

**PENGEMBANGAN APLIKASI *GEOTAGGING* PELAPORAN  
KELUHAN MASYARAKAT PADA DINAS PERHUBUNGAN KOTA  
MOJOKERTO MENGGUNAKAN *RESTFUL WEB SERVICES*  
BERBASIS *MOBILE***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Nizar Rahman Kusworo  
NIM: 175150209111022



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

# PENGESAHAN

## PENGESAHAN

PENGEMBANGAN APLIKASI *GEOTAGGING* PELAPORAN KELUHAN MASYARAKAT  
PADA DINAS PERHUBUNGAN KOTA MOJOKERTO MENGGUNAKAN *RESTFUL WEB*  
*SERVICES* BERBASIS *MOBILE*

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Nizar Rahman Kusworo  
NIM: 175150209111022

Skripsi ini telah diuji dan dinyatakan lulus pada  
24 Juli 2019  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Achmad Arwan, S.Kom, M.Kom  
NIP: 19840815 200812 1 004

Arief Andy Soebroto, S.T, M.Kom  
NIP: 19720425 199903 1 002

Mengetahui  
Ketua Jurusan Teknik Informatika



  
Dr. Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001



## PERNYATAAN ORISINALITAS

### PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 24 Juli 2019



NIZAR RAHMAN KUSWORO

NIM: 175150209111022

## PRAKATA

Segala puji bagi Allah SWT yang telah memberikan kemudahan sehingga penulis dapat menyusun dan menyelesaikan penelitian dengan judul “Pengembangan Aplikasi Geotagging Pelaporan Keluhan Masyarakat Pada Dinas Perhubungan Kota Mojokerto Menggunakan RESTFUL Web Services Berbasis *Mobile*”. Penelitian ini menjadi salah satu syarat untuk menyelesaikan pendidikan dan mengambil gelar sarjana pada Fakultas Ilmu Komputer Universitas Brawijaya. Pada bagian ini penulis menyampaikan terima kasih kepada pihak yang telah membantu diantaranya:

1. Tuhan Yang Maha Esa Allah SWT yang telah memberikan semua nikmat dan karunia-Nya sehingga dapat menyelesaikan penelitian ini.
2. Semua anggota keluarga dan teman penulis yang selalu memberikan kesabaran, nasehat, semangat dan bantuan kepada penulis dalam pengerjaan penelitian ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Jurusan Teknik informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Achmad Arwan, S.Kom, M.Kom selaku Dosen Pembimbing I yang dengan tulus, sabar dan ikhlas selama membimbing dan membantu penulis dalam menyusun penulisan ini.
5. Bapak Arief Andy Soebroto, S.T, M.Kom selaku Dosen Pembimbing II yang telah membantu dalam mengarahkan penyusunan penelitian ini dan juga memberikan bantuan pada penulisan ini.
6. Pihak Dinas Perhubungan Kota Mojokerto dan Ibu Ulfa Rahmawati selaku Narasumber yang waktunya telah di berikan kepada penulis dan memberikan izin dalam mencari informasi dalam penelitian ini

Penulis mengetahui bahwa penelitian ini masih memiliki banyak kekurangan sehingga diperlukannya kritik dan saran yang dapat membangun untuk penelitian ini. Semoga dengan dibuatnya penelitian ini mampu memberikan manfaat kepada siapa saja yang menggunakan penelitian ini sebagai acuan maupun referensi.

Malang, 24 Juli 2019

Penulis  
nizarrk@student.ub.ac.id

## ABSTRAK

**Nizar Rahman Kusworo, PENGEMBANGAN APLIKASI *GEOTAGGING* PELAPORAN KELUHAN MASYARAKAT PADA DINAS PERHUBUNGAN KOTA MOJOKERTO MENGGUNAKAN RESTFUL *WEB SERVICES* BERBASIS *MOBILE***

**Pembimbing: Achmad Arwan, S.Kom, M.Kom dan Arief Andy Soebroto, S.T, M.Kom**

Dinas Perhubungan Kota Mojokerto merupakan salah satu Satuan Kerja Perangkat Daerah (SKPD) Pemerintah Kota Mojokerto yang salah satu tugasnya melayani izin penggunaan jalan, melakukan survei kepuasan masyarakat dan menerima pegaduan keluhan masyarakat terhadap berbagai macam masalah yang terjadi di bidang transportasi dan lalu lintas. Permasalahan terkait tugas seperti masih ada masyarakat yang enggan melakukan izin penggunaan jalan, sulitnya proses penyebaran kuisisioner survei kepuasan masyarakat dan pegaduan keluhan dari masyarakat ini membuat Dinas Perhubungan kewalahan. Untuk mengatasi permasalahan tersebut pada penelitian ini dikembangkan sebuah aplikasi untuk masyarakat dapat mengurus izin penggunaan jalan, mengisi survei kepuasan masyarakat secara online dan dapat melaporkan keluhannya melalui *smartphone* dengan metode pengembangan *waterfall*. Aplikasi dibangun dengan teknologi *hybrid* menggunakan NodeJS sebagai *back-end*, VueJS sebagai *front-end framework* yang dipadukan dengan Framework7 yang menyediakan komponen untuk mendapatkan nuansa aplikasi *native* dan Apache Cordova yang berfungsi untuk mengemas aplikasi sebagai paket menggunakan *Software Development Kit* (SDK). Aplikasi ini bertujuan untuk memberikan pelayanan yang maksimal baik dalam izin penggunaan jalan, survei kepuasan masyarakat dan melaporkan keluhan. Hasil penelitian tersebut kemudian dilakukan pengujian menggunakan pengujian unit, pengujian integrasi, pengujian validasi dan pengujian *compatibility*. Hasil dari pengujian sistem yang dibangun menghasilkan 100% valid untuk semua pengujian, yang berarti sistem yang dibuat telah sesuai dengan hasil rekayasa kebutuhan dan perancangan.

**Kata kunci:** pelaporan, keluhan, izin penggunaan jalan, survei kepuasan masyarakat, *waterfall*, *nodejs*, *vuejs*, *framework7*, *apache cordova*

## ABSTRACT

**Nizar Rahman Kusworo, DEVELOPMENT OF COMMUNITY COMPLAINTS REPORTING GEOTAGGING APPLICATIONS IN TRANSPORTATION DEPARTMENT OF MOJOKERTO CITY USING MOBILE-BASED RESTFUL WEB SERVICES**

**Supervisors: Achmad Arwan, S.Kom, M.Kom and Arief Andy Soebroto, S.T, M.Kom**

*Transportation Department of Mojokerto City is one of the Regional Work Unit of the Mojokerto City Government whose task is to serve road use permits, carry out community satisfaction surveys and receive complaints regarding various problems that occur in the field of transportation and traffic. Problems related to tasks such as the existence of people who are reluctant to carry out road use permits, the difficulty of the process of distributing community satisfaction survey questionnaires and complaints from the public make the Transportation Department of Mojokerto City overwhelmed. To overcome these problems in this study an application was developed for the community to take care of road use permits, fill out a community satisfaction survey online and be able to report their complaints through a smartphone with the waterfall development method. Applications built with hybrid technology use NodeJS as a back-end, VueJS as a front-end framework combined with Framework7 that provides components to get a feel of native applications and Apache Cordova that functions to package applications as packages using the Software Development Kit (SDK). This application aims to provide maximum service both in road use permits, community satisfaction surveys and complaints. The results of the study were then tested using unit testing, integration testing, validation testing and compatibility testing. The results of the test system that was built produce 100% valid for all tests, which means the system is made in accordance with the results of engineering needs and design.*

**Keywords:** *reporting, complaints, road use permits, community satisfaction survey, waterfall, nodejs, vuejs, framework7, apache cordova*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA .....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan.....	3
1.4 Manfaat .....	3
1.5 Batasan Masalah .....	3
1.6 Sistematika Penulisan.....	3
<b>BAB II LANDASAN KEPUSTAKAAN .....</b>	<b>6</b>
2.1 Kajian Pustaka .....	6
2.2 Pelaporan Keluhan Masyarakat .....	7
2.3 Aplikasi <i>Hybrid</i> .....	8
2.4 <i>Global Positioning System</i> .....	8
2.5 <i>Geotagging</i> .....	8
2.6 <i>RESTful Web Services</i> .....	9
2.7 <i>Javascript Object Notation</i> .....	9
2.8 Apache Cordova .....	10
2.9 Framework7 .....	11
2.10 VueJS.....	11
2.11 NodeJS .....	12
2.12 Rekayasa Perangkat Lunak .....	12
2.13 <i>Software Development Life Cycle</i> .....	12
2.14 <i>Unified Modelling Language</i> .....	13
2.14.1 <i>Use Case Diagram</i> .....	13
2.14.2 <i>Sequence Diagram</i> .....	15



2.14.3	<i>Class Diagram</i> .....	17
2.15	Pengujian Perangkat Lunak .....	19
2.15.1	Pengujian <i>Black Box</i> .....	20
2.15.2	Pengujian <i>White Box</i> .....	20
BAB III	METODOLOGI .....	21
3.1	Struktur Tahap Pelaksanaan.....	21
3.2	Studi Literatur.....	21
3.3	Rekayasa Kebutuhan .....	22
3.3.1	Elisitasi kebutuhan .....	22
3.3.2	Identifikasi Aktor .....	22
3.3.3	Spesifikasi Kebutuhan .....	22
3.3.4	Pemodelan Kebutuhan .....	23
3.4	Perancangan .....	23
3.4.1	Perancangan Arsitektur .....	23
3.4.2	Perancangan Komponen.....	23
3.4.3	Perancangan Data .....	23
3.5	Implementasi.....	24
3.5.1	Implementasi Kode Program .....	24
3.5.2	Implementasi Antarmuka .....	24
3.5.3	Implementasi Basis Data.....	24
3.6	Pengujian .....	24
3.6.1	Pengujian Unit.....	24
3.6.2	Pengujian Integrasi.....	25
3.6.3	Pengujian Validasi .....	25
3.6.4	Pengujian <i>Compatibility</i> .....	25
3.7	Kesimpulan dan Saran .....	25
BAB IV	REKAYASA KEBUTUHAN .....	26
4.1	Analisis Kebutuhan .....	26
4.1.1	Elisitasi Kebutuhan.....	26
4.1.2	Gambaran Umum Sistem.....	29
4.1.3	Identifikasi Aktor .....	31
4.1.4	Aturan Penomoran.....	32
4.1.5	Kebutuhan Fungsional .....	32



4.1.6	Kebutuhan Non-fungsional .....	42
4.1.7	<i>Use Case Diagram</i> .....	43
4.1.8	Skenario <i>Use Case</i> .....	44
BAB V	PERANCANGAN DAN IMPLEMENTASI .....	79
5.1	Perancangan Sistem .....	79
5.1.1	Perancangan Arsitektur .....	79
5.1.2	Perancangan Algoritme.....	89
5.1.3	Perancangan Antarmuka.....	93
5.1.4	Perancangan Basis Data .....	96
5.2	Implementasi Sistem .....	109
5.2.1	Spesifikasi Sistem .....	109
5.2.2	Implementasi Kode Program .....	110
5.2.3	Implementasi Antarmuka .....	118
5.2.4	Implementasi Basis Data.....	122
BAB VI	PENGUJIAN .....	123
6.1	Pengujian.....	123
6.1.1	Pengujian Unit.....	123
6.1.2	Pengujian Integrasi.....	132
6.1.3	Pengujian Validasi .....	139
6.1.4	Pengujian <i>Compatibility</i> .....	150
BAB VII	PENUTUP .....	153
7.1	Kesimpulan .....	153
7.2	Saran.....	153
DAFTAR PUSTAKA.....		154
LAMPIRAN .....		156

## DAFTAR TABEL

Tabel 2. 1 Kajian Pustaka.....	6
Tabel 2. 2 HTTP <i>Method</i> .....	9
Tabel 2. 3 Penjelasan Simbol <i>Use Case Diagram</i> .....	14
Tabel 2. 4 Penjelasan Simbol <i>Sequence Diagram</i> .....	16
Tabel 2. 5 Penjelasan Simbol <i>Class Diagram</i> .....	18
Tabel 4. 1 Identifikasi Aktor .....	31
Tabel 4. 2 Aturan Penomoran .....	32
Tabel 4. 3 Kebutuhan Fungsional.....	32
Tabel 4. 4 Kebutuhan Non-fungsional .....	42
Tabel 4. 5 Skenario <i>Use Case Register</i> .....	45
Tabel 4. 6 Skenario <i>Use Case Login</i> .....	46
Tabel 4. 7 Skenario <i>Use Case</i> Lihat Laporan Keluhan .....	47
Tabel 4. 8 Skenario <i>Use Case</i> Tambah Laporan Keluhan .....	48
Tabel 4. 9 Skenario <i>Use Case Edit</i> Laporan Keluhan .....	49
Tabel 4. 10 Skenario <i>Use Case Vote</i> Laporan Keluhan.....	50
Tabel 4. 11 Skenario <i>Use Case</i> Tambah Komentar Laporan Keluhan .....	51
Tabel 4. 12 Skenario <i>Use Case Edit</i> Komentar Laporan Keluhan .....	52
Tabel 4. 13 Skenario <i>Use Case</i> Hapus Komentar Laporan Keluhan .....	53
Tabel 4. 14 Skenario <i>Use Case</i> Lihat Info Trayek .....	55
Tabel 4. 15 Skenario <i>Use Case</i> Lihat Info Pengumuman.....	56
Tabel 4. 16 Skenario <i>Use Case</i> Izin Penggunaan Jalan.....	57
Tabel 4. 17 Skenario <i>Use Case</i> Survei Kepuasan Masyarakat.....	58
Tabel 4. 18 Skenario <i>Use Case</i> Tambah Kritik dan Saran.....	59
Tabel 4. 19 Skenario <i>Use Case Edit Profile</i> .....	60
Tabel 4. 20 Skenario <i>Use Case</i> Verifikasi Laporan Keluhan .....	61
Tabel 4. 21 Skenario <i>Use Case Update</i> Status Laporan Keluhan .....	62
Tabel 4. 22 Skenario <i>Use Case</i> Cari Laporan Keluhan .....	63
Tabel 4. 23 Skenario <i>Use Case</i> Tambah Info Trayek .....	64
Tabel 4. 24 Skenario <i>Use Case Edit</i> Info Trayek .....	66
Tabel 4. 25 Skenario <i>Use Case</i> Hapus Info Trayek .....	67
Tabel 4. 26 Skenario <i>Use Case</i> Tambah Info Pengumuman .....	68
Tabel 4. 27 Skenario <i>Use Case Edit</i> Info Pengumuman .....	69
Tabel 4. 28 Skenario <i>Use Case</i> Hapus Info Pengumuman .....	70
Tabel 4. 29 Skenario <i>Use Case</i> Verifikasi Izin Penggunaan Jalan .....	71
Tabel 4. 30 Skenario <i>Use Case</i> Lihat Survei Kepuasan Masyarakat .....	73
Tabel 4. 31 Skenario <i>Use Case</i> Tambah Survei Kepuasan Masyarakat.....	74
Tabel 4. 32 Skenario <i>Use Case Edit</i> Survei Kepuasan Masyarakat.....	75
Tabel 4. 33 Skenario <i>Use Case</i> Hapus Survei Kepuasan Masyarakat.....	76
Tabel 4. 34 Skenario <i>Use Case</i> Lihat Kritik dan Saran .....	77
Tabel 4. 35 Skenario <i>Use Case Logout</i> .....	78
Tabel 5. 1 Algoritme Tambah Laporan Keluhan.....	89
Tabel 5. 2 Algoritme Lihat Info Trayek.....	90
Tabel 5. 3 Algoritme Izin Penggunaan Jalan .....	91



Tabel 5. 4 Algoritme Survei Kepuasan Masyarakat .....	91
Tabel 5. 5 Algoritme Kritik dan Saran .....	92
Tabel 5. 6 Penjelasan Detail Entitas User.....	98
Tabel 5. 7 Penjelasan Detail Entitas Laporan .....	98
Tabel 5. 8 Penjelasan Detail Entitas Izin .....	100
Tabel 5. 9 Penjelasan Detail Entitas Kritiksaran.....	101
Tabel 5. 10 Penjelasan Detail Entitas Komentar .....	102
Tabel 5. 11 Penjelasan Detail Entitas <i>Vote</i> .....	102
Tabel 5. 12 Penjelasan Detail Entitas Info.....	103
Tabel 5. 13 Penjelasan Detail Entitas Trayek .....	104
Tabel 5. 14 Penjelasan Detail Entitas Notifikasi.....	104
Tabel 5. 15 Penjelasan Detail Entitas Pertanyaan .....	105
Tabel 5. 16 Penjelasan Detail Entitas Jawaban .....	106
Tabel 5. 17 Penjelasan Detail Entitas Pertanyaan_Survei .....	106
Tabel 5. 18 Penjelasan Detail Entitas Jawaban_Survei .....	107
Tabel 5. 19 Penjelasan Detail Entitas Survei .....	108
Tabel 5. 20 Spesifikasi Perangkat Keras <i>Laptop</i> .....	109
Tabel 5. 21 Spesifikasi Perangkat Keras <i>Smartphone</i> .....	109
Tabel 5. 22 Spesifikasi Perangkat Lunak .....	110
Tabel 5. 23 <i>Source Code</i> Tambah Laporan Keluhan.....	110
Tabel 5. 24 <i>Source Code</i> Lihat Info Trayek .....	113
Tabel 5. 25 <i>Source Code</i> Izin Penggunaan Jalan.....	114
Tabel 5. 26 <i>Source Code</i> Survei Kepuasan Masyarakat .....	116
Tabel 5. 27 <i>Source Code</i> Kritik dan Saran.....	117
Tabel 6. 1 Algoritme Pengujian Unit <i>Method</i> tambahLaporan() .....	123
Tabel 6. 2 Hasil Pengujian Unit Tambah Laporan Keluhan .....	125
Tabel 6. 3 Algoritme Pengujian Unit <i>Method</i> createLapor() .....	126
Tabel 6. 4 Hasil Pengujian Unit <i>Method</i> createLapor().....	127
Tabel 6. 5 Algoritme Pengujian <i>Method</i> tambahIzin().....	128
Tabel 6. 6 Hasil Pengujian Unit <i>Method</i> tambahIzin() .....	129
Tabel 6. 7 Algoritme Pengujian <i>Method</i> createIzin() .....	130
Tabel 6. 8 Hasil Pengujian Unit <i>Method</i> createIzin() .....	131
Tabel 6. 9 Algoritme Pengujian Integrasi <i>Method</i> tambahLaporan() .....	132
Tabel 6. 10 Algoritme Pengujian Integrasi <i>Method</i> createLapor() .....	134
Tabel 6. 11 Algoritme Gabungan <i>Method</i> tambahLaporan() dan createLapor() .....	135
Tabel 6. 12 Hasil Pengujian Integrasi Tambah Laporan Keluhan.....	137
Tabel 6. 13 Hasil Pengujian Validasi.....	140
Tabel 6. 14 Perbandingan Spesifikasi Perangkat Pengujian <i>Compatibility</i> .....	150
Tabel 6. 15 Hasil Pengujian <i>Compatibility</i> .....	151

## DAFTAR GAMBAR

Gambar 2. 1 Struktur Object JSON.....	10
Gambar 2. 2 Struktur Array JSON .....	10
Gambar 2. 3 Struktur Value JSON .....	10
Gambar 2. 4 Cara Kerja Apache Cordova.....	11
Gambar 2. 5 Model Pengembangan <i>Waterfall</i> .....	13
Gambar 2. 6 Contoh <i>Use Case Diagram</i> .....	14
Gambar 2. 7 Contoh <i>Sequence Diagram</i> .....	16
Gambar 2. 8 Contoh <i>Class Diagram</i> .....	18
Gambar 3. 1 Diagram Flow Pengembangan .....	21
Gambar 4. 1 <i>Business Model As Is</i> Pelaporan Keluhan .....	27
Gambar 4. 2 <i>Business Model As Is</i> Izin Penggunaan Jalan .....	28
Gambar 4. 3 <i>Business Model As Is</i> Survei Kepuasan Masyarakat .....	28
Gambar 4. 4 Arsitektur RESTful Web Services .....	29
Gambar 4. 5 <i>Business Model To Be</i> Pelaporan Keluhan .....	30
Gambar 4. 6 <i>Business Model To Be</i> Izin Penggunaan Jalan .....	30
Gambar 4. 7 <i>Business Model To Be</i> Survei Kepuasan Masyarakat .....	31
Gambar 4. 8 <i>Use Case Diagram</i> Aplikasi Pelaporan Keluhan Bagian Satu.....	43
Gambar 4. 9 <i>Use Case Diagram</i> Aplikasi Pelaporan Keluhan Bagian Dua.....	44
Gambar 5. 1 <i>Sequence Diagram</i> Tambah Laporan Keluhan .....	80
Gambar 5. 2 <i>Sequence Diagram</i> Lihat Info Trayek.....	81
Gambar 5. 3 <i>Sequence Diagram</i> Izin Penggunaan Jalan .....	82
Gambar 5. 4 <i>Sequence Diagram</i> Survei Kepuasan Masyarakat .....	83
Gambar 5. 5 <i>Sequence Diagram</i> Kritik dan Saran .....	84
Gambar 5. 6 Diagram <i>Class</i> Aplikasi Pelaporan Mobile.....	86
Gambar 5. 7 Diagram <i>Class</i> Aplikasi Pelaporan Web Admin .....	87
Gambar 5. 8 Diagram <i>Class</i> API Aplikasi Pelaporan .....	88
Gambar 5. 9 Perancangan Antarmuka Tambah Laporan Keluhan .....	93
Gambar 5. 10 Perancangan Antarmuka Lihat Info Trayek.....	94
Gambar 5. 11 Perancangan Antarmuka Izin Penggunaan Jalan .....	95
Gambar 5. 12 Perancangan Antarmuka Survei Kepuasan Masyarakat .....	95
Gambar 5. 13 Perancangan Antarmuka Kritik dan Saran .....	96
Gambar 5. 14 <i>Entity Relationship Diagram</i> Aplikasi Pelaporan.....	97
Gambar 5. 15 Detail Entitas User.....	97
Gambar 5. 16 Detail Entitas Laporan .....	98
Gambar 5. 17 Detail Entitas Izin.....	99
Gambar 5. 18 Detail Entitas Kritiksaran .....	101
Gambar 5. 19 Detail Entitas Komentar .....	101
Gambar 5. 20 Detail Entitas <i>Vote</i> .....	102
Gambar 5. 21 Detail Entitas Info .....	103
Gambar 5. 22 Detail Entitas Trayek .....	103
Gambar 5. 23 Detail Entitas Notifikasi .....	104
Gambar 5. 24 Detail Entitas Pertanyaan.....	105
Gambar 5. 25 Detail Entitas Jawaban .....	105



Gambar 5. 26 Detail Entitas Pertanyaan\_Survei..... 106

Gambar 5. 27 Detail Entitas Jawaban\_Survei ..... 107

Gambar 5. 28 Detail Entitas Survei ..... 107

Gambar 5. 29 Implementasi Antarmuka Tambah Laporan Keluhan ..... 118

Gambar 5. 30 Implementasi Antarmuka Lihat Info Trayek..... 119

Gambar 5. 31 Implementasi Antarmuka Izin Penggunaan Jalan ..... 120

Gambar 5. 32 Implementasi Antarmuka Survei Kepuasan Masyarakat ..... 121

Gambar 5. 33 Implementasi Antarmuka Kritik dan Saran ..... 121

Gambar 5. 34 Implementasi Basis Data Menggunakan MySQL ..... 122

Gambar 6. 1 *Flowgraph Method* tambahLaporan()..... 124

Gambar 6. 2 *Flowgraph Method* createLapor()..... 127

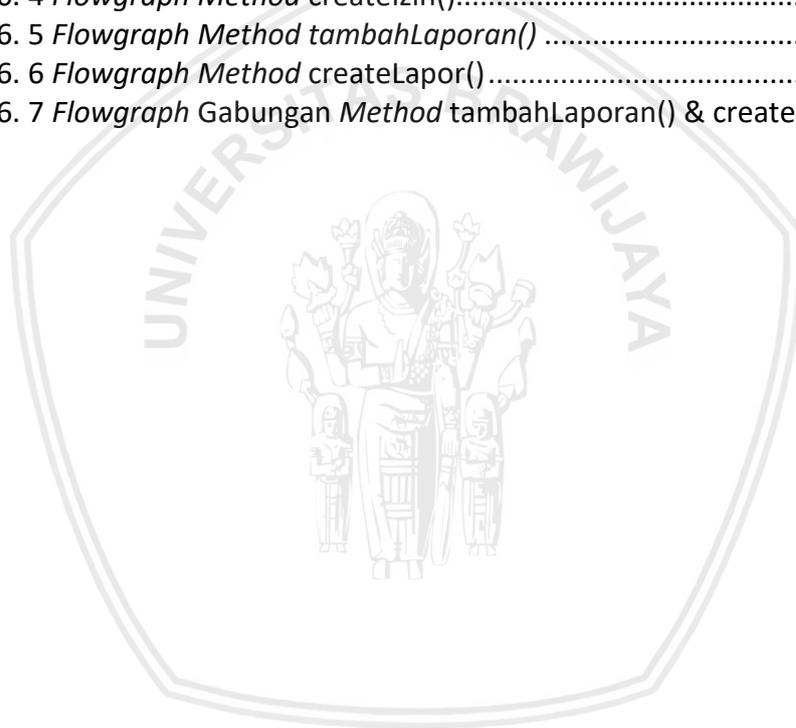
Gambar 6. 3 *Flowgraph Method* tambahIzin() ..... 129

Gambar 6. 4 *Flowgraph Method* createIzin()..... 131

Gambar 6. 5 *Flowgraph Method* tambahLaporan() ..... 133

Gambar 6. 6 *Flowgraph Method* createLapor()..... 134

Gambar 6. 7 *Flowgraph* Gabungan *Method* tambahLaporan() & createLapor() 136



## BAB I PENDAHULUAN

### 1.1 Latar belakang

Dinas Perhubungan Kota Mojokerto merupakan salah satu Satuan Kerja Perangkat Daerah (SKPD) Pemerintah Kota Mojokerto yang terbentuk sejak tahun 2008 sebagai implikasi diterapkannya Peraturan Pemerintah Nomor 41 tahun 2007 tentang Organisasi Perangkat Daerah. Dinas Perhubungan terbentuk dari unit satuan kerja yang meliputi Dinas Lalu Lintas dan Angkutan Jalan (Dishubkominfo, 2014). Salah satu tugas dari Dinas Perhubungan Kota Mojokerto adalah melayani izin penggunaan jalan dan melakukan survei kepuasan masyarakat terhadap layanan yang diberikan Dinas Perhubungan Kota Mojokerto. Selain itu pihak Dinas Perhubungan Kota Mojokerto juga menerima pengaduan keluhan masyarakat terhadap berbagai macam masalah yang terjadi di bidang transportasi dan lalu lintas.

Izin penggunaan jalan merupakan salah satu tugas dari Dinas Perhubungan Kota Mojokerto. Izin penggunaan jalan pada dasarnya diperbolehkan tetapi jalan yang digunakan harus mempunyai jalur alternatif dan mempunyai izin. Namun sayangnya masih ada masyarakat yang enggan melakukan izin penggunaan jalan kepada Dinas Perhubungan Kota Mojokerto, baik karena tidak mengetahui prosedur atau bahkan memang sengaja tidak melakukan izin terlebih dahulu.

Dinas Perhubungan Kota Mojokerto juga mempunyai tugas untuk melakukan Survei Kepuasan Masyarakat (SKM) yang merupakan kegiatan pengukuran secara komprehensif tentang tingkat kepuasan masyarakat terhadap kualitas layanan yang diberikan oleh penyelenggara pelayanan publik. Selama ini proses penyebaran kuisioner SKM masih menggunakan cara manual, yaitu membagikan kepada masyarakat yang sedang berkunjung ke kantor dinas. Yang pada akhirnya banyak ditemukan kuisioner kosong atau tidak diisi oleh pengunjung. Sulitnya proses penyebaran kuisioner ini membuat data yang masuk pada SKM sangat sedikit, sehingga hasil dari perhitungan hasil survei akan kurang relevan.

Selain tugas – tugas yang telah diuraikan sebelumnya, Dinas Perhubungan Kota Mojokerto juga sering mendapatkan pengaduan keluhan dari masyarakat terkait permasalahan di bidang transportasi dan lalu lintas. Pengaduan dari masyarakat dimanfaatkan untuk mendapatkan masukan positif maupun negatif yang bertujuan untuk terus memperbaiki kualitas produk atau jasa dari suatu perusahaan atau seseorang. Kurangnya tempat atau sarana untuk melaporkan keluhan secara langsung kepada pemerintah menyebabkan pemerintah daerah kurang cepat dalam mengatasi keluhan dari masyarakat. Masyarakat sejauh ini menggunakan media sosial seperti *Twitter*, *Facebook* dan *blog-blog* di internet untuk melaporkan keluhan-keluhannya. Tentunya menggunakan media sosial tidak menghubungkan masyarakat secara langsung kepada pemerintah yang

mengakibatkan masyarakat tidak mendapat respon langsung dari pihak pemerintah yang bersangkutan.

Mengetahui permasalahan yang terjadi pada Dinas Perhubungan Kota Mojokerto, mengutip dari Peraturan Presiden Republik Indonesia Nomor 95 Tahun 2018 Tentang Sistem Pemerintahan Berbasis Elektronik (SPBE) menyatakan bahwa untuk mewujudkan tata kelola pemerintahan yang bersih, efektif, transparan, dan akuntabel serta pelayanan publik yang berkualitas dan terpercaya diperlukan sistem pemerintahan berbasis elektronik.

Dari permasalahan tersebut perlu diatasi dengan adanya suatu perangkat yang menjadi jembatan antara masyarakat dengan instansi pemerintah khususnya Dinas Perhubungan Kota Mojokerto secara langsung untuk masyarakat dapat mengurus izin penggunaan jalan, mengisi survei kepuasan masyarakat secara online dan dapat melaporkan keluhannya meliputi masalah transportasi, angkutan jalan dan lalu lintas menggunakan sebuah foto yang diambil menggunakan kamera dari sebuah *smartphone*. *Smartphone* dapat dimanfaatkan sebagai sarana yang tepat dengan melihat cukup banyak masyarakat yang memiliki *smartphone* mulai dari kalangan bawah sampai kalangan atas. Saat ini banyak produsen *smartphone* yang menjual produknya dengan harga yang relative murah adalah salah satu faktor pendukung pertumbuhan *smartphone* di Indonesia. Untuk kelompok negara berkembang, Indonesia berada di peringkat keenam di bawah Afrika Selatan, Brasil, Filipina, Meksiko, dan Tunisia (Taylor & Silver, 2019).

Perkembangan dari pengguna *smartphone* yang semakin meningkat dapat digunakan sebagai salah satu alasan untuk mengembangkan suatu perangkat lunak berbasis *Mobile* yang akan membantu baik masyarakat dan Dinas Perhubungan Kota Mojokerto dalam menyimpan data baik data izin penggunaan jalan, data survei kepuasan masyarakat dan data keluhan-keluhan yang didapat dari masyarakat. Diketahui sebuah perangkat *smartphone* memiliki GPS (*Global Positioning System*) yang merupakan sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika untuk mendapatkan lokasi dari pengguna. Lokasi sangat dibutuhkan dalam melaporkan keluhan masyarakat sehingga proses *Geotagging* akan membantu penyimpanan informasi lokasi pengguna.

Berdasarkan latar belakang tersebut maka dikembangkan suatu sistem yang dapat menyelesaikan masalah dengan judul "**Pengembangan Aplikasi *Geotagging* Pelaporan Keluhan Masyarakat pada Dinas Perhubungan Kota Mojokerto Menggunakan RESTful Web Services Berbasis *Mobile***". Sistem ini diharapkan dapat membantu melayani dan membuat masyarakat berperan aktif dalam melaporkan keluhan kepada Dinas Perhubungan Kota Mojokerto.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat diperoleh rumusan masalah sebagai berikut:

1. Bagaimana hasil analisis kebutuhan pada sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*?
2. Bagaimana hasil perancangan dan implementasi pada sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*?
3. Bagaimana hasil pengujian pada sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*?

## 1.3 Tujuan

Adapun tujuan dari penelitian ini berdasarkan rumusan masalah yang telah dipaparkan adalah sebagai berikut:

1. Menganalisis kebutuhan sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*.
2. Merancang dan mengimplementasi sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*.
3. Menguji sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*.

## 1.4 Manfaat

Adapun manfaat yang didapat dari penelitian ini berdasarkan tujuan masalah yang telah dipaparkan adalah sebagai berikut:

1. Memberikan sarana bagi masyarakat untuk melaporkan keluhan terkait dengan masalah Dinas Perhubungan Kota Mojokerto.
2. Memberikan data keluhan masyarakat bagi Dinas Perhubungan Kota Mojokerto untuk segera merespon pada keluhan masyarakat.

## 1.5 Batasan Masalah

Agar penelitian yang dilakukan tidak terlalu luas dan terarah, maka perlu adanya batasan masalah sebagai berikut:

1. Sistem dikembangkan pada perangkat *mobile* dengan teknologi *hybrid*.
2. Sistem harus terkoneksi dengan internet untuk bisa dioperasikan.

## 1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian skripsi ini disusun dari beberapa bab sebagai berikut:

## **BAB I PENDAHULUAN**

Pada bagian bab ini menguraikan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, waktu pelaksanaan dan tempat pelaksanaan, serta sistematika penulisan skripsi pada sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* ini.

## **BAB II LANDASAN KEPUSTAKAAN**

Pada bagian bab ini menguraikan dasar-dasar teori dasar yang digunakan dalam penyelesaian proses penulisan skripsi pada sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* yang diperoleh dari referensi yang ada.

## **BAB III METODOLOGI**

Pada bagian bab ini menguraikan tentang tahapan-tahapan dalam penyelesaian proses penulisan skripsi yang terdiri dari studi literatur, analisa kebutuhan, perancangan, implementasi, pengujian dan analisis, terakhir kesimpulan dan saran serta membahas tahapan dari pengembangan sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*.

## **BAB IV REKAYASA KEBUTUHAN**

Pada bagian bab ini menguraikan deskripsi umum yang berisikan kebutuhan fungsional, non-fungsional dari sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* dan pemodelan kebutuhan menggunakan *use case diagram* dan *use case scenario*.

## **BAB V PERANCANGAN DAN IMPLEMENTASI**

Pada bagian bab ini menguraikan tentang perancangan dan implementasi dari keseluruhan sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* mulai dari perancangan arsitektur, perancangan algoritme, perancangan antarmuka dan perancangan basis data dari sistem dan implementasi dari keseluruhan sistem yang dikembangkan mulai dari arsitektur sampai antarmuka sistem spesifikasi sistem, implementasi kode program, implementasi antarmuka dan implementasi basis data.

## **BAB VI PENGUJIAN**

Pada bagian bab ini menguraikan pengujian dari hasil perancangan dan implementasi yang telah dikerjakan menggunakan *black box testing* dan *white box testing* yang kemudian dianalisis hasilnya untuk memeriksa apakah sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* telah sesuai dengan kebutuhan yang dirancang.

## BAB VII PENUTUP

Pada bagian bab ini menguraikan tentang kesimpulan mengenai pengembangan sistem yang didapat dari perancangan, implementasi dan pengujian sistem pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* disertai dengan saran untuk pengembangan selanjutnya.



## BAB II LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Terdapat tiga kajian penelitian yang digunakan sebagai bahan untuk menunjang penyelesaian skripsi ini ditunjukkan pada Tabel 2. 1 dibawah ini.

Tabel 2. 1 Kajian Pustaka

No.	Judul Penelitian	Fokus Penelitian
1.	Perancangan dan Implementasi Aplikasi Penyimpanan Lokasi dan <i>Geotagging</i> Berbasis <i>Mobile</i> Menggunakan GPS pada Android (Thunesya Widi P, 2012)	Penelitian yang ditujukan untuk Wisatawan atau biasa disebut dengan traveller biasanya memiliki mobilitas tinggi yang membutuhkan sebuah aplikasi untuk menyimpan dan mengabadikan aktivitas dalam foto yang disertai data lokasi dimana foto diambil.
2.	Sistem Informasi Geografis Pelaporan Masyarakat (SIGMA) Berbasis Foto Geotag (Ary Mardani, 2014)	Penelitian ini membahas tentang pelaporan masyarakat pada pemerintah mengenai infrastruktur yang buruk dan layanan masyarakat yang kurang memadai dengan memanfaatkan teknologi <i>geotagging</i> .
3.	Rancang Bangun Aplikasi <i>Geotagging</i> Social Report Bencana Banjir (Wisnu Wijaya, 2018)	Penelitian yang berfokus pada pembuatan peta bencana dengan memanfaatkan pelaporan dari masyarakat dengan teknologi <i>geotagging</i> untuk melaporkan kejadian bencana banjir

Penelitian pertama dengan tema *geotagging* berjudul “Perancangan dan Implementasi Aplikasi Penyimpanan Lokasi dan *Geotagging* Berbasis *Mobile* Menggunakan GPS pada Android”. Aplikasi yang ditujukan untuk Wisatawan atau biasa disebut dengan traveller biasanya memiliki mobilitas tinggi yang membutuhkan sebuah aplikasi untuk menyimpan dan mengabadikan aktivitas dalam foto yang disertai data lokasi dimana foto diambil. Penelitian ini bertujuan mencari solusi dari permasalahan tersebut dengan berfokus pada efisiensi biaya dan mobilitas yang tinggi. Hasilnya, teknologi Android dipilih sebagai solusi yang tepat di dalam membangun sistem. Fitur GPS pada Android dapat digunakan secara maksimal untuk membuat perangkat lunak berbasis LBS (*Location Based Service*) (Prasetyo, et al., 2012).

Penelitian selanjutnya dengan tema *geotagging* berjudul “Sistem Informasi Geografis Pelaporan Masyarakat (SIGMA) Berbasis Foto Geotag”. Penelitian ini membahas tentang pelaporan masyarakat pada pemerintah mengenai infrastruktur yang buruk dan layanan masyarakat yang kurang memadai dengan memanfaatkan teknologi *geotagging* melalui *metadata* GPS yang terdapat dalam

gambar atau foto, untuk mendapatkan koordinat lokasi foto yang kemudian ditampilkan pada peta (Mardani, 2014).

Penelitian lain yang pernah dilakukan dengan tema *geotagging* berjudul “Rancang Bangun Aplikasi *Geotagging Social Report* Bencana Banjir”. Penelitian yang berfokus pada pembuatan peta bencana dengan memanfaatkan pelaporan dari masyarakat dengan teknologi geotagging untuk melaporkan kejadian bencana banjir dengan menyimpan titik lokasi yang kemudian dikelola menjadi suatu data informasi yang dapat digunakan untuk pembuatan sebuah peta kejadian bencana banjir (Wijaya, et al., 2018).

Perbedaan dari penelitian-penelitian sebelumnya yang telah dijabarkan adalah pada penelitian ini permasalahan yang diangkat yaitu pelaporan keluhan oleh masyarakat yang berada di lingkup pada Dinas Perhubungan Kota Mojokerto, yaitu sebuah perangkat lunak yang dapat digunakan masyarakat untuk memberikan informasi dan melaporkan keluhan-keluhan yang terjadi di bidang transportasi, angkutan jalan dan lalu lintas kepada Dinas Perhubungan Kota Mojokerto. Pada penelitian ini menggunakan masyarakat sebagai pemeran aktif untuk melaporkan semua keluhan dan masalah-masalah pada bidang yang terkait menggunakan *smartphone* yang disertai foto terkait keluhan untuk mewujudkan lingkungan masyarakat yang lebih baik. Penelitian yang dilakukan ini menggunakan RESTful *Web Services* sebagai komunikasi data dan teknologi yang sama dengan penelitian-penelitian yang telah dipaparkan sebelumnya yaitu *GPS* dan *geotagging* yang akan berjalan pada perangkat *Mobile* dengan pendekatan *hybrid*.

## 2.2 Pelaporan Keluhan Masyarakat

Pengaduan yang didapat dari masyarakat sangat penting bagi pemerintah atau instansi lain untuk mengetahui tingkat keberhasilan dalam melakukan tugasnya. Pengaduan dari masyarakat bertujuan untuk melakukan perbaikan kekurangan dari tugas yang dilaksanakan. Proses pelaporan berawal dari masyarakat yang melakukan pengaduan kepada instansi atau pemerintah yang dapat berupa pertanyaan, saran, atau keluhan. Kemudian masyarakat bisa melakukan pengaduan melalui berbagai media seperti melalui telepon ke instansi yang berhubungan, melalui website dari instansi terkait atau masyarakat juga bisa datang langsung ke instansi yang terkait (Prasetya, et al., 2013). Pelaporan keluhan masyarakat adalah pengaduan masyarakat atas masalah-masalah yang ada di sekitarnya. Laporan dapat disampaikan berbagai media, seperti media masa, telekomunikasi, atau media sosial. (Wiyanto, et al., 2017) mengungkapkan bahwa bentuk dari pelaporan keluhan antara lain sebagai berikut:

1. Melalui media telepon pada saat jam kerja.
2. Melalui faksimile kepada instansi terkait.
3. Melalui surat elektronik atau email kepada instansi terkait.
4. Melalui korespondensi atau surat.

5. Mengunjungi instansi secara langsung.

Selain itu laporan keluhan juga bisa dari hasil foto-foto yang diambil oleh masyarakat dan dikirim ke instansi terkait. Sehingga instansi terkait dapat menangani masalah yang terjadi dengan cepat.

### 2.3 Aplikasi Hybrid

Aplikasi *Hybrid* adalah sebutan untuk aplikasi *smartphone* yang dibuat dengan teknologi *web*. Dikatakan sebagai aplikasi *Hybrid* karena aplikasi tersebut merupakan kombinasi antara aplikasi *web* dan aplikasi *native*. Aplikasi *web* merupakan aplikasi yang berada di *server web* dan diakses menggunakan *web browser*, sedangkan aplikasi *native* merupakan aplikasi yang dibuat dengan Bahasa asli sistem operasi platform seperti iOS atau Android. Aplikasi *Hybrid* juga menggabungkan keuntungan yang ada pada aplikasi *web* yang multi platform dan keuntungan aplikasi *native* yang mampu mengakses fitur-fitur native pada *smartphone* (Abdulloh, 2018).

(Abdulloh, 2018) menyatakan aplikasi *hybrid* memiliki banyak keunggulan. Diantara keunggulan aplikasi *hybrid* dibandingkan dengan aplikasi *native* adalah sebagai berikut:

1. Cukup membuat aplikasi satu kali untuk digunakan oleh berbagai sistem operasi *smartphone*.
2. Menggunakan Bahasa yang mudah dipahami yaitu CSS3, HTML5 dan javascript yang sudah sangat familiar dengan bagi *web programmer*.
3. Biaya pengembangan lebih murah
4. Waktu yang diperlukan untuk pengembangan aplikasi lebih sedikit.

### 2.4 Global Positioning System

*GPS (Global Positioning System)* adalah suatu sistem navigasi yang berbasiskan satelit yang saling berhubungan milik Departemen Pertahanan Amerika Serikat (*Department of Defense*) yang diperkenalkan pada tahun 1978. Alat yang digunakan untuk menerima sinyal yang dikirim dari satelit GPS disebut *GPS receiver*. *GPS receiver* digunakan untuk dapat mengetahui posisi dari suatu benda atau seseorang. Posisi yang didapat akan dikonversikan menjadi titik-titik yang disebut *Way-point* yang pada akhirnya diubah menjadi titik koordinat lintang dan bujur dari posisi suatu benda atau seseorang yang akan ditampilkan pada peta elektronik (Mardani, 2014).

### 2.5 Geotagging

*Geotagging* adalah suatu proses menambahkan informasi *geospasial* pada berbagai macam media digital. Media yang telah mengalami proses *geotagging* akan mempunyai informasi *geospasial* seperti koordinat garis lintang (*latitude*) dan koordinat garis bujur (*longitude*) dan mungkin juga termasuk koordinat ketinggian (*altitude*). Hal tersebut memungkinkan media dapat didapatkan posisi

secara tepat pada peta (Joshi, et al., 2010). *Geotagging* dilakukan dengan cara menerjemahkan koordinat *latitude* dan koordinat *longitude* yang didapat menjadi sebuah alamat yang dimengerti oleh manusia atau biasa disebut teknik *geocoding*.

*Geotagging* bias digunakan untuk membantu pengguna untuk menemukan informasi spesifik posisi sesuatu dari suatu daerah. *Geotagging* dapat memberikan informasi terkait lokasi dari suatu benda atau seseorang berdasarkan media digital dan sumber lainnya.

## 2.6 RESTful Web Services

REST (*Representational State Transfer*) adalah arsitektur aplikasi perangkat lunak yang dimodelkan setelah data direpresentasikan, diakses, dan di modifikasi di web. Dalam arsitektur REST, data dan fungsionalitas dianggap *resources*, dan *resources* ini diakses menggunakan URIs (*Uniform Resource Identifiers*), biasanya tautan di web. *Resources* ditindaklanjuti dengan menggunakan serangkaian operasi sederhana, terdefinisi dengan baik. Arsitektur REST pada dasarnya adalah arsitektur *client-server*, dan dirancang untuk menggunakan protokol komunikasi *stateless*, biasanya HTTP (*Hypertext Transfer Protocol*). Dalam arsitektur REST, klien dan server bertukar representasi *resources* menggunakan antarmuka dan protokol standar yang pada umumnya berformat JSON, atau XML. Aplikasi yang menggunakan prinsip-prinsip dari REST dapat disebut dengan "RESTful" menjadikannya sederhana, ringan, dan memiliki kinerja tinggi (Hamad, et al., 2010).

RESTful *Web Services* adalah aplikasi web yang dibangun di atas arsitektur REST. Mereka mengekspos *resources* (data dan fungsionalitas) melalui web URI, dan menggunakan empat metode HTTP utama untuk membuat, mengambil, memperbarui, dan menghapus *resources*. Layanan web RESTful biasanya memetakan empat metode HTTP utama ke tindakan CRUD: *create*, *retrieve*, *update*, dan *delete* yang ditunjukkan pada Tabel 2. 2.

Tabel 2. 2 HTTP Method  
Sumber: (Hamad, et al., 2010)

HTTP Method	CRUD Action
GET	<i>Retrieve data</i>
POST	<i>Create data</i>
PUT	<i>Update data</i>
DELETE	<i>Delete data</i>

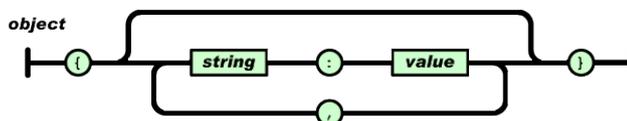
## 2.7 Javascript Object Notation

Javascript *Object Notation* (JSON) merupakan sebuah format pertukaran data yang mampu di kenali oleh berbagai *platform*. Format JSON dibuat dari bagian bahasa pemrograman javascript. JSON adalah format text yang tidak bergantung pada bahasa pemrograman manapun karena bahasa yang digunakan

adalah bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, Javascript, Perl, Python, dll. Maka dari itu, JSON ideal sebagai bahasa pertukaran-data (JSON, 1999).

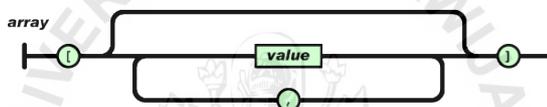
JSON memiliki bentuk sebagai berikut:

1. Objek, objek adalah pasangan *key value* yang tidak beraturan. Berikut struktur object JSON dapat dilihat di Gambar 2. 1.



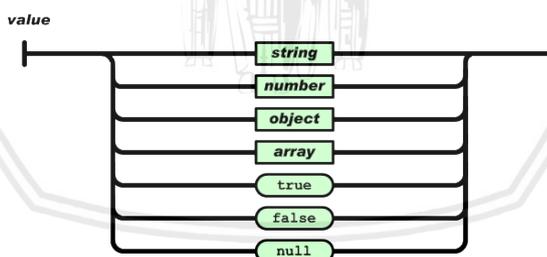
Gambar 2. 1 Struktur Object JSON  
Sumber: (JSON, 1999)

2. *Array*, *array* adalah beberapa *value* yang saling berurutan. Struktur array JSON dapat dilihat di Gambar 2. 2.



Gambar 2. 2 Struktur Array JSON  
Sumber: (JSON, 1999)

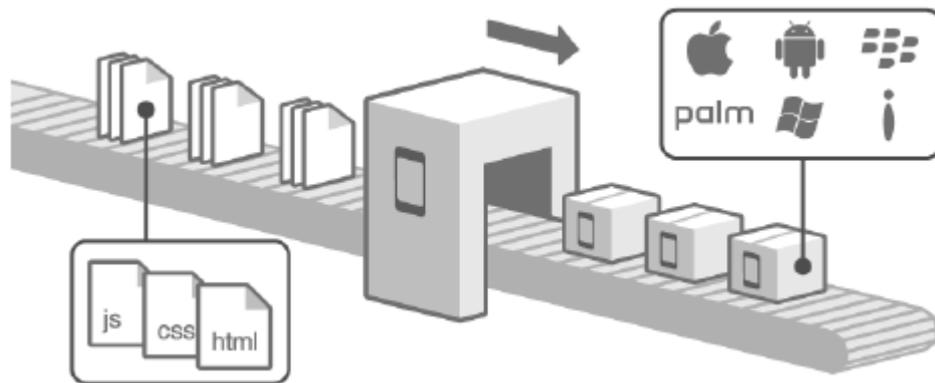
3. *Value*, *value* adalah sebuah nilai dengan tipe data tertentu, sebuah objek ataupun sebuah *array*. Berikut struktur value JSON dapat dilihat di Gambar 2. 3.



Gambar 2. 3 Struktur Value JSON  
Sumber: (JSON, 1999)

## 2.8 Apache Cordova

Apache Cordova merupakan sebuah *framework open source* yang digunakan untuk pengembangan aplikasi *mobile* yang menggunakan teknologi web standar seperti HTML5, CSS3, dan Javascript untuk pengembangan *cross-platform*. Aplikasi cordova akan dikemas menggunakan SDK (*Software Development Kit*) dari masing-masing *platform* sebagai paket aplikasi. Sebagai contoh jika SDK yang diinginkan adalah android maka harus dikemas menggunakan SDK android, demikian pula dengan *platform* yang lainnya (Tompoh, et al., 2016), ilustrasi dari cara kerja Apache Cordova ditunjukkan pada Gambar 2. 4.



Gambar 2. 4 Cara Kerja Apache Cordova  
Sumber: (Tompoh, et al., 2016)

## 2.9 Framework7

Framework7 merupakan sebuah *framework open source* dan terbuka untuk mengembangkan aplikasi *mobile hybrid*, *desktop* atau *web* dengan tampilan dan nuansa *native*. Framework7 juga bias diginakan sebagai alat *prototyping* yang menunjukkan prototipe aplikasi yang berfungsi secepat mungkin jika perlu dan bisa digunakan dengan alat (*tools*) atau *text editor* apa pun saat bekerja dengan Framework7. Framework7 digunakan untuk membuat aplikasi iOS & Android dengan HTML, CSS, dan JavaScript dengan mudah dan jelas. Framework7 tidak memaksa untuk menggunakan apa pun kecuali HTML, CSS, dan JavaScript biasa. Membuat aplikasi dengan Framework7 semudah membuat situs *web*. Dan Selain pustaka Vanilla JavaScript, Framework7 dilengkapi dengan komponen Vue.js & React untuk menghadirkan komponen-sintaks, data terstruktur dan pengikatan data (*data binding*) dengan kekuatan dan kesederhanaan Vue.js & React (Kharlampidi, 2019).

## 2.10 VueJS

VueJS merupakan salah satu *framework* modern berbasis Javascript yang berfokus pada layer *view* untuk membangun antarmuka pengguna pada aplikasi web. Dalam penggunaannya, VueJS dapat dengan mudah diintegrasikan dengan *framework* atau *library* Javascript lainnya. VueJS juga mampu secara sempurna menjalankan *Single-Page Application* yang canggih (Abdullah, 2018).

(Abdullah, 2018) menyatakan bahwa salah satu faktor yang mendukung popularitas VueJS adalah beberapa keunggulan yang dimiliki oleh *framework* ini. Berikut beberapa keunggulan yang dimiliki oleh VueJS:

1. Memiliki struktur skrip yang sederhana sehingga memudahkan pengguna pemula untuk cepat memahaminya.
2. Memiliki *performance* yang sangat baik yang akan berpengaruh pada kecepatan aplikasi.

3. Memiliki dokumentasi yang baik, sehingga memudahkan pemula dalam mempelajarinya.
4. VueJS didukung oleh Laravel, bahkan telah banyak pembahasan mengenai integrasi Laravel dan VueJS, sehingga cocok digunakan bersama Laravel.
5. VueJS mudah diintegrasikan dengan *library* Javascript lainnya seperti JQuery.

## 2.11 NodeJS

NodeJS merupakan sebuah platform yang juga disebut sebagai *runtime environment* yang mirip seperti Ruby's *Event Machine* atau Python's *Twisted* digunakan untuk membangun *real-time application*. NodeJS berasal dari bahasa campuran C++ dan juga Javascript, yang memungkinkan untuk membuat *event-driven* dalam javascript dan *Asynchronous I/O* agar meminimalkan *overhead* dan memaksimalkan skalabilitas terutama *web server*. NodeJS dieksekusi sebagai aplikasi *server*. NodeJS yang dapat berjalan pada *web browser* karena adanya *V8 Engine* yang dibuat google dengan beberapa modul bawaan yang sudah terintegrasi seperti modul *http*, *file system*, *security*, dan modul lainnya (Fajrin, 2017).

## 2.12 Rekayasa Perangkat Lunak

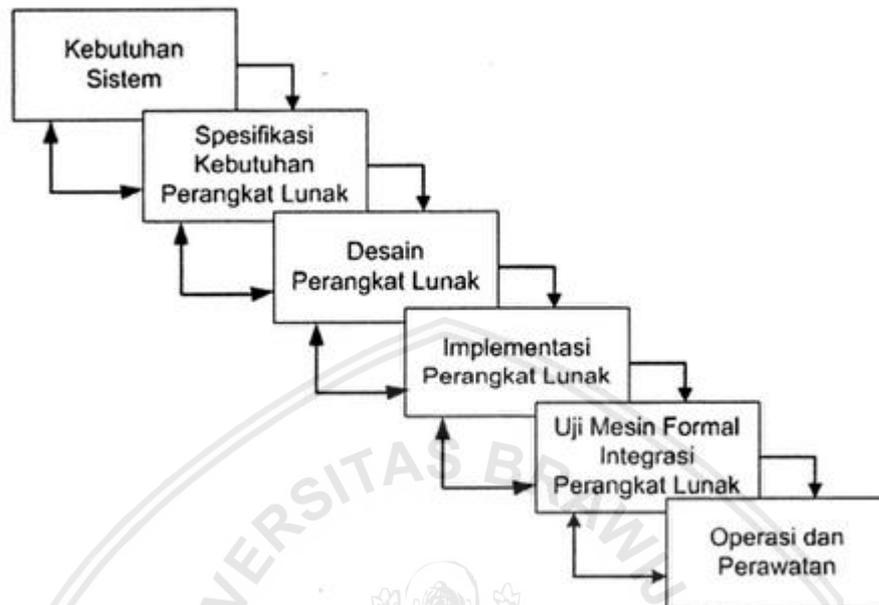
Rekayasa perangkat lunak adalah sebuah alur proses yang berkaitan dengan pengembangan dan pemeliharaan perangkat lunak dengan menerapkan teknologi dan praktik dari ilmu komputer, manajemen proyek, dan bidang-bidang lainnya. Banyak perangkat lunak yang diharapkan dapat melakukan pekerjaan dengan baik dalam menghadapi perubahan kondisi. Semua perangkat lunak juga membutuhkan keandalan yang tinggi dan harus dihasilkan secara ekonomis. Teknik rekayasa perangkat lunak akan meningkatkan fungsionalitas dan efisiensi aplikasi dan juga kemudahan dan efisiensi dari pengembang perangkat lunak (Simarmata, 2010).

## 2.13 Software Development Life Cycle

SDLC (*Software Development Life Cycle*) atau Siklus Hidup Sistem dalam rekayasa perangkat lunak adalah proses dari pengembangan sistem serta model dan metodologi yang digunakan dalam mengembangkan sistem-sistem tersebut. SDLC merupakan sebuah pola-pola yang diambil untuk mengembangkan perangkat lunak, yang terdiri dari tahap-tahap: perencanaan, analisis, desain, implementasi, uji coba dan pengelolaan (Susanto & Andriana, 2016).

Pada penelitian ini model SDLC yang digunakan adalah model *waterfall*, yang merupakan model pengembangan perangkat lunak yang paling sering digunakan. Model pengembangan ini bersifat linear dari proses paling awal pengembangan sistem yaitu perencanaan sampai proses paling akhir pengembangan sistem yaitu proses pemeliharaan. Setiap proses akan dilakukan secara runtut, proses berikutnya tidak bias dilaksanakan sebelum proses

sebelumnya selesai dilaksanakan dan tidak bisa kembali ke proses sebelumnya. Tahapan dari model pengembangan waterfall ini digambarkan pada Gambar 2. 5.



Gambar 2. 5 Model Pengembangan *Waterfall*  
Sumber: (Simarmata, 2010)

## 2.14 Unified Modelling Language

UML (*Unified Modelling Language*) adalah suatu alat bantu yang sangat penting di dunia rekayasa sistem yang berorientasi obyek. Karena UML menyediakan Bahasa pemodelan visual yang digunakan oleh pengembang sistem untuk membuat *blueprint* (cetak biru) atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2005).

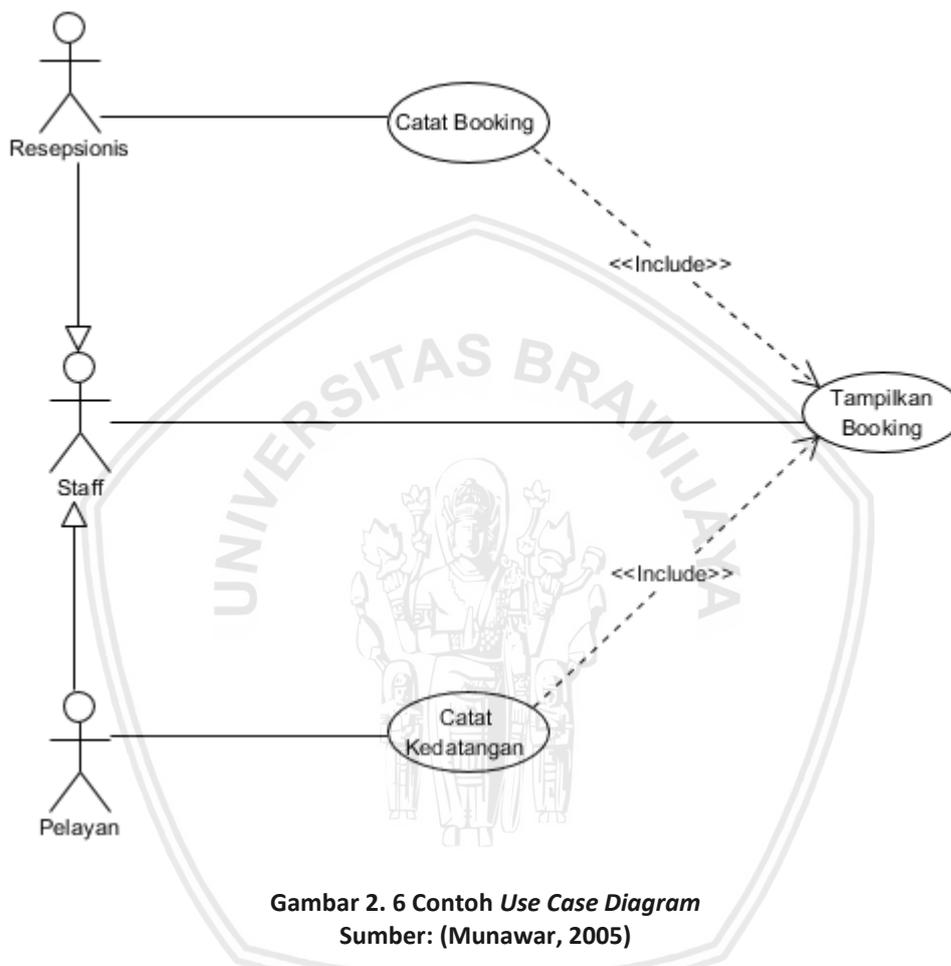
Hasil dari pemodelan tersebut merupakan pengertian dari sistem. Penggunaan Satu model UML saja tidak cukup untuk mendeskripsikan sistem secara keseluruhan, maka dibutuhkan banyak model yang saling berhubungan untuk menggambarkan suatu sistem. UML digunakan untuk menggambarkan perancangan awal dari sistem yang akan dibangun.

Pada penelitian sistem pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* ini, pemodean UML yang digunakan adalah diagram *use case*, diagram *sequence*, diagram *class*.

### 2.14.1 Use Case Diagram

Diagram *Use Case* adalah pemodelan untuk mendeskripsikan perilaku (*behaviour*) dari aplikasi yang akan dibuat. *Use case* menggambarkan sebuah

interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dapat dikatakan use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Urva & Siregar, 2015) seperti contoh *use case diagram* yang ditunjukkan pada Gambar 2. 6.

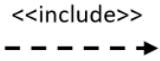
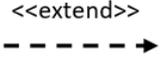


Gambar 2. 6 Contoh Use Case Diagram  
Sumber: (Munawar, 2005)

Adapun penjelasan untuk simbol-simbol yang terdapat pada *use case diagram* dapat dilihat dalam Tabel 2. 3 Penjelasan Simbol Use Case Diagram.

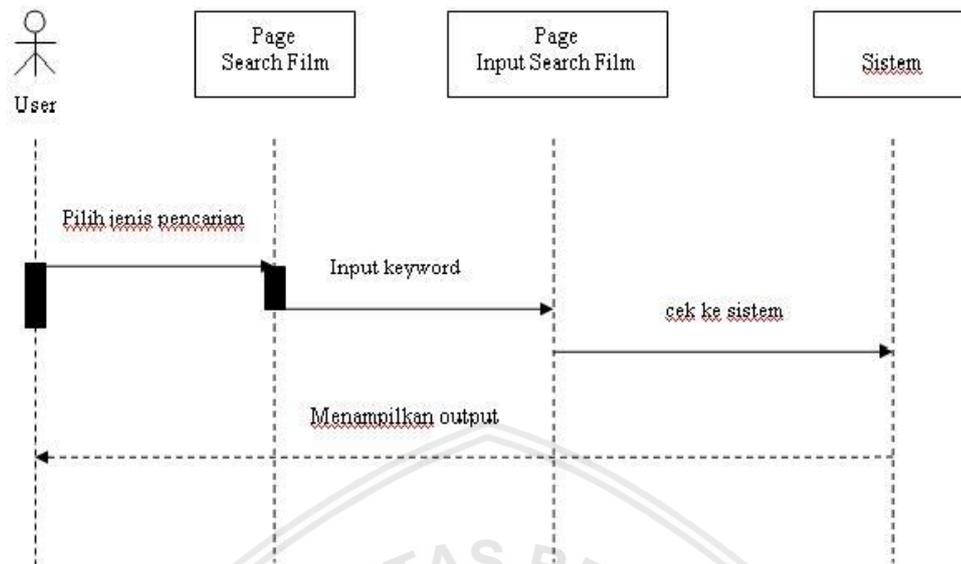
Tabel 2. 3 Penjelasan Simbol Use Case Diagram

No	Gambar	Nama	Keterangan
1		Actor	Merepresentasikan peran yang digunakan saat melakukan interaksi dengan use case.

No	Gambar	Nama	Keterangan
2		<i>Include</i>	Menunjukkan suatu use case seluruhnya adalah fungsionalitas dari use case lainnya.
3		<i>extend</i>	Menunjukkan suatu use case adalah tambahan fungsionalitas dari use case asalnya jika suatu kondisi terpenuhi.
4		<i>Association</i>	Relasi antara <i>actor</i> dengan use case
5		<i>System</i>	Menspesifikan sebuah lingkup sistem yang menampilkan sistem secara terbatas.
6		<i>Use case</i>	Daftar kebutuhan dari sebuah system yang terstruktur untuk berinteraksi dengan aktor.
7		<i>Entity</i>	Elemen yang bertanggungjawab menyimpan data atau informasi.

### 2.14.2 Sequence Diagram

Diagram *Sequence* menggambarkan perilaku objek pada use case dengan mendeskripsikan waktu hidup (*life-line*) objek dan pesan yang dikirimkan dan diterima antar objek (Urva & Siregar, 2015) seperti contoh yang ditunjukkan pada Gambar 2. 7.



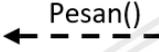
Gambar 2. 7 Contoh Sequence Diagram

Sumber: (Munawar, 2005)

Adapun penjelasan untuk simbol-simbol yang terdapat pada *sequence diagram* dapat dilihat dalam Tabel 2. 4.

Tabel 2. 4 Penjelasan Simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1		Actor	<ul style="list-style-type: none"> <li>Pengguna, proses, atau sistem lain yang berinteraksi dengan sistem informasi dan mendapatkan manfaat dari sistem.</li> <li>Berpartisipasi secara beruntun dengan cara menerima atau mengirim pesan.</li> </ul>
2		Object	<ul style="list-style-type: none"> <li>Berpartisipasi secara beruntun dengan cara menerima atau mengirim pesan.</li> </ul>
3		Lifeline	<ul style="list-style-type: none"> <li>Menandakan panjang kehidupan suatu objek</li> </ul>

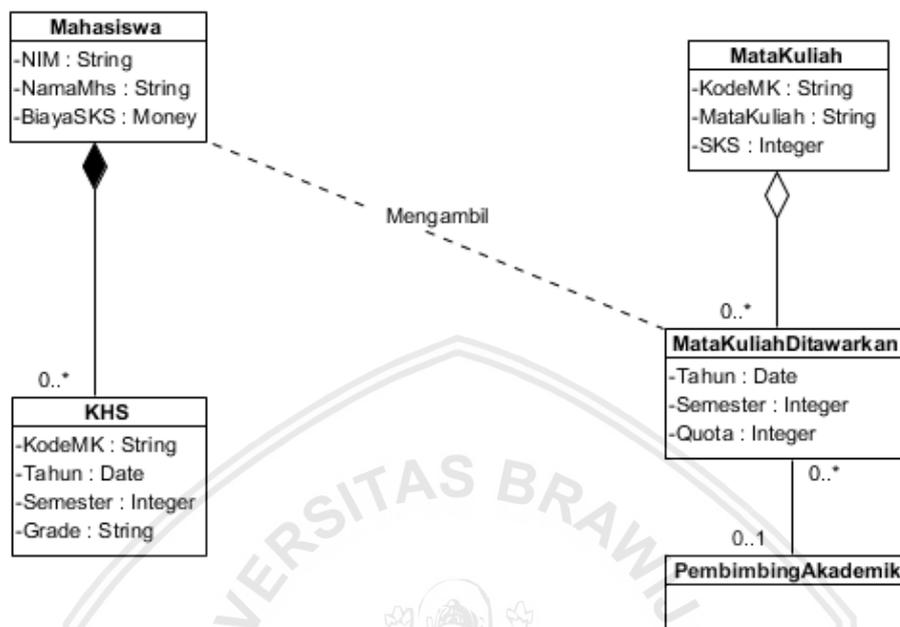
No	Gambar	Nama	Keterangan
4		<i>Activation</i>	Menandakan objek sedang aktif baik menerima atau mengirim pesan.
5		<i>Objek Message</i>	Objek mengirim satu pesan ke objek lainnya
6		<i>Return Message</i>	Hasil dari suatu kembalian dari objek tertentu, arah panah mengarah ke objek yang menerima
7		<i>Message to self</i>	Menggambarkan pesan yang dilakukan pada objek itu sendiri
8		<i>Boundary</i>	Digunakan untuk menggambarkan view
9		<i>Control Class</i>	Digunakan untuk menghubungkan boundary dengan tabel
10		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan

### 2.14.3 Class Diagram

Diagram *Class* mendeskripsikan hubungan antar kelas dan penjelasan detail dari setiap kelas, juga memperlihatkan aturan dan tugas dari entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek



yang dikoneksikan (Urva & Siregar, 2015). Berikut contoh class diagram dapat dilihat pada Gambar 2. 8.



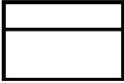
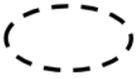
Gambar 2. 8 Contoh *Class Diagram*  
 Sumber: (Munawar, 2005)

Adapun penjelasan untuk simbol-simbol yang terdapat pada *class diagram* dapat dilihat dalam Tabel 2. 5.

Tabel 2. 5 Penjelasan Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) mempunyai perilaku dan struktur objek induk ( <i>ancestor</i> ).
2		Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.



No	Gambar	Nama	Keterangan
3		Class	Himpunan dari objek-objek yang mempunyai atribut dan operasi dari masing-masing objek.
4		Collaboration	Deskripsi dari urutan aksi yang ditampilkan dan menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
5		Realization	Operasi yang dilakukan oleh suatu objek.
6		Dependency	Hubungan yang ketika terjadi perubahan pada suatu elemen mandiri makan akan mempengaruhi elemen yang bergantung padanya.
7		Association	Hubungan yang menghubungkan objek satu dan objek yang lain.

## 2.15 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses untuk mencari kesalahan pada setiap *item* perangkat lunak, mencatat hasilnya, mengevaluasi setiap aspek pada setiap komponen (sistem) dan mengevaluasi fasilitas-fasilitas dari perangkat lunak yang akan dikembangkan (Wibisono & Baskoro, 2002).

Proses pengujian tidak terpisahkan dari perangkat lunak. Proses pengujian juga mempengaruhi masa penggunaan perangkat lunak. Semakin detail proses pengujian yang dilakukan, semakin lama juga rentang waktu yang dibutuhkan untuk satu pemeliharaan perangkat lunak dan untuk proses selanjutnya. Pengujian perangkat lunak dilakukan pada setiap tahapan pengembangan hingga pada pemeliharaan perangkat lunak. Cara pandang terhadap perangkat lunak berkembang menjadi lebih konstruktif. Pengujian tidak lagi dipandang sebagai aktivitas yang hanya dilakukan setelah pengkodean perangkat lunak selesai

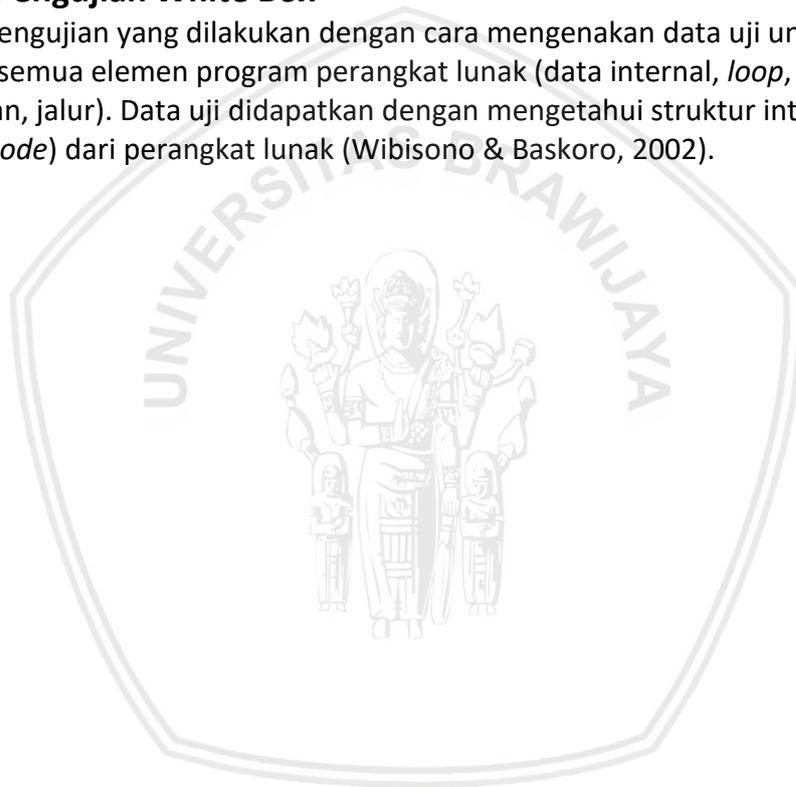
dengan batasan sebagai pendeteksi kegagalan perangkat lunak, melainkan sebagai aktivitas yang menuntun keseluruhan proses pengembangan perangkat lunak dan pemeliharaan. Pengujian pun menjadi bagian penting dari suatu konstruksi perangkat lunak (Simarmata, 2010).

### **2.15.1 Pengujian *Black Box***

Pengujian yang dilakukan dengan cara menjalankan data uji dan melakukan pengecekan apakah fungsional perangkat lunak bekerja dengan baik. Data uji didapatkan dari spesifikasi perangkat lunak, yang dalam hal ini menjelaskan fungsional perangkat lunak (Wibisono & Baskoro, 2002).

### **2.15.2 Pengujian *White Box***

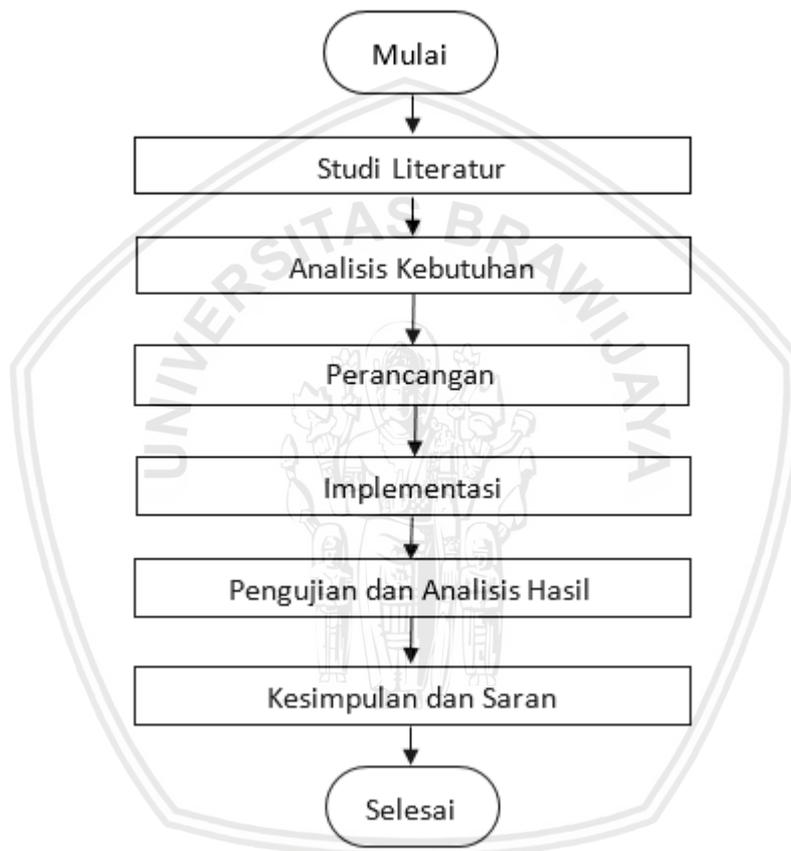
Pengujian yang dilakukan dengan cara mengenakan data uji untuk menguji semua elemen program perangkat lunak (data internal, *loop*, logika keputusan, jalur). Data uji didapatkan dengan mengetahui struktur internal (*source code*) dari perangkat lunak (Wibisono & Baskoro, 2002).



## BAB III METODOLOGI

### 3.1 Struktur Tahap Pelaksanaan

Pada bagian ini akan menjelaskan langkah-langkah dari metodologi yang berkaitan dalam proses pengembangan sistem pelaporan keluhan masyarakat. Penjelasan tersebut diantaranya adalah studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian, kemudian terakhir kesimpulan dan saran. Langkah-langkah tersebut diilustrasikan pada Gambar 3. 1 dibawah ini.



Gambar 3. 1 Diagram Flow Pengembangan

### 3.2 Studi Literatur

Studi Literatur menjelaskan teori yang digunakan untuk menunjang penyelesaian permasalahan yang sesuai dengan dasar teori dalam topik pengembangan yang dilakukan. kegiatan ini meliputi mengkaji teori yang diperoleh dari buku, jurnal, atau penelitian lainnya yang dilakukan sebelumnya. berdasarkan topik yang pembahasan pada penelitian ini. Dasar teori yang meliputi penelitian ini adalah sebagai berikut:

1. Pelaporan Keluhan Masyarakat
2. Aplikasi *Hybrid*
3. Global Positioning System

4. Geotagging
5. RESTful *Web Services*
6. Javascript Object Notation
7. Apache Cordova
8. Framework7
9. VueJS
10. NodeJS
11. Rekayasa Perangkat Lunak
12. *Software Development Life Cycle*
13. *Unified Modelling Language*
14. Pengujian Perangkat Lunak

### 3.3 Rekayasa Kebutuhan

Rekayasa kebutuhan ini menjelaskan tahapan untuk mengetahui informasi apa yang dibutuhkan dalam pengembangan sistem pelaporan keluhan masyarakat ini. Pada penelitian ini model SDLC (*Software Development Life Cycle*) yang digunakan adalah model *waterfall*. Adapun cara mendapatkan kebutuhan sistem yaitu melalui elisitasi kebutuhan, identifikasi aktor, spesifikasi kebutuhan, dan pemodelan kebutuhan.

#### 3.3.1 Elisitasi kebutuhan

Elisitasi kebutuhan dilakukan untuk mengumpulkan kebutuhan dari pihak yang berhubungan dengan sistem pelaporan keluhan masyarakat. Elisitasi kebutuhan dilakukan dengan melakukan wawancara kepada Ibu Ulfa Rahmawati selaku Pengelola Kepegawaian pada Dinas Perhubungan Kota Mojokerto. Wawancara ini bertujuan untuk mengetahui permasalahan yang ada pada Dinas Perhubungan Kota Mojokerto khususnya permasalahan yang berhubungan dengan masyarakat. Selain itu, untuk mendukung pengembangan sistem juga dikumpulkan dokumen - dokumen yang mendukung selama kegiatan tersebut dilakukan. Hasil elisitasi kebutuhan akan dimodelkan dalam bentuk proses bisnis *as is* dan menjadi dasar dari rekayasa kebutuhan.

#### 3.3.2 Identifikasi Aktor

Identifikasi aktor merupakan suatu proses mengidentifikasi hasil elisitasi kebutuhan yang dilakukan. Dengan memahami proses bisnis, *stakeholder* dan alur proses bisnis maka dapat diketahui siapa saja *stakeholder* yang terlibat yang kemudian diidentifikasi menjadi aktor yang terlibat langsung dengan sistem.

#### 3.3.3 Spesifikasi Kebutuhan

Spesifikasi kebutuhan merupakan spesifikasi terperinci dari kebutuhan sistem dan harus mencakup kebutuhan pengguna untuk suatu sistem. Tahap ini menggambarkan daftar kebutuhan yang bisa dilakukan oleh sistem. Tahap ini dilakukan dengan cara melakukan wawancara, mengambil artefak berupa form - form dan pengamatan terhadap sistem pelaporan keluhan yang lain. Hasil dari

proses ini berupa kalimat kebutuhan, baik kebutuhan fungsional maupun kebutuhan non-fungsional.

### 3.3.4 Pemodelan Kebutuhan

Pemodelan kebutuhan adalah proses untuk menggambarkan bagaimana aktor berinteraksi dengan sistem. Proses ini dilakukan dengan melakukan identifikasi fungsionalitas, aktor dan hubungan antara aktor dan sistem. Wawancara juga dilakukan dalam rangka untuk mengetahui apa saja langkah - langkah dalam sebuah fungsionalitas. Hasil dari proses ini adalah berupa *use case* diagram dan *use case scenario*.

## 3.4 Perancangan

Perancangan berisikan tentang rancangan dari sistem dengan tujuan setelah mendapatkan gambaran dengan jelas apa yang harus dikerjakan melalui proses rekayasa kebutuhan. Hasil dari perancangan ini berupa perancangan arsitektur, perancangan komponen dan perancangan data.

### 3.4.1 Perancangan Arsitektur

Perancangan arsitektur aplikasi pelaporan keluhan masyarakat ini berfokus pada perancangan menggunakan UML (*Unified Modelling Language*) yaitu *Sequence* Diagram dan *Class* Diagram untuk menunjukkan gambaran bagaimana elemen fungsional perangkat lunak disusun, diorganisasi dan distrukturkan. Diagram *sequence* didapatkan dari proses identifikasi objek, *methods* serta tahapan dalam *use case scenario*. Diagram *class* didapatkan dari proses identifikasi objek - objek yang memiliki kesamaan sifat (*attribute, methods*) serta hubungan antar objek.

### 3.4.2 Perancangan Komponen

Perancangan komponen berisi perancangan setiap komponen secara detail dari sistem pelaporan keluhan masyarakat meliputi perancangan *pseudocode* atau algoritme dan perancangan antarmuka dari sistem. Perancangan *pseudocode* didapat dari pemikiran bagaimana membuat sebuah fungsionalitas pada sebuah unit untuk mencapai tujuan tertentu. Perancangan antarmuka didapat dari proses pengamatan pada beberapa aplikasi sejenis dan pemahaman tentang spesifikasi kebutuhan.

### 3.4.3 Perancangan Data

Perancangan data pada sistem pelaporan keluhan masyarakat ini berfokus pada perancangan data yang didapat dari proses identifikasi dan analisis kebutuhan sistem dan pengguna yang didapat dari proses wawancara menjadi sebuah struktur data yang dibutuhkan dalam proses implementasi. Hasil dari tahap perancangan data ini adalah sebuah model untuk menjelaskan hubungan antar data dalam basis data menggunakan diagram ERD (*Entity Relationship Diagram*).

## 3.5 Implementasi

Implementasi adalah proses penerapan dari hasil perancangan kedalam sistem meliputi implementasi kode program, implementasi antarmuka, implementasi basis data.

### 3.5.1 Implementasi Kode Program

Implementasi kode program adalah tahap implementasi algoritme atau *pseudocode* dari proses perancangan menjadi sebuah kode program sesuai dengan bahasa pemrograman yang digunakan dalam pembuatan sistem. Dalam sistem ini nantinya terdiri dari *front-end* yang menggunakan VueJS dan *back-end* menggunakan NodeJS.

### 3.5.2 Implementasi Antarmuka

Implementasi antarmuka adalah proses penerapan antarmuka atau *user interface* yang didapatkan dari perancangan antarmuka yang telah dirancang sebelumnya pada proses perancangan komponen. Proses implementasi antarmuka menghasilkan sebuah antarmuka aplikasi yang dapat berinteraksi langsung dengan pengguna.

### 3.5.3 Implementasi Basis Data

Implementasi basis data adalah proses penerapan struktur data yang telah dirancang pada tahap perancangan data dan diimplementasikan menggunakan sistem manajemen basis data MySQL.

## 3.6 Pengujian

Pengujian adalah tahap yang dilakukan untuk mendeteksi *error* atau *bug* serta untuk mengetahui hasil yang telah diimplementasikan. Dengan ini dapat meminimalisir terjadinya kesalahan sebelum sistem digunakan. Dalam tahap pengujian ini digunakan empat metode pengujian yaitu pengujian *white box* yang menggunakan pengujian unit dan pengujian integrasi, pengujian *black box* yang menggunakan pengujian validasi dan pengujian *compatibility*.

### 3.6.1 Pengujian Unit

Pengujian unit merupakan pengujian yang menerapkan metode *white box*. Pengujian ini dilakukan dengan menguji kode program dengan menggunakan *basis path testing* untuk memeriksa jalur algoritme yang kemudian akan menghasilkan gambar *flowgraph* sehingga mendapatkan jumlah *node*, *edge* dan *predicate node* yang digunakan untuk menghitung *cyclomatic complexity* untuk mengetahui tingkat kerumitan sistem yang menghasilkan *independent path*. *Independent path* selanjutnya akan dibuatkan kasus uji pada setiap jalur yang ada untuk mengetahui apakah jalur tersebut memberikan hasil yang sesuai dengan hasil yang diinginkan.

### 3.6.2 Pengujian Integrasi

Pengujian Integrasi merupakan pengujian yang menerapkan metode *white box*. Pengujian ini dilakukan dengan cara yang hampir sama dengan pengujian unit akan tetapi pengujian ini dilakukan penggabungan dari dua atau lebih unit pada perangkat lunak. Pengujian ini akan dilakukan dengan cara menggabungkan dua atau lebih unit. Penggabungan unit akan menghasilkan algoritme dan *flowgraph* baru yang merupakan hasil penggabungan, kemudian dilakukan serangkaian perhitungan *cyclomatic complexity*, penentuan *independent path* dan pembuatan kasus uji pada setiap jalur.

### 3.6.3 Pengujian Validasi

Pengujian validasi dilakukan untuk memastikan bahwa keseluruhan fungsional sistem sudah sesuai dengan kebutuhan fungsional yang telah dijelaskan sebelumnya. Pengujian ini dilakukan dengan menggunakan metode *black box*. Pengujian ini dilakukan dengan cara membuat serangkaian prosedur kasus uji yang kemudian dibandingkan antara hasil ekspektasi dengan hasil yang didapat. Jika hasil ekspektasi memiliki kesesuaian dengan hasil yang didapat, maka akan diberikan status *valid*. Kasus uji pada pengujian kebutuhan fungsional didapatkan berdasarkan skenario *use case* yang telah dibuat.

### 3.6.4 Pengujian *Compatibility*

Pengujian *Compatibility* merupakan salah satu pengujian non fungsional yang digunakan sebagai pengujian untuk menentukan suatu *Environment* yang diharapkan dapat menjalankan sistem yang telah dikembangkan. Semakin sistem bisa digunakan di banyak jenis perangkat yang berbeda, menunjukkan semakin baik aspek kompatibilitasnya. Pengujian *compatibility* dilakukan dengan menguji setiap perangkat dengan beberapa kasus uji dan prosedur pengujian yang sama kemudian dibandingkan antara hasil ekspektasi dengan hasil yang didapat dari setiap perangkat. Jika hasil ekspektasi memiliki kesesuaian dengan hasil yang didapat pada setiap perangkat, maka akan diberikan status *valid*. Kasus uji pada pengujian *compatibility* didapatkan dari *use case scenario* yang telah dibuat.

## 3.7 Kesimpulan dan Saran

Kesimpulan dan saran pada tahap ini dilakukan saat seluruh tahapan sudah terselesaikan. Pengambilan kesimpulan didapat dari segala proses yang telah dilakukan dan berisikan hasil dari penarikan kesimpulan sebagai jawaban atas rumusan masalah yang dibuat, sedangkan saran disajikan pada saat evaluasi untuk memperbaiki kekurangan dari penelitian tersebut.

## BAB IV REKAYASA KEBUTUHAN

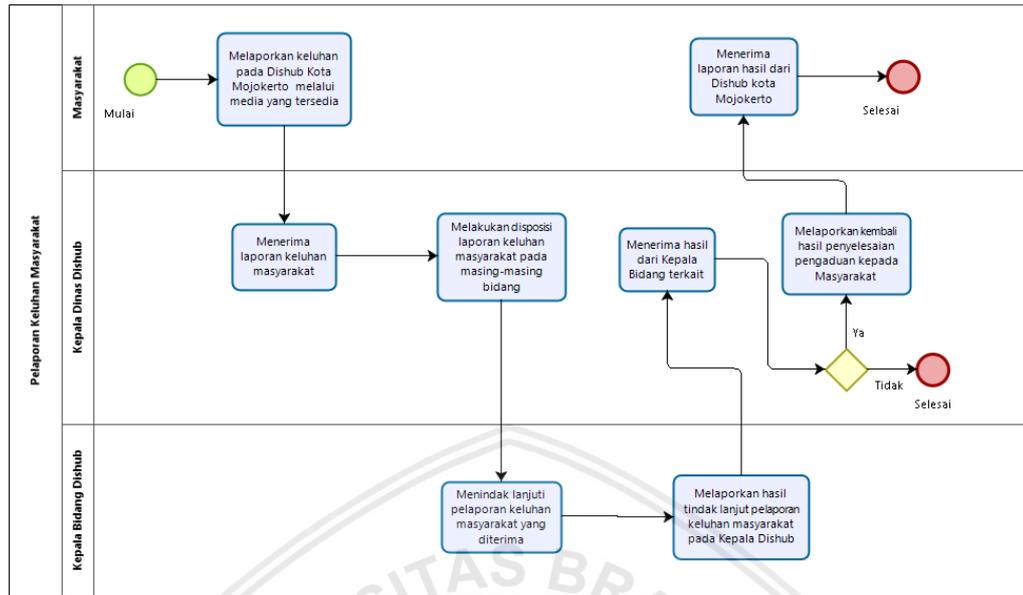
### 4.1 Analisis Kebutuhan

Tahap rekayasa kebutuhan dilakukan untuk mengidentifikasi seluruh kebutuhan yang didapatkan pengguna pada aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*. Proses dari rekayasa kebutuhan menjelaskan elisitasi kebutuhan, gambaran umum sistem, aturan penomoran, kebutuhan fungsional dan non-fungsional, pemodelan kebutuhan dengan *use case* diagram dan *use case scenario*. Deskripsi dan analisis kebutuhan fungsional sangat diperlukan agar penulis bisa memberikan gambaran sistem yang dibuat serta menjadi dokumentasi yang akan membantu pengembangan sistem selanjutnya.

#### 4.1.1 Elisitasi Kebutuhan

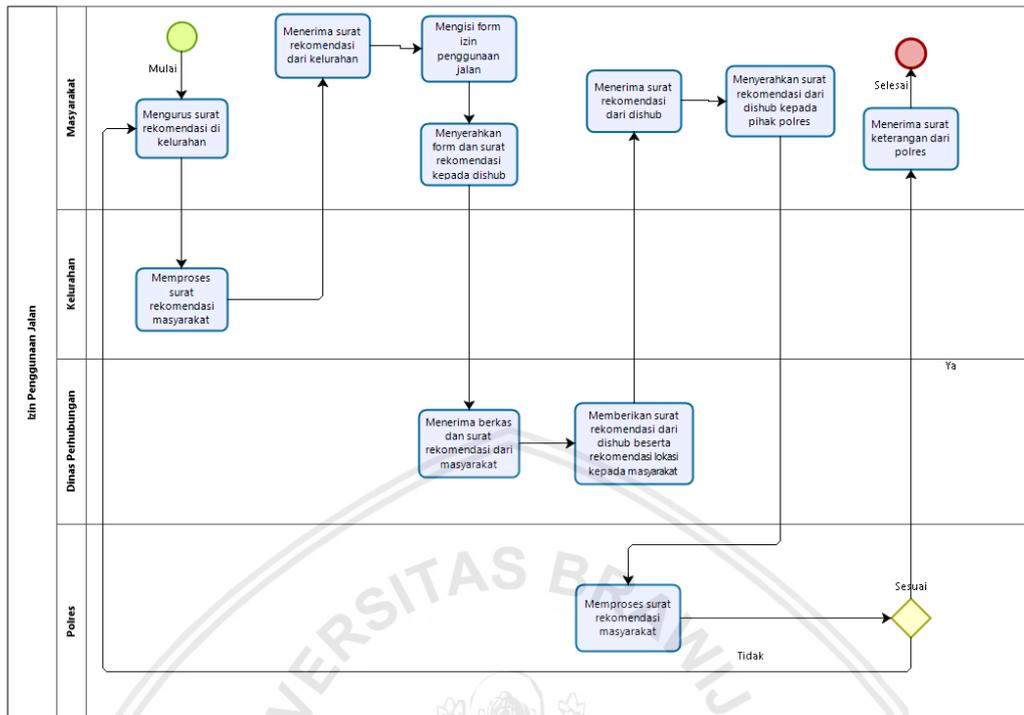
Proses elisitasi kebutuhan adalah proses dalam menemukan atau mendapatkan kebutuhan sistem melalui komunikasi dengan *customer* dan pihak lain yang berhubungan dengan sistem yang akan dikembangkan. Proses elisitasi kebutuhan dilakukan dengan cara wawancara dengan salah satu Staf Dinas Perhubungan Kota Mojokerto yaitu Ibu Ulfa Rahmawati selaku Pengelola Kepegawaian. Proses wawancara berfokus pada masalah-masalah yang dialami oleh pihak Dinas Perhubungan Kota Mojokerto khususnya yang berhubungan dengan masyarakat yaitu pelaporan keluhan masyarakat, izin penggunaan jalan, dan survei kepuasan masyarakat.

Proses pelaporan keluhan yang ada di Dinas Perhubungan kota Mojokerto masih menggunakan cara manual yaitu menggunakan media telepon, SMS, email, atau langsung menemui pihak dari Dinas terkait baik melalui petugas lapangan atau langsung ke kantor. Setelah melakukan laporan keluhan masyarakat akan menunggu tindak lanjut dari pihak Dinas Perhubungan kota Mojokerto. Proses pelaporan keluhan ditunjukkan pada Gambar 4. 1.



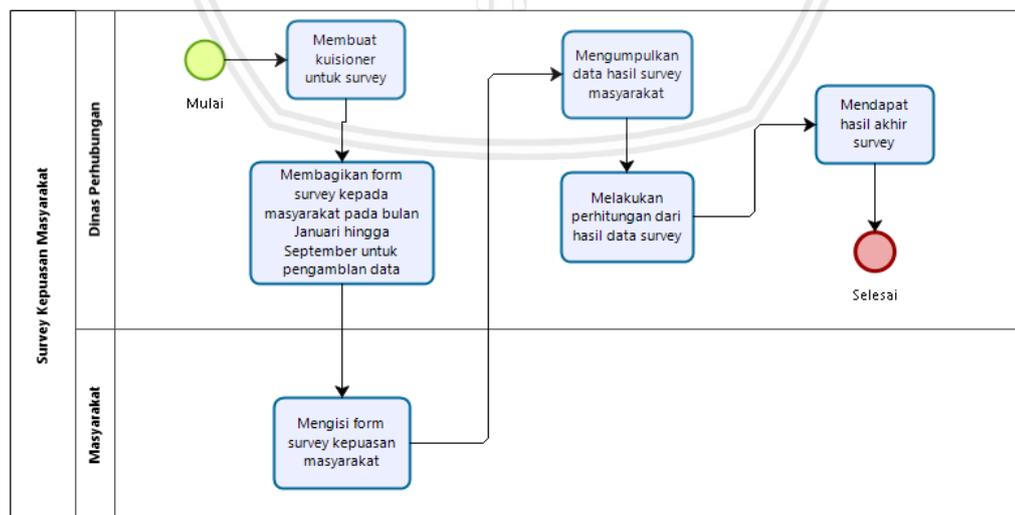
**Gambar 4. 1 Business Model As Is Pelaporan Keluhan**

Proses izin pengajuan jalan pada Dinas Perhubungan kota Mojokerto mengharuskan masyarakat untuk mengurus surat rekomendasi dari kelurahan setempat dan menyertakan lampiran berupa kartu tanda penduduk dan kartu keluarga, lalu membawa semua berkas yang diperlukan ke Dinas Perhubungan kota Mojokerto dan melakukan pengisian form yang disediakan. Proses izin pengajuan jalan ditunjukkan pada Gambar 4. 2.



Gambar 4. 2 Business Model As Is Izin Penggunaan Jalan

Proses survei kepuasan masyarakat pada Dinas Perhubungan kota Mojokerto dilakukan pada setiap bulan Januari sampai dengan September, setelah hasil data survei didapat akan dilakukan proses perhitungan data sampai mendapat hasil akhir dari survei. Proses survei kepuasan masyarakat ditunjukkan pada Gambar 4. 3.

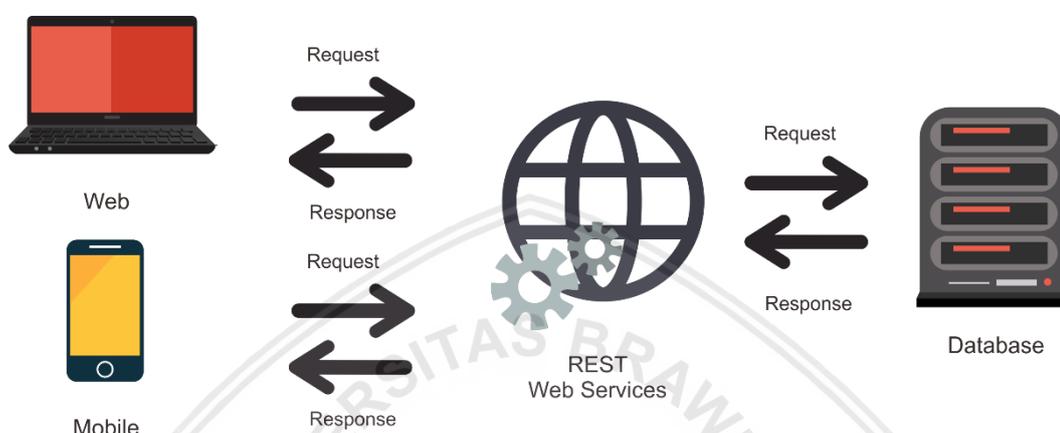


Gambar 4. 3 Business Model As Is Survei Kepuasan Masyarakat



#### 4.1.2 Gambaran Umum Sistem

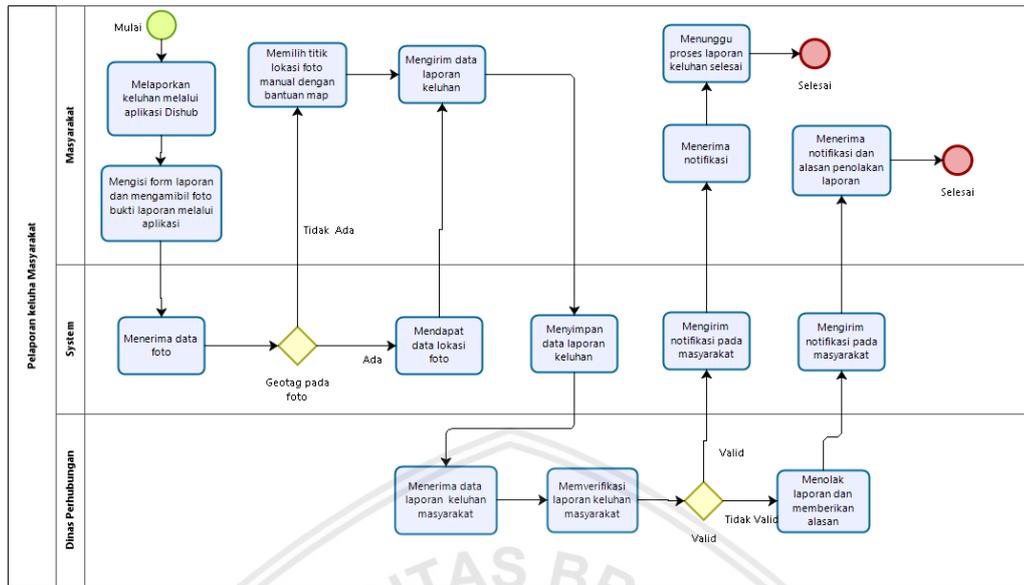
Aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* dibangun dengan teknologi *hybrid* menggunakan NodeJS sebagai *back-end* yang akan berperan sebagai RESTfull Web Services untuk menghubungkan perangkat dengan basis data melalui sebuah web services seperti yang ditunjukkan pada Gambar 4. 4.



**Gambar 4. 4 Arsitektur RESTful Web Services**

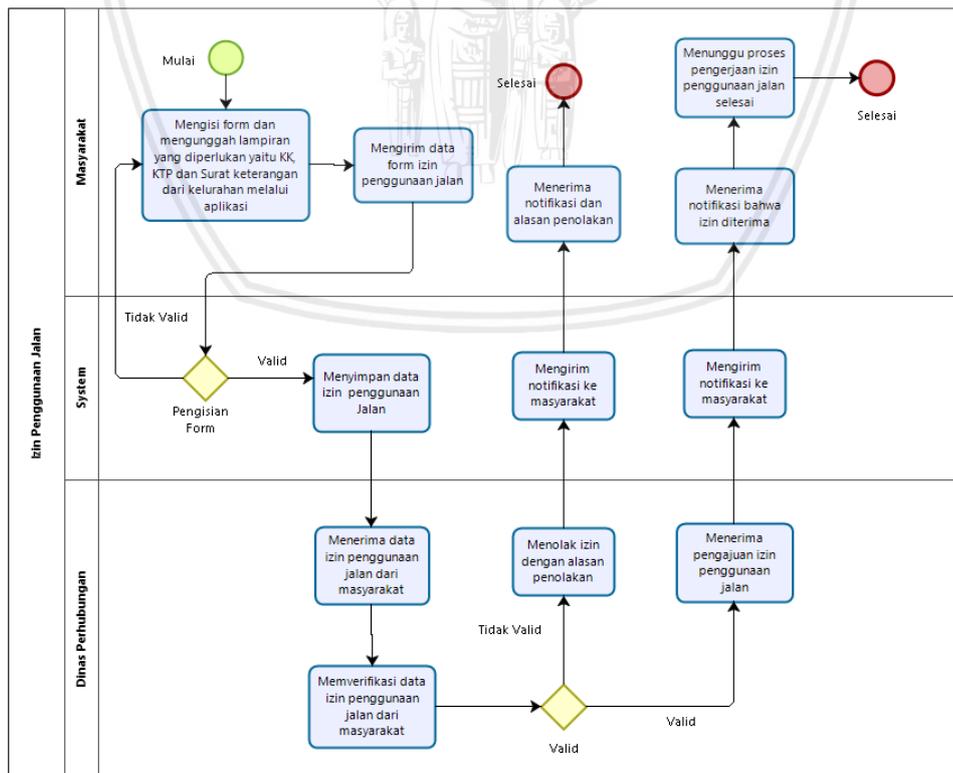
Pada bagian *front-end* menggunakan *framework* VueJS yang dipadukan dengan Framework7 yang menyediakan komponen-komponen untuk mendapatkan tampilan dan nuansa aplikasi *native* dan dibantu dengan Apache Cordova yang berfungsi untuk mengemas aplikasi sebagai paket menggunakan *Software Development Kit* (SDK) dari *platform* masing-masing seperti Android dan iOS dengan menggunakan bahasa Javascript yang menjadikannya sebuah aplikasi berbasis mobile dengan menggunakan teknologi web standar seperti HTML5, CSS3, dan Javascript.

Aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* merupakan sebuah aplikasi yang bertujuan untuk membuat masyarakat sebagai pemeran aktif untuk melaporkan semua keluhan-keluhan yang terjadi di bidang transportasi, angkutan jalan dan lalu lintas kepada Dinas Perhubungan Kota Mojokerto menggunakan *smartphone* yang disertai foto terkait keluhan yang dilengkapi dengan lokasi pengambilan foto. Proses penentuan lokasi dibantu dengan pemanfaatan teknologi GPS (*Global Positioning System*) untuk menentukan lokasi dengan tepat berdasarkan informasi yang didapat dari satelit berupa titik koordinat *latitude* dan *longitude*. Informasi lokasi yang didapatkan dari GPS akan disimpan dalam *metadata* foto dengan proses *geotagging*. Alur dari proses pelaporan keluhan masyarakat ini ditunjukkan pada Gambar 4. 5.

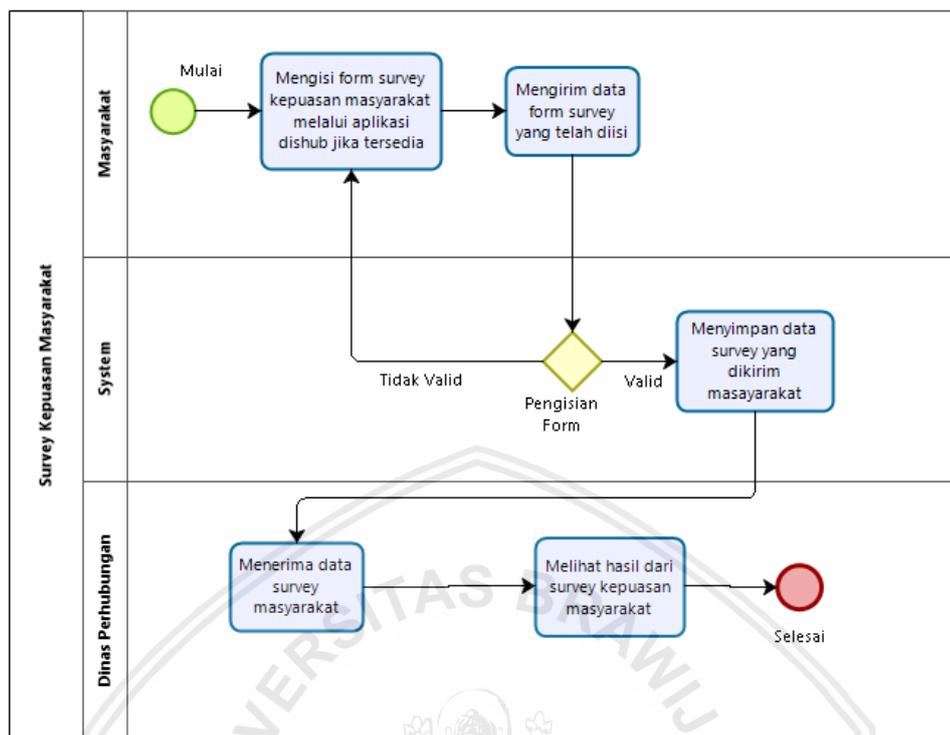


Gambar 4. 5 Business Model To Be Pelaporan Keluhan

Selain melakukan pelaporan masyarakat dapat mengurus surat rekomendasi izin penggunaan jalan, mengisi survei kepuasan masyarakat. Proses izin penggunaan jalan dan survei kepuasan masyarakat ditunjukkan pada Gambar 4. 6 dan Gambar 4. 7.



Gambar 4. 6 Business Model To Be Izin Penggunaan Jalan



Gambar 4. 7 Business Model To Be Survei Kepuasan Masyarakat

Sedangkan pada sisi Dinas Perhubungan Kota Mojokerto juga disediakan sistem berbasis web yang berisikan manajemen pengelolaan data yang ditampilkan pada masyarakat dan menampilkan laporan-laporan yang dikirim oleh masyarakat yang selanjutnya akan dilakukan proses verifikasi laporan dari masyarakat.

### 4.1.3 Identifikasi Aktor

Identifikasi aktor dilakukan dengan tujuan untuk mengidentifikasi atau sebagai representasi orang atau peran yang melakukan interaksi dengan sistem. Pada sistem ini terdapat tiga aktor seperti yang ditunjukkan pada Tabel 4. 1.

Tabel 4. 1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Guest	Guest adalah aktor yang belum mempunyai akun untuk menggunakan layanan pada aplikasi. Aktor ini hanya bisa melakukan Register dan Login untuk bisa menggunakan layanan yang tersedia.
2	User	User adalah guest yang telah melakukan registrasi dan login pada aplikasi. Aktor ini sudah bisa menggunakan semua layanan pada aplikasi.
3	Admin	Admin adalah aktor yang memiliki hak akses penuh pada aplikasi dan dapat mengelola data pada sistem.



#### 4.1.4 Aturan Penomoran

Seluruh kalimat kebutuhan harus memiliki nomor identifikasi. Berikut aturan penomoran aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* yang digunakan dalam laporan ini yang ditunjukkan pada Tabel 4. 2.

Tabel 4. 2 Aturan Penomoran

No.	Istilah	Deskripsi
1	DISHUB	Nama sistem
2	D	Kode representasi deskripsi kebutuhan fungsional
3	S	Kode representasi spesifikasi kebutuhan fungsional
2	01	Kode representasi kebutuhan fungsional
3	02	Kode representasi kebutuhan non-fungsional
4	001	Angka penomoran kebutuhan

#### 4.1.5 Kebutuhan Fungsional

Kebutuhan Fungsional bertujuan untuk menganalisis untuk mengetahui fungsi dan fitur apa saja yang harus dipenuhi oleh sistem. Dibawah ini terdapat tabel daftar kebutuhan fungsional sistem yang terdiri dari kolom berisi kode fungsi, nama fungsi dan deskripsi yang menguraikan kebutuhan dari fungsi. Kebutuhan fungsional bertujuan untuk memberikan gambaran mengenai sistem yang ditunjukkan pada Tabel 4. 3

Tabel 4. 3 Kebutuhan Fungsional

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
1	DISHUB-D-01-001	<i>Register</i>	<i>Guest</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk melakukan registrasi ke dalam sistem.
2	DISHUB-S-01-001	<i>Register</i>	<i>Guest</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk melakukan registrasi ke dalam sistem dengan memasukkan data diri yaitu nama, alamat, tempat dan tanggal lahir nomor telepon, alamat email, dan <i>password</i> .

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
3	DISHUB-D-01-002	<i>Login</i>	<i>Guest, Admin</i>	Sistem dapat memberikan fungsi bagi pengguna baik <i>guest</i> atau <i>admin</i> untuk autentikasi atau masuk ke dalam sistem.
4	DISHUB-S-01-002	<i>Login</i>	<i>Guest, Admin</i>	Sistem dapat memberikan fungsi bagi pengguna baik <i>guest</i> atau <i>admin</i> untuk masuk ke dalam sistem dengan autentikasi <i>email</i> dan <i>password</i> .
5	DISHUB-D-01-003	Lihat Laporan Keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melihat laporan keluhan yang ada.
6	DISHUB-S-01-003	Lihat Laporan Keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melihat laporan keluhan yang ada berupa judul laporan, gambar laporan, lokasi laporan dan deskripsi laporan.
7	DISHUB-D-01-004	Tambah Laporan Keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat menambahkan laporan keluhan.
8	DISHUB-S-01-004	Tambah Laporan Keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat menambahkan laporan keluhan dengan menambahkan foto beserta titik koordinat <i>map (latitude, longitude)</i> dengan proses <i>geotagging</i> dan deskripsi dari foto yang disertakan pada form laporan keluhan.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
9	DISHUB-D-01-005	Edit Laporan keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat mengubah isi dari laporan keluhannya masing-masing.
10	DISHUB-S-01-005	Edit Laporan keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat mengubah isi dari laporan keluhannya masing-masing meliputi foto, lokasi dan deskripsi dari laporan.
11	DISHUB-D-01-006	Vote Laporan Keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melakukan <i>vote</i> atau dukungan pada laporan keluhan yang ada.
12	DISHUB-S-01-006	Vote Laporan Keluhan	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melakukan <i>vote</i> atau dukungan pada laporan keluhan yang ada berdasarkan laporan keluhan yang dipilih.
13	DISHUB-D-D-01-007	Tambah Komentar Laporan Keluhan	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi <i>user</i> dan <i>admin</i> untuk dapat menambahkan komentar pada laporan keluhan yang ada.
14	DISHUB-S-D-01-007	Tambah Komentar Laporan Keluhan	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi <i>user</i> dan <i>admin</i> untuk dapat menambahkan komentar pada laporan keluhan yang ada dengan uraian dengan maksimal 500 karakter.
15	DISHUB-D-01-008	Edit Komentar Laporan Keluhan	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi <i>user</i> dan <i>admin</i> untuk dapat mengubah komentarnya pada laporan keluhan yang ada.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
16	DISHUB-S-01-008	Edit Komentar Laporan Keluhan	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi <i>user</i> dan <i>admin</i> untuk dapat mengubah komentarnya pada laporan keluhan yang ada berdasarkan komentar miliknya yang dipilih.
17	DISHUB-D-01-009	Hapus Komentar Laporan Keluhan	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi <i>user</i> dan <i>admin</i> untuk dapat menghapus komentar pada laporan keluhan yang ada.
18	DISHUB-S-01-009	Hapus Komentar Laporan Keluhan	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi <i>user</i> dan <i>admin</i> untuk dapat menghapus komentar pada laporan keluhan yang ada berdasarkan komentar miliknya yang dipilih.
19	DISHUB-D-01-010	Lihat Info Trayek	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melihat informasi terkait trayek.
20	DISHUB-S-01-010	Lihat Info Trayek	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melihat informasi terkait trayek yaitu angkutan umum serta rute dari masing-masing angkutan dan jam operasional dari pihak Dinas Perhubungan Kota Mojokerto.
21	DISHUB-D-001-011	Lihat Info Pengumuman	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melihat informasi terkait pengumuman.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
22	DISHUB-S-001-011	Lihat Info Pengumuman	User	Sistem dapat memberikan fungsi bagi <i>user</i> untuk dapat melihat informasi terkait pengumuman yaitu pengumuman penutupan jalan, atau berita seputar aktifitas pihak Dinas Perhubungan Kota Mojokerto.
23	DISHUB-D-01-012	Izin Penggunaan Jalan	User	Sistem dapat memberikan fungsi bagi <i>user</i> untuk melakukan izin penggunaan jalan.
24	DISHUB-S-01-012	Izin Penggunaan Jalan	User	Sistem dapat memberikan fungsi bagi <i>user</i> untuk melakukan izin penggunaan jalan dengan mengunggah file berupa gambar scan surat rekomendasi dari lurah setempat dan scan KTP warga untuk mendapat surat rekomendasi dari Dinas Perhubungan Kota Mojokerto.
25	DISHUB-D-01-013	Survey Kepuasan Masyarakat	User	Sistem dapat memberikan fungsi bagi <i>user</i> untuk mengisi survey terkait pendapat <i>user</i> .
26	DISHUB-S-01-013	Survey Kepuasan Masyarakat	User	Sistem dapat memberikan fungsi bagi <i>user</i> untuk mengisi survey terkait pendapat <i>user</i> seputar pelayanan dengan memilih pilihan jawaban yang tersedia pada setiap pertanyaan yang telah disediakan oleh Dinas Perhubungan Kota Mojokerto tiap tahunnya.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
27	DISHUB-D-01-014	Tambah Kritik dan Saran	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk memberikan kritik dan saran pada Dinas Perhubungan Kota Mojokerto.
28	DISHUB-S-01-014	Tambah Kritik dan Saran	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk memberikan kritik dan saran pada Dinas Perhubungan Kota Mojokerto dengan memasukkan uraian kritik dan saran pada form yang tersedia.
29	DISHUB-D-01-015	Edit Profile	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk melakukan edit detail informasi profil pada akun masing-masing.
30	DISHUB-S-01-015	Edit Profile	<i>User</i>	Sistem dapat memberikan fungsi bagi <i>user</i> untuk melakukan edit detail informasi profil meliputi nama, alamat, tempat dan tanggal lahir nomor telepon, alamat email, dan <i>password</i> pada akun masing-masing.
31	DISHUB-D-01-016	Verifikasi Laporan Keluhan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan verifikasi laporan keluhan.
32	DISHUB-S-01-016	Verifikasi Laporan Keluhan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan verifikasi laporan keluhan dengan memilih salah satu laporan keluhan yang ada.
33	DISHUB-D-01-017	Update Status Laporan Keluhan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan <i>update</i> status pada laporan keluhan yang ada.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
34	DISHUB-S-01-017	Update Status Laporan Keluhan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan <i>update</i> status yaitu menunggu, proses dan selesai pada laporan keluhan yang ada.
35	DISHUB-D-01-018	Cari Laporan Keluhan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan pencarian pada laporan keluhan.
36	DISHUB-S-01-018	Cari Laporan Keluhan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan pencarian pada laporan keluhan berdasarkan area atau wilayah tertentu.
37	DISHUB-D-01-019	Tambah Info Trayek	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menambahkan informasi terkait trayek.
38	DISHUB-S-01-019	Tambah Info Trayek	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menambahkan informasi terkait trayek yaitu angkutan umum, rute, dan jam operasional dari pihak Dinas Perhubungan Kota Mojokerto.
39	DISHUB-D-01-020	Edit Info Trayek	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk mengubah informasi terkait trayek.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
40	DISHUB-S-01-020	Edit Info Trayek	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk mengubah informasi terkait angkutan umum dan trayek dari pihak Dinas Perhubungan Kota Mojokerto berdasarkan info yang dipilih dan memasukkan data baru pada info yang dipilih.
41	DISHUB-D-01-021	Hapus Info Trayek	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menghapus informasi terkait trayek.
42	DISHUB-S-01-021	Hapus Info Trayek	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menghapus informasi terkait angkutan umum dan trayek dari pihak Dinas Perhubungan Kota Mojokerto berdasarkan info yang dipilih.
43	DISHUB-D-01-022	Tambah Info Pengumuman	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menambahkan informasi terkait pengumuman.
44	DISHUB-S-01-022	Tambah Info Pengumuman	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menambahkan informasi terkait pengumuman yaitu pengumuman penutupan jalan, atau berita seputar aktifitas yang ada dari pihak Dinas Perhubungan Kota Mojokerto.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
45	DISHUB-D-01-023	Edit Info Pengumuman	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk mengubah informasi terkait pengumuman yang ada dari pihak Dinas Perhubungan Kota Mojokerto.
46	DISHUB-S-01-023	Edit Info Pengumuman	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk mengubah informasi terkait pengumuman yang ada dari pihak Dinas Perhubungan Kota Mojokerto berdasarkan info yang dipilih dan memasukkan data baru pada info yang dipilih.
47	DISHUB-D-01-024	Hapus Info Pengumuman	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menghapus informasi terkait pengumuman yang ada dari pihak Dinas Perhubungan Kota Mojokerto.
48	DISHUB-S-01-024	Hapus Info Pengumuman	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menghapus informasi terkait pengumuman yang ada dari pihak Dinas Perhubungan Kota Mojokerto berdasarkan info yang dipilih.
49	DISHUB-D-01-025	Verifikasi Izin Penggunaan Jalan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan verifikasi pada berkas izin penggunaan jalan.
50	DISHUB-S-01-025	Verifikasi Izin Penggunaan Jalan	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melakukan verifikasi pada berkas <i>user</i> yang mengajukan penggunaan jalan dengan menerima atau menolaknya.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
51	DISHUB-D-01-026	Lihat Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melihat hasil survey kepuasan masyarakat.
52	DISHUB-S-01-026	Lihat Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melihat hasil survey kepuasan masyarakat berdasarkan survey yang dipilih.
53	DISHUB-D-01-027	Tambah Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menambahkan survey kepuasan masyarakat.
54	DISHUB-S-01-027	Tambah Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menambahkan survey kepuasan masyarakat berupa daftar pertanyaan dan pilihan jawaban yang tersedia.
55	DISHUB-D-01-028	Edit Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk mengubah data survey kepuasan masyarakat yang telah dimasukkan sebelumnya.
56	DISHUB-S-01-028	Edit Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk mengubah data survey kepuasan masyarakat yang telah dimasukkan sebelumnya berdasarkan survey yang dipilih.
57	DISHUB-D-01-029	Hapus Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menghapus survey kepuasan masyarakat.

No.	Kode Fungsi	Nama Fungsi	Aktor	Deskripsi/Spesifikasi Kebutuhan
58	DISHUB-S-01-029	Hapus Survey Kepuasan Masyarakat	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk menghapus survey kepuasan masyarakat berdasarkan survey yang dipilih.
59	DISHUB-D-01-030	Lihat Kritik dan Saran	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melihat kritik dan saran yang telah dikirimkan dari user.
60	DISHUB-S-01-030	Lihat Kritik dan Saran	<i>Admin</i>	Sistem dapat memberikan fungsi bagi <i>admin</i> untuk melihat kritik dan saran yang telah dikirimkan dari user berdasarkan kritik dan saran yang dipilih.
61	DISHUB-D-01-031	<i>Logout</i>	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi pengguna baik <i>user</i> atau <i>admin</i> untuk bisa <i>logout</i> .
62	DISHUB-S-01-031	<i>Logout</i>	<i>User, Admin</i>	Sistem dapat memberikan fungsi bagi pengguna baik <i>user</i> atau <i>admin</i> untuk bisa keluar dari aplikasi.

#### 4.1.6 Kebutuhan Non-fungsional

Dibawah ini terdapat tabel kebutuhan non-fungsional sistem yang terdiri dari kolom berisi parameter dan deskripsi yang menguraikan parameter. Kebutuhan non-fungsional bertujuan untuk mengetahui batasan-batasan fungsi yang ditawarkan sistem seperti yang ditunjukkan pada Tabel 4. 4.

Tabel 4. 4 Kebutuhan Non-fungsional

No.	Kode Fungsi	Parameter	Deskripsi
1	DISHUB-02-001	<i>Compatibility</i>	Sistem harus dapat berjalan pada <i>platform</i> android dengan versi minimal 5.0 hingga versi paling baru saat ini 9.0.

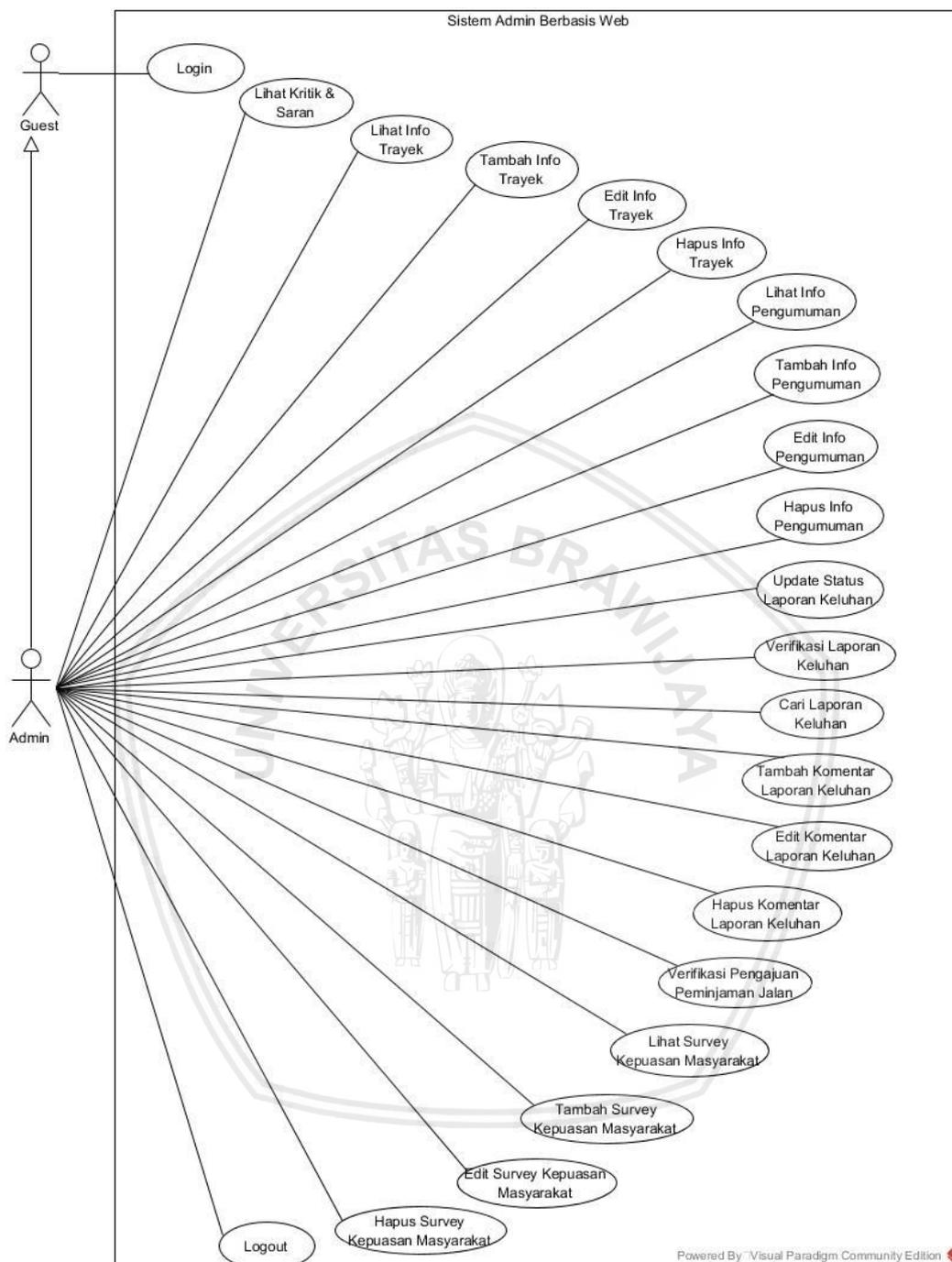
### 4.1.7 Use Case Diagram

Use Case Diagram memodelkan kebutuhan fungsional dengan aktor-aktor atau pengguna aplikasi yang terlibat pada sistem dan use case yang merepresentasikan kebutuhan fungsional. Pada aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* terdapat dua aktor yang terlibat yaitu *Guest* dan *User* yang ditunjukkan pada Gambar 4. 8.



Gambar 4. 8 Use Case Diagram Aplikasi Pelaporan Keluhan Bagian Satu

Sedangkan untuk *admin* pada sisi Dinas Perhubungan Kota Mojokerto yang digunakan untuk mengelola sumber informasi dan verifikasi laporan mempunyai satu aktor yaitu *Admin* yang ditunjukkan pada Gambar 4. 9.



Gambar 4. 9 Use Case Diagram Aplikasi Pelaporan Keluhan Bagian Dua

#### 4.1.8 Skenario Use Case

Setiap *use case* akan didetailkan terkait perilaku dalam skenario *use case*. Skenario *use case* akan berisi nama *use case*, aktor yang terlibat dalam *use case*, tujuan dari *use case*, kondisi awal yang harus dipenuhi, alur utama *use case*, alur *alternative*, dan kondisi akhir yang diharapkan.



### 1. Skenario Use Case Register

Pada skenario *use case register*, *guest* belum memiliki akun untuk melakukan *login* pada aplikasi. *guest* harus melakukan registrasi dengan memasukkan data diri di *field* pada form yang telah tersedia. Skenario *use case register* ditunjukkan pada Tabel 4. 5.

Tabel 4. 5 Skenario *Use Case Register*

<b>Register</b>	
<i>Objective</i>	<i>Guest</i> berhasil melakukan registrasi dan telah mempunyai akun untuk mengakses fitur dan layanan pada aplikasi.
<i>Actor</i>	<i>Guest</i>
<i>Pre-condition</i>	<i>Guest</i> tidak bisa menggunakan fitur dan layanan yang disediakan sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman <i>login</i>.</li> <li>2. <i>Guest</i> memilih menu untuk melakukan registrasi.</li> <li>3. Sistem menampilkan halaman <i>register</i>.</li> <li>4. <i>Guest</i> melakukan <i>input</i> data diri nama, alamat, tanggal lahir, nomor telepon, alamat <i>email</i>, dan <i>password</i> pada form yang telah tersedia.</li> <li>5. <i>Guest</i> menekan tombol <i>submit</i>.</li> <li>6. Sistem menyimpan data registrasi pengguna.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Guest</i> gagal melakukan registrasi karena email yang digunakan tidak valid atau sudah terdaftar.</li> <li>2. <i>Guest</i> gagal melakukan registrasi karena input data tidak lengkap.</li> </ol>
<i>Post-condition</i>	<i>Guest</i> telah memiliki akun yang sudah bisa digunakan untuk mengakses layanan dan fitur aplikasi.

## 2. Skenario Use Case Login

Pada skenario *use case login*, aktor baik *guest* dan *admin* belum bisa menggunakan fitur dari sistem. Aktor harus melakukan *login* terlebih dahulu sebelum menggunakan fitur sistem. Aktor bisa melakukan *login* menggunakan *email* dan *password*. Setelah melakukan *login* aktor dapat menggunakan fitur dari sistem. Usecase *scenario login* ditunjukkan pada Tabel 4. 6.

Tabel 4. 6 Skenario *Use Case Login*

<b>Login</b>	
<i>Objective</i>	Aktor dapat melakukan autentikasi pada sistem dan menggunakan semua fitur dan layanan pada sistem sesuai dengan hak akses masing-masing.
<i>Actor</i>	<i>Guest, Admin</i>
<i>Pre-condition</i>	Aktor tidak bisa menggunakan fitur dan layanan yang disediakan sistem.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor membuka aplikasi dan akan diarahkan ke halaman <i>login</i>.</li> <li>2. Sistem menampilkan halaman <i>login</i>.</li> <li>3. Aktor memasukkan <i>email</i> dan <i>password</i> yang telah terdaftar pada aplikasi.</li> <li>4. Aktor menekan tombol <i>submit</i>.</li> <li>5. Sistem akan mengarahkan actor ke halaman beranda untuk <i>user</i> dan halaman <i>dashboard</i> untuk <i>admin</i>.</li> </ol>
<i>Alternative Flow</i>	Aktor tidak bisa melakukan autentikasi karena belum terdaftar atau mengalami kesalahan pada kombinasi email dan password.
<i>Post-condition</i>	Aktor telah berhasil melakukan autentikasi dan bisa menggunakan fitur dan layanan yang disediakan sistem sesuai dengan hak akses masing-masing.

### 3. Skenario Use Case Lihat Laporan Keluhan

Pada skenario *use case* lihat laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat melihat laporan keluhan dengan memilih salah satu laporan keluhan yang ingin dilihat. Skenario *use case* lihat laporan keluhan ditunjukkan pada Tabel 4. 7.

Tabel 4. 7 Skenario *Use Case* Lihat Laporan Keluhan

Lihat Laporan Keluhan	
<i>Objective</i>	<i>User</i> dapat melihat laporan keluhan melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Sistem melakukan verifikasi <i>token</i>.</li> <li>2. Sistem menampilkan halaman beranda yang berisi semua laporan keluhan yang ada.</li> <li>3. <i>User</i> memilih salah satu laporan keluhan yang ingin dilihat.</li> <li>4. Sistem menampilkan laporan keluhan yang dipilih <i>user</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal melihat data laporan keluhannya karena terjadi kesalahan server.</li> <li>2. <i>User</i> gagal melihat data karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil melihat laporan keluhan dengan benar sesuai dengan yang ditampilkan aplikasi.

### 4. Skenario Use Case Tambah Laporan Keluhan

Pada skenario *use case* tambah laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat menambahkan laporan keluhan dengan memasukkan data pada form seperti deskripsi, foto, dan titik koordinat. Skenario *use case* tambah laporan keluhan ditunjukkan pada Tabel 4. 8.

Tabel 4. 8 Skenario *Use Case* Tambah Laporan Keluhan

Tambah Laporan Keluhan	
<i>Objective</i>	<i>User</i> dapat menambahkan laporan keluhan melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu lapor untuk menambahkan laporan keluhan.</li> <li>2. Sistem menampilkan halaman lapor.</li> <li>3. <i>User</i> mengunggah foto dengan melalui menu kamera atau galeri.</li> <li>4. Sistem menerima foto yang dipilih <i>user</i> dan menampilkan foto.</li> <li>5. <i>User</i> memilih koordinat titik lokasi laporan keluhan dengan <i>map Leaflet</i>.</li> <li>6. Sistem menampilkan alamat lengkap dari koordinat titik lokasi yang dipilih <i>user</i> dengan bantuan <i>geocoder</i>.</li> <li>7. <i>User</i> melakukan input data kategori dan deskripsi laporan.</li> <li>8. <i>User</i> menekan tombol <i>submit</i>.</li> <li>9. Sistem melakukan verifikasi <i>token</i>.</li> <li>10. Sistem menyimpan data laporan keluhan <i>user</i>.</li> </ol>

<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal melakukan input data karena data yang diinputkan user tidak lengkap atau tidak valid.</li> <li>2. <i>User</i> gagal melakukan input data karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	User berhasil menambahkan laporan keluhan dengan benar dan data laporan keluhan berhasil ditampilkan.

### 5. Skenario Use Case Edit Laporan Keluhan

Pada skenario *use case* edit laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat mengubah data laporan keluhannya dengan memasukkan data yang baru pada form. Skenario *use case* edit laporan keluhan ditunjukkan pada Tabel 4. 9.

Tabel 4. 9 Skenario *Use Case* Edit Laporan Keluhan

Edit Laporan Keluhan	
<i>Objective</i>	<i>User</i> dapat mengubah laporan keluhannya melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu laporan keluhan.</li> <li>2. Sistem menampilkan daftar laporan keluhan milik <i>user</i>.</li> <li>3. <i>User</i> memilih salah satu laporan yang ingin diubah.</li> <li>4. Sistem menampilkan halaman detail laporan.</li> <li>5. <i>User</i> memilih menu <i>edit</i> laporan.</li> <li>6. Sistem menampilkan halaman <i>edit</i> laporan.</li> <li>7. <i>User</i> memasukkan deskripsi baru pada laporan keluhannya dan menyimpan perubahan.</li> </ol>

	<ol style="list-style-type: none"> <li>8. Sistem melakukan verifikasi <i>token</i>.</li> <li>9. Sistem menyimpan perubahan data laporan keluhan.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal mengubah data karena data yang diinputkan user salah atau tidak valid.</li> <li>2. <i>User</i> gagal mengubah data karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	User berhasil mengubah laporan keluhannya dengan benar dan perubahan berhasil ditampilkan.

## 6. Skenario Use Case Vote Laporan Keluhan

Pada skenario *use case vote* laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat melakukan *vote* atau mendukung laporan keluhan yang ada pada aplikasi dengan memilih laporan keluhan mana yang ingin didukung dan memilih opsi *vote*. Skenario *use case vote* laporan keluhan ditunjukkan pada Tabel 4. 10.

Tabel 4. 10 Skenario *Use Case Vote* Laporan Keluhan

<b>Vote Laporan Keluhan</b>	
<i>Objective</i>	<i>User</i> dapat melakukan <i>vote</i> atau mendukung laporan keluhan yang ada melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih opsi <i>vote</i> pada laporan yang ingin di <i>vote</i>.</li> <li>2. Sistem melakukan verifikasi <i>token</i>.</li> <li>3. Sistem menyimpan <i>vote</i> yang dilakukan <i>user</i>.</li> </ol>



	4. Sistem menampilkan perubahan pada opsi <i>vote</i> bahwa laporan keluhan sudah di <i>vote</i> .
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal melakukan <i>vote</i> laporan keluhan karena terjadi kesalahan server.</li> <li>2. <i>User</i> gagal melakukan <i>vote</i> karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil melakukan <i>vote</i> laporan keluhan dengan benar dan jumlah <i>vote</i> bertambah.

### 7. Skenario Use Case Tambah Komentar Laporan Keluhan

Pada skenario *use case* tambah komentar laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat menambahkan komentar pada laporan keluhan dengan memilih laporan keluhan mana yang ingin dikomentari dan memilih opsi komentar. Skenario *use case* tambah komentar laporan keluhan ditunjukkan pada Tabel 4. 11.

Tabel 4. 11 Skenario *Use Case* Tambah Komentar Laporan Keluhan

Tambah Komentar Laporan Keluhan	
<i>Objective</i>	<i>User</i> dapat menambahkan komentar pada laporan keluhan melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih laporan yang ingin dikomentari.</li> <li>2. Sistem menampilkan halaman detail laporan.</li> <li>3. <i>User</i> memilih opsi komentar.</li> <li>4. Sistem menampilkan halaman komentar pada laporan keluhan yang dipilih.</li> <li>5. <i>User</i> memasukkan komentar pada form yang tersedia.</li> </ol>

	<ol style="list-style-type: none"> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan komentar yang dimasukkan <i>user</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal menambahkan komentar pada laporan keluhan karena <i>input</i> komentar tidak diisi.</li> <li>2. <i>User</i> gagal menambahkan komentar karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil menambahkan komentar laporan keluhan dengan benar dan komentar berhasil ditampilkan.

### 8. Skenario Use Case Edit Komentar Laporan Keluhan

Pada skenario *use case* edit komentar laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat menambahkan komentar pada laporan keluhan dengan memilih laporan keluhan mana yang ingin diubah komentarnya dan memasukkan komentar barunya. Skenario *use case* edit komentar laporan keluhan ditunjukkan pada Tabel 4. 12.

Tabel 4. 12 Skenario *Use Case Edit* Komentar Laporan Keluhan

Edit Komentar Laporan Keluhan	
<i>Objective</i>	<i>User</i> dapat mengubah komentarnya pada laporan keluhan melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih laporan yang ingin di komentari.</li> <li>2. Sistem menampilkan halaman detail laporan.</li> <li>3. <i>User</i> memilih opsi komentar.</li> <li>4. Sistem menampilkan halaman komentar pada laporan keluhan yang dipilih.</li> </ol>

	<ol style="list-style-type: none"> <li>5. <i>User</i> memilih komentarnya yang telah di kirim sebelumnya.</li> <li>6. <i>User</i> memilih opsi <i>edit</i> pada komentarnya.</li> <li>7. <i>User</i> mengubah komentar pada form yang tersedia.</li> <li>8. Sistem melakukan verifikasi <i>token</i>.</li> <li>9. Sistem menyimpan perubahan komentar.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal mengubah komentarnya pada laporan keluhan karena <i>input</i> komentar tidak diisi.</li> <li>2. <i>User</i> gagal mengubah komentar karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil mengubah komentarnya pada laporan keluhan dengan benar dan perubahan berhasil ditampilkan.

### 9. Skenario Use Case Hapus Komentar Laporan Keluhan

Pada skenario *use case* hapus komentar laporan keluhan, *user* telah melakukan *login* pada aplikasi. *User* dapat menghapus komentar pada laporan keluhan dengan memilih laporan keluhan mana yang ingin dihapus komentarnya yang sebelumnya telah dikirim. Skenario *use case* hapus komentar laporan keluhan ditunjukkan pada Tabel 4. 13.

Tabel 4. 13 Skenario *Use Case* Hapus Komentar Laporan Keluhan

Hapus Komentar Laporan Keluhan	
<i>Objective</i>	<i>User</i> dapat menghapus komentarnya pada laporan keluhan melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih laporan yang ingin di komentari.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman detail laporan.</li> <li>3. <i>User</i> memilih opsi komentar.</li> <li>4. Sistem menampilkan halaman komentar pada laporan keluhan yang dipilih.</li> <li>5. <i>User</i> memilih komentarnya yang telah di kirim sebelumnya.</li> <li>6. <i>User</i> memilih opsi hapus pada komentarnya.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menghapus komentar yang dipilih.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal menghapus komentarnya pada laporan keluhan karena terjadi kesalahan server.</li> <li>2. <i>User</i> gagal menghapus komentar karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil menghapus komentarnya pada laporan keluhan dengan benar dan komentar tidak tampil lagi.

#### 10. Skenario Use Case Lihat Info Trayek

Pada skenario *use case* lihat info trayek, *user* telah melakukan *login* pada aplikasi. *User* dapat melihat informasi yang telah tersedia pada aplikasi terkait trayek seperti angkutan umum, rute, dan jam operasional dari pihak Dinas Perhubungan Kota Mojokerto. Skenario *use case* lihat info trayek ditunjukkan pada Tabel 4. 14.

Tabel 4. 14 Skenario *Use Case* Lihat Info Trayek

Lihat Info Trayek	
<i>Objective</i>	<i>User</i> dapat melihat informasi terkait trayek dan angkutan umum pada aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu info pada aplikasi.</li> <li>2. Sistem melakukan verifikasi <i>token</i>.</li> <li>3. Sistem menampilkan halaman info.</li> <li>4. <i>User</i> memilih kategori trayek.</li> <li>5. Sistem menampilkan info trayek seperti angkutan umum, rute dan jam operasional.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> tidak bisa melihat informasi terkait trayek dan angkutan umum karena kesalahan server.</li> <li>2. <i>User</i> gagal melihat informasi terkait trayek dan angkutan umum karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil melihat informasi terkait trayek dan angkutan umum dengan benar.

#### 11. Skenario *Use Case* Lihat Info Pengumuman

Pada skenario *use case* lihat info pengumuman, *user* telah melakukan *login* pada aplikasi. *User* dapat melihat informasi yang telah tersedia pada aplikasi terkait pengumuman seperti pengumuman penutupan jalan, atau berita seputar aktifitas yang ada dari pihak Dinas Perhubungan Kota Mojokerto. Skenario *use case* lihat info trayek ditunjukkan pada Tabel 4. 15.

Tabel 4. 15 Skenario *Use Case* Lihat Info Pengumuman

Lihat Info Pengumuman	
<i>Objective</i>	<i>User</i> dapat melihat informasi terkait pengumuman yang ada pada aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu info pada aplikasi.</li> <li>2. Sistem melakukan verifikasi <i>token</i>.</li> <li>3. Sistem menampilkan halaman info.</li> <li>4. <i>User</i> memilih kategori trayek.</li> <li>5. <i>User</i> memilih kategori pengumuman.</li> <li>6. Sistem menampilkan info pengumuman penutupan jalan, atau berita seputar aktifitas pihak Dinas Perhubungan Kota Mojokerto.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> tidak bisa melihat informasi terkait pengumuman yang ada karena kesalahan server.</li> <li>2. <i>User</i> gagal melihat informasi terkait pengumuman karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil melihat informasi terkait pengumuman yang ada dengan benar.

## 12. Skenario *Use Case* Izin Penggunaan Jalan

Pada skenario *use case* izin penggunaan jalan, *user* telah melakukan *login* pada aplikasi. *User* dapat melakukan izin penggunaan jalan melalui aplikasi dengan mengunggah berkas yang diminta pada form aplikasi. Skenario *use case* izin penggunaan jalan ditunjukkan pada Tabel 4. 16.

Tabel 4. 16 Skenario *Use Case* Izin Penggunaan Jalan

Izin Penggunaan Jalan	
<i>Objective</i>	<i>User</i> dapat mengunggah berkas izin penggunaan jalan melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu izin penggunaan jalan pada aplikasi.</li> <li>2. Sistem menampilkan halaman izin penggunaan jalan.</li> <li>3. <i>User</i> memilih menu tambah izin.</li> <li>4. Sistem menampilkan halaman tambah izin.</li> <li>5. <i>User</i> mengunggah foto berupa scan gambar surat rekomendasi dari lurah setempat dan scan KTP dan KK.</li> <li>6. Sistem menampilkan foto yang diunggah <i>user</i>.</li> <li>7. <i>User</i> melakukan <i>input</i> data sesuai dengan <i>field</i> yang diminta seperti jenis kegiatan, lokasi kegiatan, tanggal jam kegiatan. pada form dari Dinas Perhubungan Kota Mojokerto.</li> <li>8. <i>User</i> menekan tombol <i>submit</i>.</li> <li>9. Sistem melakukan verifikasi <i>token</i>.</li> <li>10. Sistem menyimpan data izin penggunaan jalan yang dikirim <i>user</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal melakukan izin penggunaan jalan karena data <i>input</i> tidak lengkap.</li> </ol>

	2. <i>User</i> gagal melakukan input data karena <i>token</i> milik <i>user</i> tidak valid.
<i>Post-condition</i>	<i>User</i> berhasil melakukan izin penggunaan jalan dan mendapat surat rekomendasi dari pihak Dinas Perhubungan Kota Mojokerto.

### 13. Skenario Use Case Survei Kepuasan Masyarakat

Pada skenario *use case* survei kepuasan masyarakat, *user* telah melakukan *login* pada aplikasi. *User* akan disediakan sebuah survei mengenai kepuasan masyarakat terhadap fasilitas dan layanan Dinas Perhubungan Kota Mojokerto pada setiap akhir tahun. Skenario *use case* survei kepuasan masyarakat ditunjukkan pada Tabel 4. 17.

Tabel 4. 17 Skenario *Use Case* Survei Kepuasan Masyarakat

Survei Kepuasan Masyarakat	
<i>Objective</i>	<i>User</i> dapat melakukan pengisian survei kepuasan masyarakat dengan benar.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu survei kepuasan masyarakat pada aplikasi.</li> <li>2. Sistem menampilkan halaman survei yang berisi daftar pertanyaan dan jawaban.</li> <li>3. <i>User</i> mengisi survei sesuai dengan pilihan yang tersedia dengan menjawab semua pertanyaan dengan jawaban yang telah disediakan dari aplikasi.</li> <li>4. Sistem melakukan verifikasi <i>token</i>.</li> <li>5. Sistem menyimpan data survei yang telah diisi <i>user</i>.</li> </ol>

<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal melakukan survei kepuasan masyarakat karena data <i>input</i> tidak lengkap.</li> <li>2. <i>User</i> gagal melakukan survei kepuasan masyarakat karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	User telah berhasil mengisi survei kepuasan masyarakat dengan benar.

#### 14. Skenario Use Case Tambah Kritik dan Saran

Pada skenario *use case* tambah kritik dan saran, *user* telah melakukan *login* pada aplikasi. *User* dapat memberikan kritik dan saran pada Dinas Perhubungan Kota Mojokerto melalui aplikasi. Skenario *use case* tambah kritik dan saran ditunjukkan pada Tabel 4. 18.

Tabel 4. 18 Skenario *Use Case* Tambah Kritik dan Saran

Tambah Kritik dan Saran	
<i>Objective</i>	<i>User</i> dapat menambahkan kritik dan saran melalui aplikasi.
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> membuka menu kritik dan saran.</li> <li>2. Sistem menampilkan halaman kritik dan saran.</li> <li>3. <i>User</i> memilih tingkat <i>rate</i> yang disediakan yaitu tidak puas, netral, dan sangat puas.</li> <li>4. <i>User</i> mengisi form deskripsi terkait kritik dan saran.</li> <li>5. <i>User</i> menekan tombol <i>submit</i>.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan data kritik dan saran dari <i>user</i>.</li> </ol>

<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> gagal menambahkan kritik dan saran karena data <i>input</i> tidak lengkap.</li> <li>2. <i>User</i> gagal menambahkan kritik dan saran karena <i>token</i> milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil menambahkan kritik dan saran dengan benar.

### 15. Skenario Use Case Edit Profile

Pada skenario *use case edit profile*, *user* telah melakukan *login* pada aplikasi. *User* dapat melakukan perubahan data diri pada profil yang telah ada saat proses registrasi dilakukan. Skenario *use case edit profile* ditunjukkan pada Tabel 4. 19.

Tabel 4. 19 Skenario *Use Case Edit Profile*

<b>Edit Profile</b>	
<i>Objective</i>	<i>User</i> dapat mengubah data diri pada profil masing-masing <i>user</i> .
<i>Actor</i>	<i>User</i>
<i>Pre-condition</i>	<i>User</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu <i>profile</i> pada aplikasi.</li> <li>2. Sistem menampilkan halaman profil.</li> <li>3. <i>User</i> memilih opsi <i>edit profile</i>.</li> <li>4. Sistem menampilkan halaman <i>edit profile</i></li> <li>5. <i>User</i> melakukan perubahan data pada form yang telah tersedia seperti nama, alamat, telepon email , tanggal lahir dan foto.</li> <li>6. <i>User</i> menekan tombol <i>submit</i>.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan perubahan data pada profil milik <i>user</i>.</li> </ol>



<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. User gagal melakukan perubahan profil karena pengisian form tidak lengkap.</li> <li>2. <i>User</i> gagal mengubah profil karena token milik <i>user</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>User</i> berhasil melakukan perubahan data diri pada profil dan perubahan berhasil ditampilkan.

### 16. Skenario Use Case Verifikasi Laporan Keluhan

Pada skenario *use case* verifikasi laporan keluhan, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat melakukan verifikasi semua laporan keluhan yang ada baik menolak laporan keluhan atau menerima laporan keluhan. Skenario *use case* verifikasi laporan keluhan ditunjukkan pada Tabel 4. 20.

Tabel 4. 20 Skenario *Use Case* Verifikasi Laporan Keluhan

Verifikasi Laporan Keluhan	
<i>Objective</i>	<i>Admin</i> dapat melakukan verifikasi laporan keluhan yang ada melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu laporan keluhan.</li> <li>2. Sistem menampilkan halaman laporan keluhan.</li> <li>3. <i>Admin</i> melakukan verifikasi pada salah satu laporan keluhan dengan memilih salah satu opsi yaitu terima atau tolak.</li> <li>4. Sistem menampilkan notifikasi keamanan dengan pilihan opsi ya dan tidak.</li> <li>5. <i>Admin</i> memilih opsi ya.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> </ol>



	7. Sistem menyimpan perubahan status pada laporan keluhan.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal melakukan verifikasi laporan keluhan karena kesalahan server.</li> <li>2. <i>Admin</i> gagal melakukan verifikasi laporan keluhan karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> berhasil melakukan verifikasi laporan keluhan yang ada dengan benar.

### 17. Skenario Use Case Update Status Laporan Keluhan

Pada skenario *use case update* status laporan keluhan, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat melakukan update status semua laporan keluhan yang ada telah diverifikasi sebelumnya dengan tiga status yaitu menunggu, proses dan selesai. Skenario *use case update* status laporan keluhan ditunjukkan pada Tabel 4. 21.

Tabel 4. 21 Skenario *Use Case Update* Status Laporan Keluhan

<b>Update Status Laporan Keluhan</b>	
<i>Objective</i>	Admin dapat melakukan <i>update</i> status pada laporan keluhan yang telah diverifikasi sebelumnya melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu laporan keluhan.</li> <li>2. Sistem menampilkan halaman laporan keluhan.</li> <li>3. <i>Admin</i> memilih opsi proses atau selesai pada laporan keluhan yang telah diverifikasi sebelumnya pada daftar laporan keluhan.</li> <li>4. Sistem menampilkan notifikasi keamanan dengan pilihan opsi ya dan tidak.</li> </ol>

	<ol style="list-style-type: none"> <li>5. <i>Admin</i> memilih opsi ya.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan perubahan status pada laporan keluhan.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal melakukan <i>update</i> status laporan keluhan karena kesalahan server.</li> <li>2. <i>Admin</i> gagal melakukan <i>update</i> status laporan keluhan karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> berhasil melakukan <i>update</i> status pada laporan keluhan yang telah diverifikasi sebelumnya dengan benar.

### 18. Skenario Use Case Cari Laporan Keluhan

Pada skenario *use case* Cari laporan keluhan, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat melakukan mencari semua laporan keluhan yang ada dengan memasukkan *keywords* pada form pencarian. Skenario *use case* cari laporan keluhan ditunjukkan pada Tabel 4. 22.

Tabel 4. 22 Skenario *Use Case* Cari Laporan Keluhan

Cari Laporan Keluhan	
<i>Objective</i>	<i>Admin</i> dapat melakukan pencarian laporan keluhan yang ada melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu laporan keluhan.</li> <li>2. Sistem menampilkan halaman laporan keluhan.</li> <li>3. <i>Admin</i> melakukan pencarian dengan memasukkan <i>keywords</i> pada form pencarian.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Sistem melakukan verifikasi <i>token</i>.</li> <li>5. Sistem menerima input pencarian dari <i>admin</i> dan menampilkan hasil dari pencarian.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal melakukan pencarian laporan keluhan karena kesalahan server atau <i>keywords</i> yang dimasukkan tidak sesuai.</li> <li>2. <i>Admin</i> gagal melakukan pencarian laporan keluhan karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> berhasil mendapatkan hasil pencarian laporan keluhan yang ada dengan benar.

#### 19. Skenario Use Case Tambah Info Trayek

Pada skenario *use case* tambah info trayek, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat menambahkan informasi terkait trayek dan angkutan umum. Skenario *use case* tambah info trayek ditunjukkan pada Tabel 4. 23.

Tabel 4. 23 Skenario *Use Case* Tambah Info Trayek

Tambah Info Trayek	
<i>Objective</i>	<i>Admin</i> dapat menambahkan informasi terkait trayek dan angkutan umum melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu trayek pada aplikasi.</li> <li>2. Sistem menampilkan halaman trayek.</li> </ol>

	<ol style="list-style-type: none"> <li>3. <i>Admin</i> memilih opsi tambah trayek untuk menambah data trayek.</li> <li>4. Sistem menampilkan form input untuk data trayek.</li> <li>5. <i>Admin</i> memasukkan data trayek yaitu nama trayek, rute, dan koordinat <i>polyline</i> dengan <i>map Leaflet</i> sesuai dengan rute yang diinginkan.</li> <li>6. <i>Admin</i> menekan tombol <i>submit</i>.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan data trayek yang dikirim <i>admin</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal menambahkan informasi terkait trayek dan angkutan umum karena kesalahan server.</li> <li>2. <i>Admin</i> gagal menambahkan informasi terkait trayek dan angkutan umum karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil menambahkan informasi terkait trayek dan data yang baru dimasukkan tampil pada daftar info trayek pada aplikasi.

## 20. Skenario Use Case Edit Info Trayek

Pada skenario *use case edit* info trayek, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat mengubah data informasi terkait trayek dan angkutan umum yang telah dimasukkan sebelumnya. Skenario *use case edit* info trayek ditunjukkan pada Tabel 4. 24.

Tabel 4. 24 Skenario *Use Case Edit Info Trayek*

<b>Edit Info Trayek</b>	
<i>Objective</i>	<i>Admin</i> dapat mengubah informasi terkait trayek dan angkutan umum yang telah dimasukkan sebelumnya melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu trayek pada aplikasi.</li> <li>2. Sistem menampilkan halaman trayek.</li> <li>3. <i>Admin</i> memilih opsi edit trayek pada salah satu daftar trayek yang tersedia.</li> <li>4. Sistem menampilkan form input untuk edit data trayek.</li> <li>5. <i>Admin</i> mengubah data trayek yaitu nama trayek, rute, dan koordinat <i>polyline</i> dengan <i>map Leaflet</i> sesuai dengan rute yang diinginkan.</li> <li>6. <i>Admin</i> menekan tombol <i>submit</i>.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan perubahan data trayek yang dikirim <i>admin</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal mengubah informasi terkait trayek dan angkutan umum karena kesalahan server.</li> <li>2. <i>Admin</i> gagal mengubah informasi terkait trayek dan angkutan umum karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>

<i>Post-condition</i>	<i>Admin</i> telah berhasil mengubah informasi terkait trayek dan data yang baru diubah tampil pada daftar info trayek pada aplikasi.
-----------------------	---

## 21. Skenario Use Case Hapus Info Trayek

Pada skenario *use case* hapus info trayek, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat menghapus data informasi terkait trayek dan angkutan umum yang telah dimasukkan sebelumnya. Skenario *use case* hapus info trayek ditunjukkan pada Tabel 4. 25.

Tabel 4. 25 Skenario *Use Case* Hapus Info Trayek

Hapus Info Trayek	
<i>Objective</i>	<i>Admin</i> dapat menghapus informasi terkait trayek dan angkutan umum yang telah dimasukkan sebelumnya melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu trayek pada aplikasi.</li> <li>2. Sistem menampilkan halaman trayek.</li> <li>3. <i>Admin</i> memilih opsi hapus trayek pada salah satu daftar trayek yang tersedia.</li> <li>4. Sistem menampilkan notifikasi keamanan dengan opsi ya dan tidak.</li> <li>5. <i>Admin</i> memilih opsi ya.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan perubahan data trayek yang dikirim <i>admin</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal menghapus informasi terkait trayek dan</li> </ol>

	<p>angkutan umum karena kesalahan server.</p> <p>2. <i>Admin</i> gagal menghapus informasi terkait trayek dan angkutan umum karena <i>token</i> milik <i>admin</i> tidak valid.</p>
<i>Post-condition</i>	<i>Admin</i> telah berhasil menghapus informasi terkait pengumuman dan data dihapus sudah tidak tampil pada daftar info pengumuman pada aplikasi.

## 22. Skenario Use Case Tambah Info Pengumuman

Pada skenario *use case* tambah info pengumuman, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat menambahkan informasi terkait pengumuman pada Dinas Perhubungan Kota Mojokerto. Skenario *use case* tambah info pengumuman ditunjukkan pada Tabel 4. 26.

Tabel 4. 26 Skenario *Use Case* Tambah Info Pengumuman

Tambah Info Pengumuman	
<i>Objective</i>	<i>Admin</i> dapat menambahkan informasi terkait pengumuman melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu info pada aplikasi.</li> <li>2. Sistem menampilkan halaman info.</li> <li>3. <i>Admin</i> memilih opsi tambah info untuk menambah data info.</li> <li>4. Sistem menampilkan form input untuk data info.</li> <li>5. <i>Admin</i> memasukkan data info yaitu judul, foto, deskripsi dan kategori sesuai dengan yang ingin dimasukkan.</li> <li>6. <i>Admin</i> menekan tombol <i>submit</i>.</li> </ol>

	<ol style="list-style-type: none"> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan data info yang dikirim <i>admin</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal menambahkan informasi terkait pengumuman karena kesalahan server.</li> <li>2. <i>Admin</i> gagal menambahkan informasi terkait pengumuman karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil menambahkan informasi terkait pengumuman dan data yang baru dimasukkan tampil pada daftar info pengumuman pada aplikasi.

### 23. Skenario Use Case Edit Info Pengumuman

Pada skenario *use case edit* info pengumuman, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat mengubah data informasi terkait pengumuman yang ada pada Dinas Perhubungan Kota Mojokerto yang telah dimasukkan sebelumnya. Skenario *use case edit* info pengumuman ditunjukkan pada Tabel 4. 27.

Tabel 4. 27 Skenario Use Case Edit Info Pengumuman

<b>Edit Info Pengumuman</b>	
<i>Objective</i>	<i>Admin</i> dapat mengubah informasi terkait pengumuman yang telah dimasukkan sebelumnya melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu info pada aplikasi.</li> <li>2. Sistem menampilkan halaman info.</li> <li>3. <i>Admin</i> memilih opsi edit info pada salah satu daftar info untuk mengubah data info.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Sistem menampilkan form input untuk edit data info.</li> <li>5. <i>Admin</i> mengubah data info yaitu judul, foto, deskripsi dan kategori sesuai dengan yang ingin dimasukkan.</li> <li>6. <i>Admin</i> menekan tombol <i>submit</i>.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan perubahan data info yang dikirim <i>admin</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal mengubah informasi terkait pengumuman karena kesalahan server.</li> <li>2. <i>Admin</i> gagal mengubah informasi terkait pengumuman karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil mengubah informasi terkait pengumuman dan data yang baru diubah tampil pada daftar info pengumuman pada aplikasi.

#### 24. Skenario Use Case Hapus Info Pengumuman

Pada skenario *use case* hapus info pengumuman, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat menghapus data informasi terkait pengumuman yang ada pada Dinas Perhubungan Kota Mojokerto yang telah dimasukkan sebelumnya. Skenario *use case* hapus info pengumuman ditunjukkan pada Tabel 4. 28.

Tabel 4. 28 Skenario *Use Case* Hapus Info Pengumuman

Hapus Info Pengumuman	
<i>Objective</i>	<i>Admin</i> dapat menghapus informasi terkait pengumuman yang telah dimasukkan sebelumnya melalui aplikasi.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.



<p><i>Main Flow</i></p>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu info pada aplikasi.</li> <li>2. Sistem menampilkan halaman info.</li> <li>3. <i>Admin</i> memilih opsi hapus info pada salah satu daftar info untuk menghapus data info.</li> <li>4. Sistem menampilkan notifikasi keamanan dengan pilihan opsi ya dan tidak.</li> <li>5. <i>Admin</i> memilih opsi ya.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan perubahan data info yang dikirim <i>admin</i>.</li> </ol>
<p><i>Alternative Flow</i></p>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal menghapus informasi terkait pengumuman karena kesalahan server.</li> <li>2. <i>Admin</i> gagal menghapus informasi terkait pengumuman karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<p><i>Post-condition</i></p>	<p><i>Admin</i> telah berhasil menghapus informasi pengumuman dan data dihapus sudah tidak tampil pada daftar info pengumuman pada aplikasi.</p>

**25. Skenario Use Case Verifikasi Izin Penggunaan Jalan**

Pada skenario *use case* verifikasi izin penggunaan jalan, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat memverifikasi izin penggunaan jalan masyarakat melalui aplikasi dan *admin* berhak untuk memutuskan apakah izin valid atau tidak. Skenario *use case* verifikasi izin penggunaan jalan ditunjukkan pada Tabel 4. 29.

Tabel 4. 29 Skenario *Use Case* Verifikasi Izin Penggunaan Jalan

<p><b>Verifikasi Izin Penggunaan Jalan</b></p>	
<p><i>Objective</i></p>	<p><i>Admin</i> dapat melakukan verifikasi izin penggunaan jalan masyarakat dengan benar.</p>
<p><i>Actor</i></p>	<p><i>Admin</i></p>

<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu izin penggunaan jalan.</li> <li>2. Sistem menampilkan halaman izin penggunaan jalan.</li> <li>3. <i>Admin</i> melakukan verifikasi pada salah satu izin penggunaan jalan dengan memilih salah satu opsi yaitu terima atau tolak.</li> <li>4. Sistem menampilkan notifikasi keamanan dengan pilihan opsi ya dan tidak.</li> <li>5. <i>Admin</i> memilih opsi ya.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan perubahan status pada izin penggunaan jalan.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal melakukan verifikasi izin penggunaan jalan karena kesalahan server.</li> <li>2. <i>Admin</i> gagal melakukan verifikasi izin penggunaan jalan data karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil melakukan verifikasi izin penggunaan jalan dengan benar.

## 26. Skenario Use Case Lihat Survei Kepuasan Masyarakat

Pada skenario *use case* lihat survei kepuasan masyarakat, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat melihat hasil survei kepuasan masyarakat yang didapat dari *user*. Skenario *use case* lihat survei kepuasan masyarakat ditunjukkan pada Tabel 4. 30.

Tabel 4. 30 Skenario *Use Case* Lihat Survei Kepuasan Masyarakat

Lihat Survei Kepuasan Masyarakat	
<i>Objective</i>	<i>Admin</i> dapat melihat hasil survei kepuasan masyarakat yang didapat dari <i>user</i> .
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu hasil survei.</li> <li>2. Sistem melakukan verifikasi <i>token</i>.</li> <li>3. Sistem menampilkan halaman hasil survei yang berisi hasil perhitungan data survei yang telah masuk pad Dinas Perhubungan Kota Mojokerto.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal melihat hasil survei kepuasan masyarakat karena kesalahan server.</li> <li>2. <i>Admin</i> gagal melihat hasil survei kepuasan masyarakat karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil melihat hasil survei kepuasan masyarakat yang tersedia pada daftar.

### 27. Skenario *Use Case* Tambah Survei Kepuasan Masyarakat

Pada skenario *use case* tambah survei kepuasan masyarakat, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat menambahkan survei kepuasan masyarakat baru untuk *user*. Skenario *use case* tambah survei kepuasan masyarakat ditunjukkan pada Tabel 4. 31.

Tabel 4. 31 Skenario *Use Case* Tambah Survei Kepuasan Masyarakat

Tambah Survei Kepuasan Masyarakat	
<i>Objective</i>	<i>Admin</i> dapat menambahkan survei kepuasan masyarakat untuk <i>user</i> .
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu survei pada aplikasi.</li> <li>2. Sistem menampilkan halaman survei.</li> <li>3. <i>Admin</i> memilih opsi tambah survei untuk menambahkan survei baru.</li> <li>4. Sistem menampilkan form input untuk data survei.</li> <li>5. <i>Admin</i> memasukkan data survei berupa pertanyaan dan pilihan jawaban pada form yang tersedia.</li> <li>6. <i>Admin</i> menekan tombol <i>submit</i>.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan data pertanyaan survei.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal menambahkan survei kepuasan masyarakat karena terdapat <i>input</i> yang tidak diisi.</li> <li>2. <i>Admin</i> gagal menambahkan survei kepuasan masyarakat karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil menambahkan survei kepuasan masyarakat baru dan berhasil ditampilkan.

## 28. Skenario Use Case Edit Survei Kepuasan Masyarakat

Pada skenario *use case* edit survei kepuasan masyarakat, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat mengubah survei kepuasan masyarakat yang telah dimasukkan sebelumnya. Skenario *use case* edit survei kepuasan masyarakat ditunjukkan pada Tabel 4. 32.

Tabel 4. 32 Skenario *Use Case* Edit Survei Kepuasan Masyarakat

<b>Edit Survei Kepuasan Masyarakat</b>	
<i>Objective</i>	<i>Admin</i> dapat mengubah survei kepuasan masyarakat yang telah dimasukkan sebelumnya.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu survei pada aplikasi.</li> <li>2. Sistem menampilkan halaman survei.</li> <li>3. <i>Admin</i> memilih opsi edit survei pada salah satu daftar pertanyaan.</li> <li>4. Sistem menampilkan form input untuk edit data survei.</li> <li>5. <i>Admin</i> mengubah data survei berupa pertanyaan dan pilihan jawaban pada form yang tersedia.</li> <li>6. <i>Admin</i> menekan tombol <i>submit</i>.</li> <li>7. Sistem melakukan verifikasi <i>token</i>.</li> <li>8. Sistem menyimpan perubahan data pertanyaan survey.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal mengubah survei kepuasan masyarakat karena terdapat <i>input</i> yang tidak diisi.</li> </ol>

	2. <i>Admin</i> mengubah survei kepuasan masyarakat karena <i>token</i> milik <i>admin</i> tidak valid.
<i>Post-condition</i>	<i>Admin</i> telah berhasil mengubah survei kepuasan masyarakat dan perubahan berhasil ditampilkan.

**29. Skenario Use Case Hapus Survei Kepuasan Masyarakat**

Pada skenario *use case* hapus survei kepuasan masyarakat, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat menghapus survei kepuasan masyarakat yang telah dimasukkan sebelumnya. Skenario *use case* hapus survei kepuasan masyarakat ditunjukkan pada Tabel 4. 33.

**Tabel 4. 33 Skenario Use Case Hapus Survei Kepuasan Masyarakat**

<b>Hapus Survei Kepuasan Masyarakat</b>	
<i>Objective</i>	<i>Admin</i> dapat menghapus survei kepuasan masyarakat yang telah dimasukkan sebelumnya.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu survei pada aplikasi.</li> <li>2. Sistem menampilkan halaman survei.</li> <li>3. <i>Admin</i> memilih opsi hapus survei pada salah satu daftar pertanyaan.</li> <li>4. Sistem menampilkan notifikasi keamanan dengan opsi ya dan tidak.</li> <li>5. <i>Admin</i> memilih opsi ya.</li> <li>6. Sistem melakukan verifikasi <i>token</i>.</li> <li>7. Sistem menyimpan perubahan data pertanyaan survey.</li> </ol>

<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal menghapus survei kepuasan masyarakat karena kesalahan server.</li> <li>2. <i>Admin</i> gagal menghapus survei kepuasan masyarakat karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil menghapus survei kepuasan masyarakat dan tidak tampil pada daftar.

### 30. Skenario Use Case Lihat Kritik dan Saran

Pada skenario *use case* lihat kritik dan saran, *admin* berhasil masuk ke dalam aplikasi. *Admin* dapat melihat semua kritik dan saran yang telah diterima dari *user* yang telah melakukan kritik dan saran. Skenario *use case* lihat kritik dan saran ditunjukkan pada Tabel 4. 34.

Tabel 4. 34 Skenario Use Case Lihat Kritik dan Saran

Lihat Kritik dan Saran	
<i>Objective</i>	<i>Admin</i> dapat melihat semua kritik dan saran yang diterima dari user.
<i>Actor</i>	<i>Admin</i>
<i>Pre-condition</i>	<i>Admin</i> telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> memilih menu kritik dan saran.</li> <li>2. Sistem melakukan verifikasi <i>token</i>.</li> <li>3. Sistem menampilkan halaman kritik dan saran yang didapat dari <i>user</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. <i>Admin</i> gagal melihat kritik dan saran karena kesalahan server.</li> <li>2. <i>Admin</i> gagal melihat kritik dan saran karena <i>token</i> milik <i>admin</i> tidak valid.</li> </ol>
<i>Post-condition</i>	<i>Admin</i> telah berhasil melihat semua kritik dan saran yang diterima dari user.

### 31. Skenario Use Case Logout

Pada skenario *use case logout*, aktor baik *admin* atau *user* telah melakukan *login* pada aplikasi. Untuk mengakhiri sesi yang telah dibuat ketika melakukan *login* pengguna dapat melakukan *logout* dan mengakhiri sesi penggunaan aplikasi. Skenario *use case logout* ditunjukkan pada Tabel 4. 35.

Tabel 4. 35 Skenario *Use Case Logout*

<b>Logout</b>	
<i>Objective</i>	Aktor dapat melakukan <i>logout</i> dan mengakhiri sesi pemakaian aplikasi.
<i>Actor</i>	<i>Admin, User</i>
<i>Pre-condition</i>	Aktor telah berhasil melakukan autentikasi pada aplikasi dan berada pada halaman beranda.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil pada aplikasi.</li> <li>2. Sistem menampilkan halaman profil.</li> <li>3. Aktor memilih sub menu <i>logout</i> untuk keluar dari sesi pemakaian aplikasi.</li> <li>4. Sistem menghapus data <i>login</i> actor dan mengarahkan ke halaman <i>login</i>.</li> </ol>
<i>Alternative Flow</i>	Aktor gagal melakukan <i>logout</i> karena kesalahan server.
<i>Post-condition</i>	Aktor telah berhasil keluar dari aplikasi dan mengakhiri sesi pemakaian aplikasi dan diarahkan ke halaman <i>login</i> .

## BAB V PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan Sistem

Bagian Perancangan aplikasi pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* ini diawali dengan perancangan arsitektur yang menggambarkan sistem dengan UML (*Unified Modelling Language*) menggunakan diagram *sequence* dan diagram *class*, perancangan algoritme algoritme dari setiap *method* atau fungsi yang ada pada setiap *class* di diagram *class* dan perancangan antarmuka yang menggambarkan tampilan aplikasi serta semua komponen pembangun antarmuka.

#### 5.1.1 Perancangan Arsitektur

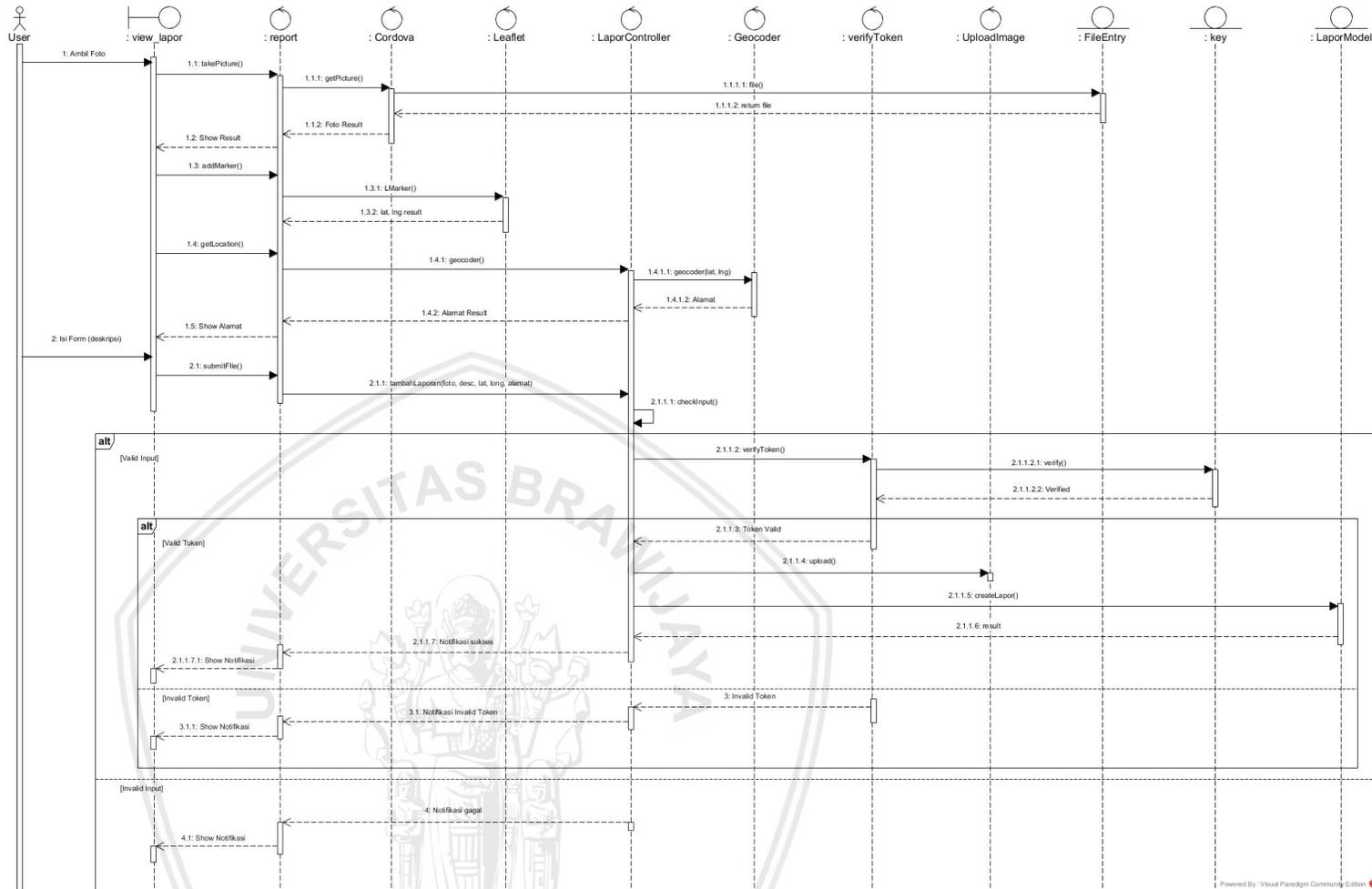
Pada bagian perancangan arsitektur aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* ini berfokus pada perancangan menggunakan UML (*Unified Modelling Language*) yaitu Diagram *Sequence* dan Diagram *Class* untuk menunjukkan Gambaran bagaimana elemen/komponen fungsional perangkat lunak disusun, diorganisasi dan distrukturkan.

##### 1. *Sequence Diagram*

*Sequence diagram* akan memodelkan bagaimana hubungan antar objek dalam aplikasi berdasarkan waktu dan menerapkan fungsi-fungsi yang ada dalam objek pada diagram *use case*. Berikut adalah lima contoh diagram *sequence* yang diterapkan pada penelitian ini yaitu tambah laporan keluhan, lihat info trayek, izin penggunaan jalan, survei kepuasan masyarakat dan kritik & saran.

##### a) Tambah Laporan Keluhan

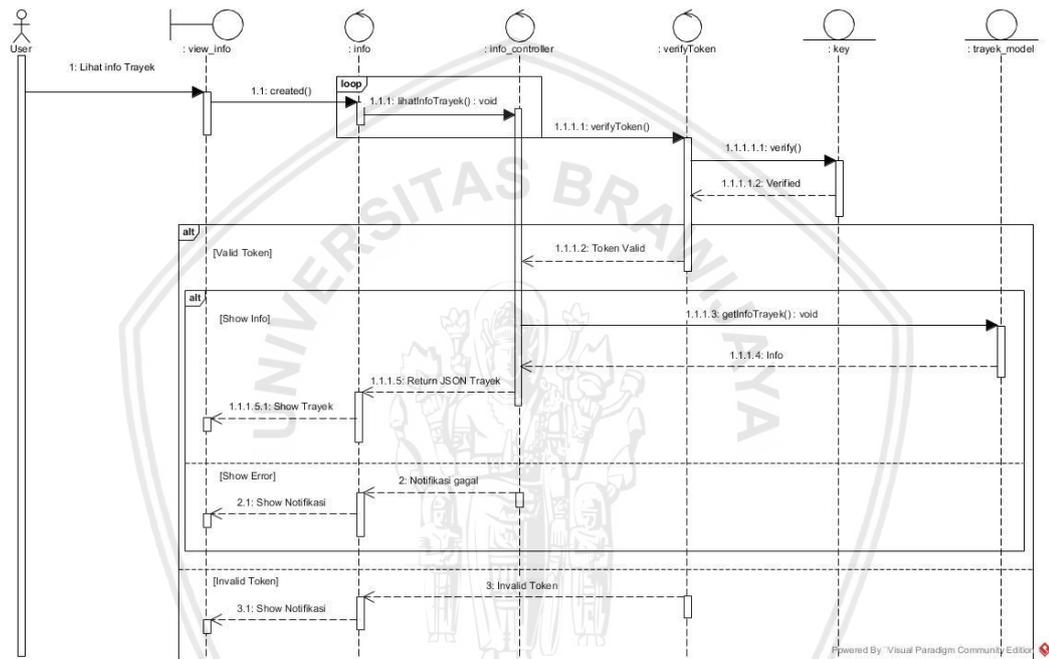
*Sequence diagram* laporan keluhan menunjukkan interaksi antara aktor user dan sistem pada saat user menambahkan laporan keluhan. Untuk menambahkan laporan keluhan user harus membuka menu lapor dan melakukan pengisian form terkait yaitu deskripsi mengenai laporan keluhan dan foto yang dilengkapi dengan *metadata geotagging* berupa koordinat *latitude* dan *longitude* yang selanjutnya akan diproses oleh Geocoder untuk melakukan proses *reverse geocode* dengan *method* *geocode()* yang mengembalikan data lokasi alamat yang selanjutnya user menambahkan laporan keluhan dengan *method* *tambahLaporan()* yang mengharuskan mengecek *token* yang didapat saat melakukan login dengan *method* *verifyToken()*. Selanjutnya akan dilakukan pengecekan apakah input yang dimasukkan user sudah sesuai dengan ketentuan. Jika sudah sesuai *method* *saveLaporan()* akan dipanggil dan menyimpan laporan pada database, dan jika tidak sesuai maka akan dimunculkan pesan error pada halaman user. Diagram *sequence* laporan keluhan ditunjukkan pada Gambar 5. 1.



Gambar 5. 1 Sequence Diagram Tambah Laporan Keluhan

**b) Lihat Info Trayek**

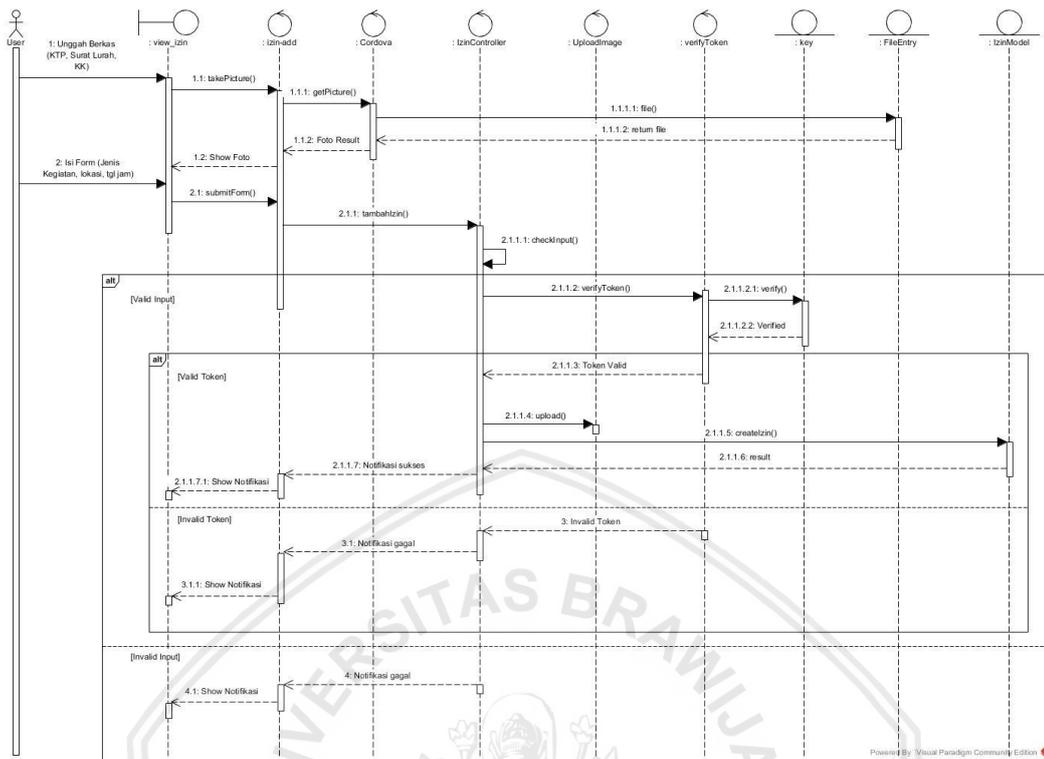
Sequence diagram laporan keluhan menunjukkan interaksi antara aktor user dan sistem pada saat user ingin melihat informasi trayek yang ada pada aplikasi. Untuk melihat info user harus membuka menu info dan aplikasi akan menampilkan informasi sesuai kategori yang diproses dengan *method* lihatInfoTrayek() yang mengharuskan mengecek *token* yang didapat saat melakukan login dengan *method* verifyToken(). Selanjutnya Jika tidak ada masalah *method* getInfoTrayek() akan dipanggil dan menampilkan informasi berdasarkan kategori, dan jika terjadi masalah maka akan dimunculkan pesan error pada halaman user. Diagram sequence lihat info ditunjukkan pada Gambar 5. 2.



Gambar 5. 2 Sequence Diagram Lihat Info Trayek

**c) Izin Penggunaan Jalan**

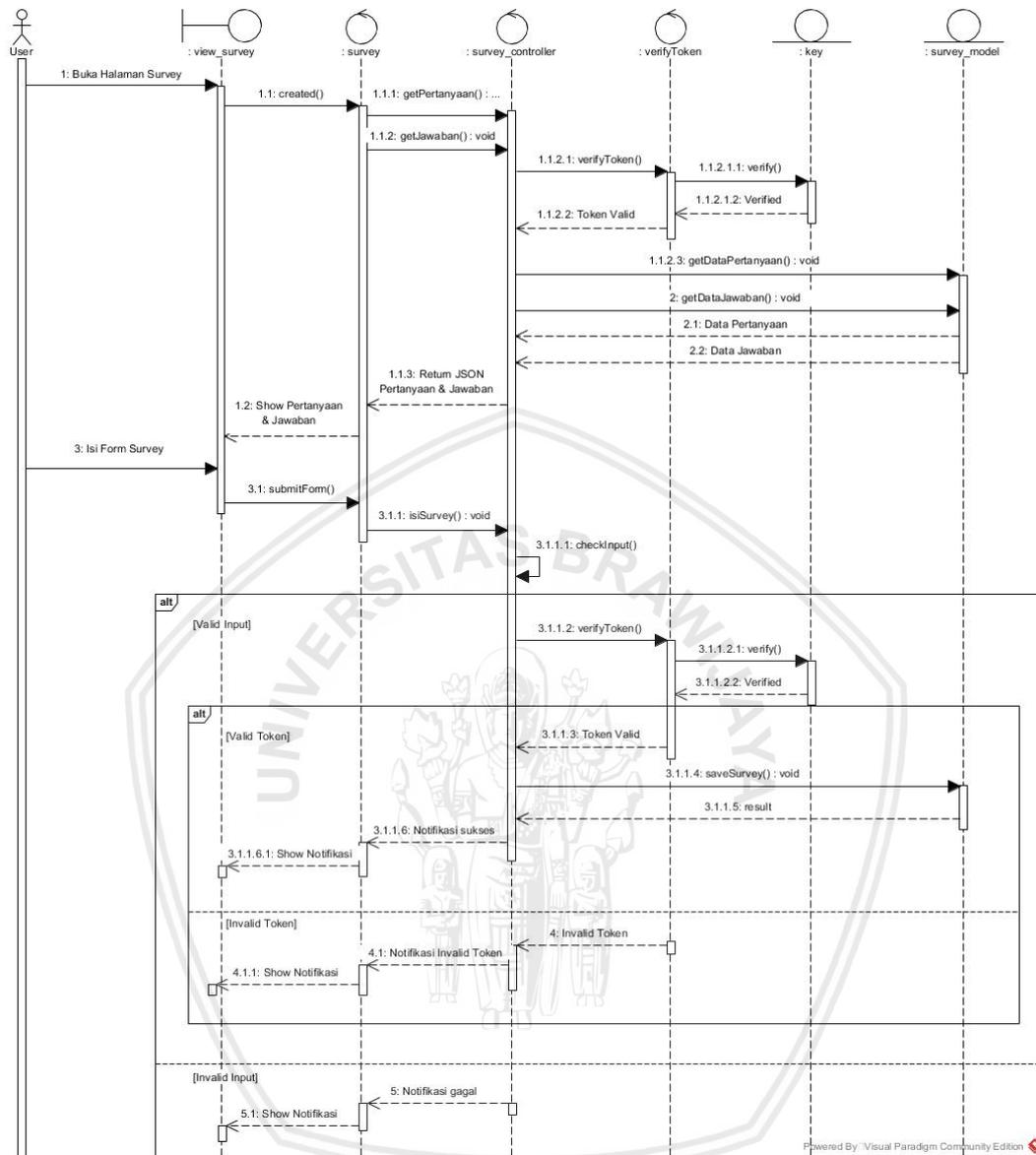
Sequence diagram izin penggunaan jalan menunjukkan interaksi antara aktor user dan sistem pada saat user melakukan izin penggunaan jalan melalui aplikasi. Untuk melakukan izin penggunaan jalan user harus membuka menu izin penggunaan jalan dan melakukan pengisian form terkait seperti deskripsi dan berkas unggah yang telah ditetapkan yang selanjutnya akan diproses dengan *method* tambahIzin() yang mengharuskan mengecek *token* yang didapat saat melakukan login dengan *method* verifyToken(). Selanjutnya akan dilakukan pengecekan apakah input yang dimasukkan user sudah sesuai dengan ketentuan. Jika sudah sesuai *method* createIzin() akan dipanggil dan menyimpan laporan pada database lalu menampilkan pesan berhasil pada user, dan jika tidak sesuai maka akan dimunculkan pesan error pada halaman user. Diagram sequence izin penggunaan jalan ditunjukkan pada Gambar 5. 3.



Gambar 5. 3 Sequence Diagram Izin Penggunaan Jalan

**d) Survei Kepuasan Masyarakat**

Sequence diagram survei kepuasan masyarakat menunjukkan interaksi antara aktor *user* dan sistem pada saat *user* melakukan survei kepuasan masyarakat melalui aplikasi. Untuk melakukan survei kepuasan masyarakat *user* harus membuka menu survei yang berisikan data pertanyaan dan jawaban yang didapat dari *method* *getPertanyaan()* dan *getJawaban()* yang digunakan *user* untuk mengisi survei kepuasan masyarakat dan melakukan pengisian survei yang selanjutnya akan diproses dengan *method* *simpanSurvey()* yang mengharuskan mengecek token yang didapat saat melakukan login dengan *method* *verifyToken()*. Selanjutnya akan dilakukan pengecekan apakah input yang dimasukkan *user* sudah sesuai dengan ketentuan dengan *method* *checkInput()*. Jika sudah sesuai *method* *createSurvey()* akan dipanggil dan menyimpan survei pada database lalu menampilkan pesan berhasil pada *user*, dan jika tidak sesuai maka akan dimunculkan pesan error pada halaman *user*. Diagram sequence survei kepuasan masyarakat ditunjukkan pada Gambar 5. 4.

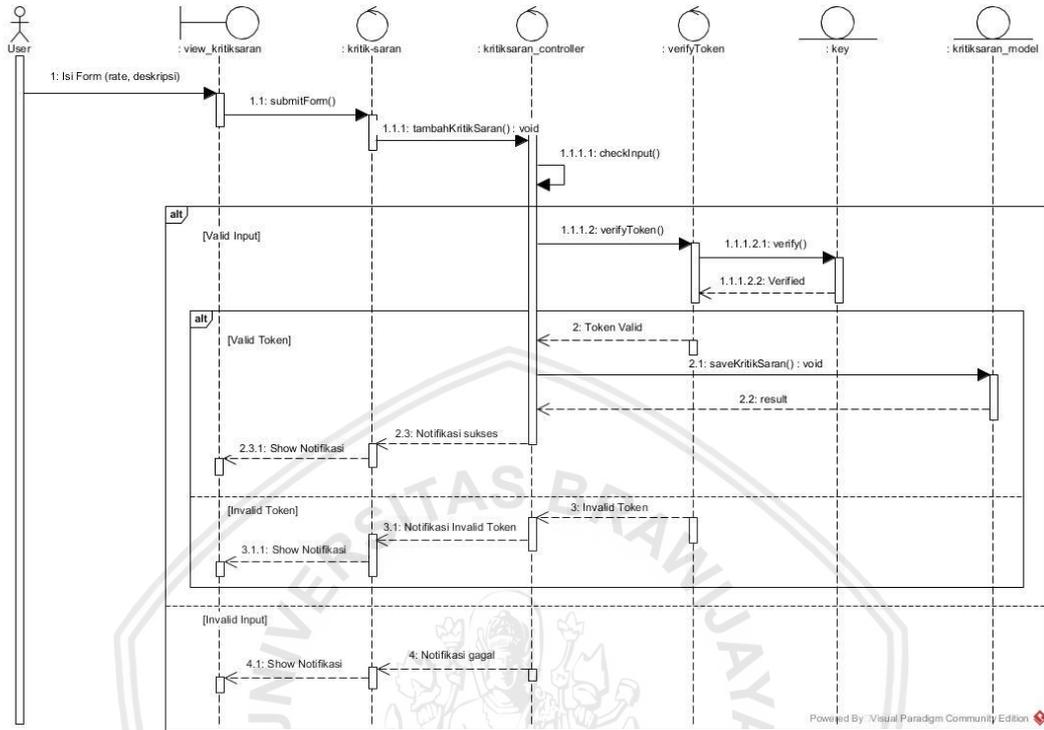


Gambar 5. 4 Sequence Diagram Survei Kepuasan Masyarakat

e) Tambah Kritik dan Saran

Sequence diagram tambah kritik dan saran menunjukkan interaksi antara aktor *user* dan sistem pada saat *user* menambahkan kritik dan saran melalui aplikasi. Untuk menambahkan kritik dan saran *user* harus membuka menu kritik saran dan melakukan pengisian form yang telah tersedia yang selanjutnya akan diproses dengan *method* tambahKritikSaran() yang mengharuskan mengecek *token* yang didapat saat melakukan login dengan *method* verifyToken(). Selanjutnya akan dilakukan pengecekan apakah input yang dimasukkan *user* sudah sesuai dengan ketentuan dengan *method* checkInput(). Jika sudah sesuai *method* createKritikSaran() akan dipanggil dan menyimpan kritik saran pada database lalu menampilkan pesan berhasil pada *user*, dan jika tidak sesuai maka akan

dimunculkan pesan error pada halaman user. Diagram tambah kritik dan saran ditunjukkan pada Gambar 5. 5.

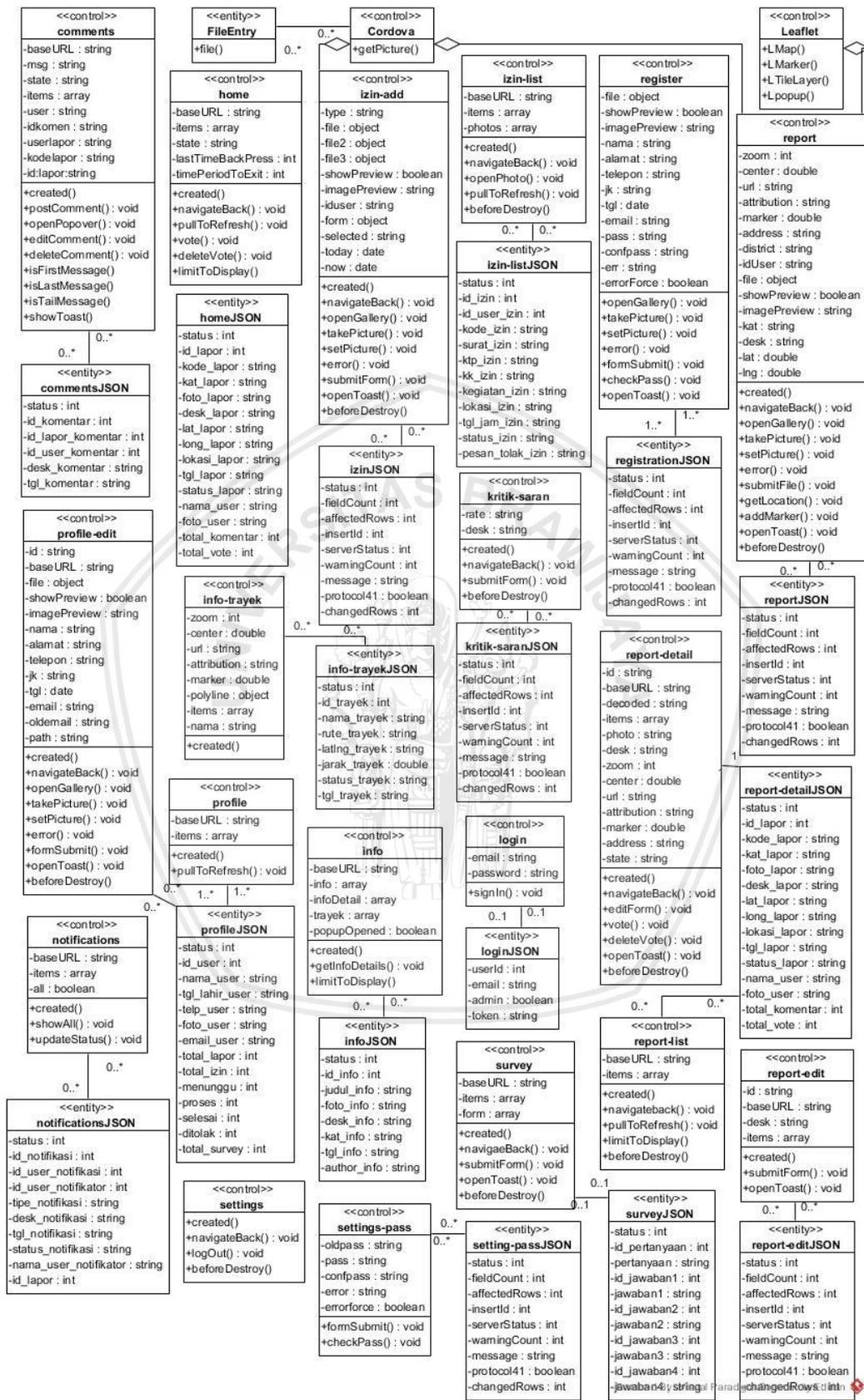


Gambar 5. 5 Sequence Diagram Kritik dan Saran

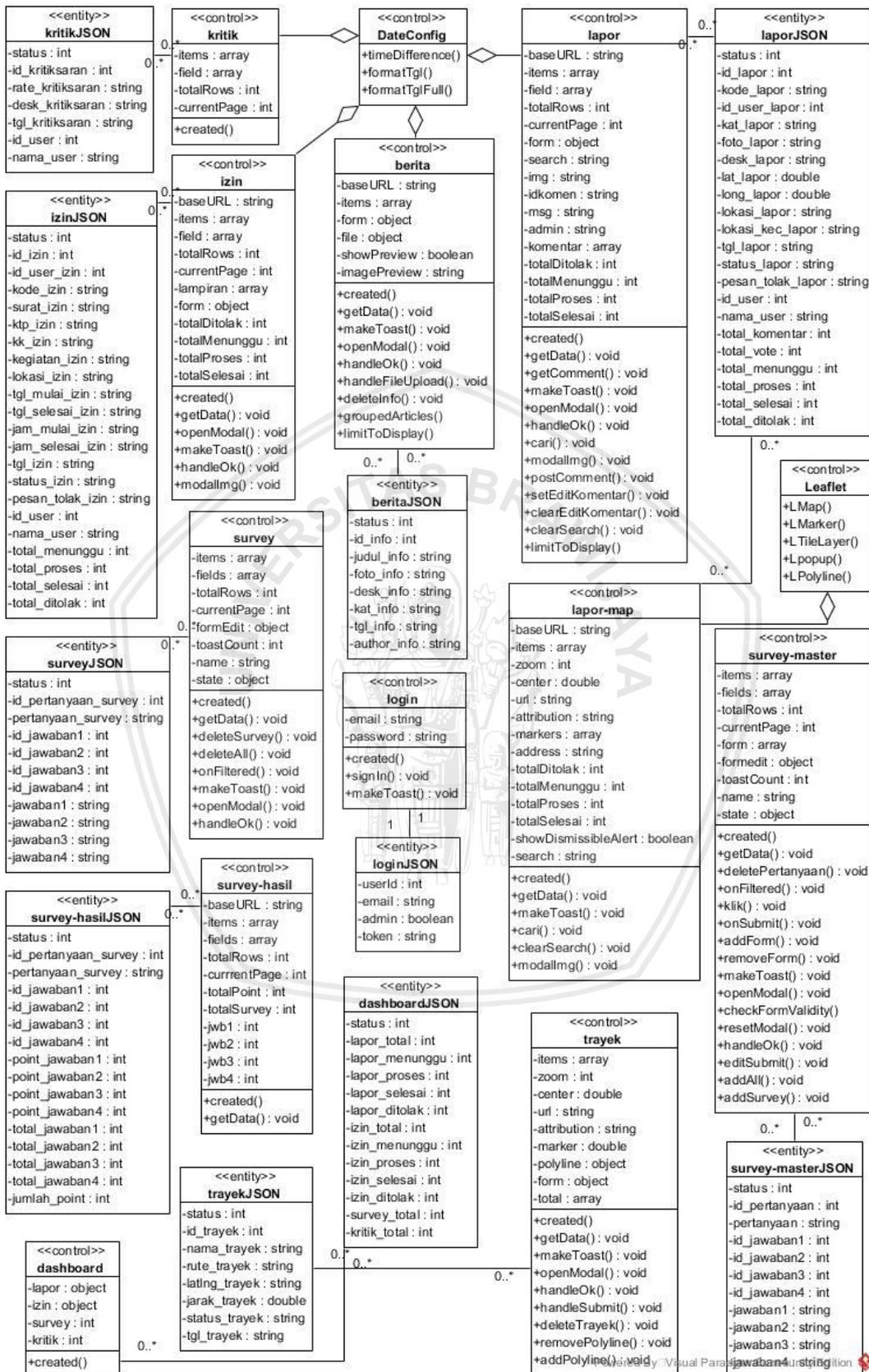
## 2. Class Diagram

*Class* diagram mendeskripsikan struktur dan deskripsi *class*, *package* dan objek serta hubungan satu sama lain. Diagram *class* pada aplikasi pelaporan keluhan masyarakat ini terbagi menjadi tiga bagian yaitu aplikasi berbasis *mobile* yang digunakan oleh *user* atau masyarakat, aplikasi berbasis web yang digunakan oleh *admin* pada Dinas Perhubungan Kota Mojokerto dan RESTful web *services* sebagai API (*Application Programming Interface*) yang ditunjukkan pada Gambar 5. 6, Gambar 5. 7 dan Gambar 5. 8.





Gambar 5. 6 Diagram Class Aplikasi Pelaporan Mobile



Gambar 5. 7 Diagram Class Aplikasi Pelaporan Web Admin



## 5.1.2 Perancangan Algoritme

Pada aplikasi pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* ini memiliki berbagai proses. Proses-proses tersebut merupakan algoritme yang diimplementasikan pada aplikasi. Pada perancangan algoritme hanya akan diperlihatkan beberapa algoritme yang menjadi pokok dalam sistem, diantaranya yaitu tambah laporan keluhan, lihat info trayek, izin penggunaan jalan, survei kepuasan masyarakat dan kritik & saran.

### 1. Algoritme Tambah Laporan Keluhan

Algoritme Tambah Laporan Keluhan berfungsi untuk menambahkan laporan keluhan kepada Dinas Perhubungan Kota Mojokerto. Algoritme ini memanfaatkan metode *geotagging* untuk menambahkan lokasi berupa *latitude* dan *longitude* pada foto yang dipilih atau diambil menggunakan kamera *smartphone*. Algoritme Tambah Laporan Keluhan ditunjukkan pada Tabel 5. 1.

Tabel 5. 1 Algoritme Tambah Laporan Keluhan

lapor controller	
No	Pseudocode
1	Mulai
2	
3	Inisialisasi userInfo
3	Inisialisasi verifyToken
3	Inisialisasi geocoder
4	Inisialisasi database
5	
5	Panggil verifyToken(userInfo)
5	
6	If (verifyToken == false) {
7	Return false
8	} else {
9	Return true
10	}
11	Endif
11	
12	Foto = file foto
13	Data = ambil input foto, kategori, deskripsi, lokasi latlng
14	Panggil geocoder(latlng)
15	If (metadata exif foto == true) {
16	Return lokasi foto
17	} else {
18	Lokasi = ambil latlng dari input
19	Panggil geocoder(lokalasi)
20	Return lokasi foto
21	}
22	Endif
23	
24	If (data == lengkap) {
25	Return data
26	Return true
27	} else {
28	data tidak lengkap
29	Return false
30	}

31	Endif
32	
33	Simpan data laporan ke database
34	
35	Selesai

## 2. Algoritme Lihat Info Trayek

Algoritme Lihat Info Trayek berfungsi untuk menampilkan informasi terkait trayek, angkutan umum dan rutenya pada masyarakat. Algoritme Lihat Info Trayek ditunjukkan pada Tabel 5. 2.

Tabel 5. 2 Algoritme Lihat Info Trayek

info_controller	
No	Pseudocode
1	Mulai
2	
3	Inisialisasi userInfo
3	Inisialisasi verifyToken
3	Inisialisasi database
4	
5	Panggil verifyToken(userInfo)
5	
5	If (verifyToken == false) {
6	Return false
7	} else {
8	Return true
9	}
10	Endif
11	
12	Info = jumlah total info trayek
13	
14	For(info) {
15	Info = Ambil info trayek dari database
16	Return info
17	}
18	Endfor
19	
20	Tampilkan Info
21	
22	Selesai

## 3. Algoritme Izin Penggunaan Jalan

Algoritme Izin Penggunaan Jalan berfungsi untuk melakukan izin penggunaan jalan melalui Dinas Perhubungan Kota Mojokerto dengan mengunggah berkas-berkas yang diminta yaitu scan surat rekomendasi dari lurah setempat dan scan kartu tanda penduduk (KTP). Algoritme Izin Penggunaan Jalan ditunjukkan pada Tabel 5. 3.

Tabel 5. 3 Algoritme Izin Penggunaan Jalan

izin controller	
No	Pseudocode
1	Mulai
2	
3	Inisialisasi userInfo
3	Inisialisasi verifyToken
3	Inisialisasi database
4	
5	Panggil verifyToken(userInfo)
5	
5	If (verifyToken == false) {
6	Return false
7	} else {
8	Return true
9	}
10	Endif
11	
12	Data = ambil data input
13	Img = ambil data gambar upload
14	
15	If (img == data gambar) {
16	Return true
17	} else {
18	Img tidak valid
19	Return false
20	}
21	Endif
22	
23	Simpan data izin penggunaan jalan ke server
24	
25	Selesai

#### 4. Algoritme Survei Kepuasan Masyarakat

Algoritme Survei Kepuasan Masyarakat berfungsi bagi masyarakat untuk mengisi survei terkait kepuasan masyarakat terhadap fasilitas terkait perhubungan dan layanan pada Dinas Perhubungan Kota Mojokerto. Algoritme Survei Kepuasan Masyarakat ditunjukkan pada Tabel 5. 4.

Tabel 5. 4 Algoritme Survei Kepuasan Masyarakat

survey controller	
No	Pseudocode
1	Mulai
2	
3	Inisialisasi userInfo
3	Inisialisasi verifyToken
3	Inisialisasi database
4	
5	Panggil verifyToken(userInfo)
5	
5	If (verifyToken == false) {
6	Return false
7	} else {
8	Return true



```

9   }
10  Endif
11
12  Survey = jumlah total set pertanyaan dan jawaban
13
14  For(survey) {
15      survey = Ambil data pertanyaan dan jawaban dari database
16      Return survey
17  }
18  Endfor
19
20  Hasil = data hasil pengisian survey
21
22  If (hasil == lengkap) {
23      Retrurn hasil
24      Return true
25  } else {
26      hasil tidak lengkap
27      Return false
28  }
29  Endif
30
31  Simpan hasil ke database
32
33  Selesai

```

### 5. Algoritme Kritik dan Saran

Algoritme Kritik dan Saran berfungsi bagi masyarakat untuk memberikan *feedback* atau kritik dan saran pada Dinas Perhubungan Kota Mojokerto agar meningkatkan kualitas layanan dan fasilitasnya. Algoritme Kritik dan Saran ditunjukkan pada Tabel 5. 5.

Tabel 5. 5 Algoritme Kritik dan Saran

kritiksaran_controller	
No	Pseudocode
1	Mulai
2	
3	Inisialisasi userInfo
3	Inisialisasi verifyToken
3	Inisialisasi database
4	
5	Panggil verifyToken(userInfo)
5	
5	If (verifyToken == false) {
6	Return false
7	} else {
8	Return true
9	}
10	Endif
11	
12	data = ambil data input
13	
14	If (data == lengkap) {
15	Retrun data
16	Return true

```

17 } else {
18     data tidak lengkap
19     Return false
20 }
21 Endif
22
23 Simpan data ke database
24
25 Selesai

```

### 5.1.3 Perancangan Antarmuka

Pada proses perancangan antarmuka aplikasi pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* ini dilakukan dengan merancang desain tata letak dan tampilan antarmuka aplikasi yang digunakan untuk berinteraksi dengan user. Pada perancangan antarmuka ini hanya akan diperlihatkan beberapa algoritme yang menjadi pokok dalam sistem, diantaranya yaitu tambah laporan keluhan, lihat info trayek, izin penggunaan jalan, survei kepuasan masyarakat dan kritik & saran.

#### 1. Perancangan Antarmuka Tambah Laporan Keluhan

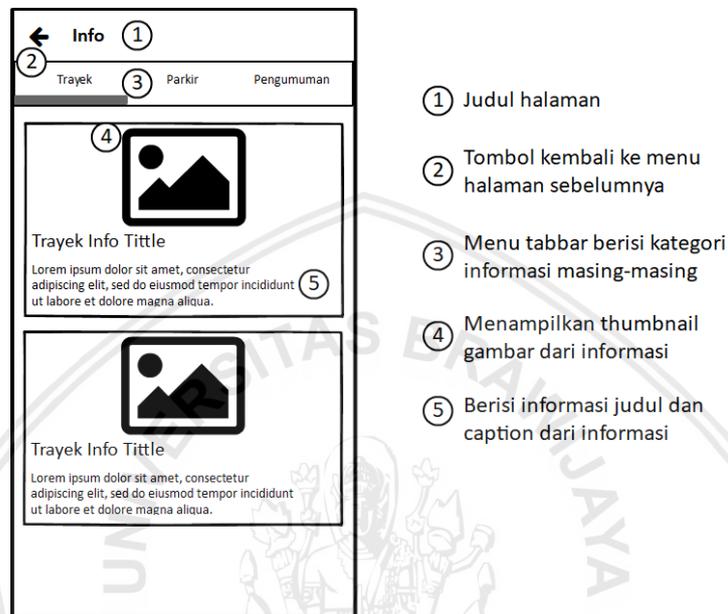
Perancangan antarmuka Tambah Laporan Keluhan adalah halaman untuk *user* atau pengguna aplikasi melakukan pelaporan keluhan. Halaman ini berisikan berbagai *input* yang diperlukan untuk melakukan pelaporan yaitu input foto, kategori, lokasi, dan deskripsi dari laporan keluhan. Perancangan antarmuka Tambah Laporan Keluhan ditunjukkan pada Gambar 5. 9.



Gambar 5. 9 Perancangan Antarmuka Tambah Laporan Keluhan

## 2. Perancangan Antarmuka Lihat Info Trayek

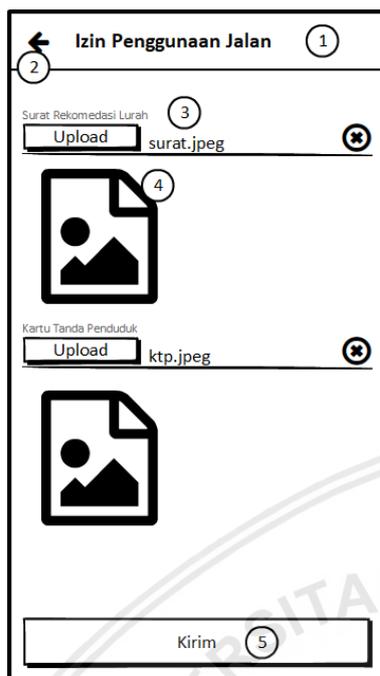
Perancangan antarmuka Lihat Info Trayek adalah halaman untuk *user* atau pengguna aplikasi untuk mengakses informasi terkait trayek yaitu angkutan umum dan rutenya. Halaman ini berisikan menu kategori informasi dan daftar informasi yang ada. Perancangan antarmuka Lihat Info Trayek ditunjukkan pada Gambar 5. 10.



Gambar 5. 10 Perancangan Antarmuka Lihat Info Trayek

## 3. Perancangan Antarmuka Izin Penggunaan Jalan

Perancangan antarmuka Izin Penggunaan Jalan adalah halaman untuk *user* atau pengguna aplikasi melakukan pengajuan izin untuk penggunaan jalan. Halaman ini berisikan *input* untuk mengunggah berkas yang diperlukan untuk mendapatkan izin penggunaan jalan. Perancangan antarmuka Izin Penggunaan Jalan ditunjukkan pada Gambar 5. 11.

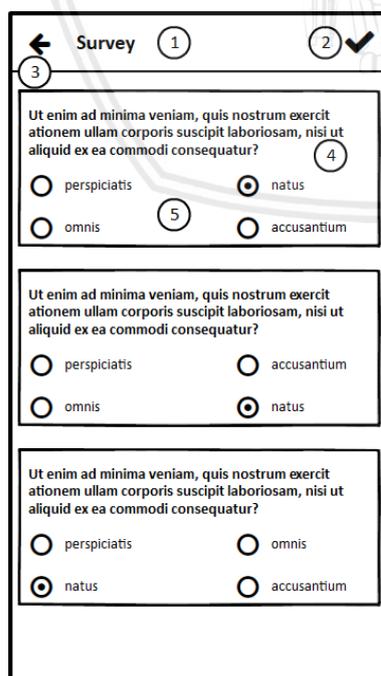


- ① Judul halaman
- ② Tombol kembali ke menu halaman sebelumnya
- ③ Input untuk unggah berkas yang diperlukan
- ④ Menampilkan preview dari gambar yang di unggah
- ⑤ Tombol untuk submit form izin penggunaan jalan

Gambar 5. 11 Perancangan Antarmuka Izin Penggunaan Jalan

#### 4. Perancangan Antarmuka Survei Kepuasan Masyarakat

Perancangan antarmuka Survei Kepuasan Masyarakat adalah halaman untuk user atau pengguna aplikasi melakukan pengisian survei yang ada. Halaman ini berisikan pertanyaan-pertanyaan dan pilihan jawaban yang akan diisi oleh. Perancangan antarmuka Survei Kepuasan Masyarakat ditunjukkan pada Gambar 5. 12.

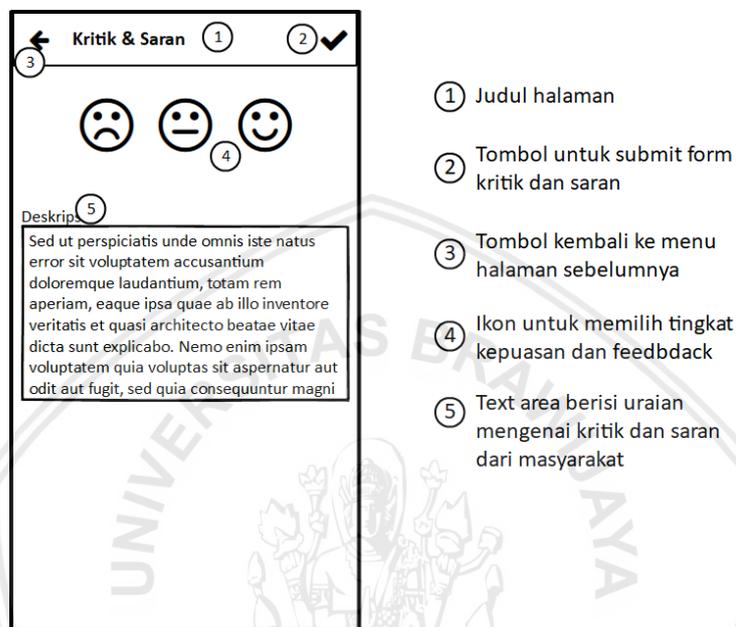


- ① Judul halaman
- ② Tombol untuk submit form survey
- ③ Tombol kembali ke menu halaman sebelumnya
- ④ Text berisi pertanyaan survey
- ⑤ Radio button berisi pilihan jawaban yang tersedia untuk menjawab pertanyaan survey

Gambar 5. 12 Perancangan Antarmuka Survei Kepuasan Masyarakat

## 5. Perancangan Antarmuka Kritik dan Saran

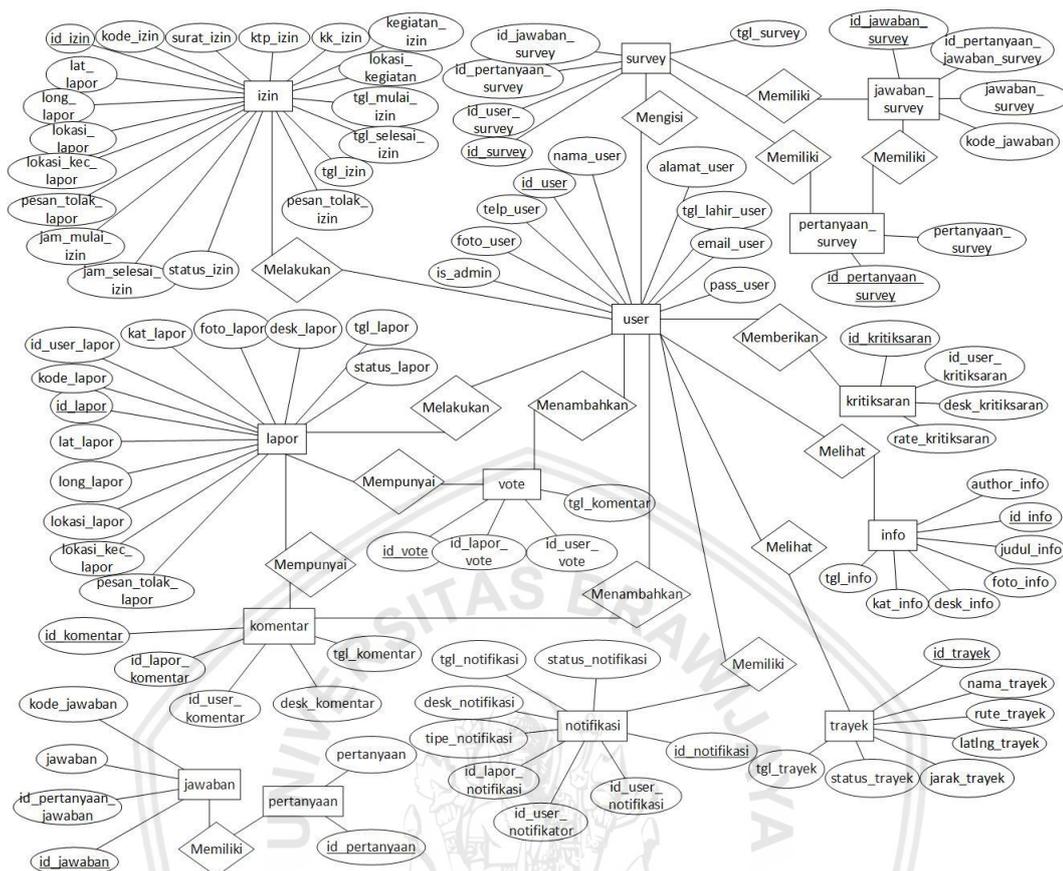
Perancangan antarmuka Kritik dan Saran adalah halaman untuk *user* atau pengguna aplikasi melakukan kritik dan saran. Halaman ini berisikan ikon untuk memilih tingkat kepuasan dan *input* berupa text area untuk menguraikan deskripsi kritik dan saran. Perancangan antarmuka Kritik dan Saran ditunjukkan pada Gambar 5. 13.



Gambar 5. 13 Perancangan Antarmuka Kritik dan Saran

### 5.1.4 Perancangan Basis Data

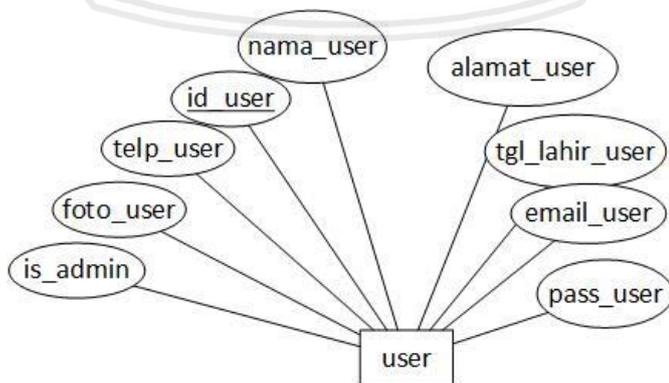
Perancangan dari basis data akan dibuat menggunakan ERD(*Entity Relationship Diagram*). ERD merupakan sebuah diagram yang berfungsi sebagai memodelkan struktur data dan hubungan antar data yaitu berupa sebuah *entity* dan *relationship* yang ditunjukkan pada Gambar 5. 14.



Gambar 5. 14 Entity Relationship Diagram Aplikasi Pelaporan

Pada Gambar 5. 14 terdapat entitas sebanyak empat belas yaitu user, laporan, izin, kritiksaran, komentar, vote, info, trayek, notifikasi, pertanyaan, jawaban, pertanyaan\_survey, jawaban\_survey dan survey.

Entitas user memiliki atribut yang berisikan data terkait pengguna seperti yang ditunjukkan pada Gambar 5. 15.



Gambar 5. 15 Detail Entitas User

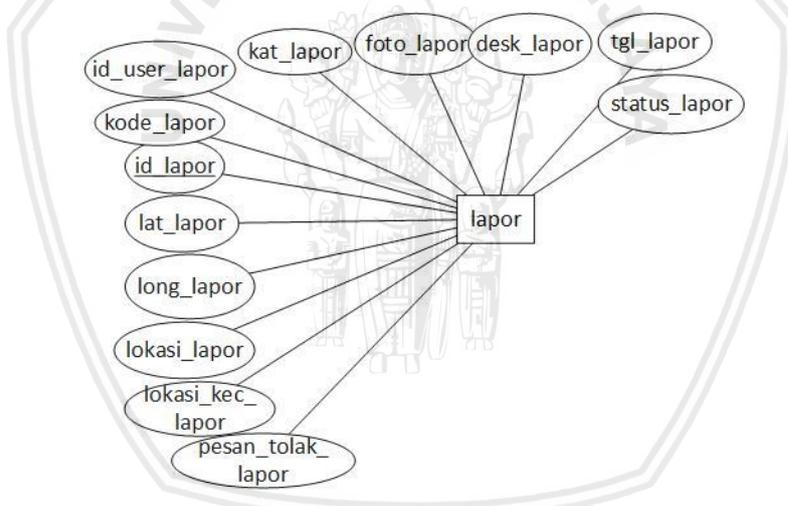
Penjelasan lengkap terkait atribut-atribut pada entitas user dapat dilihat pada Tabel 5. 6.



Tabel 5. 6 Penjelasan Detail Entitas User

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_user	Int	8	Id yang dimiliki pengguna sebagai <i>primary key</i> .
2	Nama_user	Varchar	50	Nama dari pengguna.
3	Alamat_user	Varchar	150	Alamat dari pengguna.
4	Tgl_lahir_user	Date		Tanggal lahir dari pengguna.
5	Telp_user	Varchar	15	Nomor telepon dari pengguna.
6	Foto_user	Varchar	255	URL foto dari pengguna.
7	Email_user	Varchar	50	Email dari pengguna.
8	Pass_user	Varchar	255	Password dari pengguna.
9	Is_admin	Varchar	11	Pembeda pengguna antara <i>user</i> dan <i>admin</i>

Entitas lapor memiliki atribut yang berisikan data terkait laporan keluhan seperti yang ditunjukkan pada Gambar 5. 16.



Gambar 5. 16 Detail Entitas Lapor

Penjelasan lengkap terkait atribut-atribut pada entitas lapor dapat dilihat pada Tabel 5. 7.

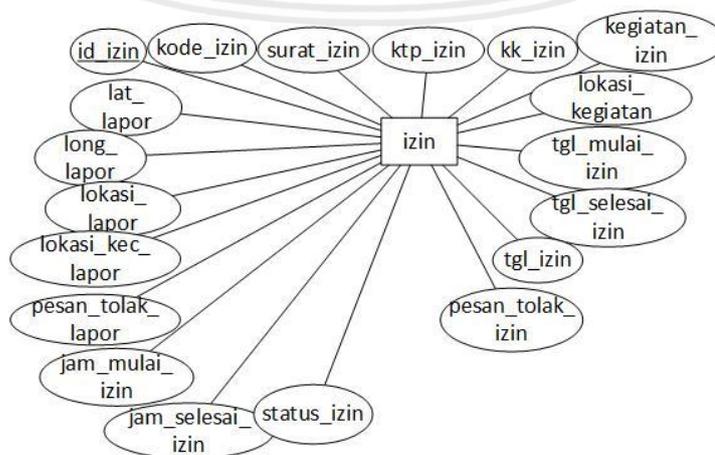
Tabel 5. 7 Penjelasan Detail Entitas Lapor

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_lapor	Int	11	Id dari laporan keluhan sebagai <i>primary key</i> .
2	Kode_lapor	Varchar	255	Kode unik dari laporan keluhan yang ditampilkan ke pengguna.



No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
3	Id_user_lapor	Int	11	Id dari pengguna yang melakukan laporan keluhan sebagai <i>foreign key</i> .
4	Kat_lapor	Varchar	255	Kategori dari laporan keluhan.
5	Foto_lapor	Varchar	150	URL foto dari laporan keluhan.
6	Desk_lapor	Text		Uraian deskripsi dari laporan keluhan.
7	Lat_lapor	Double		Koordinat <i>latitude map</i> dari laporan keluhan.
8	Long_lapor	Double		Koordinat <i>longitude map</i> dari laporan keluhan.
9	Lokasi_lapor	Varchar	255	Alamat lokasi dari laporan keluhan.
10	Lokasi_kec_lapor	Varchar	255	Distrik lokasi dari laporan keluhan.
11	Tgl_lapor	Datetime		Tanggal dan jam dari <i>user</i> ketika melakukan laporan keluhan.
12	Status_lapor	Varchar	255	Status dari laporan keluhan.
13	Pesan_tolak_lapor	Text		Pesan ketika status dari laporan keluhan adalah ditolak.

Entitas izin memiliki atribut yang berisikan data terkait izin penggunaan jalan seperti yang ditunjukkan pada Gambar 5. 17.



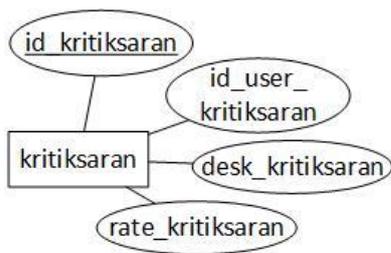
Gambar 5. 17 Detail Entitas Izin

Penjelasan lengkap terkait atribut-atribut pada entitas izin dapat dilihat pada Tabel 5. 8.

Tabel 5. 8 Penjelasan Detail Entitas Izin

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_izin	Int	11	Id dari izin penggunaan jalan sebagai <i>primary key</i> .
2	Id_user_izin	Int	11	Id dari <i>user</i> yang melakukan izin penggunaan jalan sebagai <i>foreign key</i> .
3	Kode_izin	Varchar	255	Kode unik dari izin penggunaan jalan yang ditampilkan ke pengguna.
4	Surat_izin	Varchar	255	URL scan gambar dari surat keterangan dari lurah.
5	Ktp_izin	Varchar	255	URL scan gambar dari kartu tanda penduduk.
6	Kk_izin	Varchar	255	URL scan gambar dari kartu keluarga.
7	Kegiatan_izin	Text		Jenis kegiatan yang dilakukan dari izin penggunaan jalan.
8	Lokasi_izin	Text		Lokasi kegiatan yang dilakukan dari izin penggunaan jalan.
9	Tgl_mulai_izin	Date		Tanggal kegiatan dimulai dari izin penggunaan jalan.
10	Tgl_selesai_izin	Date		Tanggal kegiatan selesai dari izin penggunaan jalan.
11	Jam_mulai_izin	Time		Jam kegiatan dimulai dari izin penggunaan jalan.
12	Jam_selesai_izin	Time		Jam kegiatan selesai dari izin penggunaan jalan.
13	Tgl_izin	Datetime		Tanggal dan jam dari <i>user</i> ketika melakukan izin penggunaan jalan.
14	Status_izin	Varchar	255	Status dari izin penggunaan jalan.
15	Pesan_tolak_izin	Text		Pesan ketika status dari izin penggunaan jalan adalah ditolak.

Entitas kritiksaran memiliki atribut yang berisikan data terkait kritik dan saran seperti yang ditunjukkan pada Gambar 5. 18.



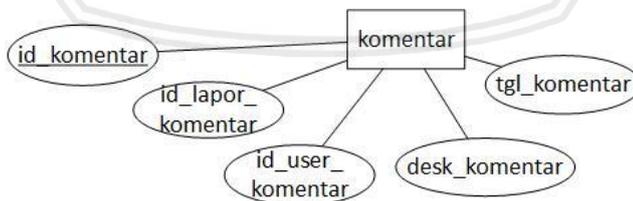
Gambar 5. 18 Detail Entitas Kritik saran

Penjelasan lengkap terkait atribut-atribut pada entitas kritik saran dapat dilihat pada Tabel 5. 9.

Tabel 5. 9 Penjelasan Detail Entitas Kritik saran

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_kritik saran	Int	11	Id dari kritik dan saran sebagai <i>primary key</i> .
2	Id_user_kritik saran	Int	11	Id dari <i>user</i> yang mengirim kritik dan saran sebagai <i>foreign key</i> .
3	Rate_kritik saran	Varchar	15	Tingkat <i>rating</i> dari <i>user</i> ketika mengirim kritik dan saran.
4	Desk_kritik saran	Text		Uraian deskripsi dari kritik dan saran.
5	Tgl_kritik saran	Datetime		Tanggal dan jam ketika <i>user</i> mengirim kritik dan saran.

Entitas komentar memiliki atribut yang berisikan data terkait komentar pada laporan keluhan seperti yang ditunjukkan pada Gambar 5. 19.



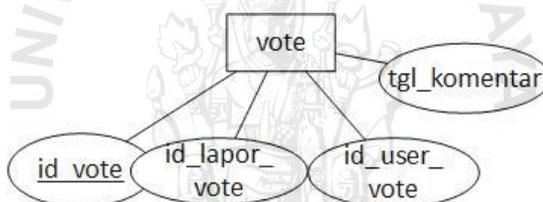
Gambar 5. 19 Detail Entitas Komentar

Penjelasan lengkap terkait atribut-atribut pada entitas komentar dapat dilihat pada Tabel 5. 10.

Tabel 5. 10 Penjelasan Detail Entitas Komentar

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_komentar	Int	11	Id dari komentar sebagai <i>primary key</i> .
2	Id_lapor_komentar	Int	11	Id dari laporan keluhan yang dikomentari sebagai <i>foreign key</i> .
3	Id_user_komentar	Int	11	Id <i>user</i> yang melakukan komentar sebagai <i>foreign key</i> .
4	Desk_komentar	Varchar	255	Uraian deskripsi komentar dari pengguna.
5	Tgl_komentar	Datetime		Tanggal dan jam ketika pengguna mengirim komentar.

Entitas *vote* memiliki atribut yang berisikan data terkait *vote* laporan keluhan seperti yang ditunjukkan pada Gambar 5. 20.



Gambar 5. 20 Detail Entitas *Vote*

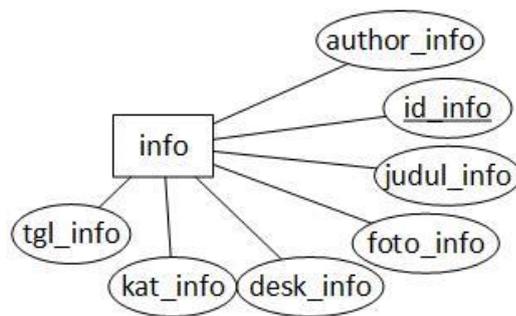
Penjelasan lengkap terkait atribut-atribut pada entitas *vote* dapat dilihat pada Tabel 5. 11.

Tabel 5. 11 Penjelasan Detail Entitas *Vote*

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_vote	Int	11	Id dari <i>vote</i> sebagai <i>primary key</i> .
2	Id_lapor_vote	Int	11	Id dari laporan keluhan yang di- <i>vote user</i> sebagai <i>foreign key</i> .
3	Id_user_vote	Int	11	Id dari <i>user</i> yang melakukan <i>vote</i> sebagai <i>foreign key</i> .
4	Tgl_vote	Datetime		Tgl dan jam ketika <i>user</i> melakukan <i>vote</i> .

Entitas *info* memiliki atribut yang berisikan data terkait informasi pengumuman seperti yang ditunjukkan pada Gambar 5. 21.





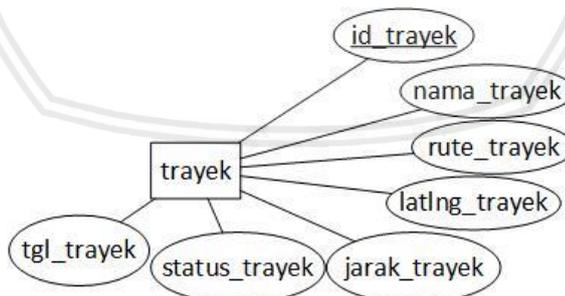
Gambar 5. 21 Detail Entitas Info

Penjelasan lengkap terkait atribut-atribut pada entitas info dapat dilihat pada Tabel 5. 12.

Tabel 5. 12 Penjelasan Detail Entitas Info

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_info	Int	11	Id dari info sebagai <i>primary key</i> .
2	Judul_info	Varchar	255	Judul dari info
3	Foto_info	Varchar	255	URL dari foto info
4	Desk_info	Text		Uraian deskripsi dari info.
5	Kat_info	Varchar	30	Kategori dari info.
6	Tgl_info	Datetime		Tanggal dan jam ketika info dibuat.
7	Author_info	Varchar	100	Penulis dari info.

Entitas trayek memiliki atribut yang berisikan data terkait informasi trayek seperti yang ditunjukkan pada Gambar 5. 22.



Gambar 5. 22 Detail Entitas Trayek

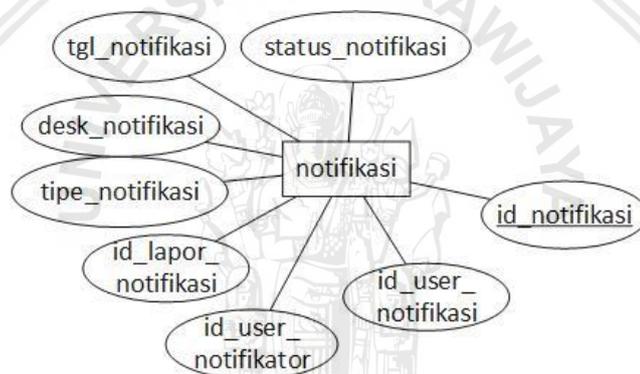
Penjelasan lengkap terkait atribut-atribut pada entitas trayek dapat dilihat pada Tabel 5. 13.



Tabel 5. 13 Penjelasan Detail Entitas Trayek

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_trayek	Int	11	Id dari trayek sebagai <i>primary key</i> .
2	Nama_trayek	Varchar	255	Nama dari trayek.
3	Rute_trayek	Text		Rute yang dilewati oleh trayek.
4	LatIng_trayek	Text		Koordinat <i>latitude</i> dan <i>longitude</i> dari rute trayek.
5	Jarak_trayek	Float		Jarak dari rute trayek.
6	Status_trayek	Varchar	255	Status dari trayek.
7	Tgl_trayek	Date		Tanggal dari trayek dibuat.

Entitas notifikasi memiliki atribut yang berisikan data terkait notifikasi yang didapat pengguna seperti yang ditunjukkan pada Gambar 5. 23.



Gambar 5. 23 Detail Entitas Notifikasi

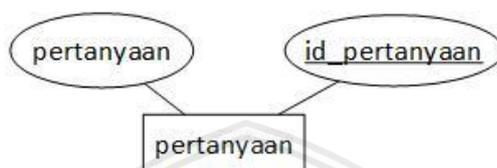
Penjelasan lengkap terkait atribut-atribut pada entitas notifikasi dapat dilihat pada Tabel 5. 14.

Tabel 5. 14 Penjelasan Detail Entitas Notifikasi

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_notifikasi	Int	11	Id dari notifikasi sebagai <i>primary key</i> .
2	Id_user_notifikasi	Int	11	Id dari pengguna yang mendapat notifikasi sebagai <i>foreign key</i> .
3	Id_user_notifikator	Int	11	Id dari pengguna yang memberi notifikasi sebagai <i>foreign key</i> .
4	Tipe_notifikasi	Varchar	255	Tipe dari notifikasi.
5	Desk_nottifikasi	Text		Uraian deskripsi dari notifikasi.

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
6	Tgl_notifikasi	Datetime		Tanggal dan jam ketika pengguna mendapat notifikasi.
7	Status_notifikasi	Varchar	255	Status dari notifikasi.

Entitas pertanyaan memiliki atribut yang berisikan data master terkait pertanyaan untuk melakukan survey seperti yang ditunjukkan pada Gambar 5. 24.



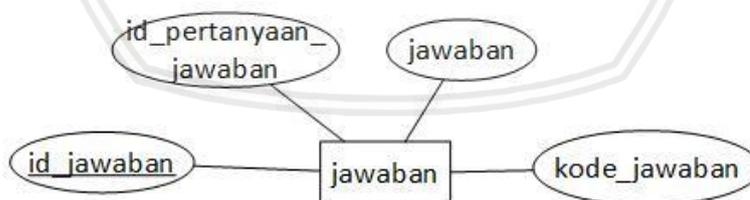
Gambar 5. 24 Detail Entitas Pertanyaan

Penjelasan lengkap terkait atribut-atribut pada entitas pertanyaan dapat dilihat pada Tabel 5. 15.

Tabel 5. 15 Penjelasan Detail Entitas Pertanyaan

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_pertanyaan	Int	11	Id dari master data pertanyaan sebagai <i>primary key</i> .
2	Pertanyaan	Text		Uraian is dari master data pertanyaan.

Entitas jawaban memiliki atribut yang berisikan data master terkait jawaban untuk melakukan survei seperti yang ditunjukkan pada Gambar 5. 25.



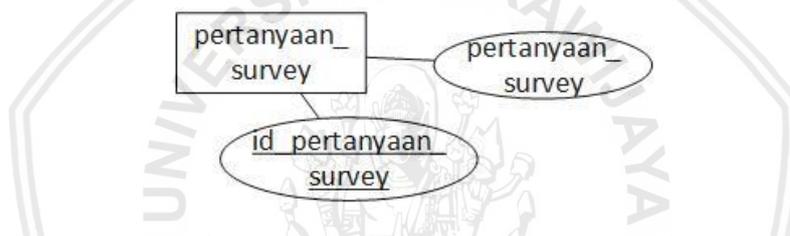
Gambar 5. 25 Detail Entitas Jawaban

Penjelasan lengkap terkait atribut-atribut pada entitas jawaban dapat dilihat pada Tabel 5. 16.

Tabel 5. 16 Penjelasan Detail Entitas Jawaban

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_jawaban	Int	11	Id dari master data jawaban <i>primary key</i> .
2	Id_pertanyaan_jawaban	Int	11	Id dari master data pertanyaan sebagai <i>foreign key</i> .
3	Jawaban	Text		Uraian isi master data jawaban.
4	Kode_jawaban	Varchar	7	Kode inisial dari master data jawaban.

Entitas pertanyaan\_survei memiliki atribut yang berisikan data terkait pertanyaan yang diberikan kepada *user* untuk menjawab survei ditunjukkan pada Gambar 5. 26.



Gambar 5. 26 Detail Entitas Pertanyaan\_Survei

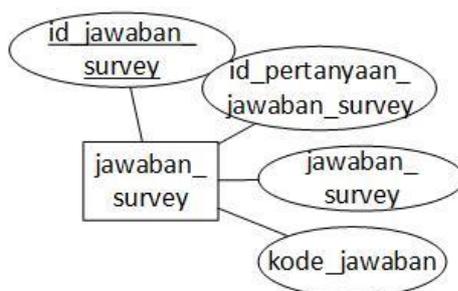
Penjelasan lengkap terkait atribut-atribut pada entitas pertanyaan\_survei dapat dilihat pada Tabel 5. 17.

Tabel 5. 17 Penjelasan Detail Entitas Pertanyaan\_Survei

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_pertanyaan_survei	Int	11	Id dari pertanyaan survei sebagai <i>primary key</i> .
2	Pertanyaan	Text		Uraian dari pertanyaan survei

Entitas jawaban\_survei memiliki atribut yang berisikan data terkait jawaban yang diberikan kepada *user* untuk menjawab survei ditunjukkan pada Gambar 5. 27.





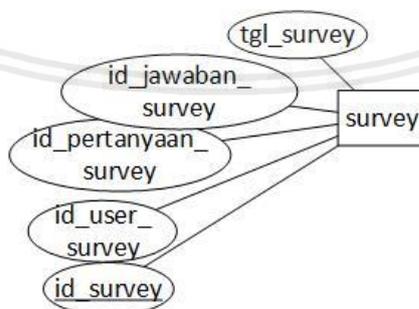
Gambar 5. 27 Detail Entitas Jawaban\_Survei

Penjelasan lengkap terkait atribut-atribut pada entitas jawaban\_surveyi dapat dilihat pada Tabel 5. 18.

Tabel 5. 18 Penjelasan Detail Entitas Jawaban\_Survei

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_jawaban_surveyi	Int	11	Id dari jawaban survei sebagai <i>primary key</i> .
2	Id_pertanyaan_jawaban_surveyi	Int	11	Id dari pertanyaan survei sebagai <i>foreign key</i> .
3	Jawaban_surveyi	Text		Uraian isi jawaban survei.
4	Kode_jawaban	Varchar	7	Kode inisial dari jawaban survei.

Entitas survei memiliki atribut yang berisikan data terkait survei kepuasan masyarakat seperti yang ditunjukkan pada Gambar 5. 28.



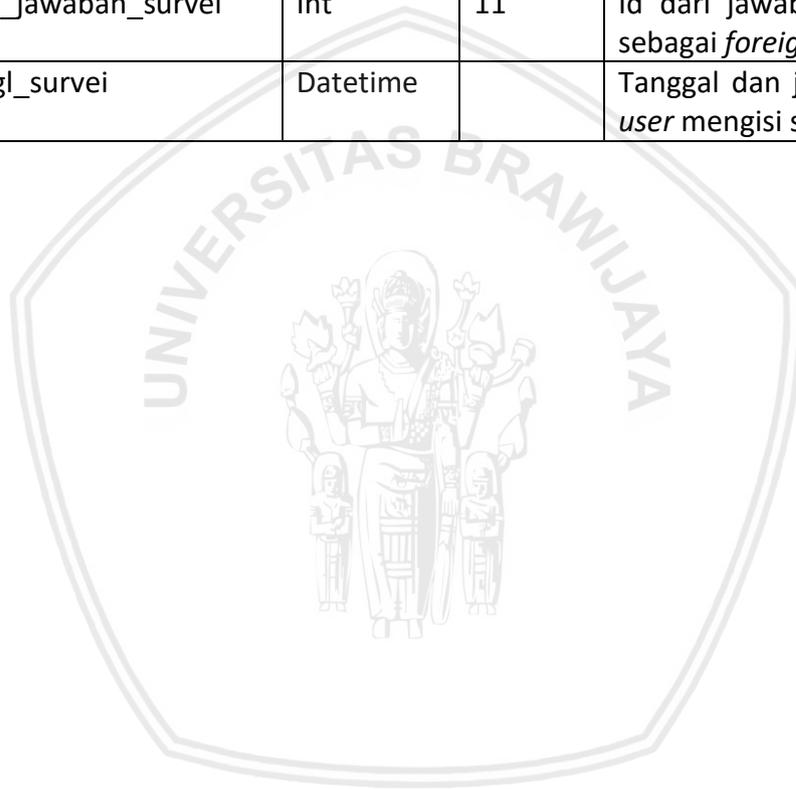
Gambar 5. 28 Detail Entitas Survei

Penjelasan lengkap terkait atribut-atribut pada entitas survei dapat dilihat pada Tabel 5. 19.



Tabel 5. 19 Penjelasan Detail Entitas Survei

No.	Nama Kolom	Tipe Data	Ukuran	Keterangan
1	Id_survei	Int	11	Id dari survei sebagai <i>primary key</i> .
2	Id_user_survei	Int	11	Id dari <i>user</i> yang mengisi survei sebagai <i>foreign key</i> .
3	Id_pertanyaan_survei	Int	11	Id dari pertanyaan survei sebagai <i>foreign key</i> .
4	Id_jawaban_survei	Int	11	Id dari jawaban survei sebagai <i>foreign key</i> .
5	Tgl_survei	Datetime		Tanggal dan jam ketika <i>user</i> mengisi survei.



## 5.2 Implementasi Sistem

Setelah proses perancangan selesai selanjutnya adalah proses implementasi sistem. Pada proses implementasi sistem akan dibahas mengenai hasil implementasi aplikasi pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile*. Pada bagian implementasi akan dibahas mengenai spesifikasi sistem, implementasi kode program, implementasi antarmuka.

### 5.2.1 Spesifikasi Sistem

Tahap implementasi sistem membutuhkan perangkat yang bekerja baik perangkat keras maupun perangkat lunak. Penjelasan mengenai spesifikasi sistem yang digunakan akan dibagi menjadi dua bagian yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

#### 1. Spesifikasi Perangkat Keras

Proses pengembangan aplikasi pelaporan keluhan masyarakat pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* dilakukan dengan menggunakan dua perangkat keras. Perangkat keras pertama adalah *laptop* untuk mengimplementasi perancangan yang sebelumnya dibuat. Perangkat keras kedua adalah sebuah *smartphone* yang digunakan untuk proses *debug* aplikasi dengan spesifikasi perangkat keras seperti yang ditunjukkan pada Tabel 5. 20 dan Tabel 5. 21.

Tabel 5. 20 Spesifikasi Perangkat Keras *Laptop*

Nama Komponen	Spesifikasi
<i>System Model</i>	Lenovo G40-45 80E1
<i>Processor</i>	AMD A8 6410 APU with AMD Radeon R5 Graphics 2.00 GHz
<i>Memory</i>	8192 MB RAM
<i>Storage</i>	Galax 120GB SSD
<i>Display</i>	14" 1366x768 (32bit)(60Hz)

Tabel 5. 21 Spesifikasi Perangkat Keras *Smartphone*

Nama Komponen	Spesifikasi
<i>System Model</i>	Honor 8x
<i>Processor</i>	Hisilicon Kirin 710 (12 nm) Octa-core (4x2.2 GHz Cortex-A73 & 4x1.7 GHz Cortex-A53)
<i>Memory</i>	3713 MB RAM
<i>Storage</i>	128 GB
<i>Display</i>	6.5 inches LTPS IPS LCD capacitive touchscreen 1080 x 2340 pixels, 19.5:9 ratio

## 2. Spesifikasi Perangkat Lunak

Proses dari pengembangan aplikasi pelaporan keluhan pada Dinas Perhubungan Kota Mojokerto berbasis *mobile* dilakukan dengan menggunakan spesifikasi perangkat lunak seperti yang ditunjukkan pada Tabel 5. 22.

Tabel 5. 22 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 Pro 64-bit Build 17134
<i>Programming Language</i>	Javascript
<i>Programming Tools</i>	Node v10.15.3
<i>Programming Environment</i>	Node.Js, Vue.Js
<i>Code Editor</i>	Visual Studio Code v1.33.1

### 5.2.2 Implementasi Kode Program

Implementasi kode program adalah tahap implementasi algoritme dari proses-proses yang telah dijelaskan sebelumnya menjadi sebuah kode program. Pada implementasi kode program hanya akan diperlihatkan beberapa algoritme yang menjadi pokok dalam sistem, diantaranya yaitu tambah laporan keluhan, lihat info trayek, izin penggunaan jalan, survei kepuasan masyarakat dan kritik & saran.

#### 1. Kode Program Tambah Laporan Keluhan

Kode program Tambah Laporan Keluhan berfungsi untuk menambahkan laporan keluhan kepada Dinas Perhubungan Kota Mojokerto. Kode program ini mengambil data dari input user yang salah satunya adalah input data foto. Kemudian foto akan dilakukan pengecekan apakah telah memiliki data GPS pada metadata EXIF (*Exchangeable Image File*) yang merupakan sebuah data yang disimpan pada sebuah gambar yang disimpan oleh kamera smartphone. Kode program ini menggunakan metode *geotagging* dengan memanfaatkan package bernama *piexif* untuk menambahkan lokasi berupa *latitude* dan *longitude* pada *metadata* foto yang yang tidak memiliki *metadata* GPS pada EXIF. Sedangkan untuk foto yang telah memiliki *metadata* GPS akan langsung di cek pada *geocoder* untuk mendapat lokasi alamat menggunakan *reverse geocode*. Kode program Tambah Laporan Keluhan ditunjukkan pada Tabel 5. 23.

Tabel 5. 23 Source Code Tambah Laporan Keluhan

No.	Source Code
1	<code>const express = require('express');</code>
2	<code>const router = express.Router();</code>
3	<code>const bodyParser = require('body-parser');</code>
4	<code>const sharp = require('sharp');</code>
5	<code>const piexif = require("piexifjs");</code>
6	<code>const fs = require("fs");</code>
7	<code>const geocoder = require('../helper/geocoder');</code>
8	<code>const upload = require('../helper/upload-image');</code>
9	<code>const response = require('../config/res');</code>
10	<code>const db = require('../config/db');</code>

```
11 const verifyToken = require('../helper/verify-token');
12 const customID = require('../helper/custom-id');
13 router.use(bodyParser.urlencoded({ limit: "50mb", extended:
14 true, parameterLimit: 50000 }));
15 router.use(bodyParser.json({ limit: "50mb" }));
16
17 const Convert = (degrees, minutes, seconds, direction) => {
18   let dd = degrees + (minutes/60) + (seconds/3600);
19
20   if (direction == "S" || direction == "W") {
21     dd = dd * -1;
22   }
23
24   return dd;
25 }
26
27 router.post('/geocode', verifyToken, (req, res) => {
28   try {
29     let arr = [];
30     if(req.body.GPSLatitude){
31       // Calculate latitude decimal
32       let latDegree = req.body.GPSLatitude[0];
33       let latMinute = req.body.GPSLatitude[1];
34       let latSecond = req.body.GPSLatitude[2];
35       let latDirection = req.body.GPSLatitudeRef;
36
37       let latFinal = Convert(latDegree, latMinute,
38 latSecond, latDirection);
39
40       // Calculate longitude decimal
41       let lonDegree = req.body.GPSLongitude[0];
42       let lonMinute = req.body.GPSLongitude[1];
43       let lonSecond = req.body.GPSLongitude[2];
44       let lonDirection = req.body.GPSLongitudeRef;
45
46       let lonFinal = Convert(lonDegree, lonMinute,
47 lonSecond, lonDirection);
48
49       geocoder.reverse({lat: latFinal, lon: lonFinal},
50 function(err, data) {
51   if (err) {
52     response.fail(err, res);
53   } else {
54     arr.push({
55       originLat: latFinal,
56       originLng: lonFinal,
57       data: data[data.length - 1]
58     });
59     console.log(arr);
60
61     response.ok(arr, res)
62   }
63 })
64 } else if (req.body.lat) {
65   let lat = req.body.lat;
66   let lng = req.body.lng;
67
```

```
68     geocoder.reverse({lat: lat, lon: lng}, function(err,
69 data) {
70     if (err) {
71         response.fail(err, res);
72     } else {
73         arr.push({
74             originLat: lat,
75             originLng: lng,
76             data: data[data.length - 1]
77         });
78         console.log(arr);
79         response.ok(arr, res)
80     }
81 });
82 } else {
83     res.status(204).json({message: 'tidak ada exif
84 data'});
85 }
86
87 } catch (error) {
88     res.status(500).json({message: error.message});
89 }
90 });
91
92 router.post('/', verifyToken, upload.single('fotoLapor'),
93 async (req, res) => {
94     try {
95         if (req.body.status == 'true') {
96             let file = fs.readFileSync(req.file.path);
97             let data = file.toString("binary");
98             let gps = {};
99             let lat = req.body.lat;
100            let lng = req.body.lng;
101
102            gps[piexif.GPSIFD.GPSLatitudeRef] = lat < 0 ? 'S' :
103 'N';
104            gps[piexif.GPSIFD.GPSLatitude] =
105 piexif.GPSHelper.degToDmsRational(lat);
106            gps[piexif.GPSIFD.GPSLongitudeRef] = lng < 0 ? 'W' :
107 'E';
108            gps[piexif.GPSIFD.GPSLongitude] =
109 piexif.GPSHelper.degToDmsRational(lng);
110
111            let exifObj = {"GPS":gps};
112            let exifbytes = piexif.dump(exifObj);
113
114            let newData = piexif.insert(exifbytes, data);
115            let newfile = new Buffer.from(newData, "binary");
116
117            fs.writeFileSync(req.file.path, newfile);
118
119        }
120
121        let path = req.file.path.replace(/\\/g, "/");
122
123        let resize = await sharp('./' +
124 path).withMetadata().toBuffer();
```

```

125     await
126     sharp(resize).withMetadata().resize(720).toFile('./' +
127     path);
128
129     let kode = await customID('lapor', 'kode_lapor', 'LPR-
130     ', 4);
131
132     let userid = req.user.userId;
133     let kat = req.body.kat;
134     let fotoLapor = '/' + path;
135     let desk = req.body.desk;
136     let lat = req.body.lat;
137     let long = req.body.lng;
138     let lokasi = req.body.lokasi;
139     let district = req.body.district;
140     let tgl = new Date();
141     let status = "Menunggu";
142
143     await db.query('INSERT INTO lapor (kode_lapor,
144     id_user_lapor, kat_lapor, foto_lapor, desk_lapor,
145     lat_lapor, long_lapor, lokasi_lapor, lokasi_kec_lapor,
146     tgl_lapor, status_lapor) values (?,?,?,?,?,?,?,?,?,?)',
147     [kode, userid, kat, fotoLapor, desk, lat, long,
148     lokasi, district, tgl, status ]);
149
150     response.ok("Berhasil menambahkan lapor!",
151     res);
152   } catch (error) {
153     console.log(error);
154     res.status(500).json({message: error.message});
155   }
156 });
157
158 module.exports = router;
159
160

```

## 2. Kode Program Lihat Info Trayek

Kode Program Lihat Info Trayek berfungsi untuk menampilkan informasi terkait trayek, angkutan umum dan rutenya pada masyarakat. Kode program ini mengembalikan data hasil dari *query* yang akan digunakan user untuk berinteraksi. Kode proram Lihat Info Trayek ditunjukkan pada Tabel 5. 24.

Tabel 5. 24 Source Code Lihat Info Trayek

No.	Source Code
1	const express = require('express');
2	const router = express.Router();
3	const bodyParser = require('body-parser');
4	const response = require('../config/res');
5	const db = require('../config/db');
6	const verifyToken = require('../helper/verify-token');
7	
8	router.use(bodyParser.urlencoded({ limit: "50mb", extended:
9	true, parameterLimit: 50000 }));
10	

```

11 router.get('/trayek', verifyToken, async (req, res) => {
12     try {
13         let result = await db.query('SELECT * FROM trayek
14 ORDER BY id_trayek DESC');
15         response.ok(result, res);
16     } catch (error) {
17         console.log(error.message);
18         res.status(500).json({message: error.message});
19     }
20 });
21
22 module.exports = router;
23

```

### 3. Kode Program Izin Penggunaan Jalan

Kode Program Izin Penggunaan Jalan berfungsi untuk melakukan izin penggunaan jalan melalui Dinas Perhubungan Kota Mojokerto dengan mengisi form izin penggunaan jalan yaitu jenis kegiatan, lokasi kegiatan jam dan tanggal kegiatan serta mengunggah berkas-berkas yang diminta yaitu scan surat rekomendasi dari lurah setempat, scan kartu tanda penduduk (KTP) dan kartu keluarga (KK). Kode program ini menggunakan package multer yang disimpan pada variable upload untuk mengunggah gambar pada server dan package sharp untuk yang disimpan pada variable sharp untuk melakukan proses *resize* gambar untuk memperkecil ukuran file. Kode Program Izin Penggunaan Jalan ditunjukkan pada Tabel 5. 25.

Tabel 5. 25 Source Code Izin Penggunaan Jalan

No.	Source Code
1	const express = require('express');
2	const router = express.Router();
3	const bodyParser = require('body-parser');
4	const sharp = require('sharp');
5	const upload = require('../helper/upload-image');
6	const response = require('../config/res');
7	const db = require('../config/db');
8	const verifyToken = require('../helper/verify-token');
9	const customID = require('../helper/custom-id');
10	
11	router.use(bodyParser.urlencoded({ limit: "50mb", extended:
12	true, parameterLimit: 50000 }));
13	router.use(bodyParser.json({ limit: "50mb" }));
14	
15	
16	router.post('/', verifyToken, upload.fields([name:
17	'fotoSurat'}, {name: 'fotoKTP'}, {name: 'fotoKK'}]), async
18	(req, res) => {
19	try {
20	let pathSurat =
21	req.files.fotoSurat[0].path.replace(/\\/g, "/");
22	let pathKTP =
23	req.files.fotoKTP[0].path.replace(/\\/g, "/");
24	let pathKK =
25	req.files.fotoKK[0].path.replace(/\\/g, "/");

```

26
27     let    resizeSurat    =    await    sharp('./' +
28 pathSurat).withMetadata().toBuffer();
29     await
30 sharp(resizeSurat).withMetadata().resize(1080).toFile('./' +
31 + pathSurat);
32
33     let    resizeKTP    =    await    sharp('./' +
34 pathKTP).withMetadata().toBuffer();
35     await
36 sharp(resizeKTP).withMetadata().resize(1080).toFile('./' +
37 + pathKTP);
38
39     let    resizeKK    =    await    sharp('./' +
40 pathKK).withMetadata().toBuffer();
41     await
42 sharp(resizeKK).withMetadata().resize(1080).toFile('./' +
43 + pathKK);
44
45     let    kode    =    await    customID('izin', 'kode_izin',
46 'IZN-', 4);
47
48     let    iduser    =    req.user.userId;
49     let    surat    =    '/' + pathSurat;
50     let    ktp    =    '/' + pathKTP;
51     let    kk    =    '/' + pathKK;
52     let    desk    =    req.body.desk;
53     let    loc    =    req.body.loc;
54     let    tglmulai    =    req.body.tglmulai;
55     let    tglend    =    req.body.tglend;
56     let    jammulai    =    req.body.jammulai;
57     let    jamend    =    req.body.jamend;
58     let    tgl    =    new Date();
59     let    status    =    'Menunggu';
60
61     await    db.query('INSERT INTO izin (id_user_izin,
62 kode_izin, surat_izin, ktp_izin, kk_izin, kegiatan_izin,
63 lokasi_izin, tgl_mulai_izin, tgl_selesai_izin,
64 jam_mulai_izin, jam_selesai_izin, tgl_izin, status_izin)
65 values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)',
66 [ iduser, kode, surat, ktp, kk, desk, loc, tglmulai,
67 tglend, jammulai, jamend, tgl, status ]);
68
69     response.ok("Berhasil menambahkan izin!", res);
70 } catch (error) {
71     console.log(error.message);
72     res.status(500).json({message: error.message});
73 }
74 });
75
module.exports = router;

```

#### 4. Kode Program Survei Kepuasan Masyarakat

Kode program Survei Kepuasan Masyarakat digunakan untuk masyarakat untuk mengisi survei terkait kepuasan masyarakat terhadap fasilitas terkait perhubungan dan layanan pada Dinas Perhubungan Kota Mojokerto. Kode



```

51     try {
52         let query = `INSERT INTO survey (id_user_survey,
53 id_pertanyaan_survey, id_jawaban_survey, tgl_survey)
54             VALUES (?, ?, ?, ?)`;
55         let result = await db.query(query, [req.user.userId,
56 req.body.soal, req.body.jwb, new Date()]);
57         response.ok(result, res);
58     } catch (error) {
59         console.log(error.message);
60         res.status(500).json({message: error.message});
61     }
62 });
63
64 module.exports = router;
65

```

## 5. Kode Program Kritik dan Saran

Kode Program Kritik dan Saran berfungsi bagi masyarakat untuk memberikan *feedback* atau kritik dan saran pada Dinas Perhubungan Kota Mojokerto agar meningkatkan kualitas layanan dan fasilitasnya. Kode program ini hanya digunakan untuk user mengirim data ke server. Kode Program Kritik dan Saran ditunjukkan pada Tabel 5. 27.

Tabel 5. 27 *Source Code* Kritik dan Saran

No.	Source Code
1	const express = require('express');
2	const router = express.Router();
3	const bodyParser = require('body-parser');
4	const response = require('../config/res');
5	const db = require('../config/db');
6	const verifyToken = require('../helper/verify-token');
7	
8	router.use(bodyParser.urlencoded({ extended: true }));
9	router.use(bodyParser.json());
10	
11	router.post('/', verifyToken, async (req, res) => {
12	try {
13	let user = req.user.userId;
14	let rate = req.body.rate;
15	let desk = req.body.desk;
16	let tgl = new Date();
17	
18	console.log(user, rate, desk, tgl);
19	
20	let result = await db.query('INSERT INTO kritiksaran
21	(id_user_kritiksaran, rate_kritiksaran, desk_kritiksaran,
22	tgl_kritiksaran) values (?, ?, ?, ?)',
23	[user, rate, desk, tgl]);
24	response.ok(result, res);
25	} catch (error) {
26	console.log(error.message);
27	}
28	});
29	module.exports = router;

### 5.2.3 Implementasi Antarmuka

Implementasi antarmuka adalah proses penerapan antarmuka atau *user interface* yang didapatkan dari perancangan antarmuka yang telah dirancang sebelumnya. Pada perancangan antarmuka ini hanya akan diperlihatkan beberapa antarmuka yang menjadi pokok dalam sistem, diantaranya yaitu tambah laporan keluhan, lihat info trayek, izin penggunaan jalan, survey kepuasan masyarakat dan kritik & saran.

#### 1. Implementasi Antarmuka Tambah Laporan Keluhan

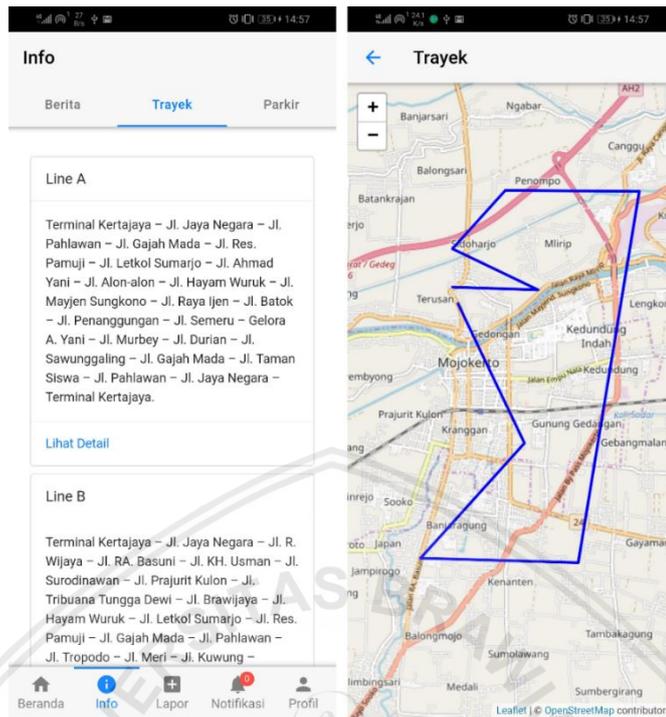
Implementasi antarmuka Tambah Laporan keluhan adalah halaman yang digunakan oleh user untuk menambahkan laporan keluhan. Pada halaman ini terdapat form yang harus diisi oleh user yaitu, foto laporan keluhan yang bisa diambil langsung dengan kamera *smartphone* atau mencari foto pada galeri *smartphone*, kategori laporan keluhan, titik lokasi laporan yang digunakan ketika foto tidak memiliki data GPS pada metadata EXIF, dan deskripsi dari laporan yang akan dikirim. Hasil dari implementasi antarmuka Tambah Laporan Keluhan ditunjukkan pada Gambar 5. 29.



Gambar 5. 29 Implementasi Antarmuka Tambah Laporan Keluhan

#### 2. Implementasi Antarmuka Lihat Info Trayek

Implementasi antarmuka Lihat Info Trayek adalah halaman yang digunakan user untuk melihat informasi terkait trayek, rute dan angkutan umum yang ada di kawasan Kota Mojokerto. Pada halaman ini terdapat list dari trayek yang ada dan rute yang dilalui oleh angkutan umum di Kota Mojokerto yang ketika user memilih salah satu trayek akan ditampilkan detail rute dengan bantuan *map*. Hasil dari implementasi antarmuka Lihat info Trayek ditunjukkan pada Gambar 5. 30.



Gambar 5. 30 Implementasi Antarmuka Lihat Info Trayek

### 3. Implementasi Antarmuka Izin Penggunaan Jalan

Implementasi antarmuka Izin Penggunaan Jalan adalah halaman yang digunakan user untuk melakukan izin penggunaan jalan pada Dinas Perhubungan Kota Mojokerto. Pada halaman ini menyediakan form yang harus diisi oleh user yaitu jenis kegiatan, lokasi kegiatan, tanggal dan jam kegiatan, dan lampiran yang berisi unggah file foto terkait surat rekomendasi dari kelurahan, foto ktp dan foto kk. Hasil dari implementasi antarmuka Izin Penggunaan Jalan ditunjukkan pada Gambar 5. 31.

Jenis Kegiatan  
-- Pilih --

Lokasi Kegiatan  
Misal: Jl. Jawa

Tanggal Mulai  
18/06/2019

Tanggal Selesai  
18/06/2019

Jam Mulai  
09:52

Jam Selesai  
09:52

Lampiran

Surat Pengantar dari Kelurahan

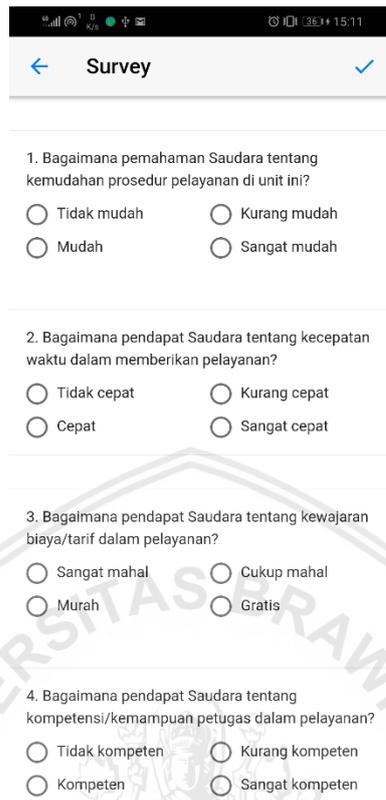
Kartu Tanda Penduduk

Kartu Keluarga

Gambar 5. 31 Implementasi Antarmuka Izin Penggunaan Jalan

#### 4. Implementasi Antarmuka Survey Kepuasan Masyarakat

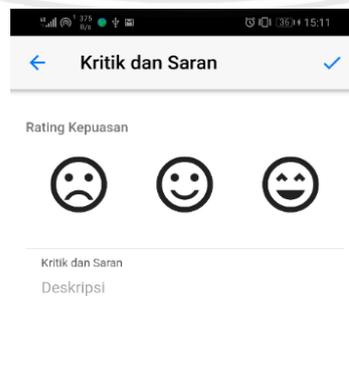
Implementasi antarmuka Survei Kepuasan Masyarakat adalah halaman yang digunakan user mengisi survei terkait kepuasan masyarakat terhadap fasilitas terkait perhubungan dan layanan pada Dinas Perhubungan Kota Mojokerto. Pada halaman ini menampilkan daftar pertanyaan dan pilihan jawaban yang tersedia. User harus menjawab semua pertanyaan yang ada sebelum mengirimnya ke server. Hasil dari implementasi Survei Kepuasan Masyarakat ditunjukkan pada Gambar 5. 32.



Gambar 5. 32 Implementasi Antarmuka Survei Kepuasan Masyarakat

### 5. Implementasi Antarmuka Kritik dan Saran

Implementasi antarmuka Kritik dan Saran adalah haaman yang digunakan user untuk melakukan kritik dan saran terkait perhubungan dan layanan pada Dinas Perhubungan Kota Mojokerto. Terdapat pilihan mengenai kepuasan user yang di representasikan dengan tiga gambar emoji yang harus dipilih dan satu input untuk menuliskan kritik dan saran dari user untuk Dinas Perhubungan Kota Mojokerto. Hasil dari implementasi antarmuka Kritik dan Saran ditunjukkan pada Gambar 5. 33.

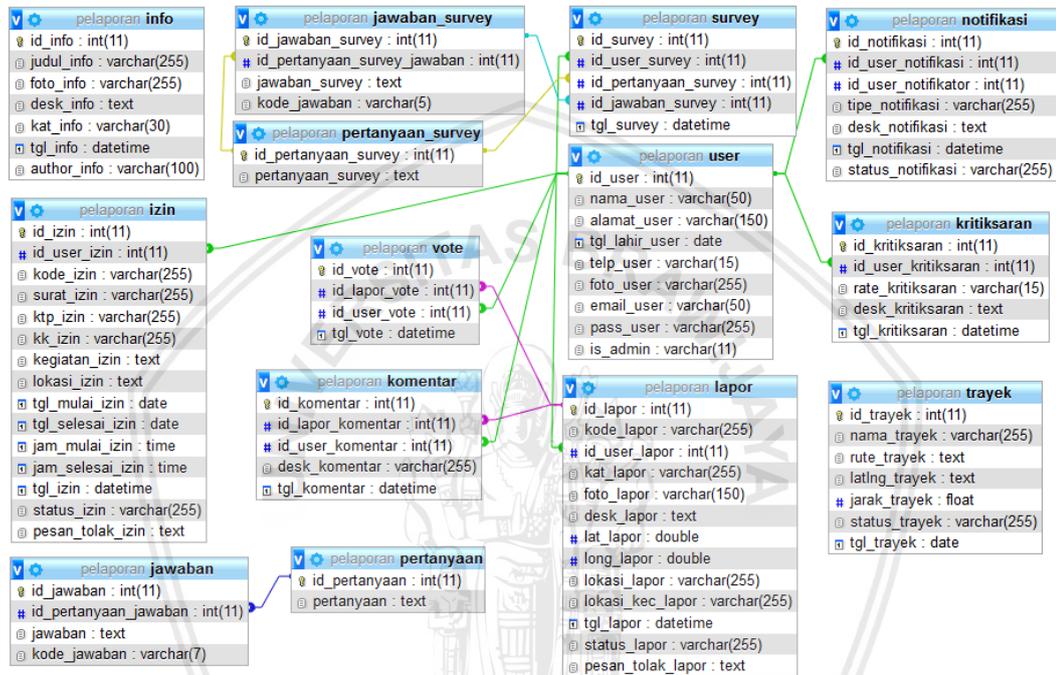


Gambar 5. 33 Implementasi Antarmuka Kritik dan Saran



## 5.2.4 Implementasi Basis Data

Implementasi dari basis data diimplementasi menggunakan sistem manajemen basis data MySQL. Seluruh tabel yang diimplementasi mengacu pada perancangan basis data yang telah dirancang sebelumnya. Dari proses perancangan tersebut ditemukan empat belas tabel yaitu user, lapor, izin, kritiksaran, komentar, vote, info, trayek, notifikasi, pertanyaan, jawaban, pertanyaan\_survei, jawaban\_survei dan survei. Hasil dari implementasi basis data ditunjukkan pada Gambar 5. 34.



Gambar 5. 34 Implementasi Basis Data Menggunakan MySQL

## BAB VI PENGUJIAN

### 6.1 Pengujian

Pengujian adalah sebuah tahapan untuk memastikan apakah sistem yang telah dikembangkan telah sesuai dengan kebutuhan. Pada tahap ini terdapat dua pengujian yang dilakukan yaitu pengujian fungsional menggunakan *black box* dan *white box* dan pengujian non fungsional menggunakan pengujian *compatibility*. Pengujian *black box* yang dilakukan menggunakan pengujian validasi sedangkan pengujian *white box* yang dilakukan menggunakan pengujian unit dan pengujian integrasi.

#### 6.1.1 Pengujian Unit

Pengujian unit bertujuan untuk melakukan uji komponen, objek, dan klas dari hasil perancangan. Pada tahap ini dilakukan pengujian untuk menyesuaikan hasil algoritme yang telah diimplementasikan dengan hasil analisis kebutuhan sebelumnya. Dalam melakukan pengujian unit dilakukan dengan metode *basis path testing*. Pada tahap pengujian ini hanya akan dipaparkan beberapa pengujian yaitu *method* tambahLaporan() pada *class* LaporController, createLapor() pada *class* LaporModel, tambahIzin() pada *class* IzinController dan createIzin() dari *class* IzinModel.

##### 1. Pengujian Unit *Method* tambahLaporan()

Pengujian unit *method* tambahLaporan() dari *class* LaporController ini dilakukan untuk merepresentasikan aliran kontrol saat user menambahkan laporan keluhan kepada Dinas Perhubungan Kota Mojokerto menggunakan algoritme yang ditunjukkan pada Tabel 6. 1.

**Tabel 6. 1 Algoritme Pengujian Unit *Method* tambahLaporan()**

tambahLaporan ()	
No	Pseudocode
1	Mulai <span style="float: right;">← (1)</span>
2	
3	Inisialisasi userInfo
4	Inisialisasi verifyToken
5	Insialisasi geocoder
6	Inisialisasi database
7	
8	Panggil verifyToken(userInfo) ← (3)
9	
10	If (verifyToken == false) { ← (4)
11	Return false ← (5)
12	} else {
13	Return true
14	}
15	Endif (6)
16	
17	Foto = file foto
18	Data = ambil input foto, kategori, deskripsi, lokasi latlng

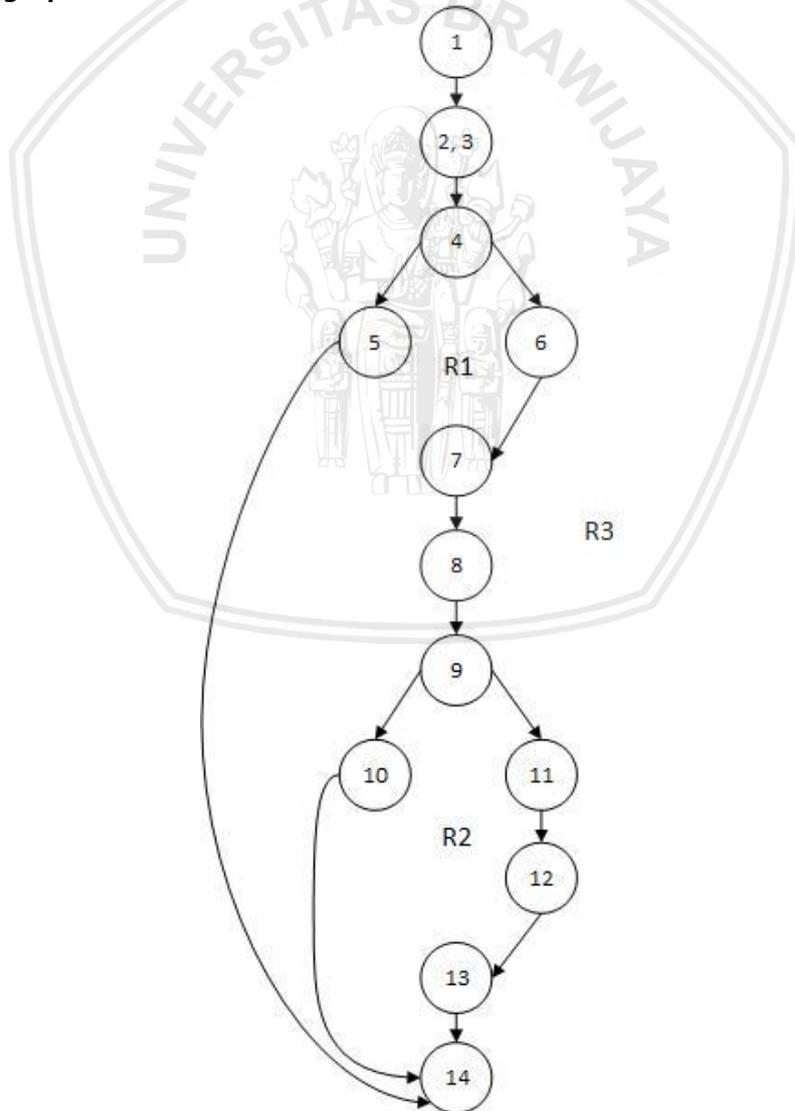
```

19 Panggil geocoder(latlng)
20 If (metadata exif foto == true) {
21   Return lokasi foto
22 } else {
23   Lokasi = ambil latlng dari input
24   Panggil geocoder(lokalasi)
25   Return lokasi foto
26 }
27 Endif
28
29 Selesai

```

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6. 1 maka akan dilakukan pengujian menggunakan teknik *basis path* yang akan menjelaskan *flowgraph* yang ditunjukkan pada Gambar 6. 1, *cyclomatic complexity*, dan jalur independen.

a) **Flowgraph**



Gambar 6. 1 *Flowgraph Method tambahLaporan()*

**b) Cyclomatic Complexity**

- $V(G) = R = 3$
- $V(G) = E - N + 2 = 15 - 14 + 2 = 3$
- $V(G) = P + 1 = 2 + 1 = 3$

**c) Jalur Independen**

- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 14
- Jalur 2 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 14
- Jalur 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 11 – 12 – 13 – 14

Hasil dari *basis path* selanjutnya digunakan untuk menerapkan kasus uji dan untuk mengetahui kevalidannya. Kasus uji dinyatakan valid ketika hasil dan hasil yang diharapkan keduanya sama. Hasil pengujian kasus uji tambah laporan keluhan ditunjukkan pada Tabel 6. 2.

Tabel 6. 2 Hasil Pengujian Unit Tambah Laporan Keluhan

Jalur	Prosedur	Hasil yang Diharapkan	Hasil	Status
1	<i>Class Test</i> memanggil <i>method verifyToken</i> dan mengembalikan nilai <i>false</i> karena <i>token</i> milik <i>user</i> tidak valid.	Sistem akan mengeluarkan user dari aplikasi dan menampilkan halaman login.	Sistem akan mengeluarkan user dari aplikasi dan menampilkan halaman login.	Valid
2	<i>Class Test</i> memanggil <i>method verifyToken</i> dan mengembalikan nilai <i>true</i> karena <i>token</i> milik <i>user</i> valid lalu memanggil <i>method geocoder</i> menggunakan parameter file foto dan mengembalikan nilai <i>true</i> karena data lokasi ditemukan pada	Sistem akan menampilkan foto yang dipilih dan data lokasi yang didapat dari metadata foto berupa koordinat pada <i>map</i> dan berupa data alamat pada label lokasi.	Sistem akan menampilkan foto yang dipilih dan data lokasi yang didapat dari metadata foto berupa koordinat pada <i>map</i> dan berupa data alamat pada label lokasi.	Valid

	metadata file foto.			
3	<p><i>Class Test</i> memanggil <i>method verifyToken</i> dan mengembalikan nilai <i>true</i> karena <i>token</i> milik <i>user</i> valid lalu memanggil <i>method geocoder</i> menggunakan parameter file foto dan mengembalikan nilai <i>false</i> karena data lokasi tidak ditemukan di metadata file foto.</p>	<p>Sistem akan menampilkan foto yang dipilih dan meminta user untuk memilih titik lokasi laporan keluhan melalui bantuan map dan menyimpan data lokasi pada metadata file foto.</p>	<p>Sistem akan menampilkan foto yang dipilih dan meminta user untuk memilih titik lokasi laporan keluhan melalui bantuan map dan menyimpan data lokasi pada metadata file foto.</p>	Valid

## 2. Pengujian Unit *Method createLapor()*

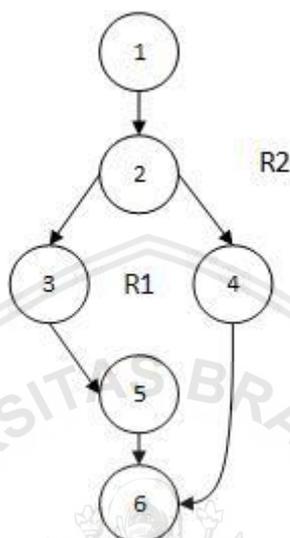
Pengujian unit *method createLapor()* dari *class LaporModel* ini dilakukan untuk merepresentasikan aliran kontrol saat user menyimpan laporan keluhan pada *database* Dinas Perhubungan Kota Mojokerto menggunakan algoritme yang ditunjukkan pada Tabel 6. 3.

Tabel 6. 3 Algoritme Pengujian Unit *Method createLapor()*

createLapor ()	
No	Pseudocode
1	Mulai
2	
3	Data = data input dari controller
4	
5	If (data == lengkap) {
6	Retrun data
7	Return true
8	} else {
9	data tidak lengkap
10	Return false
11	}
12	Endif
13	
14	Simpan data laporan ke database
	Selesai

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6. 3 maka akan dilakukan pengujian menggunakan teknik *basis path* yang akan menjelaskan *flowgraph* yang ditunjukkan pada Gambar 6.2 Gambar 6. 1, *cyclomatic complexity*, dan jalur independen.

a) **Flowgraph**



Gambar 6. 2 *Flowgraph Method createLapor()*

b) **Cyclomatic Complexity**

- $V(G) = R = 2$
- $V(G) = E - N + 2 = 6 - 6 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

c) **Jalur Independen**

- Jalur 1 = 1 – 2 – 4 – 6
- Jalur 2 = 1 – 2 – 3 – 5 – 6

Hasil dari *basis path* selanjutnya digunakan untuk menerapkan kasus uji dan untuk mengetahui kevalidannya. Kasus uji dinyatakan valid ketika hasil dan hasil yang diharapkan keduanya sama. Hasil pengujian kasus uji tambah laporan keluhan ditunjukkan pada Tabel 6. 4.

Tabel 6. 4 Hasil Pengujian Unit *Method createLapor()*

Jalur	Prosedur	Hasil yang Diharapkan	Hasil	Status
1	<i>Class Test</i> melakukan pengecekan pada parameter yang diterima dari <i>methods</i> tambahLaporan()	Sistem akan menampilkan pesan error ketika user mencoba untuk mengirim laporan keluhan.	Sistem akan menampilkan pesan error ketika user mencoba untuk mengirim laporan keluhan.	Valid



	dan terdapat parameter yang kosong atau tidak diisi.			
2	<i>Class Test</i> melakukan pengecekan pada parameter yang diterima dari <i>methods</i> tambahLaporan() dan semua parameter telah diisi.	Sistem akan menampilkan pesan sukses ketika user mengirim laporan keluhan.	Sistem akan menampilkan pesan sukses ketika user mengirim laporan keluhan.	Valid

### 3. Pengujian Unit *Method* tambahIzin()

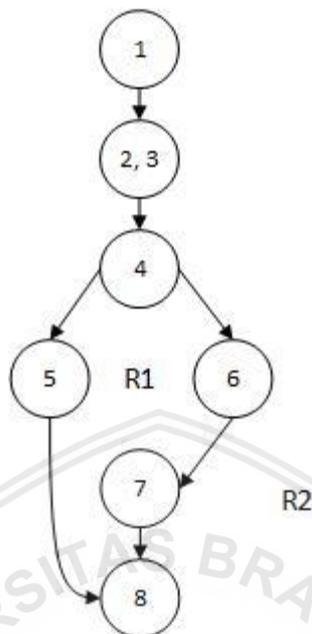
Pengujian unit *method* tambahIzin() dari *class* IzinController ini dilakukan untuk merepresentasikan aliran kontrol saat user menambahkan izin penggunaan jalan kepada Dinas Perhubungan Kota Mojokerto menggunakan algoritme yang ditunjukkan pada Tabel 6. 5.

Tabel 6. 5 Algoritme Pengujian *Method* tambahIzin()

tambahIzin ()	
No	Pseudocode
1	Mulai
2	
3	Inisialisasi userInfo
4	Inisialisasi verifyToken
5	Inisialisasi database
6	
7	Panggil verifyToken(userInfo)
8	
9	If (verifyToken == false) {
10	Return false
11	} else {
12	Return true
13	}
14	Endif
15	
16	Data = ambil data input
17	Img = ambil data gambar upload
18	
19	Selesai

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6. 5 maka akan dilakukan pengujian menggunakan teknik *basis path* yang akan menjelaskan *flowgraph* Gambar 6. 3, *cyclomatic complexity*, dan jalur independen.

a) Flowgraph



Gambar 6. 3 Flowgraph Method tambahizin()

b) Cyclomatic Complexity

- $V(G) = R = 2$
- $V(G) = E - N + 2 = 8 - 8 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

c) Jalur Independen

- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 8
- Jalur 2 = 1 – 2 – 3 – 4 – 6 – 7 – 8

Hasil dari *basis path* selanjutnya digunakan untuk menerapkan kasus uji dan untuk mengetahui kevalidannya. Kasus uji dinyatakan valid ketika hasil dan hasil yang diharapkan keduanya sama. Hasil pengujian kasus uji izin penggunaan jalan ditunjukkan pada Tabel 6. 6.

Tabel 6. 6 Hasil Pengujian Unit Method tambahizin()

Jalur	Prosedur	Hasil yang Diharapkan	Hasil	Status
1	Class Test memanggil <i>method</i> <i>verifyToken</i> dan mengembalikan nilai <i>false</i> karena <i>token</i> milik <i>user</i> tidak valid.	Sistem akan mengeluarkan user dari aplikasi dan menampilkan halaman login	Sistem akan mengeluarkan user dari aplikasi dan menampilkan halaman login	Valid



2	<i>Class Test</i> memanggil <i>method</i> <i>verifyToken</i> dan mengembalikan nilai <i>true</i> karena token milik user valid dan berhasil memanggil <i>method</i> <i>tambahIzin</i> .	Sistem akan menampilkan semua data input pada tampilan antarmuka user.	Sistem akan menampilkan semua data input pada tampilan antarmuka user.	Valid
---	---	--	--	-------

**4. Pengujian Unit Method *createlzin()***

Pengujian unit *method* *createlzin()* dari *class* *IzinModel* ini dilakukan untuk merepresentasikan aliran kontrol saat user menyimpan izin penggunaan jalan pada *database* Dinas Perhubungan Kota Mojokerto menggunakan algoritme yang ditunjukkan pada Tabel 6. 7.

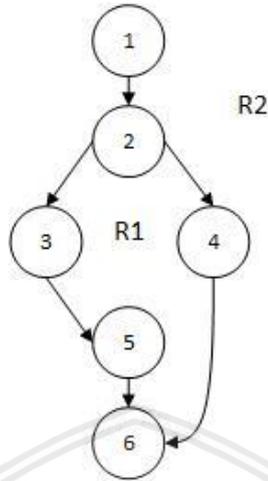
Tabel 6. 7 Algoritme Pengujian *Method* *createlzin()*

tambahIzin()	
No	Pseudocode
1	Mulai
2	
3	Data = data input dari controller
4	Img = data gambar dari controller
5	
6	If (img == data gambar    data == data lengkap) { ← 2
7	Return img, data
8	Return true } 3
9	} else {
10	Data tidak lengkap
11	Img tidak valid
12	Return false } 4
13	}
14	Endif
15	
16	Simpan data izin penggunaan jalan ke server ← 5
17	
18	Selesai ← 6

Berdasarkan algoritme yang telah dijelaskan pada Tabel 6. 7 maka akan dilakukan pengujian menggunakan teknik *basis path* yang akan menjelaskan *flowgraph* Gambar 6. 4, *cyclomatic complexity*, dan jalur independen.



a) Flowgraph



Gambar 6. 4 Flowgraph Method createlzin()

b) Cyclomatic Complexity

- $V(G) = R = 2$
- $V(G) = E - N + 2 = 6 - 6 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

c) Jalur Independen

- Jalur 1 = 1 - 2 - 4 - 6
- Jalur 2 = 1 - 2 - 3 - 5 - 6

Hasil dari *basis path* selanjutnya digunakan untuk menerapkan kasus uji dan untuk mengetahui kevalidannya. Kasus uji dinyatakan valid ketika hasil dan hasil yang diharapkan keduanya sama. Hasil pengujian kasus uji izin penggunaan jalan ditunjukkan pada Tabel 6. 8.

Tabel 6. 8 Hasil Pengujian Unit Method createlzin()

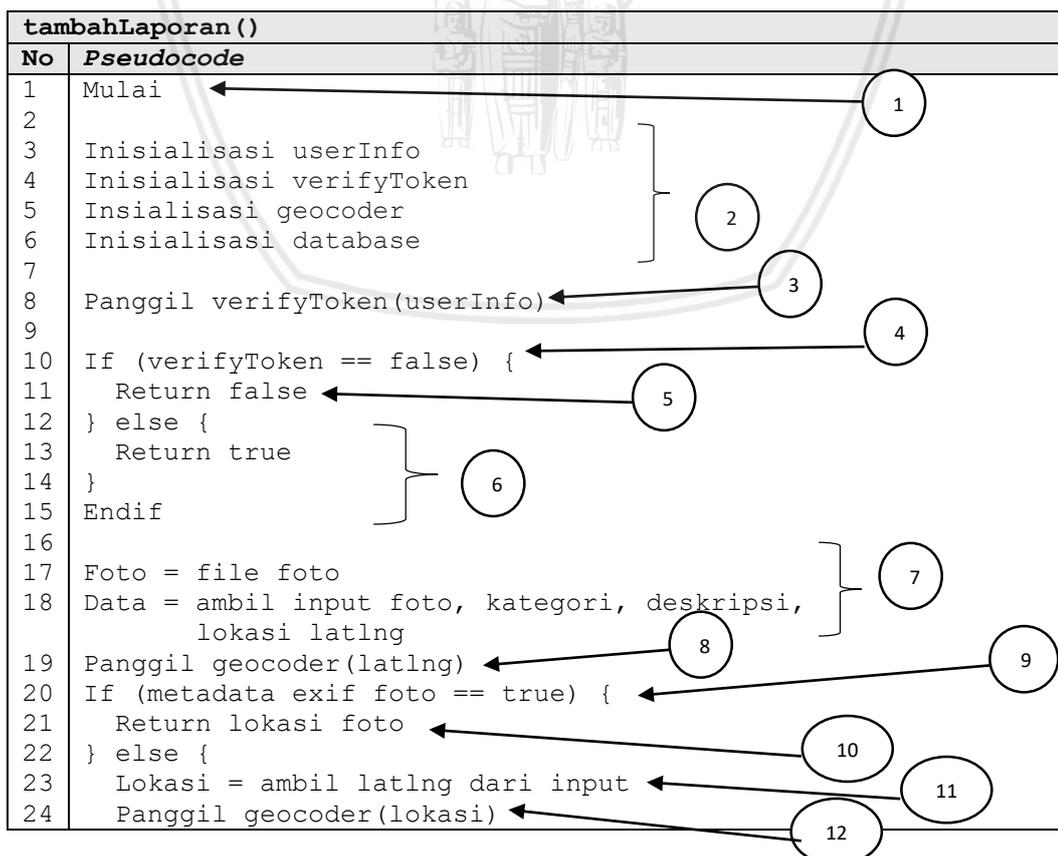
Jalur	Prosedur	Hasil yang Diharapkan	Hasil	Status
1	Class Test melakukan pengecekan pada parameter yang diterima dari <i>methods</i> tambahlzin() dan menerima parameter gambar tidak valid atau ada parameter yang kosong.	Sistem akan menampilkan pesan error ketika user mencoba untuk mengirim form izin penggunaan jalan	Sistem akan menampilkan pesan error ketika user mencoba untuk mengirim form izin penggunaan jalan	Valid

2	<i>Class Test</i> melakukan pengecekan pada parameter yang diterima dari <i>methods</i> tambahIzin() dan menerima parameter gambar valid dan parameter diisi semua dengan benar.	Sistem akan menampilkan pesan berhasil dan menyimpan data izin penggunaan jalan	Sistem akan menampilkan pesan berhasil dan menyimpan data izin penggunaan jalan	Valid
---	--	---	---	-------

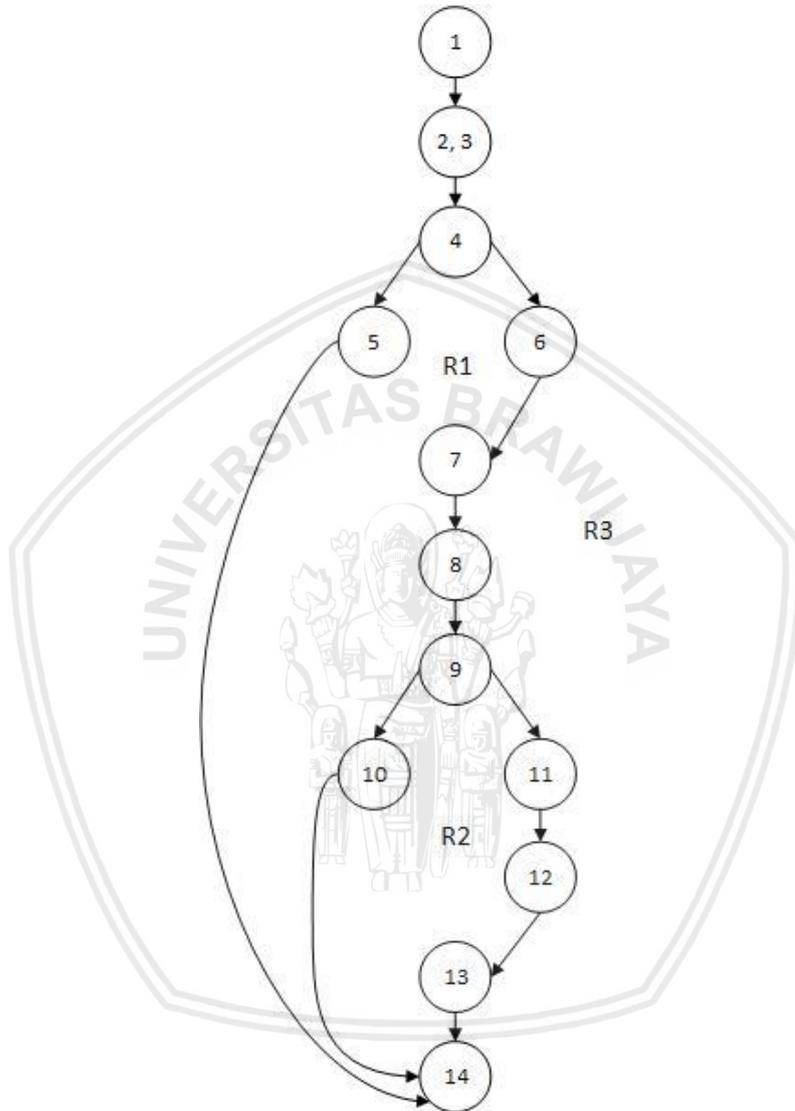
### 6.1.2 Pengujian Integrasi

Pengujian integrasi bertujuan untuk berfokus lebih pada pengujian penggabungan dari dua atau lebih unit pada perangkat lunak. Pengujian integrasi tambah laporan keluhan merupakan fungsionalitas gabungan dari *method* tambahLaporan() dari *class* LaporanController dan *method* createLapor() dari *class* LaporanModel. Berikut adalah algoritme dan *flowgraph* dari *method* tambahLaporan() dan yang ditunjukkan pada Tabel 6. 9 dan Gambar 6. 5 *Flowgraph Method tambahLaporan()*.

Tabel 6. 9 Algoritme Pengujian Integrasi *Method* tambahLaporan()



25	Return lokasi foto	}	13
26	}		
27	Endif		
28			
29	Selesai	←	14

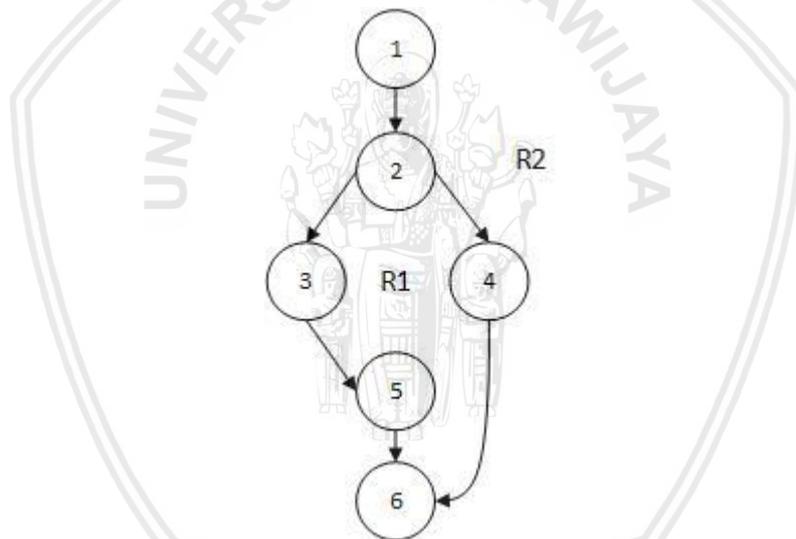


Gambar 6. 5 Flowgraph Method tambahLaporan()

Method tambahLaporan() memanggil method createLapor() untuk menyimpan data laporan keluhan yang telah dimasukkan user. Algoritme dan flowgraph method createLapor() ditunjukkan pada Tabel 6. 10 dan Gambar 6. 6.

Tabel 6. 10 Algoritme Pengujian Integrasi Method createLapor()

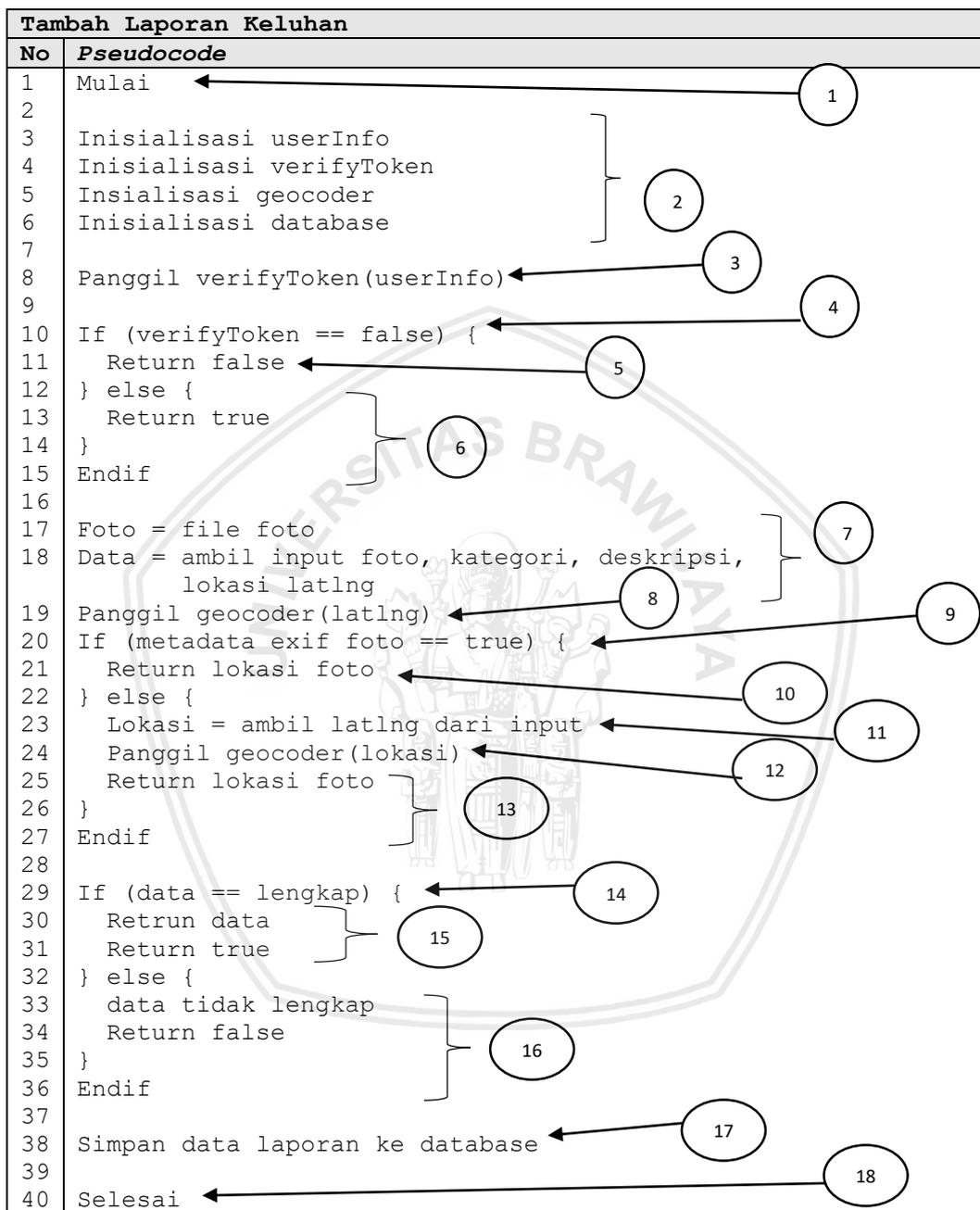
createLapor ()	
No	Pseudocode
1	Mulai
2	Data = data input dari controller
3	If (data == lengkap) {
4	Retrun data
5	Return true
6	} else {
7	data tidak lengkap
8	Return false
9	}
10	Endif
11	Simpan data laporan ke database
12	
13	
14	Selesai



Gambar 6. 6 Flowgraph Method createLapor()

Dari algoritme yang ditunjukkan pada Tabel 6. 9 dan Tabel 6. 10 menghasilkan pseudocode dan flowgraph gabungan yang ditunjukkan pada Tabel 6. 11 dan Gambar 6. 7.

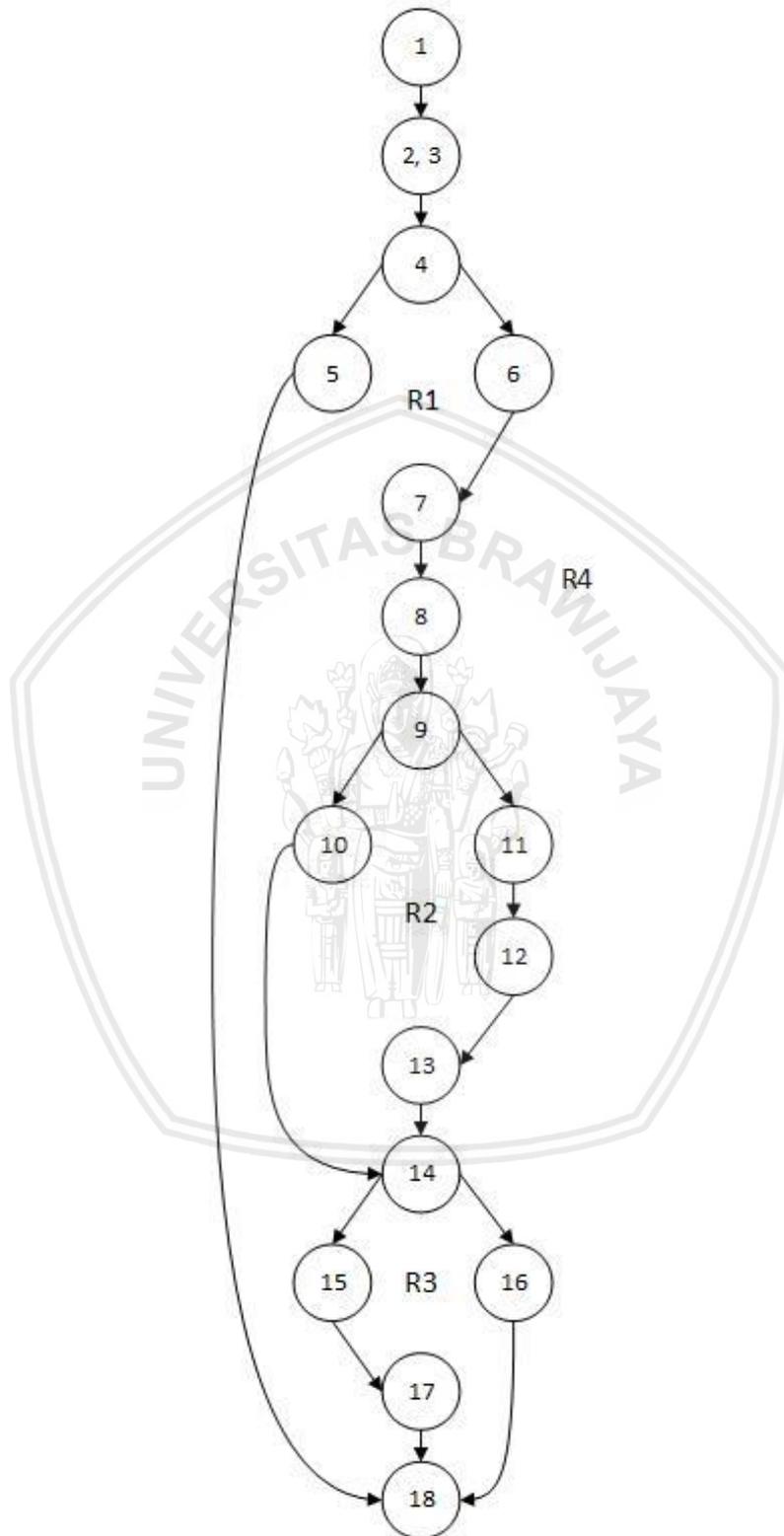
Tabel 6. 11 Algoritme Gabungan *Method* tambahLaporan() dan createLapor()



Berdasarkan algoritme yang telah dijelaskan pada Tabel 6. 11 maka akan dilakukan pengujian menggunakan teknik *basis path* yang akan menjelaskan *flowgraph* yang ditunjukkan pada Gambar 6. 7, *cyclomatic complexity*, dan jalur independen.



a) *Flowgraph*



Gambar 6. 7 *Flowgraph* Gabungan *Method* *tambahLaporan()* & *createLapor()*

**b) Cyclomatic Complexity**

- $V(G) = R = 4$
- $V(G) = E - N + 2 = 20 - 18 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

**c) Jalur Independen**

- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 18
- Jalur 2 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 14 – 15 – 17 – 18
- Jalur 3 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 11 – 12 – 13 – 14 – 16 – 18
- Jalur 4 = 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 11 – 12 – 13 – 14 – 15 – 17 – 18

Hasil dari *basis path* selanjutnya digunakan untuk menerapkan kasus uji dan untuk mengetahui kevalidannya. Kasus uji dinyatakan valid ketika hasil dan hasil yang diharapkan keduanya sama. Hasil pengujian kasus uji tambah laporan keluhan ditunjukkan pada Tabel 6. 12.

**Tabel 6. 12 Hasil Pengujian Integrasi Tambah Laporan Keluhan**

Jalur	Prosedur	Hasil yang Diharapkan	Hasil	Status
1	<i>Class Test</i> memanggil <i>method verifyToken</i> dan mengembalikan nilai <i>false</i> karena <i>token</i> milik <i>user</i> tidak valid.	Sistem akan mengeluarkan user dari aplikasi dan menampilkan halaman login	Sistem akan mengeluarkan user dari aplikasi dan menampilkan halaman login	Valid
2	<i>Class Test</i> memanggil <i>method verifyToken</i> dan mengembalikan nilai <i>true</i> karena <i>token</i> milik <i>user valid</i> , lalu memanggil <i>method tambahLaporan</i> dengan parameter file foto “foto.jpg”, kategori “Lalu Lintas” dan deskripsi “tes lapor” lalu memanggil <i>method geocoder</i> menggunakan file foto yang mengembalikan nilai <i>true</i> dan koordinat berupa <i>latitude</i> dan	Sistem akan menampilkan foto yang dipilih dan data lokasi yang didapat dari metadata foto berupa koordinat pada map dan berupa alamat pada label lokasi.	Sistem akan menampilkan foto yang dipilih dan data lokasi yang didapat dari metadata foto berupa koordinat pada map dan berupa alamat pada label lokasi.	Valid



	<i>longitude</i> sesuai pada metadata file foto dan diproses menjadi alamat yang digunakan pada <i>field</i> lokasi.			
3	<i>Class Test</i> memanggil <i>method</i> <i>verifyToken</i> dan mengembalikan nilai <i>true</i> karena <i>token</i> milik <i>user</i> valid, lalu memanggil <i>method</i> <i>tambahLaporan</i> dengan parameter file foto "foto.jpg" yang tidak memiliki data lokasi pada metadata file foto, kategori "Lalu Lintas" dan deskripsi "tes lapor" lalu memanggil <i>method</i> <i>geocoder</i> menggunakan file foto yang mengembalikan nilai <i>false</i> karena tidak ditemukan <i>latitude</i> dan <i>longitude</i> sehingga <i>field</i> lokasi kosong atau tidak diisi.	Sistem akan menampilkan foto yang dipilih dan akan menampilkan pesan error ketika user mencoba untuk mengirim laporan keluhan karena terdapat <i>field</i> yang kosong atau tidak diisi.	Sistem akan menampilkan foto yang dipilih dan akan menampilkan pesan error ketika user mencoba untuk mengirim laporan keluhan karena terdapat <i>field</i> yang kosong atau tidak diisi.	Valid
4	<i>Class Test</i> memanggil <i>method</i> <i>verifyToken</i> dan mengembalikan nilai <i>true</i> karena <i>token</i> milik <i>user</i> valid, lalu memanggil <i>method</i> <i>tambahLaporan</i> dengan parameter file foto "foto.jpg" yang tidak memiliki	Sistem akan menampilkan foto yang dipilih dan data lokasi yang didapat dari user ketika memilih titik lokasi laporan keluhan melalui bantuan <i>map</i> dan <i>leaflet</i>	Sistem akan menampilkan foto yang dipilih dan data lokasi yang didapat dari user ketika memilih titik lokasi laporan keluhan melalui bantuan <i>map</i> dan <i>leaflet</i>	Valid

	<p>data lokasi pada metadata file foto, kategori “Lalu Lintas” dan deskripsi “tes lapor” lalu memanggil <i>method geocoder</i> menggunakan file foto yang mengembalikan nilai <i>false</i> karena tidak ditemukan <i>latitude</i> dan <i>longitude</i> lalu mengisi <i>latitude</i> dan <i>longitude</i> dengan cara memilih titik lokasi dengan bantuan <i>map leaflet</i> yang disediakan dan <i>geocoder</i> akan otomatis dipanggil dan menerima data <i>latitude</i> dan <i>longitude</i> sehingga <i>field</i> lokasi akan terisi dengan alamat yang baru.</p>	<p>menyimpan data lokasi pada metadata file foto.</p>	<p>menyimpan data lokasi pada metadata file foto.</p>	
--	--	---	---	--

### 6.1.3 Pengujian Validasi

Pengujian validasi dilakukan untuk melakukan pengujian pada setiap fungsionalitas sistem yang didapat dari daftar kebutuhan fungsional untuk mencari tahu apakah aplikasi yang telah dikembangkan sudah sesuai dengan kebutuhan fungsional yang telah dijelaskan sebelumnya pada bab rekayasa kebutuhan. Hasil dari pengujian validasi ditunjukkan pada Tabel 6. 13.



Tabel 6. 13 Hasil Pengujian Validasi

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
1	Register (Main Flow)	Memasukkan data registrasi dengan lengkap dan email yang digunakan belum terdaftar.	Guest berhasil melakukan registrasi dan diarahkan kembali ke halaman <i>login</i> .	Guest berhasil melakukan registrasi dan diarahkan kembali ke halaman <i>login</i> .	Valid
2	Register (Alternate Flow 1)	Memasukkan data registrasi lengkap tetapi email yang digunakan sudah terdaftar.	Guest gagal melakukan registrasi dan akan mendapat pesan <i>error</i> "Email sudah terdaftar" dari aplikasi.	Guest gagal melakukan registrasi dan akan mendapat pesan <i>error</i> "Email sudah terdaftar" dari aplikasi.	Valid
3	Register (Alternate Flow 2)	Memasukkan data registrasi tidak lengkap atau terdapat <i>input field</i> yang tidak diisi.	Guest gagal melakukan registrasi dan akan mendapat pesan <i>error</i> pada masing-masing <i>input field</i> yang tidak diisi.	Guest gagal melakukan registrasi dan akan mendapat pesan <i>error</i> pada masing-masing <i>input field</i> yang tidak diisi.	Valid
4	Login (Main Flow)	Memasukkan email dan password sesuai dengan yang telah didaftarkan.	Aktor berhasil melakukan login dan akan diarahkan ke halaman <i>home</i> .	Guest berhasil melakukan login dan akan diarahkan ke halaman <i>home</i> .	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
5	Login (Alternate Flow)	Memasukkan email dan password tidak sesuai dengan yang telah didaftarkan atau email dan password tidak terdaftar.	Aktor gagal melakukan login dan akan mendapat pesan <i>error</i> "Kombinasi email dan password salah"	Aktor gagal melakukan login dan akan mendapat pesan <i>error</i> "Kombinasi email dan password salah"	Valid
6	Lihat Laporan Keluhan (Main Flow)	Membuka halaman home dan data laporan keluhan berhasil ditampilkan.	User berhasil melihat daftar laporan keluhan yang ada pada aplikasi.	User berhasil melihat daftar laporan keluhan yang ada pada aplikasi.	Valid
7	Tambah Laporan Keluhan (Main Flow)	Memasukkan <i>input</i> data tambah laporan keluhan sesuai dengan <i>input field</i> yang diminta dengan lengkap dan benar.	User berhasil menambahkan data laporan keluhan dan diarahkan ke halaman <i>home</i> .	User berhasil menambahkan data laporan keluhan dan diarahkan ke halaman <i>home</i> .	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
8	Tambah Laporan Keluhan ( <i>Alternate Flow</i> )	Memasukkan <i>input</i> data tambah laporan keluhan dengan tidak lengkap atau ada <i>input field</i> yang kosong.	User gagal menambakan laporan keluhan dan diberikan pesan <i>error</i> "Pengisian laporan keluhan tidak lengkap" dari aplikasi.	User gagal menambakan laporan keluhan dan diberikan pesan <i>error</i> "Pengisian laporan keluhan tidak lengkap" dari aplikasi.	Valid
9	Edit Laporan Keluhan ( <i>Main Flow</i> )	Memasukkan <i>input</i> data edit laporan keluhan sesuai dengan <i>input field</i> yang diminta dengan lengkap dan benar.	User berhasil menambahkan data laporan keluhan dan diarahkan ke halaman <i>home</i> .	User berhasil menambahkan data laporan keluhan dan diarahkan ke halaman <i>home</i> .	Valid
10	Edit Laporan Keluhan ( <i>Alternate Flow</i> )	Memasukkan <i>input</i> data edit laporan keluhan dengan tidak lengkap atau ada <i>input field</i> yang kosong.	User gagal mengedit laporan keluhan dan diberikan pesan <i>error</i> "Pengisian laporan keluhan tidak lengkap" dari aplikasi.	User gagal mengedit laporan keluhan dan diberikan pesan <i>error</i> "Pengisian laporan keluhan tidak lengkap" dari aplikasi.	Valid
11	Vote Laporan Keluhan ( <i>Main Flow</i> )	Menekan tombol <i>vote</i> pada salah satu daftar laporan keluhan.	User berhasil melakukan <i>vote</i> pada salah satu laporan keluhan.	User berhasil melakukan <i>vote</i> pada salah satu laporan keluhan.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
12	Tambah Komentar Laporan Keluhan (Main Flow)	Memasukkan komentar pada salah satu daftar laporan keluhan.	User berhasil menambahkan komentar pada salah satu laporan keluhan.	User berhasil menambahkan komentar pada salah satu laporan keluhan.	Valid
13	Tambah Komentar Laporan Keluhan (Alternate Flow)	Menekan tombol kirim tanpa mengisi <i>input</i> komentar.	User gagal menambahkan komentar pada salah satu laporan keluhan dan menerima pesan <i>error</i> "Komentar kosong".	User gagal menambahkan komentar pada salah satu laporan keluhan dan menerima pesan <i>error</i> "Komentar kosong".	Valid
14	Edit Komentar Laporan Keluhan (Main Flow)	Mengubah komentar pada salah satu daftar laporan keluhan.	User berhasil mengubah komentarnya pada salah satu laporan keluhan.	User berhasil mengubah komentarnya pada salah satu laporan keluhan.	Valid
15	Edit Komentar Laporan Keluhan (Alternate Flow)	Menekan tombol kirim tanpa mengisi <i>input</i> komentar.	User gagal mengubah komentarnya pada salah satu laporan keluhan dan menerima pesan <i>error</i> "Komentar kosong".	User gagal mengubah komentarnya pada salah satu laporan keluhan dan menerima pesan <i>error</i> "Komentar kosong".	Valid
16	Hapus Komentar Laporan Keluhan (Main Flow)	Menekan tombol hapus pada salah satu komentar milik <i>user</i> .	User berhasil menghapus komentarnya.	User berhasil menghapus komentarnya.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
17	Lihat Info Trayek ( <i>Main Flow</i> )	Membuka halaman info dan memilih tab trayek.	User berhasil melihat daftar informasi terkait trayek pada aplikasi.	User berhasil melihat daftar informasi terkait trayek pada aplikasi.	Valid
18	Lihat Info Pengumuman ( <i>Main Flow</i> )	Membuka halaman info dan memilih tab pengumuman.	User berhasil melihat daftar informasi terkait pengumuman pada aplikasi.	User berhasil melihat daftar informasi terkait pengumuman pada aplikasi.	Valid
19	Izin Penggunaan Jalan ( <i>Main Flow</i> )	Memasukkan data <i>input</i> izin penggunaan jalan dengan lengkap.	User berhasil menambahkan data izin penggunaan jalan dan diarahkan kembali ke halaman <i>list</i> izin.	User berhasil menambahkan data izin penggunaan jalan dan diarahkan kembali ke halaman <i>list</i> izin.	Valid
20	Izin Penggunaan Jalan ( <i>Alternate Flow</i> )	Memasukkan data <i>input</i> izin penggunaan jalan dengan tidak lengkap atau terdapat <i>field input</i> yang tidak diisi.	User gagal menambahkan data izin penggunaan jalan dan ditampilkan pesan <i>error</i> pada masing-masing <i>field</i> yang tidak diisi.	User gagal menambahkan data izin penggunaan jalan dan ditampilkan pesan <i>error</i> pada masing-masing <i>field</i> yang tidak diisi.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
21	Survei Kepuasan Masyarakat (Main Flow)	Mengisi seluruh <i>input</i> jawaban pada pertanyaan survei yang ada.	User berhasil menambahkan data survei kepuasan masyarakat dan diarahkan kembali ke halaman profil.	User berhasil menambahkan data survei kepuasan masyarakat dan diarahkan kembali ke halaman profil.	Valid
22	Survei Kepuasan Masyarakat (Alternate Flow)	Tidak mengisi seluruh <i>input</i> jawaban pada pertanyaan survei yang ada.	User gagal menambahkan data survei kepuasan masyarakat dan ditampilkan pesan <i>error</i> "Lengkapi pengisian form".	User gagal menambahkan data survei kepuasan masyarakat dan ditampilkan pesan <i>error</i> "Lengkapi pengisian form".	Valid
23	Kritik dan Saran (Main Flow)	Memilih <i>rate</i> dan mengisi form deskripsi dengan benar.	User berhasil menambahkan data kritik dan saran dan diarahkan kembali ke halaman profil.	User berhasil menambahkan data kritik dan saran dan diarahkan kembali ke halaman profil.	Valid
24	Kritik dan Saran (Alternate Flow)	Tidak mengisi salah satu atau semua <i>input field</i> yang ada.	User gagal menambahkan data kritik saran dan mendapat pesan <i>error</i> "Harap lengkapi pengisian form".	User gagal menambahkan data kritik saran dan mendapat pesan <i>error</i> "Harap lengkapi pengisian form".	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
25	Edit Profile (Main Flow)	Mengisi semua <i>input</i> data untuk mengubah data profil.	<i>User</i> berhasil mengubah data profil miliknya dan diarahkan kembali ke halaman profil.	<i>User</i> berhasil mengubah data profil miliknya dan diarahkan kembali ke halaman profil.	Valid
26	Edit Profile (Alternate Flow)	Tidak mengisi semua <i>input</i> data untuk mengubah data profil.	<i>User</i> gagal mengubah data profil miliknya dan mendapat pesan <i>error</i> pada masing-masing <i>field</i> yang tidak diisi.	<i>User</i> gagal mengubah data profil miliknya dan mendapat pesan <i>error</i> pada masing-masing <i>field</i> yang tidak diisi.	Valid
27	Verifikasi Laporan Keluhan (Main Flow)	Memilih salah satu opsi yang tersedia yaitu terima atau tolak.	<i>Admin</i> berhasil melakukan verifikasi laporan keluhan dan mengubah status dari laporan keluhan yang diverifikasi.	<i>Admin</i> berhasil melakukan verifikasi laporan keluhan dan mengubah status dari laporan keluhan yang diverifikasi.	Valid
28	Update Status Laporan Keluhan (Main Flow)	Memilih salah opsi selesai untuk mengakhiri proses laporan keluhan.	<i>Admin</i> berhasil melakukan <i>update</i> status dari laporan keluhan user menjadi selesai.	<i>Admin</i> berhasil melakukan <i>update</i> status dari laporan keluhan user menjadi selesai.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
29	Cari Laporan Keluhan (Main Flow)	Memasukkan kata kunci pada form pencarian laporan keluhan.	Admin berhasil mendapatkan data laporan keluhan berdasarkan kata kunci yang digunakan.	Admin berhasil mendapatkan data laporan keluhan berdasarkan kata kunci yang digunakan.	Valid
30	Tambah Info Trayek (Main Flow)	Memasukkan data info trayek sesuai dengan <i>input field</i> yang tersedia.	Admin berhasil menambahkan data info trayek dan mendapat pesan berhasil.	Admin berhasil menambahkan data info trayek dan mendapat pesan berhasil.	Valid
31	Edit Info Trayek (Main Flow)	Memasukkan data baru sesuai dengan <i>input field</i> yang tersedia untuk mengubah data pada salah satu data info trayek.	Admin berhasil mengubah data pada salah satu info trayek dan mendapat pesan berhasil.	Admin berhasil mengubah data pada salah satu info trayek dan mendapat pesan berhasil.	Valid
32	Hapus Info Trayek (Main Flow)	Menekan tombol hapus pada salah satu data info trayek.	Admin berhasil menghapus data info trayek dan mendapat pesan berhasil.	Admin berhasil menghapus data info trayek dan mendapat pesan berhasil.	Valid
33	Tambah Info Pengumuman (Main Flow)	Memasukkan data info pengumuman sesuai dengan <i>input field</i> yang tersedia.	Admin berhasil menambahkan data info pengumuman dan mendapat pesan berhasil.	Admin berhasil menambahkan data info pengumuman dan mendapat pesan berhasil.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
34	Edit Info Pengumuman (Main Flow)	Memasukkan data baru sesuai dengan <i>input field</i> yang tersedia untuk mengubah data pada salah satu data info pengumuman.	<i>Admin</i> berhasil mengubah data pada salah satu info pengumuman dan mendapat pesan berhasil.	<i>Admin</i> berhasil mengubah data pada salah satu info pengumuman dan mendapat pesan berhasil.	Valid
35	Hapus Info Trayek (Main Flow)	Menekan tombol hapus pada salah satu data info trayek.	<i>Admin</i> berhasil menghapus data info trayek dan mendapat pesan berhasil.	<i>Admin</i> berhasil menghapus data info trayek dan mendapat pesan berhasil.	Valid
36	Verifikasi Izin Penggunaan Jalan (Main Flow)	Memilih salah satu opsi yang tersedia yaitu terima atau tolak.	<i>Admin</i> berhasil melakukan verifikasi izin penggunaan jalan dan mengubah status dari izin penggunaan jalan yang diverifikasi.	<i>Admin</i> berhasil melakukan verifikasi izin penggunaan jalan dan mengubah status dari izin penggunaan jalan yang diverifikasi.	Valid
37	Lihat Survei Kepuasan Masyarakat (Main Flow)	Membuka halaman survei kepuasan masyarakat dan memilih menu hasil survei.	<i>Admin</i> berhasil melihat data hasil survei kepuasan masyarakat yang telah masuk.	<i>Admin</i> berhasil melihat data hasil survei kepuasan masyarakat yang telah masuk.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
38	Tambah Survei Kepuasan Masyarakat (Main Flow)	Memasukkan <i>input</i> data tambah survei sesuai dengan <i>input field</i> yang diminta dengan lengkap dan benar.	<i>Admin</i> berhasil menambahkan data survei kepuasan masyarakat dan mendapat pesan berhasil.	<i>Admin</i> berhasil menambahkan data survei kepuasan masyarakat dan mendapat pesan berhasil.	Valid
39	Tambah Survei Kepuasan Masyarakat (Alternate Flow)	Memasukkan <i>input</i> data tambah survei dengan tidak lengkap atau ada <i>input field</i> yang kosong.	<i>Admin</i> gagal menambahkan data survei kepuasan masyarakat dan mendapat pesan gagal.	<i>Admin</i> gagal menambahkan data survei kepuasan masyarakat dan mendapat pesan gagal.	Valid
40	Edit Survei Kepuasan Masyarakat (Main Flow)	Memasukkan <i>input</i> data edit survei sesuai dengan <i>input field</i> yang diminta dengan lengkap dan benar.	<i>Admin</i> berhasil mengubah data survei kepuasan masyarakat dan mendapat pesan berhasil.	<i>Admin</i> berhasil mengubah data survei kepuasan masyarakat dan mendapat pesan berhasil.	Valid
41	Edit Survei Kepuasan Masyarakat (Alternate Flow)	Memasukkan <i>input</i> data edit survei sesuai dengan <i>input field</i> yang diminta dengan tidak lengkap atau ada <i>input field</i> yang kosong.	<i>Admin</i> gagal mengubah data survei kepuasan masyarakat dan mendapat pesan gagal.	<i>Admin</i> gagal mengubah data survei kepuasan masyarakat dan mendapat pesan gagal.	Valid

No	Test Case	Test Scenario	Expected Results	Actual Results	Status
42	Hapus Survei Kepuasan Masyarakat (Main Flow)	Menekan tombol hapus pada salah satu data survei kepuasan masyarakat.	Admin berhasil menghapus data survei kepuasan masyarakat dan mendapat pesan berhasil.	Admin berhasil menghapus data survei kepuasan masyarakat dan mendapat pesan berhasil.	Valid
43	Lihat Kritik dan Saran (Main Flow)	Membuka halaman kritik dan saran.	Admin berhasil melihat data hasil kritik dan saran yang telah masuk.	Admin berhasil melihat data hasil kritik dan saran yang telah masuk.	Valid
44	Logout (Main Flow)	Menekan tombol logout.	Aktor berhasil keluar dari sesi aplikasi dan diarahkan ke halaman login.	Aktor berhasil keluar dari sesi aplikasi dan diarahkan ke halaman login.	Valid

### 6.1.4 Pengujian *Compatibility*

Pengujian *Compatibility* digunakan sebagai pengujian untuk menentukan suatu *environment* atau lingkungan yang diharapkan dapat menjalankan sistem yang telah dikembangkan. Pengujian *compatibility* dilakukan pada tiga perangkat android yaitu Honor 8X, Xiaomi Redmi Note 4 dan Asus Zenfone 5 yang masing - masing perangkat memiliki spesifikasi perangkat yang berbeda. Pada pengujian ini hanya akan berfokus pada aspek kompatibilitas pada aspek fungsional dan antarmuka. Berikut adalah parameter perbandingan dari tiga perangkat pengujian yang ditunjukkan pada Tabel 6. 14.

Tabel 6. 14 Perbandingan Spesifikasi Perangkat Pengujian *Compatibility*

Parameter	Perangkat		
	Honor 8X (2018)	Xiaomi Redmi Note 4 (2017)	Asus Zenfone 5 (2014)
Ukuran Layar	6.5 inches	5.5 inches	5.0 inches
Kerapatan Piksel	397 ppi	401 ppi	294 ppi
Resolusi Layar	1080 x 2340 pixels	1080 x 1920 pixels	720 x 1280 pixels
Aspek Rasio	19,5:9	16:9	16:9
Sistem Operasi	Android 9.0.1 (Pie)	Android 7.0 (Nougat)	Android 5.0.2 (Lollipop)



Pada tahap pengujian ini hanya akan diperlihatkan beberapa pengujian yang menjadi pokok dalam sistem, diantaranya yaitu tambah laporan keluhan, lihat info trayek, izin penggunaan jalan, survei kepuasan masyarakat dan kritik & saran. Berikut adalah hasil pengujian *compatibility* yang telah dilakukan dapat dilihat pada Tabel 6. 15.

Tabel 6. 15 Hasil Pengujian *Compatibility*

Test Case	Test Scenario	Expected Result	Actual Result	Hasil Pengujian		
				Honor 8X	Xiaomi Redmi Note 4	Asus Zenfone 5
Tambah Laporan Keluhan	<i>Input</i> data tambah laporan keluhan sesuai dengan <i>input field</i> yang diminta dengan lengkap dan benar.	<i>User</i> berhasil menambahkan data laporan keluhan dan diarahkan ke halaman <i>home</i> .	<i>User</i> berhasil menambahkan data laporan keluhan dan diarahkan ke halaman <i>home</i> .	Valid	Valid	Valid
Lihat Info Trayek	Membuka halaman info dan memilih tab trayek.	<i>User</i> berhasil melihat daftar informasi terkait trayek pada aplikasi.	<i>User</i> berhasil melihat daftar informasi terkait trayek pada aplikasi.	Valid	Valid	Valid
Izin Penggunaan Jalan	<i>Input</i> data izin penggunaan jalan dengan lengkap.	<i>User</i> berhasil menambahkan data izin dan diarahkan kembali ke halaman <i>list</i> izin.	<i>User</i> berhasil menambahkan data izin dan diarahkan kembali ke halaman <i>list</i> izin.	Valid	Valid	Valid

Test Case	Test Scenario	Expected Result	Actual Result	Hasil Pengujian		
				Honor 8X	Xiaomi Redmi Note 4	Asus Zenfone 5
Survei Kepuasan Masyarakat	Mengisi seluruh <i>input</i> jawaban pada pertanyaan survei yang ada.	<i>User</i> berhasil menambahkan data survei kepuasan masyarakat dan diarahkan kembali ke halaman profil.	<i>User</i> berhasil menambahkan data survei kepuasan masyarakat dan diarahkan kembali ke halaman profil.	Valid	Valid	Valid
Kritik dan Saran	Memilih <i>rate</i> dan mengisi form deskripsi dengan benar.	<i>User</i> berhasil menambahkan data kritik dan saran dan diarahkan kembali ke halaman profil.	<i>User</i> berhasil menambahkan data kritik dan saran dan diarahkan kembali ke halaman profil.	Valid	Valid	Valid

Dengan perbedaan - perbedaan yang dimiliki ketiga perangkat seperti ukuran layar, kerapatan piksel, resolusi layar, aspek rasio dan sistem operasi, hasil pengujian menunjukkan semua perangkat dapat menjalankan semua kasus uji dengan benar. Semua fungsional sistem dapat berjalan dengan baik dan tampilan antarmuka berhasil ditampilkan dengan baik.

## BAB VII PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil dari setiap tahapan pada rekayasa kebutuhan, perancangan, implementasi, dan pengujian sistem maka dapat disimpulkan seperti berikut:

1. Berdasarkan hasil rekayasa kebutuhan didapatkan tiga orang aktor, 31 kebutuhan fungsional dan satu kebutuhan non fungsional. Hasil pada tahapan rekayasa kebutuhan didapatkan dari hasil studi literatur dan hasil wawancara dengan pihak Dinas Perhubungan Kota Mojokerto.
2. Berdasarkan hasil perancangan dari proses rekayasa kebutuhan didapat hasil yaitu perancangan arsitektur menggunakan diagram *sequence* dan diagram *class*, perancangan algoritme, perancangan antarmuka dan perancangan basis data menggunakan ERD (*Entity Relationship Diagram*). Berdasarkan tahap perancangan, dilakukan tahap implementasi dengan melakukan implementasi kode program menggunakan VueJS pada bagian *front-end* dan NodeJS pada bagian *back-end*, implementasi antarmuka dan implementasi basis data menggunakan MySQL.
3. Pada tahap pengujian dilakukan tiga pengujian, yaitu pengujian unit, pengujian validasi dan pengujian *compatibility*. Pengujian unit dilakukan menggunakan *basis path testing* menunjukkan 100% valid dari lima kasus uji. Hasil pengujian validasi menunjukkan 100% valid dari 44 kasus uji pada semua kebutuhan fungsional berdasarkan skenario *use case*. Hasil dari pengujian *compatibility* menunjukkan 100% valid dari lima kasus uji yang dilakukan di tiga perangkat android dengan versi android yang berbeda-beda.

### 7.2 Saran

Berdasarkan kesimpulan yang telah diuraikan, maka yang dapat disarankan untuk penelitian selanjutnya adalah:

1. Penambahan fitur *sharing* untuk membagikan laporan keluhan atau artikel kepada pengguna lain.

## DAFTAR PUSTAKA

- Abdullah, R., 2018. *7 in 1 Pemrograman Web Tingkat Lanjut*. Jakarta: Elex Media Komputindo.
- Abdulloh, R., 2018. *Mudah Membuat Aplikasi Android dengan Ionic 3*. s.l.:Elex Media Komputindo.
- Dishubkominfo, 2014. *Profil Dinas Perhubungan Komunikasi dan Informatika Kota Mojokerto*. Mojokerto: Dishubkominfo.
- Fajrin, R., 2017. Pengembangan Sistem Informasi Geografis Berbasis Node.js untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM. *Jurnal Informatika*, 11(2).
- Hamad, H., Saad, M. & Abed, R., 2010. Performance Evaluation of RESTful Web Services for Mobile Devices. *International Arab Journal of e-Technology*, Volume 1.
- Joshi, D., Gallagher, A., Yu, J. & Luo, J., 2010. *Exploring User Image Tags for Geo-Location Inference*. New York: Eastman Kodak Company.
- JSON, 1999. *Pengenalan JSON*. [Online] Available at: <http://www.json.org/json-id.html> [Diakses 2019 Maret 10].
- Kharlampidi, V., 2019. *Introduction to Framework7 | Framework7 Documentation*. [Online] Available at: <https://framework7.io/docs/introduction.html>
- Mardani, A., 2014. Sistem Informasi Geografis Pelaporan Masyarakat (SIGMA) Berbasis Foto Geotag. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, Volume 3.
- Munawar, 2005. *Pemodelan Visual dengan UML*. 1 penyunt. Yogyakarta: Graha Ilmu.
- Prasetya, D. R., Domai, T. & Mindarti, L. I., 2013. Analisis Pengelolaan Pengaduan Masyarakat Dalam Rangka Pelayanan Publik (Studi Pada Dinas Komunikasi dan Informatika Kota Malang). *Jurnal Jurusan Administrasi Publik, Fakultas Ilmu Administrasi, Universitas Brawijaya*, Volume 1.
- Prasetyo, T. W., Prestiliano, J. & Tampake, H. S., 2012. Perancangan dan Implementasi Aplikasi Penyimpanan Lokasi dan Geotagging Berbasis Mobile Menggunakan GPS pada Android. *Jurnal Teknologi Informasi Universitas Kristen Satya Wacana*.
- Simarmata, J., 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.

Susanto, R. & Andriana, A. D., 2016. Perbandingan Model Waterfall Dan Prototyping untuk Pengembangan Sistem Informasi. *Majalah Ilmiah UNIKOM*, 14(1), pp. 41 -46.

Taylor, K. & Silver, L., 2019. Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally. *Pew Research Center*.

Tompoh, J. F., Sentinuwo, S. R. & Sinsuw, A. A. E., 2016. Rancang Bangun Aplikasi Pemesanan Menu Makanan Restoran Berbasis Android. *E-journal Teknik Informatika*, 9(1).

Urva, G. & Siregar, H. F., 2015. Pemodelan UML E-Marketing Minyak Goreng. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 1(9), pp. 92 - 101.

Wibisono, W. & Baskoro, F., 2002. Pengujian Perangkat Lunak dengan Menggunakan Model Behaviour UML. *Jurnal Ilmiah Teknologi Informasi*, 1(1), pp. 43 - 50.

Wijaya, W., Tolle, H. & Kharisma, A. P., 2018. Rancang Bangun Aplikasi Geotagging Social Report Bencana Banjir. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(7).

Wiyanto, Suyahmo & Sunarto, 2017. Pengelolaan Komplain (Keluhan) Masyarakat dalam Mewujudkan Pemerintahan yang Baik di Kota Semarang. *Jurnal Pendidikan Kewarganegaraan*, Volume 4.

