



**SISTEM INFORMASI GEOGRAFI UNTUK PEMETAAN SEBARAN  
PENJEMPUTAN IKAN DENGAN METODE *ANT COLONY OPTIMIZATION*  
(ACO)**

Studi Kasus : UD. Mitra Samudra Situbondo

**SKRIPSI**

Oleh

**Aglendy Rois Oktavirdi**

**NIM 122410101073**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2017**



**SISTEM INFORMASI GEOGRAFI UNTUK PEMETAAN SEBARAN  
PENJEMPUTAN IKAN DENGAN METODE *ANT COLONY OPTIMIZATION*  
(ACO)**

Studi Kasus : UD. Mitra Samudra Situbondo

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Sistem Informasi (S1)  
dan mencapai gelar Sarjana Komputer

Oleh

**Aglendy Rois Oktavirdi**

**NIM 122410101073**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**

## PERSEMBAHAN

Skripsi ini penulis persembahkan kepada :

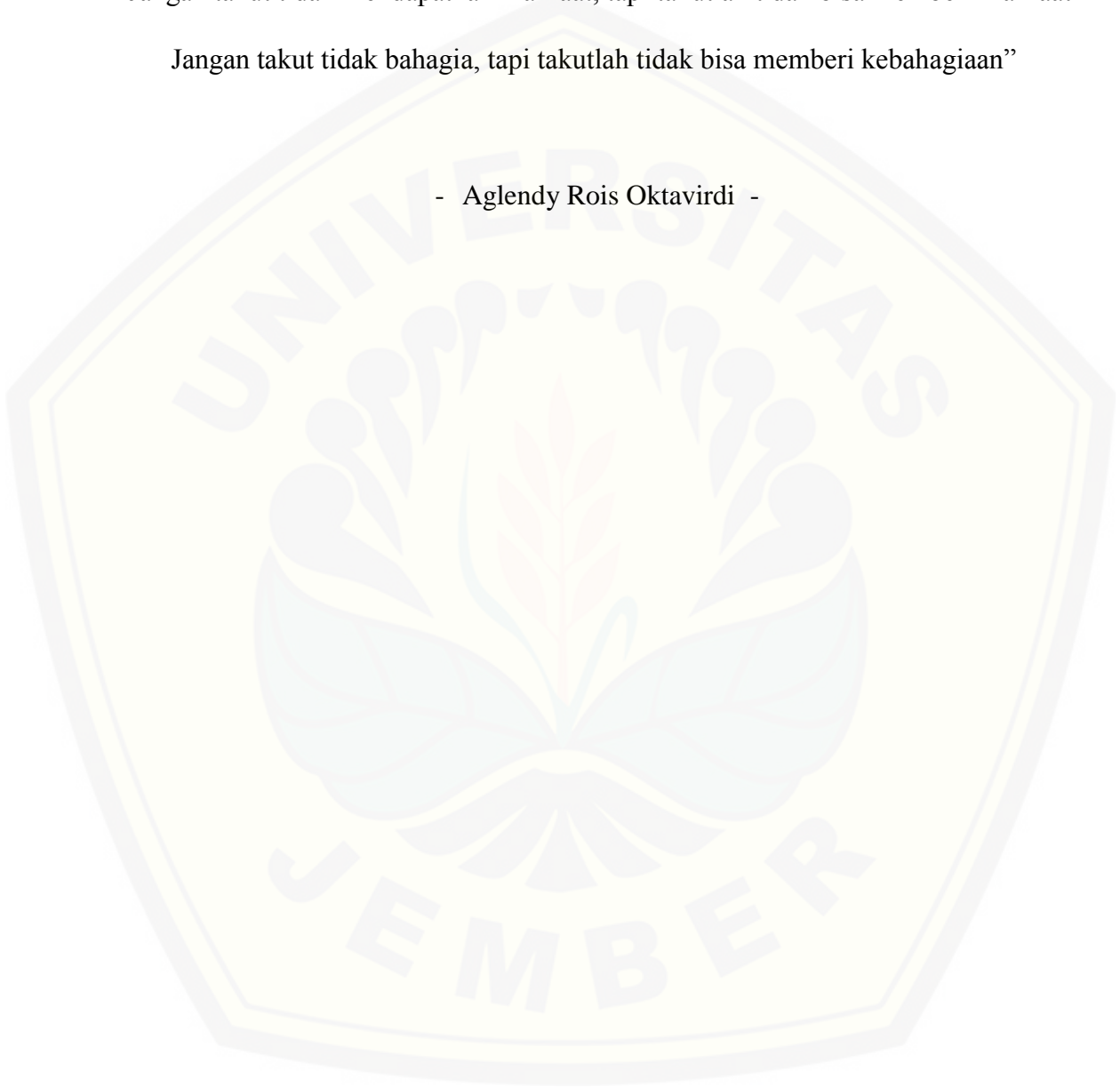
1. Bp. Tolak Atin dan Ibu Ika Yuliana yang selalu memberikan motivasi dan inspirasi penulis untuk menyusun skripsi ini;
2. Adik-adik tersayang Nandya Maulibi Safitri, Intan Mareta Navisa, dan Moch. Raihan Athallah Putra;
3. Ayu Intan Permatasari beserta keluarga yang selalu memberikan dukungan dan semangat;
4. Keluarga besar Unit Kegiatan Mahasiswa Kesenian Etalase;
5. Sahabat-sahabatku bersama dukungan dan doanya;
6. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

**MOTTO**

“Jangan takut tidak mendapatkan manfaat, tapi takutlah tidak bisa memberi manfaat

Jangan takut tidak bahagia, tapi takutlah tidak bisa memberi kebahagiaan”

- Aglendy Rois Oktavirdi -



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Aglendy Rois Oktavirdi

NIM : 122410101073

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Geografi untuk Pemetaan Sebaran Penjemputan Ikan dengan Metode *Ant Colony Optimization* (ACO) Studi Kasus : UD. Mitra Samudra”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 12 Juli 2017

Yang menyatakan,

Aglendy Rois Oktavirdi

NIM 122410101073

**SKRIPSI**

**SISTEM INFORMASI GEOGRAFI UNTUK PEMETAAN SEBARAN  
PENJEMPUTAN IKAN DENGAN METODE *ANT COLONY OPTIMIZATION*  
(ACO)**

Studi Kasus : UD. Mitra Samudra Situbondo

Oleh

**Aglendy Rois Oktavirdi**

**NIM 122410101073**

Pembimbing :

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App., Sc., Ph.D

Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya S.Si., MT.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Informasi Geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization (ACO)*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Drs. Antonius Cahya P, M.App., Sc., Ph.D

NIP. 196909281993021001

Nelly Oktavia Adiwijaya S.Si., MT.

NIP. 198410242009122008

**PENGESAHAN PENGUJI**

Skripsi berjudul “Sistem Informasi Geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO)” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D.

NIP. 196704201992011001

Priza Pandunata, S.Kom., M.Comp.Sc

NIP. 19830131201504001

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP. 196704201992011001



## RINGKASAN

**Sistem Informasi Geografi untuk Pemetaan Sebaran Penjemputan Ikan dengan Metode *Ant Colony Optimization* (ACO);** Aglendy Rois Oktavirdi, 122410101073; 2017; Program Studi Sistem Informasi Universitas Jember.

Kabupaten Situbondo yang daerah fisiknya memanjang dari barat ke timur sepanjang Selat Madura dengan panjang pantai 152 km, secara geografis sangat potensial untuk usaha pemanfaatan hasil laut. Sekitar 30% masyarakat Situbondo saat ini yang memanfaatkan hasil laut. Masyarakat Situbondo yang memanfaatkan hasil laut berprofesi sebagai nelayan, petani tambak dan pebisnis ikan.

Kondisi alur distribusi ikan dari nelayan hingga ke supplier di Kabupaten Situbondo saat ini masih menggunakan cara manual. Informasi ketersediaan ikan dan proses transaksinya hanya melalui telepon, bahkan harus datang ke gudang-gudang penyimpanan ikan milik pengambang atau pengepul untuk mengetahui ketersediaan ikan. Kondisi tersebut sangat tidak efisien dalam hal tenaga, waktu dan biaya.

Sistem Informasi Geografi untuk Pemetaan Sebaran Penjemputan Ikan dengan Metode *Ant Colony Optimization* (ACO) ini memberikan fasilitas kemudahan informasi bagi UD. Mitra Samudra maupun pengepul-pengepul yang terdaftar sebagai member melalui media yang mudah diakses. Media tersebut adalah *website* yang dapat diakses dengan mudah dengan hanya menggunakan koneksi internet. Melalui media ini memudahkan UD. Mitra Samudra dalam *monitoring* ketersediaan ikan pada setiap pengepul yang terdaftar sebagai member, lokasi pengepul yang tersebar di seluruh Kabupaten Situbondo dapat diketahui melalui peta sebaran pengepul yang disediakan oleh sistem, serta menentukan rute terpendek penjemputan ikan kepada pengepul-pengepul yang direkomendasikan oleh sistem untuk dilakukan penjemputan. Rute terpendek penjemputan ikan dihasilkan dengan menerapkan metode *Ant Colony Optimization* (ACO).

## PRAKATA

Puji syukur kepada Allah SWT, atas segala limpahan rahmat, hidayat dan karuniaNya maka penulis dapat menyelesaikan skripsi yang berjudul “Sistem Rekomendasi Pemilihan Produk Batik pada Griya Batik di Kabupaten Jember Menggunakan Metode Weighted Product”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya P, M.App., Sc., Ph.D., selaku Dosen Pembimbing Utama dan Nelly Oktavia A, S.Si, MT., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
4. Seluruh keluarga besar Landangan Family;
5. Bapak Tolak Atin dan Ibu Ika Yuliana yang selalu mendukung dan mendoakan;
6. Adik – adik Nandya Maulibi Safitri, Intan Mareta Nafisa, dan Moch. Raihan Athallah Putra;
7. Sepupu seperjuangan Alfioni Arisa Putri dan Aboerizal Ahmed Koesaery;
8. Keluarga besar Ibu Nanik Setyowati;
9. Ayu Intan Permatasari yang selalu mendukung, memberikan semangat dan motivasi;
10. Sahabat seperjuangan yang selalu menemani dan memberikan semangat serta doa Satiya Indra Budi W., Moh. Ikkal, Unzila Zulkifli, Hipolitus Kresna D.,

Ardyan Sukma Bryantara, Yudha Herlambang C.P., Marceli Aditya T., M. Hendrianto A.P., Antonius Ari Sadewo, Dinda Nurmawati, Alfiyatul Himmah;

11. Febrianto Rama Anji;
12. Teman-teman seperjuangan FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
13. Keluarga besar Unit Kegiatan Mahasiswa Kesenian (UKMK) ETALASE, Yosafat Parulian, Moch. Fikri, Bagus Ananda, Yohanis Permadi, Faris Aminullah dan anggota lain yang tidak dapat disebutkan satu persatu;
14. Semua pihak yang tidak dapat disebutkan satu persatu;

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 12 Juli 2017

Penulis

**DAFTAR ISI**

SKRIPSI.....	i
MOTTO .....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN .....	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL.....	xvii
BAB 1. PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Tujuan dan Manfaat.....	3
2.1.1. Tujuan .....	3
2.1.2. Manfaat .....	4
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	5
BAB 2. TINJAUAN PUSTAKA.....	7
2.1. Konsep Pengelolaan Bisnis Ikan Laut.....	7
2.2. Sistem Informasi Geografis (SIG).....	9
2.2.1. Konsep Sistem Informasi Geografis .....	9
2.2.2. Penelitian Terdahulu .....	9

2.3.	Google Maps API.....	10
2.4.	Teori Graph .....	11
2.5.	Rute Terpendek .....	13
2.5.1.	Metode Konvensional .....	13
2.5.2.	Metode Heuristik.....	13
2.6.	<i>Travelling Sales Problem</i> (TSP).....	14
2.7.	Ant Colony Optimzation (ACO) .....	15
2.7.1.	Definisi .....	15
2.7.2.	Kelas ACO dilihat dari sudut pandang.....	15
2.7.3.	Probabilitas Pemilihan Jalur ACO .....	16
2.7.4.	Cara kerja ACO.....	17
2.7.5.	Algoritma ACO.....	18
2.7.6.	Penelitian terdahulu.....	22
<b>BAB 3.</b>	<b>METODOLOGI PENELITIAN .....</b>	<b>23</b>
3.1.	Jenis Penelitian .....	23
3.2.	Tahapan Penelitian .....	23
3.2.1.	Tahapan Analisis Kebutuhan .....	24
3.2.2.	Tahapan Desain Sistem .....	26
3.2.3.	Tahapan Implementasi Sistem .....	27
3.2.4.	Tahapan Pengujian Sistem .....	27
3.3.	Gambaran Umum Sistem .....	28
<b>BAB 4.</b>	<b>ANALISIS DAN PENGEMBANGAN SISTEM .....</b>	<b>29</b>
4.1.	Deskripsi Umum Sistem.....	29
4.1.1.	SOP ( <i>statement of purpose</i> ) .....	29
4.2.	Pengumpulan Data .....	30
4.3.	Analisis Data .....	30

4.4.	Perancangan Sistem.....	30
4.4.1.	Analisis Kebutuhan .....	30
4.4.2.	Desain Sistem.....	31
4.5.	Pengkodean .....	44
4.6.	Pengujian .....	46
4.6.1.	Metode <i>White Box</i> .....	47
4.6.2.	Metode <i>Black Box</i> .....	54
BAB 5.	HASIL DAN PEMBAHASAN .....	56
5.1.	Implementasi Algoritma <i>Ant Colony Optimization (ACO)</i> .....	56
5.2.	Hasil Pembuatan Sistem Informasi Geografi untuk Pemetaan Sebaran Penjemputan Ikan dengan Metode <i>Ant Colony Optimization (ACO)</i> berbasis Web .....	87
5.3.	Implementasi Metode <i>Ant Colony Optimization (ACO)</i> pada Sistem .....	97
5.4.	Pembahasan Sistem Informasi Geografi untuk Pemetaan Sebaran Penjemputan Ikan dengan Metode <i>Ant Colony Optimization (ACO)</i> .....	98
5.4.1.	Kelebihan Sistem .....	99
5.4.2.	Kelemahan Sistem.....	99
BAB 6.	PENUTUP .....	101
6.1.	Kesimpulan.....	101
6.2.	Saran .....	102
DAFTAR PUSTAKA	.....	103
LAMPIRAN	.....	105
LAMPIRAN A.	<i>USE CASE</i> SKENARIO.....	105
LAMPIRAN B.	<i>ACTIVITY</i> DIAGRAM .....	112
LAMPIRAN C.	<i>SEQUENCE</i> DIAGRAM.....	118
LAMPIRAN D.	KODE PROGRAM .....	122
LAMPIRAN E.	PENGUJIAN BLACK BOX .....	132



DAFTAR GAMBAR

Gambar 2.1 Alur ikan dari nelayan hingga pabrik atau pasar..... 8

Gambar 2.2 Komponen SIG (Sumber: Qolis, 2008)..... 9

Gambar 2.3 Graf berarah dan berbobot..... 11

Gambar 2.4 Graf tidak berarah dan berbobot..... 12

Gambar 2.5 Graf berarah dan tidak berbobot..... 12

Gambar 2.6 Graf tidak berarah dan tidak berbobot..... 13

Gambar 2.7 Cara Kerja ACO..... 17

Gambar 3.1 Fase – fase model *waterfall*..... 23

Gambar 3.2 Skema *Project Definition* ..... 25

Gambar 3.3 Gambaran umum sistem..... 28

Gambar 4.1 *Bussiness Process Fiss Pick-Up Tracking*.....48

Gambar 4.2 *Usecase Diagram Fish Pick-Up Tracking* ..... 33

Gambar 4.3 *Activity Diagram Login Admin* ..... 39

Gambar 4.4 *Sequence Diagram Login Admin*..... 41

Gambar 4.5 *Class Diagram*..... 43

Gambar 4.6 *Entity Relationship Diagram*..... 44

Gambar 2.1 Alur ikan dari nelayan hingga pabrik atau pasar..... 8

Gambar 2.2 Komponen SIG (Sumber: Qolis, 2008)..... 9

Gambar 2.3 Graf berarah dan berbobot..... 11

Gambar 2.4 Graf tidak berarah dan berbobot..... 12

Gambar 2.5 Graf berarah dan tidak berbobot..... 12

Gambar 2.6 Graf tidak berarah dan tidak berbobot..... 13

Gambar 2.7 Cara Kerja ACO..... 17

Gambar 3.1 Fase – fase model *waterfall*..... 23

Gambar 3.2 Skema <i>Project Definition</i> .....	25
Gambar 3.3 Gambaran umum sistem.....	28
Gambar 4.1 <i>Bussiness Process Fiss Pick-Up Tracking</i> .....	48
Gambar 4.2 <i>Usecase Diagram Fish Pick-Up Tracking</i> .....	33
Gambar 4.3 <i>Activity Diagram Login Admin</i> .....	39
Gambar 4.4 <i>Sequence Diagram Login Admin</i> .....	41
Gambar 4.5 <i>Class Diagram</i> .....	43
Gambar 4.6 <i>Entity Relationship Diagram</i> .....	44
Gambar 4.7 <i>Listing program function GetDrivingDistance() (1)</i> .....	69
Gambar 4.8 Diagram alit fitur rute terpendek.....	70
Gambar 4.9 <i>Listing program function otentikasi() (2)</i> .....	71
Gambar 4.10 Diagram alir fitur login.....	71
Gambar 4.11 <i>Listing program function editPass() (3)</i> .....	73
Gambar 4.12 Diagram alir fitur ganti password.....	73
Gambar 5.1 Node awal dan node tujuan peta sebaran pengepul.....	57
Gambar 5.2 Rute N1 ke node selanjutnya.....	60
Gambar 5.3 Rute N2 ke node selanjutnya.....	63
Gambar 5.4 Rute N10 ke node selanjutnya.....	64
Gambar 5.5 Rute N11 menuju node tujuan.....	65
Gambar 5.6 Rute N1 ke node selanjutnya.....	68
Gambar 5.7 Rute N16 ke node selanjutnya.....	71
Gambar 5.8 Rute N15 ke node selanjutnya.....	74
Gambar 5.9 Rute N13 menuju node tujuan.....	75
Gambar 5.10 Rute N1 ke node selanjutnya.....	78
Gambar 5.11 Rute N16 ke node selanjutnya.....	81
Gambar 5.12 Rute N10 ke node selanjutnya.....	82



Gambar 5.13 Rute N11 menuju node tujuan.....	83
Gambar 5.14 Rute terpendek yang dihasilkan dari perhitungan ACO.....	87
Gambar 5.15 halaman Beranda .....	88
Gambar 5.16 halaman Fitur.....	88
Gambar 5.17 halaman kontak .....	89
Gambar 5.18 halaman Login.....	90
Gambar 5.19 halaman Dashboard Admin.....	90
Gambar 5.20 halaman Rute Penjemputan.....	91
Gambar 5.21 halaman Daftar Pengepul .....	92
Gambar 5.22 halaman Form Pengepul.....	92
Gambar 5.23 halaman Data Ikan Pengepul.....	93
Gambar 5.24 halaman Detail Ikan .....	94
Gambar 5.25 halaman Dashboard member.....	94
Gambar B. 1 <i>Activity</i> Member – Login.....	112
Gambar B. 2 <i>Activity</i> Admin – Melihat peta sebaran pengepul.....	113
Gambar B. 3 <i>Activity</i> Admin – Menentukan rute terpendek.....	113
Gambar B. 4 <i>Activity</i> Admin – Mengelola data pengepul .....	114
Gambar B. 5 <i>Activity</i> Admin – Melihat data ikan pengepul .....	115
Gambar B. 6 <i>Activity</i> Member – Mengelola Profil .....	116
Gambar B. 7 <i>Activity</i> Member – Mengelola data ikan.....	117
Gambar C. 1 <i>Sequence</i> Member - Login.....	118
Gambar C. 2 <i>Sequence</i> Admin – Menentukan rute terpendek .....	118
Gambar C. 3 <i>Sequence</i> Admin – Mengelola data pengepul .....	119
Gambar C. 4 <i>Sequence</i> Admin – Melihat data ikan pengepul .....	120
Gambar C. 5 <i>Sequence</i> Member – Mengelola profil.....	120
Gambar C. 6 <i>Sequence</i> Member – Mengelola data ikan.....	121

## DAFTAR TABEL

Tabel 4. 1 Definisi Aktor .....	34
Tabel 4. 2 Deskripsi <i>Usecase</i> .....	34
Tabel 4. 3 Skenario Login Admin.....	36
Tabel 4. 4 Potongan kode program pada fitur rute terpendek.....	45
Tabel 4. 5 <i>Test Case function GetDrivingDistance ()</i> .....	49
Tabel 4. 6 <i>Test Case function otentikasi ()</i> .....	51
Tabel 4. 7 <i>Test Case editPass ()</i> .....	53
Tabel 5. 1 Menentukan node awal dan node tujuan.....	56
Tabel 5. 2 Inisialisasi parameter ACO .....	58
Tabel 5. 3 Hasil pencarian rute dan perhitungan dari siklus 1 .....	84
Tabel 5. 4 Intensitas feromon pada setiap node yang dilewati semut 1 .....	86
Tabel A. 1 <i>Use Case</i> Skenario Login Member .....	105
Tabel A. 2 <i>Use Case</i> Skenario Melihat peta sebaran pengepul .....	106
Tabel A. 3 <i>Use Case</i> Skenario Menentukan rute terpendek .....	106
Tabel A. 4 <i>Use Case</i> Skenario Mengelola data pengepul.....	107
Tabel A. 5 <i>Use Case</i> Skenario Melihat data ikan pengepul .....	108
Tabel A. 6 <i>Use Case</i> Skenario Mengelola profil .....	109
Tabel A. 7 <i>Use Case</i> Skenario Mengelola data ikan.....	110

## **BAB 1. PENDAHULUAN**

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

### **1.1. Latar Belakang**

Kabupaten Situbondo yang daerah fisiknya memanjang dari barat ke timur sepanjang Selat Madura dengan panjang pantai 152 km, secara geografis sangat potensial untuk usaha pemanfaatan hasil laut. Dengan sumberdaya perairan sebesar itu, jika diperlakukan secara serius akan dapat mensejahterakan dan meningkatkan Pendapatan Asli Daerah (PAD) Kabupaten Situbondo sendiri. Sekitar 30% masyarakat Situbondo saat ini yang memanfaatkan hasil laut, melalui perikanan tangkap maupun perikanan budidaya. Masyarakat Situbondo yang memanfaatkan hasil laut berprofesi sebagai nelayan, petani tambak dan pebisnis ikan.

Ada dua kategori hasil tangkapan atau hasil budidaya ikan laut yang dimanfaatkan sebagai bisnis oleh masyarakat Situbondo, yaitu Ikan Palagis dan Ikan Damersal. Ikan Palagis merupakan ikan yang memiliki jaringan minyak di tubuh dan rongga dalam perut sekitar usus. Ikan Palagis hidup di permukaan laut sampai kolom perairan laut dan biasanya membentuk gerombolan serta selalu melakukan migrasi. Ikan jenis ini tujuan pasarnya adalah pasar lokal. Sedangkan kategori Ikan Damersal yaitu jenis ikan yang habitatnya berada di karang atau bagian dasar perairan.

Kondisi proses distribusi ikan dari nelayan hingga ke supplier di Kabupaten Situbondo saat ini masih menggunakan cara manual. Alur suplai ikan yaitu, pertama ikan hasil tangkapan nelayan diterima oleh juragan darat atau pengambak, kemudian dijual kepada pengepul, kemudian dijual kepada supplier, dan akhirnya dipasok ke pasar atau pabrik lokal maupun interlokal. Informasi ketersediaan ikan dan proses transaksinya hanya melalui telepon, bahkan harus datang ke gudang-gudang penyimpanan ikan milik pengambak atau pengepul untuk mengetahui ketersediaan

ikan. Banyaknya pengepul yang tersebar di Kabupaten Situbondo, membuat kesulitan di pihak supplier.

UD. Mitra Samudra merupakan badan usaha yang berperan sebagai supplier dan fokus pada bisnis Ikan laut Damersal. UD. Mitra Samudra dalam proses pengadaan ikan harus aktif setiap saat menghubungi setiap pengepul yang tersebar di Kabupaten Situbondo, bahkan harus mendatangi satu persatu gudang pengepul untuk mengetahui ketersediaan ikan. Hal ini sangat tidak efektif dan tidak efisien dalam hal biaya. Selain itu masih banyak juga lokasi pengepul yang tidak diketahui alamatnya yang tersebar di sekitar Kabupaten Situbondo.

Berdasarkan masalah tersebut dibutuhkan suatu sistem teknologi yang berbasis internet yang mampu menampilkan data ikan yang tersedia pada gudang-gudang pengepul sehingga supplier dapat mengetahui ikan yang tersedia. Sistem juga harus mampu menggambarkan secara grafis lokasi sebaran pengepul di Kabupaten Situbondo dalam bentuk peta, yaitu dengan Sistem Informasi Geografis (SIG) memudahkan UD. Mitra Samudra untuk memetakan lokasi gudang pengepul untuk melakukan penjemputan ikan serta menampilkan ikan yang tersedia pada gudang setiap pengepul. Dalam merealisasikan Sistem Informasi Geografis, *Google Maps API* merupakan salah satu pilihan utama yang dapat digunakan. Selain memiliki API yang dapat diintegrasikan dengan beberapa teknologi, *Google Maps* juga merupakan layanan gratis yang menyediakan peta satelit dan peta hybrid.

Proses penjemputan ikan di gudang ikan, dimana supir UD. Mitra Samudra mengunjungi hanya sekali dari sejumlah gudang ikan pengepul dan kembali ke tempat pemberangkatan semula. Masalah ini merupakan *Travelling Salesman Problem* (TSP) yang sulit diselesaikan dengan algoritma eksak. Algoritma yang cocok untuk permasalahan ini biasanya menggunakan algoritma heuristik (Leksono, 2009).

Berdasarkan masalah *Travelling Salesman* yang biasanya menggunakan metode heuristik untuk menemukan solusinya. Metode *Ant Colony Optimization* (ACO) merupakan salah satu metode metaheuristik yang menerapkan semut sebagai agennya

dengan update *pheromone* yang berfungsi sebagai informasi bagi koloni semut untuk dapat menemukan proses pencarian solusi yang optimal. Metode ini merupakan teknik probabilistik untuk menyelesaikan masalah komputasi dengan menemukan jalur terbaik melalui graf.

Berdasarkan latar belakang tersebut diatas, penulis akan merancang dan mengimplementasikan “*Sistem Informasi Geografis Untuk Pemetaan Sebaran Penjemputan Ikan dengan metode Ant Colony Optimization (ACO)*”. Studi kasus: UD. Mitra Samudra, Kapongan, Situbondo.

## 1.2. Rumusan Masalah

Berdasarkan uraian diatas, permasalahan yang muncul adalah:

- a. Bagaimana merancang dan membangun sistem yang dapat menyediakan informasi stok ikan dari para pengepul terhadap UD. Mitra Samudra sebagai supplier?
- b. Bagaimana mengimplementasikan metode *Ant Colony Optimzation (ACO)* untuk menemukan rute terpendek dalam penjemputan ikan dari para pengepul di Kabupaten Situbondo?
- c. Bagaimana merancang dan membangun Sistem Informasi Geografi yang dapat menggambarkan secara grafis sebaran pengepul dan memberikan visual rute terpendek dalam penjemputan ikan di Kabupaten Situbondo?

## 1.3. Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

### 1.3.1. Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

- a. Untuk merancang dan membangun sebuah sistem teknologi yang mampu menyediakan informasi ketersediaan stok ikan dari setiap pengepul terhadap UD. Mitra Samudra sebagai supplier.



- b. Untuk mengimplementasikan metode *Ant Colony Optimization* (ACO) pada penentuan rute terpendek dalam penjemputan ikan dari para pengepul.
- c. Untuk merancang dan membangun sebuah sistem informasi geografis untuk menampilkan secara grafis sebaran pengepul dan memberikan visual rute terpendek dalam penjemputan ikan di Kabupaten Situbondo.

### 1.3.2. Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Bagi Instansi  
Sabagai sarana untuk mempermudah hubungan dalam jaringan bisnis antara UD. Mitra Samudra dan pengepul-pengepul yang tersebar di Kabupaten Situbondo sebagai mitra kerjanya dalam proses *supply chain* ikan.
- b. Bagi Akademis  
Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.
- c. Bagi Peneliti  
Mengetahui bagaimana proses penerapan *ant colony optimization* (ACO) pada sistem informasi geografis untuk pemetaan sebaran penjemputan ikan pada UD. Mitra Samudra di Kabupaten Situbondo.
- d. Bagi pihak lain  
Penelitian ini dapat dijadikan bahan referensi bagi peneliti lain yang ingin mengembangkan hasil penelitian ini di kemudian hari.

### 1.4. Batasan Masalah

Terdapat beberapa batasan masalah yang dapat diangkat dalam melakukan penelitian ini, antara lain sebagai berikut:

- a. Sistem ini disimulasikan di UD. Mitra Samudra.
- b. Sistem Informasi Geografis (SIG) digunakan untuk menggambarkan secara grafis lokasi sebaran pengepul di Kabupaten Situbondo.
- c. Metode *Ant Colony Optimzation* (ACO) digunakan untuk menentukan rute terpendek dalam penjemputan ikan dari para pengepul yang dihasilkan oleh sistem.
- d. Parameter optimal berdasarkan jarak yang ditempuh dan jalan satu arah.
- e. Sistem yang dibangun berbasis web.

### **1.5. Sistematika Penulisan**

Adapun sistematika penulisan skripsi adalah sebagai berikut:

- a. Bab 1. Pendahuluan

Bab ini memaparkan latar belakang penulis dalam melakukan penelitian terhadap studi kasus, serta merumuskan masalah, tujuan, dan batasan penelitian.
- b. Bab 2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep pemetaan, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.
- c. Bab 3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.
- d. Bab 4. Pengembangan Sistem

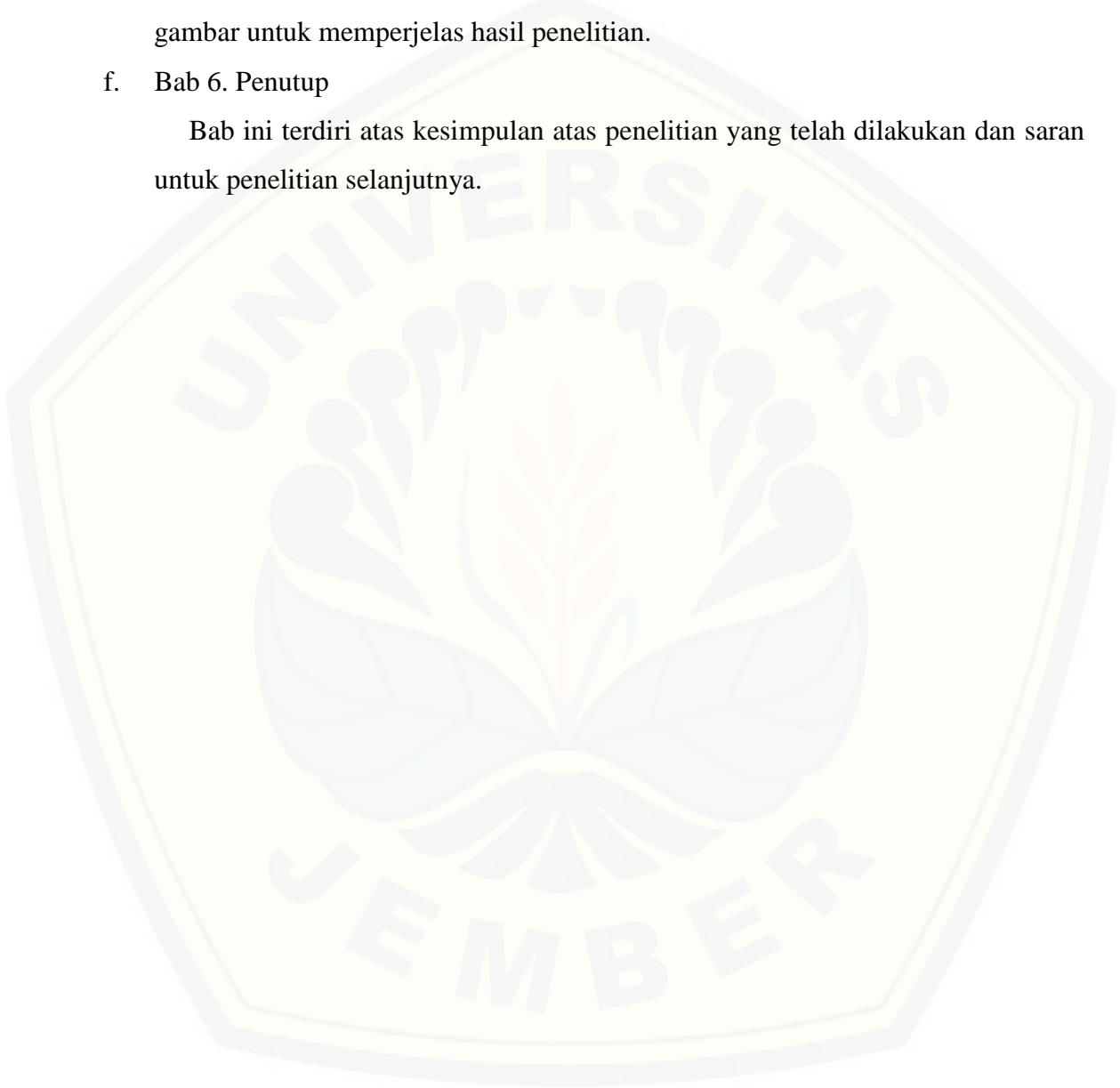
Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, kode program, dan pengujian sistem.

e. Bab 5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

f. Bab 6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.





## **BAB 2. TINJAUAN PUSTAKA**

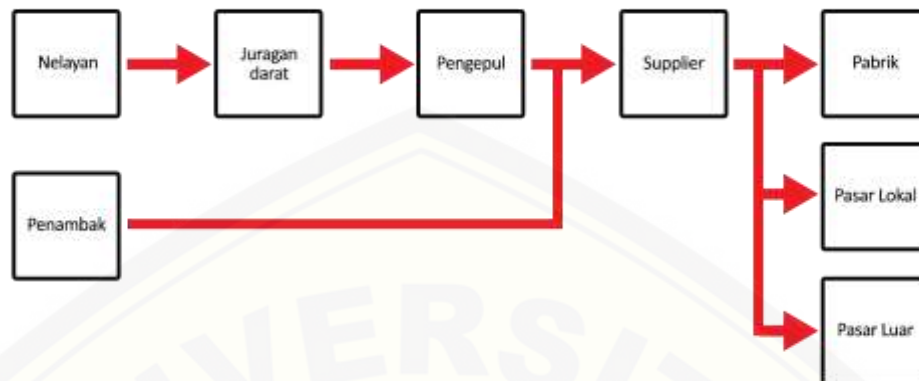
Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

### **2.1. Konsep Pengelolaan Bisnis Ikan Laut**

Berdasarkan Undang-Undang 45 Tahun 2009 tentang Perubahan atas Undang-Undang No. 31 Tahun 2004 Tentang Perikanan, yang dimaksud dengan perikanan adalah semua kegiatan yang berhubungan dengan pengelolaan dan pemanfaatan sumber daya ikan dan lingkungannya mulai dari praproduksi, produksi, pengolahan sampai dengan pemasaran yang dilaksanakan dalam suatu sistem bisnis perikanan. Dari pengertian tersebut, ternyata ruang lingkup bidang perikanan sangat luas, yang tidak hanya memanfaatkan sumberdaya ikan dan lingkungannya, tetapi juga mengelolanya.

Pengelolaan sumberdaya ikan merupakan suatu aspek yang sangat penting dalam sektor perikanan, dan ketidakmampuan dalam pengelolaan sumberdaya perikanan dapat mengakibatkan menurunnya pendapatan sektor perikanan yang berasal dari sumberdaya yang ada.

Pelaku bisnis dalam alur ikan hingga masuk pabrik atau pasar ada lima level, seperti gambar 2.1 dibawah ini.



Gambar 2.1 Alur ikan dari nelayan hingga pabrik atau pasar

- a. Level 1 (Nelayan atau Penambak)  
Nelayan merupakan pemilik perahu beserta awak kapalnya. Nelayan bertanggung jawab atas juragan darat selaku orang yang mensponsori nelayan, meliputi fasilitas dan logistik untuk kebutuhan melaut. Sedangkan penambak adalah petani atau pemilik tambak yang melakukan budidaya ikan untuk keperluan mendapatkan keuntungan. Penambak tidak harus melewati pengepul untuk menjual ikannya, penambak bisa langsung menjual ikannya ke supplier.
- b. Level 2 (Juragan darat)  
Orang yang memfasilitasi kegiatan melaut nelayan. Juragan darat mengeluarkan dana untuk bahan bakar, logistik, dan peralatan melaut, dan upah. Dengan begitu hasil tangkapan nelayan 70% adalah hak milik juragan darat.
- c. Level 3 (Pengepul)  
Orang yang mengepul dari hasil tangkapan nelayan atau juragan darat. Pengepul juga bisa bertindak sebagai juragan darat.
- d. Level 4 (Supplier)  
Orang yang membeli ikan kepada juragan darat atau pengepul. Kemudian ikan di suplai ke pabrik, pasar lokal, dan pasar luar. Supplier bisa berperan sebagai Juragan darat atau pengepul.
- e. Level 5 (Pasar atau Pabrik)

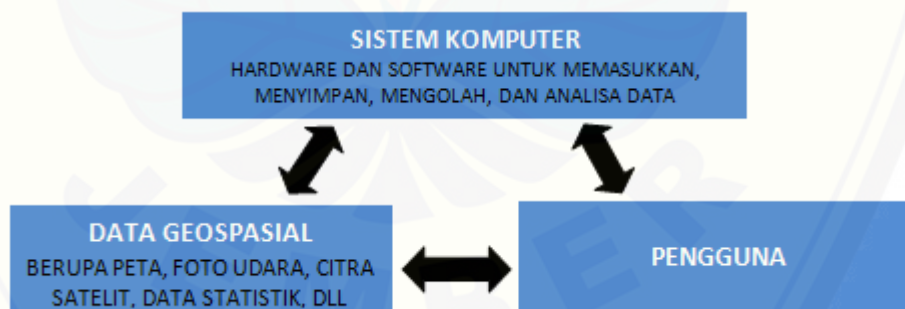
Level terakhir dari alur ikan sebelum akhirnya sampai ke konsumen atau diproduksi menjadi olahan.

## 2.2. Sistem Informasi Geografis (SIG)

### 2.2.1. Konsep Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) merupakan sistem yang berisi informasi yang berkaitan dengan pemetaan dan analisa terhadap segala sesuatu serta peristiwa yang terjadi di muka bumi (Prahasta, 2005). SIG mengintegrasikan proses pengolahan data berbasis database dengan mengguakan visualisasi yang khas melalui analisa geografis dalam bentuk peta digital.

Menurut Anon dalam Sugandi (2008) Sistem Informasi Geografis adalah suatu sistem yang menggabungkan antara data grafis (*spasial*) dan data teks (*atribut*) objek yang dihubungkan secara geografis di bumi (*georeference*). SIG juga memiliki kemampuan untuk menggabungkan data, mengatur data, dan melakukan analisa terhadap data untuk menghasilkan *output* yang dapat dijadikan pertimbangan dalam proses pengambilan keputusan pada masalah yang berhubungan dengan geografi.



Gambar 2.2 Komponen SIG (Sumber: Qolis, 2008)

### 2.2.2. Penelitian Terdahulu

Penelitian yang menggunakan Sistem Informasi Geografis (SIG) sebelumnya dilakukan oleh Arifin, dkk. dengan judul Sistem Informasi Geografis (SIG) Fasilitas Umum Kota Mojokerto Berbasis Web. Dalam penelitian ini membahas visualisasi

lokasi dengan kondisi yang sesungguhnya tentang sebaran fasilitas umum yang ada di Kota Mojokerto. Penelitian ini bertujuan untuk memudahkan berbagai pihak yang membutuhkan, seperti instansi pemerintahan, pelaku bisnis, wisatawan, maupun masyarakat umum untuk dimanfaatkan sesuai keperluan masing-masing.

Dalam penelitian ini peneliti menggunakan *Google Maps API* untuk merealisasikan SIG tersebut. Selain memiliki API yang dapat diintegrasikan dengan beberapa teknologi. Gambar-gambar yang muncul pada peta merupakan hasil dari database pada *Web Server Google*. Keseluruhan citra yang ada diintegrasikan ke dalam database pada *Google Server*, yang nantinya akan dapat dipanggil sesuai kebutuhan permintaan.

### 2.3. Google Maps API

*Google Maps API* merupakan aplikasi antarmuka yang dapat diakses lewat *javascript* agar *Google Maps* dapat ditampilkan pada halaman web yang akan dibangun (Sirenden *et al*, 2012:47). Sistem Informasi Geografis (SIG) dapat dibuat dengan memanfaatkan *Google Maps API*, dengan memanggil fungsi-fungsi yang dibutuhkan seperti menampilkan peta, menempatkan marker, dan lain-lain.

Pada *Google Maps API* terdapat 4 jenis pilihan model peta yang disediakan oleh *Google*, diantaranya adalah:

- a. *ROADMAP* : Untuk menampilkan peta 2 dimensi.
- b. *SATELLITE* : Untuk menampilkan citra satelit.
- c. *TERRAIN* : Untuk menunjukkan relief fisik permukaan bumi dan untuk menunjukkan seberapa tingginya suatu lokasi, contohnya gunung dan sungai.
- d. *HYBRID* : Untuk menunjukkan citra satelit yang di atasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan nama dan gudang pengepul).

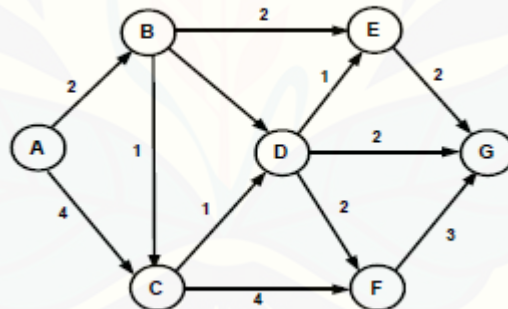
Ada 2 cara untuk mengakses data *Google Maps*, tergantung dari data yang ingin diambil dan diuraikan dari *Google Maps*, yaitu:

1. Mengakses data *Google Maps* tanpa menggunakan *API Keys*
2. Mengakses data *Google Maps* menggunakan *API Keys*

#### 2.4. Teori Graph

Macam-macam graf menurut arah dan bobotnya, graf dibagi menjadi empat bagian, yaitu:

1. Graf berarah dan berbobot : setiap edge mempunyai arah (yang ditunjukkan dengan anak panah) dan bobot. Gambar 2 adalah contoh graf berarah dan berbobot yang terdiri dari tujuh vertek yaitu bertek A, B, C, D, F, G. Vertek A mempunyai dua edge yang masing-masing menuju ke vertek B dan vertek C, vertek B mempunyai tiga edge yang masing-masing menuju vertek C, vertek D, dan vertek E. Bobot diantara keduanya pun telah diketahui.

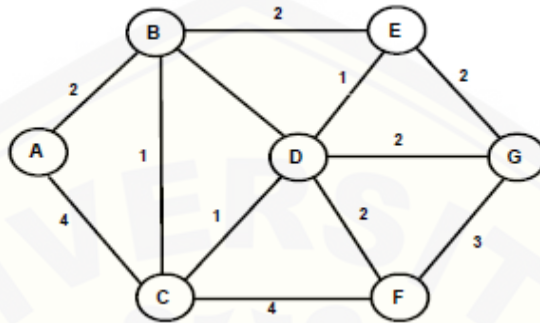


Gambar 2.3 Graf berarah dan berbobot

2. Graf tidak berarah dan berbobot : setiap edge tidak mempunyai arah tetapi mempunyai bobot. Gambar 3 adalah contoh graf tidak berarah dan berbobot. Graf terdiri dari tujuh vertek yaitu vertek A, B, C, D, E, F, G. Vertek A mempunyai dua edge yang masing-masing berhubungan dengan vertek B dan vertek C, tetapi dari masing-masing vertek tersebut tidak mempunyai arah.

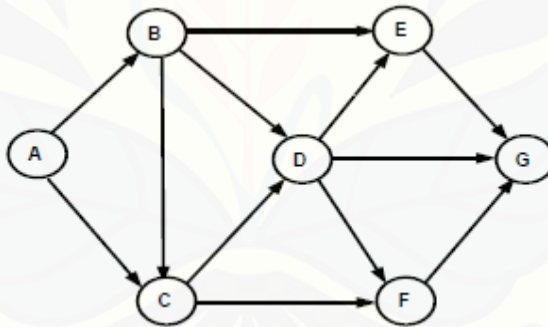


Edge yang menghubungkan vertek A dan vertek B mempunyai bobot yang teah diketahui, begitu pula dengan edge-edge yang lain.



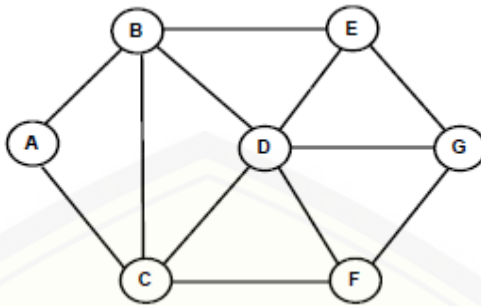
Gambar 2.4 Graf tidak berarah dan berbobot

3. Graf berarah dan tidak berbobot : setiap edge mempunyai arah tetapi tidak mempunyai bobot. Gambar 4 adalah contoh graf berarah dan tidak berbobot.



Gambar 2.5 Graf berarah dan tidak berbobot

4. Graf tidak berarah dan tidak berbobot : setiap edge tidak mempunyai arah dan tidak berbobot. Gambar 5 adalah contoh graf tidak berarah dan tidak berbobot.



Gambar 2.6 Graf tidak berarah dan tidak berbobot

## 2.5. Rute Terpendek

Permasalahan pencarian rute terpendek dapat dilakukan dengan menggunakan dua metode, yaitu metode konvensional dan metode heuristik. Metode konvensional dihitung dengan perhitungan matematis biasa, sedangkan metode heuristik dihitung dengan menggunakan sistem pendekatan.

### 2.5.1. Metode Konvensional

Metode Konvensional adalah metode yang menggunakan perhitungan matematika eksak. Metode konvensional dalam melakukan pencarian solusinya, data yang diinputkan harus lengkap, prosesnya secara algoritmik, dan outputnya harus tepat. Ada beberapa metode konvensional yang biasa digunakan untuk pencarian rute terpendek, diantaranya: algoritma *Greedy*, algoritma *Dijkstra*, algoritma *Floyd-Warshall*, dan algoritma *Bellman-Ford* (Leksono, 2009).

### 2.5.2. Metode Heuristik

Metode ini merupakan teknik yang didesain untuk memecahkan masalah dengan sedikit mengabaikan apakah solusinya bisa dibuktikan tepat, tetapi biasanya menghasilkan solusi terbaik, dalam arti optimal. Heuristik dimaksudkan untuk mendapatkan hasil secara komputasi yang lebih cepat dengan konsekuensi mengurangi presisi dan akurasi. Walaupun pada kenyataannya solusi yang dihasilkan juga mempunyai tingkat akurasi yang tinggi. Metode *Ant Colony Optimization* (ACO) merupakan salah satu metode metaheuristik yang menerapkan semut sebagai

agennya dengan update *pheromone* yang berfungsi sebagai informasi bagi koloni semut untuk dapat menemukan proses pencarian solusi yang optimal.

Algoritma metaheuristik biasanya digunakan untuk mencari solusi yang mendekati optimal pada permasalahan diskrit. Masalah diskrit tidak akan efektif jika menggunakan metode konvensional atau metode yang menggunakan perhitungan eksak. Algoritma ACO telah banyak diaplikasikan dalam berbagai bidang yang mencakup beberapa persoalan, yaitu :

- a. Traveling Salesman Problem (TSP), yaitu mencari rute terpendek dalam sebuah graph menggunakan rute Hamilton.
- b. Quadratic Assignment Problem (QAP), yaitu menugaskan sejumlah  $n$  resources untuk ditempatkan pada sejumlah  $m$  lokasi dengan meminimalisasi biaya penugasan (assignment).
- c. Vehicle Routing Problem (VRP)
- d. Pewarnaan graph, dll.

## 2.6. *Travelling Sales Problem (TSP)*

*Travelling Salesman Problem (TSP)* merupakan suatu masalah yang sangat mudah untuk dimengerti, namun memiliki kompleksitas tinggi. TSP termasuk problem optimasi diskrit, yaitu tidak ada informasi mengenai algoritma polynomial yang eksak dimana waktu komputasinya proporsional terhadap  $Nn$ .  $N$  adalah jumlah parameter yang dicari dan  $n$  adalah suatu konstanta interger, Sering juga disebut non polynomial (NP-Hard, tidak ada nilai  $n$  sehingga waktu komputasi dibatasi oleh suatu polynomial dengan pangkat  $n$ ). Bila diselesaikan secara eksak waktu komputasi yang diperlukan akan meningkat secara eksponensial seiring bertambah besarnya masalah (Leksono, 2009).

Masalah *Travelling Salesman* adalah bagaimana menentukan *Sirkuit Hamilton* biasanya diselesaikan dengan algoritma heuristik, yaitu metode yang menggunakan sistem pendekatan dalam pencarian untuk menemukan solusi yang terbaik atau



optimal. *Ant Colony Optimization* merupakan salah satu metode metaheuristik yang sering digunakan untuk menyelesaikan masalah TSP.

## 2.7. Ant Colony Optimzation (ACO)

### 2.7.1. Definisi

*Ant Colony Optimization* (ACO) diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo dalam Leksono, 2009). Semut mampu mengindera lingkungannya yang kompleks untuk mencari makanan dan kemudian kembali ke sarangnya dengan meninggalkan zat pheromone yang berisi informasi pada rute-rute yang mereka lalui.

Proses peninggalan pheromone ini dikenal sebagai stigmergy, yaitu sebuah proses memodifikasi lingkungan yang tidak hanya bertujuan untuk mengingat jalan pulang ke sarang, tetapi juga memungkinkan para semut berkomunikasi secara tidak langsung dengan koloninya.

### 2.7.2. Kelas ACO dilihat dari sudut pandang

Algoritma ACO bertujuan untuk mendapat solusi terbaik yang didasari dari pencarian rute terbaik. Algoritma ACO termasuk dalam kelas yang berbeda tergantung dari sudut pandangnya. Dilihat dari sudut pandang *artificial intelligence* (AI), algoritma ACO merupakan salah satu algoritma *swarm intelligence* yang sukses. Tujuan dari *swarm intelligence* adalah untuk mendesain *intelligence multi-agent systems* dengan mengambil inspirasi dari perilaku serangga berkoloni seperti semut, rayap, lebah, dan lain-lain. Contoh lain dari algoritma *swarm intelligence* selain ACO adalah *clustering* dan *datamining* yang terinspirasi dari karakter serangga dalam membangun sarang (Martua, 2011).

Dilihat dari sudut pandang *operation research* (OR), algoritma ACO termasuk dalam kelas metaheuristik, yaitu metode optimisasi yang dilakukan dengan memperbaiki kandidat penyelesaian secara iteratif sesuai dengan fungsi objektifnya. Metode ini mampu menghasilkan penyelesaian yang baik dalam waktu yang cepat

(*acceptable*), tetapi tidak menjamin bahwa penyelesaian yang dihasilkan merupakan penyelesaian terbaik (*optimal*). Contoh lain dari metaheuristik selain ACO adalah *evolutionary computation*, *iterated local search*, *simulated annealing*, *genetic algorithm*, dan *tabu search*.

### 2.7.3. Probabilitas Pemilihan Jalur ACO

Probabilitas semut *artificial* dalam pemilihan jalur antara sumber makanan dan sarang yaitu, ACO tersusun atas sejumlah  $m$  semut yang bekerja sama dan berkomunikasi secara tidak langsung melalui *Pheromone*. Langkah-langkah pemilihan jalur:

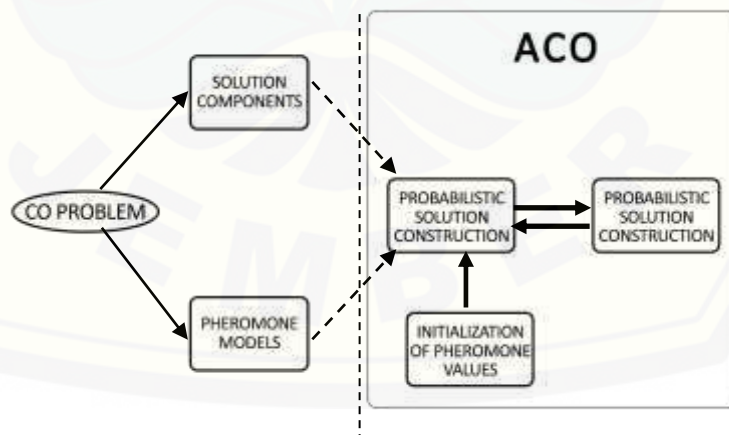
1. Setiap semut memulai tournya melalui sebuah titik yang acak.
2. Secara berulang kali, satu persatu titik yang ada dikunjungi untuk menghasilkan sebuah tour.
3. Pemilihan titik yang akan dilaluinya didasarkan pada suatu fungsi probabilitas yang disebut *state transition rule* (aturan transisi status), dengan mempertimbangkan *visibility* titik tersebut dan jumlah pheromone pada ruas yang menghubungkan titik.
4. Semut lebih suka melewati titik yang dihubungkan dengan ruas pendek dan tingkat *pheromone* tinggi
5. Setiap semut memiliki memori (*tabulist*) untuk mencegah menuju titik yang sudah dikunjungi selama tour
6. Setelah semua semut menyelesaikan tour berarti tabulist penuh
7. Aturan pembaruan pheromone global (*global pheromone updating rule*) diterapkan pada setiap semut
8. Pheromone pada setiap ruas mengalami penguapan

9. Semua semut menghitung panjang tour, lalu meninggalkan sejumlah pheromone pada setiap *edge-edge* sebanding dengan solusi yang mereka hasilkan. Ditemukan jalur terpendek
10. Semakin pendek sebuah tour yang dihasilkan oleh setiap semut, semakin besar jumlah pheromone yang ditinggalkan pada edge-edge yang mereka lalui
11. Edge yang terdapat pheromone lebih besar yang paling diminati pada tour selanjutnya
12. Semua tabu list dikosongkan kembali dan rute terpendek yang ditemukan disimpan

peran utama dari *pheromone* adalah mencegah terjadinya *stagnasi*, yaitu situasi dimana semut melakukan tour yang sama akibatnya sistem akan berhenti mencari solusi alternatif.

#### 2.7.4. Cara kerja ACO

Cara kerja ACO untuk mendapatkan solusi dapat dilihat pada gambar 2.7. dibawah ini.



Gambar 2.7 Cara Kerja ACO

Dari gambar 2.7 dapat dilihat bahwa untuk menyelesaikan masalah *Combinatorial Optimization* (CO), pertama kali definisikan komponen solusi yang akan digunakan untuk menyusun solusi. Kemudian selanjutnya, menginisialisasi nilai *pheromone* pada awal perhitungan. Nilai set *pheromone* yang dihasilkan disebut dengan *pheromone model*. Nilai awal *pheromone* untuk pemilihan solusi yang ada awalnya bernilai sama atau sebanding untuk semua probabilitas solusi. *Pheromone model* ini digunakan untuk menentukan probabilitas solusi berdasarkan komponen solusi yang ada. *Pheromone update* bertujuan untuk melakukan pencarian solusi dengan kualitas tinggi di dalam area pencarian. Proses *update* komponen solusi ini menjadi komponen penting dalam algoritma ACO.

#### 2.7.5. Algoritma ACO

Pencarian jalur terpendek dengan ACO diperlukan beberapa variabel dan langkah - langkah untuk menentukan jalur terpendek (Tyas & Prijodiprodjo, 2013), yaitu :

Langkah 1 :

- a. Inisialisasi harga parameter-parameter algoritma.

Parameter-parameter yang di inisialisasikan adalah :

1. Intensitas jejak semut antar gudang pengepul dan perubahannya ( $\tau_{ij}$ )
2. Banyak gudang pengepul ( $n$ ) termasuk koordinat ( $x,y$ ) atau jarak antar gudang pengepul ( $d_{ij}$ )
3. Gudang pengepul berangkat dan gudang pengepul tujuan
4. Tetapan siklus-semut ( $Q$ )
5. Tetapan pengendali intensitas jejak semut ( $\alpha$ ), nilai  $\alpha \geq 0$
6. Tetapan pengendali visibilitas ( $\beta$ ), nilai  $\beta \geq 0$
7. Visibilitas antar gudang pengepul =  $1/d_{ij}$  ( $\eta_{ij}$ )

8. Banyak semut ( $m$ )
  9. Tetapan penguapan jejak semut ( $\rho$ ), nilai  $\rho$  harus  $> 0$  dan  $< 1$  untuk mencegah jejak pheromone yang tak terhingga.
  10. Jumlah siklus maksimum ( $NC_{max}$ ) bersifat tetap selama algoritma dijalankan, sedangkan  $\tau_{ij}$  akan selalu diperbaharui harganya pada setiap siklus algoritma mulai dari siklus pertama ( $NC=1$ ) sampai tercapai jumlah siklus maksimum ( $NC=NC_{max}$ ) atau sampai terjadi konvergensi.
- b. Inisialisasi gudang pengepul pertama setiap semut.

Setelah inisialisasi  $\tau_{ij}$  dilakukan, kemudian  $m$  semut ditempatkan pada gudang pengepul pertama tertentu secara acak.

Langkah 2 :

Pencatatan gudang-gudang pengepul yang dilalui oleh objek semut. Setiap objek semut yang melakukan pencarian gudang pengepul tujuan akan mencatat nama gudang-gudang pengepul yang dilaluinya sehingga kita dapat mengetahui jalur perjalanan semut. UD. Mitra Samudra sebagai titik awal merupakan gudang pengepul yang pertama kali dicatat.

Langkah 3 :

Penyusunan rute kunjungan setiap semut ke setiap gudang pengepul. Sebelum melakukan pemilihan gudang pengepul tujuan dalam menelusuri gudang pengepul, dibuat suatu daftar hubungan antar gudang pengepul. Daftar hubungan gudang pengepul ini berisikan gudang-gudang pengepul yang memiliki hubungan gudang pengepul asal perjalanan. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap gudang pengepul, akan mulai melakukan perjalanan dari gudang pengepul pertama masing-masing sebagai gudang pengepul asal dan salah satu gudang pengepul lainnya sebagai gudang pengepul tujuan. Kemudian dari gudang pengepul kedua, masing-masing koloni semut akan melanjutkan perjalanan dengan memilih



salah satu dari gudang pengepul-gudang pengepul yang tidak terdapat pada daftar hubungan gudang pengepul sebagai gudang pengepul tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai gudang pengepul terakhir tidak memiliki hubungan dengan gudang pengepul lain atau hubungan yang dimiliki oleh gudang pengepul tersebut sudah tercatat dalam catatan si semut. Untuk menentukan gudang pengepul tujuan digunakan persamaan probabilitas gudang pengepul untuk dikunjungi sebagai berikut :

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in \{N - tabu_k\}} [\tau_{ik^i}]^\alpha \cdot [\eta_{ik^j}]^\beta} \text{ untuk } j \in \{N - tabu_k\} \dots 2.1$$

dengan i sebagai indeks gudang pengepul asal dan j sebagai indeks gudang pengepul tujuan.

Langkah 4 :

- a. Perhitungan panjang rute setiap semut.

Perhitungan panjang rute tertutup (length closed tour) atau  $L_k$  setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini dilakukan berdasarkan Panjang Jalur masing-masing dengan persamaan 2.2 berikut :

$$L_k = d_{tabu_k(\eta), tabu_k(1)} + \sum_{s=1}^{\eta-1} d_{tabu_k(s), tabu_k(s+1)} \dots 2.2$$

Dengan  $d_{ij}$  adalah jarak antara gudang pengepul i ke gudang pengepul j yang dihitung berdasarkan persamaan :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \dots\dots\dots 2.3$$

- b. Pencarian rute terpendek.

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

- c. Perhitungan perubahan harga intensitas jejak kaki semut antar gudang pengepul.

Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar gudang pengepul yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar gudang pengepul. Persamaan 2.4 ini adalah :

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \dots\dots\dots 2.4$$

Dengan  $\Delta\tau_{ij}$  adalah perubahan harga intensitas jejak kaki semut antar gudang pengepul setiap semut yang dihitung berdasarkan persamaan.

$$\Delta\tau_{ij}^Q = \frac{Q}{L_k} \dots\dots\dots 2.5$$

Untuk  $(i,j) \in$  gudang pengepul asal dan gudang pengepul tujuan adalah  $m$   $tabu_k$   $\Delta\tau_{ij} =$  , untuk  $(i,j)$  lainnya.

Langkah 5 :

- a. Perhitungan harga intensitas jejak kaki semut antar gudang pengepul untuk siklus selanjutnya.

Harga intensitas jejak kaki semut antar gudang pengepul pada semua lintasan antar gudang pengepul ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewati. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antar gudang pengepul untuk siklus selanjutnya dihitung dengan persamaan :

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta\tau_i \dots\dots\dots 2.6$$

- b. Atur ulang harga perubahan intensitas jejak kaki semut antar gudang pengepul.

Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar gudang pengepul perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6 :

Jika Siklus maksimum ( $N_{cmax}$ ) belum terpenuhi atau belum terjadi konvergensi, algoritma diulang lagi dari langkah 2 dengan harga parameter intensitas jejak kaki semut antar gudang pengepul yang sudah diperbaharui.

#### 2.7.6. Penelitian terdahulu

Penelitian yang menggunakan metode *Ant Colony Optimization* (ACO) yang sebelumnya dilakukan oleh Tyas dan Prijodiprodjo dengan judul Aplikasi Pencarian Rute Terbaik dengan Metode *Ant Colony Optimization* (ACO) yang membahas pencarian rute terbaik atau terpendek Patroli Jalan Raya (PJR) dalam menyelenggarakan kegiatan-kegiatan pengawasan, pengendalian lalu lintas, keamanan dan ketertiban umum di jalan, serta penindakan pelanggaran lalu lintas, tindakan pertama pada tempat kejadian perkara (TPTKP), serta setiap bentuk ancaman dan gangguan di jalan.

Algoritma *Ant Colony Optimization* (ACO) yang terinspirasi dari perilaku koloni semut yang dikenal sebagai system semut. Secara alamiah koloni semut mampu menentukan rute terpendek dalam perjalanan dari sarang menuju sumber makanan dan kembali lagi, pada saat berjalan semut meninggalkan informasi yang disebut *Pheromone* (jejak kaki semut), ditempat yang dilaluinya dan menandai rute tersebut. *Pheromone* digunakan sebagai komunikasi antar semut pada saat membangun rute [1],[2],[3].



### BAB 3. METODOLOGI PENELITIAN

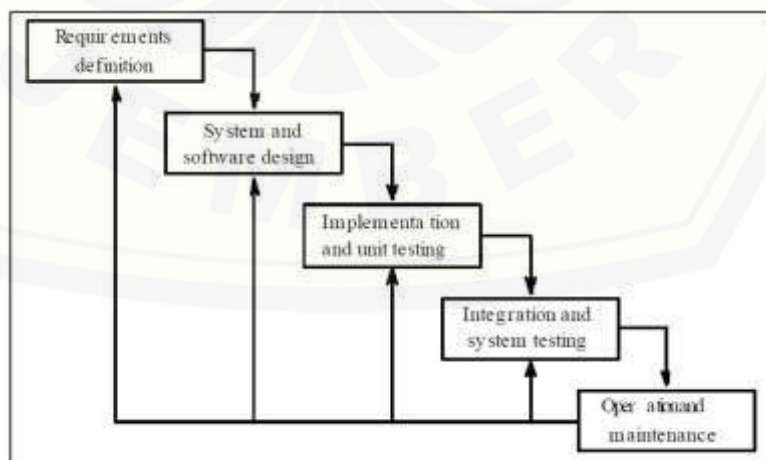
Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem menggunakan metode pengembangan.

#### 3.1. Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini adalah penelitian pengembangan, di mana penelitian ini membuat dan mengembangkan suatu sistem untuk memetakan dan menggambarkan sebaran pengepul secara grafis di Kabupaten Situbondo dan menentukan rute terpendek dalam penjemputan ikan di gudang-gudang pengepul sehingga dapat membantu UD. Mitra Samudra dalam menjalankan bisnisnya dengan menggunakan metode *Ant Colony Optimization* (ACO) dan penelitian ini bukan dimaksudkan untuk menemukan teori baru atau menguji kebenaran dari suatu teori atau metode dalam penelitian.

#### 3.2. Tahapan Penelitian

Metode yang digunakan dalam penelitian ini yakni *System Development Life Cycle* (SDLC) dengan metode *Waterfall*. Menurut Pressman (2010, p39) model *waterfall* adalah model klasik yang bersifat matematis, berurutan dalam membangun *software*. Fase-fase dalam model *waterfall* ditunjukkan pada gambar 3.1.



Gambar 3.1 Fase – fase model *waterfall*

### 3.2.1. Tahapan Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem. Metode yang digunakan adalah sebagai berikut:

#### a. Metode Pengumpulan Data

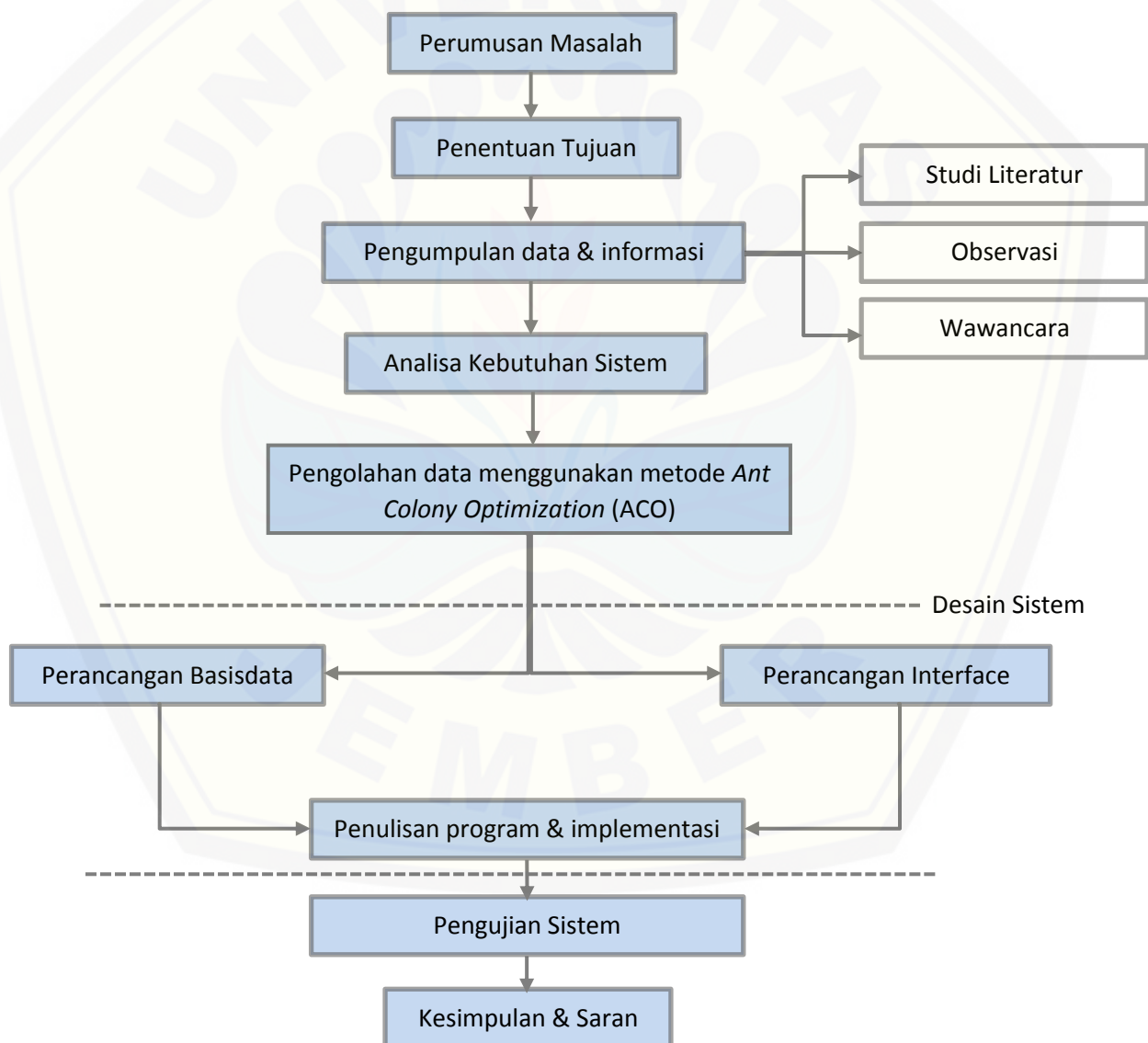
Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

1. Observasi yaitu metode pengumpulan data melalui pengamatan dan peninjauan secara langsung di lapangan atau lokasi penelitian. Dalam hal ini peneliti melakukan pengamatan dan peninjauan langsung ke UD. Mitra Samudra dan pengepul-pengepul yang tersebar di Kabupaten Situbondo.
2. Wawancara adalah salah satu cara untuk mendapatkan informasi dengan mengajukan pertanyaan-pertanyaan kepada narasumber. Melakukan wawancara kepada pemilik UD. Mitra Samudra dan pengepul-pengepul di Kabupaten Situbondo.
3. Studi literatur pada penelitian-penelitian terdahulu di berbagai buku, jurnal, skripsi dan *e-book*. Studi literatur sangat dibutuhkan guna menunjang pemahaman dan pengetahuan penulis mengenai konsep, teori dan metode yang diperlukan dalam proses pengerjaan tugas akhir ini.

#### b. Metode Analisis Data

Tahap analisis dimulai dengan menelaah data secara keseluruhan yang telah dikumpulkan dari tahap pengumpulan data baik itu studi literature, observasi dan wawancara. Langkah selanjutnya adalah menganalisa data dengan menggunakan ACO. Data input yang digunakan untuk menentukan jalur

terpendek adalah data lokasi sebaran gudang pengepul ikan dan serta data jalan di Kabupaten Situbondo. Data input untuk ACO ada dua yaitu titik awal dan titik tujuan. Output yang dihasilkan dari proses algoritma berupa jalur terpendek dari titik awal menuju titik tujuan, serta info mengenai waktu. Hal ini sangat diperlukan mengingat *software* harus dapat berinteraksi dengan elemen-elemen lain seperti *hardware*, *database*, dsb. Tahap ini sering disebut *Project Definition*. Dapat dilihat pada gambar 3.2.



Gambar 3.2 Skema *Project Definition*

### 3.2.2. Tahapan Desain Sistem

Tahap ini meliputi tahap desain sistem dengan pembuatan diagram menggunakan tools *UML Visual Paradigm*. Diagram-diagram tersebut akan digunakan sebagai acuan pembuatan sistem pada tahap implementasi kebutuhan sistem. Konsep program menggunakan *Object-Oriented Programming (OOP)*. Pada bagian desain, pengerjaan yang dilakukan diantaranya adalah:

a. *Business Process*

*Business Process* merupakan gambaran dari masuknya data serta data yang dihasilkan dari proses yang dijalankan sistem. Data yang dibutuhkan oleh sistem (*input*), keluaran data yang dihasilkan (*output*), media dari sistem (*uses*), dan tujuan dari pembuatan sistem (*goal*).

b. *Use Case Diagram*

*Use case* adalah model yang menggambarkan apa saja aktifitas yang dapat dilakukan suatu sistem dari sudut pandang pengamatan luar. *Use Case Diagram* lebih menekankan pada “apa” yang dapat dilakukan sistem, bukan “bagaimana” sistem bekerja. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Diagram *use case* dapat sangat membantu dalam penyusunan *requirements* sebuah sistem.

c. *Scenario*

Diagram skenario berfungsi untuk menjelaskan alur sistem dari fitur yang ada di *job specification* dan *job description* yang ada pada diagram *use case*. *Scenario* menjelaskan alur sistem dan keadaan yang terjadi pada *event* tertentu.

d. *Activity Diagram*

*Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aktivitas berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada eksekusi.

e. *Sequence Diagram*

*Sequence Diagram* menggambarkan aliran logika interaksi antar objek yang mengindikasikan komunikasi antar objek di dalam sistem yang disusun pada suatu urutan (*timeline*).

f. *Class Diagram*

*Class Diagram* menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*, sehingga memudahkan dalam proses pengkodean.

g. *Entity Relationship Diagram (ERD)*

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan data yang mempunyai hubungan antar relasi.

### 3.2.3. Tahapan Implementasi Sistem

Setelah desain sistem telah selesai dilakukan maka selanjutnya pada tahap ini akan dilakukan pembuatan sistem. Pembuatan sistem meliputi penulisan kode program, dan pembuatan basis data. Penulisan kode program dilakukan menggunakan tools *sublime text 2* sebagai editor dengan bahasa pemrograman *PHP* sedangkan untuk manajemen basis data yang digunakan dalam membangun sistem yaitu *DBMS MySQL*.

### 3.2.4. Tahapan Pengujian Sistem

Pengujian sistem berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem yang dibangun.

a. *White Box Testing*

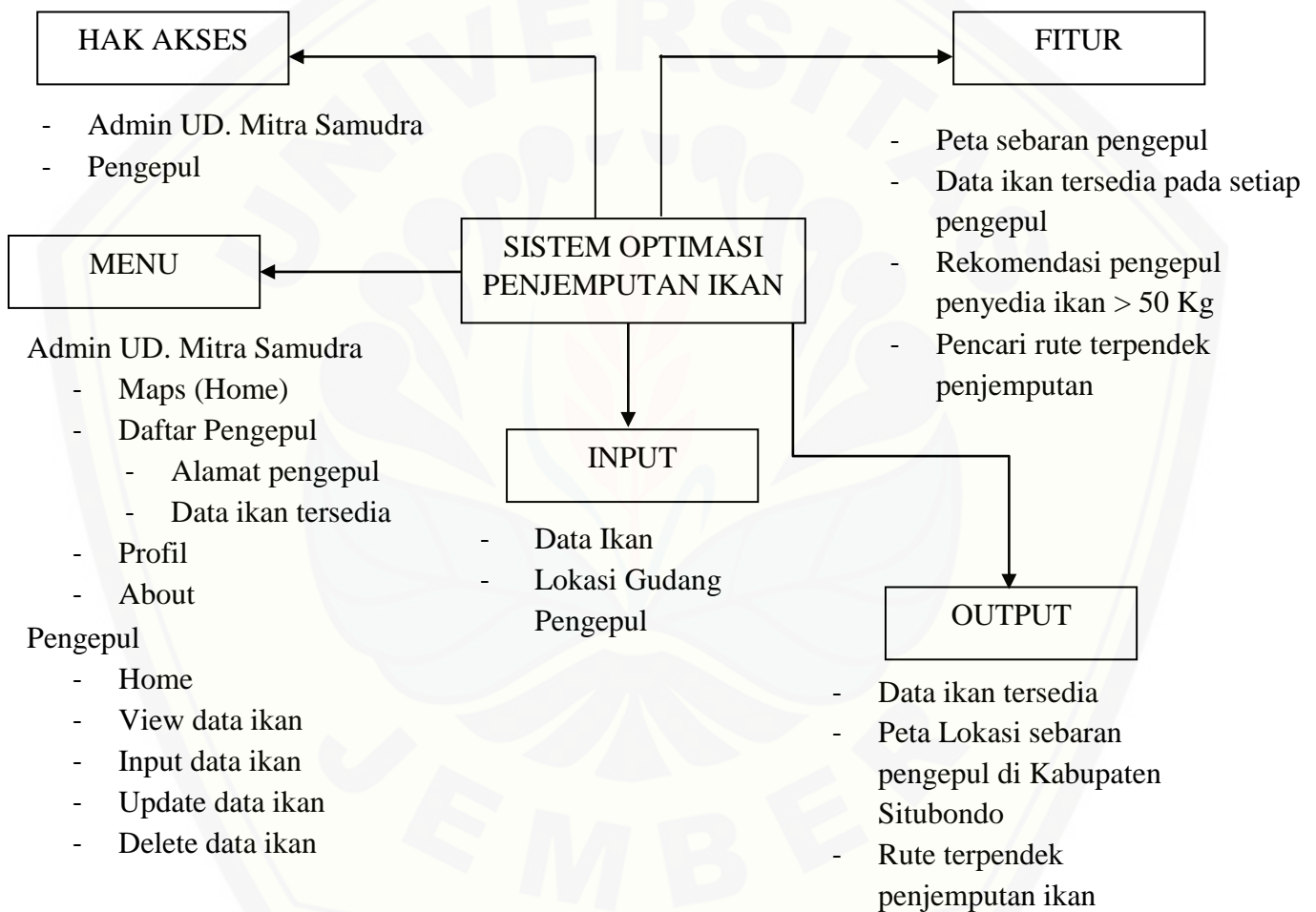
*White box testing* merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Pengujian ini dilakukan oleh (*developer*) pembuat program. Jika ada modul yang menghasilkan *output* yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan di cek dan diperbaiki, kemudian di *compile* ulang. (Pressman, 2010)

b. *Black Box Testing*



Berbeda dengan white box testing, *Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur *internal* atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Pressman, 2010).

### 3.3. Gambaran Umum Sistem



Gambar 3.3 Gambaran umum sistem



## BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini akan menjelaskan mengenai proses perancangan untuk membuat Sistem Optimasi Rute Terpendek Pengangkutan Sampah di Surabaya Menggunakan Algoritma Ant Colony Optimization (ACO). Proses perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, dilanjutkan dengan pembuatan *usecase diagram*, skenario, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relation diagram* (ERD).

### 4.1. Deskripsi Umum Sistem

Deskripsi umum sistem informasi geografi untuk pemetaan sebaran jempuran ikan dengan metode *Ant Colony Optimization* (ACO) di UD. Mitra Samudra yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (*statement of purpose*) sistem dan fungsi sistem.

#### 4.1.1. SOP (*statement of purpose*)

Sistem informasi geografi untuk pemetaan sebaran jempuran ikan dengan metode *Ant Colony Optimization* (ACO) di UD. Mitra Samudra merupakan sistem yang mampu mengelola stok ikan pada jaringan bisnis ikan yang berpusat di UD. Mitra Samudra dan mampu menentukan rute terpendek penjemputan ikan berdasarkan stok yang tersedia pada setiap pengepul yang menjadi member dalam sistem ini berdasarkan kriteria jarak antar node yang telah ditentukan dengan menggabungkan dengan perhitungan menggunakan metode *Ant Colony Optimization* (ACO). Kriteria tersebut kemudian dihitung untuk menghasilkan informasi jarak antar node dengan meninggalkan feromon pada setiap *edge* yang dilewati oleh semut. Feromon pada *edge* yang diketahui bukan *edge* dengan jarak yang terpendek tersebut kemudian mengalami proses penguapan karena jarang dilewati. Sehingga terpilih lah *edge* yang memiliki jarak terpendek dengan jumlah feromon terbanyak. Selanjutnya

hasil perhitungan rute terpendek akan dikirim kepada *user* melalui jalur yang ditampilkan secara grafis pada peta yang terdapat dalam sistem.

#### **4.2. Pengumpulan Data**

Pengumpulan data dilakukan dengan beberapa cara diantaranya adalah dengan digitasi, *download* data spasial dari situs open source, wawancara dan observasi atau tracking langsung ke lapangan. Data-data yang perlu dikumpulkan untuk sistem optimasi yang akan dibangun adalah data batas Kabupaten Situbondo, data jalan dan data sebaran pengepul di Kabupaten Situbondo.

#### **4.3. Analisis Data**

Analisis data diperlukan untuk mengetahui data yang digunakan pada sistem dapat diimplementasikan dengan benar. Pada tahap analisis data terdapat dua tahap yaitu pengolahan data dan analisis menggunakan ACO.

#### **4.4. Perancangan Sistem**

Perancangan sistem yang akan digunakan adalah perancangan model *waterfall*. Pada model *waterfall* terdapat lima tahapan perancangan yaitu analisa kebutuhan, desain sistem, penulisan kode program, pengujian program dan *maintenance* program.

##### **4.4.1. Analisis Kebutuhan**

Tahap analisa kebutuhan adalah tahap pengumpulan kebutuhan yang harus dimiliki oleh sistem untuk memenuhi fungsi yang diinginkan oleh pengguna dan kemudian dijabarkan dalam sebuah deskripsi yang jelas dan lengkap. Analisis kebutuhan yang dilakukan adalah menganalisis kebutuhan dengan mengelompokkan ke dalam kebutuhan fungsional dan non-fungsional.

##### **a. Kebutuhan Fungsional**

Kebutuhan fungsional merupakan fitur dasar yang harus dimiliki oleh perangkat lunak untuk menerima inputan data kemudian diproses hingga menghasilkan

suatu output. Sistem yang dirancang ini memiliki kebutuhan fungsional sebagai berikut:

1. Sistem memiliki dua hak akses *user* yaitu, admin UD. Mitra Samudra dan member.
  2. Sistem dapat membedakan hak akses *user* melalui login.
  3. Sistem dapat mengelola data member seperti tambah dan edit.
  4. Sistem dapat mengelola data ikan seperti tambah dan edit.
  5. Sistem dapat menampilkan peta sebaran pengepul di Kabupaten Situbondo.
  6. Sistem dapat menampilkan daftar rekomendasi jempukan ikan.
  7. Sistem dapat menemukan rute terpendek penjemputan ikan dari titik awal ke titik tujuan.
- b. Kebutuhan non Fungsional

Kebutuhan non-fungsional merupakan fungsi layanan pada sistem yang tidak secara langsung terkait pada fungsi sistem. Sistem optimasi ini memiliki kebutuhan non-fungsional sebagai berikut:

1. Sistem berbasis web.
2. Sistem dapat diakses secara bersamaan oleh *user* yang berbeda.
3. Sistem memiliki *User Interface* yang menarik dan *User Friendly*.

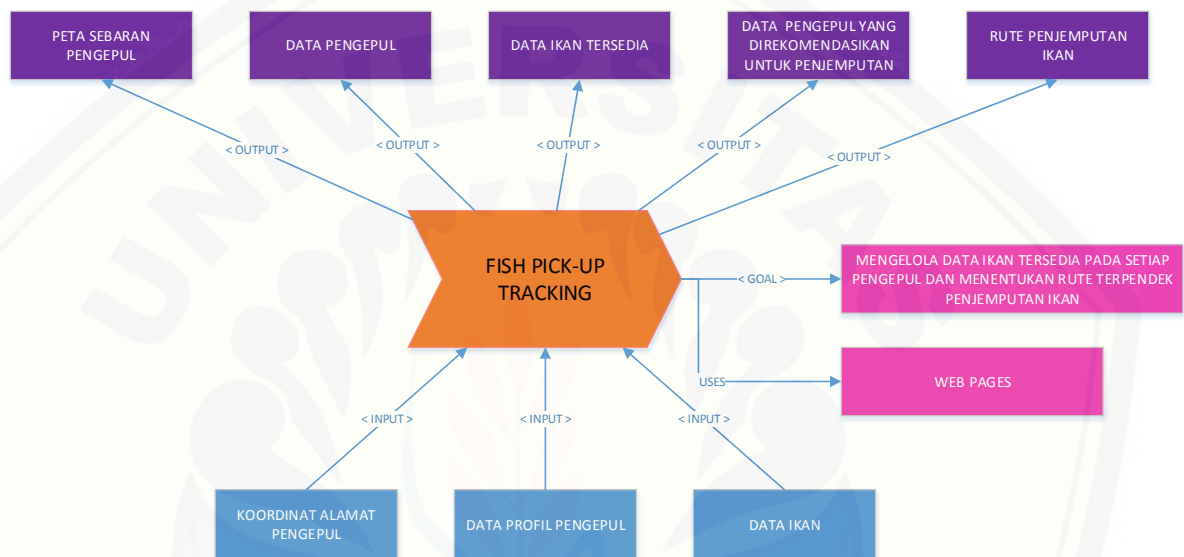
#### 4.4.2. Desain Sistem

Desain sistem yang digunakan adalah bahasa pemodelan *Unified Modeling Language* (UML). Terdapat beberapa jenis diagram dalam UML yang dapat digunakan untuk menggambarkan desain sistem sehingga dapat mendukung pembuatan sistem. Diagram-diagram tersebut adalah sebagai berikut :

##### a. *Bussiness Process*

*Bussiness Process* merupakan sebuah model diagram yang menggambarkan proses keseluruhan pada sistem yang terdiri dari *trigger*, *uses*, *input*, *supply*, *output*, dan *goal*. Input pada sistem ini adalah data member atau pengepul beserta titik

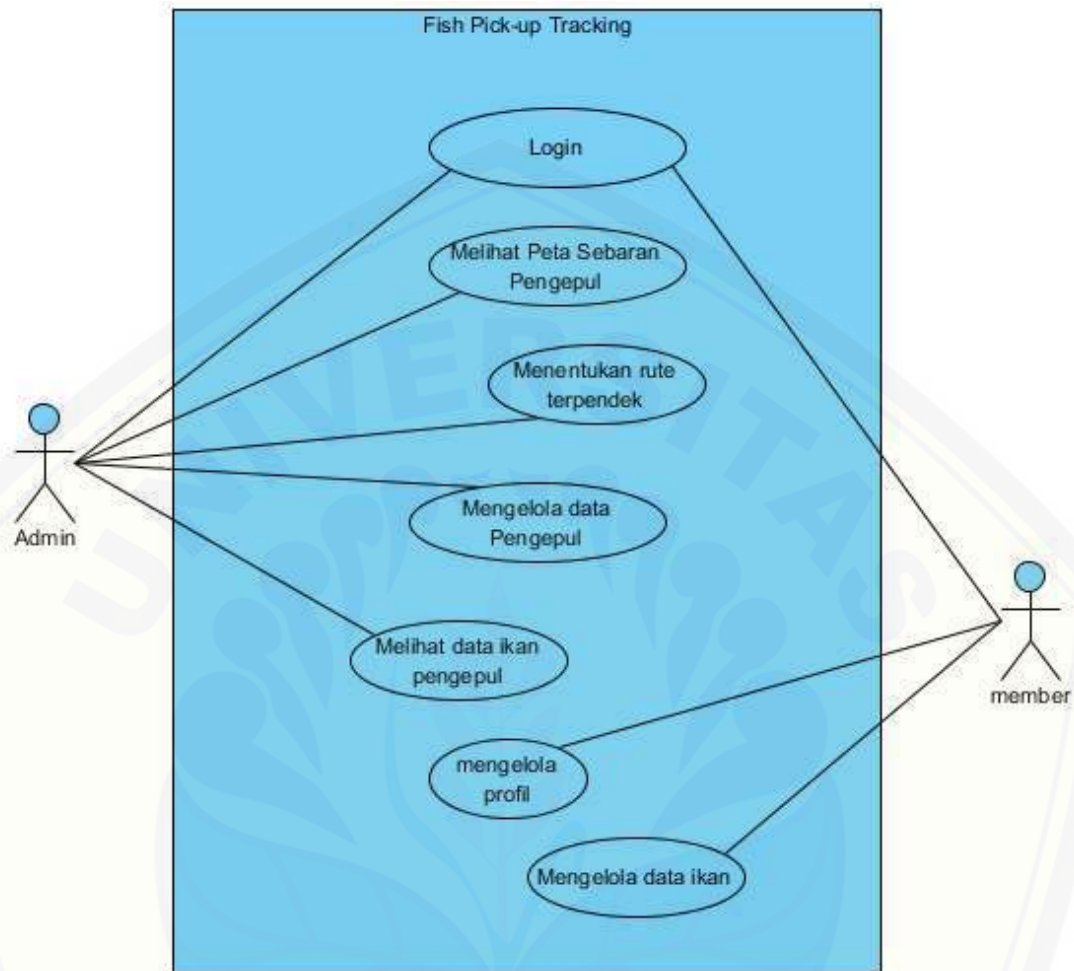
koordinatnya dan data stok ikan pengepul. Keluaran dari sistem berupa peta sebaran pengepul dan petunjuk jalan penjemputan ikan menuju setiap member atau pengepul yang direkomendasikan oleh sistem. Tujuannya adalah untuk mengetahui rute terpendek menuju member atau pengepul. Sistem ini akan dibangun berbasis web. Gambaran *Business Process* sistem dapat dilihat pada Gambar 4.1.



Gambar 4.1 *Business Process Fiss Pick-Up Tracking*

b. *Usecase Diagram*

*Usecase Diagram* merupakan identifikasi fungsi atau fitur yang ada pada sistem digambarkan berinteraksi dengan *user* sebagai akses fitur yang bisa digunakan oleh *user* tersebut. Fitur-fitur pada sistem ini terdapat 7 fitur yang digambarkan dengan elips dan terdapat 2 tipe *user*. *Usecase diagram* pada sistem ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Usecase Diagram Fish Pick-Up Tracking

#### 1. Definisi Aktor

Definisi aktor adalah user yang menggunakan sistem atau dapat berinteraksi dengan sistem dan memiliki hak akses dengan masing-masing fitur yang berbeda yang dapat digunakan oleh aktor tersebut. Penjelasan definisi aktor dapat dilihat pada Tabel 4.1.



Tabel 4. 1 Definisi Aktor

Aktor	Deskripsi
<b>Admin UD. Mitra Samudra</b>	Aktor yang memiliki hak akses penuh terhadap sistem. Aktor ini dapat menggunakan hampir semua fitur sistem meliputi <i>login</i> , melihat peta sebaran pengepul, menentukan rute terpendek, melihat daftar pengepul, mengelola data pengepul, melihat data ikan pengepul, kecuali mengelola data ikan yaitu, input dan edit.
<b>Pengepul</b>	Aktor yang memiliki akses sangat terbatas pada sistem. Aktor ini hanya dapat melakukan login dan mengelola data ikan.

## 2. Deskripsi *Usecase*

Definis *Usecase* merupakan penjelasan dari setiap *usecase diagram* yang berisi fitur-fitur dari sistem. Penjelasan definisi *usecase* dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Deskripsi *Usecase*

Usecase	Deskripsi
Login	<i>Usecase</i> yang menentukan hak akses <i>user</i> ketika memasuki dan sebagai keamanan sistem.
Melihat peta sebaran pengepul	<i>Usecase</i> yang menampilkan peta sebaran pengepul di kabupaten Situbondo sabagai member sistem.
Menentukan rute terpendek	<i>Usecase</i> yang memiliki fungsi untuk mencari rute terpendek penjemputan ikan



	berdasarkan pengepul yang dipilih sebagai hasil dari <i>Ant Colony Optimization</i> (ACO).
Melihat daftar pengepul	<i>Usecase</i> yang menampilkan daftar data pengepul sebagai member sistem
Mengelola data pengepul	<i>Usecase</i> yang menggambarkan proses pengelolaan data pengepul, yaitu input dan edit.
Melihat data ikan pengepul	<i>Usecase</i> yang menampilkan data ikan pada setiap pengepul serta detail ikan yang ada di stok pengepul
Mengelola data ikan	<i>Usecase</i> yang menggambarkan proses pengelolaan data ikan pengepul, yaitu input dan edit.

c. Skenario

Skenario merupakan penjelasan alur dari sistem yang akan dibuat berdasarkan masing-masing fitur. Setiap *usecase* pada Gambar 4.2 akan dibuat skenario normal dan skenario alternatif.

1. Skenario Login

Skenario Login merupakan langkah-langkah aksi aktor dan reaksi sistem dalam proses login berdasarkan skenario normal dan skenario alternatif. Pada proses login dapat diketahui langkah-langkah *user* dalam mengakses sistem dari halaman beranda web hingga masuk ke dashboard sistem. Hak akses login pada sistem yang dibangun memiliki dua hak akses, yaitu Admin UD. Mitra Samudra dan member. Adapun proses alur login dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Skenario Login Admin

<b>ID</b>	USC 01
<b>Nama Use Case</b>	<b>Login</b>
<b>Aktor</b>	Admin UD. Mitra Samudra
<b>Entry Condition</b>	Aktor melakukan Login
<b>Exit Condition</b>	Aktor telah memasuki sistem
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Membuka halaman web	
	2. Menampilkan halaman beranda web dengan menu <ul style="list-style-type: none"> <li>- Beranda</li> <li>- Fitur</li> <li>- Kontak</li> <li>- Log In</li> </ul>
3. Memilih menu Log In	
	4. Menampilkan form Login
5. Memasukkan username dan password	
6. Mengklik tombol login	
	7. Eksekusi validasi username dan password di dalam database
	8. Menampilkan flashdata “berhasil login”
	9. Menampilkan halaman Dashboard Admin UD. Mitra Samudra dengan menu. <ul style="list-style-type: none"> <li>- Dashboard</li> <li>- Rute Terpendek</li> <li>- Daftar Pengepul</li> </ul>

	- Data Ikan Pengepul
Alternatif (Username dan Password salah)	
	8. Menampilkan flashdata login gagal “username atau password salah”

Proses skenario login member dapat dilihat pada Lampiran A Tabel A.1.

2. Skenario Melihat peta sebaran pengepul

Skenario Melihat peta sebaran pengepul menjelaskan tentang alur aktor untuk melihat lokasi sebaran pengepul yang terdaftar pada sistem sebagai member di Kabupaten Situbondo. User melakukan login dan kemudian sistem akan menampilkan peta sebaran pengepul di halaman Dashboard. Proses lengkapnya dapat dilihat pada Lampiran A Tabel A.2.

3. Skenario Menentukan rute terpendek

Skenario Menentukan rute terpendek menjelaskan tentang alur aktor untuk menentukan rute terpendek jemputan ikan pada setiap pengepul yang direkomendasikan oleh sistem. Pada Skenario ini sistem akan menampilkan peta sebaran pengepul yang direkomendasikan untuk dilakukan jemputan dan tabel daftar rekomendasi jemputan terkini. kemudian aktor akan memilih dengan mencentang pengepul yang akan dilakukan jemputan, lalu klik tombol “TENTUKAN RUTE JEMPUTAN”. Sistem akan bereaksi dengan menampilkan rute terpendek jemputan ikan pada peta. Proses lengkapnya dapat dilihat pada Lampiran A Tabel A.3.

4. Skenario Mengelola data pengepul

Skenario Mengelola data pengepul menjelaskan tentang alur aktor untuk menambahkan data member baru atau mengubah data member. Setelah sistem menampilkan daftar pengepul, maka aktor mengklik tombol tambah data untuk menambahkan data member baru atau edit untuk mengubah data member. Kemudian sistem akan menampilkan form data baru atau form edit. Aktor akan mengisi data

pada form, kemudian klik simpan. Langkah selanjutnya sistem akan memberikan reaksi dengan pesan “data berhasil ditambahkan” atau “terjadi kesalahan dalam pengisian form”. Proses lengkapnya dapat dilihat pada Lampiran A Tabel A.4.

5. Skenario Melihat data ikan pengepul

Skenario Melihat data ikan pengepul menjelaskan tentang alur untuk melihat data ikan yang tersedia. Ketika aktor memilih menu “Data Ikan Pengepul” pada hak akses Admin, sistem akan menampilkan data ikan yang tersedia, untuk melihat detail data ikan aktor bisa mengklik tombol “Detail”. Kemudian sistem akan menampilkan data detail ikan. Proses lengkapnya dapat dilihat pada Lampiran A Tabel A.5.

6. Skenario Mengelola profil

Skenario mengelola profil menjelaskan alur aktor untuk mengelola profil. Pada skenario ini aktor dapat melihat data profil pribadi atau mengubah passwordnya sendiri dengan klik tombol edit password. Kemudian sistem akan menampilkan form ganti password. Proses lengkapnya dapat dilihat pada Lampiran A Tabel A.6.

7. Skenario Mengelola data ikan

Skenario Mengelola data ikan menjelaskan tentang alur aktor untuk menambahkan data member baru atau mengubah data stok ikan. Setelah sistem menampilkan data stok ikan, maka aktor mengklik tombol tambah data untuk menambahkan data member baru atau edit untuk mengubah data stok ikan. Kemudian sistem akan menampilkan form data baru atau form edit. Aktor akan mengisi data pada form, kemudian klik simpan. Langkah selanjutnya sistem akan memberikan reaksi dengan pesan “data berhasil ditambahkan” atau “terjadi kesalahan dalam pengisian form”. Proses lengkapnya dapat dilihat pada Lampiran A Tabel A.7.

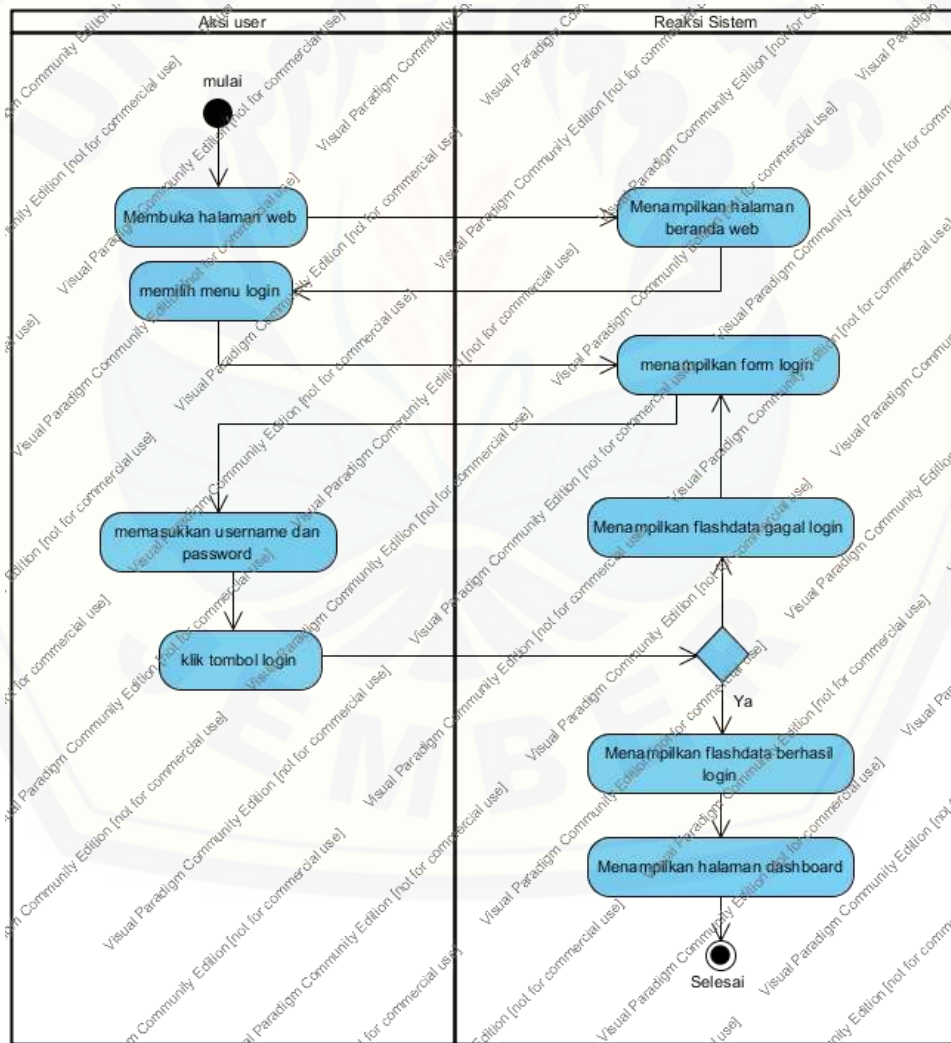
d. *Activity Diagram*

*Activity Diagram* merupakan penggambaran alur kerja aktifitas pada setiap fungsi fitur yang ada pada sistem. Pada sistem ini terdapat beberapa *usecase* yang

akan digambarkan ke dalam *activity diagram*. Masing-masing diagram akan digambarkan bagaimana suatu *activity* dimulai kemudian terdapat *decision* dan bagaimana *activity* tersebut berakhir.

### 1. Activity Diagram Login

*Activity diagram* Login menjelaskan alur kerja sistem pada proses login admin UD. Mitra Samudra atau member hingga masuk ke halaman Dashboard. Menjelaskan bagaimana sistem melakukan fungsi autentikasi hak akses aktor dalam menggunakan sistem. Seperti pada Gambar 4.3.



Gambar 4.3 Activity Diagram Login Admin



Alur diagram login member dapat dilihat pada Lampiran B Gambar B.1.

2. *Activity Diagram* Admin – Melihat peta sebaran pengepul

*Activity diagram* yang menjelaskan alur kerja sistem pada proses Admin UD. Mitra Samudra melihat peta sebaran pengepul yang terdaftar dalam sistem sebagai member di seluruh Kabupaten Situbondo. Alur diagram dapat dilihat pada Lampiran B Gambar B.2.

3. *Activity Diagram* Admin – Menentukan rute terpendek

*Activity diagram* yang menjelaskan alur kerja sistem pada proses Admin UD. Mitra Samudra menentukan rute terpendek penjemputan ikan pada pengepul yang direkomendasikan oleh sistem dan dipilih oleh admin. Alur diagram dapat dilihat pada Lampiran B Gambar B.3.

4. *Activity Diagram* Admin – Mengelola data pengepul

*Activity diagram* yang menjelaskan alur kerja sistem pada proses Admin UD. Mitra Samudra mengelola data pengepul baru atau member terdaftar seperti tambah data dan edit. Alur diagram dapat dilihat pada Lampiran B Gambar B.4.

5. *Activity Diagram* Admin – Melihat data ikan pengepul

*Activity diagram* yang menjelaskan alur kerja sistem pada proses Admin UD. Mitra Samudra melihat data stok ikan yang tersedia pada setiap pengepul. Alur diagram dapat dilihat pada Lampiran B Gambar B.5.

6. *Activity Diagram* Member – Mengelola profil

*Activity Diagram* yang menjelaskan alur kerja sistem pada proses member mengelola profil, yaitu melihat profil pribadi atau mengubah password. Alur diagram dapat dilihat pada Lampiran B Gambar B.6.

7. *Activity Diagram* Member – Mengelola data ikan



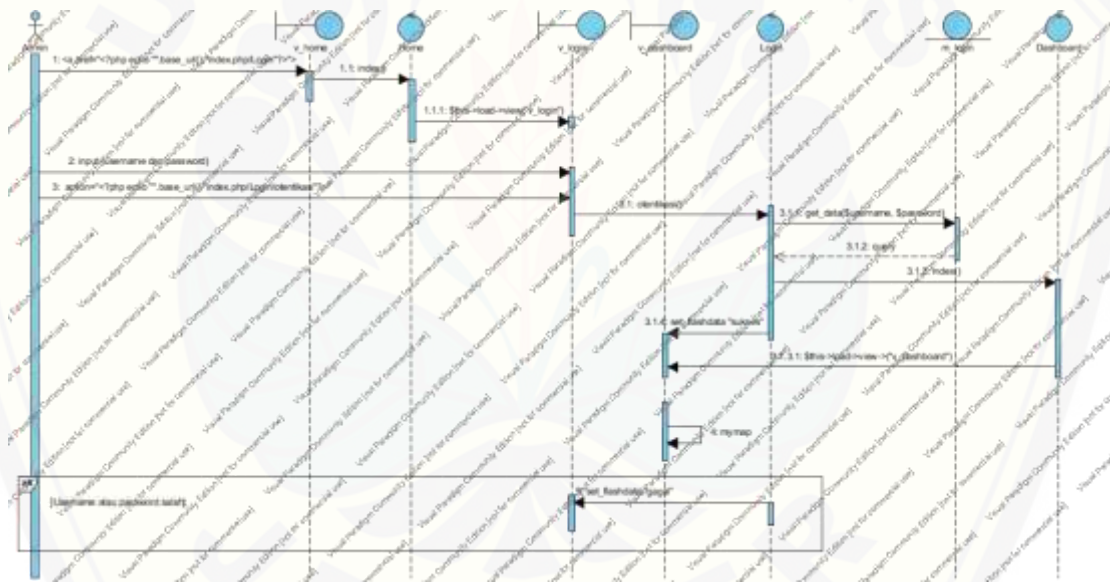
*Activity diagram* yang menjelaskan alur kerja sistem pada proses member mengelola data stok ikan seperti input data baru atau edit data stok. Alur diagram dapat dilihat pada Lampiran B Gambar B.7.

e. *Sequence Diagram*

*Sequence Diagram* data *user* merupakan dokumentasi yang menampilkan interaksi – interaksi objek di dalam sistem atau pemodelan aliran logika sistem pada fitur data *user*. Penggambaran *sequence* adalah sebagai berikut :

1. *Sequence Admin – Login*

*Sequence* ini menjelaskan *class* dan *method* yang digunakan dalam proses login. Alur diagram dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Sequence Diagram* Login Admin

Alur diagram login member dapat dilihat pada Lampiran C Gambar C.1.

2. *Sequence Admin – Melihat data ikan pengepul*

*Sequence* ini menjelaskan *class* dan *method* yang digunakan dalam proses aktor untuk melihat data ikan yang tersedia pada setiap pengepul yang terdaftar sebagai member. Alur diagram dapat dilihat pada Lampiran C Gambar C.2.

3. *Sequence Admin – Menentukan rute terpendek*

*Sequence* ini menjelaskan respon *class* dan *method* yang digunakan dalam proses aktor melakukan aksi menentukan rute terpendek. Alur diagram dapat dilihat pada Lampiran C Gambar C.3.

4. *Sequence Admin – Mengelola data pengepul*

*Sequence* ini menjelaskan respon *class* dan *method* yang digunakan dalam proses aktor melakukan aksi tambah data member baru atau edit data. Alur diagram dapat dilihat pada Lampiran C Gambar C.4.

5. *Sequence Member – Mengelola profil*

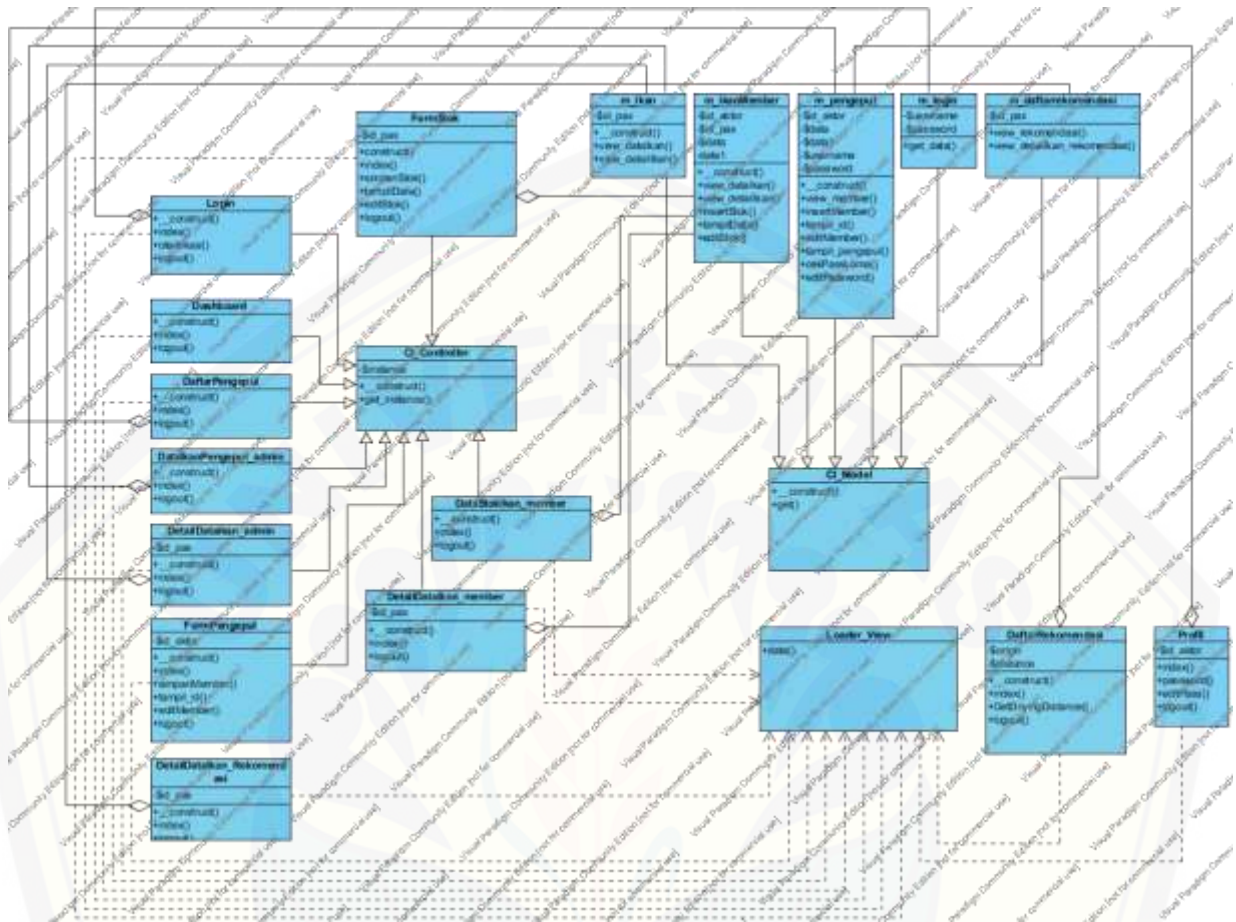
*Sequence* ini menjelaskan respon *class* dan *method* yang digunakan dalam proses aktor melakukan aksi mengelola profil, yaitu melihat profil pribadi *user* atau mengganti password. Alur diagram dapat dilihat pada Lampiran C Gambar C.5.

6. *Sequence Member – Mengelola data ikan*

*Sequence* ini menjelaskan respon *class* dan *method* yang digunakan dalam proses aktor melakukan aksi mengelola data ikan, yaitu menambahkan data baru atau edit data. Alur diagram dapat dilihat pada Lampiran C Gambar C.6.

f. *Class Diagram*

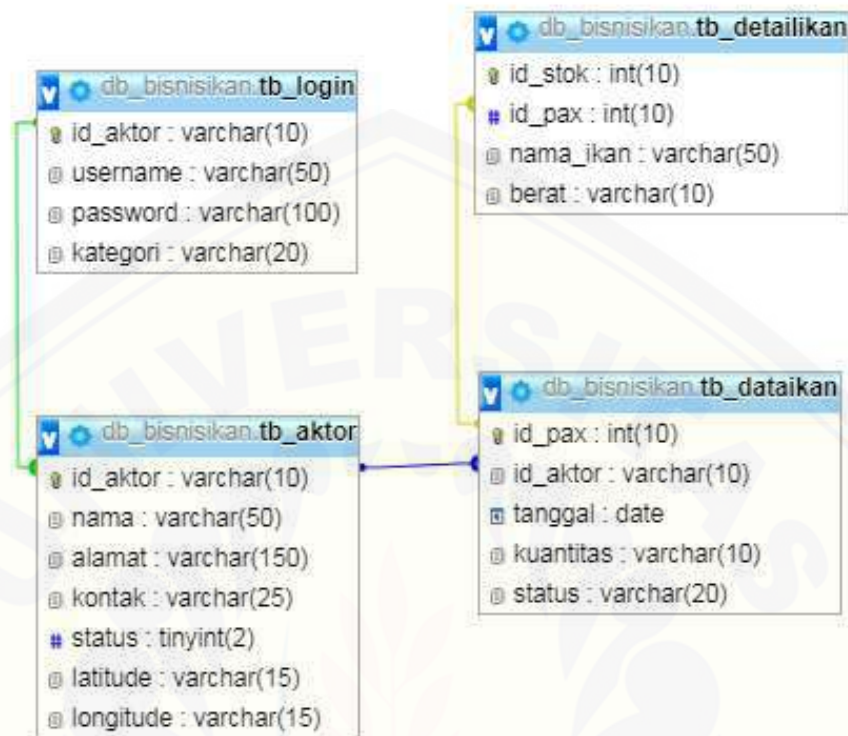
*Class* diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence diagram* yang telah dirancang, *class diagram* sistem informasi geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO) di UD. Mitra Samudra terdiri dari 5 *class model*, 12 *class controller*, dan 14 *class view*. *Class Diagram* lebih lengkapnya seperti pada gambar 4.5.



Gambar 4.5 Class Diagram

g. *Entity Relationship Diagram (ERD)*

*Entity Relationship Diagram (ERD)* menggambarkan komponen, struktur, dan relasi *database* yang digunakan dalam pembuatan sistem. *Entity relationship diagram (ERD)* yang dirancang pada sistem informasi geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization (ACO)* di UD. Mitra Samudra terdiri dari 5 entitas, 4 entitas yang berelasi dan 1 entitas yang berdiri sendiri sebagai untuk mengisi data parameter ACO. ERD sistem seperti diunjukkan pada gambar 4.6.



Gambar 4.6 Entity Relationship Diagram

#### 4.5. Pengkodean

Pengkodean atau *coding* merupakan proses menulis kode dalam membangun program berdasarkan desain sistem yang telah dibuat. Dalam pengkodean sistem informasi geografis untuk pemetaan sebaran penjemputan ikan, penulis menggunakan bahasa pemrograman *Page Hypertext Pre-processor* (PHP), *Cascading Style Sheet* (CSS), dan *Javascript*. MySQL sebagai sistem manajemen database. Pengkodean program menggunakan konsep *Object Oriented Programming* (OOP) dan menggunakan *framework CodeIgniter*. Berikut potongan kode program untuk fitur rekomendasi rute terpendek penjemputan ikan yang terdapat pada *class basefooter\_admin.php* pada *package view* yang ditampilkan pada tabel 4.4.



Tabel 4. 4 Potongan kode program pada fitur rute terpendek

Kode
<pre> &lt;?php if (isset(\$dataRekomendasi) &amp;&amp; count(\$dataRekomendasi) &gt; 1) { ?&gt; &lt;script type="text/javascript"&gt; function emaps() { console.log(tsp); var contentString = '&lt;div id="content"&gt;' + '&lt;div id="siteNotice"&gt;&lt;/div&gt;' +     '&lt;h4&gt;UD. Mitra Samudra&lt;/h4&gt;' + '&lt;div id="bodyContent"&gt;' +     '&lt;p&gt;Alamat : Jl. Banyuwangi, Landangan kranjan, Kapongan, Situbondo, Jawa Timur&lt;/p&gt;' + '&lt;p&gt;Koordinat : -7.6879373, 114.08247&lt;/p&gt;' +     '&lt;p&gt;&lt;b&gt;ADMIN FISH PICK UP TRACKING&lt;/b&gt;&lt;/p&gt;' + '&lt;/div&gt;' + '&lt;/div&gt;';      var latLng = new google.maps.LatLng(-7.6879373, 114.08247); addWaypointWithLabel(latLng, contentString); &lt;?php for (\$i = 0; \$i &lt; count(\$dataRekomendasi); \$i++) { echo "var contentString" . (\$i + 1) . " = '&lt;div id=\"content\"&gt;' . "&lt;div id=\"siteNotice\"&gt;&lt;/div&gt;" . "&lt;h4&gt;" . \$dataRekomendasi[\$i]['nama'] . "&lt;/h4&gt;" . "&lt;div id=\"bodyContent\"&gt;" . "&lt;p&gt;Alamat : " . \$dataRekomendasi[\$i]['alamat'] . "&lt;/p&gt;" . "&lt;p&gt;Koordinat : " . \$dataRekomendasi[\$i]['latitude'] . ", " . \$dataRekomendasi[\$i]['longitude'] . "&lt;/p&gt;" . "&lt;/div&gt;" . "&lt;/div&gt;"; echo "var latLng" . (\$i + 1) . " = new google.maps.LatLng(" . \$dataRekomendasi[\$i]['latitude'] . ", " . \$dataRekomendasi[\$i]['longitude'] . ");"; echo "addWaypointWithLabel(latLng" . (\$i + 1) . ", contentString" . (\$i + 1) . ");";} for (\$j = 0; \$j &lt; count(\$dataBukanRekomendasi); \$j++) { </pre>



```

echo "var                                pos={lat:". $dataBukanRekomendasi[$j]['latitude'].",
Ing:". $dataBukanRekomendasi[$j]['longitude']. "};";

echo "var marker_bukan_". $j. " = new google.maps.Marker({

    position: pos,

    map: gebMap,

    icon: 'http://maps.google.com/mapfiles/ms/icons/red-dot.png', });";

echo "var title_bukan_". $j. " = '<div id=\"content\">' .

"<div id=\"siteNotice\"></div>" .

"<h4>" . $dataBukanRekomendasi[$j]['nama'] . "</h4>" .

"<div id=\"bodyContent\">" .

"<p>Status      :      <b>BUKAN      REKOMENDASI</b></p><p>Alamat      :      "      .

$dataBukanRekomendasi[$j]['alamat'] . "</p>" .

"<p>Koordinat      :      "      .      $dataBukanRekomendasi[$j]['latitude']      .      ",      "      .

$dataBukanRekomendasi[$j]['longitude'] . "</p>" .

"</div>" .

"</div>";";

echo "var infowindow". $j. " = new google.maps.InfoWindow({

content: title_bukan_". $j. " });";

echo "marker_bukan_". $j. ".addListener('click', function() {

infowindow". $j. ".open(title_bukan_". $j. ", marker_bukan_". $j. ");});";      } ?>

    directions(1, false, false); //m(0 = SolveRoundTrip; 1 = SolveAtoZ), walking(true =
walking; false = driving), avoid }

```

#### 4.6. Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui seluruh kebutuhan sistem telah terpenuhi atau sistem layak untuk digunakan oleh *user*. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian sistem informasi geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO) di UD. Mitra

Samudra ini dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

#### 4.6.1. Metode *White Box*

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *white box* dilakukan oleh peneliti dengan cara menghitung *independent path* yaitu dengan menggunakan suatu pengukuran kuantitatif *cyclomatic complexity*, *listing program*, penentuan jalur independen, dan *test case*. Tahapan pengujian menggunakan metode *white box* meliputi:

##### a. *Listing Program*

*Listing program* merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

##### b. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari *listing program*. Diagram alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

##### c. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan rumus berikut ini.

$$V(G) = E - N + 2$$

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

d. Jalur Independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi.

e. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali. Pengujian sistem informasi geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO) di UD. Mitra Samudra menggunakan metode *white box* sebagai berikut:

1. Pengujian fitur rute terpendek penjemputan

a. *Listing* program fitur rute terpendek

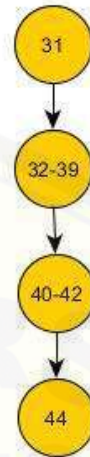
```

20 function getDrivingDistance($origin, $destination)
21 {
22     $url = "https://maps.googleapis.com/maps/api/distancematrix/json?origins={$origin}&destinations={$destination}&mode=driving&language=pl-Pl";
23     $ch = curl_init();
24     curl_setopt($ch, CURLOPT_URL, $url);
25     curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
26     curl_setopt($ch, CURLOPT_HEADER, 0);
27     curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
28     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
29     $response = curl_exec($ch);
30     curl_close($ch);
31     $response_a = json_decode($response, true);
32     $dist = $response_a['rows'][0]['elements'][0]['distance']['text'];
33     $time = $response_a['rows'][0]['elements'][0]['duration']['text'];
34
35     return array('distance' => $dist, 'time' => $time);
36 }

```

Gambar 4. 7 *Listing* program *function* GetDrivingDistance() (1)

b. Diagram alir fitur rute terpendek



Gambar 4. 8 diagram alir fitur rute terpendek

- c. Perhitungan *Cyclomatic Complexity* fitur rute terpendek

Perhitungan diagram alir pada fitur rute terpendek menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{Class users : } V(G) = E - N + 2 = 3 - 4 + 2 = 1$$

- d. Pengujian jalur program fitur rute terpendek

Pengujian jalur program fitur rute terpendek berdasarkan perhitungan *Cyclomatic Complexity* adalah sebagai berikut:

*function* GetDrivingDistance () : jalur 1 : 31-32-40-44

- e. *Test Case* fitur rute terpendek

Tabel 4. 5 *Test Case function GetDrivingDistance ()*

<i>Test Case function GetDrivingDistance ()</i>	
Jalur 1	
<i>Test Case</i>	Jika berhasil menghitung jarak, arah, dan waktu tempuh
Target yang diharapkan	Menampilkan jarak, arah, dan waktu tempuh dan menampilkan pada halaman daftar rekomendasi
Hasil pengujian	Benar
Path/Jalur	31-32-40-44

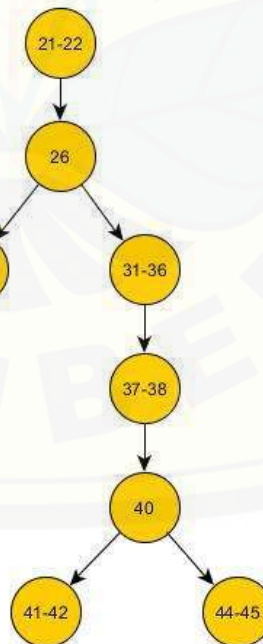
## 2. Pengujian fitur login

### a. Listing program fitur login

```
20 public function otentikasi(){
21     $username = $this->input->post('user');
22     $password = $this->input->post('password');
23     $result = $this->login->get_data($username, md5($password))>result_array();
24
25     // print_r($result);
26     if(empty($result))
27     {
28         $this->session->set_flashdata('gagal', 'Username atau Password salah');
29         redirect(base_url().'login');
30     } else{
31         $session = array (
32             'username' => $result[0]['username'],
33             'kategori' => $result[0]['kategori'],
34             'id_aktor' => $result[0]['id_aktor'],
35             'nama' => $result[0]['nama']
36         );
37         $this->session->set_userdata ('userlogin', $session);
38         $pengguna = $this->session->userdata('userlogin');
39
40         if ($userlogin['kategori'] == 'Administrator'){
41             $this->session->set_flashdata('sukses', 'Selamat datang $userlogin['nama'] 'anda login sebagai' $userlogin['kategori']);
42             redirect(base_url().'dashboard');
43         } else {
44             $this->session->set_flashdata('sukses', 'Selamat datang $userlogin['nama'] 'anda login sebagai' $userlogin['kategori']);
45             redirect(base_url().'dashboard');
46         }
47     }
48     // print_r('berhasil');
49 }
50 }
```

Gambar 4. 9 Listing program *function* otentikasi() (2)

### b. Diagram Alir fitur login



Gambar 4. 10 Diagram alir fitur login



c. Perhitungan *Cyclomatic Complexity*

Perhitungan diagram alir pada fitur login menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{Class users : } V(G) = E - N + 2 = 7 - 8 + 2 = 1$$

d. Pengujian jalur program fitur login

Pengujian jalur program fitur login berdasarkan perhitungan *Cyclomatic Complexity* adalah sebagai berikut:

*function* otentikasi () : Jalur 1 : 21-26-29

Jalur 2 : 21-26-36-38-40-42

Jalur 3 : 21-26-36-38-40-45

e. *Test Case* fitur login

Tabel 4. 6 *Test Case function otentikasi ()*

<i>Test Case function otentikasi ()</i>	
Jalur 1	
<i>Test Case</i>	Jika gagal melakukan login
Target yang diharapkan	Menampilkan kembali halaman login dan flashdata gagal
Hasil pengujian	Benar
Path/Jalur	21-26-29
Jalur 2	
<i>Test Case</i>	Jika berhasil login sebagai administrator
Target yang diharapkan	Menampilkan halaman dashboard admin dan flashdata sukses
Hasil pengujian	Benar
Path/Jalur	21-26-36-38-40-42
Jalur 3	
<i>Test Case</i>	Jika berhasil login sebagai member
Target yang diharapkan	Menampilkan halaman dashboard member dan flashdata sukses

Hasil pengujian

Benar

Path/Jalur

21-26-36-38-40-45

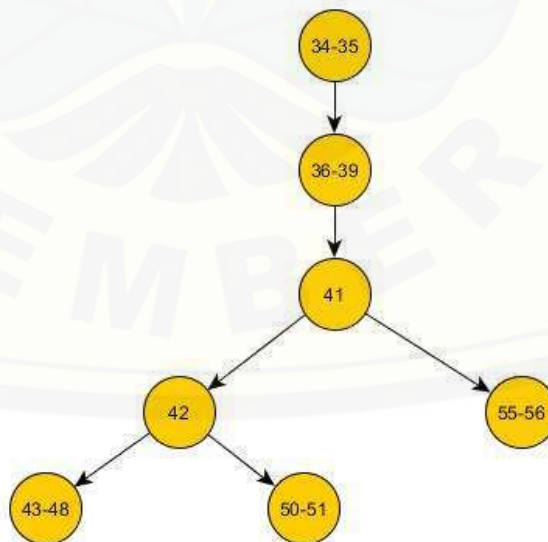
### 3. Pengujian fitur edit password

#### a. Listing program fitur edit password

```
33 function editPass(){
34     $userLogin = $this->session->userdata('userLogin');
35     $user['single_member'] = $userLogin['id_aktor'];
36     $pass_lama = $this->input->post('password_lama');
37     $pass_baru = $this->input->post('password_baru');
38     $kon_pass = $this->input->post('konfirmasi_password');
39     $cekPassLama = $this->m_pengumpul->cekPassLama($this->session->userdata('userLogin')['username'],md5($pass_lama));
40     // print_r($cekPassLama);
41     if(!empty($cekPassLama)){
42         if($pass_baru == $kon_pass){
43             $data = array(
44                 'password' => md5($pass_baru)
45             );
46             $query = $this->m_pengumpul->editPassword($userLogin['id_aktor'], $data);
47             $this->session->set_flashdata('sukses', 'Password berhasil diperbarui');
48             redirect(base_url().'index.php/Profil/index/'.$userLogin['id_aktor']);
49         }else {
50             $this->session->set_flashdata('gagal', 'Password tidak cocok');
51             redirect(base_url().'index.php/Profil/password');
52         }
53     }else {
54         // password tidak cocok
55         $this->session->set_flashdata('gagal', 'Password lama yang anda inputkan tidak sesuai');
56         redirect(base_url().'index.php/Profil/password');
57     }
58 }
59 // echo "<br>";
60 // print_r($data);
61 // print_r($data1);
62 }
```

Gambar 4. 11 Listing program function editPass() (3)

#### b. Diagram alir fitur edit password



Gambar 4. 12 diagram alir fitur edit password

c. Perhitungan *Cyclomatic Complexity*

Perhitungan diagram alir pada fitur edit password menggunakan *Cyclomatic Complexity* adalah sebagai berikut:

$$\text{Class users : } V(G) = E - N + 2 = 6 - 7 + 2 = 1$$

d. Pengujian jalur program fitur edit password

Pengujian jalur program fitur edit password berdasarkan perhitungan *Cyclomatic Complexity* adalah sebagai berikut:

*function editPass ()* : Jalur 1 : 34-39-41-42-48

Jalur 2 : 34-39-41-42-51

Jalur 3 : 34-39-41-56

e. *Test Case* fitur edit password

Tabel 4. 7 *Test Case editPass ()*

<i>Test Case editPass ()</i>	
Jalur 1	
<i>Test Case</i>	Jika password lama benar, password baru dan konfirmasi password cocok
Target yang diharapkan	Menampilkan halaman profil dan flashdata sukses
Hasil pengujian	Benar
Path/Jalur	34-39-41-42-48
Jalur 2	
<i>Test Case</i>	Jika password lama benar, password baru dan konfirmasi password tidak cocok
Target yang diharapkan	Menampilkan halaman edit password dan flashdata gagal
Hasil pengujian	Benar
Path/Jalur	34-39-41-42-51
Jalur 3	
<i>Test Case</i>	Jika password lama salah

Target yang diharapkan	Menampilkan halaman edit password dan flashdata gagal
Hasil pengujian	Benar
Path/Jalur	34-39-41-56

---

#### 4.6.2. Metode *Black Box*

*Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. Pengujian black box pada sistem informasi geografi untuk sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO) dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada setiap *usecase*. Hasil pengujian *black box* dapat dilihat pada Lampiran E (Pengujian *Black Box*).

## BAB 6. PENUTUP

Bab ini merupakan bagian akhir dari penulisan skripsi yang berisi tentang kesimpulan dan saran. Kesimpulan berisi hasil penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

### 6.1. Kesimpulan

Kesimpulan dari sistem yang dibangun pada penelitian ini adalah sebagai berikut:

1. Fitur data stok ikan pengepul yang dapat diakses oleh admin UD. Mitra Samudra dapat memudahkan UD. Mitra Samudra untuk mengetahui ketersediaan ikan pada setiap pengepul sehingga mendapatkan banyak pasokan ikan dari para pengepul yang menjadi member dalam sistem yang dibangun sekaligus meminimalisir biaya operasional UD. Mitra Samudra.
2. Penerapan Sistem Informasi Geografi untuk pemetaan sebaran pengepul di Kabupaten Situbondo berhasil dilakukan dengan memanfaatkan *Google Maps API*. Sistem yang dibuat sangat berguna untuk menjangkau seluruh pengepul yang tersebar di Kabupaten Situbondo, sehingga membuat jaringan bisnis UD. Mitra Samudra semakin meluas.
3. Penerapan metode *Ant Colony Optimization* (ACO) dalam sistem yang dibangun ini mampu menentukan rute terpendek dari permasalahan graf yang kompleks, dengan mengadopsi kemampuan alami koloni semut dalam melakukan perjalanan mencari lokasi sumber makanan hingga kembali lagi ke sarang. Masing-masing semut ( $m$ ) melakukan perjalanan melalui jalur yang berbeda. Pada jalur yang dilewati semut akan meninggalkan jejak atau feromon ( $\tau_{ij}$ ) yang berisi informasi keadaan jalur yang dilewati dan berfungsi sebagai alat komunikasi secara tidak langsung. Semakin banyak feromon pada jalur, maka akan semakin banyak semut yang melewati jalur tersebut. Demikian juga sebaliknya semakin sedikit feromon pada jalur tertentu, maka akan sedikit



semut yang melewati bahkan tidak ada. Dengan begitu akan terpilih jalur dengan nilai feromon ( $\tau_{ij}$ ) terbesar adalah rute terpendek. Permasalahan kompleks pada kasus ini yaitu menjangkau seluruh alamat pengepul yang dipilih dalam daftar rekomendasi jemputan ikan.

## 6.2. Saran

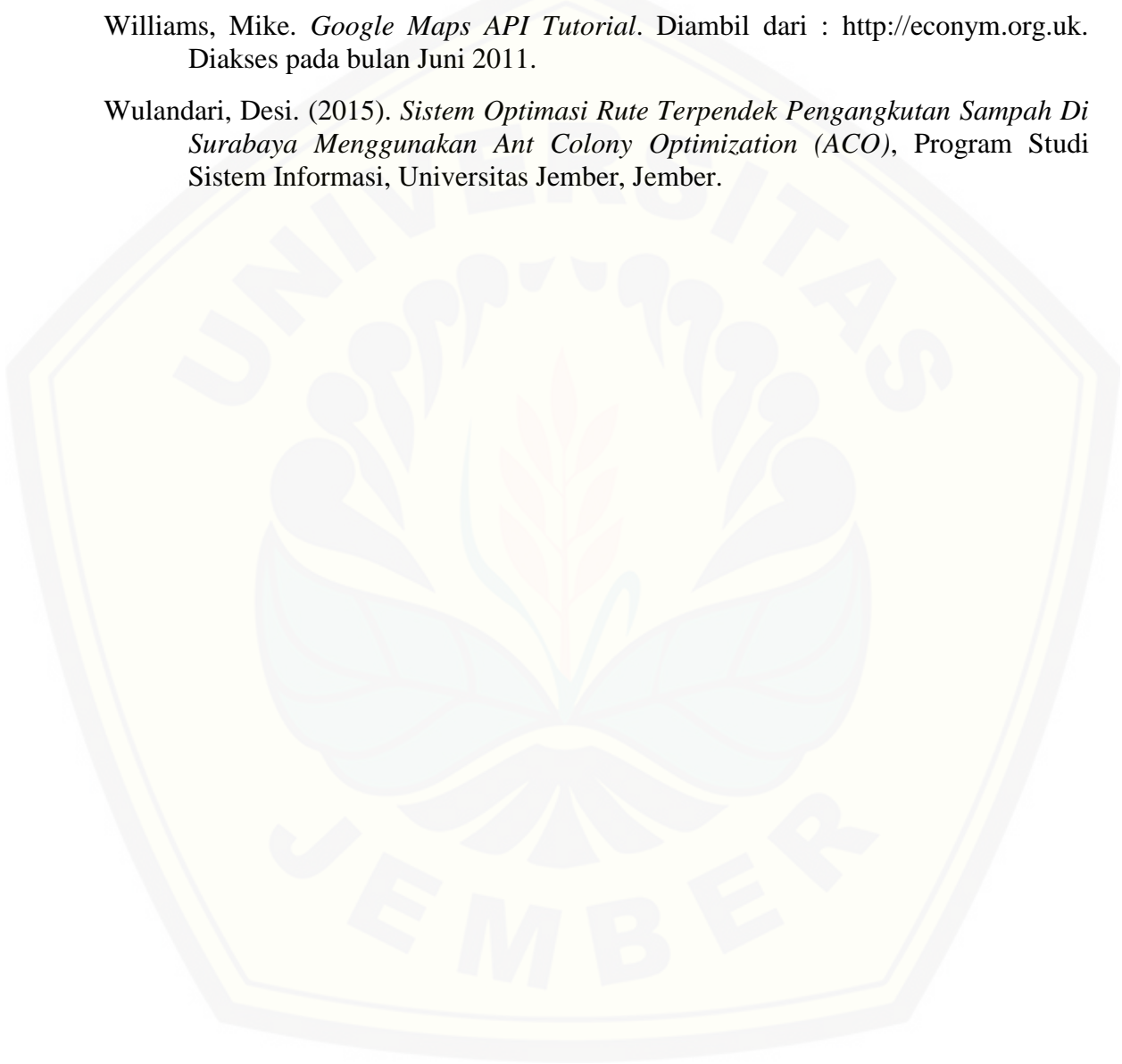
Hasil yang dicapai dari penelitian ini belum sempurna, oleh karena itu untuk meningkatkan hasil yang dicapai ada beberapa saran untuk pengembangan penelitian ini diantaranya:

1. Sistem informasi geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO) ini masih membutuhkan pengembangan lebih lanjut dengan membangun *web view* yang berbasis android, sehingga memudahkan driver UD. Mitra Samudra juga dapat melihat rute terpendek penjemputan ikan secara langsung.
2. Sistem informasi geografi untuk pemetaan sebaran penjemputan ikan dengan metode *Ant Colony Optimization* (ACO) ini membutuhkan pengembangan untuk informasi bahwa telah dilakukan penjemputan.

**DAFTAR PUSTAKA**

- Martua, Paulus Bangun. (2011). *Perancangan Algoritma Ant Colony Optimization (ACO) , untuk Penyelesaian Vehicle Routing Problem (VRP)*, Program Studi Teknik Industri, Fakultas Teknik, Universitas Indonesia, Depok.
- Hamdani, Rofiq. (2016). *Konsep Bisnis Ikan Laut Segar*. Wawancara oleh penulis, pukul 16.00.
- Leksono, Agus. (2009). *Algoritma Ant Colony Optimization (ACO) untuk menyelesaikan Traveling Salesman Problem (TSP)*, Program Studi Matematika, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Diponegoro, Semarang.
- Firdaus, Rizqi Maulana. (2015). *Perancangan Sistem Informasi Pencarian Hotel Di Wilayah Jember Dengan Menggunakan Google Maps API*, Program Studi Sistem Informasi, Universitas Jember, Jember.
- Sirenden, Bernadus Herdi., Dachi, Ester Laekha. (2012). *Buat sendiri petamu menggunakan Codeigniter dan Google Maps API*. Yogyakarta. Penerbit Andi
- Prahasta, Eddy. 2005. *Konsep- Konsep Dasar Sistem Informasi Geografis*. cetakan kedua, Informatika, Bandung.
- Arifin., Fariza, Arna dan Ahsan, Ahmad Syauqi. (2014). *Sistem Informasi Geografis (SIG) Fasilitas Umum Kota Mojokerto Berbasis Web*. Jurusan Teknik Informatika, Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember, Surabaya.
- Qolis, Nur dan Fariza, Arna. (2008). *Pemetaan dan Analisa Sebaran Sekolah Untuk Peningkatan Layanan Pendidikan di Kabupaten Kediri dengan GIS*. Institut Teknologi Sepuluh Nopember, Surabaya.
- Sugandi, Dede dan Nanin, T.S. (2008). *Handout : Sistem Informasi Geografi (SIG)*. UPI, Bandung.
- Nirwan. (2012). *Ikan Palagis dan Ikan Damersal serta Biota Laut yang Dilindungi*. Diambil dari : <http://marinescience-nirwan.blogspot.co.id/2012/06/penjelasan-mengenai-ikan-pelagis-dan.html>. Diakses pada bulan juni
- Setyawan, Andika., et al. (2013). *Perbandingan Algoritma Ant Colony Optimization, Disjktra, Tabu Search, Multiple Ant Colony System untuk Vehicle Routing Problem dengan Time Window*, Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, UNS, Surakarta.

- Tyas, Yuliyani Siyamting dan Prijodiprodjo, Widodo. (2013). *Aplikasi Pencarian Rute Terpendek dengan Metode Ant Colony Optimization (ACO)*. Yogyakarta
- Wardy, I. S. (2007). *Penggunaan graph dalam algoritma semut untuk melakukan optimisasi*, Program studi Teknik Informatika, ITB, Bandung.
- Williams, Mike. *Google Maps API Tutorial*. Diambil dari : <http://econym.org.uk>. Diakses pada bulan Juni 2011.
- Wulandari, Desi. (2015). *Sistem Optimasi Rute Terpendek Pengangkutan Sampah Di Surabaya Menggunakan Ant Colony Optimization (ACO)*, Program Studi Sistem Informasi, Universitas Jember, Jember.



## LAMPIRAN

## LAMPIRAN A. USE CASE SKENARIO

## A.1 Use Case Skenario Login Member

Tabel A. 1 Use Case Skenario Login Member

<b>ID</b>	USC 02
<b>Nama Use Case</b>	<b>Login</b>
<b>Aktor</b>	Member
<b>Entry Condition</b>	Aktor melakukan Login
<b>Exit Condition</b>	Aktor telah memasuki sistem
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Membuka halaman web	
	2. Menampilkan halaman beranda web
3. Memilih menu Log In	
	4. Menampilkan form Login
5. Memasukkan username dan password	
6. Klik tombol login	
	7. Eksekusi validasi username dan password di dalam database
	8. Menampilkan flashdata “sukses”
	9. Menampilkan halaman Dashboard Pengepul dengan menu. - Dashboard - Profil - Data Stok Ikan
Alternatif (Username dan Password salah)	

	8. Menampilkan flashdata gagal
--	--------------------------------

### A.2 Use Case Skenario Melihat peta sebaran pengepul

Tabel A. 2 Use Case Skenario Melihat peta sebaran pengepul

<b>ID</b>	USC 03
<b>Nama Use Case</b>	<b>Melihat Peta Sebaran Pengepul</b>
<b>Aktor</b>	Admin UD. Mitra Samudra
<b>Entry Condition</b>	Aktor membuka halaman dashboard
<b>Exit Condition</b>	Aktor melihat peta sebaran pengepul
<b>Reaksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor membuka halaman dashboard	
	2. Menampilkan halaman Dashboard dengan konten peta sebaran pengepul di kabupaten Situbondo
3. Melihat Peta sebaran pengepul	

### A.3 Use Case Skenario Menentukan rute terpendek

Tabel A. 3 Use Case Skenario Menentukan rute terpendek

<b>ID</b>	USC 04
<b>Nama Use Case</b>	<b>Menentukan Rute Terpendek</b>
<b>Aktor</b>	Admin UD. Mitra Samudra
<b>Entry Condition</b>	Aktor memilih menu Daftar Rekomendasi
<b>Exit Condition</b>	Aktor melihat rute jemputan yang dihasilkan oleh sistem
<b>Reaksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor memilih menu Daftar Rekomendasi	



	2. Menampilkan halaman Daftar Rekomendasi - Menampilkan Rute terpendek penjemputan ikan dan node pengepul yang mempunyai stok - Menampilkan tabel penunjuk arah - Menampilkan tabel daftar rekomendasi penjemputan terkini
--	---

#### A.4 Use Case Skenario Mengelola data pengepul

Tabel A. 4 Use Case Skenario Mengelola data pengepul

<b>ID</b>	USC 05
<b>Nama Use Case</b>	<b>Mengelola Data Pengepul</b>
<b>Aktor</b>	Admin UD. Mitra Samudra
<b>Entry Condition</b>	Aktor memilih menu Daftar Pengepul
<b>Exit Condition</b>	Aktor berhasil menambahkan atau mengubah data pengepul
Reaksi Aktor	Reaksi Sistem
<i>Normal Flow</i> “Menambahkan Data Pengepul Baru”	
1. Memilih menu Daftar pengepul	
	2. Menampilkan halaman daftar pengepul
	3. Menampilkan tabel daftar pengepul
4. Klik tombol “Tambah”	
	5. Menampilkan form pengepul
6. Mengisi data pengepul	
7. Klik “Simpan”	
	8. Mengeksekusi form pengepul

	9. Menampilkan flashdata sukses
<b>Alternatif flow (ketika terjadi kesalahan dalam pengisian form)</b>	
7. Klik “Simpan”	
	8. Menampilkan flashdata gagal
	9. Sistem akan menampilkan kembali form pengepul
10. Mengisi data	
11. Klik “Simpan”	
	12. Menampilkan flashdata sukses
<i>Normal Flow</i> “Mengubah data pengepul”	
4. Klik tombol Edit	
	5. Menampilkan form data pengepul
6. Mengubah data pengepul pada form	
7. Klik tombol simpan	
	8. Menampilkan flashdata sukses

#### A.5 Use Case Skenario Melihat data ikan pengepul

Tabel A. 5 Use Case Skenario Melihat data ikan pengepul

<b>ID</b>	USC 06
<b>Nama Use Case</b>	<b>Melihat Data Ikan Pengepul</b>
<b>Aktor</b>	Admin UD. Mitra Samudra
<b>Entry Condition</b>	Aktor memilih menu Data Ikan Pengepul
<b>Exit Condition</b>	Aktor melihat data ikan yang tersedia pada setiap pengepul
<b>Reaksi Aktor</b>	<b>Reaksi Sistem</b>
1. Memilih menu Data Ikan Pengepul	
	2. Menampilkan halaman data ikan pengepul

	3. Menampilkan tabel data ikan
4. Memilih baris ikan dan klik tombol ikon detail	
	5. Menampilkan detail data ikan pengepul dari pengepul terpilih
6. Melihat data ikan	

#### A.6 Use Case Skenario Mengelola profil

Tabel A. 6 Use Case Skenario Mengelola profil

<b>ID</b>	USC 07
<b>Nama Use Case</b>	<b>Mengelola Profil</b>
<b>Aktor</b>	Member
<b>Entry Condition</b>	Aktor memilih menu Data Stok Ikan
<b>Exit Condition</b>	Aktor berhasil menambahkan atau mengubah data stok ikan
Reaksi Aktor	Reaksi Sistem
<i>Normal Flow</i> “Menambahkan Data Ikan”	
1. Memilih menu Profil	
	2. Menampilkan halaman profil
	3. Menampilkan profil aktor
4. Mengklik tombol “edit password”	
	5. Menampilkan form ganti password
6. Mengisi form	
7. Klik “simpan”	
	8. Mengeksekusi form ganti password
	9. Menampilkan flashdata sukses
Alternatif flow (ketika terjadi kesalahan dalam pengisian form)	

6. Klik “Simpan”	
	7. Menampilkan flashdata gagal
	8. Sistem menampilkan kembali form stok ikan
9. Mengisi data yang belum diisi	
10. Klik “Simpan”	
	11. Menampilkan flashdata sukses
<i>Normal Flow</i> “Mengubah data ikan”	
4. Klik tombol Edit	
	5. Menampilkan form transaksi ikan yang telah diisi
6. Mengubah data ikan pada form stok ikan	
7. Mengklik tombol simpan	
	8. Menampilkan flashdata sukses

#### A.7 Use Case Skenario Mengelola data ikan

Tabel A. 7 Use Case Skenario Mengelola data ikan

<b>ID</b>	USC 07
<b>Nama Use Case</b>	<b>Mengelola Data Ikan</b>
<b>Aktor</b>	Member
<b>Entry Condition</b>	Aktor memilih menu Data Stok Ikan
<b>Exit Condition</b>	Aktor berhasil menambahkan atau mengubah data stok ikan
Reaksi Aktor	Reaksi Sistem
<i>Normal Flow</i> “Menambahkan Data Ikan”	
1. Memilih menu Data Stok Ikan	
	2. Menampilkan halaman data ikan

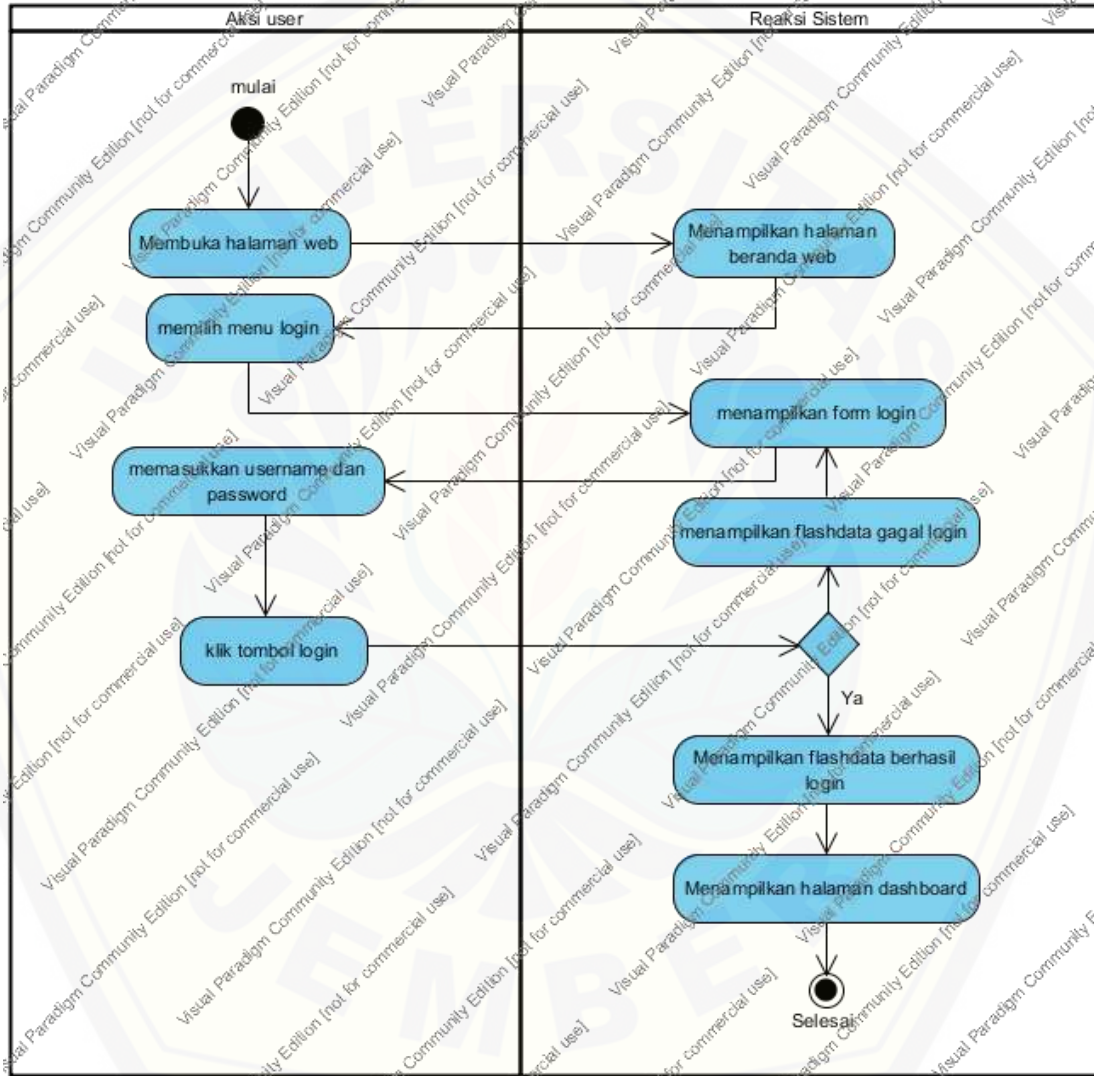
	3. Menampilkan tabel data ikan
4. Mengklik tombol “Tambah”	
	5. Menampilkan form transaksi ikan
6. Mengisi data ikan	
7. Klik “tambah”	
	8. Menambahkan data ke tabel ikan
9. Klik “Simpan”	
	10. Mengeksekusi form stok ikan
	11. Menampilkan flashdata sukses
<i>Alternatif flow (ketika terjadi kesalahan dalam pengisian form)</i>	
12. Klik “Simpan”	
	13. Menampilkan flashdata gagal
	14. Sistem menampilkan kembali form stok ikan
15. Mengisi data yang belum diisi	
16. Klik “Simpan”	
	17. Menampilkan flashdata sukses
<i>Normal Flow “Mengubah data ikan”</i>	
9. Klik tombol Edit	
	10. Menampilkan form transaksi ikan yang telah diisi
11. Mengubah data ikan pada form stok ikan	
12. Mengklik tombol simpan	
	13. Menampilkan flashdata sukses



**LAMPIRAN B. ACTIVITY DIAGRAM**

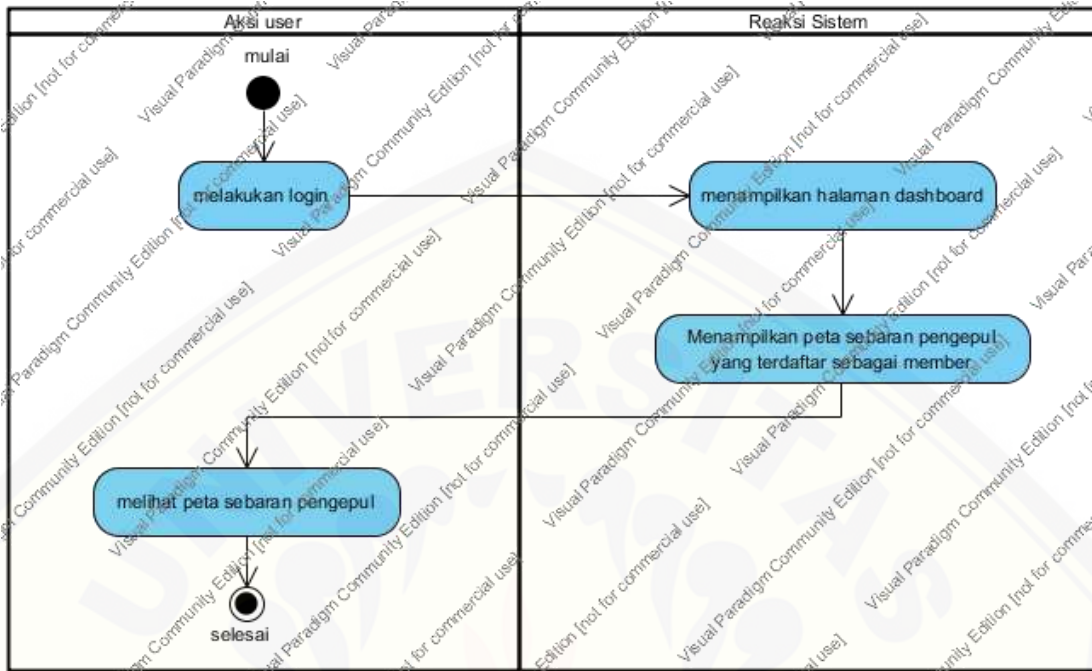
**B.1 Activity Member – Login**

No table of figures entries found.



Gambar B. 1 Activity Member – Login

**B.2 Activity Admin – Melihat peta sebaran pengepul**



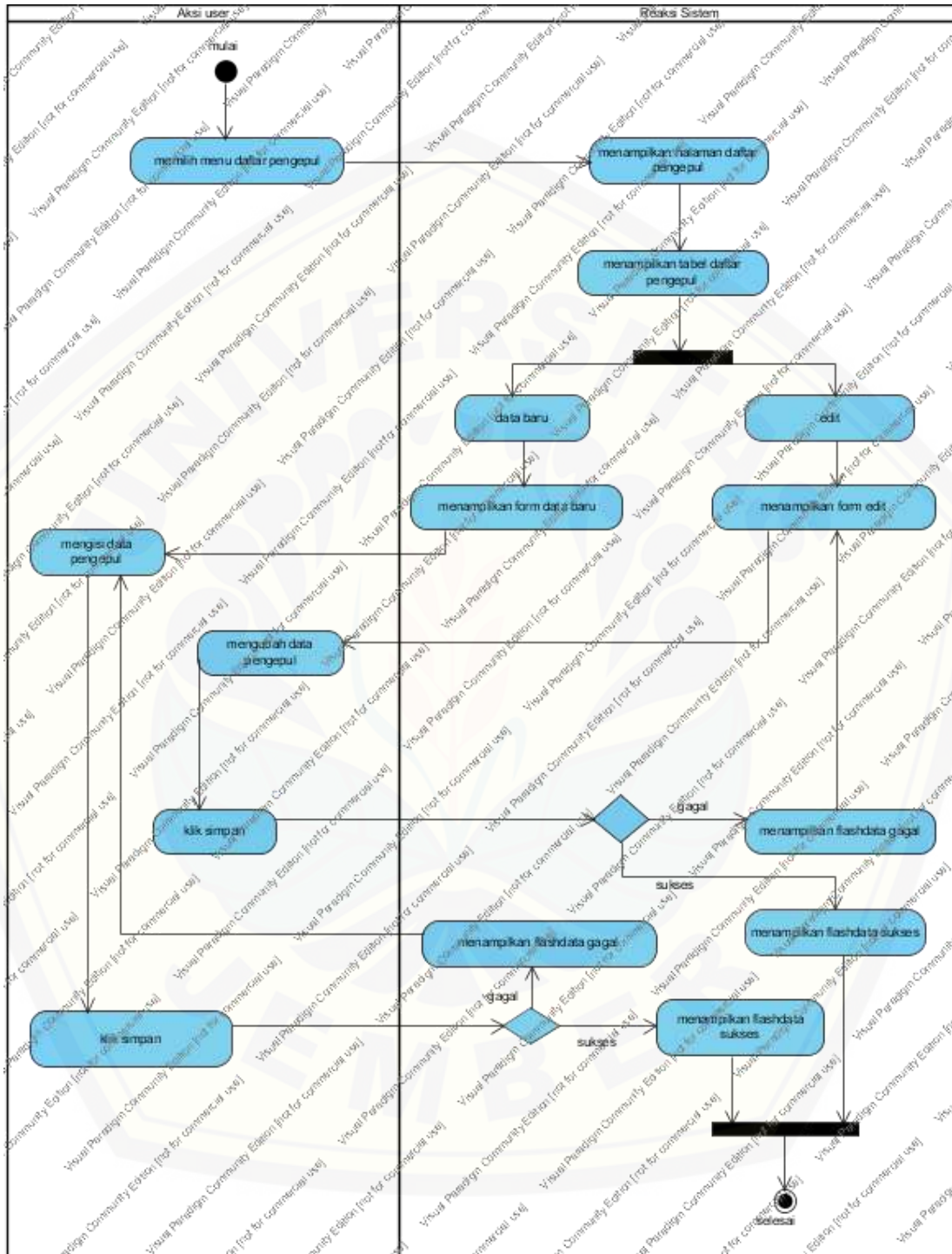
Gambar B. 2 Activity Admin – Melihat peta sebaran pengepul

B.3 Activity Admin – Menentukan rute terpendek



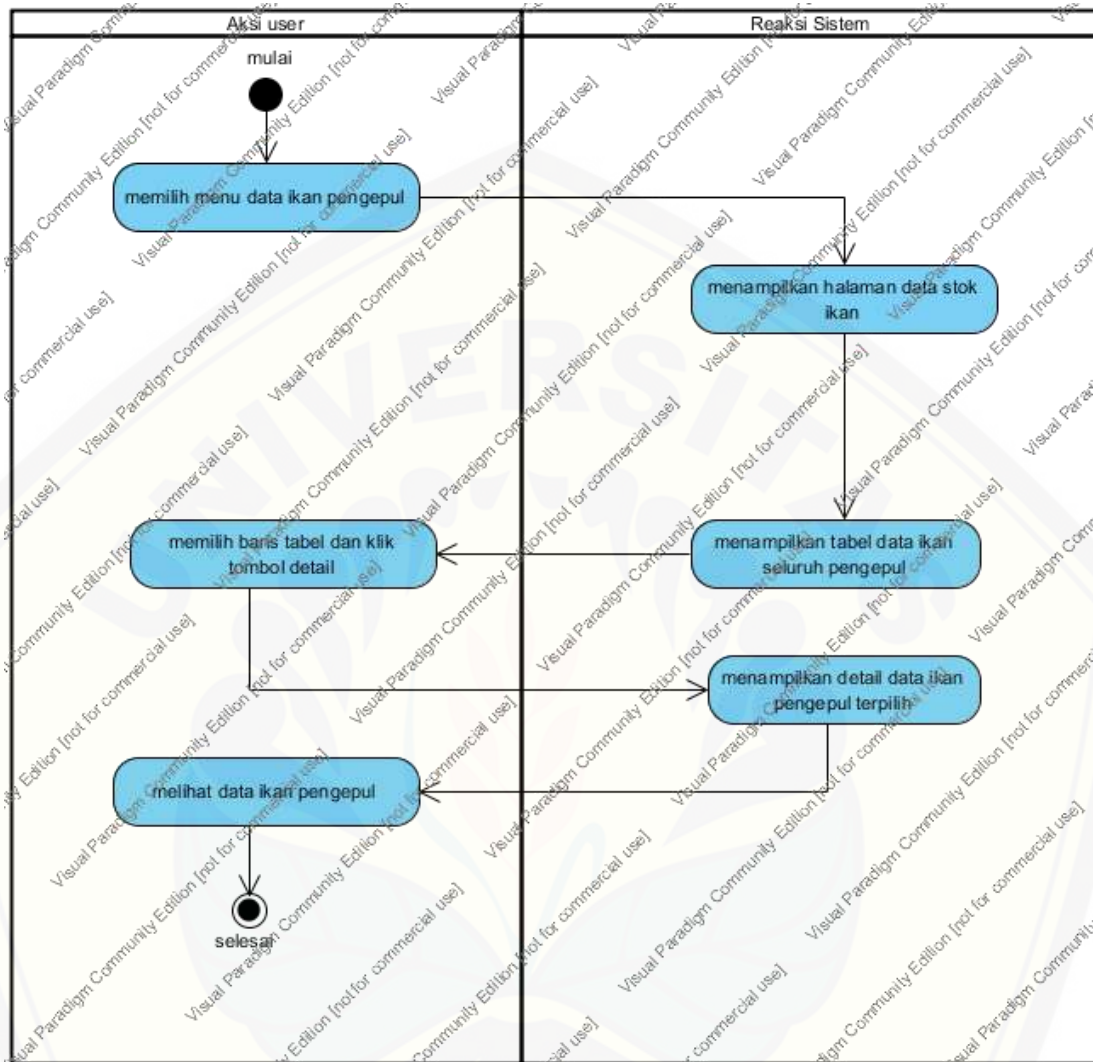
Gambar B. 3 Activity Admin – Menentukan rute terpendek

B.4 Activity Admin – Mengelola data pengepul



Gambar B. 4 Activity Admin – Mengelola data pengepul

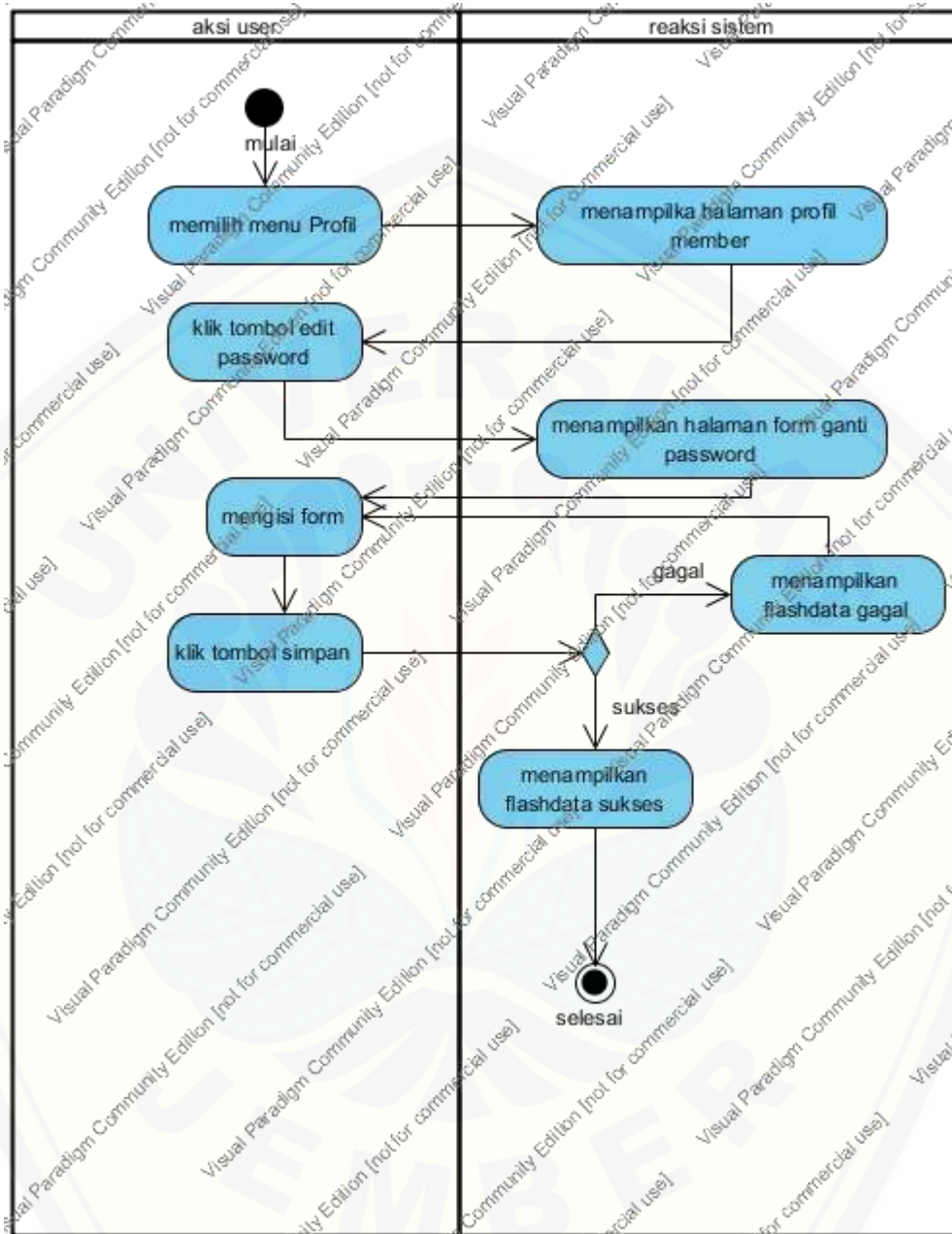
B.5 Activity Admin – Melihat data ikan pengepul



Gambar B. 5 Activity Admin – Melihat data ikan pengepul

#### B.6 Activity Member – Mengelola profil

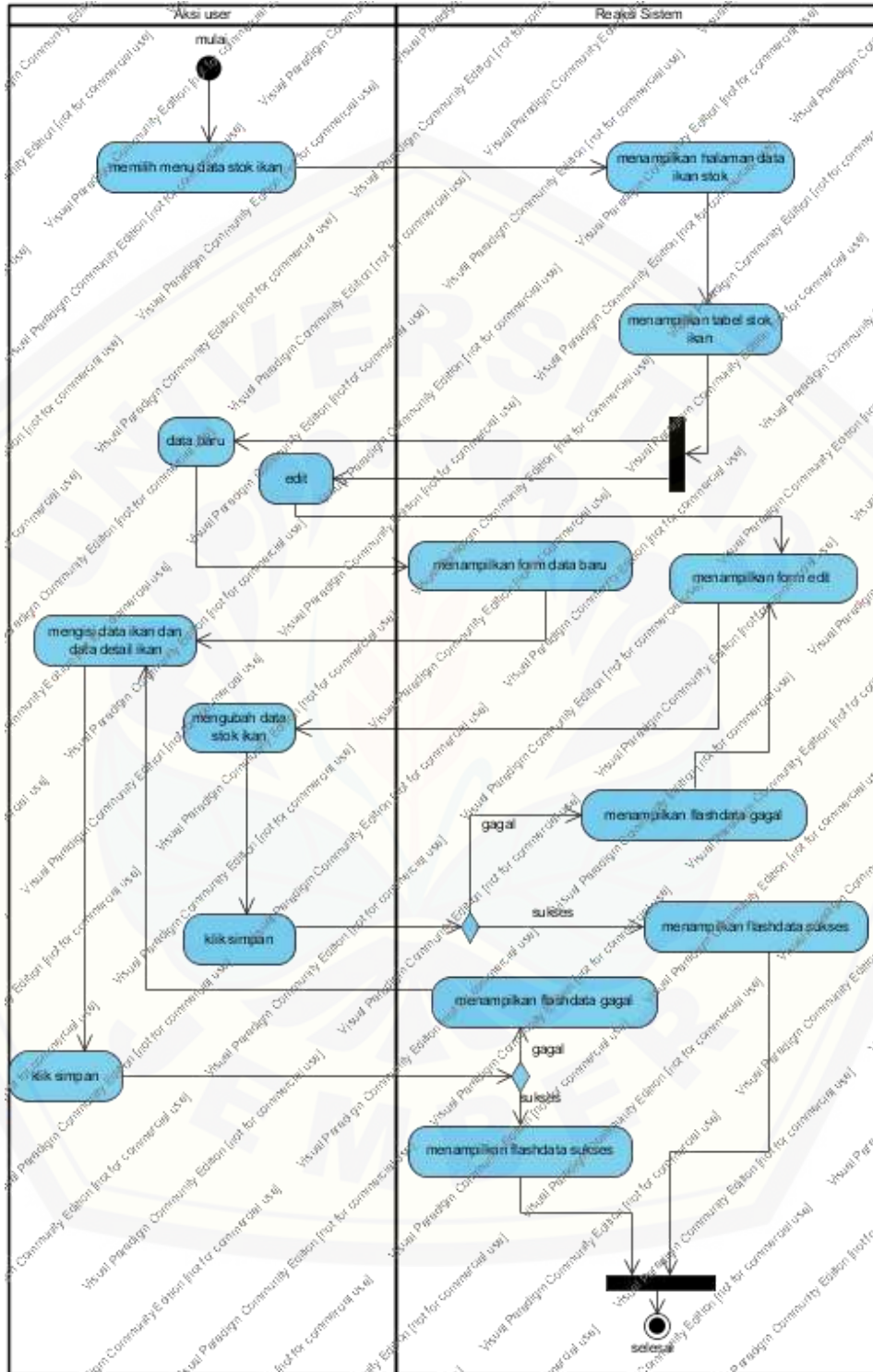




Gambar B. 6 Activity Member – Mengelola Profil

B.7 Activity Member – Mengelola data ikan

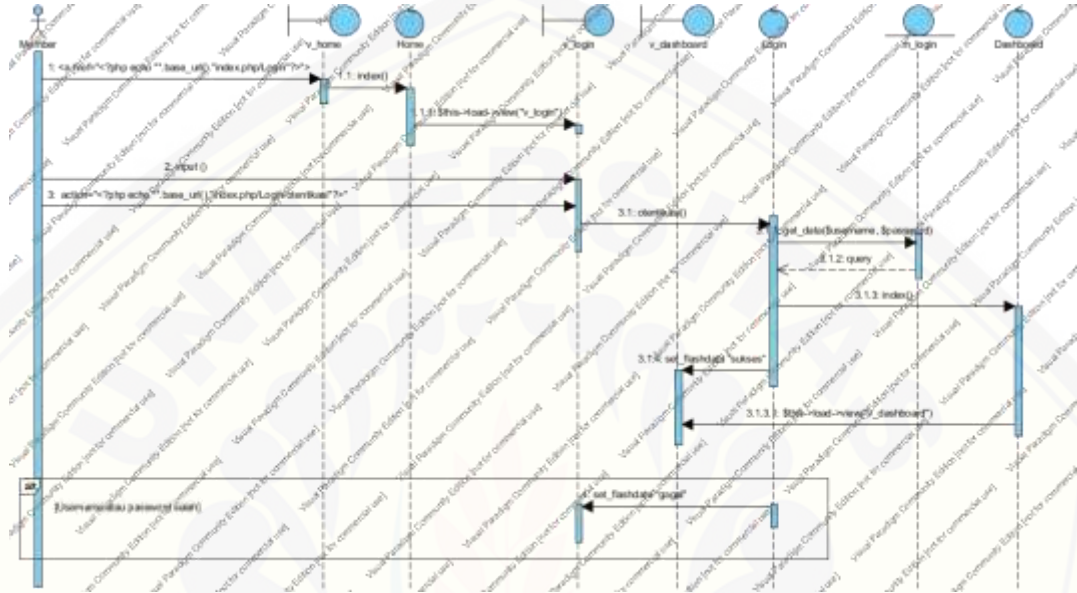




Gambar B. 7 Activity Member – Mengelola data ikan

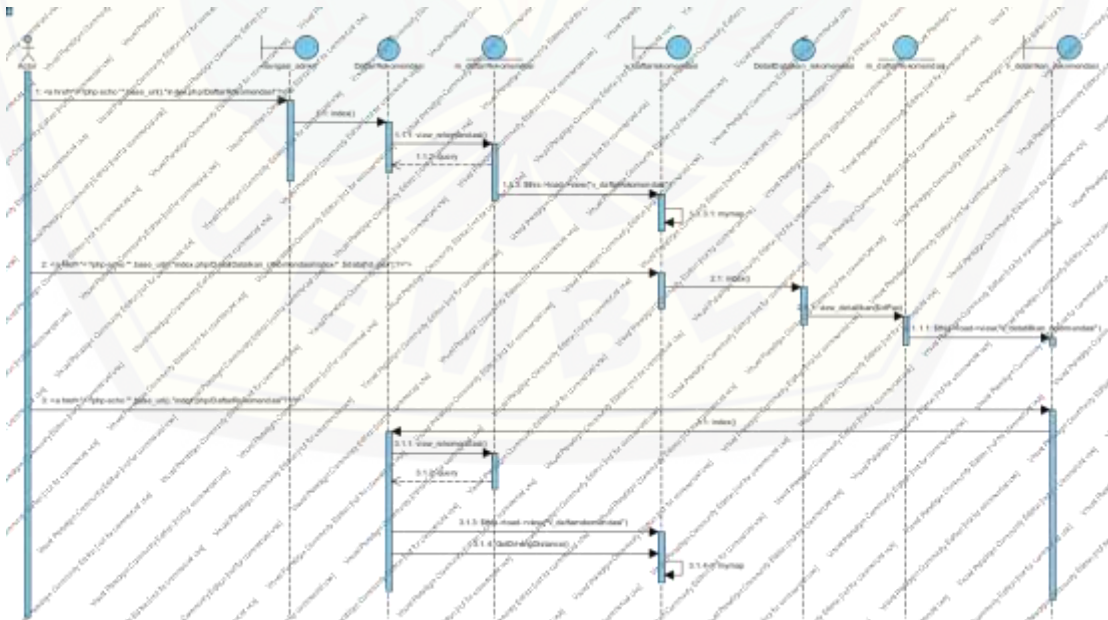
### LAMPIRAN C. SEQUENCE DIAGRAM

#### C.1 Sequence Member – Login



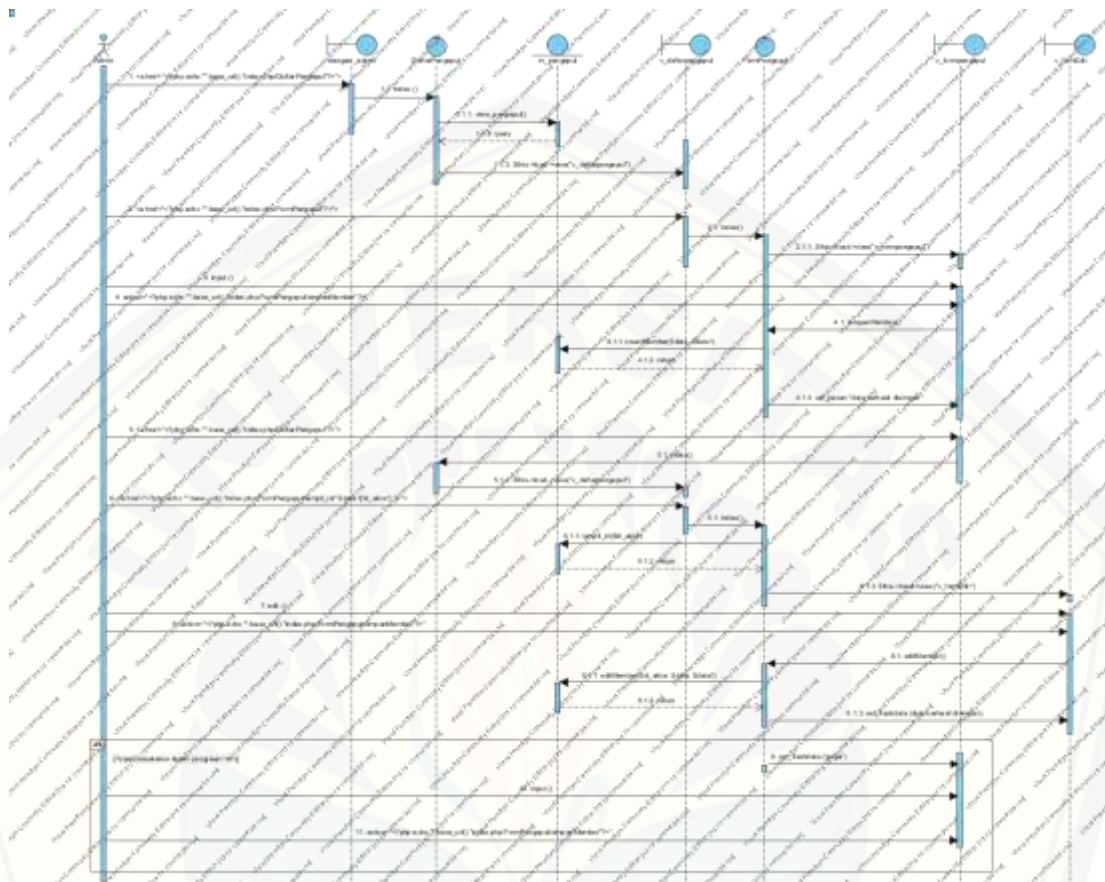
Gambar C. 1 Sequence Member - Login

#### C.2 Sequence Admin – Menentukan rute terpendek



Gambar C. 2 Sequence Admin – Menentukan rute terpendek

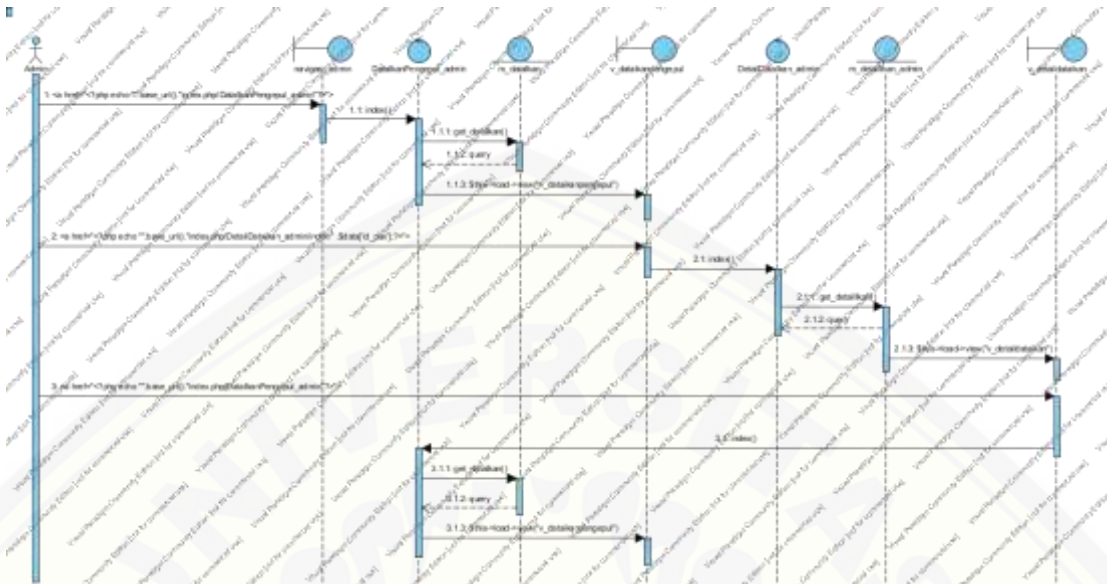
### C.3 Sequence Admin – Mengelola data pengepul



Gambar C. 3 Sequence Admin – Mengelola data pengepul

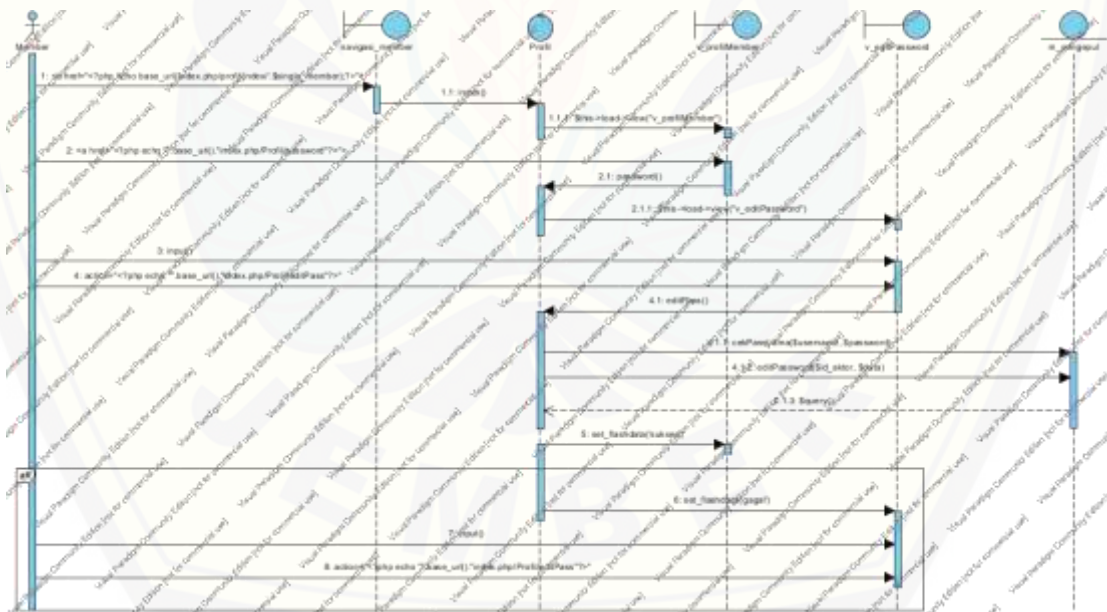
### C.4 Sequence Admin – Melihat data ikan pengepul





Gambar C. 4 Sequence Admin – Melihat data ikan pengumpul

C.5 Sequence Member – Mengelola profil



Gambar C. 5 Sequence Member – Mengelola profil

C.6 Sequence Member – Mengelola data ikan





## LAMPIRAN D. KODE PROGRAM

### D.1 Class Controller Login

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Login extends CI_Controller {
    public function __construct () {
        parent::__construct();
        $this->load->model('m_login','login');
        $this->load->model('m_pengepul');
    }
    public function index()
    {
        $this->load->view('baseheader_login');
        $this->load->view('v_login');
        $this->load->view('basefooter_login');
    }
    public function otentikasi(){
        $username          = $this->input->post('user');
        $password          = $this->input->post('password');
        $result            = $this->login->get_data($username,
md5($password))->result_array();
        if(empty($result))
        {
            $this->session->set_flashdata('gagal', 'Username atau
Password salah');
            redirect(base_url()).'Login');
        } else{
            $session = array (
                'username'          => $result[0]['username'],
                'kategori'         => $result[0]['kategori'],
                'id_aktor'        => $result[0]['id_aktor'],
                'nama'            => $result[0]['nama'] );
            $this->session->set_userdata ('userLogin', $session);
            $pengguna = $this->session->userdata('userLogin');
            if ($userLogin['kategori'] == 'Administrator'){
                $this->session->set_flashdata('sukses', 'Selamat
datang'. $userLogin['nama']. 'anda login sebagai'. $userLogin['kategori']);
                redirect(base_url()).'Dashboard');
            } else {
                $this->session->set_flashdata('sukses', 'Selamat
datang'. $userLogin['nama']. 'anda login sebagai'. $userLogin['kategori']);
                redirect(base_url()).'Dashboard');
            }
        }
    }
}

```

```

    }
}
public function logout(){
    $this->session->unset_userdata('userLogin');
}
}

```

## D.2 Class Controller Dashboard

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Dashboard extends CI_Controller {

    public function __construct () {
        parent::__construct();
        $this->load->library('session');
        $this->load->helper('url');
        $this->load->model('m_dashboard');
        $this->load->model('m_pengepul');
    }

    public function index() {
        $userLogin = $this->session->userdata('userLogin');

        if(empty($userLogin)) {
            $this->session->set_flashdata('gagal', 'Username atau Password salah');
            redirect(base_url(). 'Login');
        } else {
            if ($userLogin['kategori'] == 'Administrator'){
                // $data['dataRekomendasi']=$this->m_dashboard->view_rekomendasi();
                $data['member']= $this->db->get('tb_aktor')->result_array();
                $this->session->set_flashdata('sukses', 'Selamat datang di Halaman
                Dashboard Fish Pick-Up Tracking! anda login sebagai
                '.$userLogin['kategori']);
                $this->load->view('ADMIN/baseheader_admin');
                $this->load->view('ADMIN/navigasi_admin');
                $this->load->view('ADMIN/v_dashboard',$data);
                $this->load->view('ADMIN/basefooter_admin');
            } else {
                $data1['dataMember']=$this->m_pengepul->view_member();
                $user['single_member'] = $userLogin['id_aktor'];
                $this->session->set_flashdata('sukses', 'Selamat datang,

```

```
        '$userLogin['nama'].!' anda login sebagai '$userLogin['kategori']);
        $this->load->view('MEMBER/baseheader_member');
        $this->load->view('MEMBER/navigasi_member', $user);
        $this->load->view('MEMBER/v_dashboard', $data1);
        $this->load->view('MEMBER/basefooter_member');
    }
}

public function logout(){
    $this->session->unset_userdata('userLogin');
    redirect(base_url().'Login');
}
}
```

### D.3 Class Controller DaftarRekomendasi

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class DaftarRekomendasi extends CI_Controller {
    public function __construct () {
        parent :: __construct();
        $this->load->library('session');
        $this->load->helper('url');
        $this->load->model('m_daftarRekomendasi');
    }

    public function index()
    {
        $data['dataRekomendasi'] = $this->m_daftarRekomendasi-
        >view_rekomendasi();
        $data['dataBukanRekomendasi'] = $this->m_daftarRekomendasi-
        >view_bukan_rekomendasi();
        $this->load->view('ADMIN/baseheader_admin',$data);
        $this->load->view('ADMIN/navigasi_admin');
        $this->load->view('ADMIN/v_daftarrekomendasi',$data);
        $this->load->view('ADMIN/basefooter_admin',$data);
    }
}
```

```

function GetDrivingDistance($origin, $destination){
    $url =
    "https://maps.googleapis.com/maps/api/distancematrix/json?origins=" . $origin . "&destinations=" . $destination . "&mode=driving&language=pl-PL";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_PROXYPORT, 3128);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    $response = curl_exec($ch);
    curl_close($ch);
    $response_a = json_decode($response, true);
    $dist = $response_a['rows'][0]['elements'][0]['distance']['text'];
    $time = $response_a['rows'][0]['elements'][0]['duration']['text'];

    return array('distance' => $dist, 'time' => $time);
}

public function logout(){
    $this->session->unset_userdata('userLogin');
    redirect(base_url(). 'Login');
}
}

```

#### D.4 Class Controller FormPengepul

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class FormPengepul extends CI_Controller {
    public function __construct () {
        parent::__construct();
        $this->load->library('session');
        $this->load->helper('url');
        $this->load->model('m_pengepul');}

    public function index () {
        $this->load->view('ADMIN/baseheader_admin');
        $this->load->view('ADMIN/navigasi_admin');
        $this->load->view('ADMIN/v_formPengepul');
        $this->load->view('ADMIN/basefooter_admin'); }
}

```

```
function simpanMember() {
    $password= $this->input->post('password');
    $data = array(
        'id_aktor' => $this->input->post('id_aktor'),
        'username' => $this->input->post('username'),
        'password' => md5($password),
        'kategori' => 'Member'
    );
    $data1 = array(
        'id_aktor' => $this->input->post('id_aktor'),
        'nama' => $this->input->post('nama'),
        'alamat' => $this->input->post('alamat'),
        'kontak' => $this->input->post('kontak'),
        'status' => $this->input->post('status'),
        'latitude' => $this->input->post('latitude'),
        'longitude' => $this->input->post('longitude')
    );
    $this->m_pengumpul->insertMember($data, $data1);
    $this->load->view('ADMIN/baseheader_admin');
    $this->load->view('ADMIN/navigasi_admin');
    $this->load->view('ADMIN/v_formPengepul');
    $this->load->view('ADMIN/basefooter_admin');
}
function tampil_id($id_aktor){
    $data['single_member'] = $this->m_pengumpul->tampil_id($id_aktor);
    $this->load->view('ADMIN/baseheader_admin');
    $this->load->view('ADMIN/navigasi_admin');
    $this->load->view('ADMIN/v_formEdit', $data);
    $this->load->view('ADMIN/basefooter_admin'); }

function editMember(){
    $id_aktor = $this->input->post('id_aktor');
    $data = array(
        'id_aktor' => $this->input->post('id_aktor'),
        'username' => $this->input->post('username'),
        'password' => $this->input->post('password'),
        'kategori' => 'Member'
    );
    $data1 = array(
        'nama' => $this->input->post('nama'),
        'alamat' => $this->input->post('alamat'),
        'kontak' => $this->input->post('kontak'),
```



```
        'status' => $this->input->post('status'),
        'latitude' => $this->input->post('latitude'),
        'longitude' => $this->input->post('longitude')
    );
    $query = $this->m_pengumpul->editMember($id_aktor, $data,
    $data1);
    $this->load->view('ADMIN/baseheader_admin');
    $this->load->view('ADMIN/navigasi_admin');
    $this->load->view('ADMIN/v_formPengumpul');
    $this->load->view('ADMIN/basefooter_admin');
    }

    public function logout(){
        $this->session->unset_userdata('userLogin');
        redirect(base_url(). 'Login');
    }
}
```

#### D.5 Class Controller FormStok

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class FormStok extends CI_Controller {

    public function __construct () {
        parent::__construct();
        $this->load->library('session');
        $this->load->helper('url');
        $this->load->model('m_ikanMember');
    }

    public function index () {
        $userLogin = $this->session->userdata('userLogin');
        $user['single_member'] = $userLogin['id_aktor'];
        $this->load->view('MEMBER/baseheader_member');
        $this->load->view('MEMBER/navigasi_member', $user);
        $this->load->view('MEMBER/v_formStok');
        $this->load->view('MEMBER/basefooter_member');
    }

    public function simpanIkan(){
```

```
$data1 = array(
    'id_aktor' => $this->session->userdata('userLogin')['id_aktor'],
    'tanggal' => $_POST['tanggal'],
    'kuantitas' => array_sum($_POST['postBerat']),
    'status' => 'tersedia'
);
$this->m_ikanMember->insertIkan($data1);
$id = $this->db->insert_id();
$nm = $this->input->POST('postNama');
$data2 = array();
foreach ($nm as $key => $value) {
    $data2[] = array(
        'id_pax' => $id,
        'nama_ikan' => $_POST['postNama'][$key],
        'berat' => $_POST['postBerat'][$key]
    );
}

for($i = 0; $i < count($data2); $i++){
    $this->m_ikanMember->insertDetailIkan($data2[$i]);
}

$userLogin = $this->session->userdata('userLogin');
$user['single_member'] = $userLogin['id_aktor'];
$this->load->view('MEMBER/baseheader_member');
$this->load->view('MEMBER/navigasi_member', $user);
$this->load->view('MEMBER/v_formStok');
$this->load->view('MEMBER/basefooter_member');
}

function tampil_id($id_pax){
    $userLogin = $this->session->userdata('userLogin');
    $user['single_member'] = $userLogin['id_aktor'];
    $data['single_pax'] = $this->m_ikanMember->tampil_id($id_pax);
    $this->load->view('MEMBER/baseheader_member');
    $this->load->view('MEMBER/navigasi_member', $user);
    $this->load->view('MEMBER/v_formEdit', $data);
    $this->load->view('MEMBER/basefooter_member');
}

function editStok(){
    $id_pax = $this->input->post('id_pax');
    $data = array(
```

```

'id_pax'=>$this->input->post('id_pax'),
'tanggal'=>$this->input->post('tanggal'),
'kuantitas'=>$this->input->post('kuantitas'),
'status'=>$this->input->post('status')
);

$query = $this->m_ikanMember->editStok($id_pax, $data);
$userLogin = $this->session->userdata('userLogin');
$user['single_member'] = $userLogin['id_aktor'];
$this->load->view('MEMBER/baseheader_member');
$this->load->view('MEMBER/navigasi_member', $user);
$this->load->view('MEMBER/v_formStok');
$this->load->view('MEMBER/basefooter_member');
}

public function logout(){
    $this->session->unset_userdata('userLogin');
    redirect(base_url()).'Login';
}
}

```

#### D.6 Class Controller Profil

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Profil extends CI_Controller {
    public function __construct () {
        parent::__construct();
        $this->load->library('session');
        $this->load->helper('url');
        $this->load->model('m_pengepul');
    }

    function index($id_aktor){
        $userLogin = $this->session->userdata('userLogin');
        $user['single_member'] = $userLogin['id_aktor'];
        $data['single_aktor'] = $this->m_pengepul-
>tampil_pengepul($id_aktor)[0];
        $this->load->view('MEMBER/baseheader_member');
        $this->load->view('MEMBER/navigasi_member', $user);
        $this->load->view('MEMBER/v_profilMember', $data);
    }
}

```

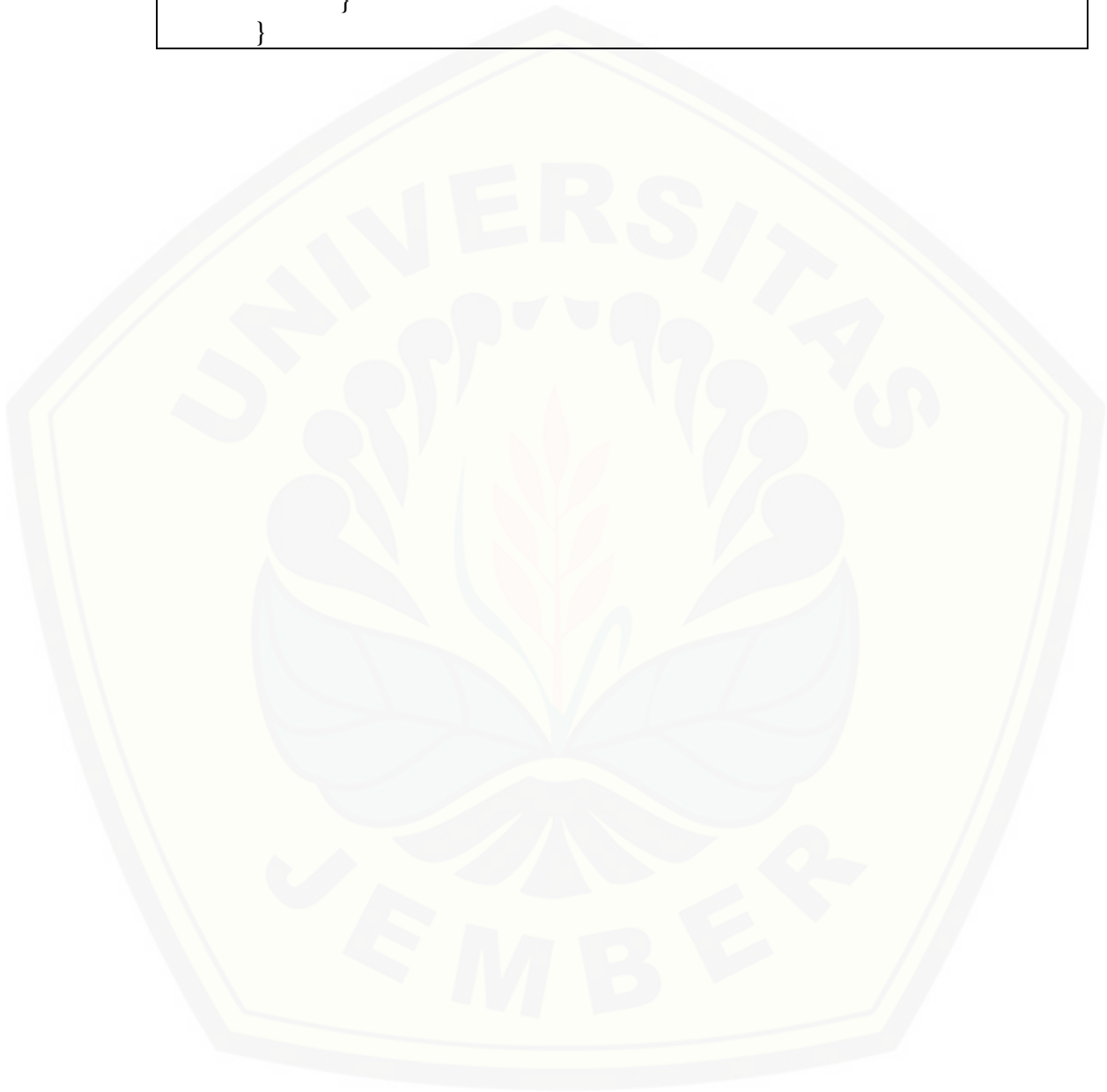
```
$this->load->view('MEMBER/basefooter_member');}

function password(){
    $userLogin = $this->session->userdata('userLogin');
    $user['single_member'] = $userLogin['id_aktor'];
    $this->load->view('MEMBER/baseheader_member');
    $this->load->view('MEMBER/navigasi_member', $user);
    $this->load->view('MEMBER/v_editPassword');
    $this->load->view('MEMBER/basefooter_member');
}

function editPass(){
    $userLogin = $this->session->userdata('userLogin');
    $user['single_member'] = $userLogin['id_aktor'];
    $pass_lama= $this->input->post('password_lama');
    $pass_baru= $this->input->post('password_baru');
    $kon_pass= $this->input->post('konfirmasi_password');
    $cekPassLama= $this->m_pengumpul->cekPassLama($this->session->userdata('userLogin')['username'],md5($pass_lama))->row_array();

    if(!empty($cekPassLama)){
        if($pass_baru === $kon_pass){
            $data = array(
                'password' => md5($pass_baru)
            );
            $query = $this->m_pengumpul->editPassword($userLogin['id_aktor'], $data);
            $this->session->set_flashdata('sukses', 'Password berhasil diperbarui');
            redirect(base_url('index.php/Profil/index/'. $userLogin['id_aktor']));
        }else {
            $this->session->set_flashdata('gagal', 'Password tidak cocok');
            redirect(base_url('index.php/Profil/password'));
        }
    }else {
        $this->session->set_flashdata('gagal', 'Password lama yang anda inputkan tidak sesuai');
        redirect(base_url('index.php/Profil/password'));
    }
}
}
```

```
public function logout(){  
    $this->session->unset_userdata('userLogin');  
    redirect(base_url().'.Login');  
}  
}
```





## LAMPIRAN E. PENGUJIAN BLACK BOX

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Home	Klik menu Home	Menampilkan halaman home yang berisi profil UD. Mitra Samudra dan deskripsi singkat sistem Fish Pick-Up Tracking	[√] Berhasil [ ] Gagal
2.	Fitur	Klik menu Fitur	Menampilkan halaman fitur yang berisi macam-macam fitur yang disediakan sistem	[√] Berhasil [ ] Gagal
3.	Kontak	Klik menu Kontak	Menampilkan halaman kontak yang berisi alamat dan kontak UD. Mitra Samudra	[√] Berhasil [ ] Gagal
4.	<i>Login</i>	Klik menu Login	Menampilkan halaman login yang berisi form username dan password	[√] Berhasil [ ] Gagal
		Mengisi kolom <i>username</i> dan <i>password</i> lalu klik	<i>Login</i> berhasil dan menampilkan halaman dashboard.	[√] Berhasil [ ] Gagal

		tombol <i>login</i>		
		Kolom <i>username</i> atau <i>password</i> tidak sesuai	Menampilkan flashdata gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5.	<i>Dashboard</i> Admin	Klik menu Dashboard	<ul style="list-style-type: none"> <li>- Menampilkan flashdata sukses</li> <li>- Menampilkan peta sebaran pengepul</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6.	Menentukan rute terpendek penjemputan ikan	Klik menu Daftar Rekomendasi	<ul style="list-style-type: none"> <li>- Menampilkan halaman daftar rekomendasi</li> <li>- Menampilkan rute terpendek penjemputan ikan berdasarkan daftar rekomendasi</li> <li>- Menampilkan tabel penunjuk arah</li> <li>- Menampilkan tabel rekomendasi</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7.	Mengelola data pengepul	Klik menu Daftar Pengepul	Menampilkan tabel daftar pengepul yang terdaftar sebagai member	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Klik tombol data baru	Menampilkan form pengepul	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol edit	Menampilkan form edit pengepul	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8.	Melihat data ikan pengepul	Klik menu Data Ikan Pengepul	<ul style="list-style-type: none"> <li>- Menampilkan halaman data ikan pengepul</li> <li>- Menampilkan tabel data ikan pengepul</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol detail	<ul style="list-style-type: none"> <li>- Menampilkan halaman detail data ikan</li> <li>- Menampilkan tabel detail data ikan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9.	Dashboard member	Klik menu dashboard	Menampilkan halaman dashboard member	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
10.	Mengelola Profil	Klik menu profil	Menampilkan halaman profil member	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Klik tombol edit password	Menampilkan halaman form edit password	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi kolom form dan klik simpan	<ul style="list-style-type: none"> <li>- Menampilkan flashdata sukses</li> <li>- Menampilkan halaman profil member</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi kolom yang tidak sesuai	Menampilkan flashdata gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
11.	Mengelola data ikan	Klik menu Data Stok Ikan	<ul style="list-style-type: none"> <li>- Menampilkan halaman data stok ikan</li> <li>- Menampilkan tabel data stok ikan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol detail	<ul style="list-style-type: none"> <li>- Menampilkan halaman detail data ikan</li> <li>- Menampilkan tabel detail data ikan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol data baru	<ul style="list-style-type: none"> <li>- Menampilkan halaman form stok</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		Klik tombol edit	Menampilkan halaman form edit stok	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
12.	Form Pengepul	Klik tombol data baru pada halaman Daftar Pengepul	Menampilkan halaman form pengepul	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi form dan klik simpan	Data berhasil ditambahkan dan menampilkan flashdata sukses	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi kolom yang tidak sesuai	Menampilkan flashdata gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
13.	Form edit pengepul	Klik tombol edit pada halaman Daftar Pengepul	Menampilkan halaman form pengepul	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
14.	Form Password	Klik tombol edit password pada halaman Profil	Menampilkan halaman form edit password	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal



15.	Form Stok	Klik tombol data baru pada halaman Data Stok Ikan	Menampilkan halaman form stok	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi kolom form stok	Menampilkan halaman data stok ikan dan menampilkan flashdata sukses	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Pengisian kolom tidak sesuai	Menampilkan flashdata gagal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
16.	Form Edit Stok	Klik tombol edit pada halaman Data Stok Ikan	Menampilkan halaman form edit stok	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal