



**RANCANG BANGUN APLIKASI PENGIRIMAN BARANG DENGAN
RUTE TERPENDEK MENGGUNAKAN METODE ANT COLONY
OPTIMIZATION PADA PT SAKA MITRA USAHA**



TUGAS AKHIR

Program Studi

S1 SISTEM INFORMASI

**UNIVERSITAS
Dinamika**

Oleh:

YOHANES KRISANTORO ADJIE

16410100002

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2021

**RANCANG BANGUN APLIKASI PENGIRIMAN BARANG DENGAN
RUTE TERPENDEK MENGGUNAKAN METODE ANT COLONY
OPTIMIZATION PADA PT SAKA MITRA USAHA**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer**



UNIVERSITAS
Dinamika

Oleh :

Nama : YOHANES KRISANTORO ADJIE
NIM : 16410100002
Program Studi : S1 Sistem Informasi

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2021

Tugas Akhir

RANCANG BANGUN APLIKASI PENGIRIMAN BARANG DENGAN ROUTE TERPENDEK MENGGUNAKAN METODE ANT COLONY OPTIMIZATION PADA PT SAKA MITRA USAHA

Dipersiapkan dan disusun oleh :
YOHANES KRISANTORO ADJIE
NIM : 16410100002

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas
Pada: Kamis, 15 Juli 2021

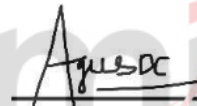
Susunan Dewan Pembahas

Pembimbing

- I. Julianto Lemantara, S.Kom., M.Eng
NIDN. 0722108601
- II. Agus Dwi Churniawan, S.Si., M.Kom
NIDN. 0723088002



Digitally signed by
Julianto Lemantara
Date: 2021.07.15
16:08:23 +07'00'



Digitally signed by
Agus Dwi
Churniawan
Date: 2021.07.16
05:54:14 +07'00'

Pembahas

Tutut Wuriyanto, M.Kom.
NIDN. 0703056702



Digitally signed by
Tutut Wuriyanto
Date: 2021.07.16
21:17:28 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



Digitally signed by
Universitas Dinamika
Date: 2021.08.05
09:20:06 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



UNIVERSITAS
Time you enjoy wasting is not wasted time
- YOHANES KRISANTORO ADJIE, 2021 -



UNIVERSITAS
Dinamika

Kupersembahkan kepada

Keluargaku yang senantiasa mendukung.

serta semua sahabat dan teman-teman yang selalu

memberi semangat.

**SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : YOHANES KRISANTORO ADJIE
NIM : 16410100002
Program Studi : S1 Sistem Informasi
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **RANCANG BANGUN APLIKASI PENGIRIMAN
BARANG DENGAN RUTE TERPENDEK
MENGUNAKAN METODE *ANT COLONY*
OPTIMIZATION PADA PT SAKA MITRA USAHA**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 12 Juli 2021

Yang menyatakan



YOHANES KRISANTORO ADJIE

NIM: 16410100002

ABSTRAK

PT Saka Mitra Usaha merupakan perantara dalam proses penjualan dan pengiriman produk makanan ringan kepada toko-toko kecil yang dilakukan oleh motoris. Motoris merupakan aktor utama dalam proses pengiriman barang. Di PT Saka Mitra Usaha terdapat 20 orang motoris yang siap melakukan pengiriman barang. Dalam melakukan proses pengiriman, motoris tidak memiliki pedoman pasti mengenai rute terdekat yang harus dilalui saat berangkat hingga kembali ke perusahaan, memungkinkan rute yang dilalui bukanlah rute terdekat, sehingga motoris membutuhkan waktu yang lebih banyak. Motoris yang menggunakan Google Maps juga harus memasukkan satu per satu destinasi toko, sehingga membutuhkan waktu yang lebih lama. Solusi yang dibuat adalah aplikasi berbasis android yang dapat memberikan rekomendasi rute terpendek. Pencarian rute terpendek ini termasuk sebuah permasalahan *Traveling Salesman Problem*, dimana seorang kurir harus melakukan pengiriman barang dengan melewati setiap rute secara optimal. Metode yang digunakan untuk menyelesaikan metode ini adalah *Ant Colony Optimization* dengan varian *Ant System*. Penggunaan *Ant System* dipilih karena banyaknya variabel yang dapat diubah sesuai kebutuhan, memungkinkan untuk mencapai hasil yang maksimal, serta jumlah destinasi yang cukup pendek yaitu 15 toko. Hasil uji coba untuk 15 toko menunjukkan bahwa aplikasi ini dapat memberikan penghitungan jarak terpendek perjalanan sebesar 40.343 kilometer, dengan nilai $\alpha = 1.5$, $\beta = 1$, $\rho = 0.9$ atau $\alpha = 2$, $\beta = 1.5$, dan $\rho = 0.5 - 0.9$. Aplikasi juga dapat memberikan rekomendasi urutan toko yang harus dilalui terlebih dahulu serta mampu mengintegrasikan urutan toko dengan navigasi akan toko yang dituju sehingga mempersingkat proses pencarian rute.

Kata Kunci : *Traveling Salesman Problem, Ant Colony Optimization, Ant System.*

KATA PENGANTAR

Puji Syukur kehadiran Tuhan Yang Maha Esa atas segala berkat dan rahmat yang telah diberikan, sehingga penulis dapat menyelesaikan penulisan dan perancangan Laporan Tugas Akhir dengan judul “Rancang Bangun Aplikasi Pengiriman Barang Dengan Rute Terpendek Menggunakan Metode *Ant Colony Optimization* Pada PT Saka Mitra Usaha”. Laporan ini disusun demi menyelesaikan Program Sarjana Komputer untuk Program Studi S1 Sistem Informasi Universitas Dinamika.

Selama menjalani proses penulisan dan perancangan, mendapatkan begitu banyak dukungan dan bantuan dari berbagai pihak, diantaranya:

1. Keluarga penulis yang senantiasa memberikan dukungan, doa, dan semangat sehingga penulisan dan perancangan Tugas Akhir dapat terselesaikan.
2. Bapak Julianto Lemantara, S.Kom., M.Eng dan Bapak Agus Dwi Churniawan, S.Si., M.Kom sebagai dosen pembimbing yang telah memberikan arahan dan saran dalam mengerjakan Tugas Akhir. Serta kepada Bapak Tutut Wuriyanto, M.Kom, sebagai dosen penguji yang telah memberikan saran dan perbaikan kepada penulis.
3. Teman-teman penulis yang terus menerus memberikan bantuan semangat dan dukungan, serta membantu mengisi kejenuhan selama pembuatan dan perancangan Tugas Akhir.

Penulis menyadari bahwa masih banyak kekurangan yang ada, baik penulisan maupun aplikasi yang dibuat, sehingga penulis berharap bahwa kritik dan saran yang diberikan dapat membuat penulis menjadi lebih baik lagi.

Surabaya, 12 Juli 2021

Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II LANDASAN TEORI	4
2.1 Penelitian Terdahulu	4
2.2 <i>Traveling Salesman Problem (TSP)</i>	5
2.3 <i>Ant Colony Optimization (ACO)</i>	5
2.4 Mapbox.....	10
2.5 <i>System Development Life Cycle (SDLC)</i>	11
BAB III METODOLOGI PENELITIAN	12
3.1 <i>Communication</i>	13
3.1.1 Project Initiation	13
3.1.2 <i>Requirement Gathering</i>	15
3.2 <i>Planning</i>	16
3.3 Modeling.....	17

3.3.1 Desain dan Analisis	17
BAB IV HASIL DAN PEMBAHASAN	25
4.1 Hasil Pencarian Rute Terpendek.....	25
4.1.1 Penjelasan Metode.....	25
4.1.2 Penghitungan Rute.....	27
4.1.3 Perbandingan Pengujian Manual.....	34
4.2 Hasil Implementasi	35
4.2.1 Pilih Toko Tujuan.....	36
4.2.2 Hitung Rute Terpendek	37
4.2.3 Navigasi.....	38
4.3 Hasil <i>Testing</i> Implementasi.....	38
4.4 Evaluasi.....	39
BAB V PENUTUP	40
5.1 Kesimpulan.....	40
5.2 Saran.....	40
DAFTAR PUSTAKA	41

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Ilustrasi pergerakan semut.....	5
Gambar 2.2 Proses ACO	8
Gambar 2.3 Alur penghitungan rute terpendek.....	9
Gambar 2.4 Pseudocode penghitungan rute terpendek	10
Gambar 2.5 Pola SDLC	11
Gambar 3.1 Alur tahapan penelitian	12
Gambar 3.2 Docflow proses bisnis pengiriman barang	14
Gambar 3.3 <i>Systemflow</i> login.....	16
Gambar 3.4 <i>Use case diagram</i>	17
Gambar 3.5 <i>Activity diagram</i> hitung rute terpendek.....	18
Gambar 3.6 <i>Activity diagram</i> navigasi perjalanan	18
Gambar 3.7 <i>Sequence diagram</i> hitung rute terpendek.....	19
Gambar 3.8 <i>Sequence diagram</i> navigasi perjalanan	20
Gambar 3.9 <i>Class diagram</i>	20
Gambar 3.10 Desain rancangan layar pilih toko	21
Gambar 3.11 Desain rancangan layar <i>dialog</i> konfirmasi.....	22
Gambar 3.12 Desain rancangan layar <i>dialog</i> pilih titik kembali	22
Gambar 3.13 Desain rancangan layar <i>loading spinner</i> proses.....	23
Gambar 3.14 Desain rancangan layar navigasi	23
Gambar 3.15 Desain rancangan layar cek keseluruhan rute	24
Gambar 4.1 Proses penghitungan jarak terpendek.....	26
Gambar 4.2 Implementasi saat terjadi kesalahan.....	36
Gambar 4.3 Implementasi pilih toko tujuan.....	37

Gambar 4.4 Implementasi halaman hitung rute terpendek	37
Gambar 4.5 Implementasi halaman navigasi konfirmasi pilih toko tujuan	38



UNIVERSITAS
Dinamika

DAFTAR TABEL

	Halaman
Tabel 2.1 Penelitian terdahulu.....	4
Tabel 3.1 Identifikasi permasalahan	14
Tabel 3.2 Analisis kebutuhan pengguna	15
Tabel 4.1 Variabel penentu	25
Tabel 4.2 Data toko	27
Tabel 4.3 Jarak antar toko	29
Tabel 4.4 Kombinasi variabel ujicoba	30
Tabel 4.5 Titik awal semut uji coba	30
Tabel 4.6 Probabilitas pilihan toko kedua untuk Semut 1	31
Tabel 4.7 Rute optimal seluruh semut pada iterasi pertama	32
Tabel 4.8 Jejak feromon baru untuk iterasi 1	33
Tabel 4.9 Detail kombinasi	34
Tabel 4.10 Pengujian manual dengan Google Maps.....	34
Tabel 4.11 Pengujian hasil aplikasi pada Google Maps	35

BAB I

PENDAHULUAN

1.1 Latar Belakang

PT Saka Mitra Usaha adalah sebuah perusahaan swasta nasional yang terletak di Jalan Jemur Andayani XV No 20 Surabaya. PT Saka Mitra Usaha bergerak dibidang jasa *Outsourcing* Manajemen Sumber Daya Manusia, menjalin kerjasama dengan beberapa perusahaan besar dalam bidang makanan ringan berupa makanan dan minuman sebagai *Event Organizer* dalam proses mengenalkan produk-produk baru. Selain itu, PT Saka Mitra Usaha juga menjadi perantara dalam proses penjualan dan pengiriman produk makanan ringan kepada toko-toko kecil yang dilakukan oleh motoris

Motoris merupakan aktor utama dalam proses pengiriman barang. Di PT Saka Mitra Usaha terdapat 20 orang motoris yang siap melakukan pengiriman barang. Sesuai namanya, motoris bergerak melakukan pengiriman dengan mengendarai motor yang dibekali dengan tas pengiriman di sebelah kiri dan kanan motornya. Ruang lingkup kerja motoris ditentukan berdasarkan kota, yaitu Kota Surabaya dan Sidoarjo. Dari ruang lingkup tersebut, tiap-tiap motoris akan ditugaskan secara spesifik berdasarkan daerahnya, seperti Surabaya Utara, Surabaya Timur, ataupun Sidoarjo Barat. Motoris ditugaskan untuk melakukan pengiriman ke kurang lebih 15 toko setiap harinya.

Proses pengiriman barang pada PT Saka Mitra Usaha diawali dari pengambilan barang terlebih dahulu ke perusahaan. Setelah itu, motoris akan mulai bergerak membawa barang yang telah diambil ke toko-toko mitra. Motoris akan melakukan pengiriman ke semua toko mitra terlebih dahulu sebelum akhirnya kembali lagi ke perusahaan. Perjalanan ke toko mitra dilakukan oleh motoris berdasarkan kebiasaan motoris dalam melakukan pengiriman.

Terdapat permasalahan yang muncul dalam proses pengiriman barang, yaitu pada efisiensi perjalanan menuju toko-toko mitra, karena sebagian motoris memulai perjalanan berdasarkan kebiasaan. Motoris tidak memiliki pedoman pasti mengenai rute terdekat yang harus dilalui saat berangkat hingga kembali ke perusahaan, memungkinkan rute yang dilalui bukanlah rute terdekat, sehingga perjalanan

menjadi tidak efisien dan motoris membutuhkan waktu yang lebih banyak. Terdapat beberapa motoris yang menggunakan Google Maps sebagai alat panduan dalam navigasi perjalanan. Dalam menggunakan Google Maps, motoris tetap harus memasukkan lokasi toko satu per satu sebagai destinasi tujuan, sehingga membutuhkan waktu yang lama dan prosesnya juga tidak praktis. Penggunaan Google Maps juga memungkinkan motoris untuk melewati toko yang sama berulang kali.

Pencarian rute terdekat dilakukan menggunakan metode *Ant Colony Optimization* (ACO), sebuah metode metaheuristik yang menirukan sikap koloni semut dalam menentukan rute terdekat antara makanan dan sarangnya (Dorigo & Stutzle, *Ant Colony Optimization*, 2004). Dalam perjalanan dari sarang menuju sumber makanan hingga kembali lagi ke sarang mereka, semut akan meninggalkan feromon, dan semut-semut lain akan melalui rute dengan feromon terbanyak, dimana rute dengan feromon terbanyak ini juga menjadi rute terpendek. Metode ACO dipilih dan digunakan pada rancang bangun aplikasi ini untuk menyelesaikan permasalahan *traveling salesman problem*, terutama untuk menemukan rute terpendek.

Berdasarkan permasalahan yang telah diuraikan diatas, muncul sebuah kebutuhan sebuah sistem atau aplikasi yang dapat mengatasi masalah tersebut, sehingga dibuatlah Rancang Bangun Aplikasi Pengiriman Barang Dengan Rute Terpendek Menggunakan Metode Ant Colony Optimization Pada PT Saka Mitra Usaha.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang permasalahan diatas, maka rumusan masalah dalam penelitian ini yaitu bagaimana merancang bangun aplikasi pengiriman barang dengan rute terpendek menggunakan metode *Ant Colony Optimization* pada PT Saka Mitra Usaha?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah di atas, maka dapat disimpulkan ruang lingkup dan batasan masalah yaitu:

1. Ruang lingkup lokasi pengiriman adalah Kota Surabaya dan Sidoarjo.
2. Motoris hanya dapat melakukan pengiriman ke Kota Surabaya atau Kota Sidoarjo saja, tidak bisa keduanya.
3. Penghitungan jarak terdekat menggunakan radius antar titik toko, sehingga jarak toko A ke toko B sama dengan toko B ke toko A.
4. Pembuatan rute dilakukan berdasarkan penghitungan radius jarak terdekat.
5. Rute jalan yang dilewati dapat berupa dua arah, sehingga rute ditentukan dari posisi awal motoris hingga melewati semua toko.
6. Toko yang telah dilewati tidak akan dilalui kembali, kecuali jarak terdekat yang ditempuh hanya dapat dicapai dengan melalui toko yang sama.
7. Jumlah toko yang dituju oleh motoris minimum 15 toko.
8. Pemetaan rute jalan menggunakan Mapbox API.
9. Aplikasi berjalan pada sistem operasi Android dengan versi minimal *Jelly Bean* (4.2.2).

1.4 Tujuan

Berdasarkan rumusan masalah diatas, dapat disimpulkan bahwa tujuan dari penelitian ini adalah untuk menghasilkan sebuah aplikasi pengiriman barang dengan rute terpendek menggunakan metode *Ant Colony Optimization* pada PT Saka Mitra Usaha sehingga motoris mendapat rute terpendek dalam proses pengiriman.

1.5 Manfaat

Berdasarkan rumusan masalah diatas, maka diperoleh beberapa manfaat yaitu:

1. Memperpendek jarak tempuh yang harus dilalui oleh motoris.
2. Mempersingkat waktu pengiriman barang.
3. Mempermudah proses integrasi toko destinasi dengan pencarian rute terdekat agar semakin praktis.

BAB II

LANDASAN TEORI

3.1 Penelitian Terdahulu

Penelitian terdahulu berupa riset dan jurnal akan digunakan untuk mendukung penulisan dan pembuatan Rancang Bangun Aplikasi Pengiriman Barang Dengan Rute Terpendek Menggunakan Metode Ant Colony Optimization Pada PT Saka Mitra Usaha.

Tabel 2.1 Penelitian terdahulu

Penelitian	Hasil Penelitian
Rancang Bangun Aplikasi Mobile Untuk Menentukan Solusi Optimal Pencarian Rute Terbaik Menggunakan Algoritma <i>Ant Colony Optimization</i> (Irawan, Setianingsih, & Arramsyah, 2018)	<ul style="list-style-type: none">• Menggunakan ACO dan <i>Simple Additive Weighing</i> (SAW) dalam menentukan rute terdekat perjalanan mobil pemadam kebakaran• Memberikan penilaian dalam pembobotan suatu kriteria tertentu, seperti kemacetan dan ketersediaan <i>hydrant</i>• Memanfaatkan <i>platform</i> android
Implementing <i>Ant Colony Optimization</i> (ACO) In Traveling Salesman Problem (Case Study At Distribution Store PB) (Nurchayanty Tanjung & Sopiah, 2014)	<ul style="list-style-type: none">• Menggunakan ACO untuk penentuan rute distribusi terdekat• Menggunakan aplikasi Matlab dalam penyusunan algoritma ACO• Menentukan nilai minimum optimal terkait biaya bahan bakar yang digunakan selama satu minggu perjalanan pada kendaraan Colt Diesel Engkel (CDE)

Dari kedua penelitian tersebut, ditemukan beberapa perbedaan dengan pembuatan aplikasi yang akan dilakukan, yaitu:

1. Tidak melakukan pembobotan prioritas (kemacetan, produk barang, dsb), karena semua lokasi pengiriman memiliki nilai prioritas yang sama.
2. Memberikan rute optimal tanpa melakukan penghitungan pengeluaran bahan bakar, disebabkan karena tidak adanya standar kendaraan motoris yang harus digunakan pada perusahaan.
3. Memberikan rekomendasi rute terdekat beserta proses *routing* perjalanan dengan pemanfaatan GPS dan Mapbox API. Dengan catatan bahwa rute terdekat diambil dari penghitungan jarak radius terbentuk garis lurus.

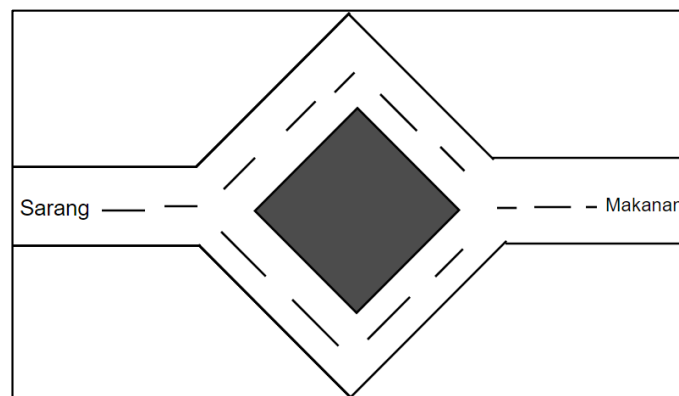
3.2 *Traveling Salesman Problem (TSP)*

Traveling salesman problem (TSP) merupakan permasalahan yang muncul dimana seorang salesman harus melewati setiap kota yang diberikan dimana jarak yang ditempuh merupakan jarak yang paling minimum, dan rute hanya boleh dilewati sekali saja (Chudasama, Shah, & Panchal, 2011). TSP masuk kedalam kategori “NP-Hard”, yaitu sebuah permasalahan yang dapat diselesaikan dalam waktu polinomial (Gupta & Poonam, 2013). TSP yang digunakan dalam rancang bangun aplikasi ini merupakan TSP simetris, dimana jarak dari kota i ke kota j sama dengan jarak dari kota j ke kota i .

3.3 *Ant Colony Optimization (ACO)*

Ant Colony Optimization (ACO) merupakan sebuah algoritma yang meniru cara bekerja koloni semut, yaitu dalam menemukan rute antara sarang dengan makanannya (Dorigo & Stutzle, 2004). Dalam pembuatan judul ini, variasi dari ACO yang digunakan oleh penulis merupakan variasi pertama yang ada dari sekian banyak variasi ACO yang lain, yaitu *Ant System*. Varian *Ant System* bukanlah sebuah varian yang kompetitif dalam penghitungan *Traveling Salesman Problem*, tetapi tetap digunakan oleh penulis karena:

1. Memiliki banyak kombinasi variabel yang mudah untuk diubah, sehingga dapat ditentukan sesuai kebutuhan sehingga dapat memenuhi jarak minimal.
2. Jumlah toko destinasi yang tidak terlalu banyak. *Ant System* dapat menunjukkan hasil yang optimal untuk destinasi hingga 75 kota. (Dorigo & Stutzle, 2010).



Gambar 2.1 Ilustrasi pergerakan semut

Dalam perjalanan hingga kembali ke sarangnya, para semut akan mengunjungi tiap titik (*node*) dan meninggalkan feromon yang nantinya akan

diikuti oleh semut-semut lain. Semut cenderung mengikuti jalan dengan jumlah feromon terbanyak, dimana jarak dengan jumlah feromon terbanyak tersebut adalah jarak terpendek. ACO digunakan pada berbagai penyelesaian optimasi, seperti *job-shop scheduling problem* (JSP), *vehicle routing problem* (VRP), hingga *traveling salesman problem* (TSP) (Raghavendra, 2015). Secara matematis, rumus ACO dapat dituliskan sebagai berikut (Katiyar, Ibraheem, & Ansari, 2015):

$$P_{ij} = \frac{[\tau_{ij}]^a [\eta_{ij}]^\beta}{\sum \tau_{ij} [\tau_{ij}]^a [\eta_{ij}]^\beta} \quad (2.1)$$

Dimana P_{ij} merupakan probabilitas terpilihnya jalur i menuju j , τ_{ij} merupakan jejak feromon pada jalur i menuju j , dan η merupakan invers ($1 / d_{ij}$) dari jarak yang dilalui antar node. a dan β merupakan variabel yang digunakan untuk mempengaruhi beban relatif dari feromon dan visibilitas. Pergerakan semut akan mempengaruhi perjalanan terhadap feromon, yaitu mengubah jejak feromon yang ditinggalkan selama perjalanan melalui evaporasi feromon, yang akan dituliskan sebagai berikut (Katiyar, Ibraheem, & Ansari, 2015):

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad \text{dimana} \quad (2.2)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{apabila } (i, j) \text{ dilalui semut} \\ 0 & \text{jika tidak} \end{cases}$$

dimana nilai ρ merupakan konstanta evaporasi, $\Delta\tau_{ij}^k$ adalah nilai perubahan jejak feromon pada titik i, j oleh semut k , Q adalah banyak feromon yang ditinggalkan oleh semut, dan L_k merupakan jarak tempuh yang dilalui oleh Semut k . Evaporasi feromon digunakan untuk menandai nilai rute yang efektif dan efisien. Semakin sering rute dilalui oleh semut, maka nilai feromon setelah evaporasi juga akan bertambah. Apabila rute jarang dilewati, maka jumlah feromon akan semakin berkurang setiap iterasinya.

Dalam proses pencarian rute terdekat ini, diawali dengan penentuan lokasi *node* (lokasi toko) berdasarkan titik koordinat lokasinya berupa garis lintang dan garis bujur. Setelah itu akan dilakukan penghitungan jarak antar *node* (*edge*) akan dilakukan menggunakan Haversine Formula. Haversine Formula merupakan sebuah metode yang digunakan untuk mengukur jarak kedua titik pada permukaan

bumi berdasarkan garis bujur dan garis lintang kedua titik (Hartanto, Furqan, Siahaan, & Fitriani, 2017). Haversine Formula dapat digambarkan sebagai berikut:

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (2.3)$$

$$c = 2 \cdot \text{atan} 2(\sqrt{a}, \sqrt{1-a}) \quad (2.4)$$

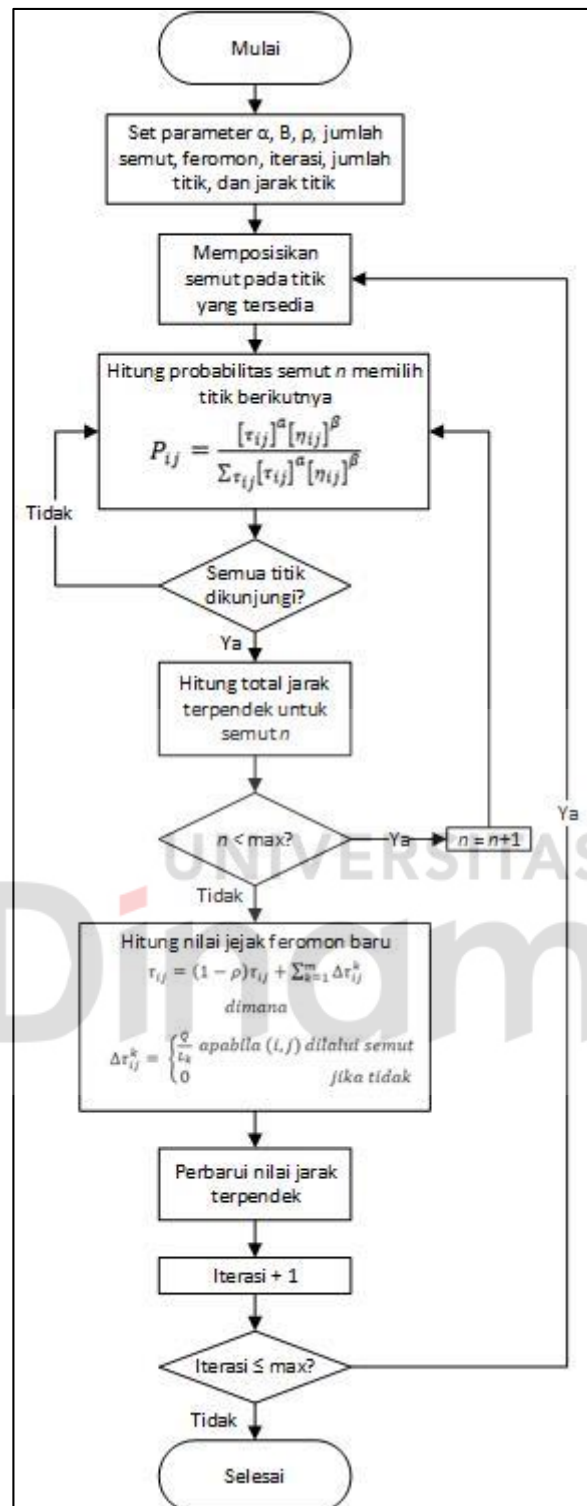
$$d = R \cdot c \quad (2.5)$$

dimana φ adalah garis lintang, λ adalah garis bujur, R adalah radius bumi, yaitu 6.371 km.

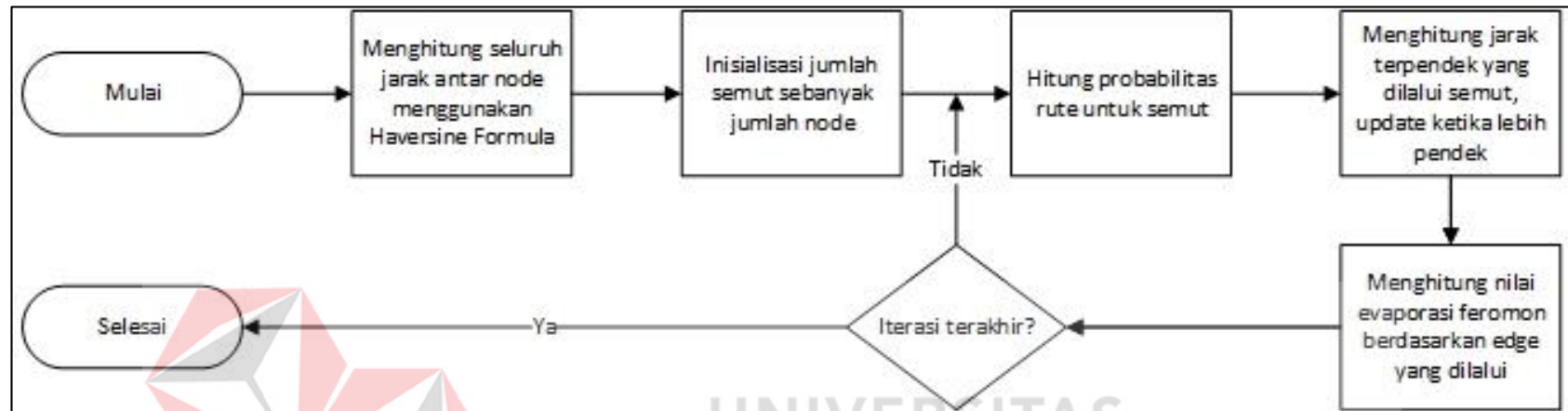
Secara sederhana, bentuk proses ACO dapat dilihat pada Gambar 2.2, sedangkan keseluruhan penghitungan rute terdekat dapat dilihat berdasarkan alur pada Gambar 2.3 dan pseudocode detail penghitungan rute terdekat pada Gambar 2.4.




UNIVERSITAS
Dinamika



Gambar 2.2 Proses ACO



Gambar 2.3 Alur penghitungan rute terpendek



```

inisialisasi alpha sebesar 1
inisialisasi beta sebesar 2
inisialisasi rho sebesar 0.1
inisialisasi iterasi sebesar 100
inisialisasi radius_terpendek sebesar 0
inisialisasi kota_terpendek
inisialisasi jarak_terpendek
inisialisasi array kota_dilewati
inisialisasi a sebesar 1
ambil data array kota dari database
inisialisasi posisi_awal dari posisi saat ini
inisialisasi jumlah_kota dari panjang array kota
jika ( a kurang dari sama dengan iterasi )
    untuk ( a kurang dari jumlah_kota )
        hitung jarak_terpendek dari dua kota dengan ACO
        jika jarak_terpendek kurang dari radius_terpendek
            radius_terpendek diupdate
            tambahkan kota kedalam kota_dilewati
            hitung evaporasi feromon
            a ditambah 1
        jika tidak
            hitung evaporasi feromon
            a ditambah 1
    jika ( a sama dengan iterasi )
        untuk ( a kurang dari jumlah_kota )
            hitung jarak_terpendek dari dua kota dengan ACO
            jika jarak_terpendek kurang dari radius_terpendek
                radius_terpendek diupdate
                tambahkan kota kedalam kota_dilewati
                a ditambah 1
            jika tidak
                a ditambah 1
        cetak radius_terpendek beserta urutan kota_dilewati

```

Gambar 2.4 Pseudocode penghitungan rute terpendek

3.4 Mapbox

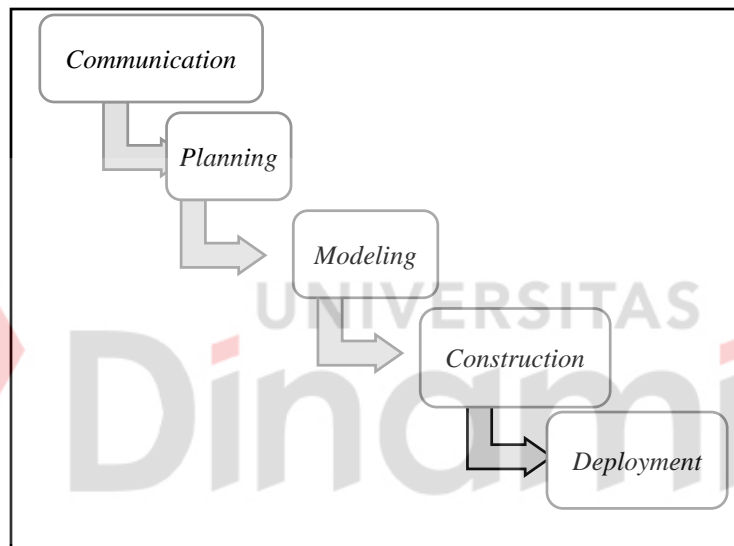
Dikutip dari profil perusahaan di websitenya, Mapbox merupakan sebuah platform data lokasi untuk aplikasi *mobile* dan website. Ditemukan pada tahun 2010, Mapbox telah digunakan oleh banyak perusahaan terkenal seperti penyedia layanan petunjuk perjalanan Lonely Planet dan situs ramalan cuaca Weather.com. Mapbox juga dapat digunakan dalam berbagai macam kasus lain, seperti pencarian toko, pelacakan aset hingga logistik.

Pada pembuatan aplikasi pencarian rute terdekat ini, Mapbox akan berperan dalam melakukan penunjuk jalan. Mapbox akan bekerja dalam memberikan

visualisasi rute akan destinasi yang harus dilalui motoris. Selain itu, Mapbox juga akan memberikan arahan navigasi selama perjalanan pengiriman barang oleh motoris.

3.5 *System Development Life Cycle (SDLC)*

SDLC model *waterfall* yang juga disebut sebagai *classic life cycle* menyarankan sebuah pendekatan yang sistematis dalam pengembangan aplikasi yang diawali dari penentuan kebutuhan pengguna dan bergerak menuju *planning*, *modeling*, *construction*, dan *deployment* (Pressman, 2010). Fase-fase dari SDLC adalah sebagai berikut:

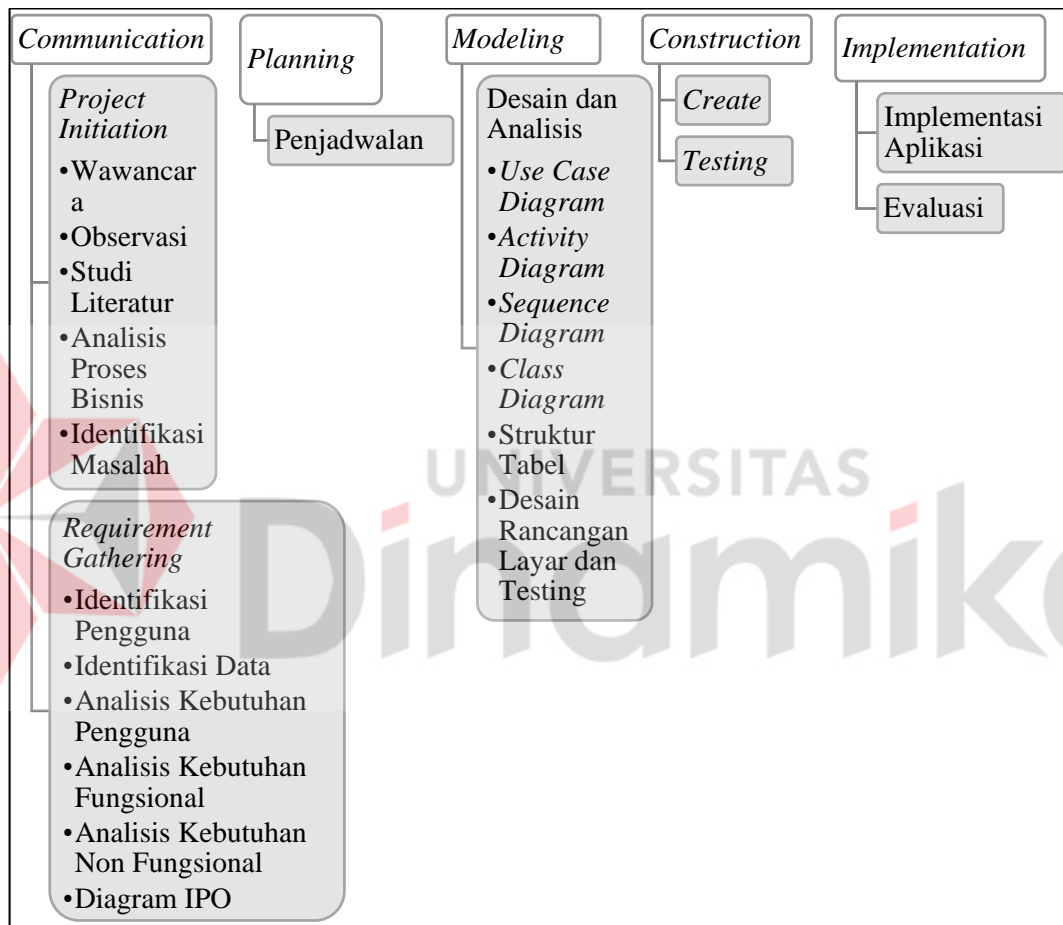


Gambar 2.5 Pola SDLC
(Sumber: Pressman 2010)

BAB III

METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah menggunakan metode SDLC *waterfall*. Metode *waterfall* memungkinkan pembuatan aplikasi terjadi secara berurutan dan sistematis. Tahapan dalam pembuatan aplikasi dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur tahapan penelitian
(Sumber: Pressman 2010)

3.1 *Communication*

3.1.1 **Project Initiation**

Proses *project initiation* dilakukan melalui tahap-tahap sebagai berikut.

1. **Wawancara**

Proses wawancara dilakukan secara langsung dengan Bapak Hario sebagai pemilik perusahaan dan Ibu Mitha dari bagian administrasi. Wawancara juga dilakukan dengan Mas Abdisa, salah seorang motoris yang ada pada PT Saka Mitra Usaha. Wawancara dilakukan untuk mengetahui alur proses bisnis pengiriman barang, kendala-kendala yang terjadi pada proses bisnis, serta harapan yang dimunculkan dari aplikasi yang akan dibuat. Detail transkrip wawancara dapat dilihat pada Lampiran 1.

2. **Observasi**

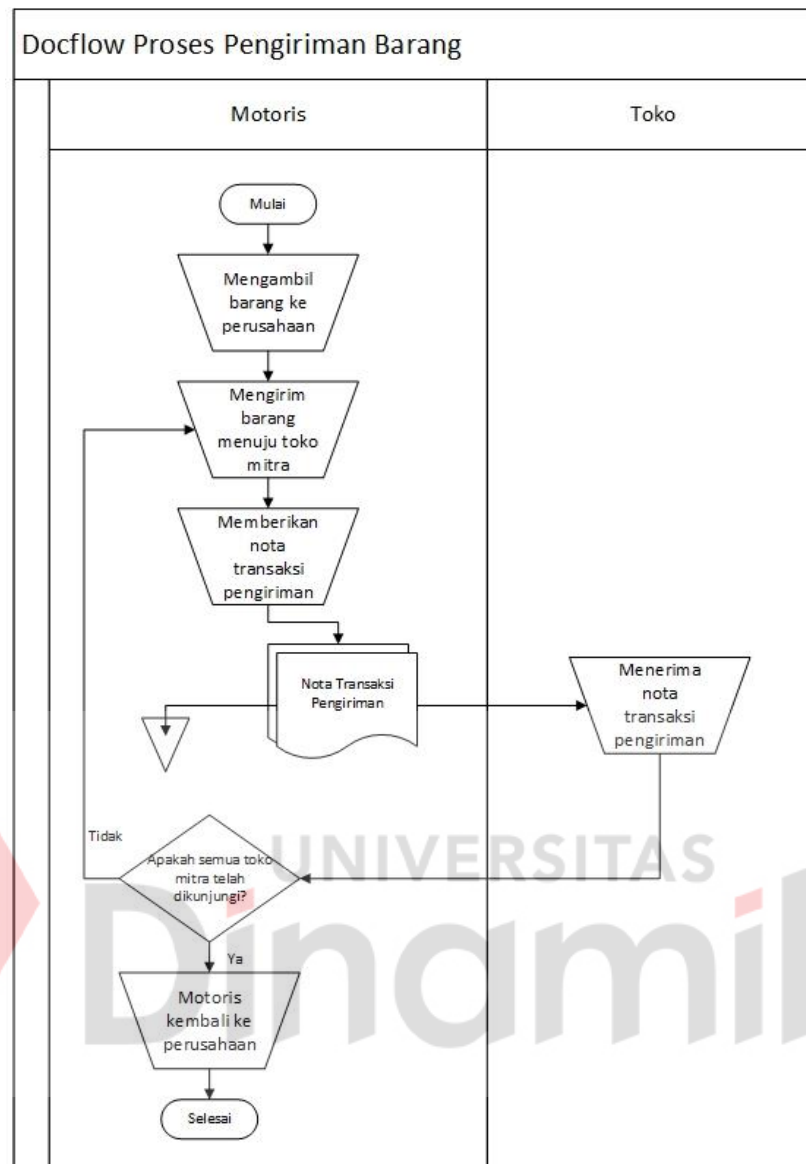
Setelah proses wawancara selesai, langkah berikutnya adalah melakukan observasi untuk melihat secara langsung proses motoris dalam melakukan proses pengiriman barang. Observasi juga dilakukan untuk melihat faktor-faktor tertentu yang mempengaruhi motoris dalam melakukan pengiriman barang serta faktor yang mempengaruhi pembuatan aplikasi pencarian rute terpendek.

3. **Studi Literatur**

Studi literatur dilakukan oleh penulis untuk mencari metode pencarian rute terpendek yang sekiranya cocok dengan kondisi proses bisnis saat ini. Penulis menemukan bahwa metode *Ant Colony Optimization* merupakan salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan *Traveling Salesman Problem*, yaitu permasalahan seorang kurir dalam memilih rute tercepat untuk melakukan pengiriman barang.

4. **Analisis Proses Bisnis**

Setelah proses observasi dan wawancara selesai dilaksanakan, dilakukanlah analisis proses bisnis yang berjalan saat ini. Analisis proses bisnis dilakukan untuk melihat secara keseluruhan alur proses pengiriman barang yang dilakukan motoris dari awal hingga selesai. Bentuk proses bisnis saat ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Docflow proses bisnis pengiriman barang

5. Identifikasi Permasalahan

Dari hasil analisis proses bisnis saat ini, ditemukan permasalahan yang muncul pada dari proses bisnis pengiriman barang seperti yang terlihat pada Tabel 3.1.

Tabel 3.1 Identifikasi permasalahan

No	Permasalahan	Dampak	Solusi
1	Tidak adanya pedoman resmi rute terdekat dari toko ke toko.	Motoris dapat menghabiskan waktu dan yang lebih banyak.	Memberikan rekomendasi rute terdekat keberangkatan hingga kembali ke perusahaan menggunakan <i>Ant Colony Optimization</i> .

3.1.2 Requirement Gathering

Proses *requirement gathering* yang terjadi pada *communication* adalah sebagai berikut:

1. Identifikasi Pengguna

Identifikasi pengguna dilakukan untuk mengetahui siapa saja yang akan menggunakan aplikasi yang dikembangkan. Dari hasil identifikasi pengguna didapatkan bahwa pengguna motoris sebagai pelaku utama proses pengiriman dan penjualan barang serta bagian administrasi dalam melakukan proses pengelolaan data master dan penugasan.

2. Identifikasi Data

Identifikasi data digunakan untuk mengetahui jenis dan banyak data yang digunakan pada aplikasi. Data yang didapatkan adalah sebagai berikut:

a. Data Motoris

Data motoris meliputi seluruh identitas terkait dengan motoris, mulai dari id motoris, nama motoris, jenis kelamin, daerah tugas, dan password.

b. Data Toko

Data toko meliputi identitas toko, mulai dari id toko, nama toko, jenis toko, alamat, daerah, status tugas, dan titik koordinatnya.

3. Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna digunakan untuk mengetahui fungsi atas kebutuhan data yang akan digunakan oleh pengguna terhadap aplikasi.

Tabel 3.2 Analisis kebutuhan pengguna

Kebutuhan Fungsi		Kebutuhan Data	Kebutuhan Informasi
Mengelola master	data	<ul style="list-style-type: none"> • Data Motoris • Data Toko 	<ul style="list-style-type: none"> • Informasi motoris • Informasi toko
Memberikan penugasan dengan motoris	toko	<ul style="list-style-type: none"> • Data Motoris • Data Toko 	<ul style="list-style-type: none"> • Informasi ID motoris • Informasi identitas toko
Melakukan pencarian rute terdekat		<ul style="list-style-type: none"> • Data Motoris • Data Toko 	<ul style="list-style-type: none"> • Informasi ID motoris • Informasi identitas toko • Informasi titik koordinat toko

4. Analisis Kebutuhan Fungsional

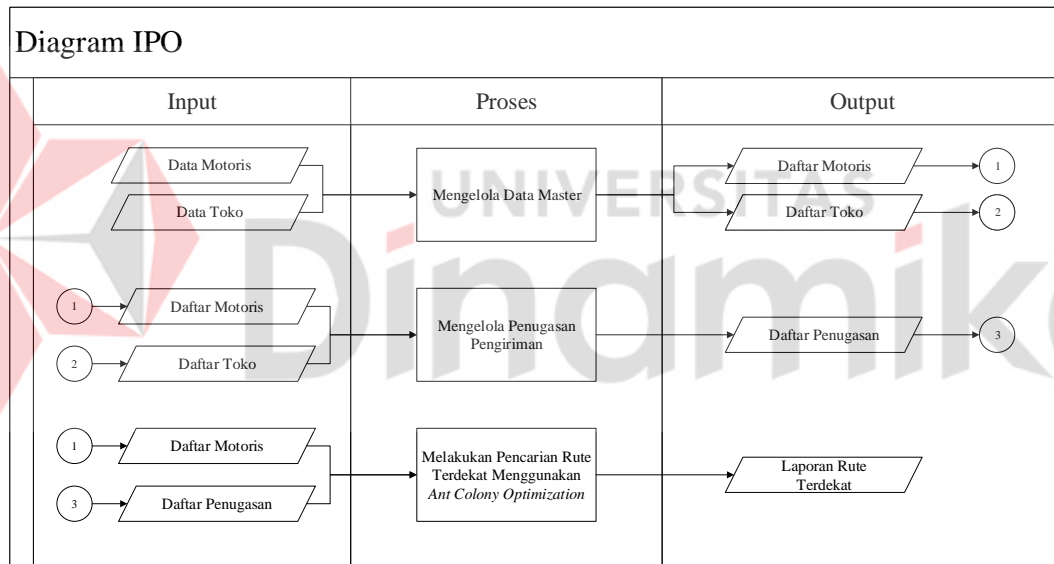
Analisis kebutuhan fungsional dibawah ini meliputi mengelola data master, memberikan penugasan toko dengan motoris, dan melakukan pencarian rute terdekat. Bentuk analisis kebutuhan fungsional dapat dilihat pada Lampiran 2.

5. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional berhubungan secara langsung dengan kinerja sistem secara umum, terutama terkait dengan standar dari berjalannya sistem. Keterangan lebih jelasnya dapat dilihat pada Lampiran 3.

6. Diagram IPO

Diagram IPO berisi tentang keseluruhan informasi terkait data, mulai dari masuk, proses, hingga hasil akhirnya. Diagram IPO dapat dilihat lebih jelas pada Gambar 3.3.



Gambar 3.3 *Systemflow* login

3.2 Planning

Proses *planning* atau perencanaan dilakukan untuk mengatur perencanaan pengerjaan aplikasi dimulai dari awal hingga akhir agar berjalan sesuai dengan jadwal yang telah ditentukan. Secara detail, bentuk jadwal pengerjaan aplikasi dapat dilihat pada Lampiran 4.

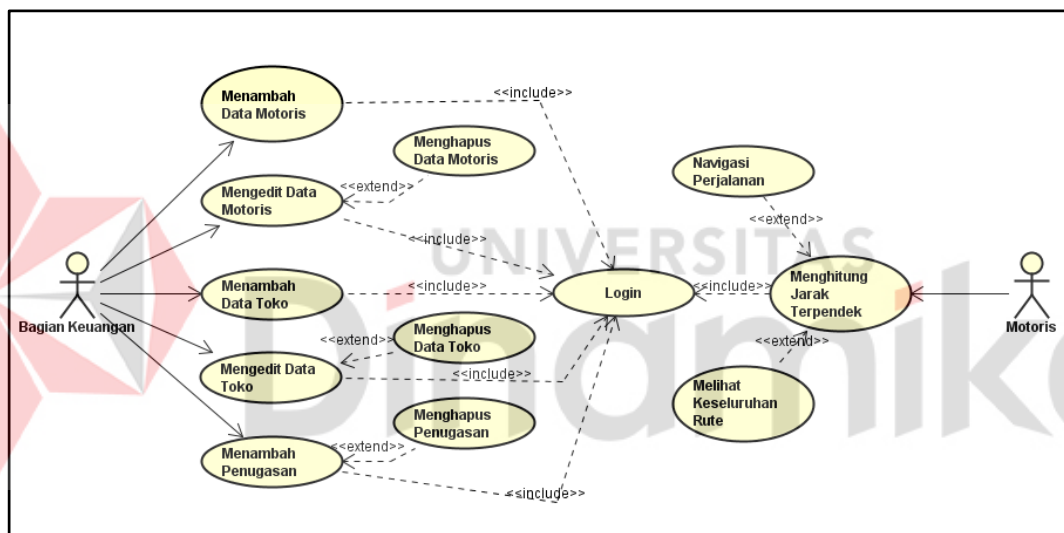
3.3 Modeling

3.3.1 Desain dan Analisis

Proses desain dan analisis digunakan untuk merancang bentuk awal aplikasi yang akan dibuat. Bentuk awal yang dimaksud meliputi penggambaran alur sistem dan data dalam bentuk *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, struktur tabel, hingga penggambaran awal bentuk desain *interface* aplikasi.

1. Use Case Diagram

Use case diagram digunakan untuk menjelaskan interaksi atau hubungan antara aktor dengan sistem yang akan dibuat. Untuk lebih detailnya, bentuk *use case diagram* dapat dilihat pada Gambar 3.4



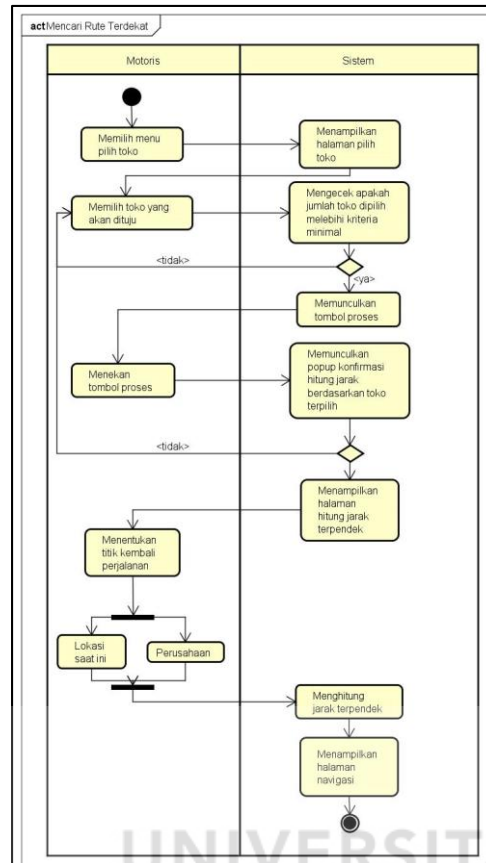
Gambar 3.4 Use case diagram

2. Activity Diagram

Activity diagram digunakan untuk menggambarkan *workflow* yang ada pada organisasi, menunjukkan keterkaitan sistem dengan manusia, serta memberikan kejelasan terhadap *use case diagram*. *Activity diagram* untuk pencarian rute terdekat dan navigasi dapat dilihat pada Gambar 3.5 dan Gambar 3.6, sedangkan keseluruhan *activity diagram* keseluruhan dapat dilihat pada Lampiran 5.

a. Activity Diagram Pencarian Rute Terpendek

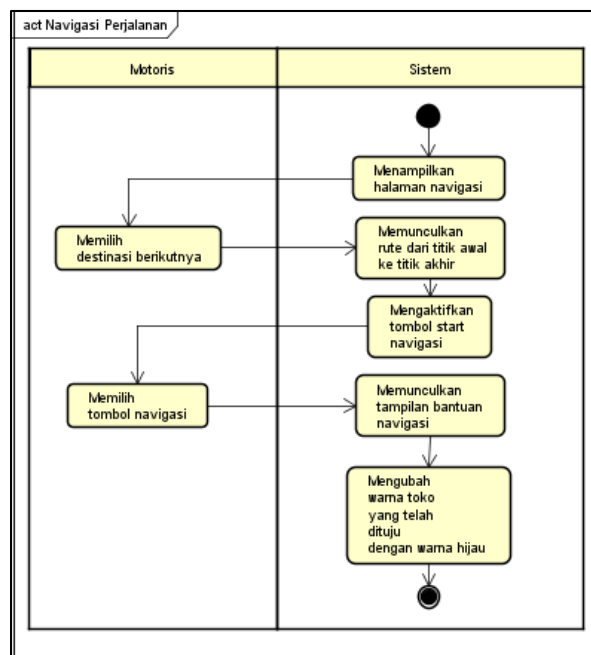
Activity diagram pencarian rute terdekat menjelaskan alur yang terjadi antara input motoris dengan respon sistem akan proses pencarian rute terdekat.



Gambar 3.5 Activity diagram hitung rute terpendek

b. Activity Diagram Navigasi Perjalanan

Activity diagram navigasi perjalanan menunjukkan alur aktivitas motoris beserta respon sistem dalam memulai navigasi perjalanan.



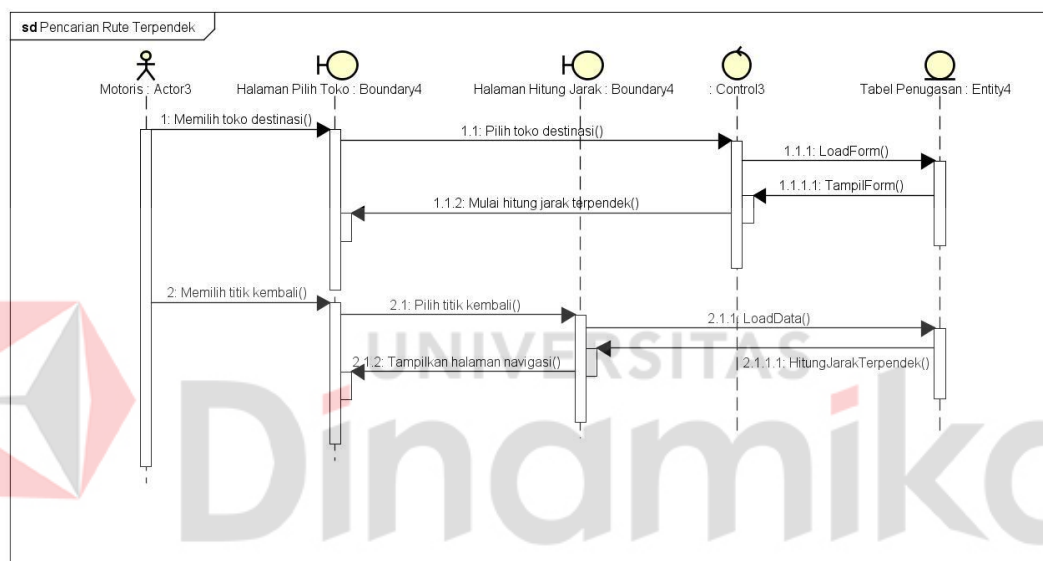
Gambar 3.6 Activity diagram navigasi perjalanan

3. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan objek dan pesan yang terdapat didalam *use case diagram*. *Sequence diagram* terkait pencarian rute terpendek dapat dilihat pada Gambar 3.7 dan Gambar 3.8, sedangkan keseluruhan *sequence diagram* dapat dilihat pada Lampiran 6.

a. Sequence Diagram Pencarian Rute Terpendek

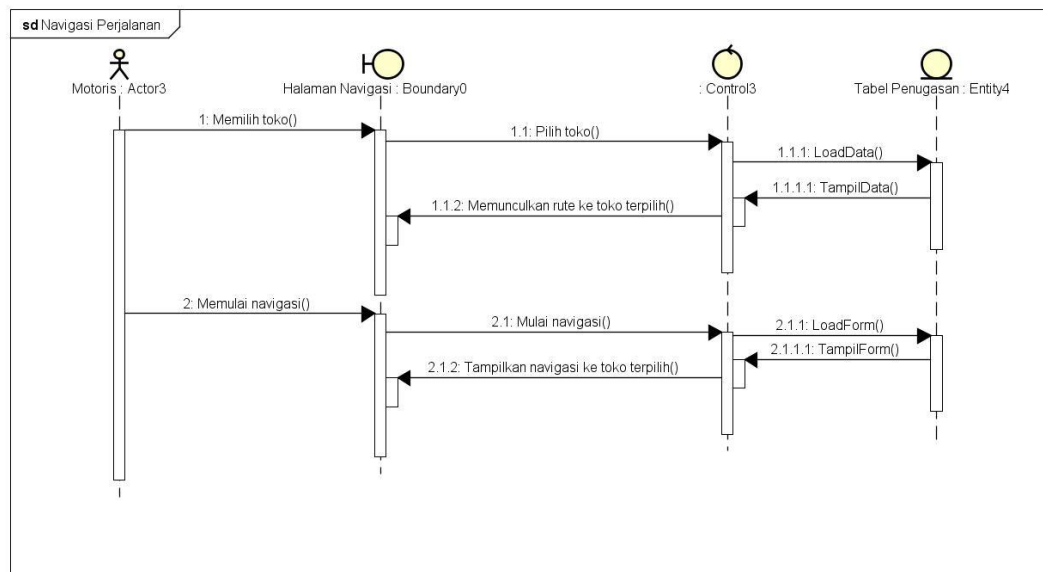
Sequence diagram pencarian rute terpendek menunjukkan proses yang dialami sistem saat motoris memilih toko dan melakukan pencarian rute terpendek.



Gambar 3.7 *Sequence diagram* hitung rute terpendek

b. Sequence Diagram Navigasi Perjalanan

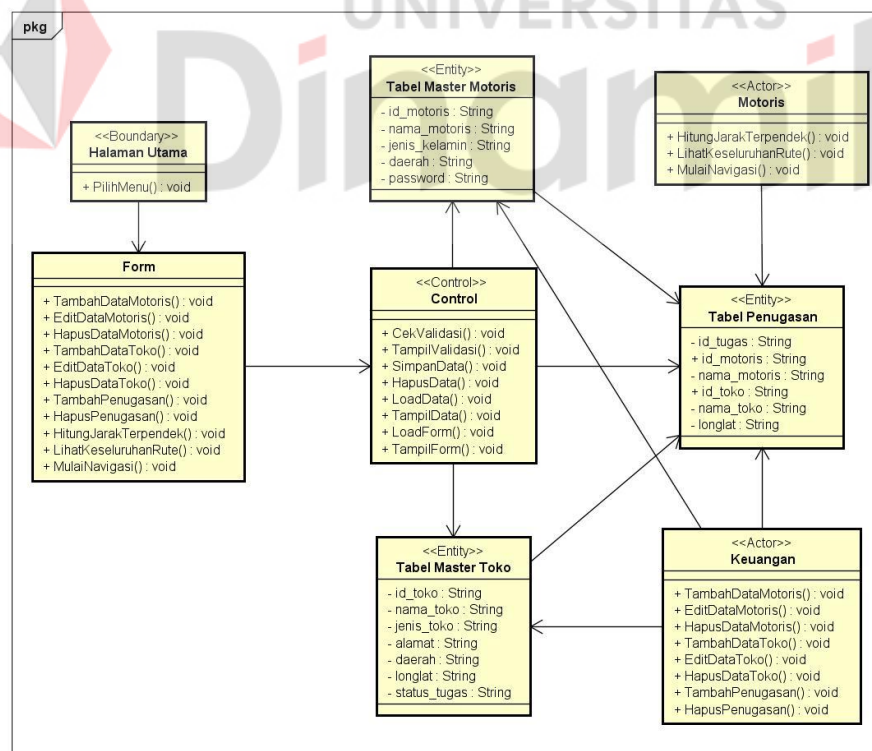
Sequence diagram navigasi perjalanan menunjukkan proses yang dialami sistem saat motoris ingin memulai navigasi menuju toko destinasi sesuai urutan dalam pencarian rute terpendek.



Gambar 3.8 Sequence diagram navigasi perjalanan

4. Class Diagram

Class diagram merupakan sebuah diagram yang menggambarkan struktur dan penjelasan objek, paket, serta hubungan satu dengan yang lain. Bentuk class diagram dapat dilihat pada Gambar 3.9.



Gambar 3.9 Class diagram

5. Struktur Tabel

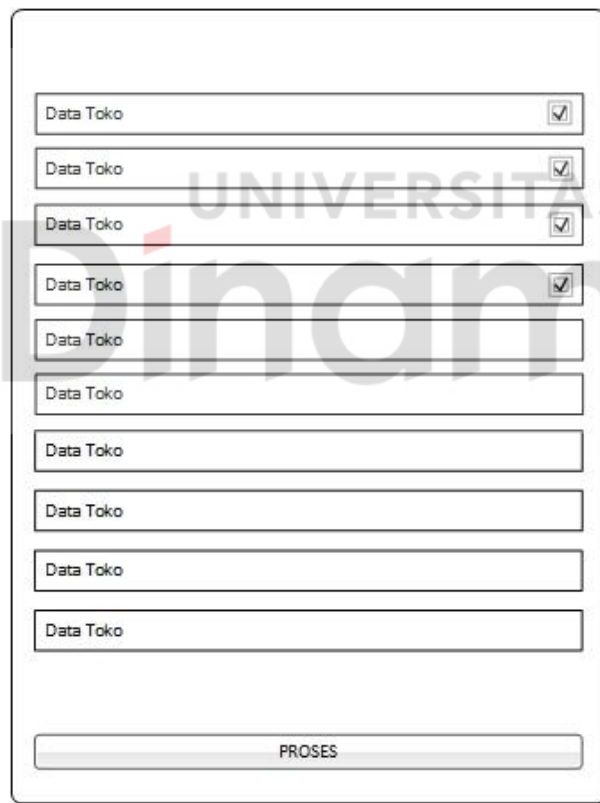
Struktur tabel menjelaskan bentuk tabel termasuk tipe data hingga karakteristik dari data-data tersebut. Struktur tabel dapat dilihat pada Lampiran 7.

6. Desain Rancangan Layar

Desain rancangan layar menggambarkan bentuk tampilan sistem yang nantinya akan dibuat. Bentuk desain rancangan layar untuk pilih toko tujuan, hitung rute terpendek, serta navigasi dapat dilihat pada Gambar 3.10 hingga gambar 3.1X, sedangkan untuk login, pengelolaan master, dan atur penugasan dapat dilihat pada Lampiran 8.

a. Desain Rancangan Layar Pilih Toko

Tampilan pilih toko didesain dengan penanda bahwa toko yang akan dituju telah dipilih, serta memunculkan *dialog* konfirmasi toko dipilih.



Gambar 3.10 Desain rancangan layar pilih toko

The image shows a mobile application screen with a list of 'Data Toko' items, each with a checkbox. A modal dialog titled 'Cari Rute Terdekat' is open, asking 'Apakah anda yakin akan mencari rute terdekat dari rute terpilih?' with 'Tidak' and 'Ya' buttons. Below the dialog are two more 'Data Toko' items and a 'PROSES' button.

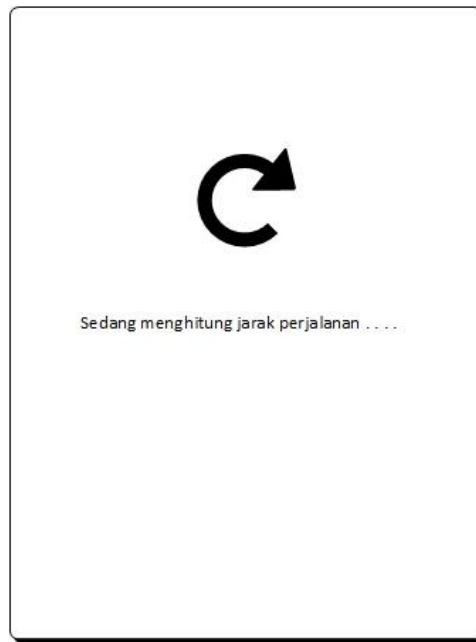
Gambar 3.11 Desain rancangan layar *dialog* konfirmasi

b. Desain Rancangan Layar Hitung Rute Terpendek

Desain layar hitung rute terpendek menampilkan *dialog* pilihan destinasi akhir yang dituju, serta *loading spinner* sebagai penanda proses penghitungan telah berjalan.

The image shows a mobile application screen with a dialog titled 'Pilih Titik Kembali' containing the text 'Tentukan titik kembali perjalanan anda'. Below the text are two buttons: 'PT Saka Mitra Usaha' and 'Lokasi Saat Ini'.

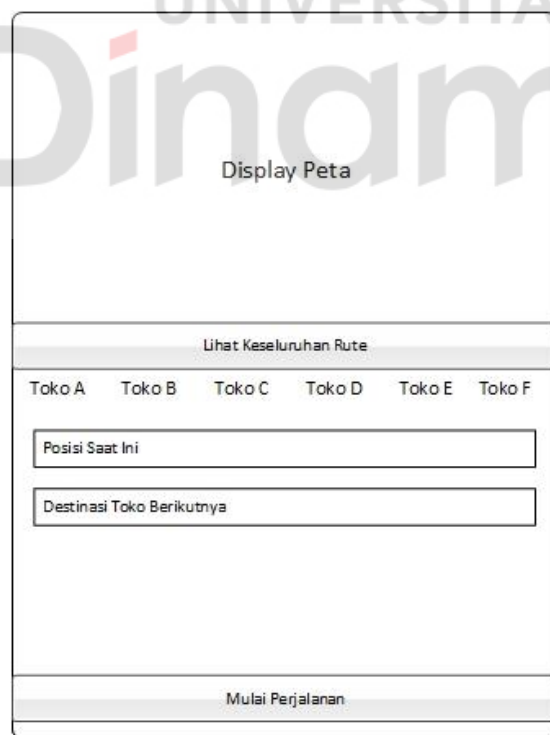
Gambar 3.12 Desain rancangan layar *dialog* pilih titik kembali




Gambar 3.13 Desain rancangan layar *loading spinner* proses

c. Desain Rancangan Layar Navigasi

Desain layar navigasi menampilkan dua fitur utama, yaitu navigasi itu sendiri serta opsi untuk melihat keseluruhan rute.



Gambar 3.14 Desain rancangan layar navigasi



Display Peta

Toko A

Toko B

Toko C

Toko D

Toko E

UNIVERSITAS
Dinamika

Gambar 3.15 Desain rancangan layar cek keseluruhan rute

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pencarian Rute Terpendek

Pada penulisan judul ini, digunakan sebanyak 15 titik lokasi toko sebagai acuan penghitungan rute terpendek dengan jumlah alpha, beta, dan evaporasi yang telah ditentukan terlebih dahulu.

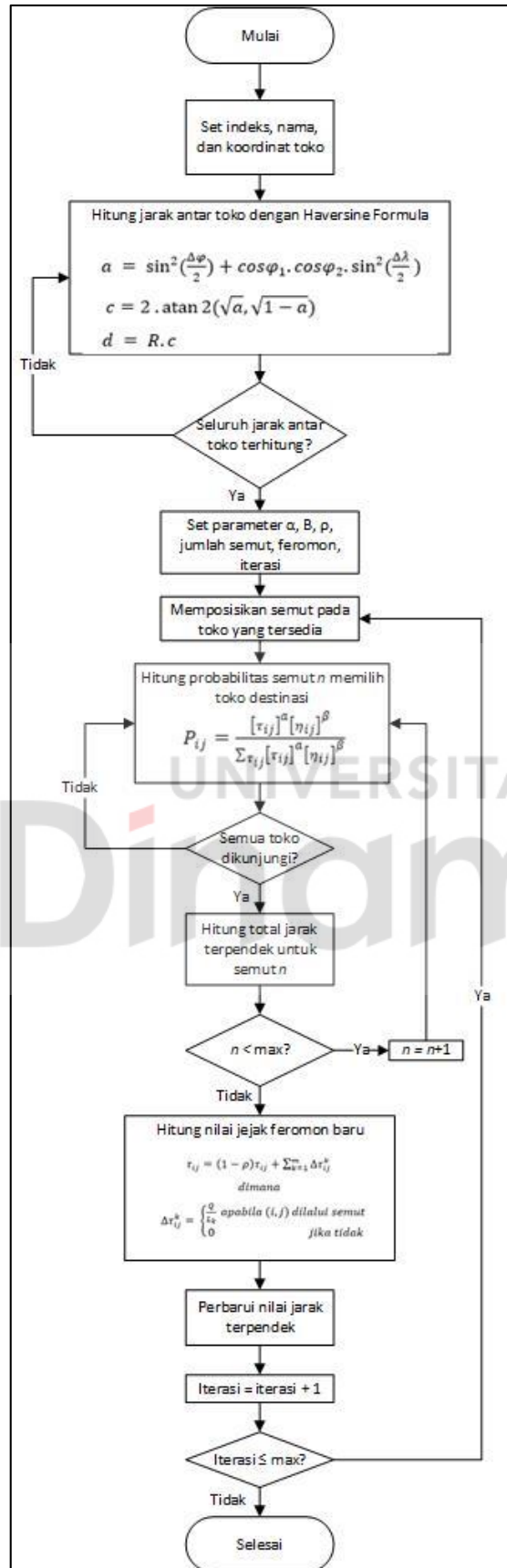
4.1.1 Penjelasan Metode

Terdapat variabel penentu yang digunakan dalam metode *Ant Colony Optimization*, diantaranya adalah alpha, beta, banyak jejak, evaporasi (rho), jumlah semut, dan jumlah iterasi. Keseluruhan variabel dapat dilihat pada Tabel 4.1.

Tabel 4.1 Variabel penentu

Nama Variabel	Nilai
Alpha	0, 0.5, 1, 1.5, 2
Beta	0, 0.5, 1, 1.5, 2, 3
Evaporasi (rho)	0.01, 0.05, 0.1, 0.5, 0.9, 1
Jumlah semut	2, 4, 8, 12, 15
Jejak feromon	1
Jumlah feromon	100
Jumlah iterasi	100

Variabel alpha dan beta masing-masing digunakan untuk menentukan prioritas dalam pemilihan rute, semakin besar alpha maka pemilihan rute akan memprioritaskan jarak antar toko, dan semakin besar beta maka pemilihan rute akan memprioritaskan besar jejak feromon yang berada pada rute toko. Evaporasi mempengaruhi banyaknya feromon yang menguap atau memudar pada jejak di setiap iterasinya. Jumlah semut menunjukkan banyaknya semut yang akan memulai perjalanan dalam setiap iterasinya. Dalam pengujian ini digunakan iterasi penghitungan sebanyak 100 kali tiap kombinasinya. Untuk lebih jelasnya, penghitungan jarak terpendek dapat dilihat pada Gambar 4.1.



Gambar 4.1 Proses penghitungan jarak terpendek

4.1.2 Penghitungan Rute

Penghitungan rute terpendek diawali dengan menentukan toko-toko yang akan digunakan sebagai destinasi. Dalam pengujian metode *Ant Colony Optimization* ini digunakan sebanyak 15 titik toko. Nama dan lokasi titik koordinat toko dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data toko

Index	Nama Toko	Koordinat
S	PT SAKA MITRA USAHA	-7.32817275728516, 112.73605094320968
0	WARUNG LAMPUNG	-7.376543149805799, 112.75735536328628
1	WARUNG RAISA	-7.402408188965115, 112.74898174029943
2	WARUNG BANG WETAN	-7.3758506350550945, 112.75765525937005
3	WARUNG ANTI GALAU	-7.384334527230434, 112.76001969381953
4	WARUNG 86	-7.394728831913482, 112.78188026374295
5	WARUNG EKA	-7.345327259739333, 112.70118556540757
6	WARUNG LESTY	-7.400015497999656, 112.72507015430347
7	WARUNG ALI	-7.354837115583754, 112.73286644800386
8	WARUNG KOPI TIKUS	-7.3713664805385894, 112.73175595125169
9	WARUNG REOG	-7.3712015001496685, 112.73178716936599
10	WARUNG OYI ALOHA	-7.3717601921809965, 112.72851987202225
11	WARUNG 46	-7.36872456211297, 112.73746937773382
12	WARUNG JOKO	-7.355982816881518, 112.72045435780366
13	WARUNG LESUH	-7.454714526106538, 112.73728108740649
14	WARUNG KITA	-7.387518112527293, 112.72824824904995

Setelah didapat titik koordinat tiap toko, maka langkah selanjutnya adalah menghitung jarak antar toko menggunakan metode Haversine Formula. Sebagai contoh, akan dihitung jarak untuk Toko 0 ke Toko 1. Menggunakan Haversine Formula, maka langkah pertama adalah mencari nilai α melalui pencarian $\Delta\varphi$ dan $\Delta\lambda$, yang secara berurutan menandakan selisih dari garis lintang dan garis bujur. Selisih derajat lintang dan bujur yang telah dihitung akan diubah terlebih dahulu kedalam satuan radian melalui perkalian dengan $\frac{\pi}{180}$.

$$\Delta\varphi = \varphi_1 - \varphi_0 = -0.02586503916 = -4.51430094 \times 10^{-4} \text{ rad}$$

$$\Delta\lambda = \lambda_1 - \lambda_0 = -0.008373622987 = -1.46147291 \times 10^{-4} \text{ rad}$$

Setelah penghitungan selisih lintang dan bujur telah dilakukan, maka dapat dilanjutkan untuk menghitung nilai $\sin^2\left(\frac{\Delta\varphi}{2}\right)$ dan $\sin^2\left(\frac{\Delta\lambda}{2}\right)$, serta menghitung nilai $\cos\varphi_0$ dan $\cos\varphi_1$. Nilai $\sin^2\left(\frac{\Delta\varphi}{2}\right)$ dan $\sin^2\left(\frac{\Delta\lambda}{2}\right)$ secara berurutan adalah $5.09472816 \times 10^{-8} \text{ rad}$ dan $5.33975766 \times 10^{-9} \text{ rad}$, sedangkan $\cos\varphi_0$ dan $\cos\varphi_1$ adalah $-0.1287449654 \text{ rad}$ dan 0.9916657481 rad . Setelah nilai

$\sin^2\left(\frac{\Delta\varphi}{2}\right)$, $\sin^2\left(\frac{\Delta\lambda}{2}\right)$, $\cos\varphi_0$, dan $\cos\varphi_1$ telah ditemukan, maka penghitungan nilai a dapat dilakukan dengan mengacu pada Rumus 2.3.

$$a = 5.09472816 \times 10^{-8} \text{ rad} + (-0.1287449654 \text{ rad}) \times 0.9916657481 \text{ rad} \\ \times 5.33975766 \times 10^{-9} \text{ rad} = 5.61987118 \times 10^{-8} \text{ rad}$$

Nilai a yang ditemukan dapat langsung digunakan untuk mencari nilai c dengan mengacu pada Rumus 2.4.

$$c = 2 \cdot \arctan 2(\sqrt{5.61987118 \times 10^{-8} \text{ rad}}, \sqrt{1 - 5.61987118 \times 10^{-8} \text{ rad}}) \\ c = 4.741253547267502 \times 10^{-4}$$

Setelah hasil c diperoleh, maka jarak antar titik toko (d) dapat langsung dikalikan dengan radius bumi sebesar 6371 kilometer, sesuai dengan Rumus 2.5.

$$d = 6371 \times 4.741253547267502 \times 10^{-4} = 3.021 \text{ km}$$

Keseluruhan penghitungan jarak antar toko dapat dilihat pada Tabel 4.3



UNIVERSITAS
Dinamika

Tabel 4.3 Jarak antar toko

Index Toko	Jarak Antar Toko (Km)														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	3.021	0.084	0.915	3.377	7.101	4.414	3.622	2.881	2.881	3.224	2.359	4.668	8.970	3.434
1	3.021	0	3.104	2.350	3.727	8.250	2.650	5.580	3.940	3.954	4.087	3.955	6.045	5.958	2.823
2	0.084	3.104	0	0.979	3.397	7.092	4.487	3.596	2.899	2.899	3.245	2.363	4.659	9.052	3.493
3	0.915	2.350	0.979	0	2.673	7.804	4.230	4.441	3.434	3.439	3.744	3.033	5.383	8.218	3.521
4	3.377	3.727	3.397	2.673	0	10.458	6.292	6.992	6.107	6.112	6.415	5.687	8.028	8.287	5.968
5	7.101	8.250	7.092	7.804	10.458	0	6.627	3.650	4.444	4.435	4.210	4.773	2.433	12.798	5.560
6	4.414	2.650	4.487	4.230	6.292	6.627	0	5.097	3.270	3.288	3.165	3.738	4.923	6.229	1.433
7	3.622	5.580	3.596	4.441	6.992	3.650	5.097	0	1.842	1.824	1.942	1.626	1.375	11.117	3.669
8	2.881	3.940	2.899	3.434	6.107	4.444	3.270	1.842	0	0.019	0.360	0.695	2.116	9.288	1.837
9	2.881	3.954	2.899	3.439	6.112	4.435	3.288	1.824	0.019	0	0.366	0.684	2.104	9.306	1.856
10	3.224	4.087	3.245	3.744	6.415	4.210	3.165	1.942	0.360	0.366	0	1.043	1.967	9.275	1.752
11	2.359	3.955	2.363	3.033	5.687	4.773	3.738	1.626	0.695	0.684	1.043	0	2.351	9.562	2.324
12	4.668	6.045	4.659	5.383	8.028	2.433	4.923	1.375	2.116	2.104	1.967	2.351	0	11.134	3.610
13	8.970	5.958	9.052	8.218	8.287	12.798	6.229	11.117	9.288	9.306	9.275	9.562	11.134	0	7.538
14	3.434	2.823	3.493	3.521	5.968	5.560	1.433	3.669	1.837	1.856	1.752	2.324	3.610	7.538	0

Setelah jarak antar toko telah didapat, maka penghitungan rute terpendek dapat langsung dimulai. Kombinasi uji coba variabel yang akan digunakan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kombinasi variabel ujicoba

Nama Variabel	Nilai
Alpha	1.5
Beta	1.0
Evaporasi (rho)	0.9
Jumlah semut	8
Jejak Feromon	1
Jumlah Feromon	100
Jumlah iterasi	100

Penggunaan jumlah semut sebanyak 8 dipilih terlebih dahulu dengan posisi titik toko berurutan. Titik awal setiap semut dapat dilihat pada Tabel 4.5.

Tabel 4.5 Titik awal semut uji coba

Semut	Index Toko	Nama Toko
Semut 1	0	WARUNG LAMPUNG
Semut 2	1	WARUNG RAISA
Semut 3	2	WARUNG BANG WETAN
Semut 4	3	WARUNG ANTI GALAU
Semut 5	4	WARUNG 86
Semut 6	5	WARUNG EKA
Semut 7	6	WARUNG LESTY
Semut 8	7	WARUNG ALI

Untuk pengujian ini, semut yang digunakan adalah Semut 1 dengan titik awal Toko 0 (WARUNG LAMPUNG). Semut 1 akan menghitung probabilitas kemungkinan setiap toko yang memungkinkan untuk dikunjungi. Untuk menghitung probabilitas Semut 1 memilih Toko 1 sebagai destinasi dapat dilakukan melalui Rumus 2.1.

$$P_{01} = \frac{[1]^{1.5}[0.331]^{1.0}}{10.719} = 0.730$$

Bentuk penghitungan untuk probabilitas toko kedua yang memungkinkan untuk dikunjungi dapat dilihat pada Tabel 4.6.

Tabel 4.6 Probabilitas pilihan toko kedua untuk Semut 1

Kota	τ	τ^α	d_{ij}	η_{ij}	$[\eta_{ij}]^\beta$	$[\tau]^\alpha[\eta_{ij}]^\beta$	$\sum[\tau]^\alpha[\eta_{ij}]^\beta$	Probabilitas Terpilih
0-1	1	1	3.021	0.331	0.331	0.331	10.719	0.020
0-2	1	1	0.084	11.932	11.932	11.932	10.719	0.730
0-3	1	1	0.915	1.093	1.093	1.093	10.719	0.067
0-4	1	1	3.377	0.296	0.296	0.296	10.719	0.018
0-5	1	1	7.101	0.141	0.141	0.141	10.719	0.009
0-6	1	1	4.414	0.227	0.227	0.227	10.719	0.014
0-7	1	1	3.622	0.276	0.276	0.276	10.719	0.017
0-8	1	1	2.881	0.347	0.347	0.347	10.719	0.021
0-9	1	1	2.881	0.347	0.347	0.347	10.719	0.021
0-10	1	1	3.224	0.310	0.310	0.310	10.719	0.019
0-11	1	1	2.359	0.424	0.424	0.424	10.719	0.026
0-12	1	1	4.668	0.214	0.214	0.214	10.719	0.013
0-13	1	1	8.970	0.111	0.111	0.111	10.719	0.007
0-14	1	1	3.434	0.291	0.291	0.291	10.719	0.018

Dari Tabel 4.6, dapat terlihat bahwa toko dengan probabilitas tertinggi untuk dituju adalah Toko 2 dengan jarak 0.084 km dari Toko 0. Keseluruhan penghitungan probabilitas untuk Semut 1 dapat dilihat pada Lampiran 9. Penghitungan akan terus dilakukan untuk toko-toko berikutnya hingga Semut 1 telah memilih semua toko dan menyelesaikan rutenya. Semut 2 hingga Semut 15 juga akan melakukan penghitungan probabilitas yang sama untuk mencapai rute optimal sesuai dengan titik awal perjalanan mereka. Setelah keseluruhan rute telah ditemukan, maka dapat langsung ditotal jarak keseluruhan, dengan tambahan titik awal dan akhir perjalanan pada PT Saka Mitra Usaha pada koordinat -7.32817275728516, 112.73605094320968. Keseluruhan rute optimal untuk iterasi pertama dapat dilihat pada Tabel 4.7

Tabel 4.7 Rute optimal seluruh semut pada iterasi pertama

Semut	Urutan Rute	Jarak (Km)
Semut 1	S-0-2-3-1-6-14-10-8-9-11-7-12-5-4-13-S	54.409
Semut 2	S-1-3-0-2-11-9-8-10-14-6-12-7-5-4-13-S	61.080
Semut 3	S-2-0-3-1-6-14-10-8-9-11-7-12-5-4-13-S	54.288
Semut 4	S-3-0-2-11-9-8-10-14-6-1-4-7-12-5-13-S	58.417
Semut 5	S-4-3-0-2-11-9-8-10-14-6-1-7-12-5-13-S	58.132
Semut 6	S-5-12-7-11-9-8-10-14-6-1-3-0-2-4-13-S	45.719
Semut 7	S-6-14-10-8-9-11-7-12-5-2-0-3-1-4-13-S	54.269
Semut 8	S-7-12-10-8-9-11-14-6-1-3-0-2-4-13-5-S	45.912

Setelah seluruh rute telah ditemukan, maka langkah berikutnya adalah memperbarui nilai jejak feromon berdasarkan variabel evaporasi feromon beserta jumlah rute dari toko awal hingga toko yang dilewati. Berdasarkan data yang didapat pada Tabel 4.7, diketahui bahwa rute untuk Toko 4 menuju Toko 3 hanya dilalui sekali yaitu dalam perjalanan Semut 5. Penghitungan jejak feromon Toko 4 ke Toko 3 dapat dilakukan dengan mengacu pada Rumus 2.2.

$$(1 - 0.01) \times 1 + \frac{100}{58.132} = 1.82$$

Untuk keseluruhan nilai jejak feromon baru pada iterasi satu dengan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Jejak feromon baru untuk iterasi 1.

Index Toko	Jejak Feromon														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0.10	0.10	11.37	3.78	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
1	0.10	0.10	0.10	6.10	3.65	0.10	3.78	1.82	0.10	0.10	0.10	0.10	0.10	0.10	0.10
2	3.78	0.10	0.10	1.94	4.47	0.10	0.10	0.10	0.10	0.10	0.10	5.17	0.10	0.10	0.10
3	9.53	5.62	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
4	0.10	0.10	0.10	1.82	0.10	0.10	0.10	1.81	0.10	0.10	0.10	0.10	0.10	11.63	0.10
5	0.10	0.10	1.94	0.10	5.42	0.10	0.10	0.10	0.10	0.10	0.10	0.10	2.29	3.53	0.10
6	0.10	7.90	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	1.74	0.10	5.62
7	0.10	0.10	0.10	0.10	0.10	1.74	0.10	0.10	0.10	0.10	0.10	2.29	11.23	0.10	0.10
8	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	7.80	7.36	0.10	0.10	0.10	0.10
9	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	7.36	0.10	0.10	7.80	0.10	0.10	0.10
10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	7.80	0.10	0.10	0.10	0.10	0.10	7.36
11	0.10	0.10	0.10	0.10	0.10	0.10	0.10	5.62	0.10	7.36	0.10	0.10	0.10	0.10	2.28
12	0.10	0.10	0.10	0.10	0.10	9.05	0.10	3.92	0.10	0.10	2.28	0.10	0.10	0.10	0.10
13	0.10	0.10	0.10	0.10	0.10	2.28	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
14	0.10	0.10	0.10	0.10	0.10	0.10	9.53	0.10	0.10	0.10	5.62	0.10	0.10	0.10	0.10

Pada penghitungan jejak feromon yang baru, semakin besar nilai jejak feromon maka semakin sering rute tersebut dipilih, semakin begitu juga sebaliknya. Penghitungan jejak feromon akan terus dilakukan setiap penghitungan keseluruhan rute dalam iterasi telah selesai.

Setelah penghitungan jejak telah dilakukan, maka penghitungan akan diulang kembali hingga mencapai batas maksimal iterasi. Langkah-langkah penghitungan akan dilakukan berulang kali dengan macam-macam kombinasi variabel sesuai dengan Tabel 4.1, dengan penghitungan tiap kombinasi variabel mencapai 750 penghitungan. Hasil penghitungan jarak terpendek yang didapat adalah sebesar 40.343 kilometer pada rute [S, 2, 0, 3, 4, 13, 1, 6, 14, 10, 8, 9, 11, 7, 12, 5, S]. Detail kombinasi untuk hasil yang didapat dapat dilihat di Tabel 4.9, sedangkan untuk hasil keseluruhan masing-masing kombinasi penghitungan dapat dilihat di Lampiran 10.

Tabel 4.9 Detail kombinasi

Banyak Semut	Alpha	Beta	Evaporasi	Jarak (km)	Rute
8	1.5	1.0	0.9	40.343	[S, 2, 0, 3, 4, 13, 1, 6, 14, 10, 8, 9, 11, 7, 12, 5, S]
8	2.0	1.5	0.5	40.343	[S, 2, 0, 3, 4, 13, 1, 6, 14, 10, 8, 9, 11, 7, 12, 5, S]
8	2.0	1.5	0.9	40.343	[S, 2, 0, 3, 4, 13, 1, 6, 14, 10, 8, 9, 11, 7, 12, 5, S]

4.1.3 Perbandingan Pengujian Manual

Berdasarkan hasil pengujian pada aplikasi, penulis melakukan pengecekan hasil pengujian radius pada penerapan kondisi jalan sebenarnya. Pengujian ini dilakukan untuk membuktikan bahwa hasil aplikasi mampu memberikan hasil rute yang lebih pendek daripada secara penghitungan manual. Uji coba manual yang dimaksud adalah memasukkan titik satu per satu setiap selesai menyelesaikan perjalanan ke salah satu toko dengan jarak terpendek pada Google Maps. Pengujian juga dilakukan menggunakan titik awal dan titik akhir PT Saka Mitra Usaha, pada titik koordinat -7.32817275728516, 112.73605094320968. Hasil pengecekan secara manual dapat dilihat pada Tabel 4.10, sedangkan untuk hasil pilihan setiap toko yang dilalui dapat dilihat pada Lampiran 11.

Tabel 4.10 Pengujian manual dengan Google Maps

Nama Toko	Jarak (Km)
PT SAKA MITRA USAHA menuju WARUNG ALI	4.4
WARUNG ALI menuju WARUNG REOG	2.7
WARUNG REOG menuju WARUNG KOPI TIKUS	0.019
WARUNG KOPI TIKUS menuju WARUNG OYI ALOHA	0.65
WARUNG OYI ALOHA menuju WARUNG 46	1.6
WARUNG 46 menuju WARUNG KITA	3.1
WARUNG KITA menuju WARUNG LESTY	2.2

Nama Toko	Jarak (Km)
WARUNG LESTY menuju WARUNG RAISA	3.5
WARUNG RAISA menuju WARUNG ANTI GALAU	3.8
WARUNG ANTI GALAU menuju WARUNG LAMPUNG	1.5
WARUNG LAMPUNG menuju WARUNG BANG WETAN	0.084
WARUNG BANG WETAN menuju WARUNG 86	5.5
WARUNG 86 menuju WARUNG JOKO	11.1
WARUNG JOKO menuju WARUNG EKA	4.3
WARUNG EKA menuju WARUNG LESUH	13.4
WARUNG LESUH menuju PT SAKA MITRA USAHA	16.8
Total Jarak	74.653

Dari hasil pengujian secara manual yang dilakukan, didapatkan bahwa total keseluruhan jarak adalah 74.653 kilometer. Untuk melakukan perbandingan dengan tepat, dilakukan pengujian yang sama menggunakan Google Maps dengan rute yang telah ditentukan pada aplikasi yang telah dirancang. Hasil penghitungan ini dapat dilihat pada Tabel 4.11.

Tabel 4.11 Pengujian hasil aplikasi pada Google Maps

Urutan Rute Toko	Jarak (Km)
PT SAKA MITRA USAHA menuju WARUNG BANG WETAN	9.6
WARUNG BANG WETAN menuju WARUNG LAMPUNG	0.65
WARUNG LAMPUNG menuju WARUNG ANTI GALAU	2
WARUNG ANTI GALAU menuju WARUNG 86	3.5
WARUNG 86 menuju WARUNG LESUH	11.5
WARUNG LESUH menuju WARUNG RAISA	7.8
WARUNG RAISA menuju WARUNG LESTY	3.7
WARUNG LESTY menuju WARUNG KITA	2.4
WARUNG KITA menuju WARUNG OYI ALOHA	3.1
WARUNG OYI ALOHA menuju WARUNG KOPI TIKUS	0.65
WARUNG KOPI TIKUS menuju WARUNG REOG	0.019
WARUNG REOG menuju WARUNG 46	0.9
WARUNG 46 menuju WARUNG ALI	5.5
WARUNG ALI menuju WARUNG JOKO	5
WARUNG JOKO menuju WARUNG EKA	4.3
WARUNG EKA menuju PT SAKA MITRA USAHA	6.6
Total Jarak	67.219

Melalui perbandingan hasil pengujian secara manual dengan menggunakan aplikasi yang telah dirancang, dapat dilihat bahwa aplikasi mampu memberikan rekomendasi jarak yang lebih pendek yaitu sejauh 67.219 kilometer, 7.434 kilometer lebih pendek daripada penghitungan manual.

4.2 Hasil Implementasi

Implementasi meliputi login, pengelolaan data master, atur penugasan, pilih toko tujuan, penghitungan jarak terdekat, dan navigasi. Untuk implementasi pilih toko tujuan, penghitungan jarak terdekat, dan navigasi dapat dilihat pada Gambar

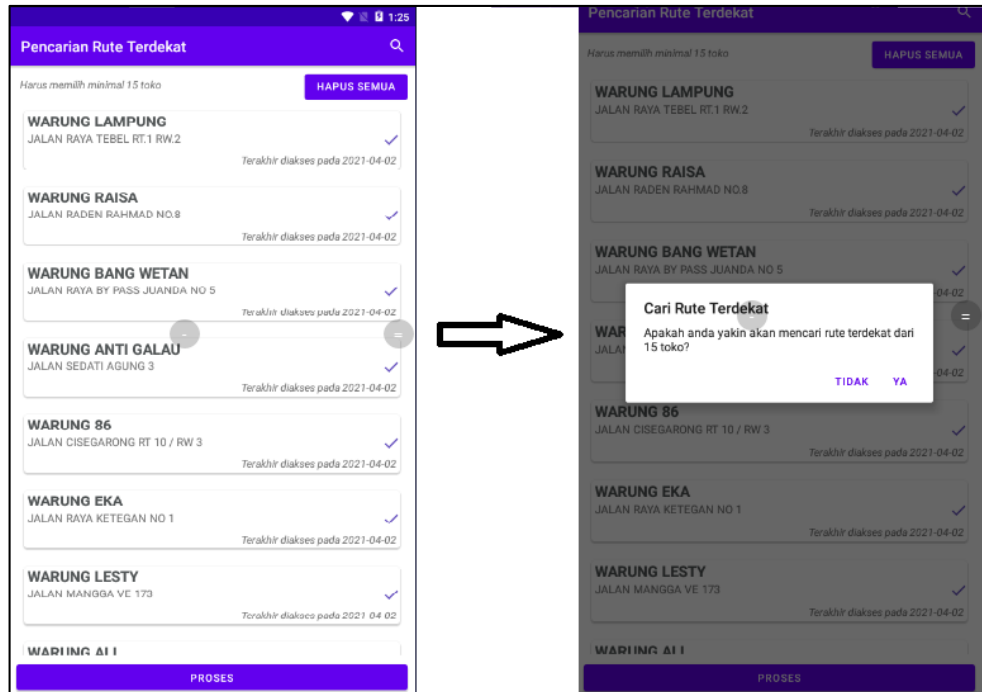
4.2 hingga Gambar 4.5, sedangkan untuk login, pengelolaan master, dan atur penugasan dapat dilihat pada Lampiran 12.

4.2.1 Pilih Toko Tujuan

Implementasi pilih toko tujuan menampilkan daftar toko penugasan yang tersedia beserta tiga tombol, yaitu pilih semua, pencarian, dan proses. Tombol pilih semua akan berubah menjadi hapus semua apabila ingin menghapus toko yang dipilih, dan tombol proses hanya akan muncul setelah 15 toko terpilih. Dialog konfirmasi akan muncul setelah motoris memilih tombol proses.



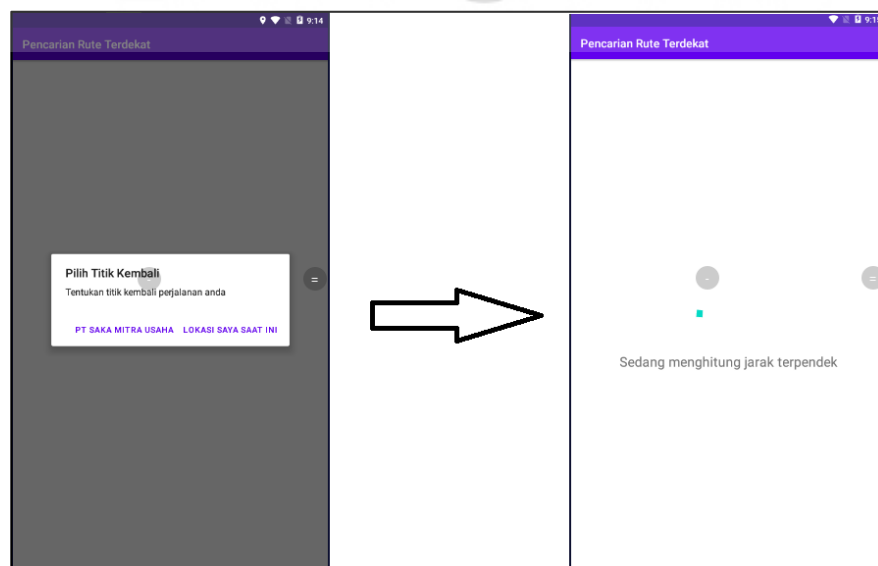
Gambar 4.2 Implementasi saat terjadi kesalahan



Gambar 4.3 Implementasi pilih toko tujuan

4.2.2 Hitung Rute Terpendek

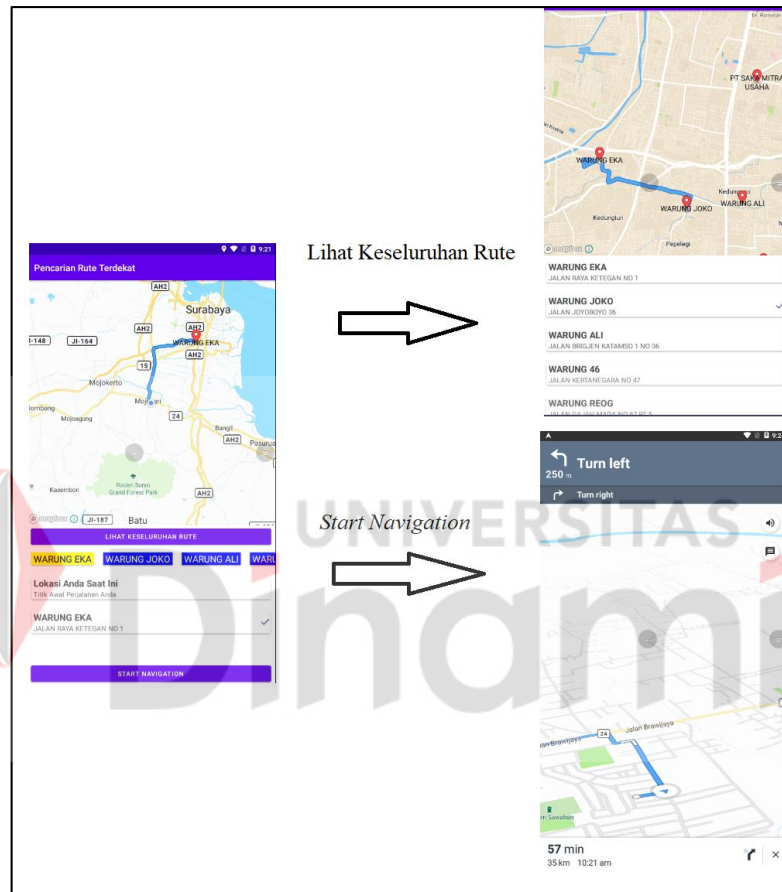
Pada penghitungan rute terpendek akan muncul dialog konfirmasi terhadap pilihan destinasi akhir motoris, akan kembali ke perusahaan atau kembali ke posisinya saat ini. Hasil konfirmasi tersebut akan mempengaruhi destinasi akhir motoris dalam perjalanan.



Gambar 4.4 Implementasi halaman hitung rute terpendek

4.2.3 Navigasi

Halaman navigasi akan memberikan tampilan peta beserta daftar toko destinasi secara berurutan. Tombol *start navigation* akan menyala biru ketika motoris memilih nama toko berikutnya yang akan dituju, serta menampilkan rute pada peta. Apabila motoris ingin melihat keseluruhan rute, motoris bisa mengaksesnya melalui tombol lihat keseluruhan rute.



Gambar 4.5 Implementasi halaman navigasi konfirmasi pilih toko tujuan

4.3 Hasil *Testing* Implementasi

Hasil *testing* akan proses input terhadap aplikasi pencarian rute terpendek adalah dapat dilihat pada Lampiran 12. Berdasarkan hasil *testing*, dapat dibuktikan bahwa:

1. Aplikasi dapat membantu motoris dalam melakukan pencarian rute terdekat. Setelah melakukan pengujian secara aplikasi dan manual, hasil pengujian menunjukkan bahwa hasil penghitungan dengan aplikasi lebih optimal dengan selisih 7.434 kilometer.

2. Aplikasi juga mampu memberikan integrasi hasil pencarian rute terpendek dengan navigasi perjalanan, sehingga motoris tidak perlu memasukkan satu per satu toko destinasi seperti pada Google Maps dan dapat langsung memulai perjalanan.

4.4 Evaluasi

Dari hasil pengujian yang telah dilakukan, maka didapatkan jarak terpendek dari 15 toko yang dituju sejauh 40.343 kilometer (67.219 kilometer secara manual), dengan kombinasi:

1. Jumlah semut = 8, $\alpha = 1.5$, $\beta = 1$, evaporasi = 0.9
2. Jumlah semut = 8, $\alpha = 2$, $\beta = 1.5$, evaporasi = 0.5 – 0.9



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang didapat pada pembuatan aplikasi pencarian rute terpendek adalah sebagai berikut:

1. Aplikasi ini dapat memberikan rekomendasi rute terpendek, sehingga mampu mempersingkat waktu perjalanan, dibuktikan dengan hasil penghitungan jarak terpendek didapat sebesar 40.343 kilometer secara radius. Kombinasi optimal yang digunakan adalah $\alpha = 1.5$, $\beta = 1$, $\rho = 0.9$ atau $\alpha = 2$, $\beta = 1.5$, dan $\rho = 0.5 - 0.9$.
2. Hasil pembuktian penghitungan jarak secara radius juga memberikan penghitungan jarak yang lebih pendek apabila dibandingkan dengan penghitungan manual, terbukti bahwa ditemukan jarak sebenarnya sejauh , 67.219 kilometer dengan jarak jalan melalui Google Maps, sedangkan hasil penghitungan manual didapat 74.653 kilometer.
3. Aplikasi ini dapat memberikan bantuan integrasi hasil pencarian rute terpendek dengan navigasi untuk membantu motoris dalam melakukan perjalanan menuju tiap-tiap toko yang telah dipilih, sehingga motoris tidak perlu mencari satu per satu destinasi tujuan seperti dengan Google Maps.
4. Aplikasi ini juga membantu bagian administrasi dalam mengelola data motoris, data toko, dan data penugasan.

5.2 Saran

Dalam pembuatannya, aplikasi pencarian rute terpendek ini masih memiliki beberapa hal yang dapat diperbaiki, diantaranya:

1. Aplikasi dapat memberikan tampilan desain yang lebih baik.
2. Aplikasi dapat langsung menghitung rute terpendek berdasarkan jarak sebenarnya, bukan radius.
3. Penghitungan rute terpendek dalam aplikasi dikombinasikan dengan *server* sehingga kecepatan komputasi dapat meningkat.

DAFTAR PUSTAKA

- Chudasama, C., Shah, S., & Panchal, M. (2011). Comparison of Parents Selection Methods of Genetic Algorithm for TSP. Maui: International Journal of Computer Applications.
- Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. Cambridge: MIT Press.
- Dorigo, M., & Stutzle, T. (2010). Ant Colony Optimization: Overview and Recent Advances. In J.-Y. P. Michel Gendreau, *Handbook of Metaheuristics* (p. 648). Boston: Springer.
- Gupta, S., & Poonam, P. (2013). Solving Travelling Salesman Problem Using Genetic Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering* , 3(6), 376-280.
- Hartanto, S., Furqan, M., Siahaan, A., & Fitriani, W. (2017). Haversine Method in Looking for the Nearest Masjid. *International Journal of Engineering Research*, 3(8), 187-195.
- Irawan, B., Setianingsih, C., & Arramsyah, I. (2018, April 1). Rancang Bangun Aplikasi Mobile Untuk Menentukan Solusi Optimal Pencarian Rute Terbaik Menggunakan Algoritma Ant Colony Optimization. *ILKOM Jurnal Ilmiah*, 10(1), 17-27.
- Katiyar, S., Ibraheem, N., & Ansari, A. Q. (2015). Ant Colony Optimization: A Tutorial Review. *National Conference on Advances in Power and Control*. Faridabad.
- Nurcahayanty Tanjung, W., & Sopiah. (2014). Implementing Ant Colony Optimization (ACO) In Traveling Salesman Problem (Case Study At Distribution Store PB). *Proceeding 7th International Seminar on Industrial Engineering and Management* (pp. OR-18 — OR-30). Bali: International Seminar on Industrial Engineering & Management.
- Pressman, R. S. (2010). *Software Engineering: A Practitioners Approach 7th Edition*. New York: McGraw-Hill.
- Raghavendra, B. V. (2015). Solving Travelling Salesmen Problem using Ant Colony Optimization Algorithm. *Journal of Information Sciences and Computing Technologies*, 170-179.