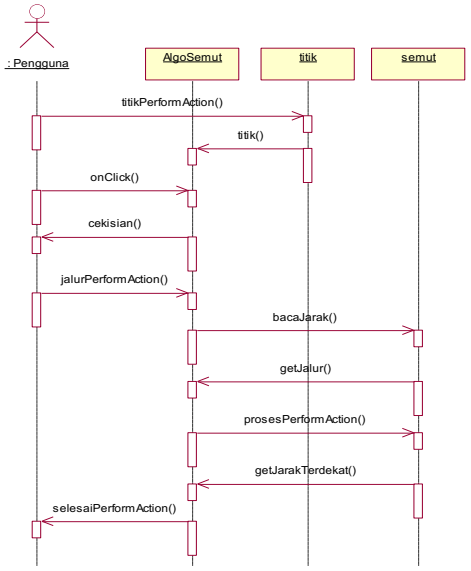


Gambar 3.5 *Class Diagram* Aplikasi Pencarian Jalur Distribusi

3.2.4 *Sequence Diagram*

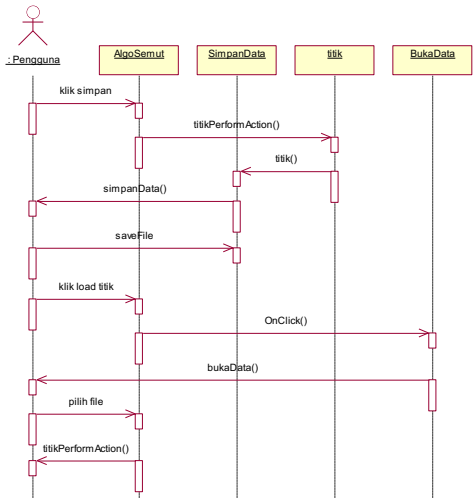
Sequence diagram digunakan untuk menggambarkan interaksi antar objek di dalam dan di sekitar system yang berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

Berikut ini adalah *sequence diagram* dari aplikasi solusi TSP menggunakan algoritma *ant colony*.



Gambar 3.6 *Sequence Diagram*
Aplikasi Proses Pencarian Jalur Distribusi

Pada *sequence diagram* diatas, seperti yang disebutkan di class diagram, class AlgoSemut adalah class utama yang menghandle komunikasi antar class, sehingga lalu lintas komunikasi dengan pengguna pun banyak dilakukan oleh class AlgoSemut kecuali pada komunikasi method titikPerformAction(), secara tidak langsung pengguna berinteraksi dengan class titik, karena class Titik menyimpan titik distribusi secara *temporary* pada aplikasi pencarian jalur distribusi.



Gambar 3.7 *Sequence Diagram* Proses *Save* dan Load Titik

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

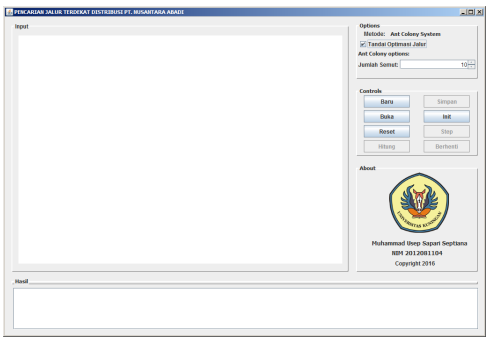
Tahap implementasi merupakan tahap penciptaan aplikasi yang yang terdiri dari penjelasan mengenai batasan implementasi, implementasi aplikasi, implementasi perangkat keras, implementasi jaringan, pengujian, dan implementasi antar muka.

4.2 Pengujian

Rancangan pengujian dilakukan dengan menggunakan teknik pengujian Black-Box yaitu berfokus pada informasi dari perangkat lunak, dengan melakukan tes case yang mempartisi domain input dari suatu program dengan cara memberikan cakupan pengujian yang mendalam dan White-box.

4.2.1 Halaman Menu Utama

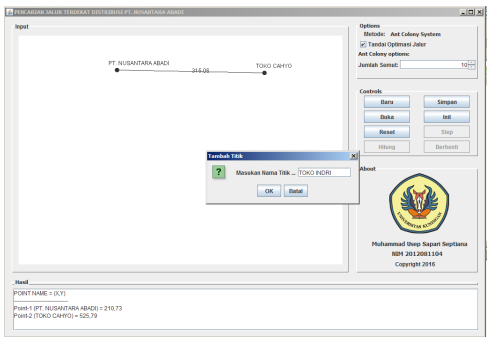
Setelah kita menjalankan aplikasi, kemudian kita masuk ke tampilan main menu seperti gambar berikut :



Gambar 4.1 Tampilan Menu Aplikasi Optimasi Jalur Terdekat

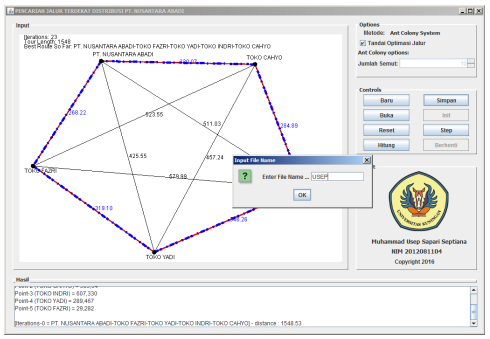
4.2.2 Halaman *Input* Titik

Secara *default,* tombol yang akan aktif setelah aplikasi dijalankan adalah tombol untuk keperluan penginputan titik jika user meng-klik *canvas* maka secara otomatis sistem akan memanggil window baru berupa proses penginputan nama titik.



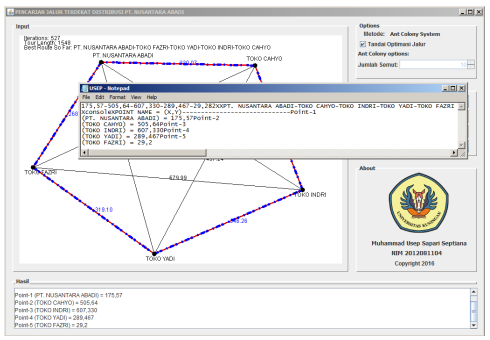
Gambar 4.2 Tampilan Penginputan Titik

4.2.3 Halaman Buka Data (*Load File*)



Gambar 4.3 Proses Load/Buka File yang sebelumnya disimpan

4.2.4 Halaman Hasil Optimasi



Gambar 4.4 Hasil Optimasi

4.3 Kasus Dan Hasil Pengujian

4.3.1 Pengujian Black Box

Digunakan untuk menguji fungsi-fungsi khusus dari perangkat

lunak yang dirancang. Sub bab ini memperlihatkan hasil pengujian yang dilakukan dalam bentuk table.

Tabel 4.3 Pengujian BlackBox

No	Fungsi yang diuji	Cara Pengu-jian	Hasil Yang diharapkan	Hasil Pengujian
1	Tombol Baru	Klik Tombol "Baru"	Mengulang Proses dengan Sepeti awal kompilasi program	Canvas menjadu bersih dan scenario tombol seperti awal pada kompilasi program
2	Tombol "Buka"	Klik Tombol "Buka"	Pop-up window untuk memilih file titik yang akan dibuka	Pop-up window muncul
4	onClick pada canvas	Dengan meng-klik pada area canvas	Muncul pop-up penginputan nama titik	Pop-up penginput-an muncul
5	Tombol Simpan	Klik tombol Simpan	Popup untuk menginputkan nama file mumcul dan file titik tersimpan	Popup input nama file muncul, dan file disimpan

4.3.2 Pengujian White Box

White Box Adalah meramalkan cara kerja perangkat lunak secara rinci, karenanya logikal path (jalur logika) perangkat lunak akan ditest dengan menyediakan test case yang akan mengerjakan kumpulan kondisi dan atau pengulangan secara spesifik Secara sekilas dapat diambil kesimpulan *white-box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Pengujian *white-box* pada proses pencarian jalur, dapat dilihat pada tabel berikut :

Tabel 4.4 Pengujian Tabel *White-Box*

Line	***Syntax***
1	for(i = 0; i < Nodes; i++){

2	visited[i] = false;
	jarak[0] = 0;
	jarakTerdekat[0] = 0;
	visited[0] = true;

3	for(i = 1; i < noNodes; i++){
	min = Integer.MAX_VALUE;
	node = -1;

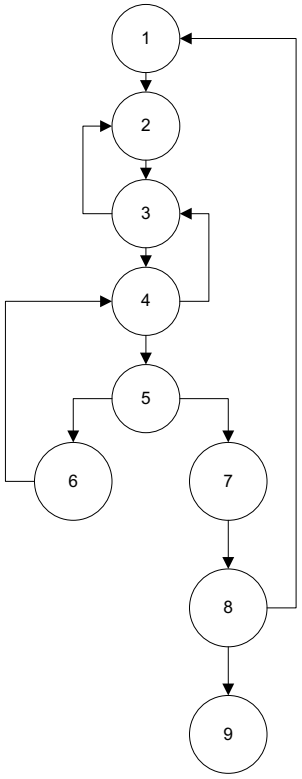
4	for(j = 0; j < noNodes; j++){
5	if((!visited[j])&&(j!=jarak[i-1])){

6	if(min > dist[jarak[i-1]][j]){
	min = dist[jarak[i-1]][j];
	node = j;
	}

7	} }

8	jarak[i] = node;
	jarakTerdekat[i] = node;
	visited[node] = true;
	}

9	jarak[noNodes] = jarak[0];
	jarakTerdekat[noNodes] = jarakTerdekat[0];
	return computeJarakLength(jarak);



Gambar 4.5 *Flow Graph* Pencarian

Jalur Terdekat

Berdasarkan *flow graph* diatas dapat dihitung nilai V(G) dengan rumus sebagai berikut :
V(G) = E – N + 2
V(G) = 12 – 9 + 2
V(G) = 5
Keterangan :
E : Jumlah edge dalam *graph*
N : Jumlah node dalam *graph*
Independent Path :
Path 1 = 1-2-3-2-3-4-3-4-5-6-4-5-7-8-1-2-3-4-5-7-8-9
Path 2 = 1-2-3-4-3-4-5-6-4-5-7-8-1-2-3-4-5-7-8-9
Path 3 = 1-2-3-2-3-4-5-6-4-5-7-8-1-2-3-4-5-7-8-9
Path 4 = 1-2-3-2-3-4-3-4-5-6-7-8-1-2-3-4-5-7-8-9
Path 5 = 1-2-3-2-3-4-3-4-5-6-7-8-9

4.4 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian Aplikasi Ant Colony dengan metode Black-Box dan WhiteBox diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional melakukan proses sesuai dengan yang diharapkan.

5. PENUTUP

5.1 Kesimpulan

Kesimpulan yang dapat diambil dengan dibangunnya perangkat lunak aplikasi pencarian jalur terdekat dengan Algoritma *Ant Colony* sebagai berikut:

1. Mapping yang dilakukan setelah penelitian ini dapat dilakukan dengan bantuan komputer dengan optimasi menggunakan *Ant Colony System*.
2. Dengan Metode Heuristik menggunakan algoritma *Ant Colony* pengguna tidak tergantung lagi dengan banyaknya titik, karena algoritma *ACO* ini dapat melayani sampai 30 titik sekaligus.

5.2 Saran

Adapun saran untuk pengembangan dan perbaikan Aplikasi pencarian jalur terpendek menggunakan Ant Colony ini adalah sebagai berikut:

1. Digantinya drawing canvas dengan peta tematik yang berupa peta jalan dan peta batas kecamatan.
2. Pengembangan dan pembangunan aplikasi ini pada perangkat mobile khususnya perangkat *Handheld* seperti Android, iPhone, Microsoft Mobile.

DAFTAR PUSTAKA

Dorigo, M. 1996. *The Ant System: Optimization by A Colony of Cooperating Agents*. IEEE Transactions on Systen Man, and Cybernetics–Part B, Vol. 26, No. 1.

Dorigo, M. dan Gambardella, L. M. 1996. *Ant Colony System: A Cooperative Learning Approach to The Traveling Salesman Problem*. Université Libre de Bruxelles Belgium.

Efendi, R. 2003. *Penerapan Algoritma Semut Untuk Pemecahan Masalah Spanning Tree pada Kasus Pemasangan Jaringan Kabel Telepon.* Tugas Akhir Jurusan Teknik Informatika, Universitas Islam Indonesia.

Hengky W, Pramana. 2006. *Aplikasi Inventory Berbasis Access*

2003. Jakarta: Elex Media Komputindo.

Kusuma Dewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.

Kusuma Dewi, S. dan Hari, P. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu.

Kristanto, Harianto. 1994. *Konsep dan Perancangan Database.* Yogyakarta: Andi Offset.

Rosa A.S dan M. Salahuddin. 2013. *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

Zakaria. 2006. *Teknologi Informasi dan Komunikasi*. Jakarta: Arya Duta.

Zuhri, Z. 2002. *Optimasi Rute dengan Algoritma Semut*. Jurnal Sistem Cerdas Volume 1 Nomor 1, Universitas Islam Indonesia.