



TUGAS AKHIR – KS141501

**PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN
DENGAN METODE RULE-BASED SYSTEM UNTUK
MENENTUKAN PRIORITAS PENGAMBILAN PESANAN
OLEH ANGKUTAN KOTA PADA APLIKASI ANGKOTIN**

**DEVELOP A DECISION SUPPORT SYSTEM USING RULE
BASED SYSTEM FOR DETERMINING PRIORITY OF
PUBLIC TRANSPORTATION ON THE ANGKOTIN
APPLICATION**

**ADITYA SEPTA SETYA BUDI
NRP 5214 100 126**

Dosen Pembimbing
Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D
Faizal Johan Atletiko, S.Kom, M.T

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR – KS141501

**PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN
DENGAN METODE RULE-BASED SYSTEM UNTUK
MENENTUKAN PRIORITAS PENGAMBILAN PESANAN
OLEH ANGKUTAN KOTA PADA APLIKASI ANGKOTIN**

**ADITYA SEPTA SETYA BUDI
NRP 5214 100 126**

Dosen Pembimbing

Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D

Faizal Johan Atletiko, S.Kom, M.T

Departemen Sistem Informasi

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018

Halaman ini sengaja dikosongkan

TUGAS AKHIR – KS141501

**DEVELOP A DECISION SUPPORT SYSTEM USING RULE
BASED SYSTEM FOR DETERMINING PRIORITY OF
PUBLIC TRANSPORTATION ON THE ANGKOTIN
APPLICATION**

ADITYA SEPTA SETYA BUDI

NRP 5214 100 126

Supervisor

Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D

Faizal Johan Atletiko, S.Kom, M.T

Information Systems Department

Faculty of Information Communication and Technology

Institute of Technology Sepuluh Nopember

Surabaya 2018

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

**PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN DENGAN
METODE RULE-BASED SYSTEM UNTUK MENENTUKAN
PRIORITAS PENGAMBILAN PESANAN OLEH ANGKUTAN
KOTA PADA APLIKASI ANGKOTIN**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember**

Oleh:

ADITYA SEPTA SETYA BUDI
NRP.5214100126

Surabaya, Januari 2018

**Plh Kepala
DEPARTEMEN SISTEM INFORMASI**



Edwin Riksakomara, S.Kom., M.T

NIP.196907252003121001

SISTEM INFORMASI

Halaman ini sengaja dikosongkan

LEMBAR PERSETUJUAN

**PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN DENGAN
METODE RULE-BASED SYSTEM UNTUK MENENTUKAN
PRIORITAS PENGAMBILAN PESANAN OLEH ANGKUTAN
KOTA PADA APLIKASI ANGKOTIN**

TUGAS AKHIR

**Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember**

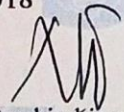
Oleh:

ADITYA SEPTA SETYA BUDI

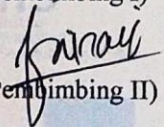
NRP. 05211440000126

**Disetujui Tim Penguji : Tanggal Ujian : 12 Januari 2018
Periode Wisuda : Maret 2018**

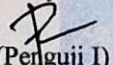
Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D


(Pembimbing I)

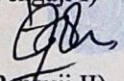
Faizal Johan Atletiko, S.Kom, M.T


(Pembimbing II)

Radityo Prasetyanto W., S.Kom, M.Kom


(Penguji I)

Irmasari Hafidz, S.Kom, M.Sc


(Penguji II)

Halaman ini sengaja dikosongkan

**PEMBUATAN SISTEM PENDUKUNG
KEPUTUSAN DENGAN METODE RULE-BASED
SYSTEM UNTUK MENENTUKAN PRIORITAS
PENGAMBILAN PESANAN OLEH ANGKUTAN
KOTA PADA APLIKASI ANGKOTIN**

Nama : Aditya Septa Setya Budi
NRP : 5214 100 126
Dosen : 1. Nur Aini Rakhmawati, S.Kom,
Pembimbing M.Sc.Eng, Ph.D
2. Faisal Johan Atletiko, S.Kom,
M.Kom
Lab : Akuisisi Data dan Desiminasi
Informasi (ADDI)

ABSTRAK

Untuk mengatasi kecenderungan penurunan penggunaan layanan transportasi angkutan kota di Kota Surabaya, dibuatlah sebuah sistem baru berbasis mobile bernama Angkotin yang memberikan alternatif untuk angkutan kota agar dapat digunakan sebagai jasa pengiriman barang. Namun dalam pengembangannya masih banyak terdapat masalah, seperti prosedur yang harus dilakukan ketika ingin melakukan pemesanan, pengambilan pesanan, serta pengiriman pesanan.

Dengan melihat permasalahan yang ada, dibuatlah sebuah sistem pendukung keputusan yang dapat membantu sopir angkutan kota dalam mengambil pesanan yang sesuai dengan kriteria masing – masing dari angkutan kota. Metode yang digunakan untuk mengembangkan sistem pendukung keputusan ini adalah sistem berbasis aturan (Rule Based System). Sistem ini mampu memberikan rekomendasi kepada sopir angkutan kota mengenai pesanan yang dapat diambil untuk dikirimkan ke tempat tujuan.

Hasil dari pengerjaan tugas akhir ini adalah sebuah sistem pendukung keputusan yang dapat membantu angkutan kota untuk menemukan pesanan yang dapat diambil berdasarkan empat faktor yang berpengaruh, yaitu jarak, arah perjalanan, kode trayek, dan status kapasitas penyimpanan angkutan kota. Dengan memperhatikan keempat faktor tersebut, sistem ini dapat menghasilkan rekomendasi pesanan pada kondisi yang sesuai melalui aplikasi Angkotin.

Kata kunci : Sistem Pendukung Keputusan, Penentuan Pesanan, Pengiriman Barang, Rule-Based System, Angkutan Kota Surabaya

**DEVELOP A DECISION SUPPORT SYSTEM
USING RULE BASED SYSTEM FOR
DETERMINING PRIORITY OF PUBLIC
TRANSPORTATION ON THE ANGKOTIN
APPLICATION**

Nama : Aditya Septa Setya Budi
NRP : 5214 100 126
Dosen : 1. Nur Aini Rakhmawati,
Pembimbing S.Kom, M.Sc.Eng, Ph.D
2. Faisal Johan Atletiko,
S.Kom, M.Kom
Lab : Akuisisi Data dan Desiminasi
Informasi (ADDI)

ABSTRACT

In overcoming of the negatif trend in utilization of urban transport (Angkot) services in Surabaya, a new mobile-based system called Angkotin provides an alternative to urban transport to be used as a freight service within the city. But in its development there are still many problems, such as procedures that must be done when making an order and picking up an orders.

By looking at the existing problems, a decision support system was developed that could help urban

transport drivers to take on a shipment orders. The method used to develop this decision support system is a rule-based system. This system is able to provide recommendations to the urban transport driver regarding orders that can be taken.

The result of this final project is a decision support system that can help public transportation to find orders that can be taken based on four factors, such as distance, direction, route code, and status of storage capacity. Based on these four factors, the system can provide an order recommendation under the appropriate conditions through the Angkotin application.

Keywords : Decision Support Systems, Dispatch Systems, Goods Shipment, Rule-Based System, Public Transport in Surabaya

KATA PENGANTAR

Puji dan syukur penulis tuturkan ke hadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan dan hidayah-Nya kepada penulis sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir dengan judul:

PEMBUATAN SISTEM PENDUKUNG KEPUTUSAN DENGAN METODE RULE-BASED SYSTEM UNTUK MENENTUKAN PRIORITAS PENGAMBILAN PESANAN OLEH ANGKUTAN KOTA PADA APLIKASI ANGKOTIN

yang merupakan salah satu syarat kelulusan pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang sudah melauangkan

waktu, tenaga dan pikirannya. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada:

1. Bapak Sigit Setyobudi dan Ibu Rini Purwokanti selaku kedua orang tua serta Putri Rigita Cahyani selaku saudara kandung dari penulis yang tiada henti memberikan dukungan dan semangat.
2. Ibu Nur Aini Rakhmawati, S.Kom, M.Sc.Eng selaku dosen wali serta dosen pembimbing dan Bapak Faisal johan Atletiko, S.Kom, M.Kom juga selaku dosen pembimbing yang senantiasa meluangkan waktu, memberikan ilmu dan petunjuk, serta memotivasi untuk kelancaran tugas akhir penulis.
3. Bapak Radityo Prasetianto W., S.Kom, dan ibu Irmasari Hafidz, S.Kom, M.Sc selaku dosen penguji yang telah memberikan saran dan kritik untuk perbaikan tugas akhir.
4. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang bermanfaat kepada penulis.
5. Hartantya Nia dan Andina Nur Damayanti selaku partner dalam mengerjakan tugas akhir ini dan

mengembangkan aplikasi Angkotin sehingga dapat mengikuti lomba Gemastik 2017.

6. Ahmad Choirun Najib, Azmi Adi Firmansyah, Faiz NFI, Hartantya Nia selaku partner dalam satu bimbingan yang selalu senantiasa bekerja bersama dan saling menyemangati dalam pengerjaan tugas akhir ini.
7. Ferdian Widyatama, Abdul Azis, Ahsanul Khuluq, dan Adam Ardiansyah yang selalu membantu dengan memberikan pemikiran untuk mengembangkan Aplikasi Angkotin dan pengerjaan tugas akhir ini
8. Teman – teman Wolfpack yang mau menjadi keluarga ketika merantau di Kota Surabaya dengan lika – liku kehidupan yang telah dijalani bersama.
9. Rekan – rekan yang tergabung dalam Laboratorium ADDI yang senantiasa membantu dan memberikan semangat dalam pengerjaan tugas akhir ini.
10. Rekan-rekan Osiris yang telah memberikan banyak pembelajaran, kenangan manis dan pahit semasa kuliah.

11. Berbagai pihak yang tidak bisa disebutkan satu persatu yang telah turut serta menyukseskan penulis dalam menyelesaikan tugas akhir.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 08 Januari 2018

Penulis,

Aditya Septa Setya Budi

DAFTAR ISI

LEMBAR PENGESAHAN...	Error! Bookmark not defined.
LEMBAR PERSETUJUAN..	Error! Bookmark not defined.
ABSTRAK.....	xi
ABSTRACT.....	xiii
KATA PENGANTAR	xv
DAFTAR ISI.....	xix
DAFTAR GAMBAR	xxiii
DAFTAR TABEL.....	xxv
DAFTAR KODE.....	xxvi
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian.....	5
1.6 Relevansi	6
2 BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Sebelumnya	7

2.2	Tinjauan Pustaka	13
2.2.1	Sistem Pendukung Keputusan.....	13
2.2.2	Komponen Sistem Pendukung Keputusan	14
2.2.3	Rule-Based System.....	16
2.2.4	CodeIgniter	19
2.2.5	Google Maps	21
2.2.6	Angkotin.....	23
3	BAB III METODOLOGI PENELITIAN	25
3.1	Tahapan Pelaksanaan Tugas Akhir	25
3.1.1	Identifikasi Masalah.....	26
3.1.2	Studi Literatur.....	26
3.1.3	Perancangan dan Pengembangan Aplikasi	27
3.1.4	Penyusunan Buku Tugas Akhir.....	33
3.2	Jadwal Penyusunan Tugas Akhir	34
3.1	BAB IV PERANCANGAN.....	37
4.1	Perancangan Penggalian Kebutuhan	37
3.1.1	Identifikasi Variabel	37
3.1.2	Identifikasi Kebutuhan Pengguna	41
4.2	Perancangan Desain.....	44

4.3	Perancangan Pengumpulan Data	44
4.3.1	Desain Database	45
4.3.2	Desain Model	48
4.3.3	Desain User Interface	53
4.3.4	Desain Arsitektur Sistem	55
4.4	Perancangan Pengujian.....	58
4.4.1	Pengujian Fungsional	58
4.4.2	Pengujian Model.....	59
4	BAB V IMPLEMENTASI	63
5.1	Lingkungan Implementasi	63
5.2	Pengembangan Sistem.....	64
5.2.1	Pembuatan Halaman Pemesanan	64
5.2.2	Pembuatan Rute Rencana Perjalanan.....	69
5.2.3	Pembuatan Model <i>Rule Based</i>	71
5.2.4	Halaman Pick Up Barang	78
5.2.5	Halaman Muatan	80
5.2.6	Halaman Pelacakan Pengiriman	81
5	BAB VI HASIL DAN PEMBAHASAN	83
6.1	Hasil	83

6.1.1	Pengujian Fungsional.....	83
6.1.2	Pengujian Model.....	87
6.2	Pembahasan	113
6.2.1.	Analisis Hasil Black Box Testing	113
6.2.2.	Analisis Hasil Luaran Model	114
6	BAB VII KESIMPULAN DAN SARAN	115
7.1	Kesimpulan	115
7.2	Saran	117
	DAFTAR PUSTAKA	119

DAFTAR GAMBAR

Gambar 2.1 Logo Google Maps	22
Gambar 2.2 Logo dan Tampilan Aplikasi Angkotin.....	23
Gambar 3.1 Skema Metodologi Penelitian	25
Gambar 3.2 Metode Pengembangan Aplikasi	28
Gambar 4.1 Ilustrasi Penghitungan untuk Menentukan Arah 40	
Gambar 4.2 Use Case Diagram untuk Pos.....	42
Gambar 4.3 Use Case Diagram untuk Driver	43
Gambar 4.4 Skema Relasi Antar Tabel	48
Gambar 4.5 Flowchart Model Rule Based	51
Gambar 4.6 Ilustrasi Pencarian Pesanan pada Aplikasi Angkotin	52
Gambar 4.7 Arsitektur Sistem Aplikasi Angkotin	56
Gambar 4.8 Komponen <i>Fuzzy Logic</i>	58
Gambar 5.1 Halaman Utama dan Pemesanan.....	69
Gambar 5.2 Tampilan Halaman Muatan	80
Gambar 5.3 Tampilan Halaman Tracking Barang	81
Gambar 6.1 Ilustrasi Test Case 1	91
Gambar 6.2 Ilustrasi Test Case 2.....	93
Gambar 6.3 Ilustrasi Test Case 3.....	94
Gambar 6.4 Ilustrasi Test Case 4.....	95
Gambar 6.5 Ilustrasi Test Case 5.....	97
Gambar 6.6 Ilustrasi Test Case 6.....	98

Gambar 6.7 Ilustrasi Test Case 7.....	100
Gambar 6.8 Ilustrasi Test Case 8.....	102
Gambar 6.9 Ilustrasi Test Case 9.....	103
Gambar 6.10 Ilustrasi Test Case 10.....	105
Gambar 6.11 Ilustrasi Test Case 11	106
Gambar 6.12 Ilustrasi Test Case 12.....	108
Gambar 6.13 Ilustrasi Test Case 13.....	110
Gambar 6.14 Ilustrasi Test Case 14.....	111

DAFTAR TABEL

Tabel 2.1 Tabel Literatur 1	7
Tabel 2.2 Tabel Literatur 2	9
Tabel 2.3 Tabel Literatur 3	11
Tabel 3.1 Jadwal Pelaksanaan Tugas Akhir	35
Tabel 4.1 Penjelasan Tabel	46
Tabel 4.2 Rencana Pembuatan <i>User Interface</i>	53
Tabel 4.3 Daftar Rencana Pengujian Fungsional	59
Tabel 4.4 Daftar Rencana Skenario Pengujian	61
Tabel 5.1 Spesifikasi Lingkungan Implementasi	63
Tabel 5.2. Lingkungan teknologi yang digunakan	64
Tabel 5.3 <i>Query</i> untuk Mengambil Data Pos	67
Tabel 5.4 Query untu Mengambil Data Pesanan dan Posisi Driver	73

DAFTAR KODE

Kode 5.1 Inisiasi Pembuatan Peta	65
Kode 5.2 Menampilkan Pos Dalam Bentuk Marker	66
Kode 5.3 Menampilkan Infowindow yang Berisi Tombol.....	68
Kode 5.4 Pencarian Rute Perjalanan	71
Kode 5.5 Pengambilan Data Pesanan dan Posisi <i>Driver</i>	72
Kode 5.6 Penghitungan Jarak Antar Titik.....	75
Kode 5.7 Implementasi Model dengan Metode <i>Rule Based</i> ..	76
Kode 5.8 Pengambilan Barang oleh Driver	79

BAB I

PENDAHULUAN

Dalam bab ini akan dijelaskan gambaran umum mengenai tugas akhir yang diangkat. Hal tersebut meliputi latar belakang masalah, perumusan masalah, batasan tugas akhir, tujuan tugas akhir, dan relevan atau manfaat kegiatan tugas akhir. Selain itu, akan dijelaskan pula sistematika penulisan tugas akhir untuk memudahkan pembaca pada saat membaca buku tugas akhir ini.

1.1 Latar Belakang

Angkutan kota atau yang sering disebut Angkot merupakan salah satu sarana transportasi umum yang memiliki biaya paling murah. Namun hal itu tidak menjamin layanan transportasi ini dapat diminati oleh masyarakat, khususnya masyarakat Kota Surabaya. Salah satu permasalahan yang sering dikeluhkan adalah terkait lamanya waktu menunggu penumpang. Hal ini diperparah dengan menurunnya jumlah penumpang yang membuat waktu menunggu sebuah angkot untuk penuh menjadi semakin lama. Sehingga tingkat produktifitas sebuah angkot saat ini menjadi sangat rendah.

Dalam menghadapi permasalahan tersebut, telah terdapat solusi yang ditawarkan yaitu dengan mengoptimalkan fungsionalitas dari angkutan kota sebagai jasa pengiriman barang. Hal ini bertujuan untuk meningkatkan produktifitas angkot dengan mengurangi waktu pemberhentian guna menunggu penumpang. Solusi ini telah didukung dengan sistem yang lebih moder yaitu menggunakan aplikasi berbasis *mobile* yang bernama “Angkotin”. Kedepannya dengan sistem yang baru, angkutan kota tidak hanya mengantarkan penumpang melainkan juga dapat mengantarkan barang.

Namun masih terdapat banyak kelemahan yang dimiliki aplikasi Angkotin ini. Salah satunya adalah perlu adanya prosedur yang digunakan dalam mengatur proses pengambilan pesanan dan pengiriman barang. Hal ini untuk menghindari terjadinya tumpang tindih dalam pengambilan pesanan dan berbagai kesalahan yang mungkin terjadi dalam pengiriman barang. Sehingga permasalahan tersebut dijadikan sebagai latar belakang dibuatnya tugas akhir ini. Tujuannya adalah untuk menentukan pesanan yang dapat diambil dan diantarkan oleh angkutan kota yang sesuai dengan kriteria pengiriman

barang. Metode yang digunakan dalam penentuan ini adalah dengan menggunakan *Rule-Based System*. Metode ini memperhitungkan beberapa aturan yang digunakan untuk menemukan sebuah keputusan berdasarkan fakta yang telah diperoleh[1].

Luaran dari pengerjaan tugas akhir ini adalah daftar faktor yang mempengaruhi penentuan pesanan dan sistem penentuan pesanan yang dapat membantu angkutan kota untuk menentukan pesanan yang dapat diambil. Sehingga diharapkan luaran tersebut dapat membantu dalam pengembangan aplikasi Angkotin sebagai solusi pelestarian angkutan kota di Kota Surabaya.

1.2 Rumusan Masalah

Rumusan masalah yang akan diselesaikan dalam tugas akhir ini adalah:

- a. Bagaimana menentukan variabel dan pembuatan model menggunakan *Rule Based System*?
- b. Bagaimana hasil luaran dari model *Rule Based System* terhadap penentuan pesanan?

1.3 Batasan Masalah

Batasan masalah terkait pengerjaan tugas akhir ini adalah:

- a. Rute trayek angkutan kota yang digunakan dalam penelitian ini hanya terbatas pada rute angkot yang terdapat di Kota Surabaya.
- b. Proses bisnis yang digunakan dalam penentuan angkot dalam pengambilan pesanan telah ditentukan dalam proses bisnis aplikasi Angkotin.
- c. Data trayek angkutan kota yang digunakan diperoleh dari data trayek Angkutan Kota Surabaya yang terdapat pada Google Maps.
- d. Data terkait perjalanan angkutan kota berdasarkan keluaran kalkulasi dari Google Maps.
- e. Peta geografis Kota Surabaya yang digunakan berdasarkan Google Maps.
- f. Data pelanggan, pos, dan sopir angkutan kota tidak didasarkan pada data sebenarnya, melainkan data dummy yang digunakan untuk mendukung pengerjaan tugas akhir.

1.4 ujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah:

- a. Mengetahui variabel dan model *Decission Support System* yang digunakan dalam penentuan angkutan kota.
- b. Memudahkan identifikasi pesanan yang dapat diambil oleh angkutan kota pada aplikasi Angkotin.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dapat dihasilkan dari pengerjaan tugas akhir ini adalah :

Bagi Peneliti

Penelitian ini dapat memberikan pengetahuan mengenai pengambilan keputusan menggunakan metode *Rule Based System*.

Bagi Pengembang Aplikasi Angkotin

Membantu mengembangkan modul penentuan angkot mana yang dapat mengambil pesanan. Karena hal ini menjadi salah satu permasalahan yang dihadapi oleh pengembang aplikasi Angkotin. Sehingga dengan adanya modul ini diharapkan dapat mempermudah

pengembangan aplikasi dan kelancaran proses bisnis Angkotin.

1.6 Relevansi

Tugas Akhir ini memiliki relevansi terhadap mata kuliah yang terdapat pada laboratorium ADDI, karena melakukan implementasi mata kuliah pada bidang keilmuan Sistem Pendukung Keputusan.

Selain itu tugas akhir ini juga bersifat untuk menyelesaikan masalah dalam pengembangan aplikasi Angkotin terkait proses penentuan agen angkot mana yang akan mengambil pesanan menggunakan aturan – aturan yang telah ditentukan.

BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini. Landasan teori akan memberikan gambaran secara umum dari landasan penjabaran tugas akhir ini.

2.1 Penelitian Sebelumnya

Berikut ini penelitian-penelitian sebelumnya yang dijadikan acuan dalam pengerjaan tugas akhir yang disajikan dalam tabel di bawah ini.

Tabel 2.1 Tabel Literatur 1

Judul	<i>Allocation of Location-Based Orders To Mobile Agents</i> (2014)
Pengarang	Cian E O'Meara, Michael Brosnan, Paul Kelly, Derek Hyland, Damian Killeen
Kesimpulan	Didalam penelitian ini dijelaskan mengenai proses yang dilakukan dalam pengalokasian agen taksi ketika terdapat pesanan yang berada pada lokasi tertentu, tujuannya adalah untuk menentukan agen mana yang memiliki prioritas paling

	<p>tinggi untuk mengambil pesanan. Kemudian disimpulkan bahwa terdapat lima tahap yang diperlukan untuk melakukan penentuan tersebut yaitu :</p> <ul style="list-style-type: none">a) Menyimpan riwayat pesanan saat ini yang mengidentifikasi kondisi yang menentukan lokasi dan waktu dimana setiap agen bebas dan dapat menerima pesanan baru;b) Menyimpan daftar lokasi, dimana setiap lokasi dalam daftar diprioritaskan untuk agen sesuai dengan ketersediaan agen dalam mencapai lokasi tersebut setelah agen bebas dari pesanan;c) Menerima pesanan baru dan mencatat lokasi dan waktu di mana pesanan ini harus dipenuhi;d) Memutuskan, berdasarkan daftar lokasi yang diprioritaskan, agen mana yang paling sesuai untuk menerima orde baru; dan
--	---

	e) Mengalokasikan pesanan baru ke agen yang teridentifikasi.
Relevansi	Proses pengalokasian agen dalam penelitian ini dijadikan rujukan dalam proses penentuan angkutan kota karena memiliki kemiripan proses bisnis.
Sumber	[2]

Tabel 2.2 Tabel Literatur 2

Judul	Sistem Berbasis Aturan Untuk Mendiagnosa Penyakit Flu Burung Secara <i>Online</i> (2006)
Pengarang	Titik Lusiani, Andhika kurniawan Cahyono
Kesimpulan	Didalam penelitian ini dijelaskan mengenai proses yang dilakukan dalam membuat sebuah sistem pakar yang dapat menghasilkan kesimpulan berdasarkan data yang dimasukkan. Metode yang digunakan pada penelitian ini sama dengan metode pengerjaan tugas akhir

	<p>ini. Sehingga proses yang terdapat pada penelitian ini dapat dijadikan acuan dalam pengerjaan tugas akhir.</p> <p>Pada proses sistem berbasis aturan untuk diagnosa penyakit flu burung secara online ini dihasilkan kesimpulan sebagai berikut :</p> <ul style="list-style-type: none">a) Sistem ini dapat melakukan validasi parameter dan proses Verifikasi, sehingga hasil desain aturan penyakit flu burung terbebas dari kesalahan penentuan parameter, Redundant Rules, Conflicting Rules, Circular Rules, Subsumed.b) Dengan treeview langkah untuk mengubah Decision Tree atau Dependency Diagram menjadi tidak diperlukan karena Decision Tree atau Dependency Diagram dapat langsung diaplikasikan.c) Berdasarkan dari hasil uji coba yang telah dilakukan, setelah melakukan
--	---

	diagnosa penyakit maka didapatkan hasil berupa jenis penyakit flu burung berdasarkan input yang telah ditentukan sebelumnya oleh pakar.
Relevansi	Metode yang digunakan dalam penelitian ini selaras dengan metode yang digunakan dalam pembuatan tugas akhir yaitu <i>Forward Chain Rule-Based System</i> .
Sumber	[3]

Tabel 2.3 Tabel Literatur 3

Judul	A decision support system and rule-based algorithm to augment the human interpretation of the 12-lead electrocardiogram (2017)
Pengarang	Andrew W. Cairns, BSc, dkk
Kesimpulan	Kesimpulan yang didapatkan dari penelitian ini yaitu sebanyak 375 interpretasi dikumpulkan. Pendekatan IPI + DDA ditunjukkan untuk

	<p>memperbaiki akurasi diagnostik sebesar 8,7%. IPI + DDA menyarankan interpretasi benar lebih sering daripada penafsiran manusia dalam 7/10 kasus. Ketepatan interpretasi manusia meningkat menjadi 70% ketika tujuh saran dihasilkan.</p>
Relevansi	<p>Penelitian ini menunjukkan bahwa metode <i>Rule Based System</i> bisa digunakan dalam sistem pendukung keputusan. Penelitian ini dilakukan pada tahun 2017 yang menandakan bahwa metode <i>Rule Based System</i> masih layak digunakan dalam penelitian.</p>
Sumber	[4]

2.2 Tinjauan Pustaka

Berikut ini dijabarkan dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir, yaitu:

2.2.1 Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan pemanipulasian data. Sistem ini digunakan untuk membantu pengambil keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat[5].

Menurut Keen (dalam Turban, 2005), Sistem pendukung keputusan adalah sistem berbasis komputer yang dibangun lewat sebuah proses adaptif dari pembelajaran, pola-pola penggunaan dan evolusi sistem[6].

Berdasarkan kedua pengertian tersebut dapat disimpulkan bahwa sistem pendukung keputusan merupakan sebuah sistem yang mampu memberikan kemampuan pemecahan masalah yang bersifat semi terstruktur dan tidak terstruktur. Selain itu SPK bukanlah sebuah alat yang digunakan sebagai pengambil keputusan,

melainkan sebuah sistem yang dapat membantu dalam menentukan keputusan mana yang akan diambil.

Penggunaan sistem pendukung keputusan dalam pengerjaan tugas akhir ini tidak terlalu kompleks, namun lebih cenderung untuk menentukan keputusan dengan cepat. Pengambilan keputusan ini bertujuan untuk menentukan pesanan mana yang dapat diambil oleh seorang *driver* angkot. Sehingga luaran dari sistem ini berupa daftar pesanan yang dapat diambil oleh *driver*.

2.2.2 Komponen Sistem Pendukung Keputusan

Menurut Turban (2011), sebuah SPK dapat terdiri dari empat buah komponen, yaitu[6]:

a. Subsistem Manajemen Data

Termasuk basis data yang berisi data-data relevan untuk situasi yang terjadi dan dikelola dalam sebuah piranti lunak yang disebut database management system (DBMS). Subsistem ini adalah bagian yang menangani semua penyimpanan maupun pengelolaan data dalam SPK.

b. Subsistem Manajemen Model

Subsistem Manajemen Model adalah sebuah paket piranti lunak yang meliputi model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang menyediakan kemampuan analitis bagi sistem dan manajemen piranti lunak yang layak. Piranti lunaknya sering disebut *model database management system* (MBMS).

c. Subsistem Antarmuka

Subsistem antarmuka berfungsi sebagai penghubung pengguna dengan sistem. Pengguna dapat berkomunikasi dan memberi perintah pada sistem dengan menggunakan komponen-komponen yang disediakan pada antarmuka.

d. Subsistem Manajemen Berbasis Pengetahuan

Subsistem ini dapat berdiri sebagai komponen sendiri atau mendukung komponen lain. Fungsinya adalah untuk menyediakan intelijen untuk kepentingan sang pengambil keputusan.

2.2.3 Rule-Based System

Rule Based System atau dikenal sebagai production system atau expert system merupakan menggunakan aturan untuk merepresentasikan ilmu atau pengetahuannya dimana pengetahuan tersebut akan dibuat pengkodean dalam sebuah sistem. Sekumpulan aturan tersebut dibuat dengan untuk menciptakan apa yang harus dilakukan dan atau menyimpulkan permasalahan yang ada. Rule based system merupakan sistem yang dibuat dengan menggunakan kondisi IF-THEN (biasanya disebut IF-THEN rules) dimana IF P then Q akan ekuivalen dengan $P \Rightarrow Q$ [1]. Syarat dari penerapan rule based system adalah kondisi atau pengetahuan yang akan diimplementasikan dapat ditulis dengan IF-THEN rules. Cakupan permasalahan yang akan digunakna tidak harus sangat besar karena banyaknya aturan yang akan dimasukkan maka sistem pakar tersebut tidak akan berjalan efektif.

Adapun beberapa elemen rule based system diantaranya [1]:

1. Kumpulan fakta (*a set of fact*). Beberapa fakta ini harus relevan dengan keadaan awal sistem.

2. Kumpulan aturan (*a set of rules*). Hal ini mengandung sekumpulan aturan yang dijalankan dalam suatu ruang lingkup permasalahan bagaimana action tersebut dituntut untuk melakukan sesuatu. Aturan yang berkaitan dengan fakta terletak pada bagian IF, sedangkan action yang akan dilakukan terletak pada bagian THEN.
3. Kriteria penghentian (*a terminate criterion*). Suatu action dikatakan berhenti apabila sudah mendapatkan hasil atau tidak ada hasil sama sekali. Hal ini penting untuk menentukan untuk mengetahui letak perulangan dan pemberhentian action yang dilakukan.

Rule based system dapat diterapkan dengan aturan sederhana sebagai berikut.

IF condition1

AND condition2

AND condition 3

...

THEN action1, action2, action3, ...

Beberapa kondisi yang menyatakan sebuah pilihan dapat diterapkan dengan aturan sederhana sebagai berikut.

IF condition
AND condition2
OR condition3
...
THEN action1, action2, action3, ...

Terdapat dua pendekatan yang digunakan dalam penerapan rule-based system antara lain [7]:

1. Forward Chaining. Pendekatan ini disebut juga data driven. Sistem dimulai dengan inisiasi nilai yang terdapat pada memori, lalu menjalankan rules atau aturan yang sudah dimasukkan hingga tujuan tercapai. Strategi ini cocok digunakan untuk data yang berskala besar.
2. Backward Chaining. Pendekatan ini disebut juga driven strategy. Sudah ditentukan targetnya, kemudian aturan yang aksinya mengandung target di-trigger. Backward chaining ini cocok untuk menelusuri fakta yang masih belum lengkap, disebut juga goal driven.

2.2.4 CodeIgniter

CodeIgniter merupakan sebuah *framework* aplikasi berbasis web yang *open source* untuk bahasa pemrograman PHP. CodeIgniter menggunakan pola desain *Model-View-Controller* (MVC), yang merupakan cara untuk mengatur aplikasi web kedalam tiga bagian yang menjadikannya lebih terstruktur[8].

CodeIgniter memiliki banyak fitur yang membuatnya berbeda dengan *framework* lainnya. Selain itu, dokumentasi dari *framework* CodeIgniter ini sangat lengkap yang mencakup seluruh aspek dalam *framework*. CodeIgniter memiliki desain dan struktur file yang sederhana, didukung dengan dokumentasi yang lengkap sehingga *framework* ini lebih mudah dipelajari. Penggunaan *framework* mempermudah dan mempercepat pembuatan program tanpa harus membuat fungsi atau class dari awal.

Fitur pada CodeIgniter yang menjadi kelebihan menggunakan *framework* CodeIgniter adalah[9]:

- a. Menggunakan konsep Model-View-Controller agar menjadi terstruktur dan memiliki standar yang jelas.

- b. URL *friendly*, dengan meminimalisasi penggunaan \$_GET dan digantikan dengan URI
- c. Kemudahan dalam mempelajari, membuat, memodifikasi dan mengintegrasikan library dan helper.

CodeIgniter dibangun dengan konsep MVC (Model-View-Controller) yang merupakan sebuah teknik pemrograman yang memisahkan *business logic* (alur piker), *data logic* (penyimpanan data), dan *presentation logic* (antarmuka aplikasi). Berikut ini adalah penjelasan konsep MVC[9].

1. Model

Model berkaitan dengan interaksi ke sebuah database seperti pengambilan data, membaca data, mengupdate data, atau menghapus data. Pada bagian Model akan berhubungan langsung dengan perintah *query* SQL.

2. View

View berkaitan dengan yang akan ditampilkan ke *end user*. View berisi halaman website yang dibuat menggunakan HTML dengan bantuan CSS. Pada View hanya khusus menampilkan data hasil dari

Model dan Controller sehingga tidak ada kode untuk melakukan koneksi ke database.

3. Controller

Controller merupakan penghubung antara Model dan View. Pada Controller terdapat class maupun fungsi yang memproses permintaan dari View kedalam struktur data di dalam Model. Tugas Controller adalah menyediakan variable yang akan ditampilkan di View, memanggil Model untuk melakukan perintah ke database, menjalankan proses logika, dll.

2.2.5 Google Maps

Google Maps diluncurkan pada tahun 2005 dan telah merevolusi aplikasi layanan peta online di internet. Berdasarkan Asynchronous JavaScript and XML (AJAX) Google Maps berhasil memperkenalkan jenis interaksi *client / server* baru untuk menjaga hubungan antara klien dan server secara berkelanjutan sehingga informasi pada peta yang disediakan selalu diperbaharui[10].

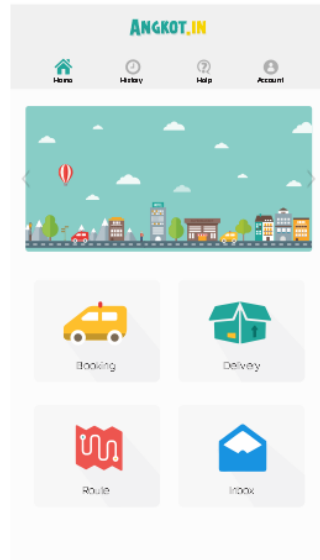


Gambar 2.1 Logo Google Maps

Selain menerapkan interaksi *client / server* yang lebih baik, Google juga menyediakan akses secara gratis bagi *programmer* untuk mengakses kode dalam bentuk Application Programming Interface (API). API terdiri dari satu set rutinitas atau fungsi yang dapat dipanggil oleh *programmer* menggunakan JavaScript, php, atau bahasa pemrograman lainnya. Google API dengan cepat diadopsi, bahkan oleh komunitas open source[10].

API dari Google Maps ini yang digunakan untuk mengembangkan aplikasi Angkotin, terutama untuk pengembangan modul – modul terkait pemilihan rute pengiriman dan modul penentuan angkutan kota. Bagian yang diambil dari API tersebut diantara lain *location*, *direction*, dan *distance*.

2.2.6 Angkotin



Gambar 2.2 Logo dan Tampilan Aplikasi Angkotin

Angkotin merupakan sebuah aplikasi pengiriman barang dengan menggunakan jasa angkutan kota di Kota Surabaya. Dengan Angkotin pengguna dapat mengirimkan barang ke berbagai daerah di Kota Surabaya melalui pos – pos yang telah pengiriman. Aplikasi ini memungkinkan pengguna untuk melakukan pengiriman barang dengan melalui perantara aplikasi. Selanjutnya barang akan diambil oleh angkutan kota untuk dikirimkan ke pos terdekat dengan lokasi tujuan pengiriman.

Selain modul pengiriman barang, pada aplikasi Angkotin juga terdapat modul penentuan harga, modul penentuan angkutan kota, dan modul *tracking* pengiriman.

Keakuratan dalam menentukan angkutan kota mana yang akan mengambil pesanan merupakan salah satu hal yang cukup krusial. Karena jika terjadi kesalahan dalam penentuan angkutan kota dapat mengakibatkan kesalahan pengiriman barang. Maka dari itu dalam tugas akhir ini akan dikembangkanlah modul penentuan angkutan kota dalam pengambilan pesanan pada aplikasi Angkotin.

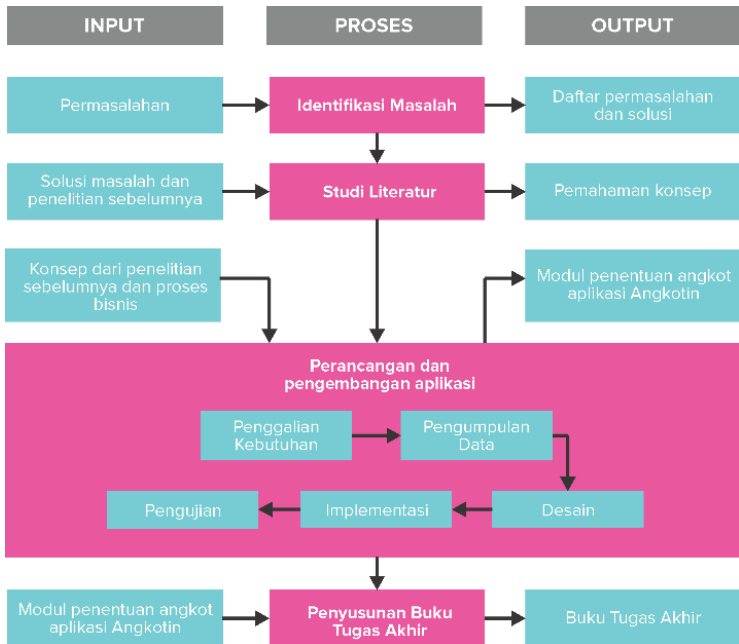
BAB III

METODOLOGI PENELITIAN

Pada bab metode penelitian akan diurutkan tahapan-tahapan apa saja yang dilakukan dalam menyelesaikan penelitian tugas akhir ini.

3.1 Tahapan Pelaksanaan Tugas Akhir

Pada sub bab ini akan dijelaskan mengenai metodologi dalam pelaksanaan tugas akhir.



Gambar 3.1 Skema Metodologi Penelitian

3.1.1 Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah terkait studi kasus, yaitu penentuan angkutan kota dalam pengambilan pesanan. Hal pertama yang dilakukan adalah mempelajari dan menganalisa proses bisnis yang terdapat pada aplikasi Angkotin, terutama yang berkaitan dengan modul penentuan angkutan kota.

Berdasarkan proses bisnis yang terdapat pada aplikasi Angkotin, apabila terdapat lebih dari satu kendaraan angkutan kota dalam satu trayek, maka dapat berpotensi menimbulkan tumpang tindih dalam pengambilan pesanan. Dan jika hal tersebut terjadi maka dapat mengakibatkan kesalahan pengiriman barang.

Oleh karena itu perlu adanya sebuah aturan yang dapat menentukan dan memprioritaskan angkutan kota mana yang dapat mengambil sebuah pesanan.

3.1.2 Studi Literatur

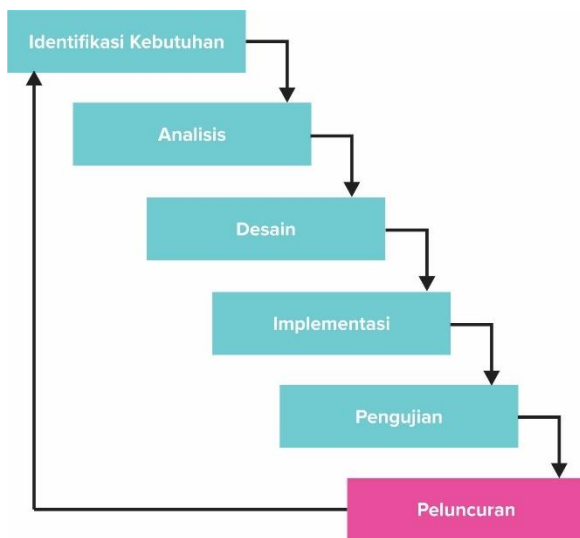
Studi literatur dilakukan dengan mengumpulkan berbagai referensi seperti buku pustaka, penelitian sebelumnya, dan dokumen terkait dalam mendukung pengerjaan tugas akhir. Studi literatur ini didasarkan pada latar belakang permasalahan yaitu mengenai penentuan

agen angkot mana yang dapat mengambil pesanan dengan memperhatikan berbagai batasan.

Untuk memperdalam solusi yang akan digunakan untuk menyelesaikan permasalahan, diperlukan studi mengenai penelitian sebelumnya yang terkait pengalokasian agen dalam mengambil pesanan pada transportasi lain seperti taxi. Selain itu juga perlu adanya studi literatur terkait metode yang digunakan dalam penentuan agen.

3.1.3 Perancangan dan Pengembangan Aplikasi

Pada tahap ini dilakukan perancangan dan pengembangan aplikasi dimulai dari analisis kebutuhan pembuatan aplikasi, mendesain aplikasi, implementasi koding, sampai pengujian aplikasi. Pembuatan aplikasi ini menerapkan metode *Waterfall* dimana proses pengembangan aplikasi dikerjakan fase ke fase. Dengan metode ini urutan proses pengerjaan menggunakan metode ini menjadi lebih teratur dari satu tahap ke tahap yang selanjutnya.



Gambar 3.2 Metode Pengembangan Aplikasi

Namun jika terdapat kekuarangan proses atau prosedur dari tahapan sebelumnya, maka tahapan pengembangan harus dilakukan mulai dari awal. Hal ini akan memakan waktu yang cukup lama.

3.1.3.1 Identifikasi Kebutuhan

Setelah mengetahui konsep dari penelitian sebelumnya serta proses bisnis yang telah ditentukan, maka tahapan selanjutnya adalah menganalisa kebutuhan apa saja yang diperlukan dalam pengembangan aplikasi. Kebutuhan disini dapat dibedakan berdasarkan pengguna

dari aplikasi seperti pelanggan, pos, dan *driver*. Selain ketiga pengguna tersebut, kebutuhan dapat dilihat dari sisi sistem itu sendiri. Karena dalam membuat sebuah sistem memerlukan modul – modul pendukung agar sesuai dengan proses bisnis yang telah ditentukan, seperti modul pengiriman barang, modul penentuan harga, modul penentuan angkutan kota, dan modul *tracking* pengiriman.

Karena pada pengerjaan tugas akhir ini lebih fokus mengerjakan modul penentuan angkutan kota, maka diperlukan identifikasi kebutuhan terkait penentuan angkutan kota. Berdasarkan pengamatan terhadap proses bisnis Angkotin, maka untuk menentukan angkutan kota mana yang dapat mengambil pesanan ditentukan berdasarkan beberapa variabel, seperti :

1. Kode Trayek

Kode trayek merupakan kode yang menentukan rute perjalanan sebuah angkutan kota. Dalam penentuan pengambilan pesanan, kode trayek angkutan kota disesuaikan dengan jalur pengiriman barang. Apabila kode trayek sesuai maka angkutan kota tersebut memenuhi satu aturan untuk dapat mengambil pesanan.

2. Jarak

Merupakan jarak antara angkutan kota dengan lokasi pemesan. Dalam penentuan angkutan kota, perlu didefinisikan jarak maksimal untuk menyaring angkutan kota mana yang dapat memenuhi aturan tersebut. Misal jarak maksimal antara angkutan kota dengan lokasi pemesan adalah sejauh 1 kilo meter.

3. Kapasitas Angkutan Kota

Merupakan kondisi yang menyatakan status kapasitas dari angkot, dimana terdapat dua status yaitu penuh dan tidak penuh. Jika kondisi angkutan kota masih memiliki tempat kosong, maka kendaraan tersebut memenuhi satu aturan untuk dapat mengambil pesanan.

4. Arah Perjalanan

Menggambarkan arah perjalanan angkutan kota dibandingkan dengan arah perjalanan dari pengiriman barang.

3.1.3.2 Desain

Berdasarkan hasil analisis kebutuhan pada tahap sebelumnya akan digunakan untuk membuat gambaran desain dari fitur dan fungsionalitas terhadap aplikasi. Pada tahap ini akan dibuat beberapa desain secara bertahap antara lain:

1. Desain Database

Pada tahap ini akan dilakukan perancangan mengenai bagaimana data disimpan dan menentukan atribut apa saja yang dibutuhkan.

2. Desain Model

Pada tahap ini akan dilakukan perancangan alur dari model berdasarkan metode Rule Based dengan menggunakan variabel yang telah ditentukan

3. Desain User Interface

Pada tahap ini akan dilakukan perancangan mengenai tampilan apa saja yang dibutuhkan untuk mendukung fungsional aplikasi.

4. Desain Arsitektur Sistem

Pada sub bab ini akan dijelaskan mengenai arsitektur sistem yang akan dibuat beserta komponennya.

3.1.3.3 Implementasi

Proses ini merupakan proses yang paling mendominasi pada tahapan konstruksi. Setelah melalui proses desain, berikutnya dilakukan proses implementasi atau *coding* terhadap model yang telah dirancang dan mengintegrasikannya dengan aplikasi Angkotin yang sudah ada. Selain itu juga dilakukan pembuatan tampilan antar muka dari aplikasi sesuai dengan kebutuhan fungsionalitas.

3.1.3.4 Pengujian

Pada tahap ini akan dilakukan pengujian aplikasi dalam rangka memastikan proses pengkodean yang telah dilakukan dapat berjalan dengan baik. Terdapat dua tujuan utama dilakukannya proses pengujian, yaitu untuk mengetahui apakah aplikasi berjalan sesuai dengan yang telah dirancang dan mencatat semua *bug* dan *error* yang ada serta menguji ketepatan dari rekomendasi hasil luaran dari model. Pengujian dilakukan setelah proses pengkodean selesai dilakukan.

3.1.4 Penyusunan Buku Tugas Akhir

Pada tahapan ini dilakukan penyusunan laporan tugas akhir sebagai bentuk dokumentasi atas terlaksananya tugas akhir ini. Dalam laporan tersebut akan mencakup sebagai berikut:

a. Bab I Pendahuluan

Pada bab pendahuluan akan dijabarkan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat, dan relevansi terhadap pengerjaan tugas akhir.

b. Bab II Tinjauan Pustaka

Bab ini akan menjelaskan mengenai penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini.

c. Bab III Metodologi Penelitian

Pada bab ini dijelaskan mengenai tahapan-tahapan apa saja yang dilakukan beserta jadwal pengerjaan tugas akhir ini.

d. Bab IV Perancangan

Pada bab ini berisi rancangan penelitian, rancangan bagaimana penelitian akan dilakukan, pemilihan objek penelitian, dan sebagainya.

e. Bab V Implementasi

Pada bab ini berisi proses pelaksanaan penelitian, bagaimana penelitian dilakukan, penerapan strategi pelaksanaan, hambatan, dan sebagainya.

f. Bab VI Analisis dan Pembahasan

Pada bab ini berisi pembahasan tentang penyelesaian permasalahan yang dikerjakan pada penelitian tugas akhir ini.

g. Bab VII Kesimpulan dan Saran

Berisi tentang kesimpulan dan saran yang ditujukan untuk kelengkapan penyempurnaan tugas akhir ini.

3.2 Jadwal Penyusunan Tugas Akhir

Rencana pelaksanaan tugas akhir ini tertuang pada Tabel 3.1 Jadwal Pelaksanaan Tugas Akhir dibawah ini.

Tabel 3.1 Jadwal Pelaksanaan Tugas Akhir

No	Kegiatan	Oktober	November	Desember	Januari
	Studi Literatur				
	Pembuatan Proses Bisnis				
	Pengumpulan Data				
	Pembuatan Model DSS				
	Perancangan dan Pengembangan Aplikasi				
	Penyusunan Buku Tugas Akhir				

BAB IV

PERANCANGAN

Pada bab ini, akan dijelaskan tentang rancangan dari tugas akhir yang terdiri dari subjek dan objek dari tugas akhir ini. Selain itu akan dijelaskan juga proses pemilihan subjek dan objek dari tugas akhir yang akan dilakukan.

4.1 Perancangan Penggalan Kebutuhan

4.1.1 Identifikasi Variabel

Dalam menentukan sebuah prioritas tentunya harus mengetahui faktor – faktor apa yang dapat mempengaruhi prioritas tersebut. Faktor – faktor ini yang nantinya akan digunakan sebagai variabel dalam pembuatan model pendukung keputusan menggunakan rule based system. Berdasarkan sistem penentuan *driver* pada Uber dan analisa kebutuhan, berikut adalah beberapa faktor yang telah diidentifikasi:

- 1) Jarak antara Pos dan Angkutan Kota

Jarak merupakan salah satu faktor penting dalam menentukan prioritas penentuan angkot yang akan mengambil pesanan. Semakin pendek jarak antara titik Pos dan angkot maka

kemungkinan angkot untuk mengambil pesanan semakin tinggi.

Untuk menentukan jarak antara pengirim barang dan angkutan kota diperoleh berdasarkan perhitungan nilai latitude (*Lat*) dan longitude (*Lon*) yang didapatkan dari API Google Maps. Hasil perhitungan akan dikonversi dalam satuan kilo meter.

Namun sebelum menghitung jarak, angkutan kota yang menjadi kandidat harus memenuhi ketiga faktor dibawah. Hal ini untuk mengurangi jumlah perhitungan yang akan diproses.

2) Trayek Angkutan Kota Surabaya

Setiap angkutan kota di Surabaya pasti memiliki kode trayek untuk mengidentifikasi rute yang akan dilewati. Kode trayek ini pula yang akan digunakan untuk menyeleksi angkutan mana yang dapat mengambil pesanan sesuai dengan rute pengiriman barang.

Kode trayek angkutan kota di Surabaya telah disediakan oleh API Google Maps. Selanjutnya

sistem akan mengambil kode trayek dari rute pengiriman barang dan dicocokkan dengan kode trayek pada masing – masing angkutan kota. Sehingga akan diperoleh kandidat dari angkutan kota yang memenuhi satu faktor untuk dapat mengambil pesanan.

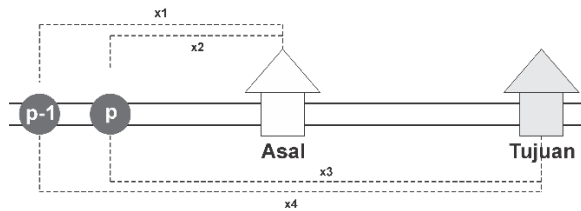
3) Kapasitas Muatan Angkutan Kota

Dalam pembuatan sistem pendukung keputusan ini, kapasitas muatan sebuah angkutan kota hanya dilihat dari dua status yaitu Penuh (*Full*) dan Tersedia (*Avalaible*). Dimana untuk merubah status tersebut akan diinisiasi oleh sopir angkutan kota. Sehingga dengan adanya status kapasitas angkutan kota akan menyeleksi kandidat dan menyisakan kandidat angkutan kota yang memiliki status Tersedia (*Available*).

4) Arah Perjalanan Angkutan Kota

Salah satu faktor yang cukup penting adalah arah perjalanan dari Angkutan Kota. Hal ini menjadi pertimbangan bagi sistem untuk memberikan pesanan yang memiliki arah yang sama dengan arah perjalanan angkot. Sehingga

sistem dituntut untuk mampu mengidentifikasi arah perjalanan angkot agar pesanan yang dikirim dapat dikirim lebih cepat ke tujuan. Untuk mengetahui arah perjalanan dari angkot, sistem perlu untuk mengetahui minimal 2 titik posisi terakhir dari angkot. Kedua titik tersebut akan dihitung masing – masing jaraknya terhadap titik asal (lokasi penjemputan barang) dan titik tujuan (lokasi tujuan pengiriman barang).



Gambar 4.1 Ilustrasi Penghitungan untuk Menentukan Arah

Keterangan :

P = Posisi terakhir angkot

P-1 = Posisi angkot sebelumnya

X1 = Jarak P-1 menuju titik Asal

X2 = Jarak P menuju titik Asal

X3 = Jarak P menuju titik Tujuan

$X4 = \text{Jarak } P-1 \text{ menuju titik Tujuan}$

Setelah mengetahui jarak dari dua posisi terakhir terhadap titik asal dan titik tujuan, maka untuk mengetahui kecocokan arah perjalanan angkot terhadap arah perjalanan pengiriman barang adalah dengan memenuhi syarat berikut :

$$X1 > X2$$

$$X4 > X3$$

$$X2 < X3$$

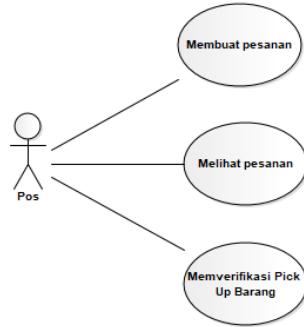
Dengan memenuhi persyaratan tersebut maka dapat diketahui bahwa posisi angkot sedang menuju titik asal dan titik tujuan.

4.1.2 Identifikasi Kebutuhan Pengguna

Proses mengidentifikasi kebutuhan pengguna sangat penting untuk mengetahui fungsional apa saja yang akan diterapkan pada sistem pendukung kebutuhan. Dengan adanya proses identifikasi kebutuhan pengguna akan memperjelas alur proses bisnis dari sistem.

Hasil dari identifikasi kebutuhan pengguna ini akan disajikan dalam bentuk use case diagram, sebagai berikut :

1) Pos

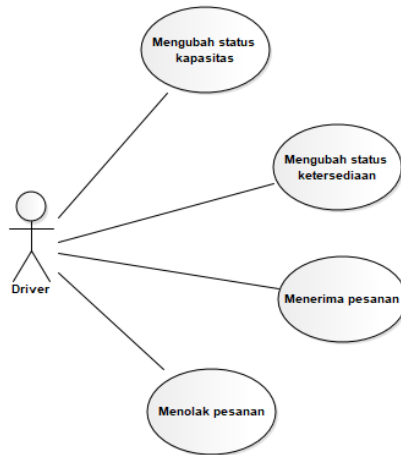


Gambar 4.2 Use Case Diagram untuk Pos

Peran dari Pos pada sistem pendukung keputusan ini adalah sebagai inisiator yaitu untuk mengajukan sebuah pesanan yang diterima dari pelanggan. Caranya adalah dengan mengisi beberapa informasi yang dibutuhkan oleh sistem. Setelah pesanan diterima maka Pos perlu untuk mengetahui status pesannya dan siapa yang akan mengambil pesanan tersebut. Selain itu Pos

juga berperan dalam memverifikasi pengambilan pesanan oleh *Driver* angkot.

2) Driver



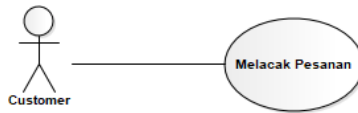
Gambar 4.3 Use Case Diagram untuk Driver

Driver dapat mengubah status kapasitas dan status ketersediaan yang akan digunakan sebagai masukan untuk sistem dalam menentukan ketersediaan driver dalam mengambil pesanan.

Selain itu driver akan menerima pesanan yang diteruskan dari hasil sistem pendukung

keputusan. Selanjutnya driver dapat menerima atau menolak pesanan tersebut.

3) Customer



Gambar 5 Use Case Diagram untuk Customer

Customer berhak mengetahui status pesanan yang telah dikirimkan melalui Pos atau istilah umumnya melacak pesanan. Pesanan dapat dilacak dengan memasukkan nomor resi yang telah diterima kedalam sistem pemantauan pengiriman barang.

4.2 Perancangan Desain

4.3 Perancangan Pengumpulan Data

Data yang digunakan dalam sistem pendukung keputusan ini merupakan data yang sifatnya cepat berubah, seperti data lokasi pengirim, data lokasi driver, rute pengiriman, dan kondisi status ketersediaan dari

angkutan kota terkini. Dimana data – data tersebut akan digunakan sebagai masukan pada sistem pendukung keputusan, terutama untuk pembuatan model.

Data yang digunakan seperti lokasi dari pengirim, lokasi angkot secara real-time, serta data trayek angkutan kota di Kota Surabaya telah tersedia di Google Maps dan dapat diambil menggunakan API dari Google Maps. Langkah yang dilakukan adalah dengan menyisipkan kode API Google Maps yang menggunakan Bahasa *javascript* pada *web client*.

Dengan mengimplementasikan beberapa kode, sistem akan dapat memperoleh lokasi dari pengirim dan driver secara otomatis. Namun untuk memperoleh kode trayek memerlukan inisiasi dari *customer* untuk mengisikan asal dan tujuan. Setelah google maps menentukan rute perjalanan yang harus dilalui barang, maka akan diketahui kode trayek angkutan kota yang akan dilalui untuk pengiriman barang.

4.3.1 Desain Database

Menyesuaikan dengan database yang terdapat pada aplikasi Angkotin, dimana database yang digunakan

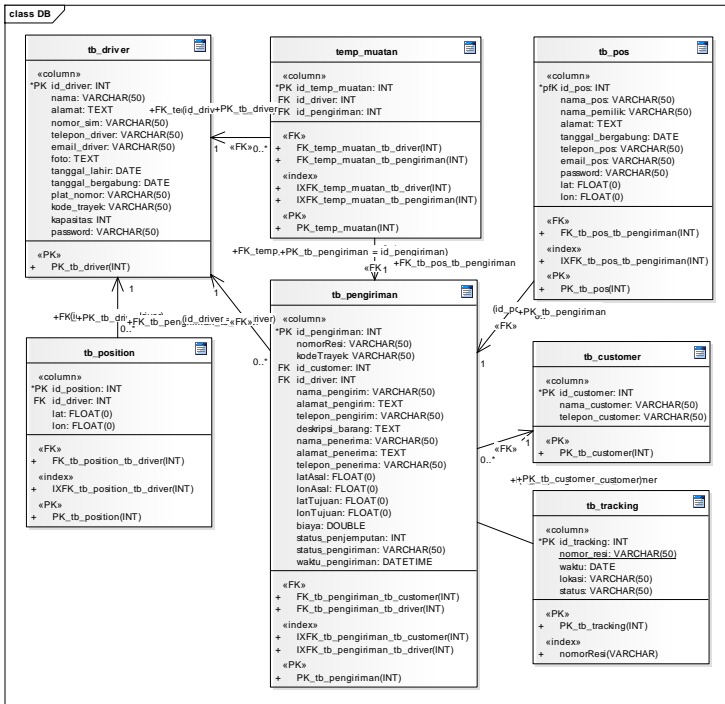
adalah MySQL. Dari 10 tabel yang terdapat pada database Aplikasi Angkotin, terdapat 7 tabel yang akan digunakan untuk mendukung pengembangan sistem pendukung keputusan ini. Pada Tabel 4.1 akan dijelaskan mengenai masing – masing tabel yang akan digunakan :

Tabel 4.1 Penjelasan Tabel

No	Tabel	Penjelasan
1	tb_driver	Berisi data yang mendiskripsikan sebuah angkot dan <i>Driver</i> -nya, seperti nama <i>Driver</i> , nomor Plat, kode trayek, dll.
2	tb_position	Memberikan gambaran posisi dari angkot secara <i>real time</i> yang disimpan setiap rentang waktu tertentu.
3	tb_pengiriman	Berisi data yang mengenai komponen pengiriman barang seperti diskripsi barang, alamat pengirim, alamat tujuan, dll.

4	tb_pos	Berisikan data mengenai Pos beserta letaknya dalam bentuk latitude dan longitude.
5	tb_customer	Berisi data pelanggan seperti nama, telepon, dan alamat.
6	tb_tracking	Menjelaskan riwayat perjalanan dari setiap pengiriman barang.
7	temp_muatan	Memberikan gambaran mengenai pesanan yang sedang dimuat setiap angkot.

Setiap tabel yang terdapat pada *database* memiliki hubungan dengan atribut yang lain. Hubungan antar tabel pada *database* dijabarkan dalam bentuk skema seperti yang terlihat pada Gambar 4.4 dibawah ini.



Gambar 4.4 Skema Relasi Antar Tabel

4.3.2 Desain Model

Perancangan model merupakan tahapan yang paling penting dalam pembuatan sistem pendukung keputusan ini. Model digunakan untuk mengklasifikasi angkutan kota yang akan menjadi kandidat untuk mengambil pesanan. Luaran dari model ini berupa daftar

pesanan yang dapat diambil oleh Driver sesuai dengan rute perjalanan pengiriman barang.

Dalam pembuatan model terdapat empat variabel yang harus dipenuhi oleh sebuah angkutan kota agar dapat menjadi kandidat, yaitu arah perjalanan angkot, kapasitas, kesesuaian kode trayek, dan jarak maksimal terhadap pengirim barang.

4.3.2.1. Input

Masukkan yang dibutuhkan untuk membuat sebuah model adalah variabel yang mempengaruhi penentuan pesanan. Variabel tersebut akan digunakan untuk membuat aturan – aturan dalam pembuatan model *rule based*. Variabel yang dibutuhkan diantaranya :

1. Data Trayek

Datat trayek yang dicari adalah data trayek yang sesuai dengan trayek dari pengiriman barang.

2. Jarak

Jarak antara angkot dengan titik asal pemesanan maksimal adalah 500 meter. Sehingga diluar jarak tersebut, model akan mengembalikan nilai *False* (Salah).

3. Kapasitas

Kapasitas dari setiap angkutan kota, model akan menampilkan pesan hanya apabila angkutan kota memiliki kapasitas yang belum penuh.

4. Arah Angkot

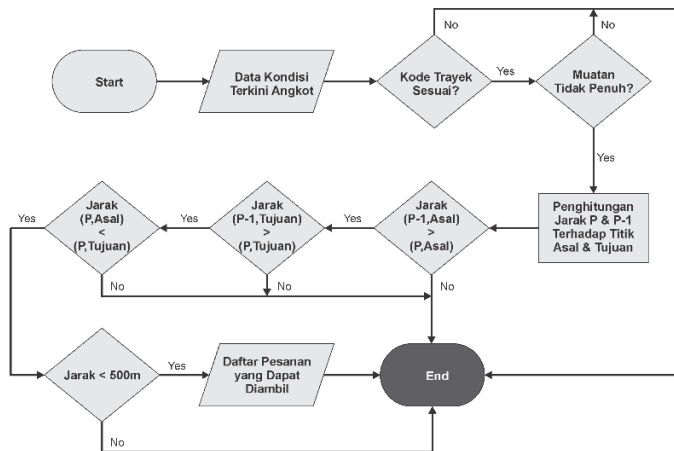
Arah dari angkot merupakan variabel yang sangat penting untuk menghindari kesalahan pengiriman barang. Untuk mendefinisikan arah dari angkutan kota perlu memperhitungkan jarak dan dua posisi terakhir dari angkutan kota seperti yang telah dijelaskan pada Gambar 4.1.

4.3.2.2. Proses

Selanjutnya keempat variabel tersebut akan dijadikan sebuah aturan (*rule*) untuk membuat sebuah *rule based systems*. Data yang masuk akan diproses melalui aturan – aturan yang telah didefinisikan. Aturan yang digunakan ada empat sesuai dengan variabel yang telah ditentukan sebelumnya, yaitu :

1. *If* (kapasitas angkot = 0 (tidak penuh))
Then...
2. *If* (kode trayek angkot = kode trayek pengiriman) *Then*...
3. *If* (arah angkot = arah pengiriman) *Then*...
4. *If* (jarak angkot < 500m) *Then*...

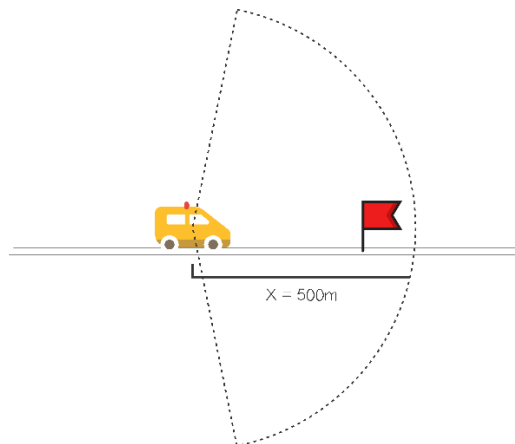
Keluaran (*Output*) dari model ini adalah sebuah daftar pesanan yang dapat diambil oleh *Driver*. Skema *rule based system* terlihat seperti Gambar 4.5 dibawah ini.



Gambar 4.5 Flowchart Model Rule Based

Model yang dikembangkan pada pembuatan sistem pendukung keputusan ini berorientasi pada *Driver*, sehingga objek yang dicari adalah pesanan yang

memenuhi kriteria untuk dapat diambil (*Pick Up*) oleh *Driver*. Gambaran dari pemodelan akan terlihat seperti Gambar 4.6 dibawah ini.



Gambar 4.6 Ilustrasi Pencarian Pesanan pada Aplikasi Angkotin

Pada aplikasi *Driver*, sistem akan melakukan pemrosesan model secara *real time* untuk mengetahui apakah ada pesanan yang dapat diambil oleh *Driver* tersebut. Apabila terdapat pesanan yang dapat diambil akan diberi tanda *Flag* untuk memberikan informasi kepada *Driver*. Jarak maksimal *Driver* terhadap pesanan adalah 500 meter, jadi diluar jarak tersebut tidak akan ada pesanan yang muncul pada peta.

4.3.2.3. Output

Keluaran dari model ini adalah berupa daftar pesanan yang dapat diambil oleh masing – masing angkutan kota. Daftar angkutan kota ini bersifat *real time*, tergantung pada kondisi dari setiap variable pada setiap waktu.

4.3.3 Desain User Interface

Pada bagian ini akan dijelaskan mengenai rancangan pembuatan tampilan (*User Interface*) pada aplikasi Angkotin, terutama yang berkaitan dengan sistem pendukung keputusan dalam pengambilan pesanan oleh angkutan kota. Rancangan tersebut akan dijabarkan pada Tabel 4.2 berikut :

Tabel 4.2 Rencana Pembuatan *User Interface*

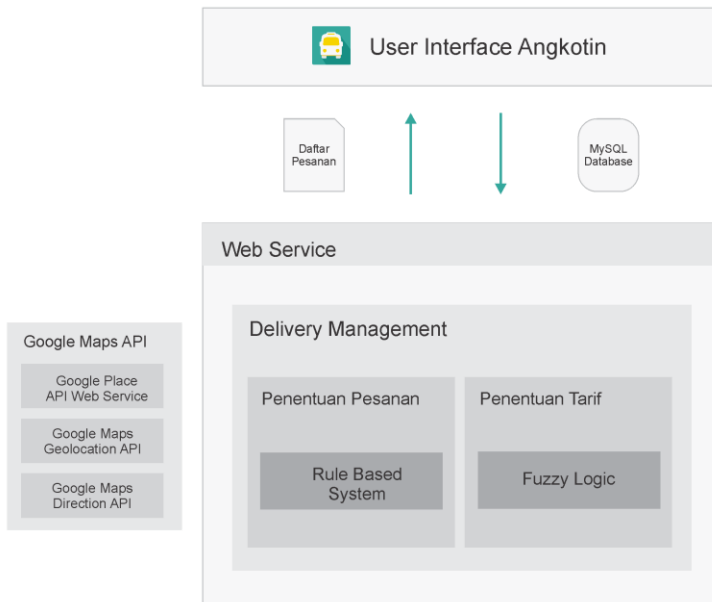
No	Tampilan	Fitur	Pengguna
1	Halaman Utama	Berisi menu : <ul style="list-style-type: none">• Tracking• Delivery	Pos
2	Halaman Pemesanan	Berisi titik – titik Pos yang digunakan untuk	Pos

		menentukan titik Asal dan titik Tujuan	
3	Halaman Detail Pemesanan	Berisi <i>Field</i> untuk mengisikan detail dari pesanan dan memberikan informasi biaya pengiriman	Pos
4.	Halaman Verifikasi Pengiriman	Berisi tombol untuk verifikasi penjemputan dan penerimaan barang dari <i>Driver</i>	Pos
5	Halaman Pesanan	Menampilkan peta Surabaya dan dapat memberikan informasi pesanan yang muncul di sekitar <i>Driver</i>	Driver
7	Halaman Detail Pesanan	Berisi daftar pesanan yang dapat diambil (<i>Pick Up</i>) oleh <i>Driver</i>	Driver
8	Halaman Muatan	Berisi daftar muatan barang yang sedang dalam pengiriman dan	Driver

		terdapat tombol “Drop barang” untuk menaruh barang di Pos Tujuan	
9	Halaman <i>Tracking</i>	Berisi <i>Field</i> untuk memasukkan nomor resi dari pengiriman guna mengetahui status pengiriman	<i>Customer</i>

4.3.4 Desain Arsitektur Sistem

Pada bagian ini akan dijelaskan mengenai rancangan dari arsitektur sistem yang akan digunakan. Gambaran dari arsitektur sistem terlihat seperti pada Gambar 4.7 dibawah ini.



Gambar 4.7 Arsitektur Sistem Aplikasi Angkotin

Fokus dari pengerjaan tugas akhir ini adalah pada proses bisnis pengiriman barang. Dimana terdapat dua proses utama dalam Aplikasi Angkotin yaitu:

1. Penentuan Pesanan

Merupakan proses untuk menentukan daftar pesanan yang diambil oleh sebuah angkot. Dalam proses penentuan tersebut terdapat empat faktor yang mempengaruhi status sebuah pesanan. Keempat

faktor itu akan dijadikan sebuah aturan dalam pembuatan model menggunakan *rule based system*.

3. Penentuan Tarif

Merupakan proses untuk menentukan tarif yang dikenakan pada suatu pengiriman barang. Tarif sendiri juga dipengaruhi beberapa faktor seperti berat, jumlah, dan volume. Metode yang digunakan untuk menentukan sebuah tarif adalah dengan menggunakan *Fuzzy Logic*.

Pada setiap proses diatas memiliki komponen yang menjadi sub-proses utama. Berikut ini merupakan penjelasan setiap komponen yang terdapat didalam kedua proses Sistem Pendukung Keputusan :

1. *Rule Based System*

Sebuah metode yang digunakan untuk memproses masukan dengan menggunakan aturan yang telah ditentukan. *Rule based system* dapat diterapkan dengan aturan sederhana sebagai berikut.

IF condition1

AND condition2

AND condition 3

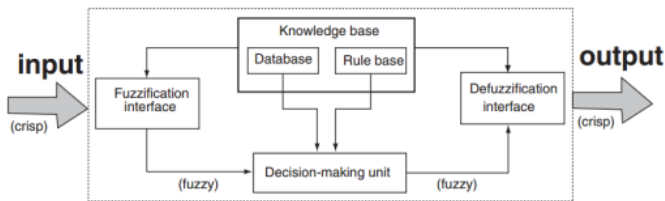
...

THEN action1, action2, action3, ...

Syarat dari penerapan *rule based system* adalah kondisi yang akan diimplementasikan dapat ditulis dengan IF-THEN rules.

2. Fuzzy Logic

Logika fuzzy digunakan untuk menerjemahkan besaran yang diekspresikan menggunakan linguistic, seperti besaran dimensi barang diekspresikan dengan kecil, sedang, dan besar.



Gambar 4.8 Komponen *Fuzzy Logic*

4.4 Perancangan Pengujian

4.4.1 Pengujian Fungsional

Pengujian fungsional dilakukan dengan melakukan unit test pada setiap fungsi yang berhasil dieksekusi dengan mengetahui apakah fitur yang dibuat pada aplikasi dapat berjalan dengan baik. Pengujian dilakukan dengan

metode *black box* dan mencatat *error* yang terjadi di setiap fitur. Pada tabel Tabel 4.3 dibawah ini terdapat daftar fitur yang akan diuji.

Tabel 4.3 Daftar Rencana Pengujian Fungsional

No	Fitur	Pengguna
F01	Menampilkan rute pengiriman	Pos
F02	Membuat pesanan	Pos
F03	Verifikasi pengambilan pesanan	Pos
F04	Verifikasi penerimaan pesanan	Pos
F05	Lacak status pengiriman	Pos, <i>Customer</i>
F06	<i>Pick Up</i> pesanan	<i>Driver</i>
F07	<i>Drop</i> Pesanan	<i>Driver</i>
F08	Lihat muatan	<i>Driver</i>

4.4.2 Pengujian Model

Pengujian rekomendasi dilakukan dengan melakukan validasi terhadap hasil dari sistem pendukung keputusan dengan melakukan simulasi rekomendasi pada 4 skenario yang berbeda. Tujuan pengujian rekomendasi dilakukan untuk mengetahui ketepatan rekomendasi yang dihasilkan dari sistem pendukung keputusan mengenai

ketersediaan angkutan kota dalam mengambil sebuah pesanan.

Terdapat dua hal yang perlu disiapkan untuk melakukan pengujian model ini, yaitu :

1. Penentuan Subjek

Subjek merupakan actor yang akan berperan dalam pengujian guna melaksanakan setiap scenario yang diujikan. Subjek pada sistem pendukung keputusan ini adalah *Driver* angkot. Setiap subjek memiliki karakteristik yang berbeda untuk menambah variasi pengujian. Untuk melakukan pengujian ini akan disiapkan 5 subjek dengan karakteristik berbeda yang akan menjalankan setiap scenario yang telah ditentukan.

2. Skenario Pengujian

Skenario merupakan rencana pelaksanaan pengujian yang telah dirancang untuk menghasilkan luaran yang diharapkan. Dalam pembuatan skenario pengujian ini mengkombinasikan beberapa variabel yang mempengaruhi sistem pendukung keputusan

yang telah dibuat. Pada Tabel 4.4 dibawah ini akan dijelaskan rancangan skenario yang akan digunakan untuk pengujian.

Tabel 4.4 Daftar Rencana Skenario Pengujian

No	Skenario	Jumlah <i>Test Case</i>
1	Menggunakan variasi kombinasi antara kode trayek, jarak, arah, dan kapasitas dari sebuah angkot dalam mengambil 1 buah pesanan.	5
2	Dengan menggunakan variasi kombinasi antara kode trayek, arah, jarak, dan kapasitas, namun akan ada minimal dua angkot dalam satu area dengan sebuah pesanan	3
3	Terdapat 3 pesanan yang akan diambil oleh	3

	subyek dengan variasi arah dan jarak	
4	Terdapat 5 pesanan yang akan diambil sebagian demi sebagian oleh subjek sesuai urutan jarak	3

Hasil dari pengujian model ini akan digunakan untuk menghitung akurasi kesuksesan model yang telah dibuat.

BAB V

IMPLEMENTASI

Pada bab ini, akan dijelaskan tentang rancangan dari tugas akhir yang terdiri dari subjek dan objek dari tugas akhir ini. Selain itu akan dijelaskan juga proses pemilihan subjek dan objek dari tugas akhir yang akan dilakukan.

5.1 Lingkungan Implementasi

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi pada Tabel 5.1 berikut:

Tabel 5.1 Spesifikasi Lingkungan Implementasi

Prosesor	Intel® Core™ i7-7200U CPU @ 2.50GHz
Memory	8 GB RAM
Sistem Operasi	Windows 10 Education
System type	64-bit Operating System, x64-based processor

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti editor, database, server; bahasa pemrograman, dan library yang disajikan dalam Tabel 5.2 berikut :

Tabel 5.2. Lingkungan teknologi yang digunakan

Websserver	Apache 2.4, Node JS
Bahasa Pemrograman	PHP 7.1, Javascript, CSS
Database	<ul style="list-style-type: none"> • MySQL (client)
Editor (IDE)	Sublime Text, Notepad++
Browser	Google Chrome 56
Library	<ul style="list-style-type: none"> • Google Maps API

5.2 Pengembangan Sistem

5.2.1 Pembuatan Halaman Pemesanan

Pesanan merupakan sebuah masukan untuk menginisiasi proses yang terdapat pada sistem pendukung keputusan. Halaman pesanan ini akan menampilkan sebuah peta dimana terdapat beberapa titik yang menunjukkan sebuah Pos. Petugas Pos akan memilih titik

mana yang akan menjadi asal pengiriman, dan titik mana yang akan menjadi tujuan pengiriman. Hal yang menjadi dasar dari pembuatan pesanan adalah peta lokasi Surabaya yang diperoleh dari API Google Maps. Seperti pada Kode 5.1 dibawah ini.

```
1. function initMap() {
2.
3.     geocoder = new google.maps.Geocoder();
4.     //Menginisiasi pembuatan peta
5.     var map = new
6.     google.maps.Map(document.getElementById('map_canvas'), {
7.         zoom: 13, //level zoom
8.         panControl : false,
9.         streetViewControl : false,
10.        mapTypeControl : false,
11.        navigationControl : false,
12.        zoomControl: false,
13.        fullscreenControl: false,
14.        center: new google.maps.LatLng(-7.278266999999999,
15.        112.79230729999995),
16.    });
```

Kode 5.1 Inisiasi Pembuatan Peta

Setelah maps diinisiasi, langkah berikutnya adalah membuat sebuah marker yang menunjukkan letak dari Pos yang telah disediakan. Kode yang digunakan dapat dilihat pada Kode 5.2.

Titik yang akan digunakan sebagai Pos diperoleh dengan cara membuat sebuah *query* untuk mengambil data Pos yang terdapat pada *database*. Selanjutnya dibuatlah

sebuah perulangan sebanyak titik Pos yang ada untuk dijadikan sebuah *marker*.

```
1. //Mengambil daftar pos yang akan ditampilkan dalam peta
2. var pos = [
3.     <?php
4.         $sql_pos="select id_pos,lat,lon from tb_pos ";
5.         $result=mysql_query($sql_pos);
6.         while($data=mysql_fetch_assoc($result)){
7.             ?>
8.             ['<?php echo $data['id_pos'];?>', <?php echo
9.             $data['lat'];?>, <?php echo $data['lon'];?>],
10.         <?php } ?>
11. ];
12. //Melakukan perulangan untuk menampilkan marker pada
13.     setiap pos
14. for (i = 0; i < pos.length; i++) {
15.     marker1 = new google.maps.Marker({
16.     position: new google.maps.LatLng(pos[i][1], pos[i][2]),
17.     map: map,
18.     icon: 'https://developers.google.com/maps/documentation
19.     /javascript/examples/full/images/beachflag.png',
20. });
```

Kode 5.2 Menampilkan Pos Dalam Bentuk Marker

Untuk menampilkan Pos pada peta membutuhkan data lokasi dari masing – masing Pos yang terdapat pada *database*. Oleh karena itu dibuatlah sebuah *Query* yang bertujuan untuk mengambil data posisi dari Pos dalam *database*. *Query* yang digunakan untuk mengambil data Pos akan dijelaskan pada Tabel 5.3 dibawah ini disertai penjelasan penggunaannya.

Tabel 5.3 Query untuk Mengambil Data Pos

<i>Query</i>	Penjelasan
<pre>SELECT id_pos, lat, lon FROM tb_pos</pre>	Untuk mengambil data posisi dari masing – masing Pos, terutama data latititude dan longitude yang digunakan untuk menampilkan Pos dalam bentuk marker ke dalam peta.

Setiap *marker* yang telah dibuat harus memiliki kemampuan untuk dapat digunakan untuk menentukan asal ataupun tujuan. Untuk itu dibuatlah sebuah *infowindow* untuk masing – masing *marker* yang memiliki dua tombol yang dapat mengambil alamat asal dan tujuan. Alamat asal dan tujuan yang berhasil diambil akan ditampilkan kedalam *field* Asal dan *field* Tujuan. Kode yang digunakan untuk menampilkan infowindow seperti terlihat pada Kode 5.3 dibawah ini.

```
1. //Content berisi tombol untuk memilih Asal dan Tujuan
```

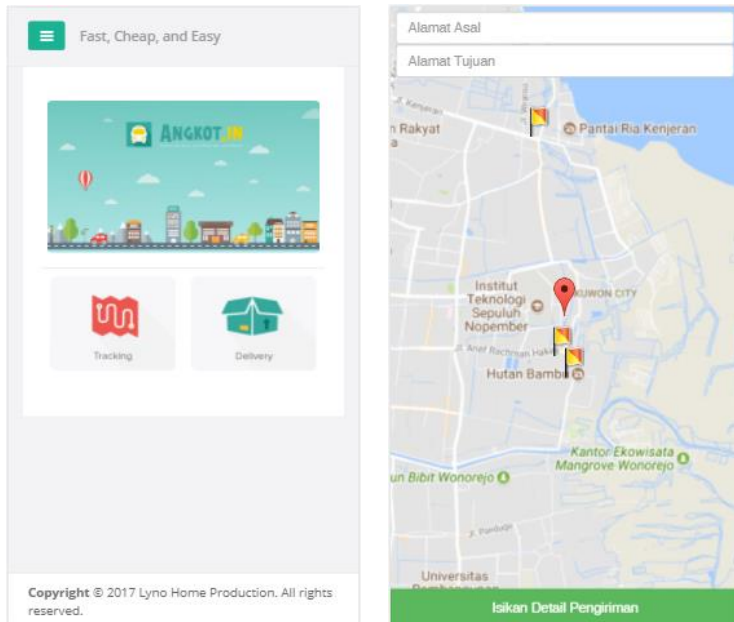
```

2. var content = '<div id="content">'+
3.   '<div id="siteNotice">'+
4.   '</div>'+
5.   '<button class="button" id="setasal">Pilih
asal</button>'+
6.   '<button class="button" id="settujuan" style="float:
right;">Pilih tujuan</button>'
7.   '</div>';
8.   infowindow.setContent(content);
9.
10. //Menampilkan infowindow untuk masing-masing marker
11. google.maps.event.addListener(marker1, 'click',
(function(marker1, i) {
12.     return function() {
13.         var id= pos[i][0];
14.         infowindow.open(map, marker1);
15.         google.maps.event.addDomListener(infowindow,
'domready', function() {
16.             $('#setasal').click(function() {
17.                 geocodePosition(marker1.getPosition());
18.             });
19.             $('#settujuan').click(function() {
20.                 geocodePosition2(marker1.getPosition());
21.                 carirute();
22.             });
23.         });
24.     }
25. })(marker1, i));
26. }

```

Kode 5.3 Menampilkan Infowindow yang Berisi Tombol

Tampilan dari halaman pemesanan yang telah dibuat akan terlihat seperti pada Gambar 5.1 dibawah ini.



Gambar 5.1 Halaman Utama dan Pemesanan

5.2.2 Pembuatan Rute Rencana Perjalanan

Berdasarkan proses sebelumnya dimana titik asal dan titik tujuan didefinisikan, maka langkah berikutnya adalah membuat rute perjalanan barang. Pada Google Maps telah disediakan kode yang digunakan untuk menentukan rute dari 2 buah titik. Dalam penentuan rute yang dimiliki Google Maps, terdapat beberapa mode

perjalanan yang dapat digunakan seperti mode jalan kaki (*Walking*), mode kendaraan pribadi (*Driving*), mode kendaraan umum (*Transit*), dan mode pesawat (*Flight*). Dalam studi kasus ini mode perjalanan yang digunakan adalah *Transit* karena angkutan kota termasuk kedalam kategori ini. Kode yang digunakan dapat dilihat pada Kode 5.1 berikut.

```
1. function carirute(){
2.   //Mendeklarasikan variabel getasal dan gettujuan
3.   var getasal = document.getElementById('asal').value;
4.   var gettujuan = document.getElementById('tujuan').value;
5.   if(getasal!="" && gettujuan !="" && getasal != gettujuan){
6.     //Memanggil fungctiion untuk mengkalkulasi rute
7.     calculateAndDisplayRoute(directionsService,
8.       directionsDisplay);
9.   } else{};
10. }
11. function calculateAndDisplayRoute(directionsService,
12.   directionsDisplay) {
13.   directionsService.route({
14.     origin: document.getElementById('asal').value,
15.     destination:
16.       document.getElementById('tujuan').value,
17.     //Angkot masuk kedalam travelmode 'TRANSIT'
18.     travelMode: 'TRANSIT'
19.   }, function(response, status) {
20.     if (status === 'OK') {
21.       directionsDisplay.setDirections(response);
22.       var route = response.routes[0];
23.       // Menampilkan infomasi untuk masing-masing rute
24.       for (var i = 0; i < route.legs.length; i++) {
25.         var routeSegment = i + 1;
26.         var steps = route.legs[i].steps;
```



```

26.         for(var j=0; j<route.legs[i].steps.length; j++){
27.             if (steps[j].travel_mode=="TRANSIT") {
28.                 //Mengambil kode trayek angkot
29.                 trayek.push(steps[j].transit.line.short_name);
30.             }
31.         }
32.     }
33.     } else {
34.         window.alert('Directions request failed due to '
+ status);
35.     } updateKodeTrayek(trayek[0]);
36. });}
37. });

```

Kode 5.4 Pencarian Rute Perjalanan

Selain rute perjalanan, proses ini juga menentukan kode trayek mana yang akan digunakan selama perjalanan pengiriman barang. Kode trayek inilah yang akan menjadi salah satu variabel dalam penentuan angkutan kota dalam pengambilan pesanan.

5.2.3 Pembuatan Model *Rule Based*

Model merupakan sebuah skema pemrosesan suatu *input* (masukan) yang akan menghasilkan sebuah *output* (keluaran) yang akan dibutuhkan untuk proses selanjutnya. Model yang digunakan pada pembuatan tugas akhir ini menggunakan metode *rule based*, dimana metode ini menggabungkan beberapa aturan (*rule*).

```

1. //Mengambil data semua pesananan
2. <?php $id_driver=$this->session->userdata('id_driver');?>
3. <?php $kode_trayek=$this->session->userdata
('kode_trayek'); ?>
4. var $orderan = [
5.     <?php
6.         //Query untuk mengambil daftar pesanan
7.         $sql_orderan="select * from tb_pengiriman a where
status=0 AND kodeTrayek='$kode_trayek'";
8.         $result2=mysql_query($sql_orderan);
9.         while($data2=mysql_fetch_assoc($result2)){
10.             ?>
11.             ['<?php echo $data2['id_pengiriman'];?>', '<?php echo
$data2['alamat_pengirim'];?>', '<?php echo
$data2['kodeTrayek'];?>', '<?php echo $data2['latAsal'];?>',
<?php echo $data2['lonAsal'];?>', '<?php echo
$data2['latTujuan'];?>', '<?php echo
$data2['lonTujuan'];?>'],
12.             <?php } ?>
13. ];
14. //Mengambil data 2 posisi driver terakhir
15. var $listPosisi = [
16.     <?php
17.         //Query untuk mengambil data posisi Driver
18.         $sql_dr="SELECT * FROM tb_position WHERE
id_driver=$id_driver ORDER BY id_position DESC LIMIT 2";
19.         $result=mysql_query($sql_dr);
20.         while($data=mysql_fetch_assoc($result)){
21.             ?>
22.             ['<?php echo $data['id_driver'];?>', '<?php echo
$data['lat'];?>', '<?php echo $data['lng'];?>'],
23.             <?php } ?>
24. ];

```

Kode 5.5 Pengambilan Data Pesanan dan Posisi *Driver*

Model yang telah dibuat membutuhkan input yang diperoleh dari database. Untuk itu terdapat dua *query* yang digunakan untuk mengambil data dari *database* seperti yang dijelaskan pada Tabel 5.4 dibawah ini.

Tabel 5.4 Query untu Mengambil Data Pesanan dan Posisi Driver

<i>Query</i>	Penjelasan
<pre>SELECT * FROM tb_pengiriman a WHERE status = 0 AND kodeTrayek = '\$kode_trayek'</pre>	<p>Untuk mengambil data pesanan yang masuk dengan parameter status kapasitas = 0 (tidak penuh) dan kode trayek pengiriman barang sama dengan kode trayek angkot.</p>
<pre>SELECT a.*,b.kapasitas FROM tb_position a LEFT JOIN tb_driver b ON a.id_driver=b.id_driver WHERE a.id_driver=4 ORDER BY a.id_position DESC LIMIT 2</pre>	<p>Untuk mengambil 2 data posisi <i>Driver</i> terakhir yang akan digunakan untuk menghitung jarak terhadap titik Asal dan titik Tujuan.</p>

Setelah seluruh data terkumpul dalam *array*, selanjutnya dilakukan perulangan untuk menghitung masing – masing posisi terhadap titik Awal dan titik Tujuan. Hasil dari pengukuran jarak ini akan menjadi masukkan dalam pemrosesan model menggunakan

metode *Rule Based*. Kode dari penghitungan jarak dapat dilihat pada Kode 5.6.

```
1. //Menghitung jarak
2. daftar.length=0;
3. for (h = 0; h < orderan.length; h++) {
4.     for (i = 0; i < listPosisi.length; i++) {
5.         var jarakAsal = distance(orderan[h][3],
6.         orderan[h][4], listPosisi[i][1], listPosisi[i][2]);
7.         var jarakTujuan = distance(orderan[h][5],
8.         orderan[h][6], listPosisi[i][1], listPosisi[i][2]);
9.         var idp = orderan[h][0];
10.         // Data tiap perhitungan disimpan dalam array
11.         daftar.push([idp, jarakAsal, jarakTujuan,
12.         orderan[h][1], orderan[h][2], listPosisi[0][3]]);
13.     }
14.     //array hasil perhitungan menjadi input untuk model
15.     hitungRuleBased(daftar);
16. //Fungsi untuk menghitung jarak menggunakan lat lon
17. function distance(lat1, lon1, lat2, lon2) {
18.     var radlat1 = Math.PI * lat1/180
19.     var radlat2 = Math.PI * lat2/180
20.     var theta = lon1-lon2
21.     var radtheta = Math.PI * theta/180
22.     var dist = Math.sin(radlat1) * Math.sin(radlat2) +
23.     Math.cos(radlat1) * Math.cos(radlat2) *
24.     Math.cos(radtheta);
25.     dist = Math.acos(dist)
26.     dist = dist * 180/Math.PI
27.     dist = dist * 60 * 1.1515
28.     dist = dist * 1.609344
29.     return dist
30. }
```

Kode 5.6 Penghitungan Jarak Antar Titik

Selanjutnya hasil dari penghitungan jarak yang tersimpan dalam *array* akan dikirimkan ke fungsi perhitungan model menggunakan metode *Rule Based*. Kode yang digunakan dalam mengimplementasikan *Rule Based* terlihat seperti pada Kode 5.7.

```
1. function hitungRuleBased(daftar) {
2.
3.   for (var i = 0; i < daftar.length/2; i++) {
4.     var a = i*2;
5.     var b = a+1;
6.
7.     //Rule untuk memeriksa kapasitas angkot
8.     if (daftar[a][5]==0) {
9.       var kapasitas=0;
10.    } else kandidat = null;
11.
12.    //Rule untuk memeriksa kode trayek angkot dan pengiriman
    barang
13.    if (kapasitas==0 && daftar[a][4]==kode) {
14.      var trayek = true;
15.    } else kandidat = null;
16.
17.    //Rule untuk memeriksa kesesuaian arah angkot dan
    pengiriman barang
18.    if (trayek==true && daftar[a][1]<daftar[b][1] &&
    daftar[a][2]<daftar[b][2] && daftar[a][1]<daftar[a][2]) {
19.      var arah = true;
20.    } else kandidat = null;
21.
22.    //Rule untuk memeriksa jarak angkot terhadap titik asal
    pesanan
23.    if (arah==true && daftar[a][1]<0.5) {
24.      kandidat[i] =
    [daftar[a][0],daftar[a][1],daftar[a][3]];
25.    } else kandidat = null;
```

```
26. }  
27. return kandidat;  
28. }
```

Kode 5.7 Implementasi Model dengan Metode *Rule Based*

Sesuai dengan yang telah dirancang sebelumnya pada Bab IV, dimana terdapat empat aturan yang berlaku dan harus dipenuhi setiap pesanan. Detail dari pembuatan model dengan empat aturan adalah sebagai berikut :

1. *If* (kapasitas angkot = 0 (tidak penuh)) *Then*...

Aturan ini menjelaskan bahwa angkot hanya bisa mendapatkan rekomendasi pesanan apabila kapasitas angkot masih tersedia. Apabila kondisi tersebut terpenuhi, maka pesanan akan diproses ke aturan berikutnya. Namun apabila angkot dalam keadaan penuh maka tidak akan ada rekomendasi pesanan yang muncul, jadi model akan mengembalikan array dengan nilai *null*.

2. *If* (kode trayek angkot = kode trayek pengiriman) *Then*...

Aturan ini menjelaskan bahwa angkot hanya bisa mendapatkan rekomendasi pesanan apabila memiliki kode trayek yang sama

dengan kode trayek pengiriman barang. Apabila kondisi tersebut terpenuhi, maka pesanan akan diproses ke aturan berikutnya. Namun apabila angkot memiliki kode trayek yang berbeda maka tidak akan ada rekomendasi pesanan yang muncul, jadi model akan mengembalikan array dengan nilai *null*.

3. *If* (arah angkot = arah pengiriman) *Then*...

Aturan ini menjelaskan bahwa angkot hanya bisa mendapatkan rekomendasi pesanan apabila memiliki arah yang sama dengan kode trayek pengiriman barang. Apabila kondisi tersebut terpenuhi, maka pesanan akan diproses ke aturan berikutnya. Namun apabila angkot memiliki arah yang berbeda maka tidak akan ada rekomendasi pesanan yang muncul, jadi model akan mengembalikan array dengan nilai *null*. Untuk mengetahui bagaimana sistem mampu melihat arah perjalanan angkot telah dijelaskan sebelumnya pada Bab IV.

4. *If* (jarak angkot < 500m) *Then*...

Aturan ini menjelaskan bahwa angkot hanya bisa mendapatkan rekomendasi pesanan apabila pada jarak kurang dari 500 meter terhadap titik asal pesanan. Apabila kondisi tersebut terpenuhi, maka pesanan akan masuk kedalam array dan ditampilkan pada peta. Namun apabila jarak angkot dengan titik asal lebih dari 500 meter maka tidak akan ada rekomendasi pesanan yang muncul, jadi model akan mengembalikan array dengan nilai *null*.

Setelah melalui proses dengan beberapa aturan, model akan mengembalikan luaran dalam bentuk *array*. *Array* tersebut berisi daftar pesanan yang dapat diambil oleh *Driver* angkot dan dimunculkan dalam peta.

5.2.4 Halaman Pick Up Barang

Pada halaman ini berisi detail daftar barang yang dapat diambil oleh *Driver* untuk dikirimkan ke Pos tujuan. Selain itu terdapat dua tombol yang berguna untuk mengambil dan menolak barang. Apabila *Driver* menolak,

maka barang akan dihapus dari array. Sedangkan apabila Driver memilih untuk mengambil barang maka kode yang akan berjalan seperti Kode 5.8 dibawah ini.

```
1. public function antarkan() {
2.     $id_driver=$this->session->userdata('id_driver');
3.     $id=$this->input->post('id');
4.     $this->model_users->update_pengiriman($id,$id_driver);
5.     //Menambah data muatan
6.     $data_muatan = array(
7.         'id_pengiriman' => $this->input->post('id')
8.     );
9.     $this->model_users->tambah_muatan( $data_muatan );
10.    $kirim = $this->model_users->kiriman($id);
11.    $st='Picked';
12.
13.    //Menambah data tracking
14.    $data_kirim = array(
15.        'kode_pengiriman'=> $id,
16.        'lokasi'           => $kirim[0]->plat_nomor,
17.        'status'           => $st,
18.    );
19.    $this->model_users->tambah_tracking( $data_kirim);
20.    $this->load->view('user/muatan');
21. }
```

Kode 5.8 Pengambilan Barang oleh Driver

Kode diatas berguna untuk menambah data muatan dari Driver dan data tracking untuk masing masing pesanan.

5.2.5 Halaman Muatan

Halaman muatan berisi daftar pesanan yang telah diambil oleh *Driver*. Data yang terdapat pada halaman ini bersifat sementara. Jadi ketika *Driver* melakukan *Drop* muatan, maka data muatan akan dihapus.



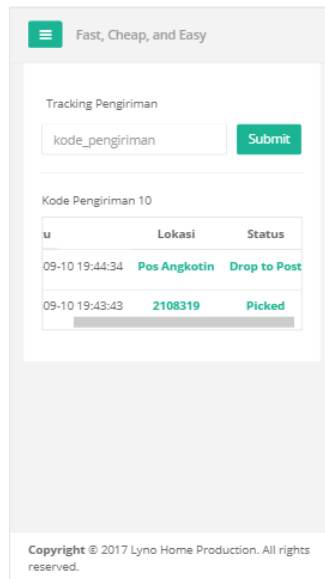
The screenshot shows a web interface with a navigation bar at the top containing 'Home', 'Muatan', and 'Logout'. Below the navigation bar, the user's name 'Hendra' is displayed in a blue header. The main content area contains a list of shipment details, each with a label and a value.

Hendra	
Nama sopir	Doran
Nomor Plat	L1892PG
Nama pengirim	Hendra
Alamat	Jl. Medokan Keputih, Keputih, Sukolilo, Kota SBY, Jawa Timur 60111, Indonesia
Telepon	081
Nama penerima	Rudi
Alamat penerima	Jl. Petojo, Pacar Keling, Tamb
Telepon penerima	082
Deskripsi barang	Buku

Gambar 5.2 Tampilan Halaman Muatan

5.2.6 Halaman Pelacakan Pengiriman

Pada halaman pelacakan muatan ini customer dan Pos dapat memeriksa status pengiriman dari barang dengan memasukkan nomor resi pengiriman.



The screenshot shows a mobile application interface for tracking shipments. At the top, there is a header with a menu icon and the text "Fast, Cheap, and Easy". Below this is a section titled "Tracking Pengiriman" which contains a text input field labeled "kode_pengiriman" and a green "Submit" button. Underneath the input field is a section titled "Kode Pengiriman 10" which displays a table of tracking events. The table has three columns: "u", "Lokasi", and "Status". The first row shows the time "09-10 19:44:34", the location "Pos Angkotin", and the status "Drop to Post". The second row shows the time "09-10 19:43:43", the tracking number "2108319", and the status "Picked". At the bottom of the page, there is a copyright notice: "Copyright © 2017 Lyno Home Production. All rights reserved."

u	Lokasi	Status
09-10 19:44:34	Pos Angkotin	Drop to Post
09-10 19:43:43	2108319	Picked

Gambar 5.3 Tampilan Halaman Tracking Barang

BAB VI

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil serta analisis terhadap hasil yang diperoleh dari proses implementasi yang telah dibahas pada bab sebelumnya.

6.1 Hasil

Berikut ini dijabarkan hasil dari implementasi yang telah dilakukan dalam sub bab sebelumnya serta proses pengujian yang dilakukan pada aplikasi. Pengujian yang dilakukan yaitu pengujian fungsional dengan metode *black box testing* dan pengujian rekomendasi berdasarkan skenario.

6.1.1 Pengujian Fungsional

Berikut ini dijabarkan hasil dari implementasi yang telah dilakukan dalam sub bab sebelumnya serta proses pengujian yang dilakukan pada aplikasi. Pengujian yang dilakukan yaitu functional testing dengan metode *black box*.

Pengujian sistem dilakukan untuk mengetahui apakah fitur yang dibuat pada aplikasi dapat berjalan dengan baik. Semua fitur akan dicoba, tetapi dokumentasi

pada buku ini hanya pada fitur-fitur utama aplikasi saja.

Tabel berikut daftar fitur yang diuji.

<i>Process</i>	<i>Input</i>	<i>Expected Result</i>	<i>Actual Result</i>
User Pos			
F01. Menampilkan rute pengirimman	Titik Asal & titik Tujuan	Bisa menampilkan rute pengiriman dan kode trayek angkot	Bisa menampilkan rute pengiriman dan kode trayek angkot. Namun terdapat beberapa kasus kode trayek tidak muncul.
	Sesuai dengan ekspektasi. Namun harus dilakukan perbaikan agar kode trayek bisa muncul pada setiap rute pengiriman.		
F02. Membuat pesanan	Titik Asal, titik Tujuan, nama pengirim, nama penerima, deskripsi barang	Sistem dapat menyimpan data pesanan	Sistem dapat menyimpan data pesanan
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		

F03. Verifikasi pengambilan pesanan	Data pesanan, <i>ID Driver</i>	<ul style="list-style-type: none"> • Pos dapat memverifikasi pesanan • Barang yang telah diverifikasi masuk kedalam muatan <i>Driver</i> 	<ul style="list-style-type: none"> • Pos dapat memverifikasi pesanan • Barang yang telah diverifikasi masuk kedalam muatan <i>Driver</i>
Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan <i>error</i> .			
F04. Verifikasi Penerimaan Pesanan	Data Pesanan, <i>ID Driver</i>	<ul style="list-style-type: none"> • Pos dapat memverifikasi barang dari <i>Driver</i> • Barang yang telah diverifikasi dihapus dari muatan <i>Driver</i> 	<ul style="list-style-type: none"> • Pos dapat memverifikasi barang dari <i>Driver</i> • Barang yang telah diverifikasi dihapus dari muatan <i>Driver</i>
Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan <i>error</i> .			
User Pos dan Customer			
F05. Lacak Status Pesanan	Nomor resi pengirimannya barang	Menampilkan riwayat dan status pengiriman barang	Menampilkan riwayat dan status pengiriman barang

	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan <i>error</i> .		
F06. Pick Up Pesanan	Data lokasi <i>Driver</i> , Data lokasi pesanan	<ul style="list-style-type: none"> • Memunculkan lokasi pesanan yang dapat diambil <i>Driver</i> • Melihat detail dari pesanan • Mengambil pesanan • Menolak pesanan • Menyimpan pesanan yang diambil kedalam database muatan 	<ul style="list-style-type: none"> • Memunculkan lokasi pesanan yang dapat diambil <i>Driver</i> • Melihat detail dari pesanan • Mengambil pesanan • Menolak pesanan • Menyimpan pesanan yang diambil kedalam database muatan
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan <i>error</i> .		
F07. Drop Pesanan	Data pesanan, ID Pos	Barang dalam muatan berubah status menjadi “Menunggu Verifikasi Pos”	Barang dalam muatan berubah status menjadi “Menunggu Verifikasi Pos”
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan <i>error</i> .		

F08. Lihat Muatan	Data pesanan	Sistem menampilkan daftar pesanan yang sedang dimuat di dalam angkot	Sistem menampilkan daftar pesanan yang sedang dimuat di dalam angkot
Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan <i>error</i> .			

6.1.2 Pengujian Model

Pengujian model dilakukan untuk mengetahui seberapa tepat rekomendasi daftar pesanan yang dihasilkan terhadap aturan - aturan yang telah ditentukan. Pengujian dilakukan dengan melakukan 4 skenario berbeda untuk diujikan kepada sistem. Masing - masing dari skenario telah memiliki hasil yang sudah diketahui. Tujuan dari pengujian ini adalah untuk mengetahui ketepatan luaran yang dihasilkan dari sistem pendukung keputusan menggunakan metode *rule based system*.

6.1.2.1 Penentuan Subjek Pengujian

Subjek yang akan digunakan dalam pengujian model ini adalah *Driver* angkot yang berjumlah 5 orang. Masing – masing subjek memiliki karakteristik yang berbeda untuk

memberikan variasi dalam pengujian. Berikut adalah kelima subjek yang akan digunakan :

ID Driver	1
Nama	Adit
Kode Trayek	O
Status Kapasitas	0 (Belum Penuh)

ID Driver	2
Nama	Berli
Kode Trayek	O
Status Kapasitas	1 (Sudah Penuh)

ID Driver	3
Nama	Budi
Kode Trayek	S
Status Kapasitas	0 (Belum Penuh)

ID Driver	4
Nama	Doran
Kode Trayek	O
Status Kapasitas	0 (Belum Penuh)

ID Driver	5
Nama	Berli
Kode Trayek	O
Status Kapasitas	0 (Belum Penuh)

6.1.2.2 Skenario Pengujian

Pembuatan skenario didasarkan pada kombinasi antar variabel yang mempengaruhi model serta pesanan yang masuk. Dengan menggunakan variasi dari keempat variabel, sehingga tercipta 4 skenario yang akan digunakan untuk pengujian model. Dibawah ini akan dijelaskan rincian dari masing – masing skenario.

Skenario 1

Menggunakan variasi kombinasi antara kode trayek, jarak, kapasitas muatan, dan arah yang berbeda dalam pengambilan satu pesanan. Hasil yang diharapkan dari skenario ini adalah untuk mengetahui perlakuan setiap angkot untuk sebuah pesanan.

Skenario 2

Dengan menggunakan variasi kombinasi antara jarak, kode trayek, arah, dan kapasitas angkot.

Fokus dari skenario ini adalah untuk mengetahui perlakuan lebih dari satu angkot terhadap sebuah pesanan. Jadi pada skenario ini akan ada minimal dua angkot yang terletak pada area yang sama dengan sebuah pesanan.

Skenario 3

Terdapat 3 pesanan yang dapat diambil oleh kelima subjek, namun harus memenuhi kondisi dari masing – masing pesanan yang berkaitan dengan arah dan jarak terhadap angkot. Hasil yang diharapkan dari skenario ini adalah masing – masing pesanan dapat diambil oleh angkot yang berbeda.

Skenario 4

Terdapat 5 pesanan yang dapat diambil oleh 3 subjek. Namun setiap angkot hanya dapat mengambil maksimal 3 pesanan. Prioritas pengambilan pesanan ditentukan oleh jarak dan kecepatan dalam pengmabilan. Hasil yang diharapkan dari skenario ini adalah pesanan yang dapat diambil untuk masing – masing angkot berbeda – beda, menyesuaikan urutan kedatangan.

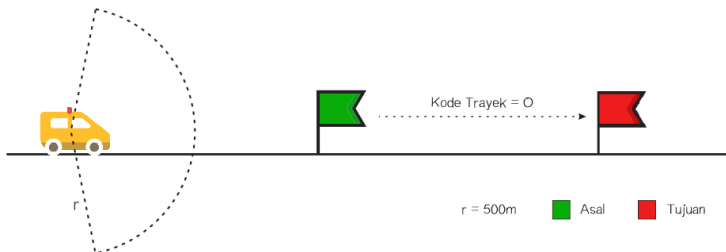
6.1.2.3 Hasil Pengujian

Pada bagian ini akan dijelaskan mengenai hasil pengujian rekomendasi untuk setiap *test case* beserta penjelasannya.

Test Case 1

Pada Test Case 1 dilakukan pengujian menggunakan skenario 1 untuk subjek dengan ID Driver 1.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 masih berada cukup jauh (>500 meter) dari titik asal pesanan.



Gambar 6.1 Ilustrasi Test Case 1

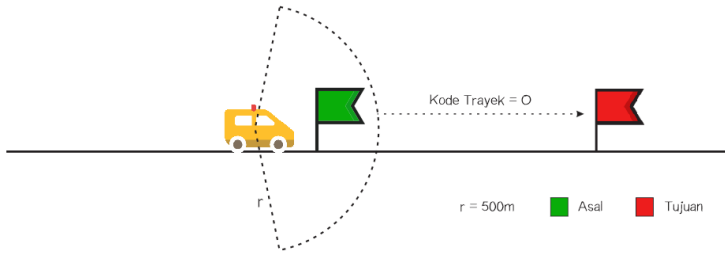
Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]

Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Expected Result	Actual Result
Tidak ada pesanan yang dapat diambil	Tidak ada pesanan yang dapat diambil
BENAR	

Test Case 2

Pada Test Case 2 dilakukan pengujian menggunakan skenario 1 untuk subjek dengan ID Driver 2.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 berada cukup dekat (<500 meter) dari titik asal pesanan.



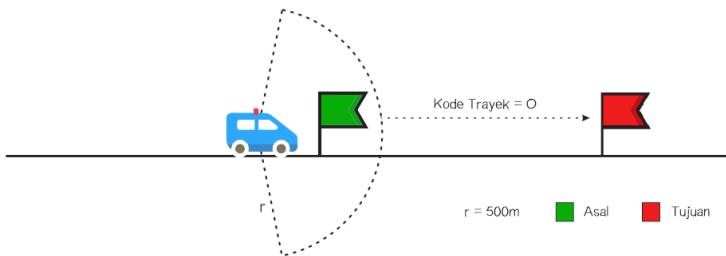
Gambar 6.2 Ilustrasi Test Case 2

Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	1 (Penuh)
Dua posisi terakhir	p [-7.290517, 112.800005] p-1 [-7.290682, 112.800336]
Expected Result	Actual Result
Tidak ada pesanan yang dapat diambil	Tidak ada pesanan yang dapat diambil
BENAR	

Test Case 3

Pada Test Case 3 dilakukan pengujian menggunakan skenario 1 untuk subjek dengan ID Driver 3.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 berada cukup dekat (<500 meter) dari titik asal pesanan namun memiliki kode trayek yang berbeda dengan kode trayek pengiriman.



Gambar 6.3 Ilustrasi Test Case 3

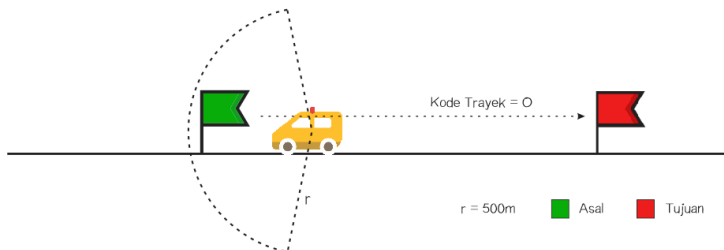
Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	S
Status kapasitas	0 (Belum Penuh)

Dua posisi terakhir	p [-7.290517, 112.800005] p-1 [-7.290682, 112.800336]
Expected Result	Actual Result
Tidak ada pesanan yang dapat diambil	Tidak ada pesanan yang dapat diambil
BENAR	

Test Case 4

Pada Test Case 4 dilakukan pengujian menggunakan skenario 1 untuk subjek dengan ID Driver 4.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 berada cukup dekat (<500 meter) dari titik asal pesanan namun berlawanan arah dengan arah pengiriman pesanan.



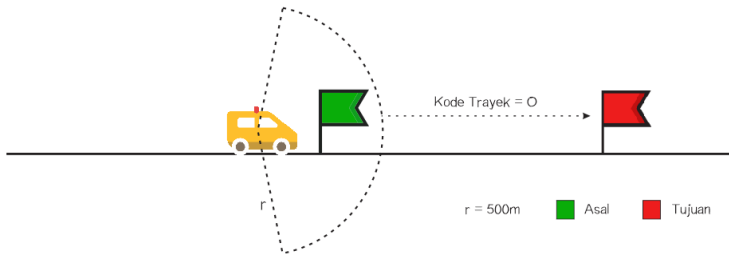
Gambar 6.4 Ilustrasi Test Case 4

Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum Penuh)
Dua posisi terakhir	p [-7.290137, 112.799175] p-1 [-7.289872, 112.798065]
Expected Result	Actual Result
Tidak ada pesanan yang dapat diambil	Tidak ada pesanan yang dapat diambil
BENAR	

Test Case 5

Pada Test Case 5 dilakukan pengujian menggunakan skenario 1 untuk subjek dengan ID Driver 5.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 berada cukup dekat (<500 meter) dari titik asal pesanan.



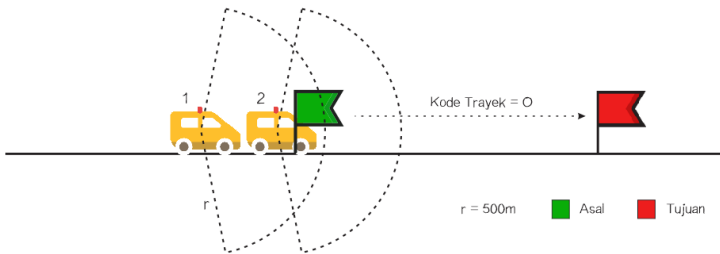
Gambar 6.5 Ilustrasi Test Case 5

Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum Penuh)
Dua posisi terakhir	p [-7.290517, 112.800005] p-1 [-7.290682, 112.800336]
Expected Result	Actual Result
Terdapat 1 pesanan muncul pada peta yang dapat diambil Driver	Terdapat 1 pesanan muncul pada peta yang dapat diambil Driver
BENAR	

Test Case 6

Pada Test Case 6 dilakukan pengujian menggunakan skenario 2 dengan dua subjek, yaitu ID Driver 1 dan ID Driver 2.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 dan angkot 2 berada pada jarak yang cukup dekat (<500 meter) dari titik asal pesanan.



Gambar 6.6 Ilustrasi Test Case 6

Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot 1	
Kode trayek	O

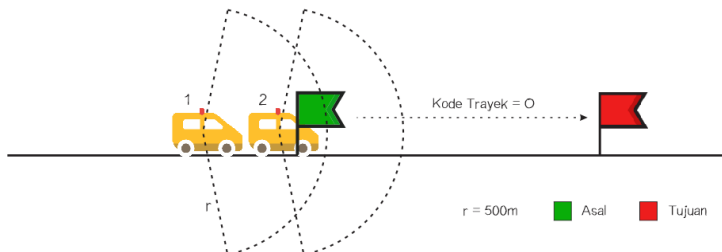
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Data Angkot 2	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.295424, 112.801465] p-1 [-7.295806, 112.801400]
Expected Result	Actual Result
Pada aplikasi kedua angkot muncul pesanan yang dapat diambil	Pada aplikasi kedua angkot muncul pesanan yang dapat diambil
BENAR	

Test Case 7

Pada Test Case 7 dilakukan pengujian menggunakan skenario 2 dengan dua subjek, yaitu ID Driver 1 dan ID Driver 2.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 dan angkot 2 berada pada jarak yang cukup dekat (<500 meter) dari titik asal

pesanan. Namun keadaan angkot 2 pada saat itu sedang penuh.



Gambar 6.7 Ilustrasi Test Case 7

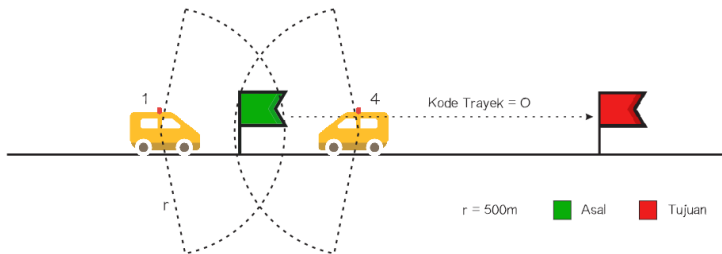
Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot 1	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Data Angkot 2	
Kode trayek	O
Status kapasitas	1 (Penuh)

Dua posisi terakhir	p [-7.295424, 112.801465] p-1 [-7.295806, 112.801400]
Expected Result	Actual Result
Pesanan hanya muncul pada aplikasi angkot 1	Pesanan hanya muncul pada aplikasi angkot 1
BENAR	

Test Case 8

Pada Test Case 8 dilakukan pengujian menggunakan skenario 2 dengan dua subjek, yaitu ID Driver 1 dan ID Driver 4.

Pada pengujian ini terdapat 1 pesanan yang terletak pada suatu titik dimana angkot 1 dan angkot 2 berada pada jarak yang cukup dekat (<500 meter) dari titik asal pesanan. Namun angkot 4 berjalan kearah yang berlawanan dengan angkot 2 dan rute pengiriman barang.



Gambar 6.8 Ilustrasi Test Case 8

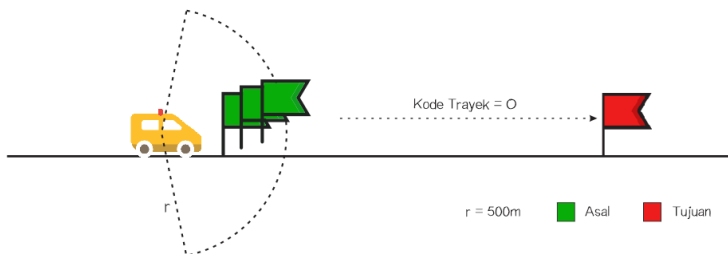
Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot 1	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.292988, 112.801744] p-1 [-7.291925, 112.801834]

Expected Result	Actual Result
Pesanan hanya muncul pada aplikasi angkot 1	Pesanan hanya muncul pada aplikasi angkot 1
BENAR	

Test Case 9

Pada Test Case 4 dilakukan pengujian menggunakan skenario 3 untuk subjek dengan ID Driver 4.

Pada pengujian ini terdapat 3 pesanan yang terletak pada suatu titik dimana angkot 1 berada cukup dekat (<500 meter) dari ketiga titik asal pesanan.



Gambar 6.9 Ilustrasi Test Case 9

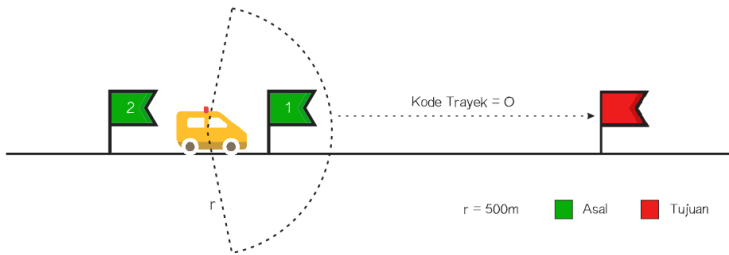
Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]

Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum Penuh)
Dua posisi terakhir	p [-7.290137, 112.799175] p-1 [-7.289872, 112.798065]
Expected Result	Actual Result
Terdapat tiga pesanan yang muncul pada aplikasi angkot dengan ID Driver 4	Terdapat tiga pesanan yang muncul pada aplikasi angkot dengan ID Driver 4
BENAR	

Test Case 10

Pada Test Case 10 dilakukan pengujian menggunakan skenario 3 untuk subjek dengan ID Driver 5.

Pada pengujian ini terdapat dua pesanan dengan titik asal yang berbeda namun terletak cukup dekat dengan angkot 5 dengan jarak <500 meter dari titik asal pesanan.



Gambar 6.10 Ilustrasi Test Case 10

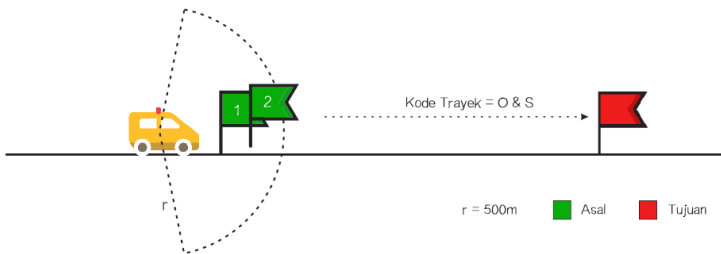
Data Pesanan 1	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Pesanan 2	
Trayek yang dilewati	O
Titik Asal	[-7.295588, 112.801470]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum Penuh)
Dua posisi terakhir	p [-7.290517, 112.800005] p-1 [-7.290682, 112.800336]
Expected Result	Actual Result

Hanya pesanan 1 yang akan muncul pada aplikasi Driver	Hanya pesanan 1 yang akan muncul pada aplikasi Driver
BENAR	

Test Case 11

Pada Test Case 11 dilakukan pengujian menggunakan skenario 3 untuk subjek dengan ID Driver 1.

Pada pengujian ini terdapat dua pesanan dengan titik asal yang sama namun memiliki kode trayek pengiriman yang berbeda. Jarak kedua pesanan dengan angkot cukup dekat (<500m).



Gambar 6.11 Ilustrasi Test Case 11

Data Pesanan 1	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]

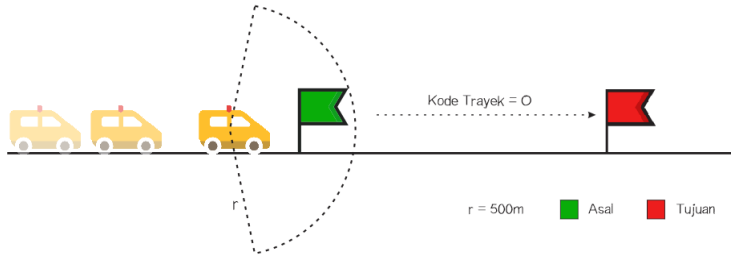
Titik Tujuan	[-7.257996, 112.757559]
Data Pesanan 2	
Trayek yang dilewati	S
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Expected Result	Actual Result
Hanya pesanan 1 yang muncul pada aplikasi Driver angkot.	Hanya pesanan 1 yang muncul pada aplikasi Driver angkot.
BENAR	

Test Case 12

Pada Test Case 12 dilakukan pengujian menggunakan skenario 4 untuk subjek dengan ID Driver 1.

Pada pengujian ini terdapat 5 pesanan yang terletak pada satu titik dimana angkot 1 datang lebih dahulu ke titik asal pesanan. Namun angkot tersebut hanya dapat membawa 3

pesanan untuk dikirimkan ke titik tujuan. Kedatangan angkot 1 ini disusul dua angkot dibelakangnya.



Gambar 6.12 Ilustrasi Test Case 12

Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Expected Result	Actual Result
Pada saat angkot 1 berada dekat dengan	Pada saat angkot 1 berada dekat dengan pesanan,

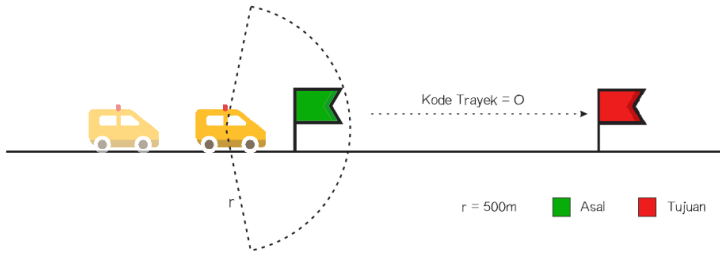
pesanan, terdapat lima pesanan yang muncul. [Action]	terdapat lima pesanan yang muncul. [Action]
Lalu angkot 1 mengambil 3 pesanan untuk diantarkan	Lalu angkot 1 mengambil 3 pesanan untuk diantarkan
BENAR	

Test Case 13

Pada Test Case 13 dilakukan pengujian menggunakan skenario 4 untuk subjek dengan ID Driver 2.

Pada pengujian ini terdapat 5 pesanan yang terletak pada satu titik dimana angkot 1 datang lebih dahulu ke titik asal dan mengambil 3 pesanan untuk dikirimkan ke titik tujuan.

Tidak lama kemudian angkot 2 datang untuk mengambil 2 pesanan yang tersisa.



Gambar 6.13 Ilustrasi Test Case 13

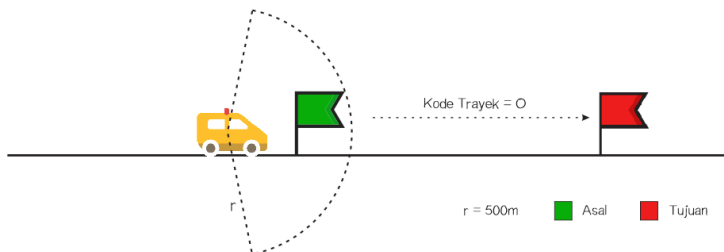
Data Pesanan	
Trayek yang dilewati	O
Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Expected Result	Actual Result
Pada saat angkot 2 berada dekat dengan pesanan, terdapat dua pesanan yang muncul. [Action]	Pada saat angkot 2 berada dekat dengan pesanan, terdapat dua pesanan yang muncul. [Action]

Lalu angkot 2 mengambil 2 pesan tersisa untuk diantarkan.	Lalu angkot 2 mengambil 2 pesan tersisa untuk diantarkan.
BENAR	

Test Case 14

Pada Test Case 14 dilakukan pengujian menggunakan skenario 4 untuk subjek dengan ID Driver 4.

Pada pengujian ini terdapat 5 pesanan yang terletak pada satu titik dimana angkot 1 datang lebih dahulu ke titik asal dan mengambil 3 pesanan diikuti angkot 2 yang mengambil 2 pesan tersisa. Tidak lama kemudian angkot 4 datang mendekati titik asal pengiriman barang.



Gambar 6.14 Ilustrasi Test Case 14

Data Pesanan	
Trayek yang dilewati	O

Titik Asal	[-7.290392, 112.799698]
Titik Tujuan	[-7.257996, 112.757559]
Data Angkot	
Kode trayek	O
Status kapasitas	0 (Belum penuh)
Dua posisi terakhir	p [-7.294797, 112.801534] p-1 [-7.294797, 112.801534]
Expected Result	Actual Result
Pada saat angkot 3 berada dekat dengan titik asal pesanan, tidak ada pesanan yang muncul. Sehingga tidak ada pesanan yang dapat diambil.	Pada saat angkot 3 berada dekat dengan titik asal pesanan, tidak ada pesanan yang muncul. Sehingga tidak ada pesanan yang dapat diambil.
BENAR	

6.2 Pembahasan

6.2.1. Analisis Hasil Black Box Testing

Pada sub bagian ini akan dilakukan pembahasan dan penarikan kesimpulan terkait serangkaian pengujian fungsional aplikasi dengan menggunakan Black Box Testing. Black Box Testing dilakukan untuk untuk menguji ketepatan aplikasi apakah hasil eksekusi dari aplikasi telah sesuai dengan apa yang telah di rancang sebelumnya. Pengujian juga dilakukan untuk menemukan kesalahan-kesalahan pada sistem dan struktur data, hal tersebut dilakukan untuk memastikan setiap fungsional aplikasi dapat berjalan dengan baik. Pengujian aplikasi dilakukan berdasarkan hasil kebutuhan fungsional web service serta terpenuhinya kebutuhan fungsional dari sisi client. Berdasarkan hasil pengujian dari 8 Fitur utama yang terkait dengan tugas akhir ini, memperlihatkan bahwasannya 7 fitur yang terdapat pada aplikasi ini dapat berjalan dengan baik tanpa adanya kesalahan atau kendala yang ditemukan. Menyisakan 1 fitur untuk menampilkan rute perjalanan yang sebenarnya telah berjalan dengan baik, namun pada beberapa kasus fitur tersebut tidak dapat

menampilkan kode trayek untuk pengiriman barang. Hal ini menjadi catatan bagi penulis untuk mengembangkan fitur pencarian rute pada API Google Maps.

6.2.2. Analisis Hasil Luaran Model

Pada sub bagian ini akan dilakukan pembahasan dan penarikan kesimpulan terkait serangkain pengujian hasil luaran dari model. Pengujian dilakukan dengan menggunakan empat skenario yang telah ditentukan kepada 3 – 5 subjek, yang dalam hal ini merupakan *Driver* angkot. Pengujian ini bertujuan untuk mengetahui apakah model yang telah dibuat menggunakan metode *Rule Based* telah berjalan dengan baik dan menghasilkan luaran yang telah diharapkan.

Hasil dari pengujian model yang telah dilakukan menunjukkan kinerja yang baik dilihat dari ketepatan luaran terhadap hasil yang diharapkan. Dari 14 *test case* yang dijalankan, semua kasus menghasilkan luaran yang sesuai dengan yang diharapkan. Hal ini menunjukkan bahwa metode *Rule Based* yang digunakan dapat berjalan dengan baik.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Setelah melakukan implementasi dan pengujian pada tugas akhir yang berjudul “Pembuatan Sistem Pendukung Keputusan Dengan Metode *Rule-Based System* Untuk Menentukan Prioritas Pengambilan Pesanan Oleh Angkutan Kota Pada Aplikasi Angkotin”, dapat disimpulkan beberapa hal sebagai berikut:

1. Pembuatan sistem pendukung keputusan menggunakan metode *rule-based system* telah berhasil dibuat dalam tugas akhir ini.
2. Aplikasi sistem pendukung keputusan ini melibatkan tiga jenis pengguna, yaitu pelanggan, pos, dan *driver* angkot.
3. Proses perhitungan model *rule based* terletak pada sisi *client* dari aplikasi *Driver*. Hal ini bertujuan untuk memudahkan *Driver* dalam memilih dan mengambil sebuah pesanan.
4. Terdapat empat variabel yang digunakan untuk menentukan prioritas pengambilan pesanan oleh

angkutan kota, yaitu jarak, arah perjalanan, status kapasitas, dan kode trayek.

5. Sistem penentuan pesanan ini tidak dapat berjalan apabila tidak terdapat jalur trayek angkot dalam sebuah rute pengiriman barang.
6. Sistem penentuan pesanan ini belum berjalan optimal pada pengiriman barang yang memiliki multi-trayek pada rute pengirimannya.
7. Hasil pengujian dari aplikasi menunjukkan performa yang sangat baik baik dari sisi fungsional maupun hasil luaran dari model. Pada pengujian fungsional, dari delapan fitur yang diuji, terdapat tujuh fitur yang berjalan baik tanpa adanya permasalahan ataupun kendala. Sedangkan pada pengujian luaran model, dari 14 kasus yang diujikan, semuanya dapat berjalan dengan baik dan sesuai dengan harapan.
8. Dengan keberhasilan dalam proses pengujian aplikasi, terutama yang berkaitan dengan pengujian luaran dari model, menunjukkan bahwa model yang dibuat menggunakan metode *Rule Based System* telah memenuhi hipotesa untuk

digunakan dalam penentuan prioritas pengambilan pesanan oleh angkutan kota.

7.2 Saran

Saran dari penulis yang dapat dilakukan untuk penelitian selanjutnya adalah sebagai berikut :

1. Kelemahan dari pengembangan aplikasi Angkotin, terutama yang berkaitan dengan penentuan angkutan kota dalam pengambilan pesanan yaitu belum adanya prosedur untuk pengiriman barang yang harus melalui dua trayek perjalanan. Sehingga diharapkan penelitian selanjutnya mampu memperbaiki kelemahan tersebut.
2. Penggunaan *database* dari pengembangan sistem pendukung keputusan ini perlu untuk diteliti lebih lanjut, terutama yang berkaitan dengan data *real time*. Sehingga diharapkan proses transaksi antara *client* dan *server* dapat berjalan lebih efektif dan efisien.
3. Data trayek angkutan kota di Kota Surabaya sebaiknya disimpan dalam *database* untuk dapat digunakan dalam pengembangan fitur dan

perbaikan pada fungsi pencarian rute serta kode trayek.

4. Penulis menyarankan untuk menambahkan fitur notifikasi pada aplikasi Angkotin untuk memperbaiki proses pemberitahuan informasi kepada *client*.
5. Pengembangan aplikasi Angkotin dalam bentuk mobile menjadi prioritas pengembangan di masa mendatang.

DAFTAR PUSTAKA

- [1] C. Grosan and A. Ajith, *Rule-Based Expert Systems*. Berlin, Heidelberg: Springer, 2011.
- [2] C. E. O'Meara, M. Brosnan, P. Kelly, D. Hyland, and D. Killeen, "Allocation of Location-Based Orders To Mobile Agents," vol. 2, no. 12, 2014.
- [3] T. Lusiani and A. kurniawan Cahyono, "SISTEM BERBASIS ATURAN UNTUK MENDIAGNOSA PENYAKIT FLU BURUNG SECARA ONLINE," pp. 156–163, 2006.
- [4] A. W. Cairns, R. R. Bond, D. D. Finlay, and S. J. Leslie, "A decision support system and rule-based algorithm to augment the human interpretation of the 12-lead electrocardiogram," 2017.
- [5] W. D. Sudjarmiko, "Sistem Pendukung Keputusan Pemberian Kredit Pada Koperasi Mitra Mandiri Sejahtera Kota Semarang," pp. 1–7, 2013.
- [6] E. Turban, J. E. Aronson, and T. Liang, *Decision Support Systems and Intelligent Systems*. 2007.
- [7] S. Mukundan, S. Ramani, and R. Chandrasekar, "A Practical Introduction to Rule Based Expert System," 2015.

- [8] A. Griffiths, *CodeIgniter 1.7 Professional Development*. Birmingham: Packt Publishing, 2010.
- [9] I. D. Id, *Framework Codeigniter: Sebuah Panduan dan Best Practice*. 2011.
- [10] M. Peterson, "International Perspectives on Maps and the Internet: An Introduction," no. March, 2016.

BIODATA PENULIS



Penulis lahir pada tanggal 10 September 1996. Merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu; SDN 2 Plosorejo Blitar, SMP Negeri 2 Blitar, dan SMA Negeri 1 Blitar.

Pada tahun 2014 pasca kelulusan SMA, penulis melanjutkan pendidikan dengan melalui jalur SBMPTN (Ujian Tulis) di Jurusan Sistem Informasi FTIK– Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 5214100126. Selama menjadi mahasiswa, penulis mengikuti berbagai kegiatan kemahasiswaan seperti beberapa kepanitiaan seperti Information System Expo serta pernah menjabat sebagai Produser di salah satu media di ITS yang bernama ITS TV. Selain itu, kegiatan seperti Latihan Ketrampilan Manajemen Mahasiswa pun pernah diikuti. Di bidang akademik, penulis pernah mengikuti lomba Gemastik 2017 dengan melombakan produk dari tugas akhir ini.

Pada tahun keempat, karena penulis memiliki ketertarikan di bidang pengolahan data, maka penulis mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Pada akhirnya penulis mengambil topik Sistem Pendukung Keputusan sebagai topik pegerjaan tugas akhir ini. Penulis dapat dihubungi melalui *email* di lookadityasepta@gmail.com.