



**PEMBANGUNAN APLIKASI MOBILE IDENTIFIKASI PARKIR  
NON RESMI BERDASARKAN KONSEP CROWDSOURCING  
MENGUNAKAN GEOTAGGING**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Indah Ratna Sari  
NIM: 175150209111029



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2020

## PERSETUJUAN

PEMBANGUNAN APLIKASI *MOBILE* IDENTIFIKASI PARKIR NON RESMI  
BERDASARKAN KONSEP *CROWDSOURCING* MENGGUNAKAN *GEOTAGGING*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Indah Ratna Sari  
NIM: 175150209111029

Skripsi ini telah diuji dan dinyatakan lulus pada  
3 Juni 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Faizatul Amalia, S.Pd., M.Pd.  
NIK: 201309 860821 2 001

Dosen Pembimbing 2



Adam Hendra Brata, S.Kom., M.T., M.Sc.  
NIP: 19900105 201903 1 009

Mengetahui  
Ketua Jurusan Teknik Informatika



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





## 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 pustaka.

Apabila ternyata didalam 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, 3 Juni 2020



Indah Ratna Sari

NIM: 175150209111029





## ABSTRAK

**Indah Ratna Sari, Pembangunan Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep *Crowdsourcing* Menggunakan *Geotagging*.**

**Pembimbing: Faizatul Amalia, S.Pd., M.Pd. dan Adam Hendra Brata, S.Kom., M.T., M.Sc.**

Parkir merupakan suatu kebutuhan pada masyarakat pemilik kendaraan, jika ingin memarkirkan kendaraannya di suatu tempat yang mudah dicapai. Selain itu, kawasan parkir juga perlu disediakan petugas parkir yang dapat mengarahkan kendaraan agar tertata rapi dan tidak mengganggu arus lalu lintas jalan umum demi keamanan dan kenyamanan. Keberadaan petugas parkir sekaligus dapat menjaga keamanan kendaraan yang diparkirkan di tempat resmi dari tindakan kriminal. Maka dari itu perlunya informasi laporan mendorong masyarakat untuk menyampaikan permasalahan parkir di Kota Malang. Karena laporan masyarakat pada Dinas Perhubungan terkait belum tersampaikan dengan baik sehingga membuat masyarakat kebingungan harus menyampaikan laporan parkir tidak resmi. Dengan perkembangan teknologi mobile yang kian pesat, pemanfaatan teknologi yang dapat digunakan oleh masyarakat untuk berbagi mengenai informasi parkir non resmi, adalah berbagi lokasi informasi laporan parkir tidak resmi dengan memanfaatkan informasi geospasial yang memanfaatkan fitur *geotagging*. Selain itu aplikasi yang akan dikembangkan dalam penelitian ini (*GoTindak*) memanfaatkan fitur *crowdsourcing* dimana informasi laporan parkir non resmi dapat langsung diperoleh dari pengguna lain. Secara umum pada aplikasi *GoTindak* menggunakan teknologi *geotagging* yang dapat membantu pengguna mengetahui lokasi pada saat melaporkan keluhan atau kritik pada parkir non resmi di wilayah Kota Malang. Informasi laporan yang terdapat pada aplikasi *GoTindak* berisi detail mengenai lokasi, waktu, status laporan, deskripsi dan bukti foto maupun gambar. Selain itu, fitur yang terdapat pada website yaitu admin dapat mengubah status laporan yang telah dikirimkan oleh pengguna. Di samping itu, terdapat 4 tahapan status laporan yaitu belum ditindak lanjuti, sedang proses, selesai, dan tidak disetujui. Teknik yang digunakan dalam penelitian ini adalah *prototyping*. Selanjutnya pengujian yang dilakukan adalah pengujian validasi, unit, pengujian *usability* yang menggunakan *SUS (System Usability Scale)* sebagai instrumen. Hasil dari pengujian fungsional adalah status valid serta aplikasi berjalan sesuai kebutuhan pengguna. Pengujian *usability* yang menghasilkan nilai 80,5 dimana nilai tersebut termasuk kategori sangat baik dan berhasil.

**Kata kunci:** *Android, prototyping, usability, parkir non resmi, crowdsourcing*



## ABSTRACT

**Indah Ratna Sari, Pembangunan Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep *Crowdsourcing* Menggunakan *Geotagging*.**

**Supervisors: Faizatul Amalia, S.Pd., M.Pd. dan Adam Hendra Brata, S.Kom., M.T., M.Sc.**

*Parking is a necessity for the owner of the vehicle if they want to park their vehicles somewhere easy to reach. Besides, the parking area also needs to be provided parking personnel who can steer the vehicle to be organized neatly and do not interfere with the flow of public road traffic for safety and comfort. The existence of the parking attendant can maintain the safety of vehicles that are parked in the official place of criminal activity.*

*Therefore, the need for the report *Infomrasi* encourages the public to convey parking problems in the city of Malang. Because the community reports on the related transportation Office has not been conveyed well to make the community confusion should submit an unofficial parking report. With the rapid development of mobile technology, the utilization of technology that can be used by the public to share about non-official parking information is to share the location of unofficial parking report information by utilizing geospatial information that makes up geotagging features. In addition to the applications to be developed, this research (*GoTindak*) utilizes crowdsourcing only where information on non-official parking reports can be directly obtained from other users. In general, the application uses geotagging technology that can help users know the location when reporting complaints or criticism of non-official parking in the area of Malang. The report information in the *Go follow* app contains details about the location, time, report status, description, and evidence of photos and images. Besides, the features of the website that are admins can change the status of the reports that users have submitted. Also, there are 4 stages of report status which are not followed up, is processing, finished, and not approved. The techniques used in this study are prototyping. Furthermore, testing is a validation test, a unit, a usability test that uses *SUS* (*System Usability Scale*) as an instrument. The result of functional testing is the valid status as well as the application running according to user needs. Usability testing that resulted in a value of 80.5 where the value belongs to the category is excellent and successful.*

**Keywords:** *Android, prototyping, usability, non official parking, crowdsourcing*





## DAFTAR ISI

PERSETUJUAN.....	ii
PERNYATAAN ORISINALITAS.....	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	5
BAB 2 LANDASAN KEPUSTAKAAN.....	12
2.1 Kajian Pustaka.....	12
2.2 DISHUB Kota Malang (Dinas Perhubungan).....	13
2.3 Parkir.....	14
2.4 <i>Crowdsourcing</i> .....	15
2.5 <i>Android</i> .....	15
2.6 <i>Java</i> .....	16
2.7 <i>Android Studio</i> .....	17
2.8 <i>Javascript Object Notation (JSON)</i> .....	17
2.9 <i>Geotagging</i> .....	18
2.10 Codeigniter.....	19
2.11 Rekayasa Perangkat Lunak.....	19
2.12 <i>Software Development Life Cycle</i> .....	20
2.13 Model Prototipe.....	21
2.14 UML.....	22
2.14.1 <i>Use case Diagram</i> .....	22





2.14.2 Use case Scenario .....	23
2.14.3 Sequence Diagram .....	24
2.14.4 Class Diagram .....	25
2.14.5 Entity Relationship Diagram .....	26
2.15 Pengujian Perangkat Lunak .....	27
2.15.1 Pengujian Unit .....	27
2.15.2 Pengujian Validasi .....	27
2.15.3 Pengujian Usability .....	28
2.15.4 Pengujian Compatibility .....	28
<b>BAB 3 METODOLOGI PENELITIAN .....</b>	<b>30</b>
3.1 Studi Literatur .....	30
3.2 Pengumpulan Data .....	31
3.3.1 Wawancara .....	31
3.2.1 Kuesioner .....	32
3.3 Analisis Kebutuhan .....	32
3.4 Perancangan Sistem .....	32
3.5 Implementasi <i>Prototyping</i> .....	33
3.6 Evaluasi Pengguna .....	33
3.7 Implementasi .....	34
3.8 Pengujian .....	34
3.8 Penarikan Kesimpulan dan Saran .....	35
<b>BAB 4 REKAYASA KEBUTUHAN .....</b>	<b>36</b>
4.1 Gambaran Umum Sistem .....	36
4.1.1 Deskripsi Sistem .....	36
4.1.2 Gambaran Sistem .....	36
4.2 Elisitasi Kebutuhan .....	37
4.3 Identifikasi Aktor .....	39
4.4 Analisis Kebutuhan Sistem .....	39
4.4.1 Kebutuhan Fungsional Iterasi Pertama .....	12
4.4.2 Kebutuhan Fungsional Iterasi Kedua .....	20
4.4.3 Kebutuhan Non Fungsional .....	21
4.5 Pemodelan Kebutuhan .....	22
4.5.1 Use case Diagram .....	22





4.5.2 Use case Scenario .....	27
4.6 Analisis Data .....	57
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>59</b>
5.1 Perancangan Sistem .....	59
5.1.1 Perancangan Sequence Diagram .....	59
5.1.2 Perancangan Class Diagram .....	63
5.1.3 Perancangan Basis Data .....	76
5.1.4 Perancangan Komponen .....	80
5.1.5 Perancangan Antarmuka .....	82
5.2 Implementasi Sistem .....	107
5.2.1 Spesifikasi Sistem .....	108
5.2.2 Batasan-batasan Implementasi .....	109
5.2.3 Implementasi Basis Data .....	109
5.2.4 Implementasi Kode Progam .....	110
5.2.5 Implementasi Antarmuka .....	115
<b>BAB 6 PENGUJIAN .....</b>	<b>122</b>
6.1 Pengujian Unit .....	122
6.1.1 Pengujian Unit <i>Method post()</i> .....	122
6.1.2 Pengujian Unit <i>Method onCreateView()</i> .....	124
6.1.3 Pengujian Unit <i>Method getAllProses()</i> .....	126
6.1.4 Pengujian Unit <i>Method ubah()</i> .....	128
6.2 Pengujian Validasi .....	129
6.2.1 Pengujian Validasi Sistem Berbasis <i>Mobile</i> .....	130
6.2.2 Pengujian Validasi Sistem Berbasis <i>Website</i> .....	139
6.3 Pengujian Usabilitas .....	148
6.3.1 Prosedur Pengujian .....	148
6.3.2 Hasil Pengujian .....	151
6.4 Pengujian <i>Compatibility</i> .....	152
6.5 Analisis Hasil Pengujian .....	153
6.5.1 Analisis Hasil Pengujian Unit .....	153
6.5.2 Analisis Hasil Pengujian Validasi .....	153
6.5.3 Analisis Hasil Pegujian Usabilitas .....	154
6.5.4 Analisis Hasil Pengujian <i>Compatibility</i> .....	154





BAB 7 PENUTUP .....	155
7.1 Kesimpulan .....	155
7.2 Saran .....	156
DAFTAR REFERENSI .....	158
LAMPIRAN A HASIL WAWANCARA DAN KUESIONER .....	161
A.1 Lampiran Hasil Wawancara Dengan Kepala Bidang Pengelolaan Perparkiran .....	165
A.2 Lampiran Lembar Validasi Kuesioner Pengguna (Pelapor) .....	168
A.3 Lampiran Hasil Kuesioner Pengguna (Pelapor) .....	178
LAMPIRAN B EVALUASI PENGGUNA TERHADAP <i>PROTOTYPE</i> .....	182
B.1 Lampiran Lembar Evaluasi Pengguna (Pelapor) .....	182
LAMPIRAN C DOKUMEN PENDUKUNG .....	191
C.1 Kartu Identitas Petugas Parkir .....	191
C.2 Bukti Fisik Karcis Parkir Resmi .....	192
C.3 Surat Permohonan Data Skripsi .....	194
LAMPIRAN D PENGUJIAN .....	196
D.1 Kuesioner Pengujian <i>Usability</i> Pengguna (Pelapor) .....	196
D.2 Kuesioner Pengujian <i>Usability</i> Administrator .....	206
LAMPIRAN E DOKUMENTASI PENELITIAN .....	210
E.1 Dokumentasi Wawancara .....	210
E.2 Dokumentasi Validasi Hasil Wawancara .....	210
E.3 Dokumentasi Pengujian .....	211
E.4 Dokumentasi Pendukung .....	212





## DAFTAR GAMBAR

Gambar 2.1 Bagan Struktur Organisasi Dinas Pehubungan Kota Malang .....	14
Gambar 2.2 <i>JSON</i> dengan <i>object</i> .....	17
Gambar 2.3 <i>JSON</i> dengan <i>Array</i> .....	18
Gambar 2.4 <i>JSON</i> dengan <i>Value</i> .....	18
Gambar 2.5 Model Prototipe .....	21
Gambar 2.6 Item <i>System Usability Scale</i> (SUS) .....	28
Gambar 3.1 Diagram Alir Metodologi Penelitian .....	30
Gambar 4.1 Cara Kerja Sistem Aplikasi GoTindak .....	37
Gambar 4.2 Pengajuan Pelaporan Dinas Perhubungan Kota Malang .....	38
Gambar 4.3 Usulan Sistem yang akan dibuat .....	40
Gambar 4.4 Kode Kebutuhan .....	41
Gambar 4.5 <i>Use case Diagram</i> Pelapor Iterasi Pertama .....	23
Gambar 4.6 <i>Use case Diagram</i> Pelapor Iterasi Kedua .....	24
Gambar 4.7 <i>Use case diagram</i> Administrator Iterasi Pertama .....	25
Gambar 4.8 <i>Use case Diagram</i> Administrator Iterasi Kedua .....	26
Gambar 5.1 Sequence diagram Membuat Laporan .....	60
Gambar 5.2 Sequence Diagram Melihat Laporan .....	61
Gambar 5.3 Sequence Diagram Melihat Laporan Sedang Proses .....	62
Gambar 5.4 Sequence Diagram Mengupload Foto Sedang Proese .....	63
Gambar 5.5 Class Diagram Pelapor Iterasi Pertama .....	64
Gambar 5.6 <i>Class Diagram Package Activity</i> Iterasi Pertama .....	65
Gambar 5.7 <i>Class Diagram Package Fragment</i> Iterasi Pertama .....	66
Gambar 5.8 <i>Class Diagram Package Adapter</i> Iterasi Pertama .....	67
Gambar 5.9 <i>Class Diagram Package Model</i> Iterasi Pertama .....	68
Gambar 5.10 Class Diagram Package Service Iterasi Pertama .....	69
Gambar 5.11 Class Diagram Pelapor Iterasi Kedua .....	70
Gambar 5.12 Class Diagram Package Activity Iterasi Kedua .....	71
Gambar 5.13 Class Diagram Package Fragment Iterasi Kedua .....	72
Gambar 5.14 <i>Class Diagram Package Adapter</i> Iterasi Kedua .....	73





Gambar 5.15 <i>Class Diagram Package Model</i> Iterasi Kedua.....	74
Gambar 5.16 <i>Class Diagram Package Service</i> Iterasi Kedua .....	74
Gambar 5.17 <i>Class Diagram Administrator</i> Iterasi Pertama.....	75
Gambar 5.18 <i>Class Diagram Administrator</i> Iterasi Kedua .....	76
Gambar 5.19 Perancangan <i>Entity Relationship Diagram</i> Iterasi Pertama.....	77
Gambar 5.20 Perancangan <i>Entity Relationship Diagram</i> Iterasi Kedua.....	79
Gambar 5.21 <i>Screenflow</i> aplikasi sisi pengguna .....	83
Gambar 5.22 Halaman Login.....	84
Gambar 5.23 Halaman Register.....	85
Gambar 5.24 <i>Screenflow</i> aplikasi sisi pelapor.....	86
Gambar 5.25 Halaman Beranda.....	87
Gambar 5.26 Halaman Detail Laporan.....	88
Gambar 5.27 Halaman Tambah Laporan.....	89
Gambar 5.28 Halaman Riwayat Laporan .....	90
Gambar 5.29 Halaman Profil Pengguna.....	91
Gambar 5.30 <i>Screenflow</i> aplikasi sisi administrator .....	92
Gambar 5.31 Halaman Login Website .....	93
Gambar 5.32 Halaman Dashboard Website.....	94
Gambar 5.33 Halaman Laporan Belum Ditindaklanjuti.....	95
Gambar 5.34 Halaman Laporan Sedang Proses.....	97
Gambar 5.35 Halaman Edit Status Laporan.....	98
Gambar 5.36 Halaman Melihat Artikel Identifikasi Parkir.....	99
Gambar 5.37 Halaman Melihat Detail Artikel Identifikasi Parkir.....	100
Gambar 5.38 Halaman Detail Laporan.....	101
Gambar 5.39 Halaman Laporan.....	102
Gambar 5.40 Halaman Beranda.....	103
Gambar 5.41 Halaman Riwayat Laporan.....	104
Gambar 5.42 Halaman Profil.....	105
Gambar 5.43 Halaman Edit Status Laporan Website.....	107
Gambar 5.44 Implementasi Basis Data.....	110





Gambar 5.45 Implementasi Antarmuka Halaman Login..... 115

Gambar 5.46 Implementasi Antarmuka Halaman Register Akun..... 116

Gambar 5.47 Implementasi Antarmuka Halaman Daftar Laporan..... 116

Gambar 5.48 Implementasi Antarmuka Halaman Detail Laporan..... 117

Gambar 5.49 Implementasi Antarmuka Halaman Laporan Pelapor..... 117

Gambar 5.50 Implementasi Antarmuka Halaman Profil Pengguna..... 118

Gambar 5.51 Implementasi Antarmuka Website Halaman Login ..... 118

Gambar 5.52 Implementasi Antarmuka Website Halaman Dashboard ..... 119

Gambar 5.53 Implementasi Antarmuka Website Halaman Status Belum Ditinjau ..... 119

Gambar 5.54 ..... 119

Gambar 5.55 Implementasi Antarmuka Website Halaman Status Selesai ..... 120

Gambar 5.56 Implementasi Antarmuka Website Halaman Status Tidak Disetujui ..... 121

Gambar 5.57 Implementasi Antarmuka Website Halaman Detail Laporan..... 121

Gambar 6.1 *Flow Graph Method post()* ..... 123

Gambar 6.2 *Flow Graph Method onCreateView()* ..... 125

Gambar 6.3 *Flow Graph Method getAllProses()* ..... 127

Gambar 6.4 Algoritme Method ubah() ..... 128

Gambar 6.6 Hasil Pengujian Compatibilty dengan Tool SortSite..... 153





## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	12
Tabel 2.2 Kelebihan dan Kekurangan SDLC Prototipe .....	22
Tabel 2.3 Komponen pada <i>Use case Scenario</i> .....	24
Tabel 2.4 Elemen pada <i>Sequence Diagram</i> .....	24
Tabel 2.5 Elemen pada <i>Class Diagram</i> .....	26
Tabel 2.6 Elemen pada <i>Entity Relationship Diagram</i> .....	27
Tabel 4.1 Identifikasi Aktor .....	39
Tabel 4.2 Kebutuhan Fungsional Pengguna Iterasi Pertama .....	12
Tabel 4.3 Kebutuhan Fungsional Administrator Iterasi Pertama .....	16
Tabel 4.4 Kebutuhan Fungsional Pelapor Iterasi Kedua .....	20
Tabel 4.5 Kebutuhan Fungsional Administrator Iterasi Kedua .....	21
Tabel 4.6 Kebutuhan Non Fungsional .....	21
Tabel 4.7 <i>Use case scenario</i> Register iterasi pertama .....	27
Tabel 4.8 <i>Use case scenario Login</i> iterasi pertama .....	28
Tabel 4.9 <i>Use case scenario</i> Membuat Laporan Parkir Iterasi Pertama .....	29
Tabel 4.10 <i>Use case scenario</i> Daftar Laporan Iterasi Pertama .....	30
Tabel 4.11 <i>Use scenario</i> Melihat Detail Laporan Iterasi Pertama .....	31
Tabel 4.12 <i>Use case scenario</i> Memberi Dukungan Laporan Iterasi Pertama .....	31
Tabel 4.13 <i>Use case scenario</i> Melihat Riwayat Laporan Iterasi Pertama .....	32
Tabel 4.14 <i>Use case scenario</i> Melihat Profil Iterasi Pertama .....	33
Tabel 4.15 <i>Use case scenario</i> Mengedit Profil Iterasi Pertama .....	33
Tabel 4.16 <i>Use case scenario</i> Mengupload Foto Profil Iterasi Pertama .....	34
Tabel 4.17 <i>Use case scenario</i> Mengedit <i>Password</i> Iterasi Pertama .....	35
Tabel 4.18 <i>Use case scenario</i> Menambah Komentar Iterasi Pertama .....	36
Tabel 4.19 <i>Use case scenario</i> Melihat Komentar Iterasi Pertama .....	37
Tabel 4.20 <i>Use case scenario Logout</i> Iterasi Pertama .....	37
Tabel 4.21 <i>Use case scenario</i> Memilih Fitur <i>Camera</i> Iterasi Pertama .....	38
Tabel 4.22 <i>Use case scenario</i> Memilih Fitur <i>Gallery</i> .....	39
Tabel 4.23 <i>Use case scenario</i> Melihat Artikel Identifikasi parkir Iterasi Kedua .....	40





Tabel 4.24 <i>Use case scenario</i> Melihat Detail Indentifikasi Parkir .....	41
Tabel 4.25 <i>Use case scenario</i> Login Iterasi Pertama.....	41
Tabel 4.26 <i>Use case scenario</i> Melihat Jumlah Laporan Iterasi Pertama.....	42
Tabel 4.27 <i>Use case scenario</i> Melihat Laporan Belum Ditindaklanjuti Iterasi Pertama.....	43
Tabel 4.28 <i>Use case scenario</i> Melihat Detail Laporan Belum Ditindaklanjuti Iterasi Pertama.....	44
Tabel 4.29 <i>Use case scenario</i> Mengubah Status Laporan Belum Ditindaklanjuti Iterasi Pertama.....	45
Tabel 4.30 <i>Use case scenario</i> Mencari Status Laporan Belum Ditindaklanjuti Iterasi Pertama.....	46
Tabel 4.31 <i>Use case scenario</i> Melihat Laporan Sedang Proses Iterasi Pertama... ..	47
Tabel 4.32 <i>Use case scenario</i> Melihat Detail Laporan Sedang Proses Iterasi Pertama.....	47
Tabel 4.33 <i>Use case scenario</i> Mengubah Status Laporan Sedang Proses Iterasi Pertama.....	48
Tabel 4.34 <i>Use case scenario</i> Mencari Status Laporan Sedang Proses.....	50
Tabel 4.35 <i>Use case scenario</i> Melihat Laporan Selesai Iterasi Pertama.....	50
Tabel 4.36 <i>Use case scenario</i> Melihat Detail Laporan Selesai Iterasi Pertama ....	51
Tabel 4.37 <i>Use case scenario</i> Mencari Laporan Selesai Iterasi Pertama.....	52
Tabel 4.38 <i>Use case scenario</i> Melihat Laporan Tidak disetujui Iterasi Pertama ..	53
Tabel 4.39 <i>Use case scenario</i> Mencari Laporan Tidak disetujui Iterasi Pertama..	54
Tabel 4.40 <i>Use case scenario</i> Mengubah Status Laporan Tidak disetujui Iterasi Pertama.....	55
Tabel 4.41 <i>Use case scenario</i> Logout Iterasi Pertama .....	55
Tabel 4.42 <i>Use case scenario</i> Mengupload Foto Sedang Proses Iterasi Kedua ....	56
Tabel 5.1 Rancangan Tabel Entitas <i>Users</i> .....	77
Tabel 5.2 Rancangan Tabel Entitas Laporan .....	78
Tabel 5.3 Rancangan Tabel Entitas Komentar .....	78
Tabel 5.4 Rancangan Tabel Entitas Like.....	79





Tabel 5.5 Rancangan Tabel Entitas Laporan .....	79
Tabel 5.6 Algoritme Membuat Laporan Parkir .....	80
Tabel 5.7 Algoritme Melihat Daftar Laporan .....	81
Tabel 5.8 Algoritme Melihat Laporan Sedang Proses .....	82
Tabel 5.9 Algoritme Mengupload Foto Sedang Proses .....	82
Tabel 5.10 Penjelasan Antarmuka Halaman <i>Login</i> .....	84
Tabel 5.11 Penjelasan Antarmuka Halaman Register .....	85
Tabel 5.12 Penjelasan Antarmuka Halaman Beranda .....	87
Tabel 5.13 Penjelasan Antarmuka Halaman Detail Laporan .....	89
Tabel 5.14 Penjelasan Antarmuka Halaman Tambah Laporan .....	90
Tabel 5.15 Penjelasan Antarmuka Halaman Riwayat Laporan .....	90
Tabel 5.16 Penjelasan Antarmuka Halaman Profil Pengguna .....	91
Tabel 5.17 Penjelasan Antarmuka Halaman <i>Login Website</i> .....	93
Tabel 5.18 Penjelasan Antarmuka Halaman <i>Dashboard</i> .....	94
Tabel 5.19 Penjelasan Antarmuka Halaman Laporan Belum Ditindaklanjuti .....	96
Tabel 5.20 Penjelasan Antarmuka Halaman Laporan Sedang Proses .....	97
Tabel 5.21 Penjelasan Antarmuka Halaman Edit Status Laporan .....	99
Tabel 5.22 Penjelasan Antarmuka Halaman Melihat Artikel Identifikasi Parkir .....	100
Tabel 5.23 Penjelasan Antarmuka Halaman Melihat Detail Artikel Identifikasi Parkir .....	101
Tabel 5.24 Penjelasan Antarmuka Halaman Detail Laporan .....	102
Tabel 5.25 Penjelasan Antarmuka Halaman Tambah Laporan .....	103
Tabel 5.26 Penjelasan Antarmuka Halaman Beranda .....	104
Tabel 5.27 Penjelasan Antarmuka Halaman Riwayat Laporan .....	105
Tabel 5.28 Penjelasan Antarmuka Halaman Profil Pengguna .....	106
Tabel 5.29 Penjelasan Antarmuka Halaman Edit Status Laporan .....	107
Tabel 5.30 Spesifikasi Perangkat Laptop .....	108
Tabel 5.31 Spesifikasi Perangkat Smartphone <i>Android</i> .....	108
Tabel 5.32 Spesifikasi Perangkat Lunak .....	109
Tabel 5.33 Kode Program Membuat Laporan .....	110





Tabel 5.34 Penjelasan Kode Program Membuat Laporan .....	111
Tabel 5.35 Kode Program Melihat Laporan .....	112
Tabel 5.36 Penjelasan Kode Program Melihat Laporan .....	113
Tabel 5.37 Kode Program Laporan Sedang Proses .....	113
Tabel 5.38 Penjelasan Kode Program Laporan Sedang Proses .....	114
Tabel 5.39 Kode Program Mengupload Foto Sedang Proses .....	114
Tabel 5.40 Penjelasan Kode Program Mengupload Foto Sedang Proses .....	115
Tabel 6.1 Algoritme <i>Method post()</i> .....	122
Tabel 6.2 Kasus Uji Pengujian Unit <i>Method post()</i> .....	124
Tabel 6.3 Algoritme <i>Method onCreateView()</i> .....	125
Tabel 6.4 Kasus Uji Pengujian Unit <i>Method onCreateView()</i> .....	126
Tabel 6.5 Algoritme <i>Method getAllProses()</i> .....	126
Tabel 6.6 Kasus Uji Pengujian Unit <i>Method getAllProses()</i> .....	127
Tabel 6.7 Algoritme <i>Method ubah()</i> .....	128
Tabel 6.8 Kasus Uji Pengujian Unit <i>Method getAllProses()</i> .....	129
Tabel 6.9 Kasus Uji Pengujian Validasi Akun <i>Email</i> Terdaftar .....	130
Tabel 6.10 Kasus Uji Pengujian Validasi Akun Belum <i>Email</i> Terdaftar .....	130
Tabel 6.11 Kasus Uji Pengujian Validasi <i>Email</i> atau <i>Password</i> Tidak Terisi .....	131
Tabel 6.12 Kasus Uji Pengujian Validasi Register Akun Berhasil .....	131
Tabel 6.13 Kasus Uji Pengujian Validasi <i>Form</i> Register Tidak Terisi Lengkap .....	132
Tabel 6.14 Kasus Uji Pengujian Validasi Membuat Laporan Parkir .....	132
Tabel 6.15 Kasus Uji Pengujian Validasi Melihat Laporan .....	133
Tabel 6.16 Kasus Uji Pengujian Validasi Melihat Detail Laporan .....	133
Tabel 6.17 Kasus Uji Pengujian Validasi Memberi Dukungan Laporan .....	134
Tabel 6.18 Kasus Uji Pengujian Validasi Melihat Riwayat Laporan .....	134
Tabel 6.19 Kasus Uji Pengujian Validasi Melihat Profil .....	135
Tabel 6.20 Kasus Uji Pengujian Validasi Mengedit Profil .....	135
Tabel 6.21 Kasus Uji Pengujian Validasi Mengupload Foto Profil .....	135
Tabel 6.22 Kasus Uji Pengujian Validasi Mengedit <i>Password</i> .....	136
Tabel 6.23 Kasus Uji Pengujian Validasi Menambah Komentar .....	136





Tabel 6.24 Kasus Uji Pengujian Validasi Melihat Komentar .....	137
Tabel 6.25 Kasus Uji Pengujian Validasi <i>Logout</i> .....	137
Tabel 6.26 Kasus Uji Pengujian Memilih Fitur <i>Camera</i> .....	138
Tabel 6.27 Kasus Uji Pengujian Memilih Fitur <i>Gallery</i> .....	138
Tabel 6.28 Kasus Uji Pengujian Validasi Melihat Artikel Identifikasi Parkir .....	138
Tabel 6.29 Kasus Uji Pengujian Validasi Melihat Detail Artikel Identifikasi Parkir .....	139
Tabel 6.30 Kasus Uji Pengujian Validasi Akun <i>Email</i> Terdaftar .....	139
Tabel 6.31 Kasus Uji Pengujian Validasi Akun Belum <i>Email</i> Terdaftar .....	140
Tabel 6.32 Kasus Uji Pengujian Validasi <i>Email</i> atau <i>Password</i> Tidak Terisi .....	140
Tabel 6.33 Kasus Uji Pengujian Validasi Melihat Jumlah Laporan .....	141
Tabel 6.34 Kasus Uji Pengujian Validasi Melihat Laporan Belum Ditindaklanjuti .....	141
Tabel 6.35 Kasus Uji Pengujian Validasi Melihat Detail Laporan Belum Ditindaklanjuti .....	142
Tabel 6.36 Kasus Uji Pengujian Validasi Mengubah Laporan Belum Ditindaklanjuti .....	142
Tabel 6.37 Kasus Uji Pengujian Validasi Mencari Status Laporan Belum Ditindaklanjuti .....	143
Tabel 6.38 Kasus Uji Pengujian Validasi Melihat Laporan Sedang Proses .....	143
Tabel 6.39 Kasus Uji Pengujian Validasi Melihat Detail Laporan Sedang Proses .....	143
Tabel 6.40 Kasus Uji Pengujian Validasi Mengubah Status Laporan Sedang Proses .....	144
Tabel 6.41 Kasus Uji Pengujian Validasi Mencari Status Laporan Sedang Proses .....	144
Tabel 6.42 Kasus Uji Pengujian Validasi Melihat Laporan Selesai .....	145
Tabel 6.43 Kasus Uji Pengujian Validasi Melihat Detail Laporan Selesai .....	145
Tabel 6.44 Kasus Uji Pengujian Validasi Mencari Status Laporan Selesai .....	146
Tabel 6.45 Kasus Uji Pengujian Validasi Melihat Laporan Tidak disetujui .....	146
Tabel 6.46 Kasus Uji Pengujian Validasi Mencari Laporan Tidak disetujui .....	147





Tabel 6.47 Kasus Uji Pengujian Validasi Mengubah Status Laporan Tidak disetujui

147

Tabel 6.48 Kasus Uji Pengujian Validasi *Logout*

148

Tabel 6.49 Kasus Uji Pengujian Validasi Mengupload Foto Sedang Proses

148

Tabel 6.50 Parameter Bobot Skor Usabilitas (SUS)

149

Tabel 6.51 Daftar Pernyataan Kuesioner Pengujian *Usability* Pelapor

149

Tabel 6.52 Daftar Pernyataan Kuesioner Pengujian *Usability* Admin

150

Tabel 6.53 Data Hasil Pengujian Usabilitas Aplikasi *Mobile* Pelapor

151

Tabel 6.54 Data Hasil Pengujian Usabilitas Website Administrator

151

Tabel 6.55 Data Hasil Normalisasi dan Skor SUS Aplikasi *Mobile* Pelapor

152

Tabel 6.56 Data Hasil Normalisasi dan Skor SUS Website Administrator

152





## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Kota Malang mengalami peningkatan jumlah kendaraan roda dua maupun roda empat setiap tahunnya. Berdasarkan data dari Badan Pusat Statistik Kota Malang jumlah kendaraan bermotor tahun 2016-2017 di Kota Malang mencapai 571.337 unit dengan jumlah kendaraan roda empat 95.320 unit dan kendaraan bermotor roda dua 476.017 unit (Statistik, 2018). Jumlah kendaraan bermotor ini belum termasuk jumlah dari kendaraan pendatang dari luar Kota Malang untuk menempuh pendidikan ataupun hanya sekedar berlibur di Kota Malang. Tingginya jumlah kendaraan bermotor mengakibatkan tingginya arus kendaraan pada ruas jalan raya di Kota Malang dan meningkatnya kebutuhan akan lahan parkir. Dengan jumlah kendaraan yang semakin meningkat maka dibutuhkan lahan parkir resmi dari Dinas Perhubungan Kota Malang supaya tidak terjadi parkir non resmi yang dapat menyebabkan kemacetan pada ruas jalan raya, tindakan yang tidak diinginkan, dan tarif parkir yang tidak sesuai dengan harga tarif parkir resmi.

Berdasarkan data dari hasil wawancara pada Dinas Perhubungan Kota Malang mengenai penetapan lokasi tempat parkir, sepanjang tahun 2018 terdapat kurang lebih 765 titik lokasi dan posisi parkir di Kota Malang. Kondisi ini tentunya menuntut pemerintah daerah selaku pelayanan publik, untuk dapat memberikan fasilitas sarana dan prasarana dalam pengaturan arus kendaraan serta memberikan jasa pelayanan parkir yang memadai bagi warga Malang. Kawasan parkir tidak serta merta dapat disediakan di sembarang tempat, namun perlu kajian mendalam agar dampak dari kawasan parkir tersebut tidak mengganggu ketertiban umum. Selain itu, kawasan parkir juga perlu disediakan petugas parkir yang dapat mengarahkan kendaraan agar tertata rapi dan tidak mengganggu arus lalu lintas jalan umum demi keamanan dan kenyamanan. Keberadaan petugas parkir sekaligus dapat menjaga keamanan kendaraan yang diparkirkan di tempat resmi dari tindakan kriminal.

*Crowdsourcing* merupakan suatu metode yang dilakukan oleh suatu kelompok atau individu, berkabolarasi untuk melakukan suatu tugas; misalnya berbagi informasi maupun mengerjakan sesuatu yang telah ditentukan (Estellés-arolas dan González-ladrón-de-guevara, 2012). Saat ini, informasi laporan dari masyarakat yang disampaikan hanya mencakup daerah tertentu dan tidak adanya perhimpunan data (*crowdsourcing*). Konsep *crowdsourcing* dapat mendorong masyarakat untuk menyampaikan permasalahan parkir di Kota Malang. Karena laporan masyarakat pada Dinas Perhubungan terkait belum tersampaikan dengan baik sehingga membuat masyarakat kebingungan harus menyampaikan laporan parkir tidak resmi. Sedangkan untuk melaporkan masalah parkir tidak resmi, Dinas Perhubungan Kota Malang hanya menggunakan layanan melalui pesan berupa SMS aduan Kota Malang di *hotline* yang tersedia di *website* resmi dinas perhubungan Kota Malang untuk mempermudah penyampaian laporan, meskipun telah disediakan sarana tersebut, hal ini masih kurang efektif karena informasi yang tersedia kurang terkoordinasi dengan baik dan dari





banyaknya laporan tidak bisa semua ditindaklanjuti karena kurangnya informasi yang tidak lengkap dan tidak terpercaya.

Terdapat sebuah penelitian mengenai modul perancangan sistem *e-complaint* berbasis *geotagging* dan *geofencing* yang berfokus pada pelayanan publik dengan memberikan informasi atau keluhan kepada pemerintah dalam bentuk sistem *e-complaint*. Sistem *e-complaint* yang diinstal pada *smartphone* yang dilengkapi dengan teknik *geotagging* untuk menentukan secara tepat di mana lokasi keluhan dibuat dan metode *geofencing* untuk menentukan wilayah mana keluhan tersebut berada. Dalam penelitian ini, peneliti menjelaskan proses merancang modul untuk sistem *e-complaint* berbasis *geotagging* dan *geofencing*. Dengan akurasi 94%, *geofencing* adalah teknik yang cocok untuk diterapkan. Selain itu, teknik *geotagging* menggunakan EXIF dari gambar digital dianggap teknik yang paling cocok untuk sistem *e-complaint* daripada teknik *geotagging* lainnya. Penelitian ini dilakukan pada tahun 2017 yang berjudul “*Designing Module e-complaint System Based on Geotagging and Geofencing*” (Puspa, et al., 2017). *Geotagging* merupakan suatu prosedur untuk menambahkan informasi metadata geografis pada media yang berbeda seperti menambahkan informasi geografis pada gambar, audiovisual, situs internet, pesan SMS, QR code, maupun RSS berupa metadata geospasial. Umumnya informasi yang ditambahkan terdiri atas koordinat geografis *GPS*, lokasi tempat atau alamat, deskripsi informasi dari tempat tersebut, dan penanda tempat.

Penelitian ini dilakukan pada sebuah aplikasi berbasis *mobile Android* pada sebuah *smartphone*. Teknologi perangkat bergerak *smartphone* menjadi kebutuhan yang sudah sangat erat dalam kehidupan sehari-hari. *Smartphone* sebagai media komunikasi yang memiliki berbagai macam fitur, juga fleksibel dapat dibawa ke mana saja dan kapan saja serta memiliki *user friendly interface*. Pengguna *smartphone* di Indonesia yang menggunakan operating system *Android* mengalami peningkatan sebesar 90% yang dibuktikan dari data *Mobile Operating System Market*. Berdasarkan data tersebut, aplikasi *mobile crowdsourcing* pelaporan parkir non resmi yang dikembangkan akan menggunakan platform *Android*. Hal itu juga membuat penyebaran informasi pada masyarakat semakin mudah dengan adanya internet. Aktivitas penyebaran informasi per tahun ini mengalami peningkatan yang signifikan. Berdasarkan survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) tahun 2017, pengguna internet di Indonesia meningkat menjadi 143,26 juta jiwa atau setara 54,7% dari total populasi penduduk di Indonesia (APJII, 2018). Masyarakat sudah terbiasa menggunakan *smartphone* untuk membantu aktivitas sehari-hari. Aplikasi merupakan sebuah perangkat lunak yang dipasang pada sebuah perangkat *smartphone* yang sudah banyak digunakan di dunia digital seperti sekarang ini. *Smartphone* dapat membantu dalam menyelesaikan berbagai bidang permasalahan dengan *software* yang ada sesuai dengan kebutuhannya.

Dalam penelitian ini akan dikembangkan sebuah aplikasi dengan nama *GoTindak*. *Gotindak* merupakan merupakan sebuah aplikasi *mobile social crowdsourcing* yang akan dikembangkan menggunakan platform *android*. Di samping itu, aplikasi *GoTindak*





bertujuan sebagai wadah berbagi informasi mengenai pelaporan terkait parkir non resmi. Aplikasi GoTindak ditujukan kepada seluruh elemen masyarakat khususnya di Kota Malang untuk dapat membagikan informasi berupa keluhan atau kritik mengenai pelaporan parkir non resmi di wilayah sekitar. Oleh karena itu, aplikasi GoTindak bekerjasama dengan Dinas Perhubungan Kota Malang pada Manajemen Bidang Perparkiran agar informasi laporan dapat langsung tersampaikan pada pihak terkait, yang nantinya akan ditindaklanjuti. Aplikasi GoTindak terdiri dari 2 aplikasi yang saling terhubung. Aplikasi GoTindak untuk pengguna berbasis *mobile* dan *website* GoTindak untuk admin pada Manajemen Bidang Perparkiran. Secara umum pada aplikasi GoTindak menggunakan teknologi *geotagging* yang dapat membantu pengguna mengetahui lokasi pada saat melaporkan keluhan atau kritik pada parkir non resmi di wilayah Kota Malang. Informasi laporan yang terdapat pada aplikasi GoTindak berisi detail mengenai lokasi, waktu, status laporan, deskripsi dan bukti foto maupun gambar. Selain itu, fitur yang terdapat pada *website* yaitu admin dapat mengubah status laporan yang telah dikirimkan oleh pengguna. Di samping itu, terdapat 4 tahapan status laporan yaitu belum ditindak lanjuti, sedang proses, selesai, dan tidak disetujui. Berdasarkan latar belakang diatas maka penulis mengangkat skripsi dengan judul “Pembangunan Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep *Crowdsourcing* Menggunakan *Geotagging*”.

## 1.2 Rumusan Masalah

Rumusan masalah yang akan dipaparkan dalam penelitian ini yaitu:

1. Bagaimana hasil analisis kebutuhan pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*?
2. Bagaimana hasil rancangan dan implementasi pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*?
3. Bagaimana hasil pengujian fungsional pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*?
4. Bagaimana hasil pengujian usability pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*?

## 1.3 Tujuan Penelitian

Adapun tujuan adanya penelitian ini yaitu:

1. Menganalisis kebutuhan pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*.
2. Mengimplementasikan sebuah aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* dengan teknologi *geotagging* yang dapat memudahkan masyarakat dalam melaporkan adanya parkir non resmi.
3. Mengetahui hasil pengujian fungsional pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*.





4. Mampu mengetahui tingkat usability pada aplikasi *mobile* identifikasi parkir non resmi konsep *crowdsourcing* menggunakan *geotagging* sehingga pengguna dapat mengoperasikan sistem dengan mudah.

#### 1.4 Manfaat

Manfaat yang diinginkan melalui tugas akhir berikut, ditinjau dari banyaknya sisi pengguna sebagai berikut:

##### A. Manfaat Bagi Penulis

Penelitian ini diharapkan dapat menambah pengalaman dan pemahaman sebuah informasi dalam menerapkan ilmu yang didapat selama perkuliahan dan dapat memberikan wawasan ilmu pengetahuan.

##### B. Manfaat Bagi Masyarakat

1. Penelitian ini memudahkan masyarakat dalam menyampaikan informasi laporan kepada DISHUB Kota Malang mengenai permasalahan parkir yang tidak resmi.
2. Tersedianya media informasi bagi masyarakat mengenai permasalahan identifikasi parkir non resmi dengan aplikasi *mobile*.
3. Masyarakat dapat berpartisipasi dalam menyampaikan laporan permasalahan mengenai parkir tidak resmi melalui aplikasi *mobile*.

##### C. Manfaat Bagi Dinas Perhubungan Kota Malang

1. Mempermudah DISHUB Kota Malang untuk mendapatkan informasi secara cepat dan akurat dalam laporan yang disampaikan oleh masyarakat mengenai parkir tidak resmi.
2. Mempermudah DISHUB dalam menindaklanjuti laporan masyarakat mengenai parkir yang tidak resmi.
3. Mempermudah DISHUB Kota Malang untuk mengetahui lokasi yang tepat sesuai laporan masyarakat mengenai parkir yang tidak resmi.

#### 1.5 Batasan Masalah

Adapun batasan masalah dalam melakukan penelitian ini sebagai berikut:

1. Perancangan serta implementasi dalam membangun aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* hanya dapat berjalan pada sistem operasi *Android* dengan API 16 (*Jelly Bean*).
2. Objek dalam penelitian ini ditujukan pada permasalahan parkir di Kota Malang.
3. Aplikasi dapat berjalan ketika terhubung dengan internet.
4. Penelitian yang dilakukan penulis pada aplikasi *mobile* identifikasi parkir non resmi hanya sebagai media informasi mengenai laporan masyarakat terkait identifikasi parkir non resmi.





5. Sistem identifikasi parkir non resmi untuk admin berbasis *website* dan dapat digunakan pada *browser*.
6. Penelitian yang dilakukan oleh penulis dalam mengimplementasikan aplikasi hanya sebatas uji coba, tidak sampai tahap komersialisasi.

## 1.6 Sistematika Penulisan

Keterangan secara merata dari semua isi yang ditulis pada bagian per-babnya terdiri:

### BAB 1 PENDAHULUAN

Pada bab ini memaparkan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat, batasan masalah, serta sistematika penulisan dalam membangun aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging*.

### BAB 2 LANDASAN KEPUSTAKAAN

Bab ini mendeskripsikan mengenai kajian pustaka serta landasan teori sebagai referensi dalam melakukan penelitian terkait aplikasi *mobile* identifikasi parkir non resmi. Dalam hal ini, landasan kepustakaan sangat penting dalam hal membantu peneliti untuk menunjang penulisan laporan penelitian.

### BAB 3 METODOLOGI PENELITIAN

Pada bab ini membahas mengenai metode penelitian mengenai alur kerja dalam membangun sistem dan juga digunakan dalam membantu peneliti untuk alur penulisan dokumen agar penelitian ini dapat terarah dengan baik.

### BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan dibahas tentang rekayasa kebutuhan yang didapatkan pada proses penggalian kebutuhan saat melakukan wawancara dan menyebarkan kuesioner. Dalam mendefinisikan kebutuhan yang dilakukan dengan menggambarkan deskripsi sistem, melakukan pemodelan kebutuhan dengan *use case diagram*, dan menjelaskan *main flow* menggunakan *use case scenario*.

### BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan mengenai proses perancangan pada sistem sesuai dengan analisis kebutuhan yang telah didefinisikan sebelumnya. Setelah selesai melakukan perancangan, selanjutnya melakukan implementasi pada sistem berdasarkan proses perancangan yang dihasilkan.

### BAB 6 PENGUJIAN

Pada bab ini membahas mengenai proses pengujian sistem dan analisis hasil dari pembangunan sistem yang dilakukan. Tahapan pengujian bertujuan untuk mengetahui setiap fungsi yang telah diimplementasikan sudah berjalan dengan baik.

### BAB 7 PENUTUP





Bab ini menjelaskan kesimpulan dan saran dari penelitian yang telah dilakukan. Kesimpulan menjawab dari rumusan masalah dalam penelitian yang telah didefinisikan. Selanjutnya saran berisi penjelasan mengenai pengembangan lebih lanjut pada penelitian berikutnya.





## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Bab ini peneliti akan menggali mengenai landasan pustaka yang terkait dengan judul untuk memperoleh landasan ilmiah, serta tinjauan pustaka dari penelitian sebelumnya yang telah dilakukan sebagai bahan perbandingan, mengenai kelebihan dan kekurangan yang mana sebagai rujukan untuk dilaksanakannya penelitian ini dalam membangun Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep *Crowdsourcing* Menggunakan *Geotagging*.

Tabel 2.1 Kajian Pustaka

No	Judul	Penulis	Fokus Penelitian
1	<i>Designing Module e-complaint System Based on Geotagging and Geofencing</i>	(Puspa, et al., 2017)	Penelitian ini berfokus membahas perancangan sistem <i>e-complaint</i> menggunakan teknik <i>geotagging</i> sehingga dapat menentukan lokasi pengaduan secara tepat
2.	Penerapan Konsep <i>Geotagging</i> Pada Aplikasi Tanggap Darurat Bencana Berbasis <i>Android</i>	(Nunu, et al., 2019)	Penelitian ini berfokus pada penyampaian informasi bencana di wilayah Majalengka berbasis <i>Android</i> dengan menggunakan teknik <i>geotagging</i>
3	Aplikasi Rekomendasi Spot Area Wisata Berbasis <i>Android</i> dengan Teknik <i>Geotag</i>	(Mochamad, et al., 2017)	Penelitian ini berfokus pada penyampaian informasi rekomendasi area wisata kepada wisatawan dengan menyesuaikan kriteria yang ada serta detail lengkap informasi pada setiap tempat wisata

Tinjauan pustaka pada penelitian pertama dilakukan pada tahun 2017 dengan judul "*Designing Module e-complaint System Based on Geotagging and Geofencing*." Pada penelitian tersebut berfokus pada sistem pelayanan publik dalam pemerintah dengan merancang sistem pengaduan masyarakat atau *e-complaint*. Sistem *e-complaint* yang dirancang berbasis *geotagging* dan *geofencing* yang diharapkan dapat membantu masyarakat dalam menyampaikan informasi atau keluhan. Terdapat dua teknik yang digunakan pada sistem *e-complaint* tersebut, yang pertama teknik *geotagging* yang diharapkan dapat menentukan wilayah keluhan yang disampaikan dengan tepat sedangkan teknik





*geofencing* berfungsi untuk menentukan keberadaan wilayah keluhan yang disampaikan (Puspa, et al., 2017). Sedangkan fokus penelitian pada skripsi penulis dengan judul pembangunan aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* terdapat kemiripan pada teknik yang digunakan yaitu teknik *geotagging*. Secara umum pada aplikasi GoTindak menggunakan teknologi *geotagging* yang dapat membantu pengguna mengetahui lokasi pada saat melaporkan keluhan atau kritik pada parkir non resmi di wilayah Kota Malang.

Penelitian kedua dengan judul “Penerapan Konsep *Geotagging* Pada Aplikasi Tanggap Darurat Bencana Berbasis *Android*” yang dilakukan pada tahun 2019. Penelitian ini berfokus pada aplikasi untuk menyampaikan informasi bencana pada wilayah Majalengka dengan akurat dan lengkap. Aplikasi tanggap darurat yang dikembangkan menggunakan teknik *geotagging* agar memudahkan masyarakat dalam menyampaikan informasi bencana pada aplikasi *Android* (Nunu, et al., 2019). Sedangkan fokus penelitian pada skripsi penulis dengan judul pembangunan aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* terdapat kemiripan pada teknik yang digunakan yaitu teknik *geotagging*. Secara umum pada aplikasi GoTindak menggunakan teknologi *geotagging* yang dapat membantu pengguna mengetahui lokasi pada saat melaporkan keluhan atau kritik pada parkir non resmi di wilayah Kota Malang. Dengan harapan masyarakat dapat menyampaikan informasi dengan akurat dan lengkap dengan memberikan foto kejadian, lokasi, serta keterangan laporan yang disampaikan.

Penelitian ketiga dengan judul “Aplikasi Rekomendasi Spot Area Wisata Berbasis *Android* dengan Teknik *Geotag*” yang dilakukan pada tahun 2017. Penelitian ini berfokus pada penyampaian informasi rekomendasi area wisata kepada wisatawan dengan menyesuaikan kriteria yang ada serta detail lengkap informasi pada setiap tempat wisata. Teknik yang digunakan penelitian ini yaitu teknik *geotagging* dengan tujuan agar memudahkan masyarakat dalam memberikan informasi wisata secara detail dan lengkap (Mochamad, et al., 2017). Sedangkan fokus penelitian pada skripsi penulis dengan judul pembangunan aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* terdapat kemiripan pada teknik yang digunakan yaitu teknik *geotagging*. Secara umum pada aplikasi GoTindak menggunakan teknologi *geotagging* yang dapat membantu pengguna mengetahui lokasi pada saat melaporkan keluhan atau kritik pada parkir non resmi di wilayah Kota Malang. Dengan harapan masyarakat dapat menyampaikan informasi dengan akurat dan lengkap dengan memberikan foto kejadian, lokasi, serta keterangan laporan yang disampaikan.

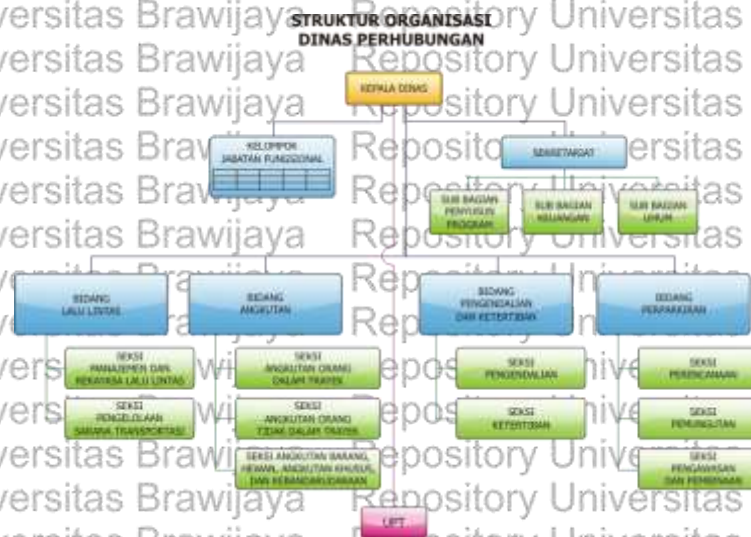
## 2.2 DISHUB Kota Malang (Dinas Perhubungan)

Tugas pokok DISHUB Kota Malang diatur dalam Peraturan Daerah nomor 6 tahun 2012. Dinas Perhubungan Kota Malang memiliki tanggung jawab kepada Walikota melalui Sekretaris Daerah yang memiliki fungsi sebagai pelaksana





otonomi Daerah pada bidang perhubungan untuk melaksanakan tugas pokok. Dalam melakukan penyusunan tugas pokok, dinas perhubungan memiliki fungsi melakukan perumusan kebijakan teknis operasional yang salah satunya teknis lalu lintas dan parkir. Karena sering bertambahnya penduduk di Kota Malang, pihak dishub harus dapat meningkatkan pelayanan khususnya pada bidang transportasi. Maka dari itu tugas DISHUB untuk memberikan kenyamanan pada masyarakat terhadap pelayanan perparkiran yang aman dan tertib (Dishub, 2015). Pada penelitian ini lebih difokuskan pada Bidang Perparkiran. Tugas pokok bidang perparkiran sebagai pengelola dan menyelenggarakan parkir yang berada pada tepi jalan umum. Berikut struktur organisasi pada DISHUB Kota Malang yang akan dijelaskan pada gambar 2.1



Gambar 2.1 Bagan Struktur Organisasi Dinas Pehubungan Kota Malang

Sumber: (Dishub, 2016)

### 2.3 Parkir

Parkir dideskripsikan sebagai kendaraan tidak bergerak yang sifatnya sementara sedangkan berhenti merupakan kendaraan yang sifatnya sementara tetapi pengemudi tetap berada dalam kendaraan. Parkir merupakan suatu kebutuhan pada masyarakat pemilik kendaraan, jika ingin memarkirkan kendaraannya di suatu tempat yang mudah dicapai. Area tempat parkir yang dimaksud yaitu tempat resmi yang diselenggarakan oleh pemerintah ataupun pihak lain milik badan usaha. Konsumen parkir membutuhkan tempat yang aman ketika sedang melakukan aktivitas parkir. Dalam lampiran Keputusan Menteri Dalam Negeri no 43 tahun 1980 tercantum penyelenggaraan perparkiran yang membahas mengenai Perparkiran Daerah yang menyatakan bahwa setiap daerah harus mampu melakukan pembinaan dan pengelolaan parkir dengan terpadu dan terkendali. Hal itu bertujuan untuk menjamin agar mewujudkan penataan lingkungan perkotaan dan kelancaraan lalu lintas (Dishub, 2017).





## 2.4 Crowdsourcing

Crowdsourcing terdiri dari dua kata yaitu *crowd* dan *source*. “*Crowd*” diartikan sebagai kerumunan orang, sedangkan “*sourcing*” dari kata kerja *source* yang artinya sumber daya. Sehingga dapat diartikan secara keseluruhan bahwa crowdsourcing merupakan suatu aktivitas atau tindakan yang dilakukan oleh masyarakat dan dibagikan secara bebas, terbuka ke banyak orang yang terhubung dalam suatu jaringan atau terkoneksi dengan internet (Howe, 2006). Jadi konsep di dalam crowdsourcing dapat disimpulkan bahwa adanya keterlibatan yang tidak terbatas, juga tidak memandang latar belakang; baik amatir atau profesional, kewarnegaraan, pendidikan, agama. Konsep ini berlaku bagi setiap orang yang ingin memberikan kontribusinya atau solusinya atas suatu permasalahan yang dibagikan oleh individu, perusahaan atau institusi, baik dibayar secara royalti ataupun dengan sukarela.

Crowdsourcing merupakan jenis aktifitas online yang dilakukan oleh individu, lembaga, organisasi atau perusahaan dengan memberikan tugas kepada kelompok atau individu melalui panggilan terbuka yang sifatnya fleksibel. Setiap orang yang melakukan crowdsourcing mendapatkan keuntungan kepuasan bersama dengan jenis kebutuhan yang diberikan baik dari segi ekonomi, pengakuan sosial dan mengembangkan keterampilan individu (Estellés-arolas, 2012).

## 2.5 Android

*Android* adalah sistem operasi berbasis *mobile* yang dipopulerkan dan dikembangkan oleh *Android, Inc* dan pada tahun 2005 resmi dibeli oleh *Google*. *Android* sebagai sistem operasi berbasis *linux* yang telah dirancang khusus, untuk perangkat bergerak yang disebut juga dengan *smartphone* atau telepon pintar. Sistem operasi *Android* bersifat *open source* sehingga sangat cocok untuk melakukan pengembangan pada perangkat *mobile* secara terus-menerus. Di samping itu, sebagai *developer* bisa secara bebas membangun aplikasi *Android* tanpa adanya gangguan hak cipta (Yudhanto, 2017). Dalam membangun aplikasi *Android* terdapat tiga bagian utama yang harus dipahami, yaitu *intent*, *service*, dan *content provider*. *Intent* merupakan komponen pada *Android* yang berfungsi untuk komunikasi agar dapat berpindah antar halaman. Selain itu, *intent* juga dapat berkomunikasi dengan *service* yang sedang aktif. *Service* yaitu komponen yang berjalan secara *background* dan tidak memiliki *Graphic User Interface* (GUI). Untuk dapat menjalankan setiap *service* harus berada dalam *thread* utama pada proses aplikasi. *Content Provider* adalah komponen yang digunakan pada aplikasi berupa kumpulan data aplikasi secara spesifik. *Content provider* memiliki cara untuk mengakses data yang tersedia yang dibutuhkan oleh *activity* (Nazruddin, 2015). Sistem operasi *Android* memiliki beberapa keunggulan, yaitu:





### 1. Lengkap (*Complete Platform*)

*Android* sebagai sistem operasi yang memiliki berbagai macam *tools* dalam membangun suatu aplikasi *mobile* sehingga memberikan peluang bagi *developer* untuk melakukan pengembangan aplikasi.

### 2. Bebas (*Open Source Platform*)

*Android* sebagai sistem operasi memberikan kebebasan untuk pengembang dalam membuat aplikasi *mobile*, karena lisensi pada *platform Android* bersifat *open source*.

### 3. Gratis (*Free Platform*)

*Android* sebagai sistem operasi yang bebas karena untuk mengembangkan tidak memerlukan tambahan biaya apapun serta dalam mendistribusikan aplikasi bersifat bebas.

## 2.6 Java

*Java* pertama kali dipelopori oleh James Gosling yang merupakan *engineer* dari *Sun Microsystem*. Pada tahun 1991 *Java* mulai dibangun. Selain itu, tahun 1995 *Java* dirilis dengan versi *alpha* dan *beta*. *Java* termasuk bahasa yang populer pada saat ini. Karena itu, terdapat beberapa versi *Java* yang diciptakan yaitu *J2EE* untuk Aplikasi *Enterprise* dan *J2ME* khusus aplikasi *mobile*. *Java* dapat bergerak pada berbagai macam *platform* sistem operasi sehingga dapat dikatakan bahasa *java* sangat populer (Bachtiar, 2018).

*Java* merupakan bahasa orientasi objek yang diinterpretasikan. Untuk menginterpretasikan *Java* harus menginstal sistem *host* terlebih dahulu. *Java* termasuk dalam bahasa orientasi objek untuk mengembangkan aplikasi mandiri, aplikasi berbasis internet, aplikasi pada *smartphone* yang dapat berkomunikasi lewat internet atau jaringan internet. *Java* bukan lagi bahasa yang dikhususkan untuk memperindah halaman web tetapi *Java* untuk mengembangkan aplikasi *enterprise* berskala besar yang berbasis jaringan. *Java* dirancang khusus untuk bergantung pada *platform* perangkat keras dan sistem operasi. Terdapat slogan yang diusung pada *Java* yaitu "*write once, run everywhere*", slogan tersebut memiliki makna kode program yang berjalan pada satu *platform* tidak perlu dikompilasi ulang agar berjalan pada *platform* yang lain. *Java* dapat berjalan pada sembarang (JVM) *Java Virtual Machine* yang tidak mempedulikan arsitektur pada komputer. Bahasa *Microsoft C#* atau (dibaca *C Sharp*) merupakan bahasa lanjutan yang terinspirasi dari pendekatan bahasa *Java*. Fitur-fitur dalam bahasa *Java* menjadikan adaptasi dalam perkembangan Bahasa *Microsoft C#*. Di samping itu, pengembangan aplikasi *Android* menggunakan bahasa *Java*. Dengan demikian, bahasa populer yang paling banyak digunakan dari bahasa pemrograman yang ada adalah bahasa *Java* terutama pada aplikasi *web client-server* (Hariyanto, 2014).





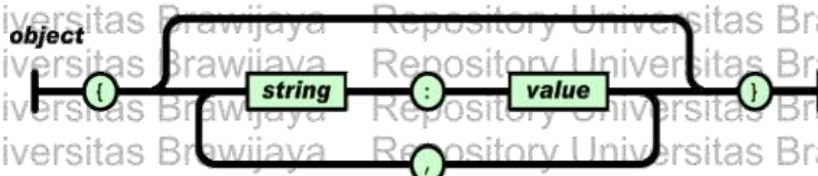
## 2.7 Android Studio

*Android studio* merupakan pengembangan terpadu dari *IDE* atau (*Integrated Development Enviroment*) dan telah resmi menjadi pengembangan aplikasi *android* dengan didasarkan dari *IntelliJ IDEA*. *Android studio* memiliki sifat yang *open source* sehingga sangat memudahkan bagi pengembang. *Android studio* saat ini telah menggantikan *tool Eclipse* dengan mempersiapkan *tool* untuk mengembangkan *android* yang saling berintegrasi untuk melakukan aktivitas *develop* serta *debugging* pada aplikasi. Fitur yang terdapat dalam *android studio* yaitu dapat melakukan *build system* dengan basis *gradle* yang fleksibel, fitur emulator yang memudahkan *develop* serta dapat melakukan uji fitur dengan lengkap. Lisensi *Apache* yang menaungi *android studio*. Tahap awal *preview IDE* dimulai dari versi 1.0, kemudian lanjut ke versi 0.8 atau versi beta pada Juni 2014. *Tools final android studio* diluncurkan pada Desember 2014 yang dapat diunduh pada *browser Window, MacOS* serta *Linux* (Hermawan, 2011).

## 2.8 Javascript Object Notation (JSON)

*JSON* atau *java script object notation* mendeskripsikan *format* pertukaran struktur data yang ringan pada komputer, sehingga memudahkan untuk ditullis dan dibaca oleh pengembang. Hal itu mempermudah mesin untuk melakukan proses *generate*. *Format* keluaran pada *JSON* berbentuk teks dengan menampilkan *format* struktur data yang bersifat *independent* yang tidak akan memepengaruhi dalam bahasa pemrograman lainnya (Kurniaji, 2015). Dalam *format JSON* terdapat dua struktur yaitu *object* dan *array*. Daftar *value* dari *JSON* berbentuk *number, string, array, obejct, true, false* atau *null*. *Format data JSON object* mendeskripsikan sebuah pasangan set nama dan nilai yang tidak berurutan, *JSON object* diawali menggunakan tanda tag kurung kurawal terbuka “{” dan diakhiri tanda kurung kurawal tertutup “}” pada *format JSON* itu sendiri. Setiap nilai diikuti dengan tanda titik dua “:” sebagai nilai variabel itu sendiri, jika terdapat set nilai maka dipisahkan dengan tanda koma “,”. Untuk *format JSON array* mendeskripsikan set nilai yang berurutan yang ditandai dengan tanda tag kurung siku buka “[” dan diakhiri dengan kurung siku tutup “]”. Untuk setiap nilai pada *array* dibedakan dengan tanda koma “,” (*JSON*, 2019). Berikut penjelasan bentuk yang digunakan daam *format JSON* yang akan dijelaskan pada gambar2.2, gambar2.3 dan gambar2.4

### 1. Object



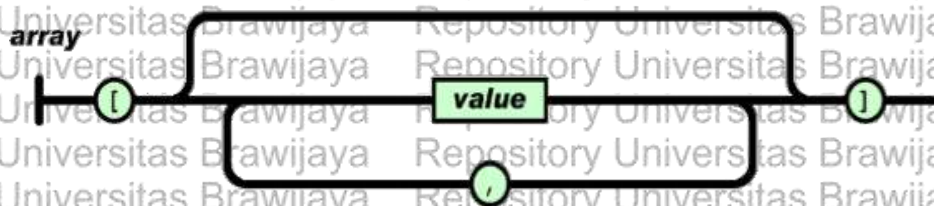
Gambar 2.2 *JSON* dengan *object*

Sumber: (*JSON*, 2019)





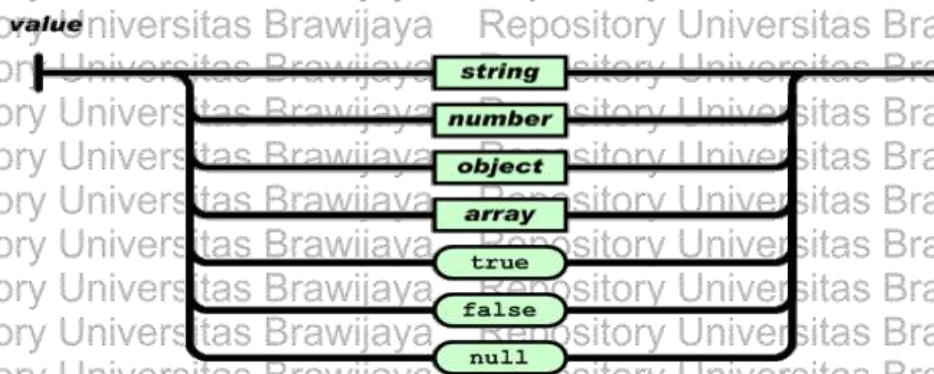
## 2. Array



Gambar 2.3 JSON dengan Array

Sumber: (JSON, 2019)

## 3. Value



Gambar 2.4 JSON dengan Value

Sumber: (JSON, 2019)

## 2.9 Geotagging

*Geotagging* merupakan teknik *tagging* untuk menambahkan sebuah metadata geospasial pada berbagai macam media seperti konten web, *e-book*, dokumen, narasi, serta video dan aplikasi media sosial lainnya. Dengan menggunakan alat media sosial dan aplikasi, pengguna dapat menambahkan informasi geospasial yang terintegrasi oleh penerima *GPS* dan dapat dilakukan secara otomatis. Informasi yang akan ditambahkan oleh *GPS* yaitu koordinat *latitude* dan *longitude*. Pada suatu media dapat diletakkan suatu Teknik *geotagging* yang terdiri dari beberapa metode yaitu (Nandipati, 2011):

1. *Geocoding* dengan teknik manual mendeskripsikan fungsi penambahan secara manual untuk melakukan *input* koordinat dan mencari lokasi untuk melakukan proses *upload* pada internet.
2. Menggunakan kamera digital dengan menggunakan fungsi *GPS*, sehingga untuk sebagian kamera sudah terlengkapi dengan fungsi *GPS* langsung. Sehingga memiliki informasi lokasi pada data yang bersangkutan.
3. Menggunakan teknik kamera digital menggunakan fungsi *GPS* yang terpisah, lalu terletak pada kamera, dan belum dilengkapinya *GPS geotagging* dengan cara menyinkronkan kamera dengan fitur *GPS* pada sebuah *software*.





Dalam mengimplementasikan teknik *geotagging* dapat ditemukan pada banyak domain seperti di twitter, halaman web, artikel, dan sebagainya. Terdapat berbagai jenis pendekatan bahwa informasi mengenai geospasial dapat tersimpan dalam media informasi. Pendekatan yang digunakan oleh penulis menggunakan konteks manual karena dapat merekam lokasi secara langsung serta lokasi yang ditentukan pada server sedangkan teknik *geotagging* dapat melakukan perekaman lokasi secara langsung atau *geoparsing dan geocoding* (Hunter, 2012).

## 2.10 Codeigniter

*Codeigniter* merupakan kerangka kerja dengan mengembangkan aplikasi (*Application Development Framework*) dalam pembuatan suatu kerangka kerja program yang lebih cepat dengan menggunakan PHP yang lebih sistematis. Dengan menggunakan kerangka kerja CI, pengembang dapat dengan mudah membangun aplikasi dengan cepat, dengan mengikuti kerangka yang telah disediakan oleh kerangka CI. *Codeigniter* tidak perlu membuat program dari awal, karena *library* fungsi-fungsi pada *framework* CI dapat membuat program dengan cepat. Pemrograman CI hanya perlu mengakses *library* yang tersedia untuk menggunakan antarmuka dan struktur logika yang sederhana. Teknik yang dapat digunakan pada *framework* CI yaitu konsep MVC (*Model View Controller*).

MVC yaitu teknik pemrograman yang membagi program dalam tiga komponen bagian yaitu *model*, *view* dan *controller*. *Model* merupakan implementasi logika pada data dalam sistem aplikasi. Objek model umumnya digunakan untuk mengambil data dari *database* serta menyimpan data ke dalam *database*. *View* merupakan komponen untuk mengimplementasikan antarmuka *user interface* yang dibuat berdasarkan data dari model. *Controller* merupakan komponen yang digunakan untuk memproses interaksi antar pengguna yang merupakan *file* utama dalam setiap modul saat menggunakan CI. *Controller* bekerja dengan *model* dan memilih *view* yang digunakan untuk mengolah data. Dengan demikian, prinsip dari MVC memodularkan program (Sidik, 2018).

## 2.11 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak merupakan penyusunan dengan memakai metode atau konsep rekayasa yang memiliki tujuan untuk mempelajari berbagai perencanaan dan dapat menghasilkan perangkat lunak yang berkualitas dengan ditinjau dari aspek waktu dan biaya serta perangkat lunak dapat bekerja secara optimal. Untuk memenuhi kebutuhan customer maka banyak permintaan untuk dibuat perangkat lunak yang akhirnya menjadi tidak digunakan karena tidak memenuhi kebutuhan pengguna atau juga dapat dipengaruhi customer yang tidak ingin mengubah dari hal manual menjadi otomatis. Oleh karena itu sangat dibutuhkan rekayasa perangkat lunak supaya perangkat lunak yang telah dibuat tidak menjadi perangkat lunak yang tidak digunakan tetapi dapat dimanfaatkan secara efisien. Tentunya untuk tercapainya semua itu dibutuhkan pendekatan yang terencana atau sistematis serta kreativitas dengan pemilihan kriteria yang akan digunakan dalam persyaratan tertentu. Untuk memenuhi kebutuhan





customer maka dibutuhkan perangkat lunak yang baik yang terfokus pada pelanggan bukan hanya terfokus pada tim pengembang perangkat lunak (Simarmata, 2010).

Proses perangkat lunak dapat dikatakan sebagai sekumpulan yang bertujuan untuk melakukan pengembangan atau perubahan pada perangkat lunak. Proses perangkat lunak terbagi menjadi empat aktivitas secara umum, yang terdiri dari:

1. Pengumpulan Spesifikasi (*Specification*), merupakan langkah yang menentukan sistem perangkat lunak bagian apa saja yang harus dikerjakan serta terdapat batasan dalam melakukan pengembangan perangkat lunak.
2. Pengembangan (*Development*), merupakan langkah untuk memproduksi atau membangun sistem perangkat lunak yang mana telah dilakukan perancangan perangkat lunak sebelumnya.
3. Validasi (*Validation*), merupakan langkah untuk mengetahui perangkat lunak yang dikembangkan apakah sudah memenuhi kebutuhan dari pelanggan itu sendiri.
4. Evolusi (Evolusi), merupakan langkah untuk mengetahui perubahan yang terjadi pada kebutuhan perangkat lunak agar permintaan kebutuhan dari pelanggan dapat terpenuhi.

## 2.12 Software Development Life Cycle

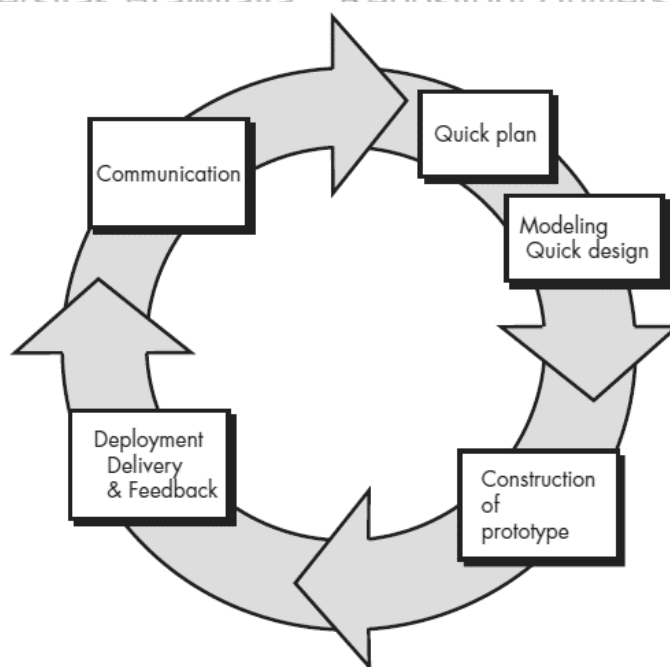
*Software Development Life Cycle* atau dapat juga disebut *System Software Development Life Cycle* merupakan proses untuk melakukan pembangunan atau pengembangan dan perubahan pada sistem perangkat lunak dengan menggunakan rancangan dan menggunakan metodologi yang pernah digunakan oleh orang untuk membangun sistem perangkat lunak yang telah dibuat sebelumnya. Dalam mengembangkan perangkat lunak harus berdasarkan dengan aturan-aturan yang sebelumnya teruji dengan baik. Dalam proses pembuatan perangkat lunak diharuskan memilih siklus tahapan supaya menghasilkan perangkat lunak yang berkualitas baik. Terdapat beberapa jenis aktivitas dari sebuah tahapan SDLC yang dapat diselesaikan dalam urutan aktivitas tertentu dengan model-model dan metodologi pengembangan perangkat lunak. Dalam pemilihan SDLC yang akan dijadikan metodologi dalam pengembangan perangkat lunak pasti memiliki kelebihan dan kelemahannya pada setiap model SDLC. Hal yang paling diperhatikan harus mengetahui karakteristik *customer* itu sendiri dan karakteristik pada pengembang. Pada penelitian ini menggunakan model prototipe karena model prototipe dianggap pendekatan yang paling baik karena biasanya pengguna tidak dapat menganalisis fitur atau fungsi yang tersedia pada perangkat lunak secara detail. Dengan demikian pengembang dapat melakukan beberapa kali perbaikan pada perangkat lunak sampai pelanggan merasa puas dengan perangkat lunak yang dikembangkan (Pressman, 2010).





### 2.13 Model Prototipe

Proses pengembangan perangkat lunak yang diinginkan oleh pengguna dilihat pada permasalahan analisis kebutuhan. Pada penelitian ini menggunakan model prototipe untuk membangun sistem. Pemilihan model prototipe ini sangat cocok untuk menganalisis kebutuhan pengguna secara terperinci karena sering kali pengguna mendapatkan masalah untuk mengungkapkan kebutuhan secara terperinci tanpa melihat gambaran umum proses yang tak diragukan lagi. Model prototipe digunakan sebagai penyambung ketidajelasan pengguna terhadap pengembangan sebuah perangkat lunak. Terdapat beberapa jenis pada model prototipe yaitu *incremental prototyping*, *rapid prototyping*, *extreme prototyping*, dan *evolutionary prototyping*. Penelitian ini menggunakan model *evolutionary prototyping* untuk mengembangkan perangkat lunak, pendekatan prototipe ini dirasa sangat baik untuk membantu *stakeholder* serta tim pengembang dalam memikirkan spesifikasi kebutuhan yang tidak pasti (Pressman, 2010).



**Gambar 2.5 Model Prototipe**

Sumber: (Pressman, 2010).

Pada model prototipe di atas, yang harus dilakukan pertama kali yaitu komunikasi antara pengembang atau *developer* dengan pengguna. Pada proses pengembangan perangkat lunak tim pengembang mengadakan pertemuan dengan pemangku kepentingan untuk mengetahui keseluruhan tujuan pengembangan, mengidentifikasi kebutuhan yang diketahui oleh *stakeholder*, dan memikirkan iterasi yang akan didefinisikan lebih jauh. Kemudian, melakukan perencanaan iterasi prototipe dengan melakukan rancangan cepat dan melakukan perancangan cepat pada rancangan antarmuka pengguna atau disebut juga





format tampilan. Rancangan cepat yang dilakukan merupakan awal untuk memulai intruksi pembuatan prototipe yang kemudian rancangan cepat ini diserahkan ke *stakeholder* untuk dilakukan dievaluasi yang sebelumnya sudah dibuat. Kemudian *stakeholder* memberikan umpan balik yang fungsinya untuk memperbaiki spesifikasi kebutuhan. Untuk memenuhi kebutuhan *stakeholder* maka pengembang melakukan perbaikan pada prototipe proses ini dapat dikatakan sebagai iterasi (Pressman, 2010). Pada model prototipe terdapat kelemahan dan kelebihan sebagai berikut:

**Tabel 2.2 Kelebihan dan Kekurangan SDLC Prototipe**

Kelebihan	Kekurangan
Tim pengembang dan pengguna dapat saling terlibat dalam pengembangan perangkat lunak.	Adanya penambahan pada spesifikasi kebutuhan, menyebabkan tim pengembang banyak mengalah kepada pengguna.
Model kerja yang terlihat dapat membantu pengguna mendapatkan permasalahan secara mendalam.	Adakalanya pengguna mengalami kebingungan dalam

## 2.14 UML

UML atau *Unified Modeling Language* yaitu bahasa yang berfungsi memodelkan berbagai artefak rekayasa perangkat lunak dengan metode yang mendukung pemrograman berorientasi objek. UML hanya berfungsi untuk memodelkan perangkat lunak. Penggunaan UML pada perangkat lunak dapat diwujudkan dalam bentuk visual diagram dan simbol. Selanjutnya, simbol dan diagram yang selesai dibangun untuk diimplementasikan menjadi kode program. Untuk membuat UML telah disediakan program untuk dapat membuat kode program berdasarkan perancangan diagram UML. Diagram UML dapat diimplementasikan dengan menggunakan berbagai bahasa pemrograman dengan ketentuan mendukung pemrograman berorientasi objek. Dalam hal ini, UML sangat dibutuhkan untuk memodelkan visualisasi, menspesifikasikan, dan melakukan dokumentasi sistem aplikasi perangkat lunak. Pada UML terdapat berbagai jenis diagram yang memiliki perbedaan fungsi (Shalahuddin, 2018).







### 2.14.1 Use case Diagram

*Use case diagram* adalah deskripsi diagram dengan fungsi dari sebuah sistem dengan menggambarkan interaksi fungsionalitas yang dilakukan oleh aktor terhadap sistem yang dipakai. *Use case* yaitu aktivitas aktor terhadap sistem. Dengan kata lain, *use case* memiliki kegunaan untuk mengetahui fungsionalitas pada sistem yang saling berperilaku atau berinteraksi antar aktor. Komponen dalam *use case diagram* tersusun atas *use case*, aktor, asosiasi, ekstensi, generalisasi, *include*, dan *extend* (Yasin, 2012). Simbol pada diagram *use case* akan dijelaskan dalam tabel 2.1 berikut:





Table 2.1 Penjelasan Elemen *Use case Diagram*

No	Elemen	Simbol	Deskripsi
1	<i>Actor</i>	 Actor	Elemen aktor yang digambarkan dengan simbol orang. Akan tetapi tidak semua aktor adalah orang, dapat berupa sebuah sistem lain yang saling berinteraksi dengan sistemnya sendiri.
2	<i>Use case</i>	 Use Case	Elemen fungsionalitas pada sistem yang dilakukan oleh aktor untuk saling bertukar informasi pada sistem. Untuk itu penulisan <i>use case</i> menggunakan kata kerja.
3	<i>Association</i>		Sebuah interaksi yang menghubungkan antara <i>user</i> atau pengguna <i>use case</i> .
4	<i>Extend</i>		<i>Extend</i> ialah hubungan tambahan pada <i>use case</i> yang menjelaskan proses lanjutan ke sebuah <i>use case</i> sebelumnya.
5	<i>Include</i>		<i>Include</i> ialah relasi tambahan pada <i>use case</i> yang mana untuk menjalankan fungsi pada sistem memerlukan <i>use case</i> sebelumnya, karena merupakan syarat untuk menjalankan <i>use case</i> ini.
6	Generalisasi		Generalisasi ialah hubungan dua buah <i>use case</i> dimana fungsi salah satu <i>use case</i> merupakan fungsi <i>use case</i> yang lebih umum.

Sumber: (Rossa & Shalahuddin, 2014)

#### 2.14.2 *Use case Scenario*

Skenario *use case* atau *use case scenario* ialah proses urutan kejadian jalannya *use case* pada fungsionalitas yang disediakan oleh *use case* dari sisi aktor dan sistem. Tabel yang akan dijelaskan pada *use case scenario* meliputi keterangan lebih lanjut pada fungsioanal sistem yang digambarkan pada *use case diagram*. Dalam konteks ini untuk mendefinisikan *use case scenario* harus sebanding dengan jumlah *use case diagram* yang dibentuk. Dengan kata lain *use case scenario* dibentuk untuk mempresentasikan *use case* yang terdapat pada *use case diagram* (Shalahuddin & Rossa, 2015). Selanjutnya terdapat deskripsi *scenario* yang akan diuraikan dalam tabel 2.3.



Tabel 2.3 Komponen pada *Use case Scenario*

Nama Komponen	Deskripsi
<i>Objectives</i>	Menjelaskan object dari <i>use case</i>
<i>Actors</i>	Merupakan <i>abstraction</i> dari pengguna yang mengaktifkan jalannya sistem, tidak semua pengguna adalah orang bisa berupa sebuah sistem sendiri ataupun sistem lain.
<i>Precondition</i>	Memaparkan proses aktivitas sistem sebelum masuk ke dalam <i>use case</i> .
<i>Main Flow</i>	Memaparkan jalannya <i>use case</i> .
<i>Alternative Flow</i>	Memaparkan alur jalannya sistem dengan memberikan alternative ketika sistem tidak berjalan normal.
<i>Post Condition</i>	Memaparkan hasil kondisi setelah <i>use case</i> dilakukan.

Sumber: (Rossa & Shalahuddin, 2014)

### 2.14.3 Sequence Diagram



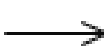

Sequence Diagram mendeskripsikan gambaran perilaku pada sebuah *scenario* atau gambaran antar objek pada *use case* dengan memaparkan *message* yang dikirimkan dan diterima antar objek dalam urutan waktu. Dengan kata lain, sequence diagram dibangun untuk memenuhi kebutuhan fungsional pada sistem, sehingga dapat mengetahui interaksi pada setiap objek. Dengan demikian, untuk membuat sequence harus diketahui terlebih dahulu objek yang digunakan pada *use case* dan juga *method* yang dimiliki pada kelas yang menginisiasi objek. Skenario pada *use case* juga dibutuhkan untuk membuat sequence diagram. Pada prinsipnya, banyaknya *use case* yang dijelaskan alur interaksinya pada *use case* diagram, maka semakin banyak pula *use case diagram* yang akan digambarkan (Shalahuddin & Rossa, 2015). Selanjutnya terdapat deskripsi tiap elemen dari *sequence* yang diuraikan dalam tabel 2.4.

Tabel 2.4 Elemen pada *Sequence Diagram*

Elemen	Simbol	Deskripsi
<i>Actor</i>		Nama aktor dapat berupa gambar orang, tetapi tidak semua aktor adalah orang. Dapat berupa kata benda yang terdapat di awal nama aktor. Aktor dapat juga berupa sistem lain yang saling berinteraksi dengan sistem lainnya.
<i>Lifeline</i>		<i>Lifeline</i> atau garis hidup merupakan kehidupan suatu objek pada sistem.



Tabel 2.4 Elemen pada *Sequence diagram* (Lanjutan)

Elemen	Simbol	Deskripsi
<i>Object</i>		Objek merupakan suatu pesan yang saling berinteraksi pada objek
<i>Activation</i>		<i>Activation</i> atau waktu aktif merupakan tahapan yang dilakukan objek
<i>Message</i>		Hubungan yang memuat informasi pada pada objek, dengan mengirimkan pesan dari satu <i>participant</i> ke <i>participant</i> yang lain atau dari <i>lifeline</i> satu ke <i>lifeline</i> yang lain.
<i>Self Message</i>		Menggambarkan hubungan pesan pada objek itu sendiri.
<i>Asynchronous Message</i>		Menyatakan hubungan suatu objek mengirimkan informasi pada objek lainnya.
<i>Return Message</i>		Hubungan pada objek yang telah selesai dieksekusi dan memberikan kembalian pada objek.

Sumber: (Rossa &amp; Shalahuddin, 2014)

#### 2.14.4 Class Diagram

*Class diagram* mendefinisikan struktur sistem yang digambarkan dalam kelas-kelas untuk membangun suatu sistem. Dalam *class diagram* terdapat suatu kelas yang memiliki atribut dan metode. Perancangan kelas yang dibuat harus mencakup seluruh kebutuhan pada sistem. Dengan demikian tujuan *class diagram* dibuat agar dokumentasi perancangan dan perangkat lunak dapat saling berkesinambungan. Terdapat elemen penyusun struktur kelas yang baik pada diagram kelas yaitu *main class*, *view*, *controller*, dan model. Struktur pada *main class* memiliki fungsi awal pada saat program pertama kali dijalankan. Struktur *view* berfungsi mengatur tampilan *view* pada sistem. Terdapat kelas *controller* yang mengatur jalannya proses bisnis pada suatu perangkat lunak. Selanjutnya terdapat model yang berfungsi untuk membungkus atau mendefinisikan data (Shalahuddin & Rossa, 2015). Selanjutnya terdapat deskripsi tiap elemen dari *class diagram* yang diuraikan dalam tabel 2.5.



Tabel 2.5 Elemen pada *Class Diagram*

Elemen	Simbol	Skripsi
<i>Class</i>		Struktur kelas yang terdapat pada sistem.
<i>Interface</i>		Dalam konteks ini, sama seperti <i>interface</i> yang terdapat pada pemrograman OOP.
<i>Association</i>		Elemen yang merelasikan antar kelas dengan artian umum.
<i>Direct Association</i>		Merupakan kelas yang menghubungkan antara kelas yang satu dan digunakan juga dengan beberapa kelas.
<i>Generalisasi</i>		Penghubung antar kelas dengan arti hubungan umum ke khusus.
<i>Dependency</i>		Penghubung antara kelas yang memiliki hubungan ketergantungan antar kelas.
<i>Agregation</i>		Penghubung antar kelas dengan arti agregat pada semua bagian.

Sumber: (Rossa &amp; Shalahuddin, 2014)




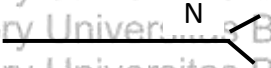
### 2.14.5 Entity Relationship Diagram

ERD kepanjangan dari *Entity Relationship Diagram* yang merupakan pemodelan awal pada basis data. Kegunaan ERD untuk melakukan pemodelan basis data rasional. Terdapat dua bentuk pada diagram ERD yaitu bentuk *physical* dan *conceptual*. Bentuk ERD *physical* menggambarkan hubungan relasi antar entitas yang digambarkan dalam bentuk tabel dan komponen dalam tabel itu sendiri. Sedangkan ERD *conceptual* menggambarkan hubungan relasi antar komponen yang dilengkapi dengan kunci relasi untuk menghubungkan entitas dengan relasi yang dimiliki (Shalahuddin & Rossa, 2015). Selanjutnya terdapat deskripsi tiap elemen dari *ERD* yang diuraikan dalam tabel 2.6.





Tabel 2.6 Elemen pada *Entity Relationship Diagram*

Simbol	Deskripsi
	Entitas merupakan data inti yang akan disimpan pada basis data, biasanya penamaan pada entitas menggunakan kata benda.
	Relasi merupakan penghubung antara entitas.
	Atribut merupakan kolom data yang tersimpan dalam suatu entitas.
	Asosiasi menghubungkan antara relasi dan entitas yang mana kedua ujungnya memungkinkan memiliki jumlah pemakaian.

Sumber: (Rossa & Shalahuddin, 2014)

## 2.15 Pengujian Perangkat Lunak

Diperlukannya pengujian perangkat lunak dengan tujuan mengetahui apakah setiap fungsi yang telah diimplementasikan sudah berjalan dengan baik serta untuk mengetahui kepuasan pengguna saat menjalankan aktivitas pada sistem aplikasi. Berikut merupakan pengujian yang akan dilakukan pada penelitian ini.

### 2.15.1 Pengujian Unit

Pengujian unit yaitu pengujian yang terletak pada unit atau *individual components* dalam sistem aplikasi yang diimplementasi. Pengujian ini bertujuan untuk memfokuskan komponen pada unit terkecil dari suatu modul perangkat lunak. Pengujian unit dilakukan dengan cara melihat kode program pada sistem untuk dapat diketahui kesalahan batas modul pada jalur kontrol suatu kode program, sehingga akan menghasilkan kompleksitas (Sommerville, 2011). Teknik pengujian ini dilakukan dengan menggunakan *basis path testing*. Dalam hal ini, *basis path testing* berfungsi untuk mengetahui pengujian dalam setiap jalur algoritme yang telah diimplementasikan.

### 2.15.2 Pengujian Validasi

Pengujian validasi yaitu pengujian yang berdasarkan kebutuhan terdefinisi sebelumnya. Teknik yang dipakai pada pengujian validasi yaitu *blackbox testing*. Pengujian validasi dilakukan apakah fungsional pada sistem telah berhasil dan valid sesuai dengan masukan dan keluaran yang dihasilkan (Sommerville, 2011).





### 2.15.3 Pengujian *Usability*

Pengujian *usability* yaitu pengujian pada sistem aplikasi untuk mengevaluasi pengalaman pengguna dalam menjalankan suatu sistem yang mangacu kemudahan akses pengguna. Dalam hal ini, bertujuan untuk mengetahui seberapa mudah antarmuka pengguna pada sistem aplikasi yang telah dirancang. Selain itu, untuk mengetahui tingkat level dari *usability*. Pengujian *usability* langsung diujikan pada calon pengguna sehingga penilaian dilakukan secara objektif. Instrumen yang digunakan pada pengujian *usability* yaitu menggunakan metode SUS atau (*System Usability Scale*) (Nielsen, 2012).

#### 2.15.3.1 *System Usability Scale* (SUS)

Dalam mendapatkan nilai *usability* dapat menggunakan skala SUS dengan menyediakan ukuran skala "*quick and dirty*" yang handal untuk mengukur skala *usability*. Pada skala SUS terdiri dari sepuluh pernyataan sederhana yang dapat memberikan pandangan secara global. Metode SUS terdiri dari 10 pernyataan yang terdapat pada kuesioner. Adapun cara dalam pengujian SUS yaitu dihitung menggunakan skala 1 sampai 5 setiap pernyataan, yang mana untuk setiap pernyataan terdapat 5 opsi jawaban. Berikut merupakan item pernyataan metode SUS pada gambar 2.6.

No	Template Question
1	I think that I would like to use this system frequently.
2	I found the system unnecessarily complex.
3	I thought the system was easy to use.
4	I think that I would need the support of a technical person to be able to use this system.
5	I found the various functions in this system were well integrated.
6	I thought there was too much inconsistency in this system.
7	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9	I felt very confident using the system.
10	I needed to learn a lot of things before I could get going with this system.

Gambar 2.6 Item *System Usability Scale* (SUS)

### 2.15.4 Pengujian *Compatibility*

Pengujian *compatibility* adalah teknik pengujian non fungsional yang dilakukan untuk menguji kompatibilitas pada suatu sistem, apakah dapat digunakan pada browser, sistem operasi, perangkat tampilan, hardware dengan lingkungan yang berbeda-beda. Dalam aplikasi web perbedaan perambah terkadang dapat memberikan hasil yang tidak sama dan ini tanpa mempengaruhi





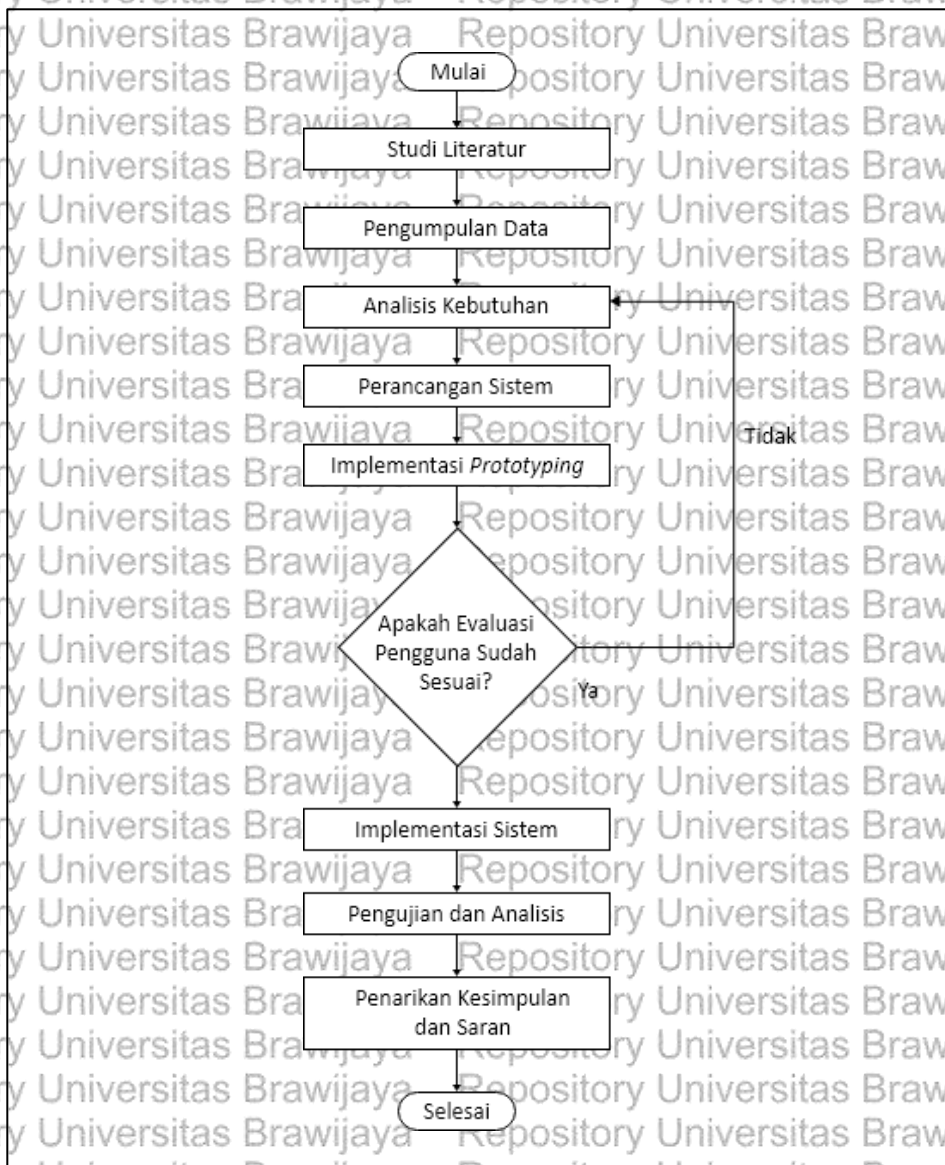
tingkatan pada standar isi HTML. Terkadang tidak tersedianya *plug-in* pada *browser* serta konfigurasi tertentu yang tidak tersedia dapat menampilkan hasil yang berbeda. Terdapat beberapa kasus, dalam permasalahan kompatibilitas contohnya saat melakukan *download* tidak dapat dilakukan, karena tidak tersedianya *plugin* pengunduhan sehingga isi *content* pada sistem menjadi tidak lengkap. Permasalahan seperti ini sering terjadi karena perbedaan *browser* dapat mengubah tata letak halaman, *font style* dapat berubah sehingga tidak dapat terbaca, lalu beberapa *form* mungkin tidak dapat terorganisasi dengan baik. Pengujian kompatibilitas bertujuan untuk mengungkap permasalahan pada aplikasi web sebelum dibuat *online*. Dalam hal ini, pengujian kompatibilitas berusaha mengungkap kesalahan atau masalah saat mengeksekusi yang dapat diketahui melalui perbedaan-perbedaan konfigurasi (Pressman, 2010). Dalam pengujian compatibility menggunakan bantuan aplikasi SortSite. SortSite dapat melakukan lebih dari 1200 checkpoint setiap kali menguji satu halaman (PowerMapper, 2019).





## BAB 3 METODOLOGI PENELITIAN

Bab ini membahas metodologi yang akan digunakan pada Pembangunan Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep *Crowdsourcing* Menggunakan *Geotagging*. Pada penelitian ini menggunakan tipe penelitian implementatif. Dalam metodologi ini mempunyai tahap-tahap observasi yang dikerjakan. Berikut tergambar dalam alur bagan 3.1 dibawah ini.



Gambar 3.1 Diagram Alir Metodologi Penelitian

### 3.1 Studi Literatur

Studi literatur berfungsi untuk mendapatkan teori dalam melakukan penyusunan pada penelitian yang digunakan sebagai dasar dalam membangun sistem Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep





*Crowdsourcing* Menggunakan *Geotagging*. Untuk mendapatkan literatur dalam memperoleh referensi ataupun penelitian yang berkaitan dengan buku, *paper*, e-book, jurnal dan media lain sebagainya. Beberapa referensi yang terkait dalam penelitian ini di antaranya adalah:

1. Dinas Perhubungan Kota Malang
2. Parkir
3. *Crowdsourcing*
4. *Android*
5. Java
6. *Android Studio*
7. *JSON*
8. *Geotagging*
9. Codeigniter
10. Rekayasa Perangkat Lunak
11. *Software Development Life Cycle*
12. Model Prototipe
13. *UML*
  - a. *Use case Diagram*
  - b. *Activity Diagram*
  - c. *Class Diagram*
  - d. *Sequence Diagram*
14. Pengujian Perangkat Lunak
  - a. Pengujian Unit
  - b. Pengujian Validasi
  - c. Pengujian *Usability*
  - d. Pengujian *Compatibility*

### 3.2 Pengumpulan Data

Proses ini mempunyai tujuan agar lebih mudah saat melakukan pengumpulan keterangan informasi berupa data yang mana untuk menentukan keperluan program atau mencari keperluan pengguna yang nantinya sistem dapat bermanfaat bagi pengguna. Pada penelitian ini akan digunakan prinsip wawancara dan kuesioner dengan melakukan pengumpulan informasi berupa data. Pada pengumpulan data ini menggunakan teknik *simple random sampling*.

#### 3.3.1 Wawancara

Prinsip ini mempunyai tujuan mendapatkan informasi yang dibutuhkan untuk dapat memenuhi keperluan sistem. Hasil wawancara terdapat pada halaman lampiran. Wawancara dilakukan kepada pengguna parkir khususnya pengendara sepeda motor atau mobil serta melakukan wawancara pada DISHUB Kota Malang. Informasi yang akan digali mengenai prosedur parkir resmi dan non resmi dari DISHUB Kota Malang pada Bidang Perparkiran dengan Bapak Hary Dwi Yuniarto, S.Psi selaku seksi pendataan pada bidang perparkiran dan informasi mengenai alur proses untuk menyampaikan informasi laporan mengenai permasalahan parkir di





Kota Malang, informasi tentang laporan mengenai permasalahan identifikasi parkir dari laporan masyarakat, informasi mengenai tindak lanjut DISHUB ketika menerima laporan parkir dari masyarakat.

### 3.2.1 Kuesioner

Kuesioner disebarakan pada beberapa sampel masyarakat pada pengguna parkir motor dan mobil. Sampel diambil di ruang lingkup Kota Malang pada Kecamatan Lowokwaru. Dengan memberikan angket kuesioner berisi pertanyaan yang telah disusun sebelumnya pada lampiran, selain itu mudah dimengerti dan bersifat kombinasi. Untuk jawaban pertanyaan sudah tersedia terlebih dahulu. Pertanyaan yang akan diajukan adalah permasalahan parkir non resmi yang ada di Kota Malang.

### 3.3 Analisis Kebutuhan

Tahapan selanjutnya adalah menganalisis kebutuhan sistem. Tujuan dari analisis kebutuhan ini adalah membangun pemahaman mengenai karakteristik dari ruang lingkup permasalahan dan kumpulan kebutuhan untuk mendapatkan sebuah solusi yang efektif. Kebutuhan sistem harus digali agar sistem nantinya akan berjalan sesuai dengan fungsinya, juga membuat sebuah sistem menjadi terarah. Analisis kebutuhan dilakukan untuk menggali kebutuhan pada sistem, kebutuhan tersebut merupakan kebutuhan dari pengguna. Kebutuhan ini nantinya akan menjadi sebuah fungsi yang harus ada pada sistem. Adapun proses untuk menganalisis kebutuhan berdasarkan data jawaban hasil wawancara yang terdapat pada lampiran yaitu pada dinas perhubungan Kota Malang bagian bidang perparkiran dengan seksi pemetaan dan menyebarkan kuesioner pada pengguna parkir.

Adapun tahapan untuk analisis kebutuhan terdapat 5 tahapan yaitu, identifikasi aktor, analisis kebutuhan fungsional, pemodelan *use case diagram*, skenario *use case* dan analisis kebutuhan non fungsional. Identifikasi aktor adalah mengidentifikasi aktor-aktor yang akan berinteraksi dengan sistem, selanjutnya melakukan analisis kebutuhan fungsional yaitu dengan menentukan layanan-layanan sistem yang dibutuhkan oleh aktor yang kemudian akan dimodelkan dengan *use case diagram* untuk menggambarkan interaksi aktor dengan sistem. Skenario *use case* digunakan untuk menjelaskan lebih detail dari masing-masing *use case* yang terdapat dari *use case diagram*. Kemudian dilakukan analisis kebutuhan non fungsional.

### 3.4 Perancangan Sistem

Tahapan selanjutnya yaitu perancangan sistem sebagai penentu kebutuhan aplikasi yang diperoleh dalam proses analisis kebutuhan pembangunan aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* pada *platform* android. Pada perancangan sistem terdapat tiga proses yaitu perancangan UML, perancangan komponen atau algoritme, perancangan antarmuka sistem, dan perancangan data. Pada tahap





perancangan sistem akan dilakukan beberapa tahapan proses iterasi. Tahapan perancangan sistem pada penelitian ini yaitu:

#### 1. Perancangan UML

Perancangan UML berfungsi untuk memodelkan artefak rekayasa perangkat lunak dengan metode yang mendukung pemrograman berorientasi objek. UML sangat dibutuhkan untuk memodelkan visualisasi, menspesifikasikan dan melakukan dokumentasi sistem aplikasi perangkat lunak. Pada UML terdapat berbagai jenis diagram yang memiliki perbedaan fungsi. Pada penelitian ini perancangan UML terdiri dari *sequence diagram* dan *class diagram*. *Sequence Diagram* adalah gambaran perilaku pada *scenario* atau gambaran antar pada *use case diagram* dengan memaparkan *message* yang dikirimkan dan diterima antar objek dalam urutan waktu. *Class Diagram* adalah struktur sistem yang digambarkan dalam kelas-kelas untuk membangun suatu sistem.

#### 2. Perancangan Komponen

Perancangan Komponen menggambarkan perancangan unit terkecil yang dirancang untuk membuat algoritme pada suatu *method* dengan tujuan untuk menjelaskan proses pada fungsi yang terdapat dalam sistem.

#### 3. Perancangan Data

Perancangan data yang akan dibahas pada sistem aplikasi menggunakan *database MySQL* yang akan digambarkan dalam bentuk ERD.

#### 4. Perancangan Antarmuka

Perancangan antarmuka ini merupakan tahapan dalam *wireframing* atau gambaran awal antarmuka dengan merancang tata letak pada layar antarmuka.

### 3.5 Implementasi *Prototyping*

Tahap selanjutnya yaitu melakukan tahapan implementasi *prototyping*. Dalam melakukan *prototyping* harus mengarahkan dengan perancangan yang telah dibuat sebelumnya. *Prototyping* berupa tampilan *user interface* dan menggambarkan beberapa fitur dari sistem aplikasi. Nantinya pada tahap implementasi *prototyping* akan melalui proses iterasi untuk memperoleh hasil dari evaluasi pengguna. Dalam hal ini, tahap iterasi *prototyping* mestinya dilakukan sesuai dengan kebutuhan sistem. Jika sudah sesuai dengan kebutuhan pengguna, selanjutnya akan mendapatkan hasil *prototyping* versi final yang akan dijadikan rujukan implementasi sistem dengan versi akhir atau final.

### 3.6 Evaluasi Pengguna

Proses evaluasi pengguna dilakukan, setelah proses *prototyping* selesai dibuat sebelumnya. Selanjutnya, *prototyping* yang telah selesai sebelumnya diberikan kepada pengguna agar dilakukan evaluasi pengguna. Tujuannya untuk





mengetahui kepuasan dan pengalaman pengguna saat menjalankan *prototyping* pada sistem. Pengguna akan melakukan evaluasi *prototyping* dengan mengacu pada permasalahan kebutuhan yang telah didefinisikan. Setelah itu, hasil dari evaluasi pengguna akan diproses kembali sesuai dengan umpan balik pengguna. Umpan balik atau *feedback* yang diberikan pengguna akan direview oleh peneliti yang nantinya dilakukan perancangan antarmuka prototipe kembali berdasarkan hasil dari umpan balik.

### 3.7 Implementasi

Tahapan implementasi pada sistem dilakukan mengacu pada analisis dan perancangan sistem. Aplikasi yang akan diimplementasi pada dalam penelitian ini menerapkan aplikasi *mobile Android* dengan menggunakan kode mesin pemrograman *java* dan *framework codeigniter* untuk membangun aplikasi *website* dan *PHP* untuk menangani server ataupun proses lainnya. Tahapan implementasi yang akan dilakukan yaitu spesifikasi bagian sistem, implementasi pada kelas, mengimplementasikan *user interfaces* dan mengimplementasikan data. Spesifikasi lingkungan sistem menjelaskan perangkat keras maupun lunak untuk membangun sistem. Implementasi kelas adalah hasil dari realisasi *class diagram* yang sudah dibuat pada tahap perancangan arsitektur, serta mengimplementasikan *user interfaces*.

### 3.8 Pengujian

Setelah proses implementasi pada sistem selesai, maka tahap selanjutnya melakukan pengujian pada sistem. Pengujian dilakukan untuk tahu akan aplikasi yang dibangun dapat bekerja sesuai spesifikasi dari analisis kebutuhan. Pengujian ini bertujuan untuk mengetahui fungsionalitas pada sistem dapat berjalan sesuai kebutuhan pengguna atau tidak. Terdapat 4 pengujian yang dilakukan pada penelitian ini yaitu:

#### 1. Pengujian Unit

Pengujian unit mendeskripsikan pengujian pada unit atau *individual components* dalam sistem aplikasi yang akan diimplementasi. Tujuan pengujian ini untuk memfokuskan komponen unit terkecil dari suatu modul perangkat lunak. Pengujian unit dilakukan dengan cara melihat kode program pada sistem untuk dapat diketahui kesalahan batas modul pada jalur *control* suatu kode program. Teknik pengujian unit yang dilakukan menggunakan *basis path testing*.

#### 2. Pengujian Validasi

Pengujian validasi berfungsi untuk mengetahui sistem yang dibangun sudah sesuai dengan kebutuhan yang didefinisikan. Tujuannya apakah fungsional pada sistem telah berhasil dan valid sesuai dengan masukan serta keluaran yang dihasilkan.





### 3. Pengujian *Usability*

Pengujian *Usability* untuk mengevaluasi pengalaman pengguna dalam menjalankan suatu sistem yang mengacu kemudahan akses pengguna. Tujuannya untuk mengetahui seberapa mudah antarmuka pengguna pada sistem aplikasi yang telah dirancang.

### 4. Pengujian *Compatibility*

Pengujian *compatibility* dilakukan untuk menguji kompatibilitas pada suatu sistem, apakah dapat digunakan pada lingkungan yang berbeda. Pengujian kompatibilitas bertujuan untuk mengungkapkan permasalahan pada aplikasi web sebelum dibuat *online*. Dalam hal ini pengujian kompatibilitas berusaha mengungkap kesalahan atau masalah saat mengeksekusi yang dapat diketahui melalui perbedaan konfigurasi. Pengujian *usability* bertujuan untuk menguji aplikasi dan menentukan apakah *user* dapat menggunakan aplikasi dengan baik dan juga dapat memahami antarmuka dari aplikasi.

## 3.8 Penarikan Kesimpulan dan Saran

Setelah melalui tahapan-tahapan sebelumnya, yaitu pada analisis kebutuhan, perancangan sistem, implementasi, pengujian serta analisis selesai dilakukan, maka tahapan selanjutnya dilakukan penarikan kesimpulan dari hasil yang diperoleh dari penelitian. Kesimpulan akan menjawab hasil dari rumusan masalah yang telah diuraikan. Selanjutnya peneliti membuat saran yang mengacu dari jawaban kesimpulan. Pembuatan saran didasarkan pada penelitian yang nantinya akan ada pengembangan lebih lanjut dengan memberikan masukan, dan juga perbaikan pada sistem agar lebih baik dan dapat memberi manfaat bagi penelitian ke depannya.





## BAB 4 REKAYASA KEBUTUHAN

Pada rekayasa kebutuhan membahas penjelasan secara umum pada sistem yang dikembangkan yaitu aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* berbasis *Android*. Dalam hubungan ini, gambaran umum pada sistem meliputi deskripsi dan gambaran pada sistem, aktivitas untuk mengumpulkan kebutuhan, pengguna atau pemangku kepentingan yang terlibat dalam menggunakan aplikasi dan pembahasan spesifikasi kebutuhan fungsional dan non-fungsional pada sistem.

### 4.1 Gambaran Umum Sistem

Aplikasi *mobile* identifikasi parkir non resmi dengan fitur *geotagging* berdasarkan konsep *crowdsourcing* atau disebut juga dengan aplikasi GoTindak. Aplikasi GoTindak merupakan layanan berbasis aplikasi berupa konsep sosial *crowdsourcing* yang ditujukan untuk melayani pengaduan masyarakat terkait maraknya parkir non resmi di wilayah Kota Malang.

#### 4.1.1 Deskripsi Sistem

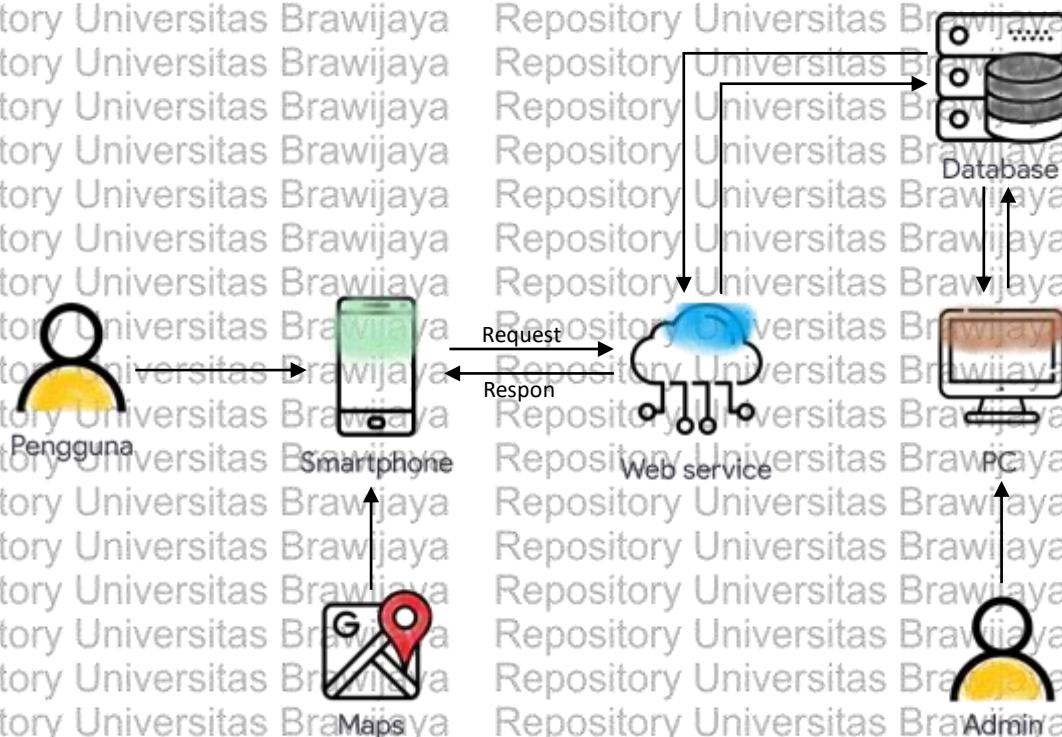
GoTindak adalah sebuah aplikasi *mobile social crowdsourcing* yang akan dikembangkan menggunakan *platform Android*. Di samping itu, aplikasi GoTindak bertujuan sebagai wadah berbagi informasi mengenai pelaporan terkait parkir non resmi. Aplikasi GoTindak ditujukan kepada seluruh elemen masyarakat khususnya di Kota Malang untuk dapat membagikan informasi berupa keluhan atau kritik mengenai pelaporan parkir non resmi di wilayah sekitar. Oleh karena itu, aplikasi GoTindak bekerjasama dengan Dinas Perhubungan Kota Malang pada Manajemen Bidang Perparkiran agar informasi laporan dapat langsung tersampaikan pada pihak terkait, yang nantinya akan ditindaklanjuti. Aplikasi GoTindak terdiri dari 2 aplikasi yang saling terhubung. Aplikasi GoTindak untuk pengguna berbasis *mobile* dan *website* GoTindak untuk admin pada Manajemen Bidang Perparkiran. Secara umum pada aplikasi GoTindak menggunakan teknologi *geotagging* yang dapat membantu pengguna mengetahui lokasi pada saat melaporkan keluhan atau kritik pada parkir non resmi di wilayah Kota Malang. Informasi laporan yang terdapat pada aplikasi GoTindak berisi detail mengenai lokasi, waktu, status laporan, deskripsi dan bukti foto maupun gambar. Selain itu, fitur yang terdapat pada *website* yaitu admin dapat mengubah status laporan yang telah dikirimkan oleh pengguna. Di samping itu, terdapat 4 tahapan status laporan yaitu belum ditindaklanjuti, sedang proses, selesai, dan tidak disetujui.

#### 4.1.2 Gambaran Sistem

Aplikasi Go-Tindak memiliki 2 sub-sistem, aplikasi untuk pengguna dan *website* untuk admin pada Dinas Perhubungan Kota Malang bidang manajemen perparkiran. Secara umum pada aplikasi GoTindak dilakukan oleh pengguna dengan mengambil foto atau gambar pada tempat kejadian adanya keluhan parkir non resmi yang akan dilaporkan, kemudian mengisikan informasi berupa lokasi



serta deskripsi yang sebelumnya pengguna sudah mengaktifkan GPS pada *smartphone*. Setelah berhasil mengambil foto pada objek keluhan parkir non resmi, selanjutnya informasi akan dikirim ke server melalui jaringan internet. Informasi pelaporan akan dikirimkan melalui *web service* yang tersimpan dalam *database*, dan informasi diterima oleh admin. Selanjutnya, admin dapat melakukan tindak lanjut pada laporan yang telah dikirimkan oleh pengguna melalui *web service* yang tersimpan pada *database*. Kemudian pengguna akan menerima status laporan melalui *smartphone*. Berikut merupakan gambaran sistem aplikasi GoTindak yang dapat dilihat dalam gambar 4.1.



Gambar 4.1 Cara Kerja Sistem Aplikasi GoTindak

## 4.2 Elisitasi Kebutuhan

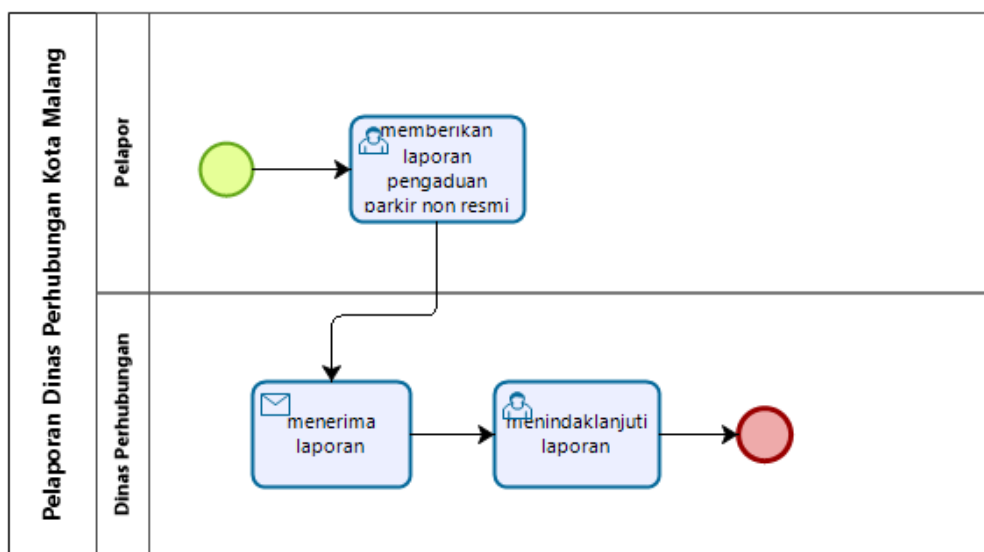
Dalam membangun suatu sistem, peranan pengguna sangat dibutuhkan. Disamping itu, elisitasi kebutuhan merupakan tahapan awal dalam proses analisis kebutuhan yang bertujuan untuk mengumpulkan kebutuhan dari para pemangku kepentingan. Sistem aplikasi diharapkan dapat menyelesaikan permasalahan pengguna yang terjadi pada saat menjalankan aktivitas. Oleh karena itu, pihak pengembang harus menjalankan komunikasi dengan pengguna agar dapat mengetahui kebutuhan pada sistem aplikasi yang dibangun. Dalam hal ini, elisitasi kebutuhan sangat diperlukan pada aplikasi yang dibangun agar masalah pengguna dapat terpecahkan.





Pada tahapan penelitian ini, peneliti menggunakan metode wawancara dan menyebarkan kuesioner kepada para pengguna. Pada gambaran sistem aplikasi telah dijelaskan bahwa terdapat 2 jenis pengguna yaitu sistem aplikasi bagi pengguna *mobile* yang diambil dari beberapa sampling masyarakat dan sistem *website* bagi pengguna pada Dinas Perhubungan Kota Malang. Dalam konteks ini, peneliti melakukan wawancara pada pihak Dinas Perhubungan Kota Malang yang berkaitan dengan manajemen bidang perpajakan dan hasil dari wawancara dapat dilihat pada lampiran A. Selain itu, peneliti menyebarkan kuesioner pada beberapa masyarakat pengguna parkir tepi jalan umum di kecamatan Lowokwaru Kota Malang dan hasil dari kuesioner yang telah peneliti sebarkan dapat dilihat pada lampiran B. Dalam konteks ini hasil yang didapat saat proses penggalan kebutuhan.

Berdasarkan proses penggalan kebutuhan yang telah dilakukan, maka diperoleh proses bisnis *as is* dalam alur pelaporan parkir non resmi antara masyarakat dengan Dinas Perhubungan Kota Malang. Berikut alur proses bisnis akan dijelaskan pada gambar 4.2 di bawah ini.



**Gambar 4.2 Pengaluan Pelaporan Dinas Perhubungan Kota Malang (*as is*)**

Dalam gambar 4.2 dijelaskan alur pemodelan proses bisnis *as is* pada pelaporan Dinas Perhubungan Kota Malang yang dilakukan oleh masyarakat sebagai pelapor. Proses bisnis yang terjadi yaitu pada tahapan awal pelapor memberikan laporan pengaduan mengenai parkir non resmi via sms kepada pihak DISHUB, kemudian DISHUB menerima laporan pelapor. Setelah laporan diterima, selanjutnya pihak DISHUB menindaklanjuti laporan dari masyarakat. Terkadang pihak DISHUB ada yang tidak menindaklanjuti laporan maupun tidak dan pelaporpun tidak mengetahui apakah laporan mereka sudah ditindaklanjuti atau belum.





### 4.3 Identifikasi Aktor

Pada tahap identifikasi aktor bertujuan untuk mengetahui aktor siapa saja yang terlibat dalam aplikasi GoTindak. Dalam hal ini, identifikasi aktor memberikan gambaran mengenai interaksi aktor terhadap sistem yang akan digunakan. Pada aplikasi GoTindak terdapat 3 aktor yang terlibat yaitu pengguna, pelapor dan administrator. Aktor yang terlibat dalam aplikasi GoTindak akan dijelaskan pada tabel 4.1.

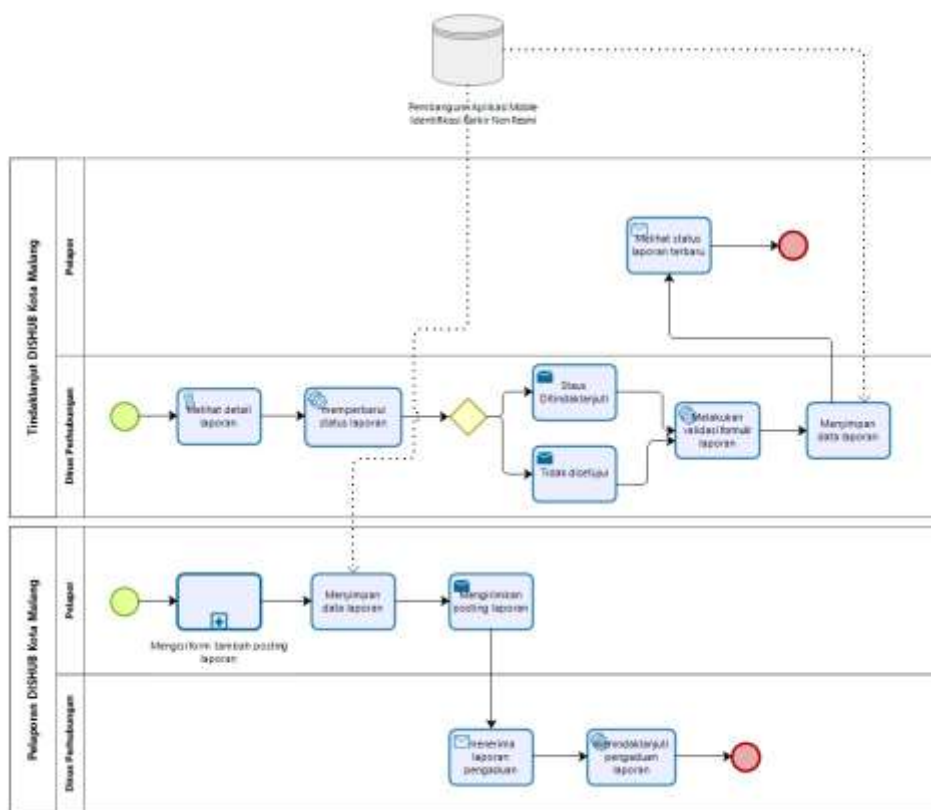
Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Pengguna sebagai aktor atau masyarakat yang menggunakan aplikasi GoTindak pada <i>smartphone Android</i> , yang sudah melakukan register dalam sistem tetapi belum <i>login</i> .
Pelapor	Pelapor sebagai aktor atau masyarakat yang telah mengakses register dan <i>login</i> pada aplikasi GoTindak, sehingga dapat melakukan proses pelaporan pada sistem terkait identifikasi parkir non resmi.
Administrator	Administrator sebagai pemangku kepentingan yang memiliki hak akses penuh untuk mengelola data pelaporan terkait identifikasi parkir non resmi.

### 4.4 Analisis Kebutuhan Sistem

Berdasarkan hasil elisitasi yang diperoleh proses bisnis *to be* yang dibuat setelah adanya sistem, maka dalam penelitian ini mengusulkan proses bisnis dengan mengembangkan suatu sistem. Pelaporan parkir non resmi yang akan dikembangkan menggunakan *platform Android*. Di samping itu, sistem ini ditujukan kepada seluruh elemen masyarakat khususnya di Kota Malang untuk dapat membagikan informasi berupa keluhan atau kritik mengenai pelaporan parkir non resmi di wilayah sekitar, yang bekerjasama dengan Dinas Perhubungan Kota Malang pada Manajemen Bidang Perparkiran agar informasi laporan dapat langsung tersampaikan pada pihak terkait, yang nantinya akan ditindaklanjuti. Dari kegiatan yang telah dijabarkan, maka usulan sistem yang akan dibuat akan digambarkan pada pemodelan bisnis pada Gambar 4.3 di bawah ini.





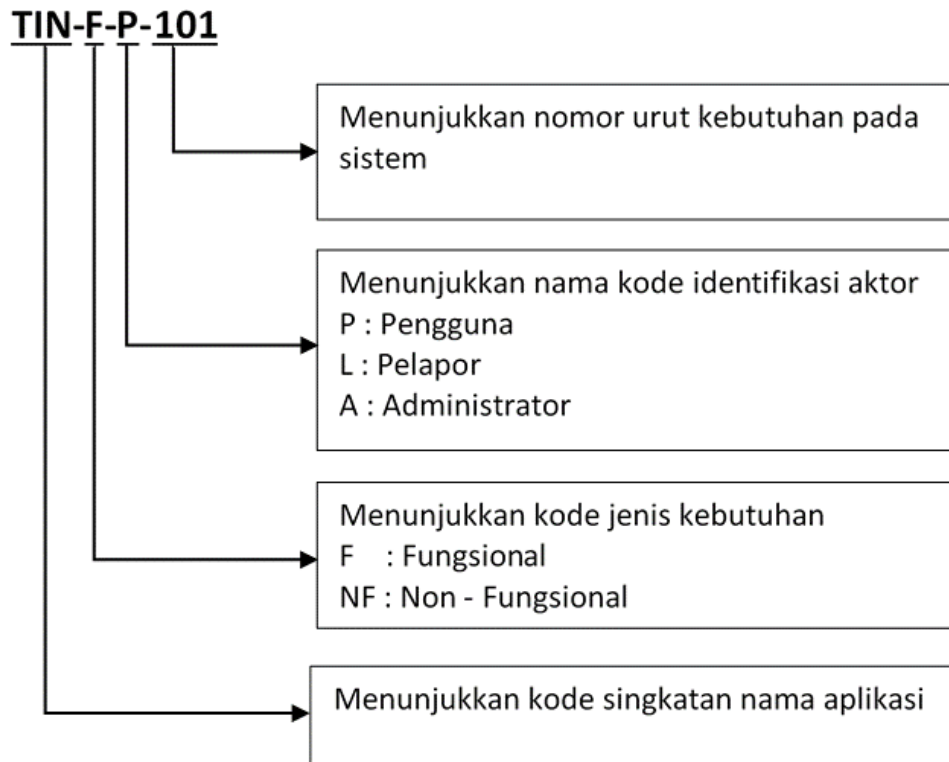
**Gambar 4.3 Usulan Sistem yang akan dibuat**

Untuk gambar 4.3 merupakan pemodelan proses bisnis usulan sistem yang akan dikembangkan. Pertama diawali dengan pelapor mengisi formulir laporan pengaduan parkir non resmi pada aplikasi GoTindak. Data laporan yang dikirim akan diterima oleh pihak admin pada Dinas Perhubungan Kota Malang. Selanjutnya admin melihat detail laporan yang dikirimkan pelapor pada sistem website. Admin dapat mengubah status laporan menjadi status ditindaklanjuti ataupun tidak disetujui. Setelah itu, admin melakukan validasi laporan dan menyimpan data laporan yang telah diperbarui. Sehingga pelapor dapat melihat status laporan yang telah diperbarui.

Selanjutnya melakukan analisis kebutuhan untuk mengetahui kebutuhan apa saja yang terdapat pada sistem. Nantinya kebutuhan tersebut akan dikembangkan menjadi sebuah sistem. Tahapan analisis kebutuhan pada sistem sangat penting dalam tahapan SDLC. Agar proses analisis kebutuhan semakin mudah maka membutuhkan dua jenis kebutuhan pada sistem. Terdapat dua jenis kebutuhan pada sistem yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional mendeskripsikan proses atau interaksi yang dapat dilakukan oleh sistem, sehingga kebutuhan fungsional harus mampu mempresentasi tujuan dari permasalahan pengguna yang telah diperoleh sebelumnya. Sedangkan pada kebutuhan non-fungsional mendeskripsikan kebutuhan yang secara langsung tidak berkaitan dengan interaksi pada sistem yang diberikan kepada pengguna. Di samping itu untuk mendefinisikan kebutuhan pada sistem perlu menggunakan kode identitas pada suatu kebutuhan. Tujuan diberikan kode identitas pada



kebutuhan sistem untuk mempermudah proses dokumentasi serta identifikasi pada suatu kebutuhan. Untuk aturan pemberian nomor, akan diberi kode TIN-F-P-101 setiap kodenya memiliki arti, TIN menunjukkan kode nama sistem aplikasi yaitu GoTindak, F menunjukkan kode untuk kebutuhan fungsional, P menunjukkan identifikasi aktor yang terlibat contohnya Pelapor, 101 merupakan kode nomor kebutuhan pada sistem. Untuk lebih jelasnya berikut penjelasan struktur penamaan kode pada kebutuhan fungsional sistem yang dijelaskan dalam gambar 4.2 sebagai berikut:



Gambar 4.4 Kode Kebutuhan





#### 4.4.1 Kebutuhan Fungsional Iterasi Pertama

Pada sub bab kebutuhan fungsional akan dijelaskan mengenai kebutuhan sistem, nantinya akan menjadi fitur pada aplikasi GoTindak. Untuk mendetailkan kebutuhan fungsional pada sistem tergantung pada setiap spesifikasi yang dimiliki oleh kebutuhan fungsional sistem tersebut. Oleh sebab itu, kebutuhan fungsional ini merupakan fitur utama pada sistem yang akan digunakan oleh aktor. Simbol F mendeskripsikan kebutuhan fungsional pada sistem. Dalam Tabel 4.2 akan dijelaskan iterasi pertama pada aktor.

Tabel 4.2 Kebutuhan Fungsional Pengguna Iterasi Pertama

No	Kode	Nama Fungsi	Deskripsi
1.	TIN-F-P-01	<i>Login</i>	<p>Sistem mampu menyediakan layanan <i>login</i> bagi aktor/pengguna yang sudah terdaftar sebelumnya pada aplikasi GoTindak.</p> <p>1.1 Pengguna harus mengisi <i>formulir login</i> yang sudah terdaftar dengan memasukkan data pada <i>field email</i> dan <i>password</i>.</p>
2.	TIN-F-P-02	<i>Register</i>	<p>Sistem mampu menyediakan layanan register atau mendaftar sebagai pengguna baru pada sistem.</p> <p>2.1 Pengguna harus mengisi <i>formulir register</i> yang disediakan oleh sistem dengan memasukkan data pada <i>field</i> nama lengkap, nomor <i>handphone</i>, <i>username</i>, <i>email</i>, <i>password</i>, dan <i>confirm password</i>. <i>Field</i> tersebut wajib diisi oleh pengguna.</p>





Tabel 4.2 Kebutuhan Fungsional Pelapor Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
1.	TIN-F-L-01	Membuat laporan parkir	<p>Sistem mampu menyediakan fungsi bagi pelapor untuk menambahkan data informasi laporan identifikasi parkir non resmi.</p> <p>1.1 Pelapor harus mengisi <i>formulir</i> tambah laporan yang disediakan oleh sistem dengan menambahkan data pada <i>field</i> gambar, kategori, deskripsi dan lokasi kejadian. <i>Field</i> tersebut harus diisi lengkap oleh pelapor.</p>
2.	TIN-F-L-02	Melihat daftar laporan	<p>Sistem mampu menyediakan fitur bagi pelapor untuk menampilkan seluruh daftar laporan pada sistem.</p> <p>2.1 Sistem mampu menampilkan seluruh informasi laporan. Data informasi yang ditampilkan meliputi nama pelapor, foto profil pelapor, waktu laporan, lokasi kejadian, deskripsi, jumlah <i>like</i>, gambar, status laporan, dan kategori laporan.</p>
3.	TIN-F-L-03	Melihat detail laporan	<p>Sistem dapat menyediakan fungsi bagi pelapor untuk menampilkan detail informasi.</p> <p>3.1 Sistem harus dapat menampilkan detail informasi laporan identifikasi parkir, data informasi yang ditampilkan meliputi: foto profil pelapor, nama pelapor, gambar deskripsi, lokasi kejadian, deskripsi, tanggal laporan, status laporan, kategori, jumlah like, dan komentar.</p>





Tabel 4.2 Kebutuhan Fungsional Pelapor Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
4.	TIN-F-L-04	Memberi dukungan laporan	<p>Sistem dapat menyediakan fungsi memberi dukungan atau <i>vote</i> pada laporan yang dilakukan oleh pengguna.</p> <p>4.1 Sistem harus dapat menampilkan jumlah dukungan, serta melakukan <i>upvote</i> dan <i>downvote</i> pada laporan pengguna.</p>
5.	TIN-F-L-05	Melihat riwayat laporan	<p>Sistem dapat menyediakan fungsi untuk menampilkan informasi riwayat atau <i>history</i> laporan yang sudah dikirimkan oleh pengguna.</p> <p>5.1 Sistem harus dapat menampilkan informasi riwayat yang sudah pelapor kirimkan, dengan menampilkan data informasi meliputi status laporan, kategori laporan, gambar, deskripsi, dan lokasi kejadian.</p>
6.	TIN-F-L-06	Melihat profil	<p>Sistem mampu menyediakan fitur untuk menampilkan profil pelapor.</p> <p>6.1 Sistem mampu menampilkan informasi profil pelapor, data yang ditampilkan meliputi foto profil, nama lengkap, nomor <i>handphone</i>, <i>username</i>, <i>email</i>, dan <i>password</i>.</p>
7.	TIN-F-L-07	Mengedit profil	<p>Sistem mampu menyediakan fungsi untuk mengedit profil atau akun pelapor yang sudah terdaftar.</p> <p>7.1 Sistem mampu menyediakan fitur untuk mengubah profil pelapor, profil pelapor yang dapat diubah yaitu pada <i>field</i> nama lengkap, nomor <i>handphone</i>, <i>username</i> dan <i>email</i>.</p>





Tabel 4.2 Kebutuhan Fungsional Pelapor Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
8.	TIN-F-L-08	Mengunggah foto profil	<p>Sistem mampu menyediakan fungsi untuk mengunggah foto profil pelapor.</p> <p>8.1 Sistem mampu menyediakan fungsi untuk mengunggah foto profil pelapor berupa <i>file</i> yang berekstensi <i>image</i>.</p>
9.	TIN-F-L-09	Mengedit <i>password</i>	<p>Sistem dapat menyediakan fungsi untuk mengedit <i>password</i> pelapor.</p> <p>9.1 Sistem mampu menyediakan fitur untuk mengedit <i>password</i> pelapor dengan <i>mengisikan field new password</i> dan <i>confirm new password</i>.</p>
10.	TIN-F-L-10	Menambah komentar	<p>Sistem dapat menyediakan fungsi untuk menambahkan komentar pada laporan pengguna.</p> <p>10.1 Sistem mampu menyediakan fungsi untuk menambahkan komentar dengan <i>mengisikan field tulis komentar</i> pada setiap laporan ketika pengguna memilih tombol detail laporan.</p>
11	TIN-F-L-11	Melihat komentar	<p>Sistem mampu menyediakan fungsi untuk menampilkan daftar komentar.</p> <p>11.1 Sistem mampu menyediakan fungsi menampilkan komentar pengguna pada setiap laporan pengguna, dengan data nama pengguna, foto profil pengguna, dan tulisan komentar pengguna.</p>
12	TIN-F-L-12	<i>Logout</i>	<p>Sistem harus dapat menyediakan fungsi <i>logout</i> pada sistem, sehingga aktor dapat keluar dari sistem.</p>





Tabel 4.2 Kebutuhan Fungsional Pelapor Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
13.	TIN-F-L-13	Memilih fitur <i>camera</i>	Sistem dapat menyediakan fungsi untuk menampilkan fitur <i>camera</i> pada saat mengambil foto. 13.1 Sistem harus dapat menyediakan fungsi menampilkan fitur <i>camera</i> ketika pengguna memilih button ambil <i>camera</i> maka sistem harus dapat mengambil foto berekstensi <i>image</i> .
14.	TIN-F-L-14	Memilih fitur <i>gallery</i>	Sistem dapat menyediakan fungsi untuk tampil fitur <i>gallery</i> pada saat memilih photo dari fitur <i>gallery</i> . 14.1 Sistem harus dapat menyediakan fungsi menampilkan fitur <i>gallery</i> ketika pengguna memilih button <i>gallery</i> maka sistem harus dapat menampilkan foto berekstensi <i>image</i> dari fitur <i>gallery</i> .

Tabel 4.3 Kebutuhan Fungsional Administrator Iterasi Pertama

No	Kode	Nama Fungsi	Deskripsi
1.	TIN-F-A-01	<i>Login</i>	Sistem mampu menyediakan layanan <i>login</i> bagi administrator yang sudah terdaftar sebelumnya pada <i>website</i> GoTindak. 1.1 Admin harus mengisi <i>formulir login</i> yang sudah terdaftar sebelumnya pada <i>website</i> GoTindak.





Tabel 4.3 Kebutuhan Fungsional Administrator Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
2.	TIN-F-A-02	Melihat jumlah laporan	<p>Sistem mampu menyediakan fungsi untuk menampilkan jumlah data laporan dalam sistem.</p> <p>2.1 Sistem mampu menyediakan fungsi untuk tampilkan jumlah total laporan, total pengguna, total laporan belum ditindaklanjuti, laporan sedang proses, laporan selesai dan laporan tidak disetujui.</p>
3.	TIN-F-A-03	Melihat laporan belum ditindaklanjuti	<p>Sistem dapat menyediakan fungsi untuk menampilkan data laporan belum ditindaklanjuti.</p> <p>3.1 Sistem harus dapat menampilkan data laporan dengan status belum ditindaklanjuti dalam bentuk tabel dengan informasi yang terdiri dari id pelapor, deskripsi, foto, tanggal, status laporan, dan <i>button</i> proses.</p>
4.	TIN-F-A-04	Melihat detail laporan belum ditindaklanjuti	<p>Sistem mampu menyediakan fungsi menampilkan detail laporan belum ditindaklanjuti.</p> <p>4.1 Sistem harus dapat menampilkan detail laporan belum ditindaklanjuti dalam bentuk <i>form</i> dengan rincian detail informasi status, kategori, deskripsi, lokasi serta tombol <i>button cancel</i> dan <i>save</i>.</p>
5.	TIN-F-A-05	Mengubah status laporan belum ditindaklanjuti	<p>Sistem dapat menyediakan fungsi untuk mengubah status laporan belum ditindaklanjuti.</p> <p>5.1 Sistem mampu menyediakan <i>form</i> untuk mengubah data laporan belum ditindaklanjuti yang terdiri dari status laporan dan kategori laporan.</p>





Tabel 4.3 Kebutuhan Fungsional Administrator Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
6.	TIN-F-A-06	Mencari status laporan belum ditindaklanjuti	Sistem dapat menyediakan fungsi untuk melakukan pencarian pada status laporan belum ditindaklanjuti berdasarkan id pelapor, deskripsi dan alamat.
7.	TIN-F-A-07	Melihat laporan sedang proses	Sistem dapat menyediakan fungsi untuk untuk menampilkan data laporan sedang proses. 7.1 Sistem harus dapat menampilkan data laporan dengan status sedang proses dalam bentuk <i>datatable</i> dengan informasi yang terdiri dari id pelapor, deskripsi, foto, tanggal, status laporan, dan <i>button</i> proses
8.	TIN-F-A-08	Melihat detail laporan sedang proses	Sistem mampu menyediakan fungsi untuk menampilkan detail laporan sedang proses. 8.1 Sistem mampu menampilkan detail laporan sedang proses dalam bentuk <i>form</i> dengan rincian detail informasi status, kategori, deskripsi, lokasi, serta tombol <i>button cancel</i> dan <i>save</i> .
9.	TIN-F-A-09	Mengubah status laporan sedang proses	Sistem dapat menyediakan fungsi untuk mengubah status laporan sedang proses. 9.1 Sistem harus dapat menyediakan <i>form</i> untuk mengubah data laporan sedang proses yang terdiri dari status laporan dan kategori laporan.
10.	TIN-F-A-10	Mencari status laporan sedang proses	Sistem dapat menyediakan fungsi untuk melakukan pencarian data status laporan sedang proses berdasarkan id pelapor, deskripsi, dan alamat.





Tabel 4.3 Kebutuhan Fungsional Administrator Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
11.	TIN-F-A-11	Melihat laporan selesai	<p>Sistem dapat menyediakan fungsi untuk menampilkan data laporan selesai.</p> <p>11.1 Sistem harus dapat menampilkan data laporan dengan status selesai dalam bentuk <i>datatable</i> dengan informasi yang terdiri dari id pelapor, deskripsi, foto, tanggal, status laporan, dan button proses</p>
12.	TIN-F-A-12	Melihat detail laporan selesai	<p>Sistem mampu menyediakan fungsi untuk menampilkan detail laporan selesai</p> <p>12.1 Sistem harus dapat menampilkan detail laporan sedang proses dalam bentuk <i>form</i> dengan rincian detail informasi status, kategori, deskripsi, lokasi kejadian serta tombol <i>button cancel</i> dan <i>save</i>.</p>
13.	TIN-F-A-13	Mencari laporan selesai	<p>Sistem dapat menyediakan fungsi untuk melakukan pencarian data status laporan sedang proses berdasarkan id pelapor, deskripsi, dan alamat.</p>
14.	TIN-F-A-14	Melihat laporan tidak disetujui	<p>Sistem dapat menyediakan fungsi untuk menampilkan data laporan tidak disetujui.</p> <p>14.1 Sistem harus dapat menampilkan data laporan dengan status selesai dalam bentuk <i>datatable</i> dengan informasi yang terdiri dari id pelapor, tanggal, status laporan, dan button proses.</p>





Tabel 4.3 Kebutuhan Fungsional Administrator Iterasi Pertama (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
15.	TIN-F-A-15	Mencari laporan tidak disetujui	Sistem mampu menyediakan fungsi untuk pencarian data status laporan tidak disetujui berdasarkan id pelapor, deskripsi, dan alamat.
16.	TIN-F-A-16	Megubah status laporan tidak disetujui	Sistem dapat menyediakan fungsi untuk mengubah status laporan tidak disetujui.  16.1 Sistem mampu menyediakan button untidak disetujui untuk mengubah data laporan dari status tidak disetujui menjadi <i>untidak disetujui</i> .
17.	TIN-F-A-17	<i>Logout</i>	Sistem mampu menyediakan fungsi <i>logout</i> pada sistem, sehingga aktor dapat keluar dari sistem.

#### 4.4.2 Kebutuhan Fungsional Iterasi Kedua

Kebutuhan fungsional iterasi pertama pada sistem aplikasi GoTindak telah terdefinisi. Selanjutnya melakukan uji coba rancangan antarmuka atau prototipe pada aplikasi GoTindak yang diserahkan kepada calon pengguna untuk dilakukan evaluasi-evaluasi. Setelah memperoleh evaluasi pengguna, prototipe yang dibuat sebelumnya akan memberikan *feedback* yang nantinya berguna untuk proses iterasi pada spesifikasi kebutuhan. Dengan kata lain, adanya umpan balik dapat memahami kebutuhan yang dilakukan pada proses iterasi selanjutnya. Hasil proses evaluasi yang akan dideskripsikan dalam tabel. Pada tabel 4.4 dan tabel 4.5 akan dideskripsikan pada kebutuhan fungsional iterasi kedua dari aplikasi GoTindak pada masing-masing aktor.

Tabel 4.4 Kebutuhan Fungsional Pelapor Iterasi Kedua

No	Kode	Nama Fungsi	Deskripsi
1.	TIN-F-L-15	Melihat artikel identifikasi parkir	Sistem mampu menyediakan fungsi untuk menampilkan artikel mengenai identifikasi parkir  1.1 Sistem mampu menampilkan daftar artikel identifikasi parkir. Data informasi yang ditampilkan meliputi judul artikel dan gambar.





Tabel 4.4 Kebutuhan Fungsional Pelapor Iterasi Kedua (Lanjutan)

No	Kode	Nama Fungsi	Deskripsi
2.	TIN-F-L-16	Melihat detail artikel identifikasi parkir	Sistem dapat menyediakan fungsi untuk menampilkan isi detail artikel identifikasi parkir.  2.1 Sistem harus dapat menampilkan detail artikel identifikasi parkir dengan rincian detail informasi judul artikel, gambar dan isi dari artikel.

Tabel 4.5 Kebutuhan Fungsional Administrator Iterasi Kedua

No	Kode	Nama Fungsi	Deskripsi
1.	TIN-F-A-18	Mengupload foto sedang proses	Sistem dapat menyediakan fungsi untuk mengupload foto pada menu status laporan sedang proses.  1.1 Sistem harus dapat menyediakan fungsi untuk mengupload foto ketika laporan sampai tahap status sedang proses, file yang di <i>upload</i> harus berekstensi <i>image</i> .

#### 4.4.3 Kebutuhan Non Fungsional

Kebutuhan Fungsional adalah kebutuhan yang sifatnya tidak wajib untuk dilakukan tetapi sangat dibutuhkan untuk mengetahui kualitas jalannya sistem pada saat beroperasi dilingkungan sistem tersebut. Pada kebutuhan non fungsional diberikan kode huruf NF yang menunjukkan kebutuhan non fungsional dari Aplikasi GoTindak pada tabel 4.6.

Tabel 4.6 Kebutuhan Non Fungsional

No	Kode	Nama Fungsi	Deskripsi
1.	TIN-NF-01	<i>Usability</i>	Sistem mampu memberikan kemudahan akses pada pengguna dengan mendesain aplikasi sedemikian rupa.
2.	TIN-NF-02	<i>Compatibility</i>	Sistem mampu dieksekusi pada <i>browser edge, opera, safari, firefox</i> dan <i>chrome</i> .





## 4.5 Pemodelan Kebutuhan

Tahapan sebelumnya dalam proses analisis kebutuhan sistem didapatkan, maka untuk mengetahui hasil kebutuhan sistem, perlu mengerjakan pemodelan kebutuhan. Kebutuhan tersebut dimodelkan pada tahapan perancangan. Selanjutnya saat memodelkan kebutuhan yang dikembangkan di antaranya *use case diagram* serta *use case scenario*. *Use case diagram* yang akan dibikin dengan tujuan untuk mengetahui fungsi yang dilakukan oleh pemangku kepentingan dalam aplikasi GoTindak. Tahapan selanjutnya dibuat *use case scenario* dengan tujuan untuk mengetahui setiap tahapan kebutuhan pada *use case diagram* yang dibikin sebelumnya.

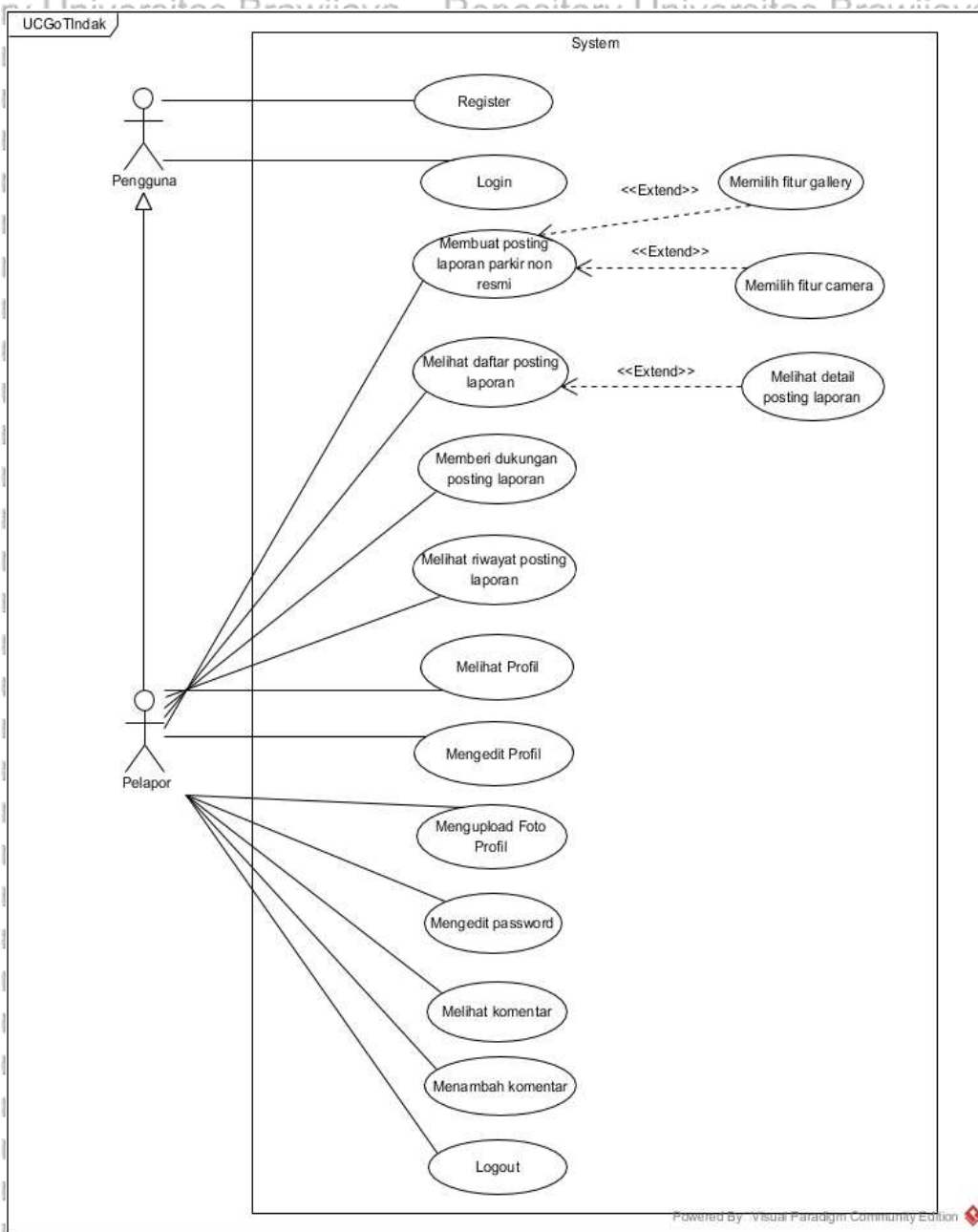
### 4.5.1 Use case Diagram

*Use case diagram* adalah menggambarkan kebutuhan fungsionalitas pada sistem dari pengguna. *Use case diagram* merupakan alat bantu terbaik untuk melakukan pemodelan kebutuhan guna mengetahui aktivitas pengguna sehubungan sebuah sistem, sehingga mudah memahami alur cerita sistem pada proses pengembangan berdasarkan kebutuhan fungsional. Selain itu, untuk menentukan pengguna mana saja yang berhak untuk menjalankan fungsi pada sistem aplikasi. Terkait dengan proses analisis kebutuhan sebelumnya maka akan digambarkan 2 buah *use case diagram*.

#### 4.5.1.1 Use case diagram Pelapor Iterasi Pertama

Gambaran *use case diagram* pada aplikasi *mobile* GoTindak dengan melibatkan pengguna pelapor akan dijelaskan pada gambar 4.5. *Use case diagram* yang akan dijelaskan yaitu diagram pada proses iterasi pertama. Dalam hal ini, penulisan kebutuhan sesuai dengan kebutuhan sebelumnya yang telah didefinisikan.





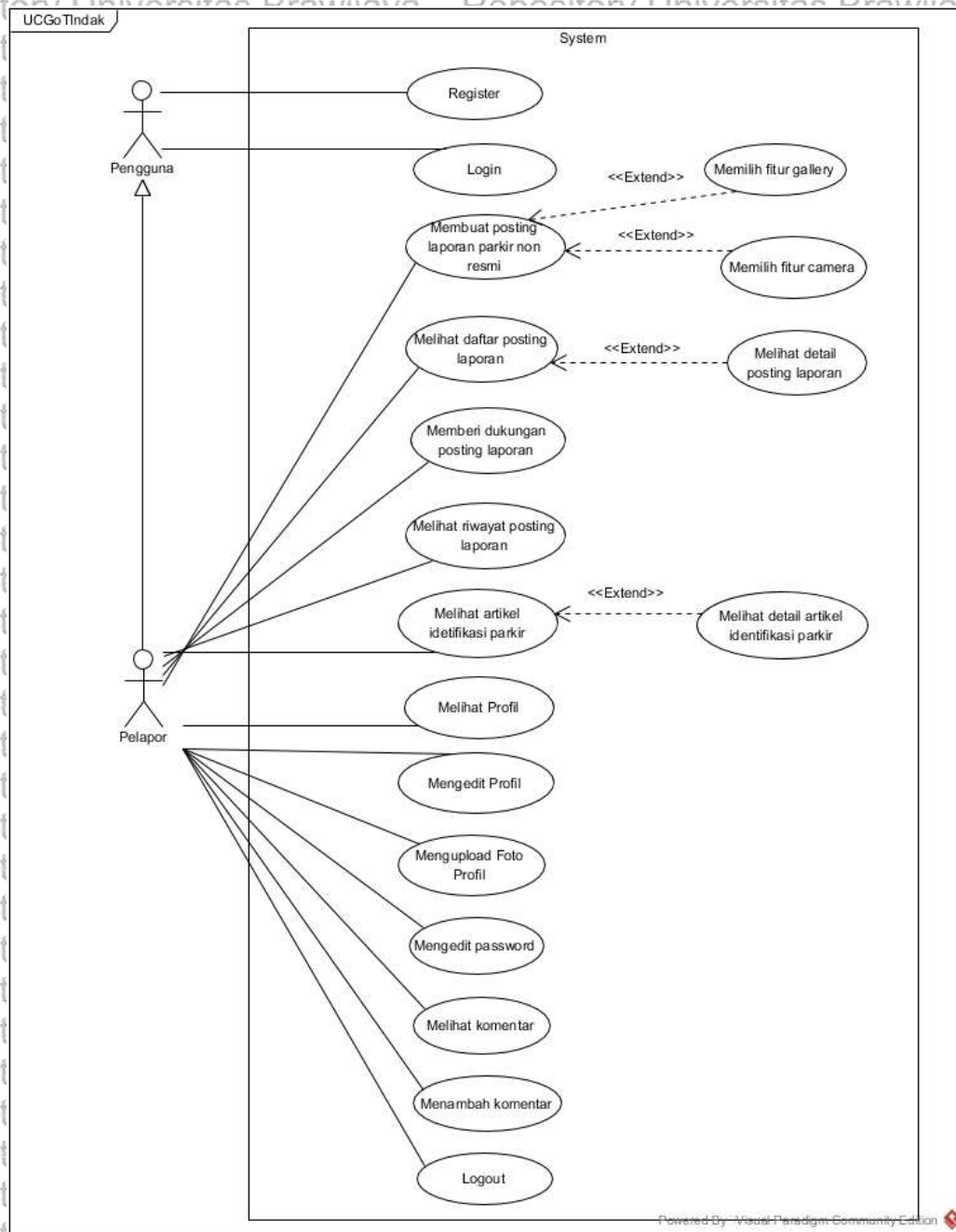
Gambar 4.5 Use case Diagram Pelapor Iterasi Pertama





#### 4.5.1.2 Use case diagram Pelapor Iterasi Kedua

Gambaran *use case diagram* pada aplikasi *mobile* GoTindak dengan melibatkan pengguna pelapor akan dijelaskan pada gambar 4.6. *Use case diagram* yang akan dijelaskan yaitu diagram pada proses iterasi kedua yang terdapat penambahan fungsionalitas untuk dimasukkan ke dalam sistem. Dalam hal ini, kebutuhan yang akan dituliskan sesuai dengan kebutuhan sebelumnya pada iterasi kedua yang telah disempurnakan.

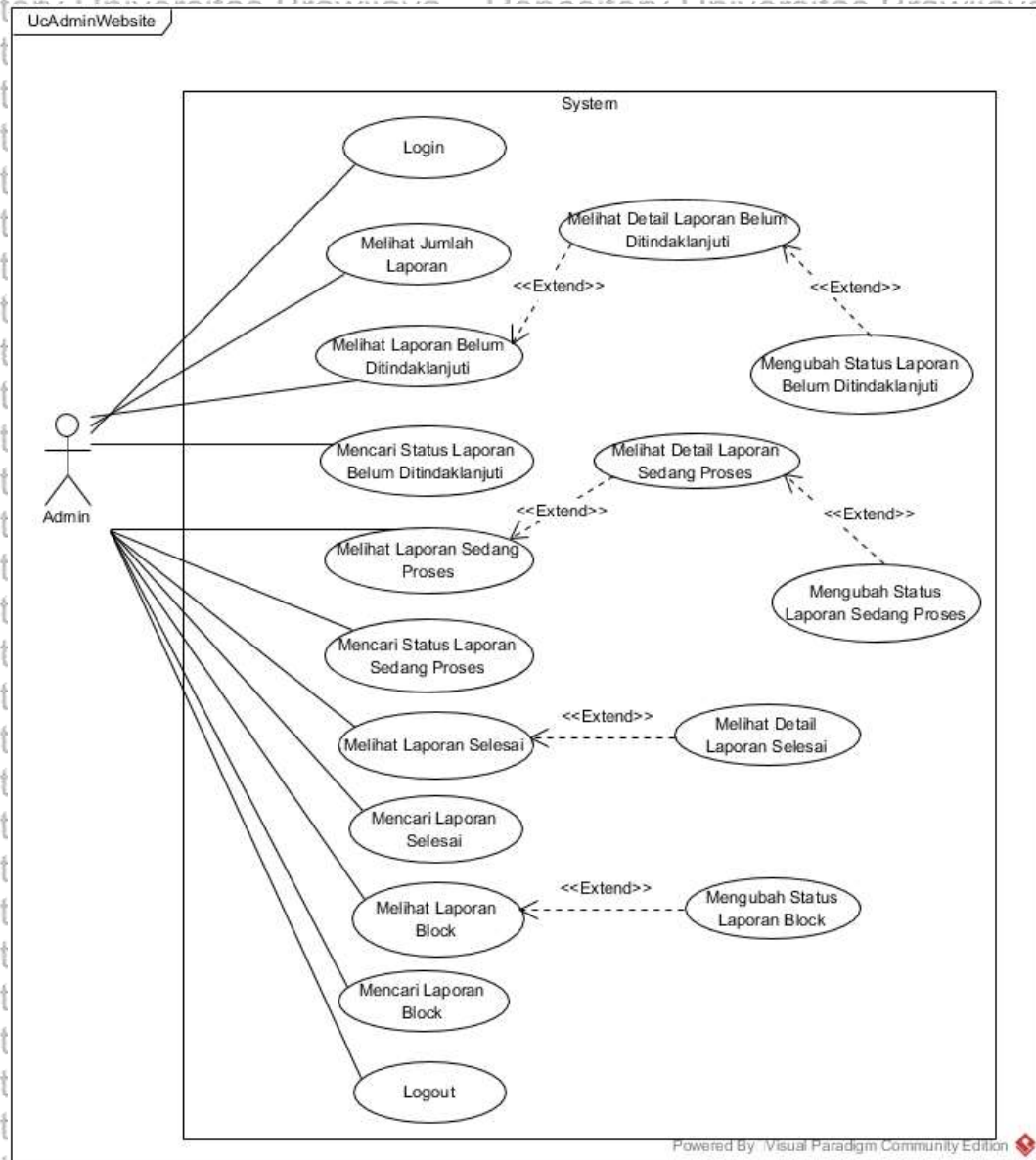


Gambar 4.6 Use case Diagram Pelapor Iterasi Kedua



#### 4.5.1.3 Use case diagram Administrator Iterasi Pertama

Gambaran *use case diagram* pada *website* admin GoTindak dengan melibatkan pengguna administrator akan dijelaskan pada gambar 4.7. *Use case diagram* yang akan dijelaskan yaitu diagram pada proses iterasi pertama. Dalam hal ini, penulisan kebutuhan sesuai dengan kebutuhan sebelumnya yang telah didefinisikan.



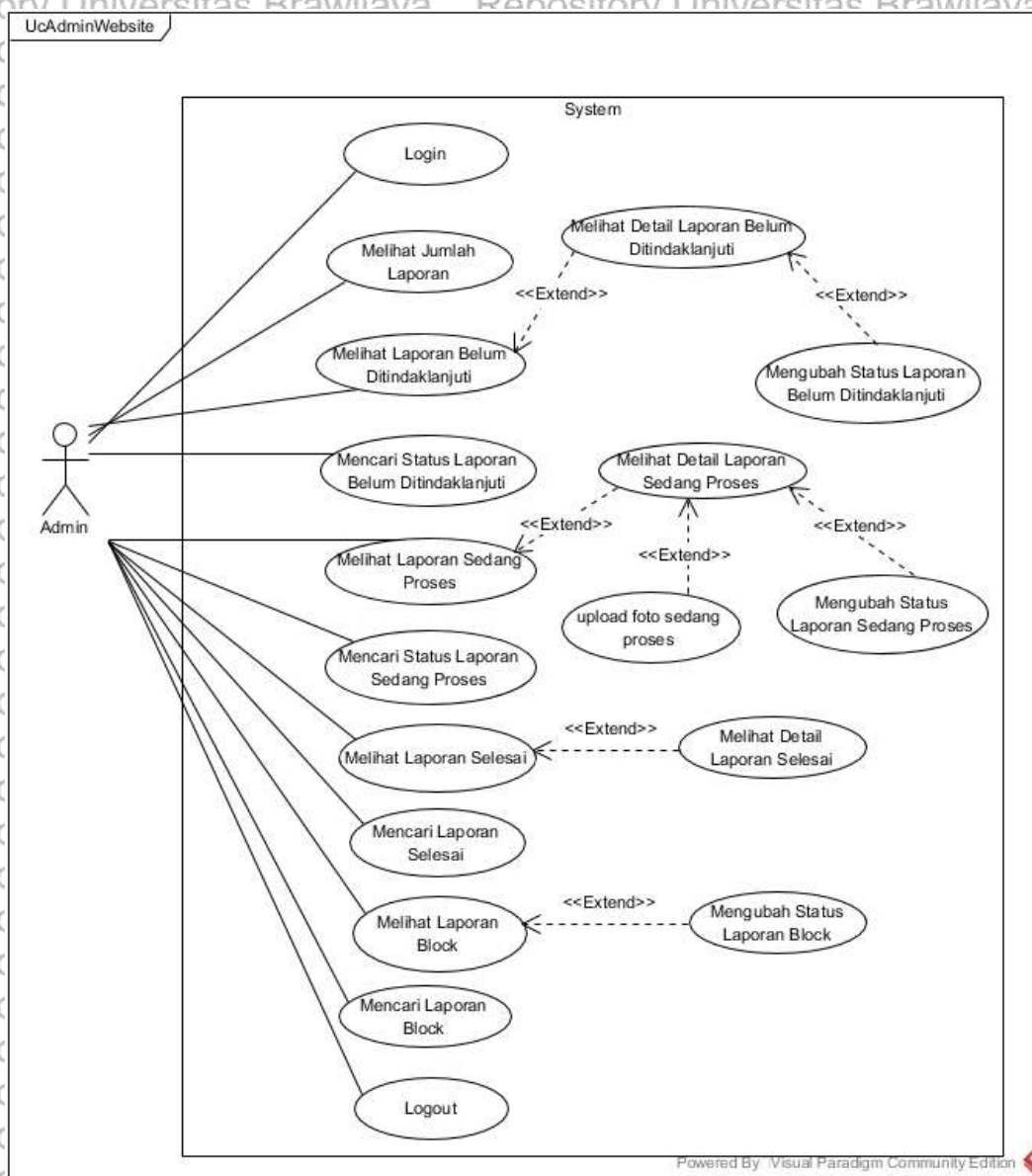
Gambar 4.7 Use case diagram Administrator Iterasi Pertama





#### 4.5.1.4 Use case diagram Administrator Iterasi Kedua

Gambaran *use case diagram* pada *website* admin GoTindak dengan melibatkan pengguna administrator akan dijelaskan pada gambar 4.8. *Use case diagram* yang akan dijelaskan yaitu diagram pada proses iterasi kedua yang terdapat penambahan fungsionalitas untuk dimasukkan ke dalam sistem. Dalam hal ini, kebutuhan yang akan dituliskan sesuai dengan kebutuhan sebelumnya pada iterasi kedua yang telah disempurnakan.



Gambar 4.8 Use case Diagram Administrator Iterasi Kedua





#### 4.5.2 Use case Scenario

Setelah pemodelan *use case diagram* dilakukan, maka tahapan berikutnya melakukan pemodelan *use case scenario* dengan menguraikan tahapan *use case* yang lebih detail. *Use case scenario* menjelaskan fungsi dari *use case* serta fungsi apa yang akan dicapai ketika *use case* dijalankan oleh aktor. Dengan demikian, *use case scenario* mendeskripsikan jalannya *scenario* pada fungsionalitas *use case* yang ditulis sebelumnya. Dalam *use case scenario* akan dispesifikasikan dalam tahap analisis yang terdiri dari beberapa komponen standar *use case scenario* yaitu nama *use case* yang dideskripsikan, kode kebutuhan, aktor, tujuan, *Pre-Condition*, *Main Flow*, *Alternative Flow*, dan *post condition*.

##### 4.5.2.1 Use case scenario Pelapor Iterasi Pertama

Berikut penjelasan *use case scenario* dari masing-masing *use case* yang melibatkan aktor pelapor dan telah didefinisikan sebelumnya. Definisi *use case scenario* akan dijelaskan dalam tabel 4.7 sampai dengan tabel 4.22.

#### 1. Use case scenario Register

Tabel 4.7 Use case scenario Register iterasi pertama

<b>Use case</b>	Register	
<b>Kode Kebutuhan</b>	TIN-F-P-02	
<b>Aktor</b>	Pengguna	
<b>Tujuan</b>	Use case ini sebagai langkah pengguna untuk melakukan pendaftaran akun pada sistem, sehingga pengguna dapat menggunakan fungsionalitas pada sistem	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	1. Sistem menampilkan formulir untuk login	
<b>Main Flow</b>	2. Sistem menampilkan halaman formulir register 5. Sistem melakukan validasi data akun pengguna 6. Sistem akan menyimpan data akun pengguna pada database dan mengarahkan pada halaman awal sistem	1. Aktor melakukan klik pada teks daftar 3. Aktor mengisi form registrasi yang terdiri dari nama lengkap, nomor handphone, <i>username</i> , <i>email</i> , <i>password</i> , <i>confirm password</i> 4. Aktor melakukan klik pada <i>button register</i>



Tabel 4.7 *Use case scenario* Register iterasi pertama (Lanjutan)

<b>Alternative Flow</b>	1. Apabila data yang diinputkan tidak lengkap, maka sistem menampilkan <i>field</i> harus di isi 2. Apabila inputan <i>password</i> dan <i>confirm password</i> tidak sama, maka sistem menampilkan isian <i>password</i> harus sama	
<b>Post Condition</b>	Sistem menampilkan pesan bahwa <i>register</i> sukses	Aktor berhasil melakukan <i>register</i> akun dan memiliki akun untuk <i>login</i>

2. *Use case scenario Login*Tabel 4.8 *Use case scenario Login* iterasi pertama

<b>Use case</b>	<i>Login</i>	
<b>Kode Kebutuhan</b>	TIN-F-P-01	
<b>Aktor</b>	Pengguna	
<b>Tujuan</b>	<i>Use case</i> ini sebagai langkah pengguna agar dapat masuk ke dalam sistem sehingga dapat menggunakan fungsionalitas pada sistem	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	1. Sistem menampilkan <i>formulir</i> untuk <i>login</i>	
<b>Main Flow</b>	3. Sistem melakukan proses validasi akun yang sudah terdaftar pada <i>database</i> sistem 4. Sistem menampilkan halaman awal aplikasi	1. Aktor mengisi <i>form</i> pada <i>login</i> dengan menginputkan <i>email</i> dan <i>password</i> . 2. Aktor menekan tombol <i>login</i>





Tabel 4.8 Use case scenario Login iterasi pertama (Lanjutan)

<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Apabila <i>email</i> dan <i>password</i> yang diinputkan salah maka sistem akan menampilkan pesan <i>email</i> atau <i>password</i> tidak terdaftar</li> <li>2. Jika salah satu data tidak lengkap maka sistem menampilkan pesan bahwa data anda belum diisi</li> </ol>
<b>Post Condition</b>	Aktor berhasil masuk ke dalam aplikasi dan menampilkan halaman beranda

## 3. Use case scenario Membuat Laporan Parkir

Tabel 4.9 Use case scenario Membuat Laporan Parkir Iterasi Pertama

<b>Use case</b>	Membuat	
<b>Kode Kebutuhan</b>	TIN-F-L-01	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor membuat laporan untuk melaporkan adanya keluhan parkir non resmi	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor berada pada halaman beranda</li> </ol>	
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan formulir untuk menambah laporan</li> <li>5. Sistem melakukan penyimpanan data pada <i>database</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol plus pada bagian bawah navigasi</li> <li>3. Aktor mengisi <i>form</i> tambah laporan yang terdiri dari <i>field</i> gambar, kategori, deskripsi dan lokasi kejadian</li> <li>4. Aktor menekan tombol "Post"</li> </ol>



Tabel 4.9 *Use case scenario* Membuat Laporan Parkir Iterasi Pertama (Lanjutan)

<b>Alternative Flow</b>	1. Apabila data <i>form</i> yang diisi tidak lengkap maka sistem akan menampilkan <i>field</i> harus diisi	
<b>Post Condition</b>	Sistem berhasil mengirimkan data laporan pengguna pada <i>database</i> dan tampil pada halaman beranda	Aktor berhasil menambahkan laporan parkir dan kembali kehalaman beranda

4. *Use case scenario* Melihat Daftar LaporanTabel 4.10 *Use case scenario* Daftar Laporan Iterasi Pertama

<b>Use case</b>	Melihat Daftar Laporan	
<b>Kode</b>	TIN-F-L-02	
<b>Kebutuhan</b>		
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melihat laporan dari pengguna lain.	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman beranda
<b>Main Flow</b>	2. Sitem akan melakukan pengambilan data laporan dari <i>database</i>	1. Aktor menekan tombol home yang berada pada navigasi bawah
<b>Alternative Flow</b>	1. Apabila tidak ada koneksi internet pada palikasi, maka sistem tidak akan menampilkan data dari <i>database</i>	
<b>Post Condition</b>	Sistem menampilkan daftar laporan pengguna lain	Aktor berhasil melihat daftar laporan dari pengguna lain





## 5. Use case scenario Melihat Detail Laporan

Tabel 4.11 Use scenario Melihat Detail Laporan Iterasi Pertama

<b>Use case</b>	Melihat Detail Laporan	
<b>Kode Kebutuhan</b>	TIN-F-L-03	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melihat detail laporan pengguna lain	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem.</li> <li>2. Aktor telah masuk pada halaman beranda</li> </ol>
<b>Main Flow</b>	2. Sistem akan melakukan pengambilan data laporan dari <i>database</i>	1. Aktor memilih salah satu <i>cardview</i> daftar laporan
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem menampilkan detail laporan pengguna lain	Aktor berhasil melihat detail daftar laporan yang dipilih

## 6. Use case scenario Memberi Dukungan Laporan

Tabel 4.12 Use case scenario Memberi Dukungan Laporan Iterasi Pertama

<b>Use case</b>	Memberi Dukungan Laporan	
<b>Kode Kebutuhan</b>	TIN-F-L-04	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Aktor memberikan dukungan atau <i>upvote</i> pada laporan pengguna lain.	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman beranda</li> </ol>





Tabel 4.12 *Use case scenario* Memberi Dukungan Laporan Iterasi Pertama (Lanjutan)

<b>Main Flow</b>	2. Sistem akan melakukan pengambilan data dukungan dari <i>database</i>	1. Aktor menekan <i>icon</i> dukungan atau <i>upvote</i> pada <i>card view</i> paoting laporan
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem akan menampilkan jumlah dukungan yang telah di <i>upvote</i>	Aktor berhasil melakukan <i>upvote</i> atau dukungan pada laporan pengguna

#### 7. *Use case scenario* Melihat Riwayat Laporan

Tabel 4.13 *Use case scenario* Melihat Riwayat Laporan Iterasi Pertama

<b>Use case</b>	Melihat Riwayat Laporan	
<b>Kode Kebutuhan</b>	TIN-F-L-05	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Aktor melihat riwayat laporan yang sudah sebelumnya	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman riwayat	
<b>Main Flow</b>	2. Sistem akan melakukan pengambilan data laporan berdasarkan <i>id user</i> dari <i>database</i>	1. Aktor menekan tombol riwayat yang berada pada navigasi bawah
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem akan menampilkan riwayat laporan pengguna	Aktor berhasil melihat riwayat laporan yang sudah di sebelumnya





## 8. Use case scenario Melihat Profil

Tabel 4.14 Use case scenario Melihat Profil Iterasi Pertama

<b>Use case</b>	Melihat Profil	
<b>Kode Kebutuhan</b>	TIN-F-L-06	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melihat profil	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem.
<b>Main Flow</b>	2. Sistem mengambil data profil dari <i>database</i>	1. Aktor memilih menu profil pada navigasi bawah
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem menampilkan halaman untuk data informasi profil pengguna	Aktor berhasil masuk pada halaman profil

## 9. Use case scenario Mengedit Profil

Tabel 4.15 Use case scenario Mengedit Profil Iterasi Pertama

<b>Use case</b>	Mengedit Profil	
<b>Kode Kebutuhan</b>	TIN-F-L-07	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat mengedit profil	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Akor telah berada pada halaman detail informasi profil



Tabel 4.15 *Use case scenario* Mengedit Profil Iterasi Pertama (Lanjutan)

<b>Main Flow</b>	2. Sistem menampilkan data informasi <i>formulir</i> edit profil 5. Sistem akan mengirimkan data profil yang telah diperbarui pada <i>database</i>	1. Aktor memilih tab menu profil pada navigasi bawah 3. Aktor mengisi <i>formulir</i> nama lengkap, nomor handphone, <i>username</i> dan <i>email</i> pada ubah profil 4. Aktor menekan tombol simpan profil
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil memperbarui data profil dan berhasil disimpan pada <i>database</i> , selanjutnya data profil akan ditampilkan pada sistem	Aktor berhasil mengedit profil

10. *Use case scenario* Mengupload Foto ProfilTabel 4.16 *Use case scenario* Mengupload Foto Profil Iterasi Pertama

<b>Use case</b>	Mengupload Foto Profil	
<b>Kode Kebutuhan</b>	TIN-F-L-08	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat memperbarui foto profil	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Akor telah berada pada halaman detail informasi profil
<b>Main Flow</b>	3. Sistem menampilkan pilhan upload foto melalui <i>camera</i> atau melalui <i>gallery</i> 5. Sistem akan mengirimkan data profil yang telah diperbarui pada <i>database</i>	1. Aktor memilih tab menu profil pada navigasi bawah 2. Aktor menekan <i>icon photo</i> 4. Aktor melakukan upload foto profil 5. Aktor menekan tombol simpan profil



Tabel 4.16 *Use case scenario* Mengupload Foto Profil Iterasi Pertama (Lanjutan)

<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil memperbarui foto profil dan berhasil disimpan pada <i>database</i> , selanjutnya data profil akan ditampilkan pada sistem	Aktor berhasil mengupload foto profil

11. *Use case scenario* Mengedit PasswordTabel 4.17 *Use case scenario* Mengedit Password Iterasi Pertama

<b>Use case</b>	Mengedit Password	
<b>Kode Kebutuhan</b>	TIN-F-L-09	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat memperbarui <i>password</i>	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah berada pada halaman detail informasi profil</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan data informasi <i>formulir</i> edit profil</li> <li>5. Sistem akan mengirimkan data <i>password</i> yang telah diperbarui pada <i>database</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor memilih tab menu profil pada navigasi bawah</li> <li>3. Aktor mengisi <i>formulir</i> pada <i>field new password</i> dan <i>confirm new password</i></li> <li>4. Aktor menekan tombol simpan profil</li> </ol>
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil memperbarui data <i>password</i> dan berhasil disimpan pada <i>database</i>	Aktor berhasil mengedit <i>password</i>



12. *Use case scenario* Menambah KomentarTabel 4.18 *Use case scenario* Menambah Komentar Iterasi Pertama

<b>Use case</b>	Menambah Komentar	
<b>Kode Kebutuhan</b>	TIN-F-L-10	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat menambahkan komentar pada laporan pengguna	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor berada pada halaman beranda</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan data laporan</li> <li>4. Sistem menampilkan detail informasi laporan</li> <li>7. Sistem melakukan penyimpanan data komentar pada <i>database</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol plus pada bagian bawah navigasi</li> <li>3. Aktor menekan <i>icon</i> komentar pada salah satu <i>cardview</i> daftar laporan</li> <li>5. Aktor mengisi komentar pada <i>field</i> tulis komentar</li> <li>6. Aktor menekan tombol "Kirim"</li> </ol>
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem berhasil mengirimkan data komentar pengguna pada <i>database</i> dan tampil pada laporan pengguna	Aktor berhasil menambahkan komentar pada laporan parkir pengguna lainnya



13. *Use case scenario* Melihat KomentarTabel 4.19 *Use case scenario* Melihat Komentar Iterasi Pertama

<b>Use case</b>	Melihat Komentar	
<b>Kode Kebutuhan</b>	TIN-F-L-11	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melihat komentar pada laporan pengguna	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor berada pada halaman beranda</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan data laporan</li> <li>4. Sistem menampilkan detail komentar informasi laporan</li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol plus pada bagian bawah navigasi</li> <li>3. Aktor menekan <i>icon</i> komentar pada salah satu <i>cardview</i> daftar laporan</li> </ol>
<b>Alternative Flow</b>	-	
<b>Post Condition</b>	Sistem menampilkan halaman komentar pada poting laporan pengguna	Aktor berhasil melihat komentar pada laporan pengguna

14. *Use case scenario* LogoutTabel 4.20 *Use case scenario* Logout Iterasi Pertama

<b>Use case</b>	Logout	
<b>Kode Kebutuhan</b>	TIN-F-L-12	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melakukan <i>logout</i> pada aplikasi	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem



Tabel 4.20 *Use case scenario Logout* Iterasi Pertama

<b>Main Flow</b>	3. Sistem akan mengeluarkan pelapor dari aplikasi	1. Aktor memilih menu profil pada navigasi bawah 2. Aktor menekan <i>icon logout</i>
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem akan menampilkan halaman awal berupa <i>login</i> pada sistem.	Aktor berhasil <i>logout</i> atau keluar dari aplikasi

15. *Use case scenario Memilih Fitur Camera*Tabel 4.21 *Use case scenario Memilih Fitur Camera* Iterasi Pertama

<b>Use case</b>	Memilih Fitur <i>Camera</i>	
<b>Kode Kebutuhan</b>	TIN-F-L-12	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat memilih fitur <i>camera</i> untuk mengupload foto pada saat membuat laporan	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor berada pada halaman beranda
<b>Main Flow</b>	2. Sistem menampilkan formulir untuk menambah laporan 4. Sistem akan menampilkan <i>camera</i> dari <i>smartphone</i> pelapor 7. Sistem mengirimkan foto kejadian yang telah diambil pada halaman <i>camera</i> pada <i>database</i> .	1. Aktor menekan tombol plus pada bagian bawah navigasi 3. Aktor mengisi <i>form</i> dengan memilih fitur <i>camera</i> 5. Aktor mengambil foto kejadian dari <i>camera smartphone</i> 6. Aktor menekan tombol "Centang" pada <i>camera</i>



Tabel 4.21 *Use case scenario* Memilih Fitur *Camera* Iterasi Pertama (Lanjutan)

<b>Alternative Flow</b>	1. Apabila aktor tidak mengambil foto, maka <i>form</i> akan dianggap tidak lengkap maka sistem menampilkan pesan “Foto belum diambil”	
<b>Post Condition</b>	Sistem berhasil menampilkan foto kejadian dari <i>database</i> dan tampil pada halaman beranda	Aktor berhasil mengambil foto melalui <i>camera</i>

16. *Use case scenario* Memilih Fitur *Gallery*Tabel 4.22 *Use case scenario* Memilih Fitur *Gallery*

<b>Use case</b>	Memilih Fitur <i>Gallery</i>	
<b>Kode Kebutuhan</b>	TIN-F-L-13	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat memilih fitur <i>gallery</i> untuk mengupload foto pada saat membuat laporan	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor berada pada halaman beranda
<b>Main Flow</b>	2. Sistem menampilkan <i>formulir</i> untuk menambah laporan 4. Sistem akan menampilkan <i>gallery</i> dari <i>smartphone</i> pelapor 7. Sistem mengirimkan foto kejadian yang telah diambil pada halaman <i>gallery</i> pada <i>database</i> .	1. Aktor menekan tombol plus pada bagian bawah navigasi 3. Aktor mengisi <i>form</i> dengan memilih fitur <i>gallery</i> 5. Aktor mengambil foto kejadian dari <i>gallery smartphone</i> 6. Aktor menekan tombol “Centang” pada <i>gallery</i>



Tabel 4.22 *Use case scenario* Memilih Fitur *Gallery*

<b>Alternative Flow</b>	1. Apabila aktor tidak mengambil foto <i>form</i> akan dianggap tidak lengkap maka sistem menampilkan pesan "Foto belum diambil"	
<b>Post Condition</b>	Sistem berhasil menampilkan foto kejadian dari <i>database</i> dan tampil pada halaman beranda	Aktor berhasil mengambil foto melalui <i>gallery</i>

#### 4.5.2.2 *Use case scenario* Pelapor Iterasi Kedua

Berikut penjelasan *use case scenario* dari masing-masing *use case* yang melibatkan aktor pelapor dan telah didefinisikan sebelumnya. Definisi *use case scenario* akan dijelaskan dalam tabel 4.23 sampai dengan tabel 4.24.

##### 1. *Use case scenario* Melihat Artikel Identifikasi Parkir

Tabel 4.23 *Use case scenario* Melihat Artikel Identifikasi parkir Iterasi Kedua

<b>Use case</b>	Melihat artikel identifikasi parkir	
<b>Kode Kebutuhan</b>	TIN-F-L-15	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melihat artikel mengenai identifikasi parkir	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman artikel
<b>Main Flow</b>	2. Sitem akan melakukan pengambilan data artikel	1. Aktor menekan tombol artikel yang berada pada navigasi bawah
<b>Alternative Flow</b>	-	
<b>Post Condition</b>	Sistem menampilkan artikel pada halaman informasi identifikasi parkir	Aktor berhasil melihat artikel identifikasi parkir





## 2. Use case scenario Melihat Detail Artikel Identifikasi Parkir

Tabel 4.24 Use case scenario Melihat Detail Indentifikasi Parkir

<b>Use case</b>	Melihat Detail Artikel Identifikasi Parkir	
<b>Kode Kebutuhan</b>	TIN-F-L-16	
<b>Aktor</b>	Pelapor	
<b>Tujuan</b>	Pelapor dapat melihat detail artikel mengenai identifikasi parkir	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman artikel</li> </ol>
<b>Main Flow</b>	2. Sistem akan melakukan pengambilan data artikel	1. Aktor memilih salah satu <i>cardview</i> daftar artikel
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem menampilkan detail artikel pada halaman informasi identifikasi parkir	Aktor berhasil melihat detail artikel identifikasi parkir

### 4.5.2.3 Use case scenario Administrator Iterasi Pertama

Berikut penjelasan *use case scenario* dari masing-masing *use case* yang melibatkan aktor administrator dan telah didefinisikan sebelumnya. Definisi *use case scenario* akan dijelaskan dalam tabel 4.25 sampai dengan tabel 4.42.

#### 1. Use case scenario Login

Tabel 4.25 Use case scenario Login Iterasi Pertama

<b>Use case</b>	<i>Login</i>
<b>Kode Kebutuhan</b>	TIN-F-A-01
<b>Aktor</b>	Administrator
<b>Tujuan</b>	<i>Use case</i> ini sebagai langkah admin agar dapat masuk ke dalam sistem sehingga dapat menggunakan fungsionalitas pada sistem





Tabel 4.25 Use case scenario Login Iterasi Pertama (Lanjutan)

Perlakuan	Sistem	Aktor
<b>Pre-Condition</b>	1. Sistem menampilkan formulir untuk login	
<b>Main Flow</b>	3. Sistem melakukan proses validasi akun yang sudah terdaftar pada database sistem 4. Sistem menampilkan halaman awal aplikasi	1. Aktor mengisi form pada login dengan menginputkan email dan password. 2. Aktor menekan tombol login
<b>Alternative Flow</b>	1. Apabila email dan password yang diinputkan salah maka sistem akan menampilkan pesan email atau password tidak terdaftar 2. Jika salah satu data tidak lengkap maka sistem menampilkan pesan bahwa data anda belum diisi	
<b>Post Condition</b>		Aktor berhasil masuk ke dalam aplikasi dan menampilkan halaman beranda

## 2. Use case scenario Melihat Jumlah Laporan

Tabel 4.26 Use case scenario Melihat Jumlah Laporan Iterasi Pertama

<b>Use case</b>	Melihat Jumlah Laporan	
<b>Kode</b>	TIN-F-A-02	
<b>Kebutuhan</b>		
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat jumlah laporan pada dashboard	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan login pada sistem





Tabel 4.26 *Use case scenario* Melihat Jumlah Laporan Iterasi Pertama (Lanjutan)

<b>Main Flow</b>	2. Sistem akan mengambil data jumlah laporan pada <i>database</i>	1. Aktor memilih menu <i>dashboard</i>
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil menampilkan jumlah laporan yang terdiri dari total <i>user</i> , total laporan, total laporan belum ditindaklanjuti, total laporan sedang diproses, total laporan selesai, total laporan tidak disetujui	Aktor berhasil masuk pada halaman <i>dashboard</i>

3. *Use case scenario* Melihat Laporan Belum Ditindaklanjuti

Tabel 4.27 *Use case scenario* Melihat Laporan Belum Ditindaklanjuti Iterasi Pertama

<b>Use case</b>	Melihat Laporan Belum Ditindaklanjuti	
<b>Kode Kebutuhan</b>	TIN-F-A-03	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat laporan belum ditindaklanjuti	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman awal
<b>Main Flow</b>	2. Sistem mengambil data laporan dengan status belum ditindaklanjuti	1. Aktor memilih menu belum ditindaklanjuti
<b>Alternative Flow</b>	-	-





Tabel 4.27 *Use case scenario* Melihat Laporan Belum Ditindaklanjuti Iterasi Pertama (Lanjutan)

<b>Post Condition</b>	Sistem berhasil menampilkan laporan dari <i>database</i> dalam bentuk <i>datatable</i> yang terdiri dari informasi id pelapor, deskripsi, alamat, foto, tanggal, status dan button proses	Aktor berhasil masuk pada halaman laporan belum ditindaklanjuti
-----------------------	---	---

#### 4. *Use case scenario* Melihat Detail Laporan Belum Ditindaklanjuti

Tabel 4.28 *Use case scenario* Melihat Detail Laporan Belum Ditindaklanjuti Iterasi Pertama

<b>Use case</b>	Melihat Detail Laporan Belum Ditindaklanjuti	
<b>Kode Kebutuhan</b>	TIN-F-A-04	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat detail laporan belum ditindaklanjuti	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman awal</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem mengambil data laporan dengan status belum ditindaklanjuti</li> <li>4. Sistem menampilkan halaman detail laporan belum ditindaklanjuti</li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu belum ditindaklanjuti</li> <li>3. Aktor menekan tombol proses pada daftar laporan yang dipilih</li> </ol>
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem berhasil menampilkan detail laporan dari <i>database</i> dalam bentuk <i>formulir</i> yang terdiri dari informasi status, kategori, deskripsi, dan alamat.	Aktor berhasil masuk pada halaman detail laporan belum ditindaklanjuti





## 5. Use case scenario Mengubah Status Laporan Belum Ditindaklanjuti

Tabel 4.29 Use case scenario Mengubah Status Laporan Belum Ditindaklanjuti Iterasi Pertama

Use case	Mengubah Status Laporan Belum Ditindaklanjuti	
Kode	TIN-F-A-05	
Kebutuhan		
Aktor	Administrator	
Tujuan	Aktor dapat mengubah status laporan belum ditindaklanjuti	
Perlakuan	Sistem	Aktor
Pre-Condition	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman belum ditindaklanjuti</li> <li>3. Aktor telah masuk pada halaman detail laporan</li> </ol>	
Main Flow	<ol style="list-style-type: none"> <li>2. Sistem menampilkan formulir data laporan pada edit laporan belum ditindaklanjuti</li> <li>4. Sistem melakukan validasi terhadap formulir yang telah diperbarui</li> <li>6. Sistem mengirimkan data formulir laporan yang telah diperbarui ke <i>database</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol proses pada daftar laporan yang dipilih</li> <li>3. Aktor memperbarui laporan dengan mengubah status pada laporan halaman belum ditindaklanjuti</li> <li>5. Aktor menekan tombol <i>save change</i></li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Aktor dapat mengganti status laporan menjadi "Di Proses" apabila laporan masih tahap pengerjaan</li> <li>2. Aktor dapat mengganti status laporan menjadi "Selesai" apabila laporan sudah selesai pengerjaan</li> <li>3. Aktor dapat mengubah status laporan menjadi "Tidak disetujui" apabila status</li> </ol>	





		laporan tidak sesuai ketentuan.
<b>Post Condition</b>	Sistem berhasil menampilkan status laporan berdasarkan status dan tersimpan dalam <i>database</i>	Aktor berhasil mengganti status laporan pada halaman belum ditindaklanjuti

#### 6. Use case scenario Mencari Status Laporan Belum Ditindaklanjuti

Tabel 4.30 Use case scenario Mencari Status Laporan Belum Ditindaklanjuti Iterasi Pertama

<b>Use case</b>	Mencari Status Laporan Belum Ditindaklanjuti	
<b>Kode Kebutuhan</b>	TIN-F-A-06	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat mencari status laporan belum ditindaklanjuti	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman awal sistem</li> </ol>	
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem mengambil data laporan dengan status belum ditindaklanjuti</li> <li>4. Sistem akan memproses hasil pencarian kata kunci</li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu belum ditindaklanjuti</li> <li>3. Aktor mengisikan kata kunci pencarian dengan mengetikkan informasi id pelapor, deskripsi, alamat, serta status</li> </ol>
<b>Alternative Flow</b>	-	
<b>Post Condition</b>	Sistem berhasil menampilkan pencarian kata kunci pada halaman belum ditindaklanjuti	Aktor berhasil melakukan pencarian pada halaman status belum ditindaklanjuti





## 7. Use case scenario Melihat Laporan Sedang Proses

Tabel 4.31 Use case scenario Melihat Laporan Sedang Proses Iterasi Pertama

<b>Use case</b>	Melihat Laporan Sedang Proses	
<b>Kode Kebutuhan</b>	TIN-F-A-07	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat laporan sedang proses	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman awal</li> </ol>
<b>Main Flow</b>	2. Sistem mengambil data laporan dengan status sedang proses	1. Aktor memilih menu sedang proses
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil menampilkan laporan dari <i>database</i> dalam bentuk <i>datatable</i> yang terdiri dari informasi id pelapor, deskripsi, alamat, foto, tanggal, status dan button proses	Aktor berhasil masuk pada halaman laporan sedang proses

## 8. Use case scenario Melihat Detail Laporan Sedang Proses

Tabel 4.32 Use case scenario Melihat Detail Laporan Sedang Proses Iterasi Pertama

<b>Use case</b>	Melihat Detail Laporan Sedang Proses	
<b>Kode Kebutuhan</b>	TIN-F-A-08	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat detail laporan sedang proses	





Tabel 4.32 *Use case scenario* Melihat Detail Laporan Sedang Proses Iterasi Pertama (Lanjutan)

Perlakuan	Sistem	Aktor
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman awal</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem mengambil data laporan dengan status sedang proses</li> <li>4. Sistem menampilkan halaman detail laporan sedang diproses</li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu sedang proses</li> <li>3. Aktor menekan tombol proses pada daftar laporan yang dipilih</li> </ol>
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil menampilkan detail laporan dari <i>database</i> dalam bentuk <i>formulir</i> yang terdiri dari informasi status, kategori, deskripsi, dan alamat.	Aktor berhasil masuk pada halaman detail laporan sedang diproses.

### 9. *Use case scenario* Mengubah Status Laporan Sedang Proses

Tabel 4.33 *Use case scenario* Mengubah Status Laporan Sedang Proses Iterasi Pertama

<b>Use case</b>	Mengubah Status Laporan Sedang Proses
<b>Kode Kebutuhan</b>	TIN-F-A-09
<b>Aktor</b>	Administrator
<b>Tujuan</b>	Aktor dapat mengubah status laporan sedang proses





Tabel 4.33 *Use case scenario* Mengubah Status Laporan Sedang Proses Iterasi Pertama (Lanjutan)

Perlakuan	Sistem	Aktor
<b>Pre-Condition</b>		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman sedang proses</li> <li>3. Aktor telah masuk pada halaman detail laporan</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem menampilkan <i>formulir</i> data laporan pada edit laporan sedang proses</li> <li>4. Sistem melakukan validasi terhadap <i>formulir</i> yang telah diperbarui</li> <li>6. Sistem mengirimkan data <i>formulir</i> laporan yang telah diperbarui ke <i>database</i></li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol proses pada daftar laporan yang dipilih</li> <li>3. Aktor memperbarui laporan dengan mengubah status pada laporan halaman sedang proses</li> <li>5. Aktor menekan tombol <i>save change</i></li> </ol>
<b>Alternative Flow</b>		<ol style="list-style-type: none"> <li>1. Aktor dapat mengganti status laporan menjadi "Selesai" apabila laporan sudah selesai pengerjaan</li> <li>2. Aktor dapat mengubah status laporan menjadi "Tidak disetujui" apabila status laporan tidak sesuai ketentuan.</li> </ol>
<b>Post Condition</b>	Sistem berhasil menampilkan status laporan berdasarkan status sedang proses dan tersimpan dalam <i>database</i>	Aktor berhasil mengganti status laporan pada halaman sedang proses





## 10. Use case scenario Mencari Status Laporan Sedang Proses

Tabel 4.34 Use case scenario Mencari Status Laporan Sedang Proses

<b>Use case</b>	Mencari Status Laporan Sedang Proses	
<b>Kode Kebutuhan</b>	TIN-F-A-10	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat mencari status laporan sedang proses	
<b>Perlakuan Pre-Condition</b>	<b>Sistem</b>	<b>Aktor</b>
		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman awal sistem</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem mengambil data laporan dengan status sedang proses</li> <li>4. Sistem akan memproses hasil pencarian kata kunci</li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu sedang proses</li> <li>3. Aktor mengisikan kata kunci pencarian dengan mengetikkan informasi id pelapor, deskripsi, alamat, serta status</li> </ol>
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem berhasil menampilkan pencarian kata kunci pada halaman sedang proses	Aktor berhasil melakukan pencarian pada halaman status sedang proses

## 11. Use case scenario Melihat Laporan Selesai

Tabel 4.35 Use case scenario Melihat Laporan Selesai Iterasi Pertama

<b>Use case</b>	Melihat Laporan Selesai	
<b>Kode Kebutuhan</b>	TIN-F-A-11	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat laporan selesai	



Tabel 4.35 *Use case scenario* Melihat Laporan Selesai Iterasi Pertama (Lanjutan)

Perlakuan	Sistem	Aktor
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman awal
<b>Main Flow</b>	2. Sistem mengambil data laporan dengan status selesai	1. Aktor memilih menu selesai
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem berhasil menampilkan laporan dari <i>database</i> dalam bentuk datatable yang terdiri dari informasi id pelapor, deskripsi, alamat, foto, tanggal, status dan button proses	Aktor berhasil masuk pada halaman laporan selesai

12. *Use case scenario* Melihat Detail Laporan SelesaiTabel 4.36 *Use case scenario* Melihat Detail Laporan Selesai Iterasi Pertama

<b>Use case</b>	Melihat Detail Laporan Selesai	
<b>Kode Kebutuhan</b>	TIN-F-A-12	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat detail laporan selesai	
Perlakuan	Sistem	Aktor
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman awal





**Tabel 4.36 Use case scenario Mencari Laporan Selesai Iterasi Pertama (Lanjutan)**

<b>Main Flow</b>	2. Sistem mengambil data laporan dengan status selesai 4. Sistem menampilkan halaman detail laporan selesai	1. Aktor memilih menu selesai 3. Aktor menekan tombol proses pada daftar laporan yang dipilih
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil menampilkan detail laporan dari <i>database</i> dalam bentuk <i>formulir</i> yang terdiri dari informasi status, kategori, deskripsi, dan alamat.	Aktor berhasil masuk pada halaman detail laporan selesai

### 13. Use case scenario Mencari Laporan Selesai

**Tabel 4.37 Use case scenario Mencari Laporan Selesai Iterasi Pertama**

<b>Use case</b>	Mencari Laporan Selesai	
<b>Kode Kebutuhan</b>	TIN-F-A-13	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat mencari status laporan selesai	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman awal sistem





**Tabel 4.37 Use case scenario Mencari Laporan Selesai Iterasi Pertama (Lanjutan)**

<b>Main Flow</b>	2. Sistem mengambil data laporan dengan status selesai 4. Sistem menampilkan halaman detail laporan selesai	1. Aktor memilih menu selesai 3. Aktor mengisikan kata kunci pencarian dengan mengetikkan informasi id pelapor, deskripsi, alamat, serta status
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem berhasil menampilkan pencarian kata kunci pada halaman selesai	Aktor berhasil melakukan pencarian pada halaman status selesai

#### 14. Use case scenario Melihat Laporan Tidak disetujui

**Tabel 4.38 Use case scenario Melihat Laporan Tidak disetujui Iterasi Pertama**

<b>Use case</b>	Melihat Laporan Tidak disetujui	
<b>Kode Kebutuhan</b>	TIN-F-A-14	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melihat laporan tidak disetujui	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>		1. Aktor telah melakukan <i>login</i> pada sistem 2. Aktor telah masuk pada halaman awal
<b>Main Flow</b>	2. Sistem mengambil data laporan dengan status tidak disetujui	1. Aktor memilih menu tidak disetujui
<b>Alternative Flow</b>	-	-





Tabel 4.38 *Use case scenario* Melihat Laporan Tidak disetujui Iterasi Pertama (lanjutan)

<b>Post Condition</b>	Sistem berhasil menampilkan laporan dari <i>database</i> dalam bentuk <i>datatable</i> yang terdiri dari informasi id pelapor, deskripsi, alamat, foto, tanggal, status dan button proses	Aktor berhasil masuk pada halaman laporan tidak disetujui
-----------------------	---	---

15. *Use case scenario* Mencari Laporan Tidak disetujui

Tabel 4.39 *Use case scenario* Mencari Laporan Tidak disetujui Iterasi Pertama

<b>Use case</b>	Mencari Laporan Tidak disetujui	
<b>Kode Kebutuhan</b>	TIN-F-A-15	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat mencari status laporan tidak disetujui	
<b>Perlakuan Pre-Condition</b>	<b>Sistem</b>	<b>Aktor</b>
		<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman awal sistem</li> </ol>
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>2. Sistem mengambil data laporan dengan status tidak disetujui</li> <li>4. Sistem menampilkan halaman detail laporan tidak disetujui</li> </ol>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu tidak disetujui</li> <li>3. Aktor mengisikan kata kunci pencarian dengan mengetikkan informasi id pelapor, deskripsi, alamat, serta status</li> </ol>
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem berhasil menampilkan pencarian kata kunci pada halaman tidak disetujui	Aktor berhasil melakukan pencarian pada halaman status tidak disetujui





## 16. Use case scenario Mengubah Status Laporan Tidak disetujui

Tabel 4.40 Use case scenario Mengubah Status Laporan Tidak disetujui Iterasi Pertama

<b>Use case</b>	Mengubah Status Laporan Tidak disetujui	
<b>Kode</b>	TIN-F-A-16	
<b>Kebutuhan</b>		
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat mengubah status laporan tidak disetujui	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman sedang proses</li> </ol>	
<b>Main Flow</b>	2. Sistem mengirimkan data formulir laporan yang telah diperbarui ke <i>database</i> .	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>untidak disetujui</i> pada daftar laporan yang dipilih</li> </ol>
<b>Alternative Flow</b>		
<b>Post Condition</b>	Sistem berhasil menampilkan status laporan dan tersimpan dalam <i>database</i>	Aktor berhasil mengganti status laporan pada halaman tidak disetujui

## 17. Use case scenario Logout

Tabel 4.41 Use case scenario Logout Iterasi Pertama

<b>Use case</b>	Logout	
<b>Kode</b>	TIN-F-A-17	
<b>Kebutuhan</b>		
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat melakukan <i>logout</i> pada sistem	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> </ol>	



Tabel 4.41 *Use case scenario Logout Iterasi Pertama (lanjutan)*

<b>Main Flow</b>	3. Sistem akan mengeluarkan admin dari aplikasi	1. Aktor menekan tombol <i>user</i> yang berada pada pojok kanan atas header 2. Aktor menekan <i>icon logout</i>
<b>Alternative Flow</b>	-	-
<b>Post Condition</b>	Sistem akan menampilkan halaman awal berupa <i>login</i> pada sistem.	Aktor berhasil <i>logout</i> atau keluar dari sistem

#### 4.5.2.4 *Use case scenario Administrator Iterasi Kedua*

Berikut penjelasan *use case scenario* dari masing-masing *use case* yang melibatkan aktor administrator dan telah didefinisikan sebelumnya. Definisi *use case scenario* akan dijelaskan dalam tabel 4.45

##### 1. *Use case scenario* Mengupload Foto Sedang Proses

Tabel 4.42 *Use case scenario Mengupload Foto Sedang Proses Iterasi Kedua*

<b>Use case</b>	Mengupload Foto sedang Proses	
<b>Kode Kebutuhan</b>	TIN-F-A-18	
<b>Aktor</b>	Administrator	
<b>Tujuan</b>	Aktor dapat mengupload foto pada halaman sedang proses	
<b>Perlakuan</b>	<b>Sistem</b>	<b>Aktor</b>
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Aktor telah melakukan <i>login</i> pada sistem</li> <li>2. Aktor telah masuk pada halaman sedang proses</li> <li>3. Aktor telah masuk pada halaman detail laporan</li> </ol>	





Tabel 4.42 *Use case scenario* Mengupload Foto Sedang Proses Iterasi Kedua (Lanjutan)

<b>Main Flow</b>	<p>2. Sistem menampilkan <i>formulir</i> data laporan pada edit laporan sedang proses</p> <p>4. Sistem melakukan validasi terhadap <i>formulir</i> yang telah diperbarui</p> <p>6. Sistem mengirimkan data <i>formulir</i> laporan yang telah diperbarui ke <i>database</i></p>	<p>1. Aktor menekan tombol proses pada daftar laporan yang dipilih</p> <p>3. Aktor memperbarui laporan dengan mengupload foto mengenai tindak lanjut laporan</p> <p>5. Aktor menekan tombol <i>save change</i></p>
<b>Alternative Flow</b>		<p>1. Aktor dapat mengganti status laporan menjadi "Selesai" apabila laporan sudah selesai pengerjaan</p> <p>2. Aktor dapat mengubah status laporan menjadi "Tidak disetujui" apabila status laporan tidak sesuai ketentuan.</p>
<b>Post Condition</b>	Sistem berhasil menampilkan foto dan tersimpan dalam <i>database</i>	Aktor berhasil mengupload foto laporan pada halaman sedang proses

#### 4.6 Analisis Data

Tahapan analisis data sangat diperlukan sebelum melakukan pengembangan pada sebuah sistem. Analisis data bertujuan untuk mengetahui struktur data penyimpanan yang akan diimplementasikan pada aplikasi GoTindak. Berdasarkan analisis data yang telah dilakukan maka data yang akan digunakan pada aplikasi GoTindak sebagai berikut:

- Data pengguna yang digunakan pada sistem aplikasi untuk melakukan registrasi, *login*, serta *setting* akun pengguna. Maka data pengguna yang dibutuhkan memiliki atribut berupa id pengguna, nama lengkap, nomor *handphone*, *username*, *email*, *password*, dan foto.
- Data laporan yang digunakan pada sistem aplikasi untuk melakukan laporan dan melihat informasi laporan. Maka data laporan yang dibutuhkan memiliki atribut id, pengunggah, deskripsi, alamat, tanggal, status, kategori, gambar0, catatan dan gambar1.





- c. Data komentar yang digunakan pada sistem aplikasi untuk menambahkan komentar pada laporan dan melihat komentar. Maka data laporan yang dibutuhkan memiliki atribut *id pengguna*, *id laporan*, dan *komentar*.
- d. Data dukungan yang digunakan pada sistem aplikasi untuk menambahkan jumlah dukungan pengguna berupa *upvote*. Maka data dukungan yang dibutuhkan memiliki *atribut id pengguna* dan *id laporan*.
- e. Data admin yang digunakan pada sistem *website* untuk melakukan *login* pengguna. Maka data admin yang dibutuhkan memiliki atribut *id admin*, *username*, *email* dan *password*.





## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini akan membahas perancangan dan implementasi pada aplikasi informasi laporan identifikasi parkir. Sebelum melakukan tahapan implementasi pada sistem, terlebih dahulu melakukan perancangan sistem. Selanjutnya, perancangan sistem ini dilakukan berdasarkan hasil analisis kebutuhan yang telah dispesifikasikan sebelumnya. Dalam hal ini, tujuan dari perancangan sistem sebagai landasan untuk mengimplementasikan sistem. Setelah dilakukan perancangan sistem tahapan selanjutnya yaitu implementasi pada sistem berdasarkan perancangan yang dibuat sebelumnya.

### 5.1 Perancangan Sistem

Perancangan sistem merupakan cara untuk mendefinisikan kebutuhan *stakeholder* pada sistem yang sebelumnya telah dilakukan analisis kebutuhan fungsional. Perancangan sistem juga bertindak sebagai dasar untuk melanjutkan ke tahap implementasi agar meminimalisir terjadinya resiko pada sistem yang dikembangkan. Perancangan pada aplikasi GoTindak dibagi menjadi 5 tahap di antaranya perancangan *class diagram*, perancangan *sequence diagram*, perancangan komponen, perancangan data, dan perancangan antarmuka.

#### 5.1.1 Perancangan Sequence Diagram

Perancangan *Sequence Diagram* menggambarkan interaksi pada sebuah sistem ketika digunakan. Dalam hal ini, perancangan *sequence diagram* menggambarkan perilaku pada sebuah *use case scenario*. Diagram ini menunjukkan aliran pesan dari satu objek ke objek yang lain yang telah didefinisikan kebutuhannya. Komponen utama dalam *sequence diagram* yaitu objek dan pesan. Pada sistem aplikasi GoTindak terdiri dari 2 sub sistem, yaitu aplikasi GoTindak untuk pelapor dan *website* GoTindak untuk administrator. Selanjutnya, pada perancangan *sequence diagram* ini akan digambarkan 4 diagram, yang terdiri dari 2 diagram aplikasi GoTindak dengan aktor pelapor dan 2 diagram *website* GoTindak dengan aktor administrator.

##### 5.1.1.1 Sequence diagram Iterasi Pertama

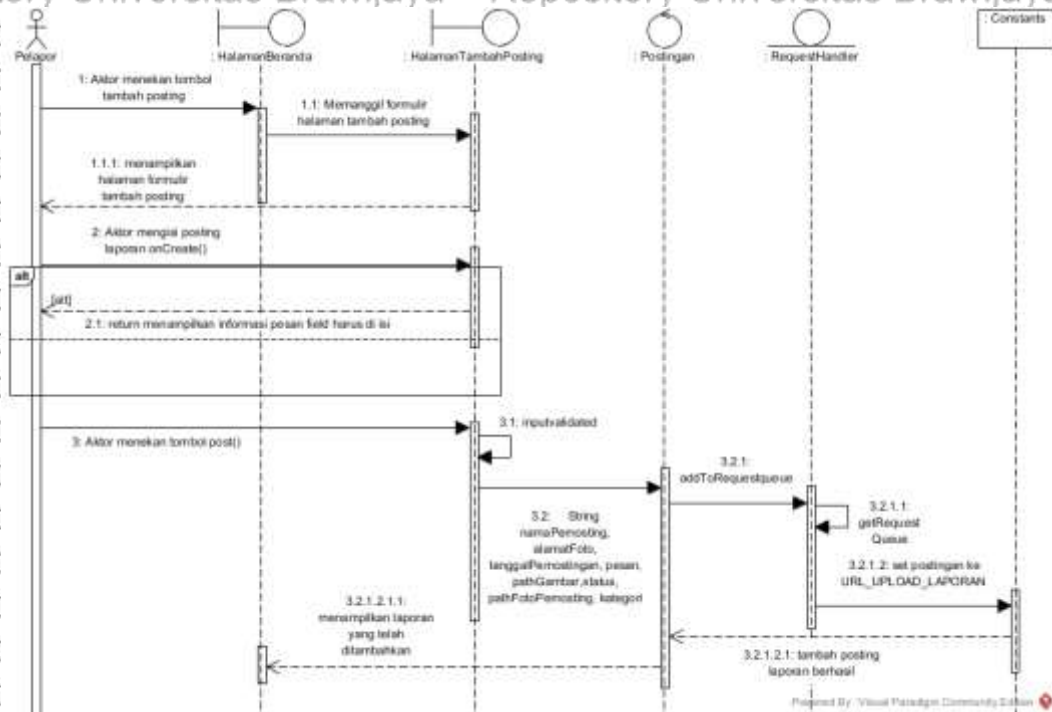
###### 5.1.1.1.1 Membuat Laporan Parkir

*Sequence diagram* membuat laporan menjelaskan bagaimana alur pelapor dalam melakukan interaksi pada sistem aplikasi GoTindak. Terdapat 5 objek pada *sequence* membuat laporan yaitu halaman beranda, halaman tambah laporan, *request handler* dan *constant*. Pertama aktor menekan tombol tambah, kemudian akan memanggil *formulir* tambah laporan. Selanjutnya aktor mengisi laporan dengan memanggil *method onCreate()*, jika aktor tidak mengisi laporan dengan lengkap maka akan menampilkan pesan *field* harus diisi. Sesudah itu aktor menekan tombol post dengan memanggil *method post()* dan *formulir* akan melakukan proses *inputValidate()* yang terdiri dari *string* nama pemosting, alamat





foto, tanggal laporan, pesan, pathGamabr, status, pathFotoPemosting dan kategori. Selanjutnya data laporan dibuat objek dengan memanggil method *addToRequestgueue* dan *getRequestQueue*. Kemudian data laporan dikirimkan pada service *url\_upload\_laporan* dengan mengembalikan bentuk data menjadi *JSON*, jika berhasil maka data laporan akan ditampilkan dalam halaman beranda. *Sequence diagram* membuat laporan akan dijelaskan pada keterangan gambar di bawah.

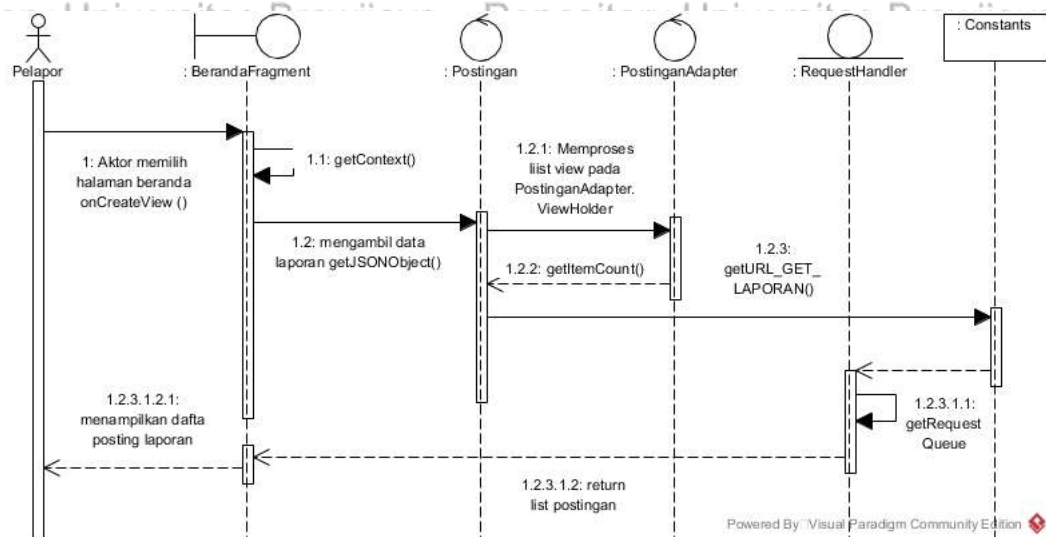


Gambar 5.1 Sequence diagram Membuat Laporan

#### 5.1.1.1.2 Melihat Daftar Laporan

Sequence diagram melihat daftar laporan menjelaskan bagaimana alur pelapor dalam melakukan interaksi pada sistem aplikasi GoTindak. Terdapat 5 objek pada sequence melihat daftar laporan yaitu beranda *fragment*, *adapter*, *request handler* dan *constants*. Pertama aktor memilih halaman beranda dengan memanggil *method oncreateView()* pada *boundary BerandaFragment*. Selanjutnya memanggil data laporan dengan *format getJsonObject()* dan mengembalikan objek dalam bentuk *list view* dengan memanggil *method Adapter ViewHolder* pada *control Adapter*. Kemudian mengembalikan pada *control* dengan *method getItemCount()* yang berfungsi menampilkan laporan dalam bentuk *listview*. Selanjutnya data laporan memanggil objek *url\_get\_laporan* pada *service constants* dengan mengembalikan bentuk data menjadi *JSON* dengan *method return list laporan*, jika berhasil maka data laporan akan ditampilkan dalam halaman beranda. *Sequence diagram* membuat laporan akan dijelaskan pada Gambar 5.2 dibawah ini.



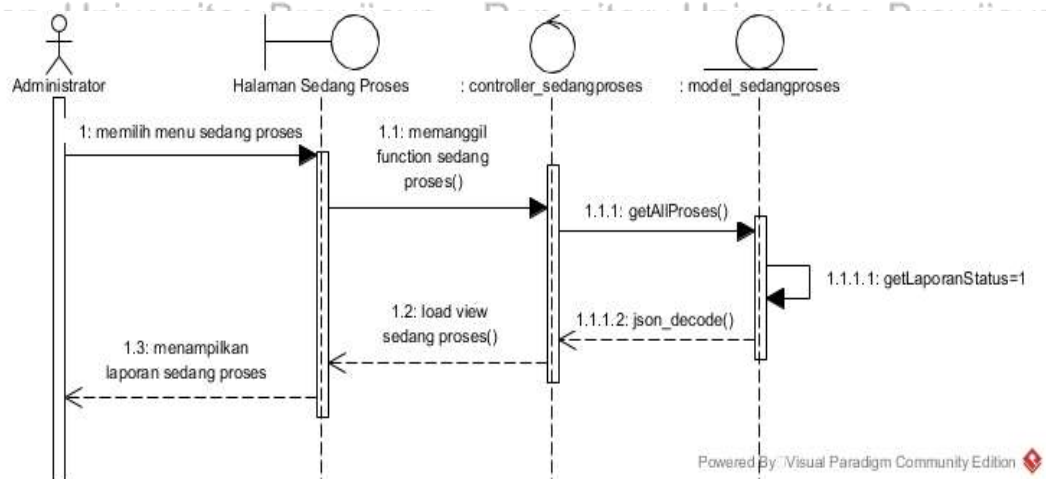


Gambar 5.2 Sequence Diagram Melihat Laporan

#### 5.1.1.1.3 Melihat Laporan Sedang Proses

*Sequence diagram* melihat laporan sedang proses menjelaskan bagaimana alur admin dalam melakukan interaksi pada sistem *website* GoTindak. Terdapat 3 *lifeline* pada *sequence* melihat daftar laporan yaitu *boundary* halaman sedang proses, *controller* sedang proses, dan entitas model sedang proses. Pertama admin memilih menu sedang proses, kemudian memanggil *function* sedang proses yang terdapat pada *controller* sedang proses. Selanjutnya *controller* sedang proses memanggil *method* *getAllProses* pada model sedang proses. Untuk dapat menampilkan laporan dengan status sedang proses maka model sedang proses mengembalikan *getLaporanStatus=1* yang artinya hanya laporan yang memiliki status sedang proses saja yang akan ditampilkan. Sesudah itu model sedang proses mengembalikan data dalam bentuk *JSON* *decode()* pada *controller* sedang proses. Sehingga data laporan akan ditampilkan pada halaman sedang proses dengan memanggil *fungsi* *load view* sedang proses. *Sequence diagram* melihat laporan sedang proses akan dijelaskan pada Gambar 5.3 dibawah ini.





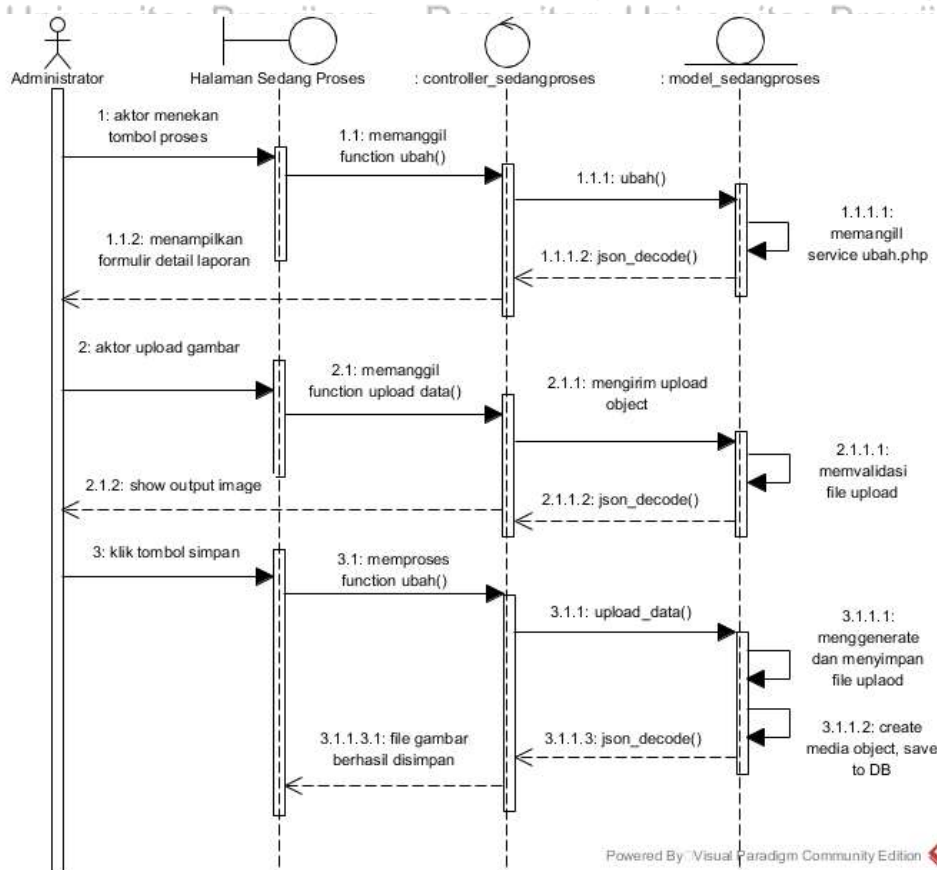
Gambar 5.3 Sequence Diagram Melihat Laporan Sedang Proses

### 5.1.1.2 Sequence diagram Iterasi Kedua

#### 5.1.1.2.1 Mengupload foto sedang proses

*Sequence diagram* mengupload foto sedang proses menjelaskan bagaimana alur admin dalam melakukan interaksi pada sistem *website* GoTindak. Terdapat 3 *lifeline* pada *sequence* melihat daftar laporan yaitu *boundary* halaman sedang proses, *controller* sedang proses, dan entitas model sedang proses. Pertama aktor menekan tombol proses kemudian memanggil *function* *ubah()* pada *controller* sedang proses. Selanjutnya *controller* sedang proses memanggil *method* *ubah()* pada model sedang proses. Untuk dapat menampilkan laporan dengan status sedang proses maka model sedang proses *return service* *ubah.php*. Sesudah itu model sedang proses mengembalikan data dalam bentuk *JSON\_decode()* pada *controller* sedang proses. Sehingga akan menampilkan *formulir* detail laporan. Setelah itu aktor melakukan upload gambar dengan memanggil *function* *upload data()* dengan mengirimkan *upload object* pada model sedang proses dan memvalidasi file upload gambar. Selanjutnya mengembalikan data dalam bentuk *JSON\_decode()* pada *controller* sedang proses sehingga akan menampilkan *output image*. Ketika aktor melakukan klik tombol simpan akan memproses *function* *ubah()* dan memproses *upload\_data()*. Setelah itu melakukan *generate* dan menyimpan file upload pada *database* dengan mengembalikan data berupa *JSON\_decode()*, sehingga file gambar berhasil disimpan. *Sequence diagram* mengupload foto sedang proses akan dijelaskan pada Gambar 5.4 di bawah ini.





Gambar 5.4 Sequence Diagram Mengupload Foto Sedang Proese

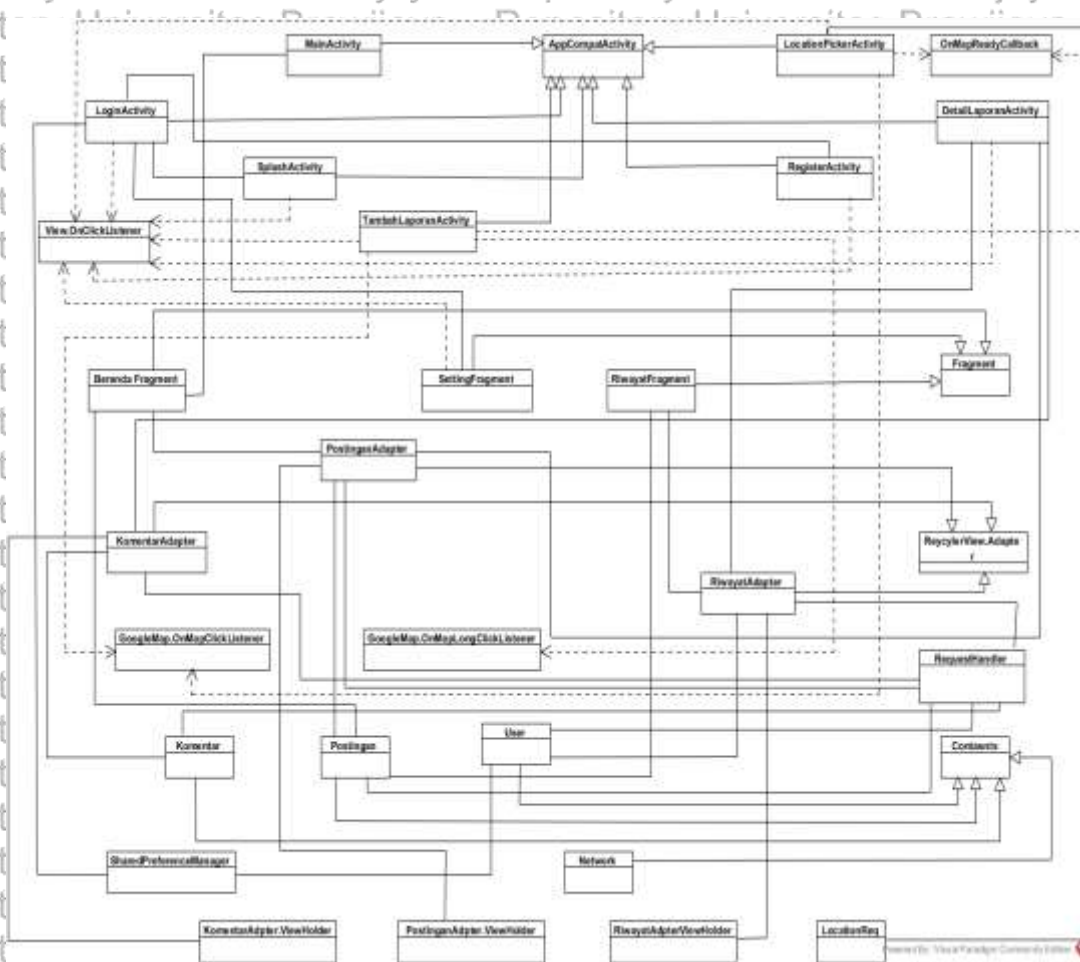
### 5.1.2 Perancangan Class Diagram

Perancangan class diagram merupakan perancangan untuk menggambarkan serta mendokumentasikan kode program dari sistem aplikasi yang akan dibangun. Oleh karena itu pada *class diagram* ini, akan digambarkan pada 2 sub sistem yaitu sistem aplikasi *mobile* untuk aktor pelapor dan sistem aplikasi *website* untuk administrator. Tujuan dari perancangan *class diagram* untuk memodelkan pandangan statis pada sistem aplikasi.

#### 5.1.2.1 Perancangan Class Diagram Pelapor Iterasi Pertama

Perancangan *class diagram* pelapor iterasi pertama, dirancang untuk sub sistem aplikasi *mobile* yang terdiri dari 5 *package* yaitu *activity*, *fragment*, *adapter*, *model*, dan *service*. Berkaitan dengan itu, pada masing-masing *package* terdiri dari kelas-kelas yang akan didefinisikan. Pada gambar 5.5 akan digambarkan perancangan *class diagram* umum pada aplikasi *mobile* GoTindak.





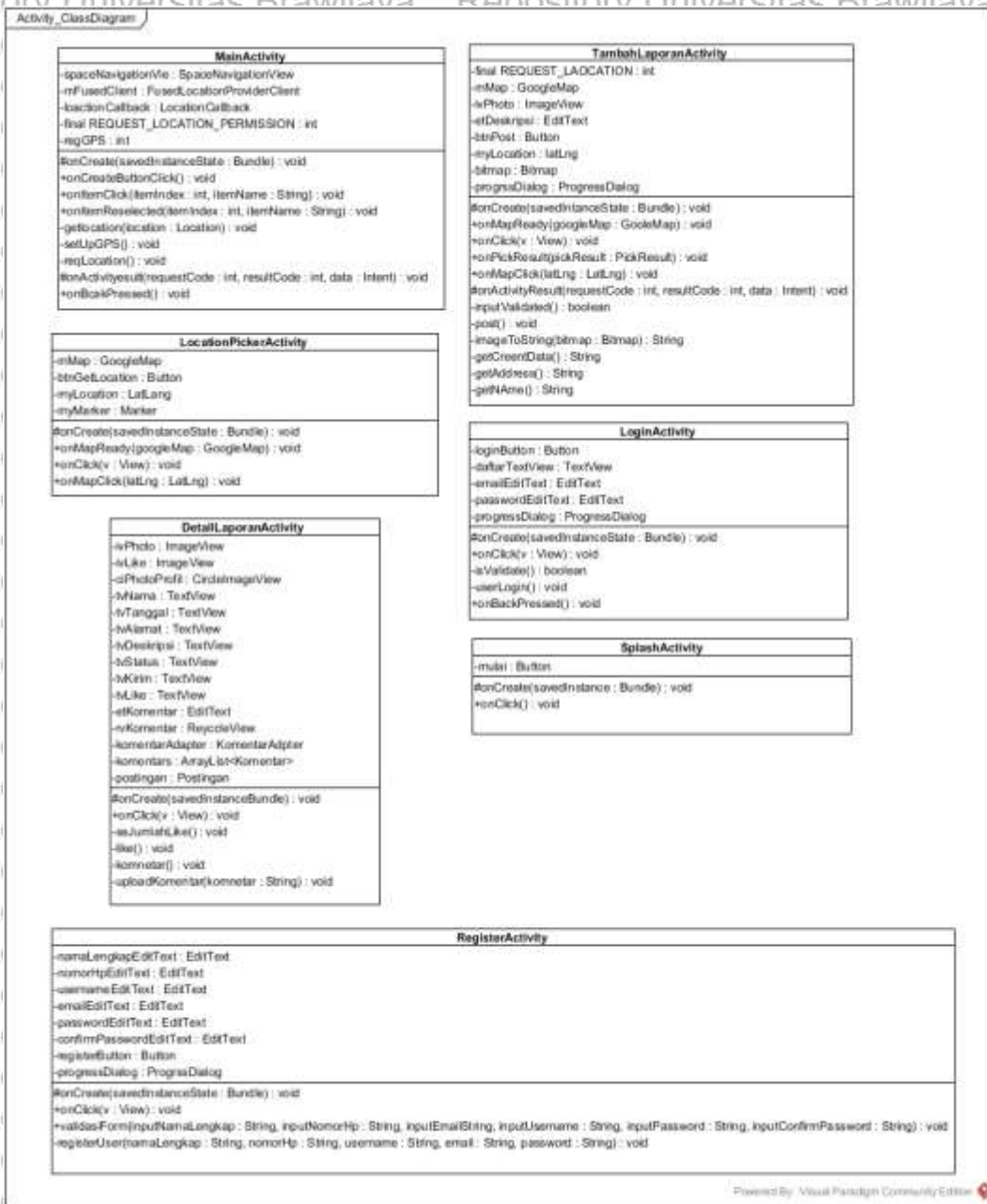
Gambar 5.5 Class Diagram Pelapor Iterasi Pertama





### 1. Detail Class Activity Aplikasi Mobile Iterasi Pertama

Package activity terdiri DetailInformasiActivity, DetailLaporanActivity, LocationPickerActivity, LoginActivity, MainActivity, RegisterActivity, SplashActivity, TambahLaporanActivity. Package activity ini berfungsi untuk menampilkan serta mengatur layout yang akan berinteraksi antara user dengan sistem. Detail package activity akan dijelaskan pada gambar 5.6 di bawah ini.



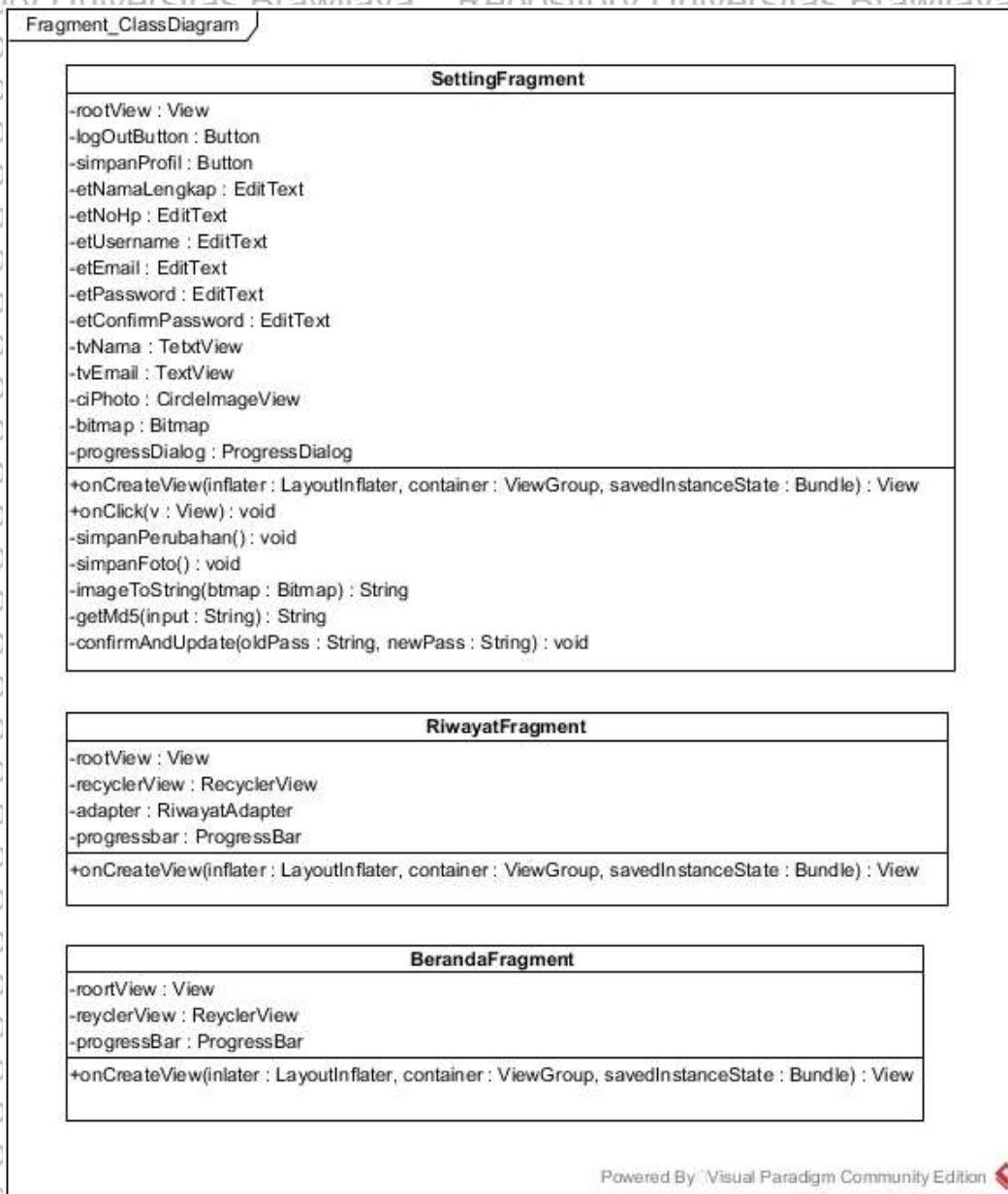
Gambar 5.6 Class Diagram Package Activity Iterasi Pertama





## 2. Detail Class Fragment Aplikasi Mobile Iterasi Pertama

Selanjutnya terdapat *class fragment* yang merupakan bagian *user interface* pada *activity*. *Layout fragment* terletak pada halaman *activity* sehingga terjadi penggabungan *layout* atau dapat dikatakan sebagai *sub-activity*. *Fragment* tidak dapat berjalan sendiri tanpa adanya *activity*. Detail *package fragment* akan dijelaskan pada gambar 5.7 di bawah ini.



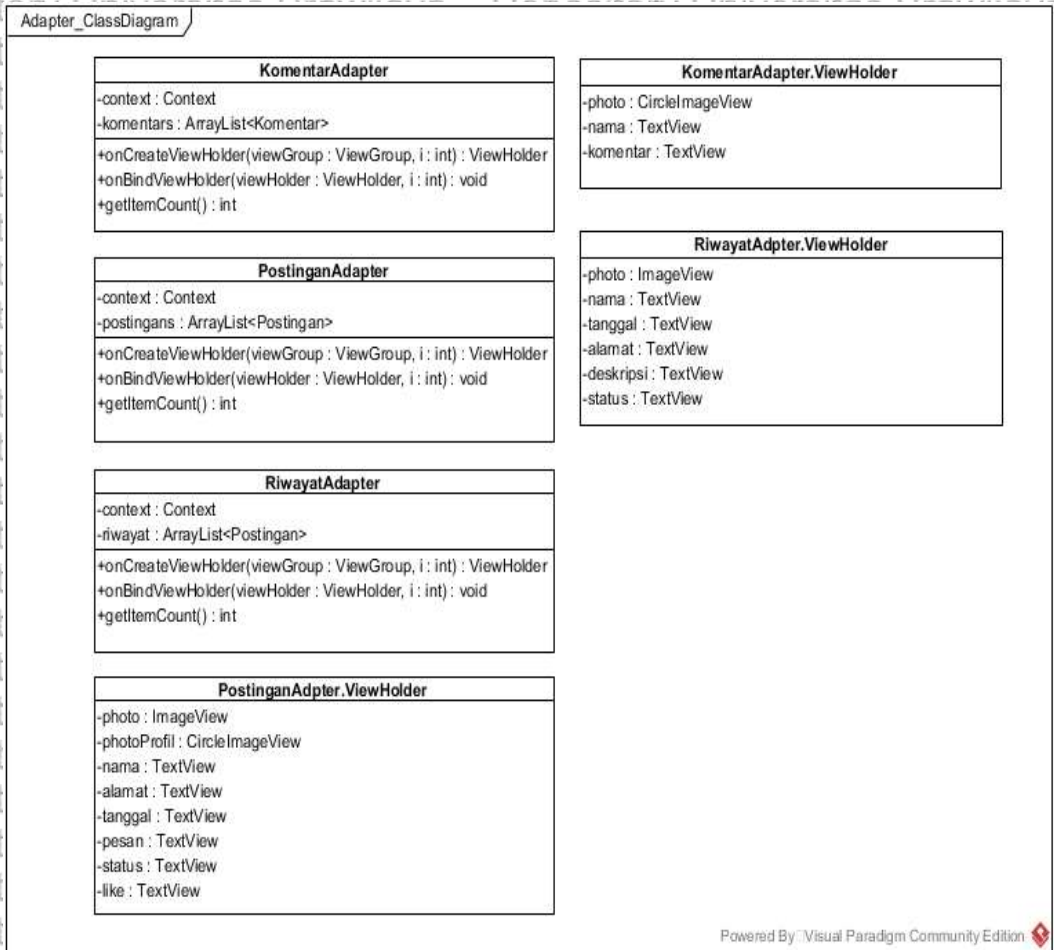
Gambar 5.7 Class Diagram Package Fragment Iterasi Pertama





### 3. Detail Class Adapter Aplikasi *Mobile* Iterasi Pertama

*Package adapter* mengatur *item* pada *listview* serta menjembatani antara *adapter view* atau *list view* dengan data. *Adapter* juga berperan penting dalam mengelola data model. *Package adapter* terdiri dari komentar *adapter*, *adapter*, *adapter viewholder*, riwayat *adapter*, riwayat *adapter viewholder*, dan komentar *adapter viewholder*. Detail *package adapter* akan dijelaskan pada gambar 5.8 di bawah ini.

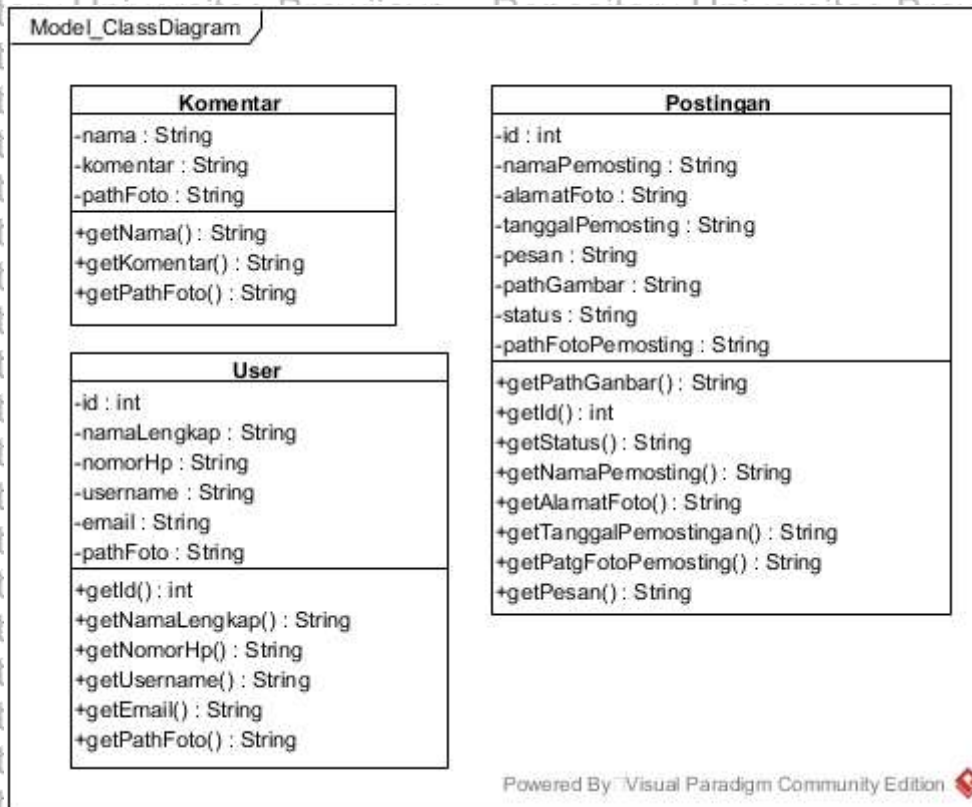


Gambar 5.8 Class Diagram Package Adapter Iterasi Pertama

### 4. Detail Class Model Aplikasi *Mobile* Iterasi Pertama

Detail *class* pada model berfungsi untuk menyimpan informasi untuk ditampilkan kepada pengguna dengan membaca data dari hasil *request server*. *Package model* terdiri dari komentar, *user*. Detail *package model* akan dijelaskan pada gambar 5.9 di bawah ini.



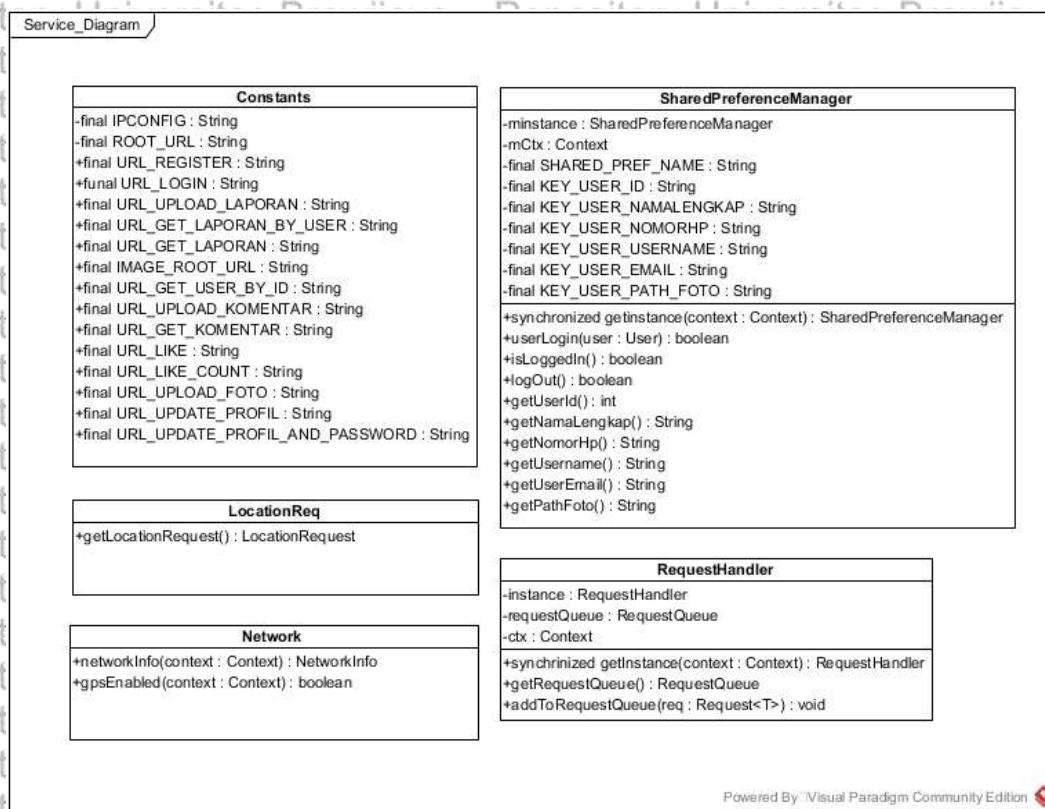


Gambar 5.9 Class Diagram Package Model Iterasi Pertama

## 5. Detail Class Service Aplikasi Mobile Iterasi Pertama

Detail kelas diagram pada *package service* terdiri dari *constants*, *RequestHandler*, *LocationReq*, *Network*, *SharePreferenceManager*. *Class constant* berfungsi sebagai *service* yang menampung variabel-variabel seperti nama *Base\_URL*, *request handler* berfungsi untuk mengirimkan data, *network* digunakan sebagai *internet permission*, *locationreq* untuk mendapatkan *request lokasi*, dan *SharePreferenceManager* digunakan untuk menambahkan *session login* dan *logout*. Detail *package service* akan dijelaskan pada gambar 5.10 di bawah ini.



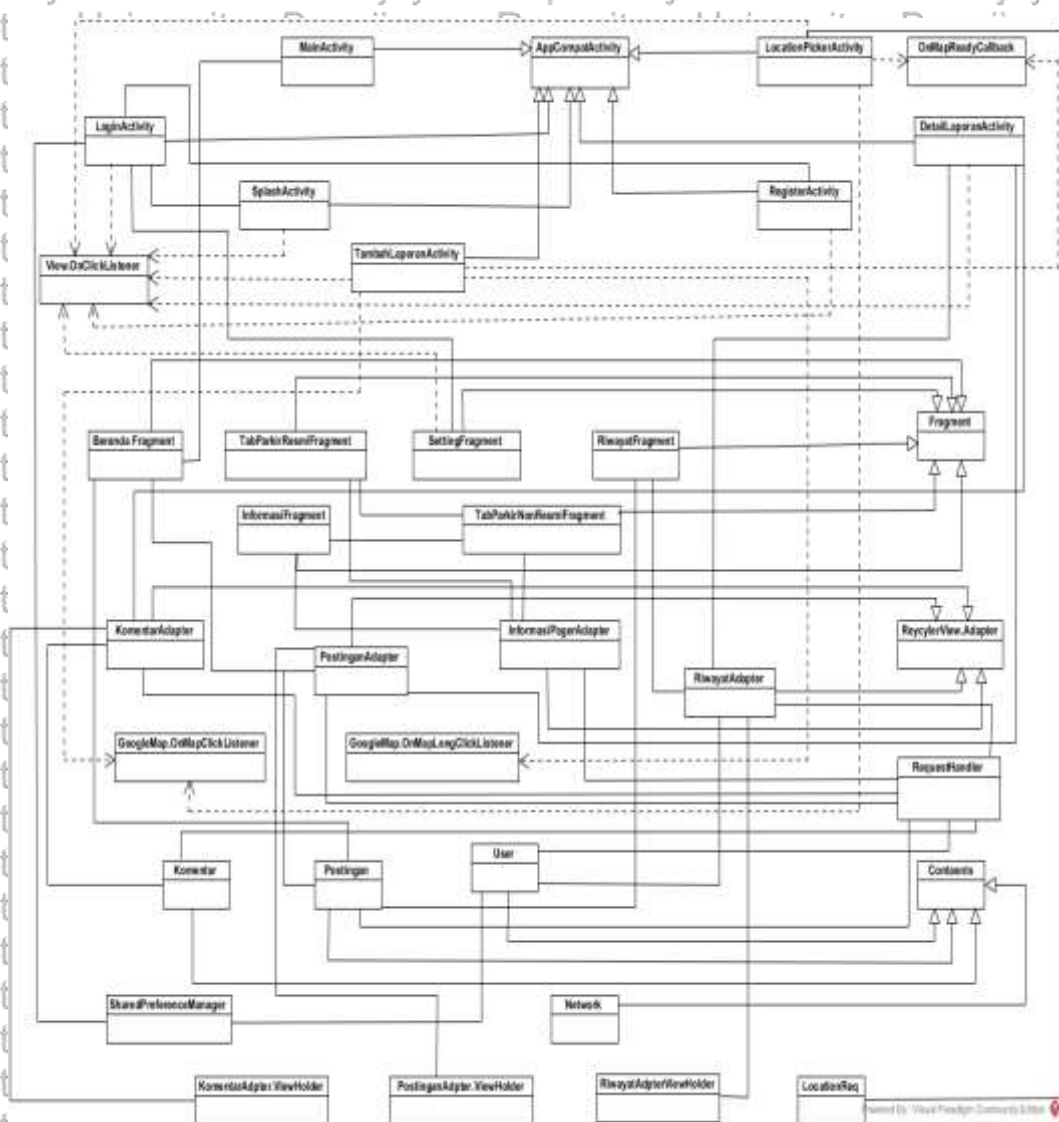


Gambar 5.10 Class Diagram Package Service Iterasi Pertama

### 5.1.2.2 Perancangan Kelas Diagram Pelapor Iterasi Kedua

Perancangan class diagram pelapor iterasi kedua, dirancang untuk sub sistem aplikasi *mobile* yang terdiri dari 5 *package* yaitu *activity*, *fragment*, *adapter*, *model*, dan *service*. Berkaitan dengan itu, pada masing-masing *package* terdiri dari kelas-kelas yang akan didefinisikan. Dalam konteks ini, terdapat proses iterasi dengan melakukan penambahan *InformasiPagerAdapter*, *InformasiFragment*, *TabParkirResmiFragment*, *TabParkirNonResmiFragment*. Pada gambar 5.11 akan digambarkan perancangan *class diagram* umum iterasi kedua pada aplikasi *mobile GoTindak*.



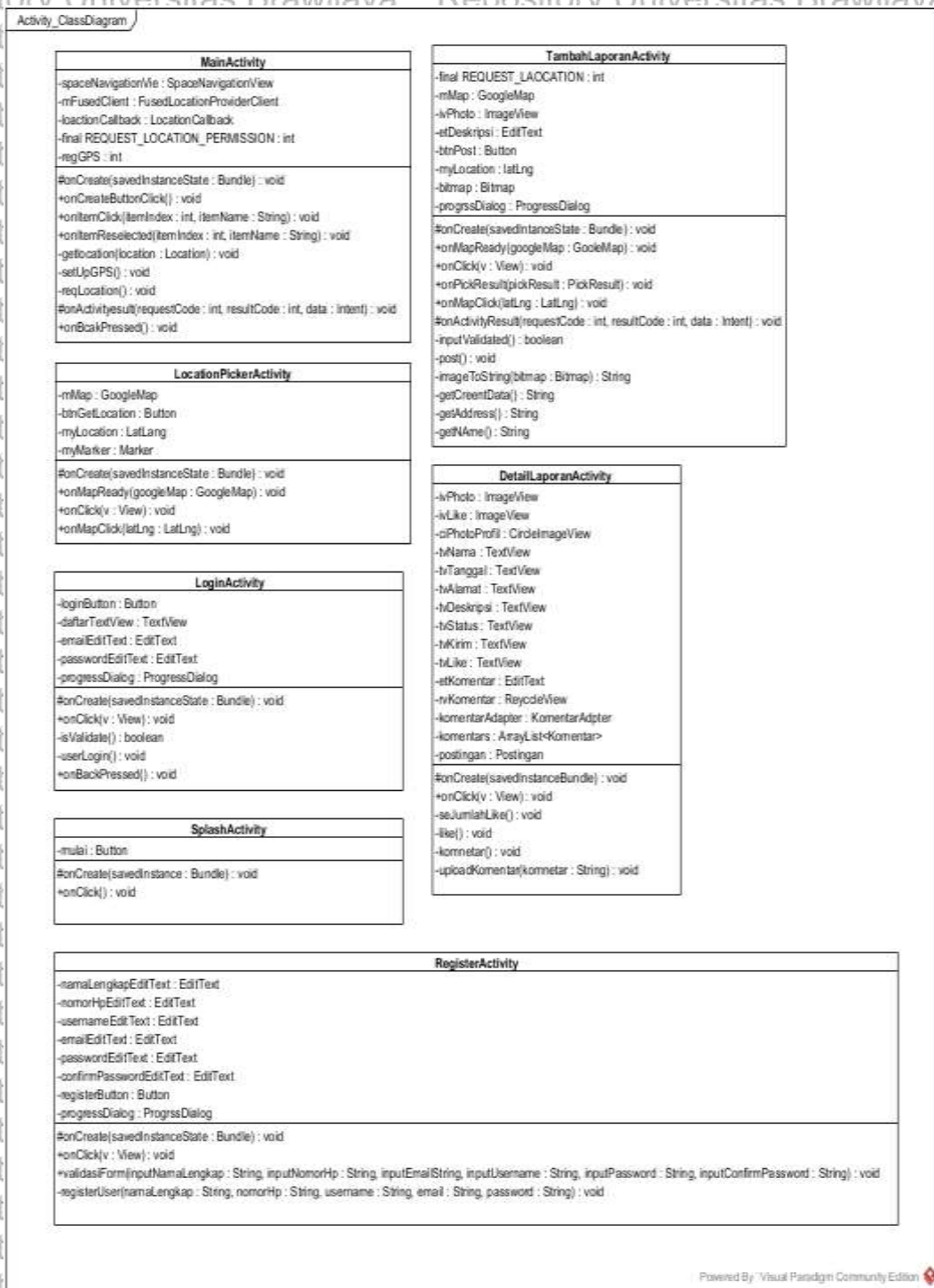


Gambar 5.11 Class Diagram Pelapor Iterasi Kedua

### 1. Detail Class Activity Aplikasi *Mobile* Iterasi Kedua

*Package activity* terdiri *DetailInformasiActivity*, *DetailLaporanActivity*, *LocationPickerActivity*, *LoginActivity*, *MainActivity*, *RegisterActivity*, *SplashActivity*, *TambahLaporanActivity*. *Package activity* ini berfungsi untuk menampilkan serta mengatur layout yang akan berinteraksi antara *user* dengan sistem. Detail *package activity* akan dijelaskan pada gambar 5.12 di bawah ini.





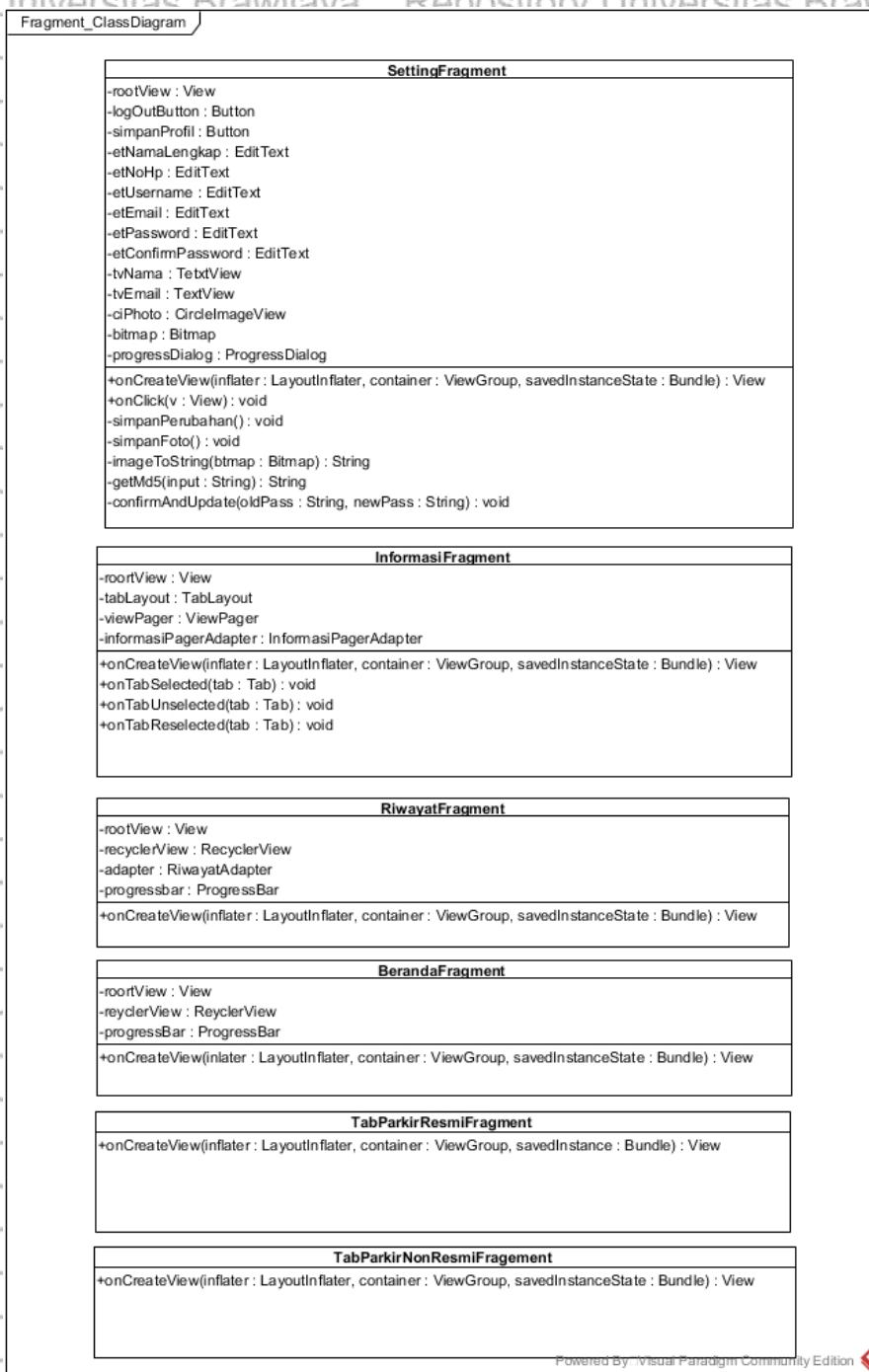
Gambar 5.12 Class Diagram Package Activity Iterasi Kedua





## 2. Detail Class Fragment Aplikasi *Mobile* Iterasi Kedua

Selanjutnya terdapat *class fragment* yang merupakan bagian *user interface* pada *activity*. *Layout fragment* terletak pada halaman *activity* sehingga terjadi penggabungan *layout* atau dapat dikatakan sebagai *sub-activity*. *Fragment* tidak dapat berjalan sendiri tanpa adanya *activity*. *Class fragment* pada iterasi kedua terdiri dari *SettingFragment*, *InformasiFragment*, *TabParkirNonResmiFragment*, *TabParkirResmiFragment*, *BerandaFragment*, *RiwayatFragment*, *InformasiFragment*. Detail *package fragment* akan dijelaskan pada gambar 5.13 di bawah ini.



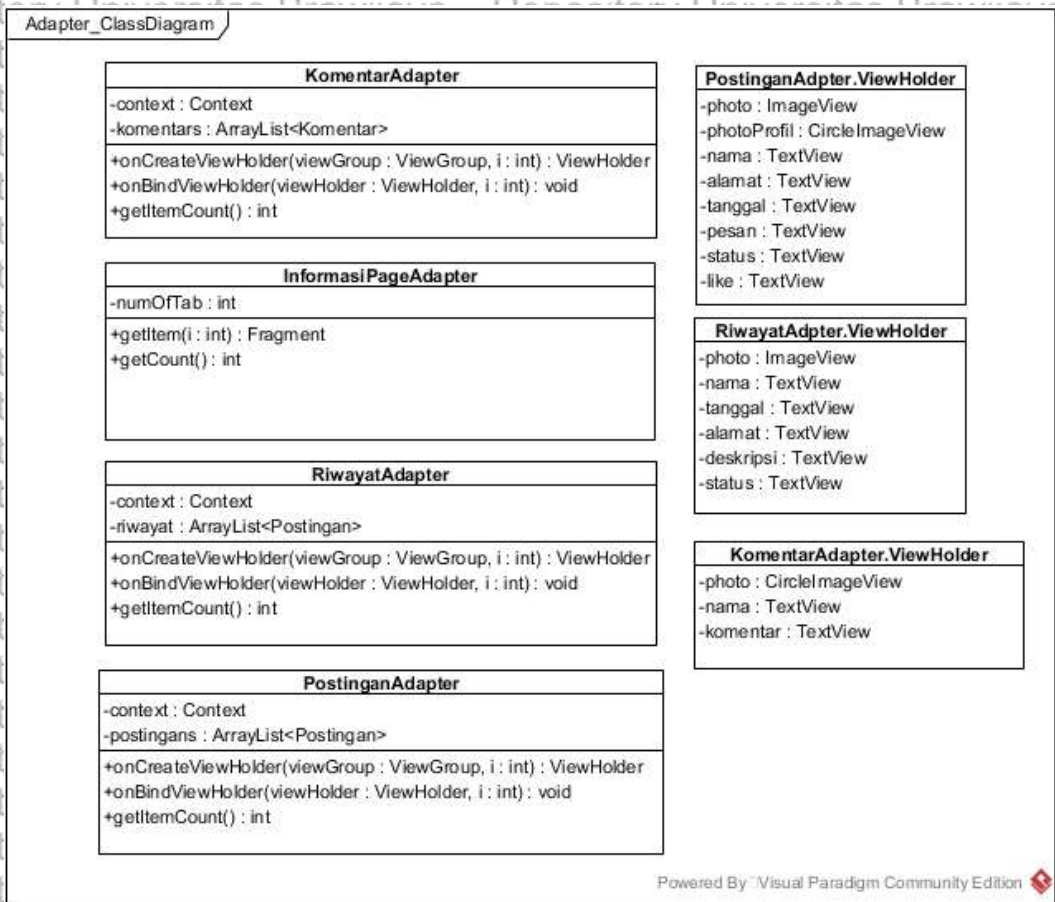
Gambar 5.13 Class Diagram Package Fragment Iterasi Kedua





### 3. Detail Class Adapter Aplikasi *Mobile* Iterasi Kedua

*Package adapter* mengatur item pada *listview* serta menjembatani antara *adapter view* atau *list view* dengan data. *Adapter* juga berperan penting dalam mengelola data model. *Package adapter* terdiri dari *komentar adapter*, *adapter*, *adapter viewholder*, *riwayat adapter*, *riwayat adapter viewholder*, dan *komentar adapter viewholder*, dan *informasi pager adapter*. Detail *package adapter* akan dijelaskan pada gambar 5.14 di bawah ini.



Gambar 5.14 Class Diagram Package Adapter Iterasi Kedua

### 4. Detail Class Model Aplikasi *Mobile* Iterasi Kedua

Detail class pada model berfungsi untuk menyimpan informasi untuk ditampilkan kepada pengguna dengan membaca data dari hasil *request server*. *Package model* terdiri dari *komentar*, *user*. Detail *package model* akan dijelaskan pada gambar 5.15 di bawah ini.



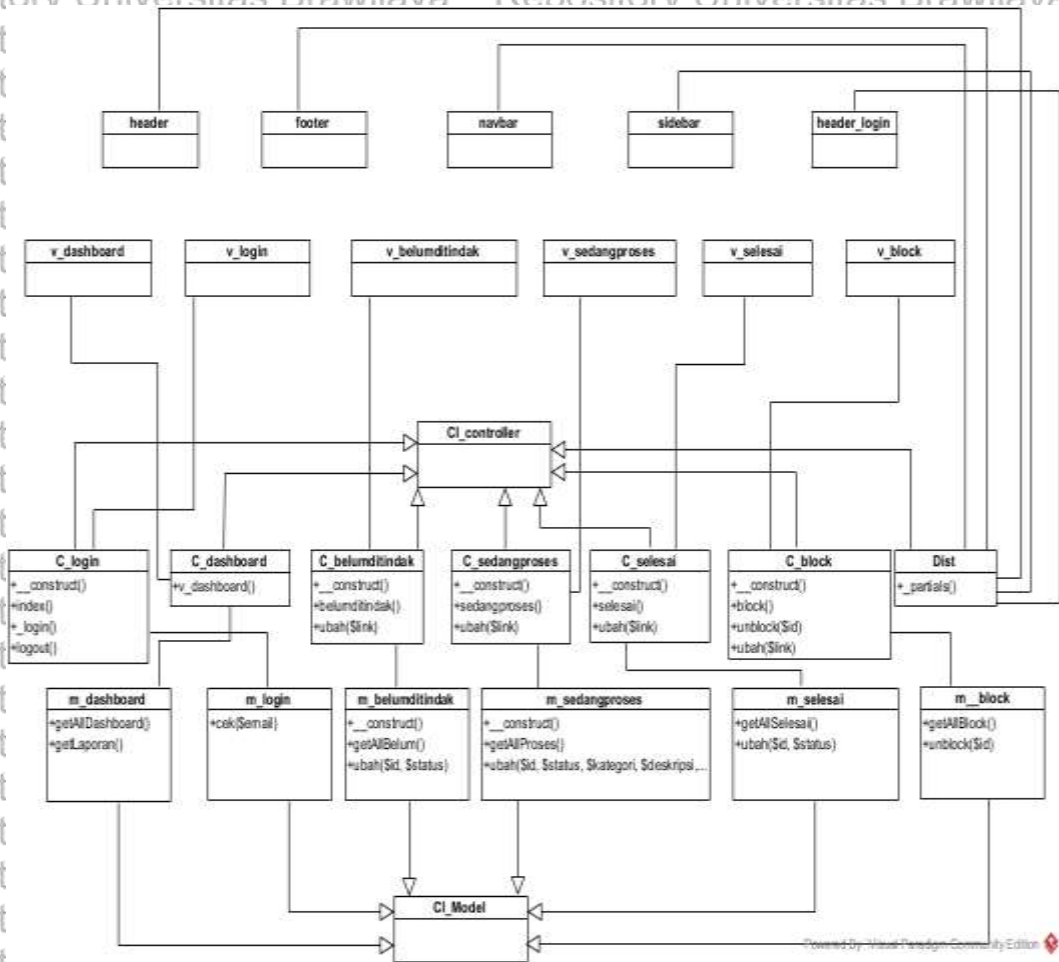






### 5.1.2.3 Perancangan Class Diagram Administrator Iterasi Pertama

Perancangan *class diagram* administrator iterasi pertama, dirancang untuk sub sistem *website* menggunakan model *MVC* atau *model view controller* yang terdiri dari 3 kelas yaitu *model*, *view*, dan *controller*. Berkaitan dengan itu, pada masing-masing kelas terdiri dari kelas-kelas yang akan didefinisikan. Kelas model terdiri dari *m\_dashboard*, *m\_login*, *m\_belumditindak*, *m\_sedangproses*, *m\_selesai*, *m\_tidak disetujui*. Selanjutnya pada *controller* terdapat *c\_login*, *c\_dashboard*, *c\_belumditindak*, *c\_sedangproses*, *c\_selesai*, *c\_tidak disetujui* dan *dist*. Pada kelas *view* terdiri dari *v\_dashboard*, *v\_login*, *v\_belumditindak*, *v\_sedangproses*, *v\_selesai*, *v\_tidak disetujui* dan *partials*, *header*, *footer*, *navbar*, *sidebar*, dan *header\_login*. Pada gambar 5.17 akan digambarkan perancangan *class diagram* umum pada sistem *website* GoTindak.



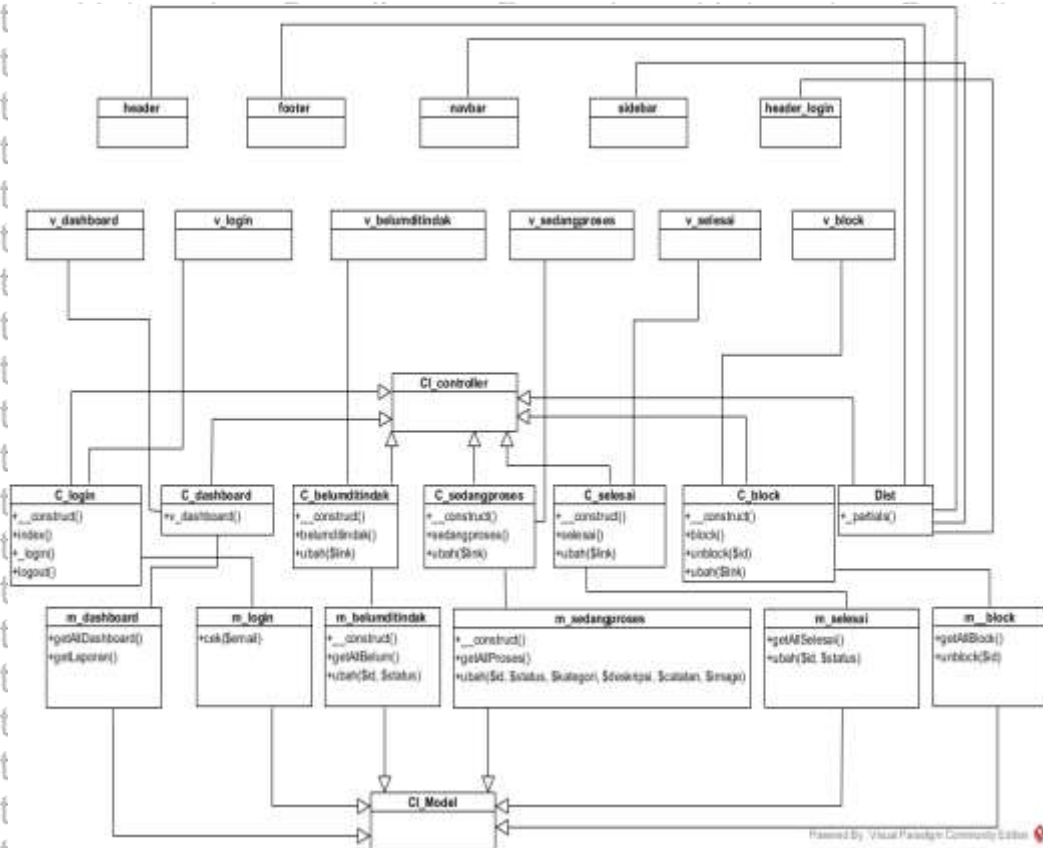
Gambar 5.17 Class Diagram Administrator Iterasi Pertama

### 5.1.2.4 Perancangan Class Diagram Administator Iterasi Kedua

Perancangan *class diagram* administrator iterasi kedua dengan tambahan atribut *Scatatan* dan *Simage* pada model *m\_sedangproses*, dirancang untuk sub sistem *website* menggunakan model *MVC* atau *model view controller* yang terdiri dari 3 kelas yaitu *model*, *view*, dan *controller*. Berkaitan dengan itu, pada masing-masing kelas terdiri dari kelas-kelas yang akan didefinisikan. Kelas model terdiri



dari *m\_dashboard*, *m\_login*, *m\_belumditindak*, *m\_sedangproses*, *m\_selesai*, *m\_tidak\_disetujui*. Selanjutnya pada *controller* terdapat *c\_login*, *c\_dashboard*, *c\_belumditindak*, *c\_sedangproses*, *c\_selesai*, *c\_tidak\_disetujui* dan *dist*. Pada kelas *view* terdiri dari *v\_dashboard*, *v\_login*, *v\_belumditindak*, *v\_sedangproses*, *v\_selesai*, *v\_tidak\_disetujui* dan *\_partials* *header*, *footer*, *navbar*, *sidebar*, dan *header\_login*. Pada gambar 5.18 akan digambarkan perancangan *class diagram* umum pada sistem *website* GoTindak.



Gambar 5.18 Class Diagram Administrator Iterasi Kedua

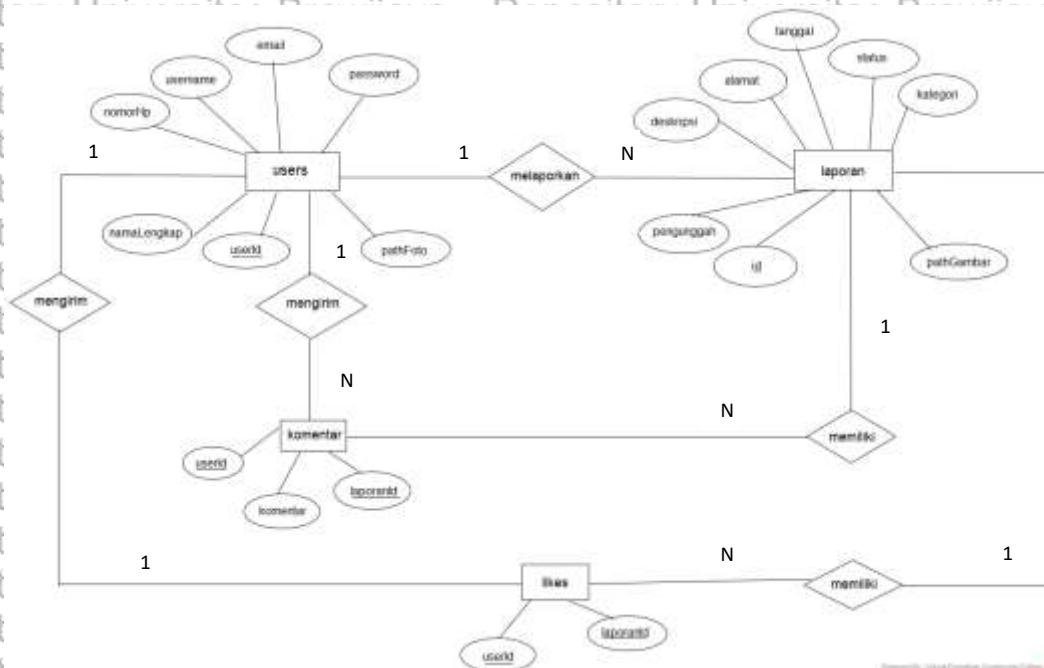
### 5.1.3 Perancangan Basis Data

Perancangan basis data yang akan dibahas pada aplikasi GoTindak menggunakan *database* MySQL yang akan digambarkan dalam bentuk ERD atau *Entity Relationship Diagram*. Pada aplikasi GoTindak terdapat 5 entitas yang terdiri dari *likes*, *laporan*, *komentar*, *users*, dan *admin*. Pada perancangan basis data terdapat proses iterasi dengan penambahan atribut pada entitas *laporan* dengan tujuan memenuhi kebutuhan pengguna.

#### 5.1.3.1 Perancangan Basis Data Iterasi Pertama

Perancangan basis data iterasi pertama tidak terdapat penambahan ataupun pengurangan pada kebutuhan fungsional pengguna. Gambar 5.19 akan menjelaskan perancangan basis data pada aplikasi GoTindak.





**Gambar 5.19 Perancangan Entity Relationship Diagram Iterasi Pertama**

Gambar 5.19 menjelaskan perancangan basis data yang direpresentasikan dalam bentuk perancangan *entity relationship diagram*, yang terdapat 4 entitas yaitu entitas *users*, entitas *komentar*, entitas *laporan*, entitas *likes*.

Entitas *Users* mempresentasikan rancangan tabel dari pengguna aplikasi GoTindak yang berfungsi untuk menyimpan data pengguna sebagai pelapor. Entitas *user* memiliki tujuh kolom dengan tipe data yang beragam. Entitas *users* dapat terisi jika pengguna melakukan registrasi pada aplikasi GoTindak. Tabel 5.1 akan menjelaskan lebih detail rancangan tabel entitas *users*.

**Tabel 5.1 Rancangan Tabel Entitas *Users***

Nama Atribut	Tipe Data	Keterangan
<i>userId</i>	int	Id yang dimiliki pengguna sebagai kunci utama
<i>namaLengkap</i>	varchar	Nama lengkap dari pengguna
<i>nomorHP</i>	varchar	nomorHP yang dimiliki pengguna
<i>username</i>	varchar	Username dari pengguna
<i>email</i>	varchar	Email yang dimiliki pengguna
<i>password</i>	varchar	Password dari pengguna untuk akses masuk
<i>pathFoto</i>	varchar	PathFoto pengguna





Entitas Laporan mempresentasikan rancangan tabel laporan dari aplikasi GoTindak yang berfungsi untuk menyimpan data laporan dari pengguna. Entitas laporan memiliki delapan kolom dengan tipe data yang beragam. Entitas laporan dapat terisi jika pengguna melakukan laporan kejadian parkir non resmi pada aplikasi GoTindak. Tabel 5.2 akan menjelaskan lebih detail rancangan tabel entitas laporan.

**Tabel 5.2 Rancangan Tabel Entitas Laporan**

Nama Atribut	Tipe Data	Keterangan
Id	Int	Id pada laporan
Pengunggah	Int	Pengunggah laporan
Deskripsi	Varchar	Deskripsi laporan yang ditulis oleh pengguna
Alamat	Varchar	Alamat kejadian pada laporan
Tanggal	Varchar	Tanggal laporan yang diunggah
Status	Varchar	Status pada laporan
Kategori	Varchar	Kategori pada laporan
pathGambar	Varchar	pathGambar yang diunggah oleh pengguna

Entitas Komentar mempresentasikan rancangan tabel dari komentar pada aplikasi GoTindak yang berfungsi untuk menyimpan data komentar yang dikirimkan oleh pengguna sebagai pelapor. Entitas komentar memiliki tiga kolom dengan tipe data yang beragam. Entitas komentar dapat terisi jika pengguna mengirim pada aplikasi GoTindak. Tabel 5.3 akan menjelaskan lebih detail rancangan tabel entitas users.

**Tabel 5.3 Rancangan Tabel Entitas Komentar**

Nama Atribut	Tipe Data	Keterangan
userId	Int	Id pengguna
LaporanId	Int	LaporanId sbegaaai id pada laporan
Komentar	varchar	Komentar yang akan diisi oleh pengguna

Entitas Like mempresentasikan rancangan tabel dari like pada aplikasi GoTindak yang berfungsi untuk memberikan dukungan pada laporan pengguna. Entitas like dapat terisi jika pengguna mengirim dukungan pada laporan. Tabel 5.4 akan menjelaskan lebih detail rancangan tabel entitas like.



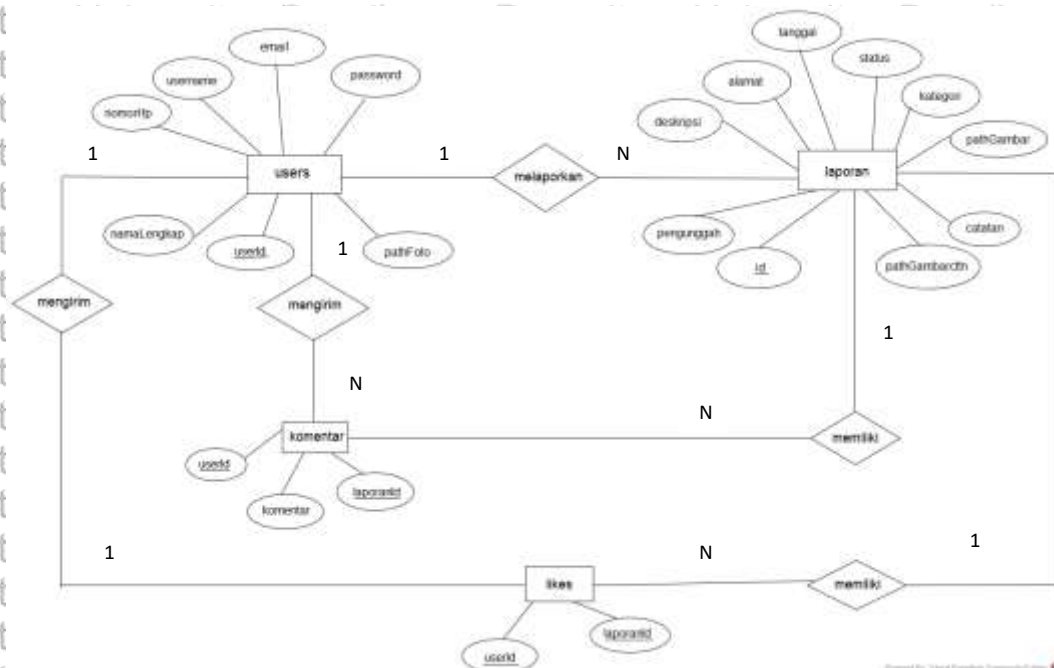


Tabel 5.4 Rancangan Tabel Entitas Like

Nama Atribut	Tipe Data	Keterangan
userId	Int	Id pengguna
LaporanId	Int	LaporanId sebagai id pada laporan

5.1.3.2 Perancangan Basis Data Iterasi Kedua

Perancangan basis data iterasi kedua terdapat penambahan kolom pada entitas laporan yaitu kolom catatan dan kolom *pathGamabrcttn* untuk memenuhi kebutuhan fungsional pengguna. Gambar 5.20 akan menjelaskan perancangan basis data iterasi kedua pada aplikasi GoTindak.



Gambar 5.20 Perancangan Entity Relationship Diagram Iterasi Kedua

Entitas Laporan mempresentasikan rancangan tabel laporan dari aplikasi GoTindak yang berfungsi untuk menyimpan data laporan dari pengguna. Entitas laporan memiliki 10 kolom dengan tipe data yang beragam. Entitas laporan dapat terisi jika pengguna melakukan laporan kejadian parkir non resmi pada aplikasi GoTindak. Tabel 5.5 akan menjelaskan lebih detail rancangan tabel entitas laporan.

Tabel 5.5 Rancangan Tabel Entitas Laporan

Nama Atribut	Tipe Data	Keterangan
Id	Int	Id pada laporan
Pengunggah	Int	Pengunggah laporan





Tabel 5.5 Rancangan Tabel Entitas Laporan (Lanjutan)

Nama Atribut	Tipe Data	Keterangan
Deskripsi	Varchar	Deskripsi laporan yang ditulis oleh pengguna
Alamat	Varchar	Alamat kejadian pada laporan
Tanggal	Varchar	Tanggal laporan yang diunggah
Status	Varchar	Status pada laporan
Kategori	Varchar	Kategori pada laporan
pathGambar	Varchar	pathGambar yang diunggah oleh pengguna
catatan	Varchar	Catatan berupa tanggapan pada laporan
pathGambarcttn	Varchar	Gambar berupa tindak lanjut pada laporan pengguna

#### 5.1.4 Perancangan Komponen

Perancangan komponen mendefinisikan perancangan unit terkecil yang dirancang untuk membuat algoritme pada suatu *method*, dengan tujuan untuk menjelaskan proses pada fungsi yang terdapat dalam sistem. Penulisan pada perancangan komponen menggunakan *pseudocode*. Terdapat 4 perancangan komponen yang akan dijelaskan yaitu 2 komponen pada sub sistem aplikasi *mobile* dan 2 sub sistem pada *website*.

##### 5.1.4.1 Algoritme Membuat Laporan Parkir

Algoritme ini menjelaskan tugas ketika pelapor membuat laporan pada sistem aplikasi *mobile*. Penjelasan mengenai algoritme membuat laporan parkir akan dijelaskan pada tabel 5.6 dibawah ini.

Nama kelas : `TambahLaporanActivity`

Nama metode : `post ()`

Tabel 5.6 Algoritme Membuat Laporan Parkir

No	Kode Program
1	Start
1	Inisialisasi
2	if (memproses validasi inputan)
3	Meginisialisasi <code>stringRequest (Mengirim.Method.POST,</code>
3	<code>dengan service pada Constants.URL_UPLOAD LAPORAN) {</code>
4	<code>Deklarasi onResponse {</code>
5	<code>if (server merespon) {</code>
6	<code>menampilkan toast dengan</code>
6	<code>text, "Uploadsucceeded".show();</code>
7	<code>}</code>
8	<code>else {</code>
8	<code>menampilkan toast dengan text.show();</code>
8	<code>}</code>
9	<code>progressDialog.dismiss();}}</code>





10	Deklarasi <code>onErrorResponse {progressDialog.dismiss();}</code>
11	Menambahkan parameter <code>getParams()</code>
11	Mengirim params nama ke server
11	Mengirim params deskripsi ke server
11	Mengirim params pengunggah ke server
11	Mengirim params tanggal ke server
11	Mengirim params status ke server
11	Mengirim params kategori ke server
11	Mengirim params alamat ke server
11	Mengirim params image ke server
12	<code>return params;</code>
13	Menambahkan the request ke <code>RequestQueue</code> Selesai

#### 5.1.4.2 Algoritme Melihat Daftar Laporan

Algoritme ini menjelaskan tugas ketika pelapor melihat daftar laporan pada sistem aplikasi *mobile*. Penjelasan mengenai algoritme melihat daftar laporan parkir akan dijelaskan pada tabel 5.7 dibawah ini.

Nama kelas : `BerandaFragment`

Nama metode : `onCreateView()`

**Tabel 5.7 Algoritme Melihat Daftar Laporan**

No	Kode Program
1	Start
2	Inisialisasi <code>onCreateView</code>
3	<code>inflater.inflate(R.layout.fragment_beranda</code>
3	<code>Menambahkan recyclerView dengan findViewById</code>
3	<code>berandaRecyclerView</code>
3	<code>Menambahkan recyclerView dengan findViewById</code>
3	<code>progressBarBeranda);</code>
3	Mengatur layout manager menjadi <code>LinearLayoutManager</code>
4	Melakukan <code>Request.Method.GET, url,</code>
5	Mengirim <code>Response.Listener&lt;String&gt;</code>
5	Inisialisai <code>onResponse(String response)</code>
6	Memproses <code>ArrayList</code> Laporan
7	Mengirim <code>Response.ErrorListener()</code>
7	Inisialisai <code>onErrorResponse(VolleyError error)</code>
8	Menambahkan permintaan pada request to queue
9	End

#### 5.1.4.3 Algoritme Melihat Laporan Sedang Proses

Algoritme ini menjelaskan tugas ketika admin melihat laporan sedang proses pada sistem *website*. Penjelasan mengenai algoritme melihat laporan sedang proses akan dijelaskan pada tabel 5.8 dibawah ini.

Nama kelas : `m_sedangproses.php`

Nama metode : `getAllProses()`





**Tabel 5.8 Algoritme Melihat Laporan Sedang Proses**

No	Uraian
1	Start
2	Memanggil client = client Maka response = client memanggil request dengan method GET dan URL ke service getLaporanstatus.php dengan status=1 Mengambil result = response <i>JSON_decode</i> jadi array Mengembalikan fungsi result
3	End

#### 5.1.4.4 Algoritme Mengupload Foto Sedang Proses

Algoritme ini menjelaskan tugas ketika admin mengunggah foto sedang proses pada laporan sedang proses dalam sistem *website*. Penjelasan mengenai algoritme mengunggah laporan sedang proses akan dijelaskan pada tabel 5.9 dibawah ini.

Nama kelas : `m_sedangproses.php`

Nama metode : `ubah()`

**Tabel 5.9 Algoritme Mengupload Foto Sedang Proses**

No	Uraian
1	Start
2	Memanggil client = client Menampilkan variabel image
3	IF (image = null) (
4	Maka response = client memanggil request dengan method GET dan URL ke service ubah.php tanpa image)
5	Else (
6	Maka response = client memanggil request dengan method GET dan URL ke service ubah.php dengan image)
7	END IF
7	Menampilkan url service ubah.php Mengambil result = response <i>JSON_decode</i> jadi array Mengembalikan fungsi result
8	End

#### 5.1.5 Perancangan Antarmuka

Perancangan antarmuka pada aplikasi GoTindak bertujuan untuk memodelkan rancangan pada sistem dalam bentuk *wireframe*. Perancangan antarmuka dibuat untuk menghasilkan rancangan sistem yang dapat dipahami oleh pengguna. Pada sub bab ini akan berfokus pada perancangan sisi pengguna, sisi pelapor dan sisi administrator. Dalam hal ini, alur sistem yang bekerja akan dijelaskan menggunakan *screenflow* sehingga mudah untuk dipahami.



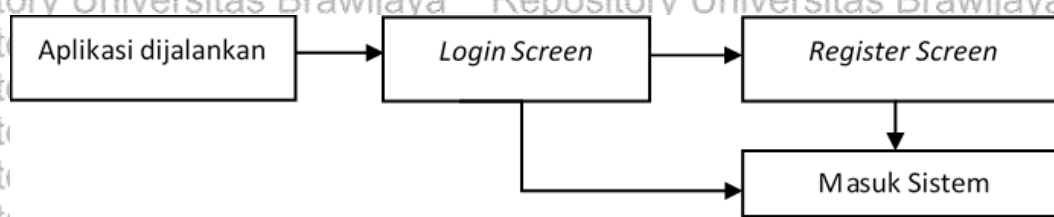


### 5.1.5.1 Perancangan Antarmuka Iterasi Pertama

Tahapan dalam perancangan antarmuka dengan mengidentifikasi kebutuhan pengguna yang telah didefinisikan sebelumnya pada proses analisis kebutuhan. Selanjutnya mengidentifikasi objek pada kebutuhan untuk diterapkan pada antarmuka pengguna, sehingga dapat memberikan suatu aksi pada objek. Sesudah itu, merancang tata letak pada layar antarmuka pengguna, sehingga akan diperoleh tahapan perancangan antarmuka iterasi pertama. Antarmuka yang akan dirancang berfokus pada pengguna, pelapor dan administrator.

#### 5.1.5.1.1 Perancangan Antarmuka Aktor Pengguna

Perancangan antarmuka pada aktor pengguna dirancang untuk menampilkan sistem dari sisi aktor pengguna sehingga dapat masuk dalam sistem. Gambar 5.21 menjelaskan rancangan *screenflow* aplikasi dari sisi aktor pengguna.



Gambar 5.21 *Screenflow* aplikasi sisi pengguna

Alur sistem pada antarmuka sisi pengguna ketika dijalankan terdiri dari halaman *login* dan halaman register. Tahapan awal pada saat menjalankan sistem akan membuka halaman *login*, ketika *user* belum mempunyai akun maka akan dialihkan pada halaman register dengan mengklik tombol daftar. Sehingga pengguna dapat masuk dalam sistem.

#### 1. Halaman *Login*

Halaman *login* merupakan halaman awal pada aplikasi sehingga pengguna dapat masuk dalam sistem. Gambar 5.22 menjelaskan rancangan *wireframe* pada halaman *login*.





**Gambar 5.22** Halaman Login

Penjelasan elemen pada *wireframe* halaman *login* akan dideskripsikan dalam tabel 5.10 berikut.

**Tabel 5.10** Penjelasan Antarmuka Halaman *Login*

No	Nama Objek	Tipe	Deskripsi
1	Tulisan <i>Login</i>	<i>TextView</i>	Menampilkan tulisan <i>login</i>
2	<i>Field Email</i>	<i>EditText</i>	Untuk memasukkan <i>email</i>
3	<i>Field Password</i>	<i>EditText</i>	Untuk menginputkan <i>password</i>
4	Tombol <i>Login</i>	<i>Button</i>	Tombol <i>login</i> untuk masuk dalam sistem
5	<i>Link Registrasi</i>	<i>TextView</i>	Untuk berpindah halaman register

## 2. Halaman Register

Halaman register merupakan halaman untuk mendaftarkan akun pengguna pada aplikasi sehingga pengguna dapat masuk dalam sistem. Gambar 5.23 menjelaskan rancangan *wireframe* pada halaman register.





**Gambar 5.23** Halaman Register

Penjelasan elemen pada *wireframe* halaman register akan dideskripsikan dalam tabel 5.11 berikut.

**Tabel 5.11** Penjelasan Antarmuka Halaman Register

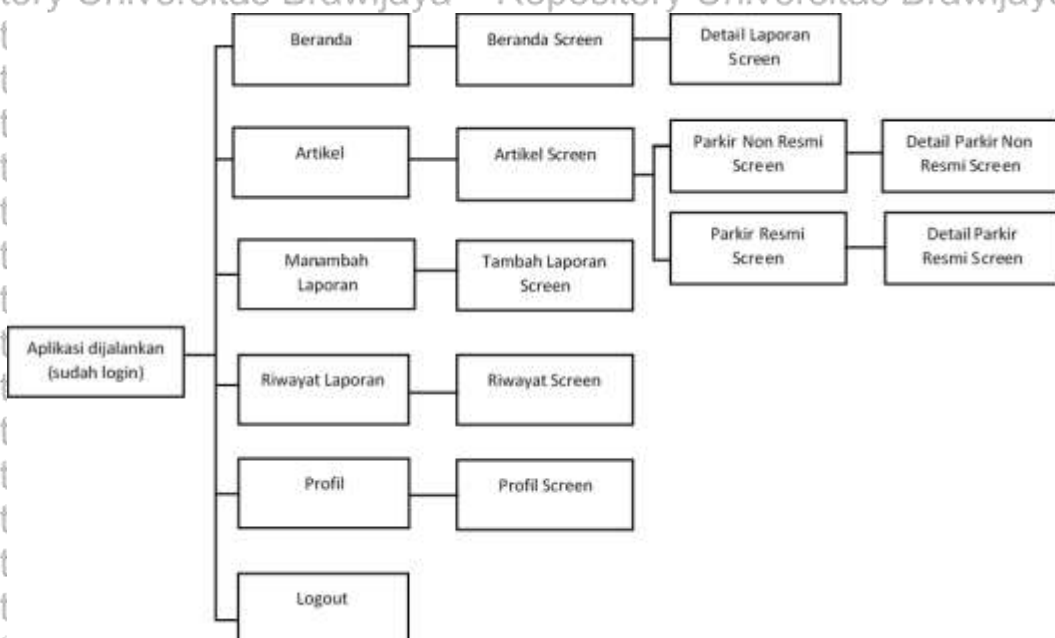
No	Nama Objek	Tipe	Deskripsi
1	Tulisan Register	<i>TextView</i>	Menampilkan tulisan register
2	Field Nama Lengkap	<i>EditText</i>	Untuk menginputkan nama lengkap
3	Field No. Hp	<i>EditText</i>	Untuk menginputkan nomor handphone
4	Field Username	<i>EditText</i>	Untuk menginputkan <i>username</i>
5	Field Email	<i>EditText</i>	Untuk menginputkan <i>email</i>
6	Field Password	<i>EditText</i>	Untuk menginputkan <i>password</i>
7	Field Confirm Password	<i>EditText</i>	Untuk menginputkan <i>confirm password</i>
8	Button Register	Button	Tombol register untuk masuk dalam sistem





### 5.1.5.1.2 Perancangan Antarmuka Aktor Pelapor

Perancangan antarmuka pada aktor pelapor dirancang untuk menampilkan sistem dari sisi aktor pelapor sehingga dapat mengoperasikan sistem. Gambar 5.24 menjelaskan rancangan alur sistem aplikasi dari sisi aktor pelapor.



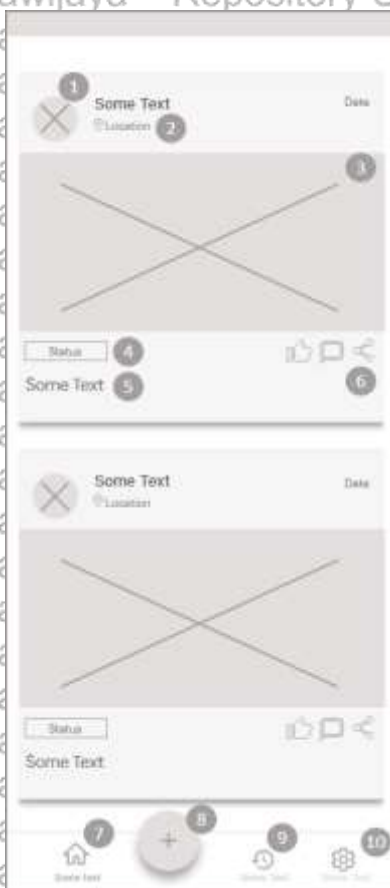
**Gambar 5.24 Screenflow aplikasi sisi pelapor**

Alur sistem pada antarmuka sisi pelapor ketika dijalankan terdiri dari antarmuka utama yaitu halaman beranda, halaman artikel, halaman menambah laporan, halaman riwayat laporan, halaman profil dan halaman *logout*. Tahapan awal pada saat menjalankan sistem akan membuka halaman beranda, dengan posisi ketika *user* sudah melakukan *login*.

#### 1. Halaman Beranda

Halaman beranda merupakan halaman awal aplikasi ketika pengguna sudah melakukan *login*. Halaman beranda terdiri dari laporan yang telah pengguna kirimkan sebelumnya. Pada halaman beranda pengguna dapat melihat laporan parkir resmi maupun non resmi dan juga dapat melihat laporan berdasarkan status laporan. Status laporan yang dimaksud terdiri laporan belum ditindaklanjuti, laporan sedang proses, dan laporan selesai. Gambar 5.25 menjelaskan rancangan *wireframe* pada halaman beranda.





Gambar 5.25 Halaman Beranda

Penjelasan elemen pada *wireframe* halaman beranda akan dideskripsikan dalam tabel 5.12 berikut.

Tabel 5.12 Penjelasan Antarmuka Halaman Beranda

No	Nama Objek	Tipe	Deskripsi
1	Foto Pengguna	<i>ImageView</i>	Foto profil pengguna
2	Lokasi	<i>TextView</i>	Lokasi laporan
3	Gambar	<i>ImageView</i>	Gambar laporan parkir
4	Status laporan	<i>TextView</i>	Status laporan pengguna
5	Deskripsi laporan	<i>TextView</i>	Deskripsi laporan pengguna
6	Ikon Vote, komentar	<i>ImageView</i>	Ikon <i>vote</i> untuk memberikan dukungan pada laporan dan ikon komentar untuk memberikan komentar pada laporan





Tabel 5.12 Penjelasan Antarmuka Halaman Beranda (Lanjutan)

No	Nama Objek	Tipe	Deskripsi
7	Ikon menu beranda	<i>ImageView</i>	Menu beranda untuk berpindah ke halaman beranda
8	Ikon menu tambah laporan	<i>ImageView</i>	Menu tambah laporan untuk menambahkan pengguna
9	Ikon menu riwayat	<i>ImageView</i>	Menu riwayat untuk berpindah ke menu riwayat
10	Ikon menu setting	<i>ImageView</i>	Menu setting menampilkan profil pengguna

## 2. Halaman Detail Laporan

Halaman detail laporan merupakan *section* dari list laporan pengguna. Selanjutnya pengguna dapat melihat komentar pengguna lainnya dan menambahkan komentar pada laporan pengguna. Gambar 5.26 menjelaskan rancangan *wireframe* pada halaman detail laporan.



Gambar 5.26 Halaman Detail Laporan

Penjelasan elemen pada *wireframe* halaman detail laporan akan dideskripsikan dalam tabel 5.13 berikut.



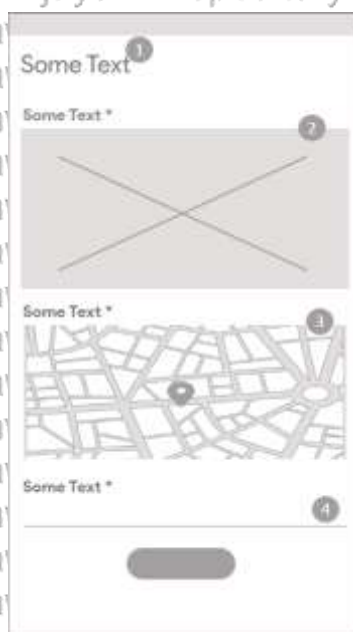


Tabel 5.13 Penjelasan Antarmuka Halaman Detail Laporan

No	Nama Objek	Tipe	Deskripsi
1	Gambar	<i>ImageView</i>	Gambar laporan parkir
2	Status Laporan	<i>TextView</i>	Status laporan pengguna
3	Ikon vote dan komentar	<i>ImageView</i>	Ikon vote untuk memberikan dukungan pada laporan dan ikon komentar untuk memberikan komentar pada laporan
4	Deskripsi Laporan	<i>TextView</i>	Deskripsi laporan pengguna
5	Komentar Pengguna	<i>TextView</i>	Berisi komentar pengguna
6	Button kirim komentar	<i>TextView</i>	Untuk mengirim komentar pada laporan pengguna

### 3. Halaman Tambah Laporan

Halaman tambah laporan merupakan halaman untuk menambahkan laporan terkait peristiwa yang dialami oleh pengguna berkaitan dengan parkir. Gambar 5.27 menjelaskan rancangan *wireframe* pada halaman tambah laporan.



Gambar 5.27 Halaman Tambah Laporan

Penjelasan elemen pada *wireframe* halaman tambah laporan akan dideskripsikan dalam tabel 5.14 berikut.





Tabel 5.14 Penjelasan Antarmuka Halaman Tambah Laporan

No	Nama Objek	Tipe	Deskripsi
1	Nama halaman	<i>TextView</i>	Nama halaman tambah laporan
2	Gambar	<i>ImageView</i>	Menambahkan gambar
3	Maps	<i>Maps</i>	Menambah lokasi kejadian
4	Deskripsi	<i>TextView</i>	Menambahkan deskripsi laporan
5	Button Post	<i>Button</i>	Tombol untuk menambahkan laporan

#### 4. Halaman Riwayat Laporan

Halaman riwayat laporan merupakan halaman untuk mengetahui riwayat laporan pengguna yang telah dikirimkan sebelumnya. Gambar 5.28 menjelaskan rancangan *wireframe* pada halaman riwayat laporan.



Gambar 5.28 Halaman Riwayat Laporan

Penjelasan elemen pada *wireframe* halaman riwayat laporan akan dideskripsikan dalam tabel 5.15 berikut.

Tabel 5.15 Penjelasan Antarmuka Halaman Riwayat Laporan

No	Nama Objek	Tipe	Deskripsi
1	Nama halaman	<i>TextView</i>	Nama halaman riwayat laporan
2	Gambar	<i>ImageView</i>	Gambar laporan pengguna

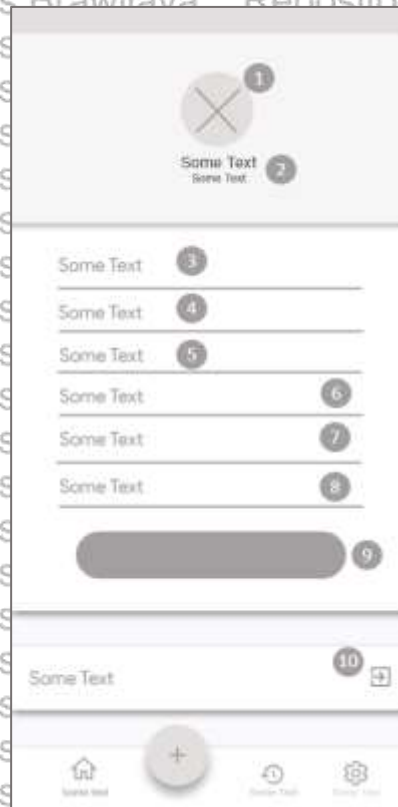




3	Lokasi	<i>TextView</i>	Lokasi laporan pengguna
4	Deskripsi	<i>TextView</i>	Menampilkan deskripsi laporan
5	Status	<i>TextView</i>	Menampilkan status laporan

### 5. Halaman Profil Pengguna

Halaman profil pengguna merupakan halaman untuk menampilkan data pengguna yang sebelumnya sudah melakukan pendaftaran akun pada sistem. Gambar 5.29 menjelaskan rancangan *wireframe* pada halaman profil pengguna.



Gambar 5.29 Halaman Profil Pengguna

Penjelasan elemen pada *wireframe* halaman profil pengguna akan dideskripsikan dalam tabel 5.16 berikut.

Tabel 5.16 Penjelasan Antarmuka Halaman Profil Pengguna

No	Nama Objek	Tipe	Deskripsi
1	Foto Pengguna	<i>ImageView</i>	Untuk menampilkan foto pengguna
2	<i>Username</i>	<i>TextView</i>	Menampilkan <i>username</i> pengguna



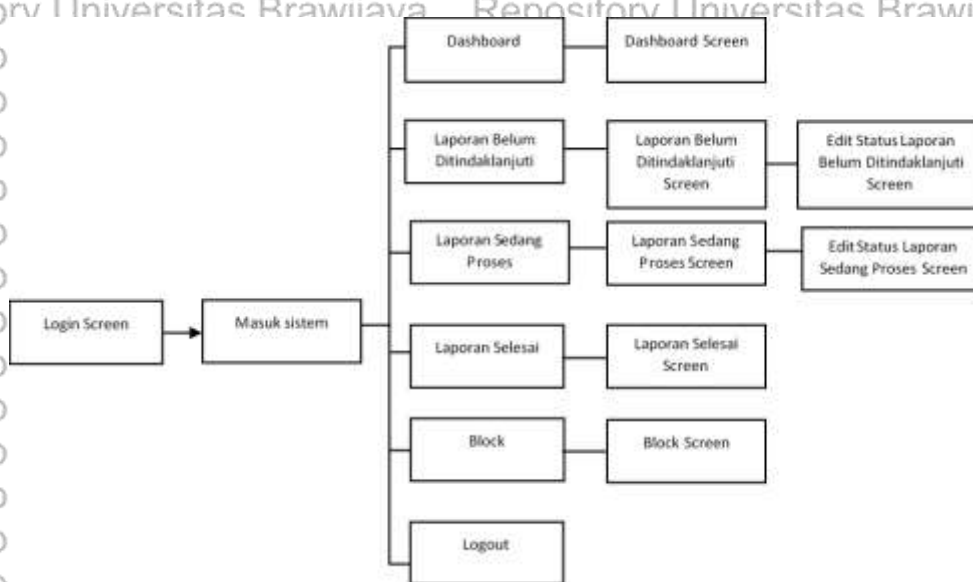


Tabel 5.16 Penjelasan Antarmuka Halaman Profil Pengguna (lanjutan)

No	Nama Objek	Tipe	Deskripsi
3	Nama Lengkap	<i>TextView</i>	Menampilkan nama lengkap pengguna
4	No. Hp	<i>TextView</i>	Menampilkan nomor hp pengguna
5	<i>Username</i>	<i>TextView</i>	Menampilkan <i>username</i> pengguna
6	<i>Email</i>	<i>TextView</i>	Menampilkan <i>email</i> pengguna
7	<i>Password</i>	<i>TextView</i>	Menampilkan <i>password</i> pengguna
8	Ulangi <i>Password</i>	<i>TextView</i>	Untuk mengulangi <i>password</i>
9	Button simpan profil	<i>Button</i>	Tombol untuk menyimpan profil pengguna
10	Button <i>logout</i>	<i>Button</i>	Tombol untuk <i>logout</i>

### 5.1.5.1.3 Perancangan Antarmuka Aktor Administrator

Perancangan antarmuka pada aktor pelapor dirancang untuk menampilkan sistem *website* dari sisi aktor administrator sehingga admin dapat mengelola data laporan. Gambar 5.30 menjelaskan rancangan *screenflow* aplikasi dari sisi aktor administrator.

Gambar 5.30 *Screenflow* aplikasi sisi administrator

Alur sistem pada antarmuka sisi administrator ketika dijalankan akan membuka halaman *login* terlebih dahulu. Selanjutnya ketika admin sudah *login* maka akan menampilkan halaman awal *dashboard*. Halaman utama dari sistem *website* yaitu halaman *dashboard*, halaman laporan belum ditindaklanjuti,





halaman laporan sedang proses, laporan selesai, dan halaman laporan tidak disetujui.

### 1. Halaman *Login Website*

Halaman *login* merupakan halaman awal pada aplikasi sehingga admin dapat masuk dalam sistem. Gambar 5.31 menjelaskan rancangan *wireframe* pada halaman *login*.



**Gambar 5.31** Halaman *Login Website*

Penjelasan elemen pada *wireframe* halaman *login* akan dideskripsikan dalam tabel 17 berikut.

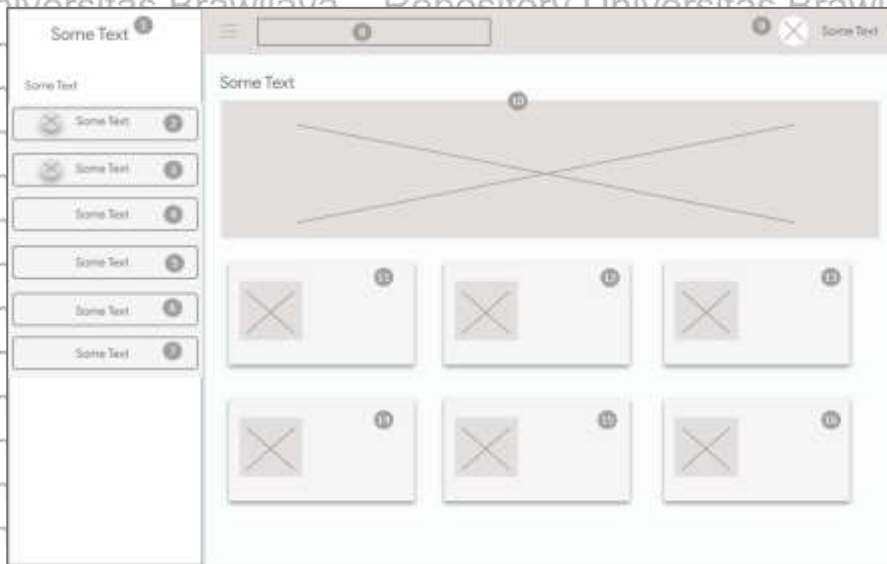
**Tabel 5.17** Penjelasan Antarmuka Halaman *Login Website*

No	Nama Objek	Tipe	Deskripsi
1	Tulisan <i>Login</i>	<i>TextView</i>	Menampilkan tulisan <i>login</i>
2	<i>Field Email</i>	<i>EditText</i>	Untuk memasukkan <i>email</i>
3	<i>Field Password</i>	<i>EditText</i>	Untuk menginputkan <i>password</i>
4	Tombol <i>Login</i>	<i>Button</i>	Tombol <i>login</i> untuk masuk dalam sistem

### 2. Halaman *Dashboard Website*

Halaman *dashboard website* merupakan halaman untuk menampilkan jumlah keseluruhan data laporan yang telah masuk pada sistem. Gambar 5.32 menjelaskan rancangan *wireframe* pada halaman *login website*. Terdapat jumlah total *user*, total laporan, total laporan belum ditindaklanjuti, total laporan selesai dan total laporan tidak disetujui. Gambar 5.32 menjelaskan rancangan *wireframe* pada halaman *dashboard*.





**Gambar 5.32** Halaman Dashboard Website

Penjelasan elemen pada *wireframe* halaman *dashboard website* akan dideskripsikan dalam tabel 5.18 berikut.

**Tabel 5.18** Penjelasan Antarmuka Halaman *Dashboard*

No	Nama Objek	Tipe	Deskripsi
1	Tulisan <i>Dashboard</i>	<i>TextView</i>	Menampilkan tulisan <i>dashboard</i>
2	Menu <i>Dashboard</i>	<i>TextView</i>	Tombol menu <i>dashboard</i>
3	Menu Laporan	<i>TextView</i>	Tombol menu laporan
4	Menu Belum Ditindaklanjuti	<i>TextView</i>	Tombol menu belum ditindaklanjuti
5	Menu Sedang Proses	<i>TextView</i>	Tombol menu sedang proses
6	Menu Selesai	<i>TextView</i>	Tombol menu selesai
7	Menu Tidak disetujui	<i>TextView</i>	Tombol menu tidak disetujui
8	Search	<i>Search</i>	Untuk melakukan pencarian
9	Profil	<i>ImageView</i>	Menampilkan image admin

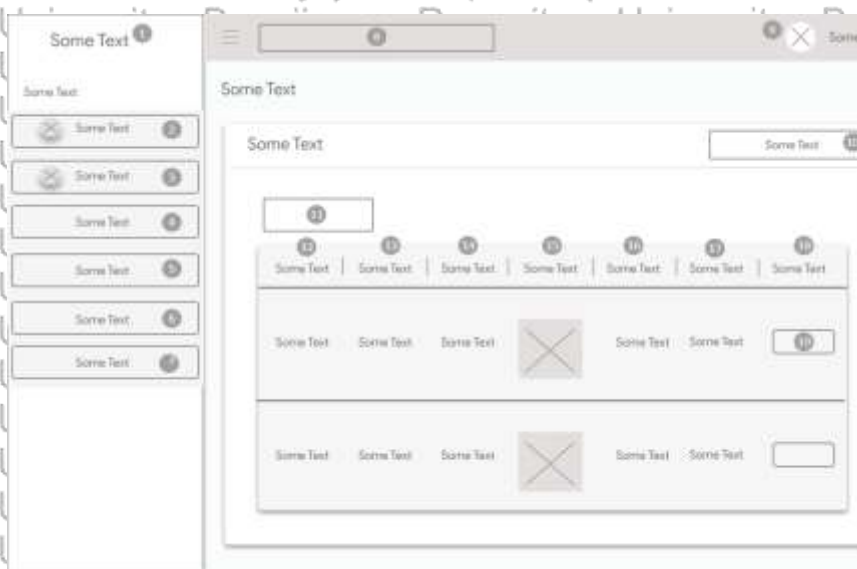


Tabel 5.18 Penjelasan Antarmuka Halaman *Dashboard* (Lanjutan)

No	Nama Objek	Tipe	Deskripsi
10	Gambar	<i>ImageView</i>	Menampilkan gambar
11	Total <i>user</i>	<i>TextView</i>	Menampilkan jumlah total <i>user</i>
12	Total laporan	<i>TextView</i>	Menampilkan jumlah total laporan
13	Total laporan belum ditindaklanjuti	<i>TextView</i>	Menampilkan jumlah laporan belum ditindaklanjuti
14	Total laporan sedang proses	<i>TextView</i>	Menampilkan jumlah laporan sedang proses
15	Total laporan selesai	<i>TextView</i>	Menampilkan jumlah laporan selesai
16	Total laporan tidak disetujui	<i>TextView</i>	Menampilkan jumlah laporan tidak disetujui

### 3. Halaman Laporan Belum Ditindaklanjuti *Website*

Halaman laporan belum ditindaklanjuti merupakan halaman untuk menampilkan laporan dengan status belum ditindaklanjuti. Data laporan ditampilkan dalam bentuk *datatable*. Gambar 5.33 menjelaskan rancangan *wireframe* pada halaman laporan belum ditindaklanjuti.



Gambar 5.33 Halaman Laporan Belum Ditindaklanjuti

Penjelasan elemen pada *wireframe* halaman laporan belum ditindaklanjuti pada *website* akan dideskripsikan dalam tabel 5.19 berikut.





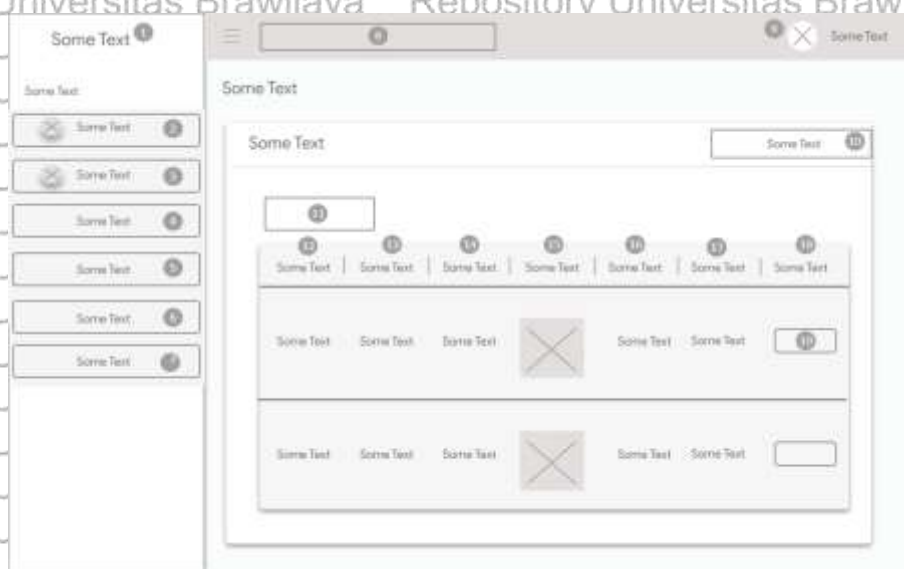
Tabel 5.19 Penjelasan Antarmuka Halaman Laporan Belum Ditindaklanjuti

No	Nama Objek	Tipe	Deskripsi
1	Tulisan GoTindak	<i>TextView</i>	Menampilkan tulisan GoTindak
2	Menu <i>Dashboard</i>	<i>TextView</i>	Tombol menu <i>dashboard</i>
3	Menu Laporan	<i>TextView</i>	Tombol menu laporan
4	Menu Belum Ditindaklanjuti	<i>TextView</i>	Tombol menu belum ditindaklanjuti
5	Menu Sedang Proses	<i>TextView</i>	Tombol menu sedang proses
6	Menu Selesai	<i>TextView</i>	Tombol menu selesai
7	Menu Tidak disetujui	<i>TextView</i>	Tombol menu tidak disetujui
8	<i>Search</i>	<i>Search</i>	Untuk melakukan pencarian
9	<i>Profil</i>	<i>ImageView</i>	Menampilkan <i>image</i> admin
10	<i>Search</i>	<i>Search</i>	Untuk mencari data laporan
11	Menu show laporan	<i>ListView</i>	Untuk menampilkan jumlah list tabel yang akan ditampilkan
12	Id pelapor	<i>TextView</i>	Menampilkan id pelapor
13	Deskripsi	<i>TextView</i>	Menampilkan deskripsi
14	Alamat	<i>TextView</i>	Menampilkan alamat
15	Foto	<i>ImageView</i>	Menampilkan gambar laporan
16	Tanggal	<i>TextView</i>	Menampilkan tanggal
17	Status	<i>TextView</i>	Menampilkan status laporan
18	Proses	<i>TextView</i>	Menampilkan button proses
19	Button Proses	<i>Button</i>	Tombol untuk memproses laporan dengan mengubah status laporan

#### 4. Halaman Laporan Sedang Proses Website

Halaman laporan sedang proses merupakan halaman untuk menampilkan laporan dengan status sedang proses. Data laporan ditampilkan dalam bentuk *datatable*. Gambar 5.34 menjelaskan rancangan *wireframe* pada halaman laporan sedang proses.





**Gambar 5.34 Halaman Laporan Sedang Proses**

Penjelasan elemen pada *wireframe* halaman laporan belum ditindaklanjuti pada *website* akan dideskripsikan dalam tabel 5.20 berikut.

**Tabel 5.20 Penjelasan Antarmuka Halaman Laporan Sedang Proses**

No	Nama Objek	Tipe	Deskripsi
1	Tulisan GoTindak	<i>TextView</i>	Menampilkan tulisan GoTindak
2	Menu <i>Dashboard</i>	<i>TextView</i>	Tombol menu <i>dashboard</i>
3	Menu Laporan	<i>TextView</i>	Tombol menu laporan
4	Menu Belum Ditindaklanjuti	<i>TextView</i>	Tombol menu belum ditindaklanjuti
5	Menu Sedang Proses	<i>TextView</i>	Tombol menu sedang proses
6	Menu Selesai	<i>TextView</i>	Tombol menu selesai
7	Menu Tidak disetujui	<i>TextView</i>	Tombol menu tidak disetujui
8	Search	<i>Search</i>	Untuk melakukan pencarian
9	Profil	<i>ImageView</i>	Menampilkan image admin
10	Search	<i>Search</i>	Untuk mencari data laporan
11	Menu show laporan	<i>ListView</i>	Untuk menampilkan jumlah list tabel yang akan ditampilkan





**Tabel 5.20 Penjelasan Antarmuka Halaman Laporan Sedang Proses (Lanjutan)**

12	Id pelapor	<i>TextView</i>	Menampilkan id pelapor
13	Deskripsi	<i>TextView</i>	Menampilkan deskripsi
14	Alamat	<i>TextView</i>	Menampilkan alamat
15	Foto	<i>ImageView</i>	Menampilkan gambar laporan
16	Tanggal	<i>TextView</i>	Menampilkan tanggal
17	Status	<i>TextView</i>	Menampilkan status laporan
18	Proses	<i>TextView</i>	Menampilkan button proses
19	Button Proses	<i>Button</i>	Tombol untuk memproses laporan dengan menguang status laporan

### 5. Halaman Edit Status Laporan Website

Halaman edit status laporan merupakan halaman untuk mengganti status laporan menjadi belum ditindaklanjuti, sedang proses, selesai dan tidak disetujui. Admin dapat mengubah status laporan sesuai dengan keadaan laporan. Gambar 5.35 menjelaskan rancangan *wireframe* pada halaman edit status laporan.



**Gambar 5.35 Halaman Edit Status Laporan**

Penjelasan elemen pada *wireframe* halaman *login* akan dideskripsikan dalam tabel 5.21 berikut.





Tabel 5.21 Penjelasan Antarmuka Halaman Edit Status Laporan

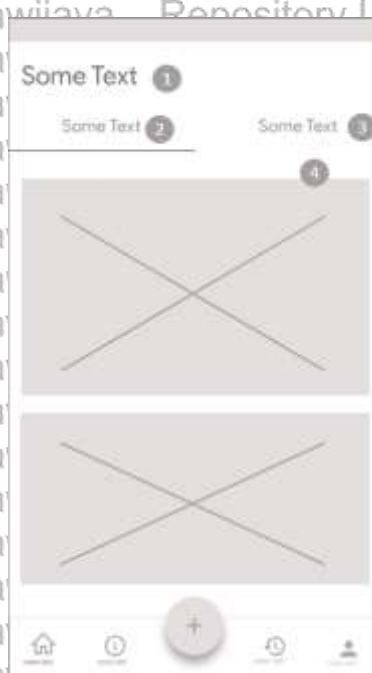
No	Nama Objek	Tipe	Deskripsi
1	Tulisan Proses	<i>TextView</i>	Menampilkan tulisan proses
2	Status	<i>Dropdown</i>	Menampilkan status laporan
3	Kategori	<i>Dropdown</i>	Menampilkan kategori laporan
4	Deskripsi	<i>TextView</i>	Menampilkan deskripsi laporan
5	Lokasi	<i>TextView</i>	Menampilkan lokasi laporan oleh pengguna
6	Button close	<i>Button</i>	Menutup <i>form</i> edit
7	Button Save	<i>Button</i>	Menyimpan <i>form</i> edit

### 5.1.5.2 Perancangan Antarmuka Iterasi Kedua

Tahapan perancangan antarmuka iterasi kedua pada aplikasi GoTindak terdapat beberapa perbaikan dan penambahan kebutuhan terkait dengan temuan masalah pada antarmuka yang telah didefinisikan sebelumnya. Oleh karena itu perlu adanya perbaikan antarmuka pada aplikasi GoTindak.

#### 5.1.5.2.1 Perancangan Antarmuka Halaman Melihat Artikel Identifikasi Parkir

Halaman melihat artikel identifikasi parkir merupakan penambahan kebutuhan dari sisi pelapor. Selain itu juga terdapat perbaikan tata letak pada menu *navigasi bottom*. Gambar 5.36 menjelaskan rancangan *wireframe* pada halaman melihat artikel identifikasi parkir laporan.



Gambar 5.36 Halaman Melihat Artikel Identifikasi Parkir





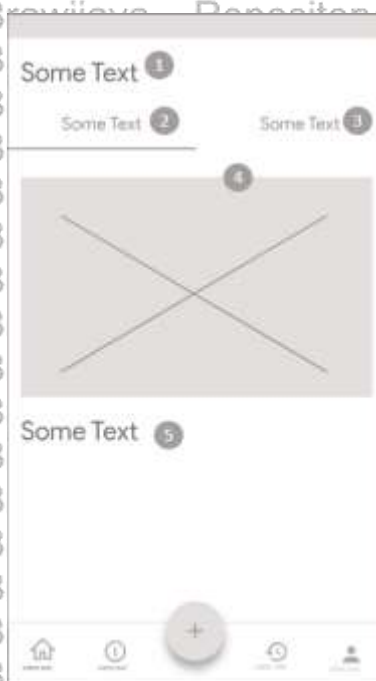
Penjelasan elemen pada *wireframe* halaman melihat artikel identifikasi parkir akan dideskripsikan dalam tabel 5.22 berikut.

**Tabel 5.22 Penjelasan Antarmuka Halaman Melihat Artikel Identifikasi Parkir**

No	Nama Objek	Tipe	Deskripsi
1	Tulisan Identifikasi Parkir	<i>TextView</i>	Menampilkan tulisan identifikasi parkir
2	Tab menu parkir non resmi	<i>Button</i>	Tombol menu artikel parkir non resmi
3	Tab menu parkir resmi	<i>Button</i>	Tombol menu artikel parkir non resmi
4	Gambar artikel	<i>ImageView</i>	Menampilkan gambar artikel

#### 5.1.5.2.2 Perancangan Antarmuka Halaman Melihat Detail Artikel Identifikasi Parkir

Halaman melihat detail artikel identifikasi parkir merupakan penambahan kebutuhan dari sisi pelapor. Selain itu juga terdapat perbaikan tata letak pada menu *navigasi bottom*. Gambar 5.37 menjelaskan rancangan *wireframe* pada halaman melihat detail artikel identifikasi parkir.



**Gambar 5.37 Halaman Melihat Detail Artikel Identifikasi Parkir**

Penjelasan elemen pada *wireframe* halaman melihat detail artikel identifikasi parkir akan dideskripsikan dalam tabel 5.23 berikut.



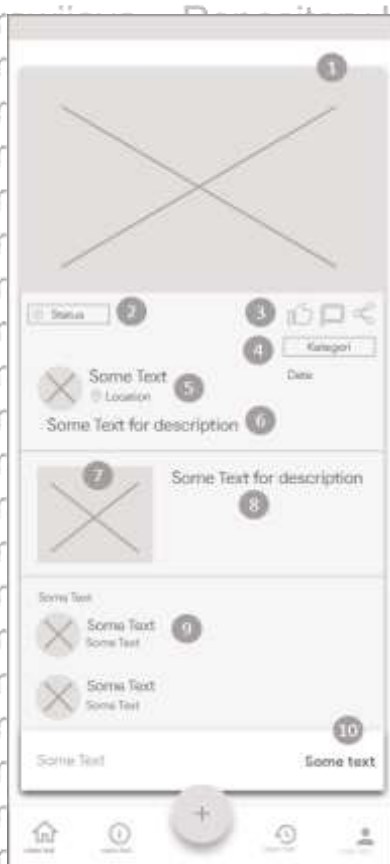


**Tabel 5.23 Penjelasan Antarmuka Halaman Melihat Detail Artikel Identifikasi Parkir**

No	Nama Objek	Tipe	Deskripsi
1	Tulisan Identifikasi Parkir	<i>TextView</i>	Menampilkan tulisan identifikasi parkir
2	Tab menu parkir non resmi	<i>Button</i>	Tombol menu artikel parkir non resmi
3	Tab menu parkir resmi	<i>Button</i>	Tombol menu artikel parkir resmi
4	Gambar artikel	<i>ImageView</i>	Menampilkan gambar artikel
5	Deskripsi	<i>TextView</i>	Menampilkan deskripsi artikel

#### 5.1.5.2.3 Perancangan Antarmuka Halaman Melihat Detail Laporan

Halaman detail laporan merupakan section dari list laporan pengguna. Selanjutnya pengguna dapat melihat komentar pengguna lainnya dan menambahkan komentar pada laporan pengguna. Pada iterasi kedua terdapat penambahan dan perbaikan tata letak. Gambar 5.38 menjelaskan rancangan *wireframe* pada halaman detail laporan.



**Gambar 5.38 Halaman Detail Laporan**





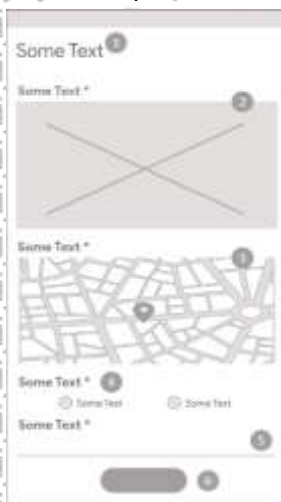
Penjelasan elemen pada *wireframe* halaman detail laporan akan dideskripsikan dalam tabel 5.24 berikut.

**Tabel 5.24 Penjelasan Antarmuka Halaman Detail Laporan**

No	Nama Objek	Tipe	Deskripsi
1	Gambar	<i>ImageView</i>	Gambar laporan parkir
2	Status Laporan	<i>TextView</i>	Status laporan pengguna
3	Ikon vote dan komentar	<i>ImageView</i>	Ikon vote untuk memberikan dukungan pada laporan dan ikon komentar untuk memberikan komentar pada laporan
4	Kategori	<i>TextView</i>	Menampilkan kategori laporan
5	Lokasi	<i>TextView</i>	Menampilkan lokasi laporan
6	Deskripsi Laporan	<i>TextView</i>	Deskripsi laporan pengguna
7	Gambar tanggapan	<i>ImageView</i>	Menampilkan foto tindak lanjut laporan
8	Catatan	<i>TextView</i>	Menambahkan catatan
9	Komentar Pengguna	<i>TextView</i>	Berisi komentar pengguna
10	Button kirim komentar	<i>TextView</i>	Untuk mengirim komentar pada laporan pengguna

#### 5.1.5.2.4 Perancangan Antarmuka Halaman Laporan

Halaman tambah laporan merupakan halaman untuk menambahkan laporan terkait peristiwa yang dialami oleh pengguna berkaitan dengan parkir. Pada iterasi kedua terdapat penambahan dan perbaikan tata letak. Gambar 5.39 menjelaskan rancangan *wireframe* pada halaman laporan.



**Gambar 5.39 Halaman Laporan**





Penjelasan elemen pada *wireframe* halaman tambah laporan akan dideskripsikan dalam tabel 5.25 berikut.

**Tabel 5.25 Penjelasan Antarmuka Halaman Tambah Laporan**

No	Nama Objek	Tipe	Deskripsi
1	Nama halaman	<i>TextView</i>	Nama halaman tambah laporan
2	Gambar	<i>ImageView</i>	Menambahkan gambar
3	Maps	<i>Maps</i>	Menambah lokasi kejadian
4	Kategori	<i>RadioButton</i>	Memilih kategori laporan
5	Deskripsi	<i>TextView</i>	Menambahkan deskripsi laporan
6	Button Post	<i>Button</i>	Tombol untuk menambahkan laporan

#### 5.1.5.2.5 Perancangan Antarmuka Halaman Beranda

Halaman beranda merupakan halaman awal aplikasi ketika pengguna sudah melakukan *login*. Halaman beranda terdiri dari list laporan yang telah pengguna kirimkan sebelumnya. Pada halaman beranda pengguna dapat melihat laporan parkir resmi maupun non resmi dan juga dapat melihat laporan berdasarkan status laporan. Status laporan yang dimaksud terdiri laporan belum ditindaklanjuti, laporan sedang proses, dan laporan selesai. Pada iterasi kedua terdapat penambahan dan perbaikan tata letak. Gambar 5.40 menjelaskan rancangan *wireframe* pada halaman beranda.



**Gambar 5.40 Halaman Beranda**





Penjelasan elemen pada *wireframe* halaman beranda akan dideskripsikan dalam tabel 5.26 berikut.

**Tabel 5.26 Penjelasan Antarmuka Halaman Beranda**

No	Nama Objek	Tipe	Deskripsi
1	Foto Pengguna	<i>ImageView</i>	Foto profil pengguna
2	<i>Username</i>	<i>TextView</i>	Menampilkan <i>username</i> pengguna
3	Lokasi	<i>TextView</i>	Lokasi laporan
4	Gambar	<i>ImageView</i>	Gambar laporan parkir
5	Status laporan	<i>TextView</i>	Status laporan pengguna
6	Deskripsi laporan	<i>TextView</i>	Deskripsi laporan pengguna
7	Ikon Vote, komentar	<i>ImageView</i>	Ikon vote untuk memberikan dukungan pada laporan dan ikon komentar untuk memberikan komentar pada laporan
8	Kategori	<i>TetxView</i>	Menampilkan kategori laporan

#### 5.1.5.2.6 Perancangan Antarmuka Halaman Riwayat Laporan

Halaman riwayat laporan merupakan halaman untuk mengetahui riwayat laporan pengguna yang telah dikirimkan sebelumnya. Pada iterasi kedua terdapat penambahan dan perbaikan tata letak. Gambar 5.41 menjelaskan rancangan *wireframe* pada halaman riwayat laporan.



**Gambar 5.41 Halaman Riwayat Laporan**





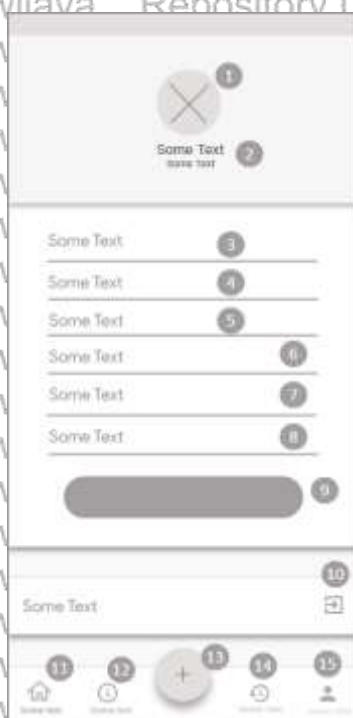
Penjelasan elemen pada *wireframe* halaman riwayat laporan akan dideskripsikan dalam tabel 5.27 berikut.

**Tabel 5.27 Penjelasan Antarmuka Halaman Riwayat Laporan**

No	Nama Objek	Tipe	Deskripsi
1	Nama halaman	<i>TextView</i>	Nama halaman riwayat laporan
2	Gambar	<i>ImageView</i>	Gambar laporan pengguna
3	<i>Username</i>	<i>TextView</i>	<i>Username</i> laporan pengguna
4	Lokasi	<i>TextView</i>	Lokasi laporan pengguna
5	Deskripsi	<i>TextView</i>	Menampilkan deskripsi laporan
6	Status	<i>TextView</i>	Menampilkan status laporan
7	Kategori	<i>TextView</i>	Menampilkan kategori laporan

#### 5.1.5.2.7 Perancangan Antarmuka Halaman Profil

Halaman profil pengguna merupakan halaman untuk menampilkan data pengguna yang sebelumnya sudah melakukan pendaftaran akun pada sistem. Pada iterasi kedua terdapat penambahan dan perbaikan tata letak. Gambar 5.42 menjelaskan rancangan *wireframe* pada halaman profil pengguna.



**Gambar 5.42 Halaman Profil**

Penjelasan elemen pada *wireframe* halaman profil pengguna akan dideskripsikan dalam tabel 5.29 berikut.





Tabel 5.28 Penjelasan Antarmuka Halaman Profil Pengguna

No	Nama Objek	Tipe	Deskripsi
1	Foto Pengguna	<i>ImageView</i>	Untuk menampilkan foto pengguna
2	<i>Username</i>	<i>TextView</i>	Menampilkan <i>username</i> pengguna
3	Nama Lengkap	<i>TextView</i>	Menampilkan nama lengkap pengguna
4	No. Hp	<i>TextView</i>	Menampilkan nomor hp pengguna
5	<i>Username</i>	<i>TextView</i>	Menampilkan <i>username</i> pengguna
6	<i>Email</i>	<i>TextView</i>	Menampilkan <i>email</i> pengguna
7	<i>Password</i>	<i>TextView</i>	Menampilkan <i>password</i> pengguna
8	Ulangi <i>Password</i>	<i>TextView</i>	Untuk mengulangi <i>password</i>
9	Button simpan profil	<i>Button</i>	Tombol untuk menyimpan profil pengguna
10	Button <i>logout</i>	<i>Button</i>	Tombol untuk <i>logout</i>
11	Ikon menu beranda	<i>ImageView</i>	Menu beranda untuk berpindah ke halaman beranda
12	Ikon menu informasi parkir	<i>ImageView</i>	Menu untuk melihat informasi seputar parkir resmi dan parkir tidak resmi
13	Ikon menu tambah laporan	<i>ImageView</i>	Menu tambah laporan untuk menambahkan pengguna
14	Ikon menu riwayat	<i>ImageView</i>	Menu riwayat untuk melihat riwayat laporan pengguna
15	Ikon menu setting	<i>ImageView</i>	Menu setting menampilkan profil pengguna

#### 5.1.5.2.8 Perancangan Antarmuka Halaman Edit Status Laporan *Website*

Halaman edit status laporan merupakan halaman untuk mengganti status laporan menjadi belum ditindaklanjuti, sedang proses, selesai dan tidak disetujui. Admin dapat mengubah status laporan sesuai dengan keadaan laporan. Pada iterasi kedua terdapat penambahan dan perbaikan tata letak. Gambar 5.43 menjelaskan rancangan *wireframe* pada halaman edit status laporan.





**Gambar 5.43 Halaman Edit Status Laporan Website**

Penjelasan elemen pada *wireframe* halaman edit status laporan website akan dideskripsikan dalam tabel 5.29 berikut.

**Tabel 5.29 Penjelasan Antarmuka Halaman Edit Status Laporan**

No	Nama Objek	Tipe	Deskripsi
1	Tulisan Proses	<i>TextView</i>	Menampilkan tulisan proses
2	Status	<i>DropDown</i>	Menampilkan status laporan
3	Kategori	<i>DropDown</i>	Menampilkan kategori laporan
4	Deskripsi	<i>TextView</i>	Menampilkan deskripsi laporan
5	Lokasi	<i>TextView</i>	Menampilkan lokasi laporan oleh pengguna
6	Catatan	<i>TextView</i>	Menambahkan catatan
7	Upload Gambar	<i>ImageView</i>	Menambahkan gambar tindakan lanjut
8	Button close	<i>Button</i>	Menutup <i>form</i> edit
9	Button Save	<i>Button</i>	Menyimpan <i>form</i> edit

## 5.2 Implementasi Sistem

Setelah melakukan perancangan pada sistem, tahapan selanjutnya yaitu melakukan implementasi sistem. Hasil perancangan yang telah didefinisikan sebelumnya akan dilakukan proses implementasi sistem. Dalam hal ini, implementasi sistem didapatkan berdasarkan analisis kebutuhan yang





sebelumnya telah dideskripsikan sesuai dengan perancangan. Implementasi sistem yang akan dideskripsikan mengenai spesifikasi sistem yang terdiri dari spesifikasi perangkat lunak, spesifikasi perangkat keras, batasan dalam melakukan implementasi, implementasi basis data, selanjutnya implementasi kode program dan implementasi *user interface* atau antarmuka.

### 5.2.1 Spesifikasi Sistem

Pengembangan sistem aplikasi GoTindak membutuhkan spesifikasi sistem untuk menunjang proses pengembangan. Lingkungan spesifikasi sistem yang harus diperhatikan, untuk mengimplementasikan sistem dari sisi perangkat keras ataupun perangkat lunak dan juga sistem operasi yang digunakan.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Perangkat keras yang dibutuhkan dalam membangun sistem aplikasi GoTindak secara keseluruhan ini menggunakan perangkat keras laptop dan juga perangkat keras *smartphone Android* yang berfungsi untuk menguji aplikasi. Spesifikasi pada masing-masing perangkat keras akan dijelaskan pada tabel 5.30 dan tabel 5.31 berikut.

**Tabel 5.30 Spesifikasi Perangkat Laptop**

Nama Komponen	Deskripsi Spesifikasi
Nama Model	ASUS A442U
Memory (RAM)	8GB
Processor	Intel Core i5-8250U, up to 3.4 GHz
Penyimpanan Internal	HDD 1TB
Arsitektur Sistem	64bit
Sistem Operasi	Windows 10 Pro

**Tabel 5.31 Spesifikasi Perangkat Smartphone *Android***

Nama Komponen	Deskripsi Spesifikasi
Nama Model	OPPO A39
Memory (RAM)	3 GB
Processor	Eight Core
Penyimpanan Internal	32 GB
Arsitektur Sistem	Cortex A-53
Sistem Operasi	<i>Android</i> 5.1 (Lollipop)





### 5.2.1.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang mendukung dalam membangun sistem aplikasi GoTindak berbasis *Android*, secara keseluruhan ini menggunakan beberapa jenis perangkat lunak. Perangkat lunak yang digunakan akan dideskripsikan dalam tabel berikut.

Tabel 5.32 Spesifikasi Perangkat Lunak

Nama Komponen	Deskripsi Spesifikasi
Editor Dokumentasi	Microsoft Office Word 2019
Editor Perancangan	1. Visual Paradigm 15.1 Community Edition 2. Adobe XD
Editor	Visual Studio Code, <i>Android Studio</i>
Bahasa	HTML 5, PHP, Java
<i>Platform Database</i>	MYSQL
Web Server	XAMPP
Browser	Google Chrome
<i>Database Server</i>	PHPMyAdmin

### 5.2.2 Batasan-batasan Implementasi

Batasan implementasi pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* menggunakan *geotagging* atau aplikasi GoTindak memiliki beberapa batasan, yang terdiri dari:

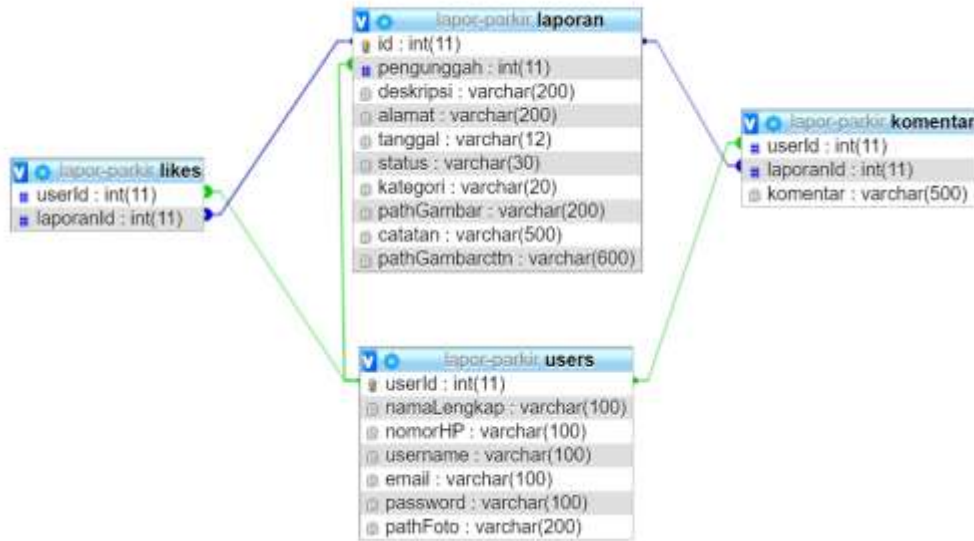
1. Sistem *website* yang dibangun menggunakan kerangka kerja CodeIgniter.
2. Aplikasi *mobile* yang dibangun menggunakan *Android studio* hanya dapat berjalan minimal pada smartphone *Android* versi 5.0 (Lollipop).
3. Aplikasi GoTindak dapat berjalan ketika ada koneksi internet.
4. Aplikasi GoTindak dalam melakukan pertukaran data menggunakan *web service* dengan format *JSON*.
5. Dalam pengembangan sistem menggunakan model *prototyping* dengan proses 2 kali iterasi.

### 5.2.3 Implementasi Basis Data

Implementasi basis data dilakukan mengikuti perancangan ERD yang telah dibuat sebelumnya. Dalam pengimplementasi basis data aplikasi GoTindak menggunakan basis data MySQL. Implementasi basis data pada aplikasi GoTindak terdapat 4 tabel yang nantinya akan digunakan. Tabel yang digunakan terdiri dari tabel laporan, tabel *users*, tabel komentar dan tabel likes. Gambar 5.44



merupakan gambar basis data yang diimplementasikan pada sistem sebagai berikut.



Gambar 5.44 Implementasi Basis Data

### 5.2.4 Implementasi Kode Program

Implementasi kode program merupakan tahapan implementasi yang telah dilakukan pada perancangan komponen sebelumnya. Setelah melakukan proses perancangan, komponen selanjutnya mengimplementasikan kode program menggunakan bahasa java untuk aplikasi *mobile* dan menggunakan kerangka kerja codeigniter pada sistem *website*.

#### 5.2.4.1 Implementasi Kode Program Membuat Laporan

Implementasi kode program berikut merupakan kode program untuk melakukan tugas membuat laporan parkir. Dalam mengimplementasikan kode program berdasarkan pada perancangan komponen yang telah didefinisikan sebelumnya pada algoritme membuat laporan. Tabel 5.33 menjelaskan implementasi program membuat laporan.

Nama kelas : `TambahLaporanActivity`

Nama metode : `post ()`

Tabel 5.33 Kode Program Membuat Laporan

No	Kode Program
1	<code>private void post(){</code>
2	<code>if (inputValidated()) {</code>
3	<code>StringRequest stringRequest = new</code>
4	<code>StringRequest (Request.Method.POST,</code>
5	<code>Constants.URL_UPLOAD LAPORAN, new</code>
6	<code>Response.Listener&lt;String&gt; () {</code>





```

7      @Override
8      public void onResponse(String response) {
9          if
10             (response.split(",")[0].split(":")[1].equals(
11                 "false")){
12                 Toast.makeText(TambahLaporanActivity.this,"Up
13                     loadsucceeded",Toast.LENGTH_LONG).show();
14                 finish();
15             }
16             else {
17                 Toast.makeText(TambahLaporanActivity.this,res
18                     ponse,Toast.LENGTH_LONG).show();
19             }
20             progressDialog.dismiss();
21         }
22         }, new Response.ErrorListener() {
23             @Override
24             public void onErrorResponse(VolleyError error) {
25                 progressDialog.dismiss();
26             }
27         }) {
28             @Override
29             protected Map<String, String> getParams() throws
30                 AuthFailureError {
31                 Map<String,String> params = new HashMap<>();
32                 params.put("nama",getName());
33                 params.put("deskripsi",etDeskripsi.getText().toS
34                     tring());
35                 params.put("pengunggah",String.valueOf(SharedPre
36                     ferenceManager.getInstance(TambahLapor
37                         anActivity.this).getUserId());
38                 params.put("tanggal",getCurrentDate());
39                 params.put("status","0");
40                 params.put("kategori",radioGroup.getCheckedRadio
41                     ButtonId(==R.id.rbNonResmi?"0":"1");
42                 params.put("alamat",getAddress());
43                 params.put("image",imageToString(bitmap));
44             }
45             return params;
46         }
47     };
48     RequestHandler.getInstance(this).addToRequestQueue(
49         stringRequest);
50     }
51 }

```

Tabel 5.34 Penjelasan Kode Program Membuat Laporan

Baris	Penjelasan
1	Penamaan <i>method post</i>
2	Memproses validasi inputan
3-6	Menginisialisasi string request dengan mengirimkan <i>method post</i> pada <i>service constant</i> url_upload_laporan





7-8	Deklarasi dengan mengirimkan <i>on response</i> pada server
9-15	Jika server merespon maka akan menampilkan <i>toast text upload success</i>
16-21	Jika tidak menampilkan <i>toast</i> dengan <i>text.show</i>
22-27	Melakukan deklarasi respon dengan progress dialog dismiss
28-29	Memanggil parameter <i>getParam</i>
30-47	Mengirim param string pada server
48-51	Menambahkan <i>request</i> pada <i>queue</i>

#### 5.2.4.2 Implementasi Kode Program Melihat Laporan

Implementasi kode program berikut merupakan kode program untuk melakukan tugas melihat laporan parkir. Dalam mengimplementasikan kode program berdasarkan pada perancangan komponen yang telah didefinisikan sebelumnya pada algoritme melihat laporan. Tabel 5.36 menjelaskan implementasi program melihat laporan.

Nama kelas : BerandaFragment

Nama metode : onCreateView()

Tabel 5.35 Kode Program Melihat Laporan

No	Kode Program
1	public View onCreateView(LayoutInflater inflater, ViewGroup
2	container, Bundle savedInstanceState) {
3	rootView = inflater.inflate(R.layout.fragment_beranda,
4	container, false);
5	recyclerView =
6	rootView.findViewById(R.id.berandaRecyclerView);
7	progressBar =
8	rootView.findViewById(R.id.progressBarBeranda);
9	recyclerView.setLayoutManager(new LinearLayoutManager(getCo
10	ntext()));
11	StringRequest stringRequest = new
12	StringRequest(Request.Method.GET, Constants.URL_GET_LAPORAN,
13	new Response.Listener<String>() {
14	@Override
15	public void onResponse(String response) {
16	final ArrayList<Laporan> laporan = new ArrayList<>();
17	}
18	},
19	new Response.ErrorListener() {
20	@Override
21	public void onErrorResponse(VolleyError error) {
22	}
23	}
24	});
25	requestQueue.add(stringRequest);
26	}





Tabel 5.36 Penjelasan Kode Program Melihat Laporan

Baris	Penjelasan
1-2	Menginisialisasi oncreateview
3-4	Melakukan inflate pada layout fragment beranda
5-6	Menambahkan reyclerviewview sesuai dengan id pada berandareyclerview
7-8	Menambahkan reyclerviewview sesuai dengan id pada progressbarberanda
9-10	Mengatur layout manager menjadi <i>linier</i> layout manager
11-13	Melakukan request metode get utl
14-19	Mengirim response listener dan menginisiasi respon pada server dengan memproses data arraylist
20-25	Menginisiasi respon error pada volley
26	Menambahkan permintaan pada request ke queue

### 5.2.4.3 Implementasi Kode Program Melihat Laporan Sedang Proses

Implementasi kode program berikut merupakan kode program untuk melakukan tugas melihat laporan sedang proses. Dalam mengimplementasikan kode program berdasarkan pada perancangan komponen yang telah didefinisikan sebelumnya pada algoritme melihat laporan sedang proses. Tabel 5.37 menjelaskan implementasi program melihat laporan sedang proses.

Nama kelas : `m_sedangproses.php`

Nama metode : `getAllProses()`

Tabel 5.37 Kode Program Laporan Sedang Proses

No	Kode Program
1	<code>public function getAllProses()</code>
2	<code>{</code>
3	<code>    \$client = new Client();</code>
4	<code>    \$response = \$client-</code>
5	<code>    &gt;request('POST', 'http://localhost/laporparkir/v1/getLap-</code>
6	<code>    oranStatus.php?status=1');</code>
7	<code>    </code>
8	<code>    \$result = JSON_decode(\$response-&gt;getBody()-</code>
9	<code>    &gt;getContents(), true);</code>
10	<code>    return \$result;</code>
11	<code>}</code>





Tabel 5.38 Penjelasan Kode Program Laporan Sedang Proses

Baris	Penjelasan
1	Deklarasi function getAllproses
2-3	Memanggil <i>client</i>
4-6	Maka akan mengirim respon pada client dengan memanggil request method get dan url ke service getlaporanstatus.php dengan status sama dengan satu
7-9	Mengambil result dengan mengirimkan response <i>JSON_decode</i> jadi array
10-11	Mengembalikan fungsi <i>result</i>

#### 5.2.4.4 Implementasi Kode Program Mengupload Foto Sedang Proses

Implementasi kode program berikut merupakan kode program untuk melakukan tugas mengunggah foto sedang proses. Dalam mengimplementasikan kode program berdasarkan pada perancangan komponen yang telah didefinisikan sebelumnya pada algoritme mengunggah foto sedang proses. Tabel 5.39 menjelaskan implementasi program mengunggah foto sedang proses.

Nama kelas : `m_sedangproses.php`

Nama metode : `ubah()`

Tabel 5.39 Kode Program Mengupload Foto Sedang Proses

No	Kode Program
1	<code>public function ubah(\$id, \$status, \$kategori, \$deskripsi,</code>
2	<code>\$catatan, \$image) / start</code>
3	<code>{</code>
4	<code>    \$client = new Client();</code>
5	<code>    print_r(\$image);</code>
6	<code>    if (empty(\$image)) {</code>
7	<code>        \$response = \$client-&gt;request('GET',</code>
8	<code>        "http://localhost/lapor-</code>
9	<code>        parkir/v1/ubah.php?id=\$id&amp;status=\$status&amp;kategori=\$kategori</code>
10	<code>        &amp;deskripsi=\$deskripsi&amp;catatan=\$catatan");</code>
11	<code>    } else {</code>
12	<code>        </code>
13	<code>        \$response = \$client-&gt;request('GET',</code>
14	<code>        "http://localhost/lapor-</code>
15	<code>        parkir/v1/ubah.php?id=\$id&amp;status=\$status&amp;kategori=\$kategori</code>
16	<code>        &amp;deskripsi=\$deskripsi&amp;catatan=\$catatan&amp;image=\$image");</code>
17	<code>    }</code>
18	<code>    print_r("http://localhost/lapor-</code>
19	<code>    parkir/v1/ubah.php?id=\$id&amp;status=\$status&amp;kategori=\$kategori</code>
20	<code>    &amp;deskripsi=\$deskripsi&amp;catatan=\$catatan&amp;image=\$image");</code>
21	<code>    \$result = <i>JSON_decode</i>(\$response-&gt;getBody()-</code>
22	<code>    &gt;getContents(), true);</code>
23	<code>    return \$result;</code>
24	<code>}</code>
25	<code>}</code>





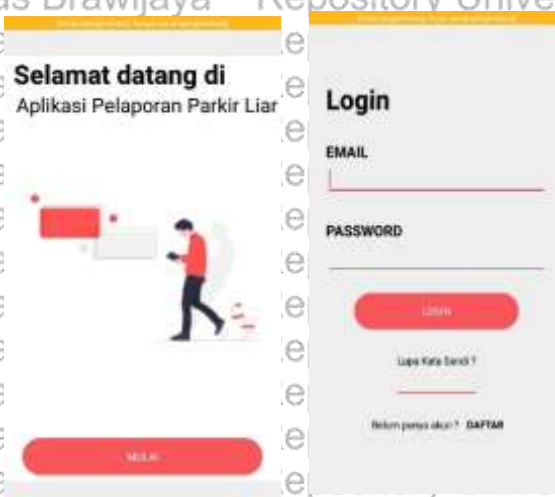
Tabel 5.40 Penjelasan Kode Program Mengupload Foto Sedang Proses

Baris	Penjelasan
1-2	Deklarasi <i>function</i> ubah()
3-4	Memanggil client
5	Menampilkan <i>variable image</i>
6	Jika image sama dengan null
7-10	Maka akan mengirim response kepada client dengan memanggil request menggunakan method Get dan url ke service ubah.php tanpa image
11-18	Respon kepada client dengan memanggil <i>request method get</i> dan url ke service.php dengan image
19-21	Menampilkan url service ubah.php
22-23	Mengambil result dengan mengirim response <i>JSON_decode</i> jadi array
24	Mengembalikan fungsi result
25	<i>end</i>

## 5.2.5 Implementasi Antarmuka

### 5.2.5.1 Implementasi Antarmuka Halaman Login

Berikut merupakan implementasi *user interface* halaman login dan terdapat informasi memasukkan email, password dan button login. Juga terdapat bantuan lupa kata sandi. Jika pengguna belum mempunyai akun maka terdapat link yang mengarahkan untuk melakukan registrasi akun pada halaman register. Implementasi *user interface* login ditunjukkan dalam Gambar 5.45



Gambar 5.45 Implementasi Antarmuka Halaman Login





### 5.2.5.2 Implementasi Antarmuka Halaman Register Akun

Implementasi *user interface* halaman register yang terdapat field nama lengkap, nomor hp, *username*, *email*, *password* dan *confirm password* serta terdapat *button register*. Ketika pengguna memasukkan password maka tampilan isi password menjadi tak terlihat. Implementasi *user interface* register ditunjukkan dalam Gambar 5.46.

Gambar 5.46 Implementasi Antarmuka Halaman Register Akun

### 5.2.5.3 Implementasi Antarmuka Melihat Daftar Laporan

Implementasi *user interface* halaman melihat daftar laporan terdapat informasi setiap *section* untuk pengguna. Isi dalam *section* tersebut terdapat nama pengguna, lokasi, judul, status laporan, jenis parkir, tanggal dan gambar. Implementasi antarmuka melihat daftar laporan ditunjukkan dalam Gambar 5.47.



Gambar 5.47 Implementasi Antarmuka Halaman Daftar Laporan





#### 5.2.5.4 Implementasi Antarmuka Halaman Detail Laporan

Implementasi *user interface* halaman detail laporan terdapat informasi setiap *section* untuk pengguna. Isi dalam *section* tersebut terdapat nama pengguna, lokasi, judul, status laporan, jenis parkir, tanggal dan gambar. Pengguna juga dapat menambahkan komentar dan melihat komentar. Implementasi antarmuka halaman detail laporan ditunjukkan dalam Gambar 5.48.



Gambar 5.48 Implementasi Antarmuka Halaman Detail Laporan

#### 5.2.5.5 Implementasi Antarmuka Halaman Riwayat Laporan

Berikut merupakan hasil implementasi halaman laporan yang telah pengguna laporkan. Pengguna juga dapat melihat status laporan apakah sudah ditindaklanjuti atau belum. Implementasi *user interface* halaman riwayat laporan ditunjukkan dalam Gambar 5.49.



Gambar 5.49 Implementasi Antarmuka Halaman Laporan Pelapor





### 5.2.5.6 Implementasi Antarmuka Halaman Profil Pengguna

Berikut merupakan halaman profil pengguna, yang mana pengguna dapat melakukan *update* profil atau mengganti *password* baru.



Gambar 5.50 Implementasi Antarmuka Halaman Profil Pengguna

### 5.2.5.7 Implementasi Antarmuka Website Halaman Login

Berikut merupakan halaman *login* untuk web yang mana halaman awal yang muncul ketika admin mengakses *website* pelaporan masyarakat. Pengguna harus melakukan *login* terlebih dahulu pada halaman ini untuk dapat mengakses web tersebut. Implementasi *user interface* halaman login ditunjukkan dalam Gambar 5.51.



Gambar 5.51 Implementasi Antarmuka Website Halaman Login





### 5.2.5.8 Implementasi Antarmuka Website Halaman Dashboard

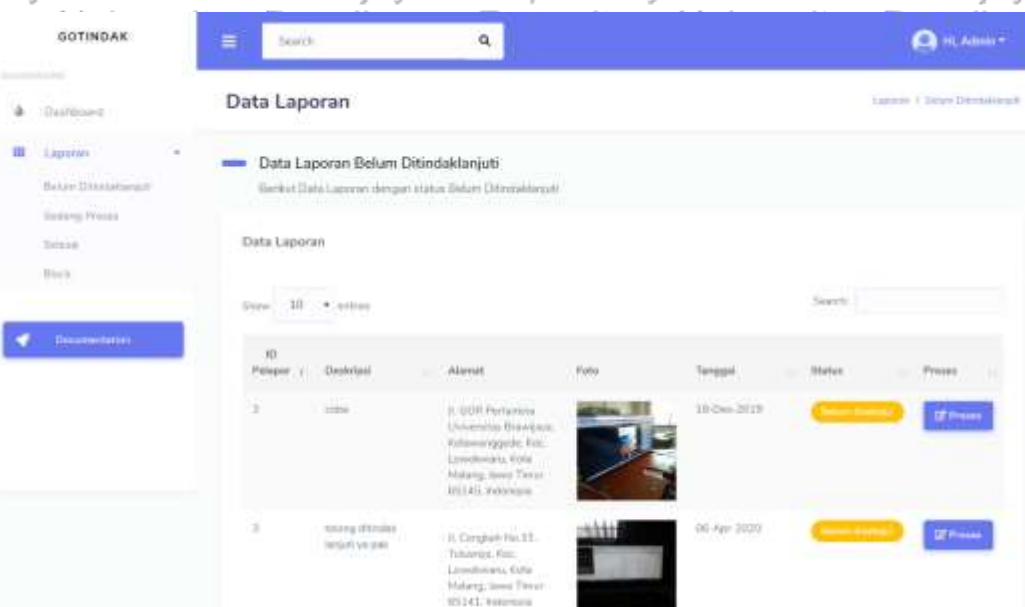
Berikut merupakan implementasi halaman *dashboard* pada *website*, dengan menampilkan data jumlah laporan, total laporan selesai, diproses, dan laporan report. Implementasi *user interface* halaman Dashboard laporan ditunjukkan dalam Gambar 5.52.



Gambar 5.52 Implementasi Antarmuka Website Halaman Dashboard

### 5.2.5.9 Implementasi Antarmuka Website Halaman Status Belum Ditinjau

Berikut merupakan implementasi halaman laporan dengan status belum ditinjau merupakan halaman yang menampilkan secara seluruh laporan yang memiliki status belum ditinjau. Berikut hasil dari implementasi halaman status belum ditinjau. Implementasi *user interface* halaman status belum ditinjau ditunjukkan dalam Gambar 5.53.



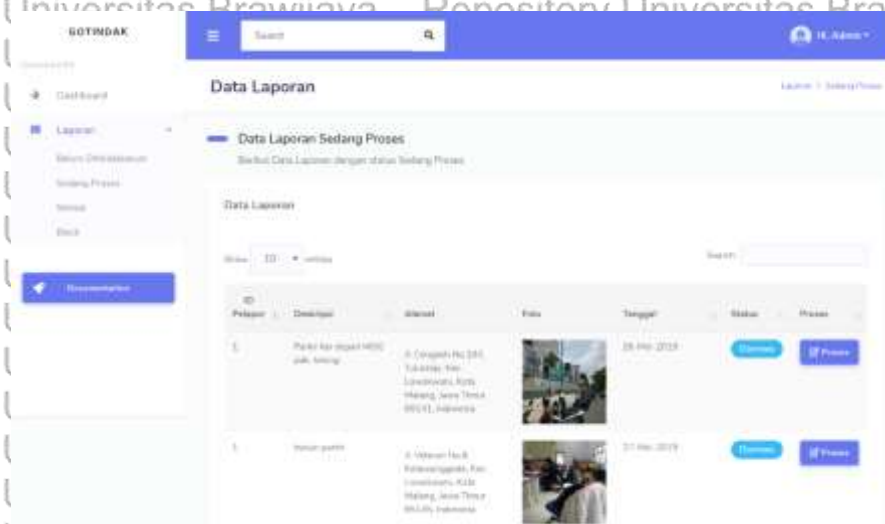
Gambar 5.53 Implementasi Antarmuka Website Halaman Status Belum Ditinjau





#### 5.2.5.10 Implementasi Antarmuka Website Halaman Status Sedang Diproses

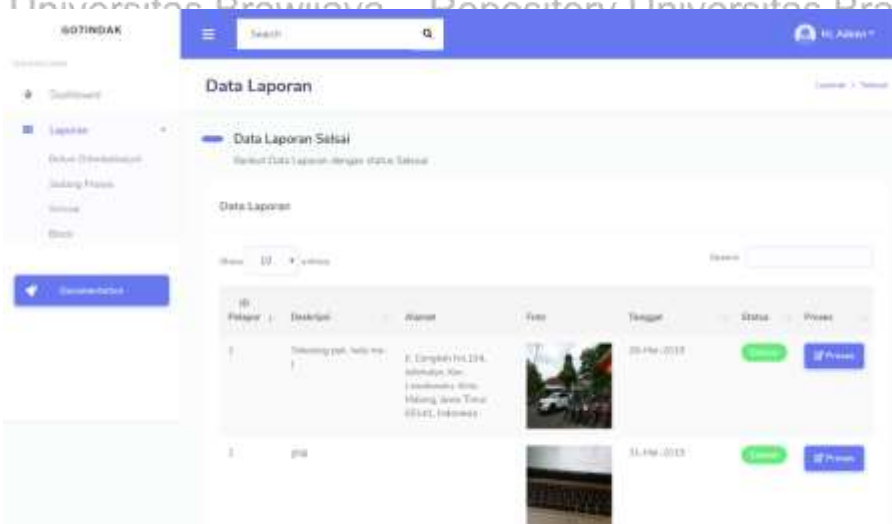
Berikut merupakan implementasi halaman laporan dengan status sedang diproses merupakan halaman yang menampilkan secara seluruh laporan yang memiliki status sedang diproses. Berikut hasil dari implementasi halaman status sedang diproses. Implementasi *user interface* halaman status sedang proses ditunjukkan dalam Gambar 5.54.



Gambar 5.54 Implementasi Antarmuka Website Halaman Status Sedang Diproses

#### 5.2.5.11 Implementasi Antarmuka Website Halaman Status Selesai

Berikut merupakan implementasi halaman laporan dengan status sedang selesai merupakan halaman yang menampilkan secara seluruh laporan yang memiliki status selesai. Berikut hasil dari implementasi halaman status selesai. Implementasi *user interface* halaman status selesai ditunjukkan dalam Gambar 5.55.



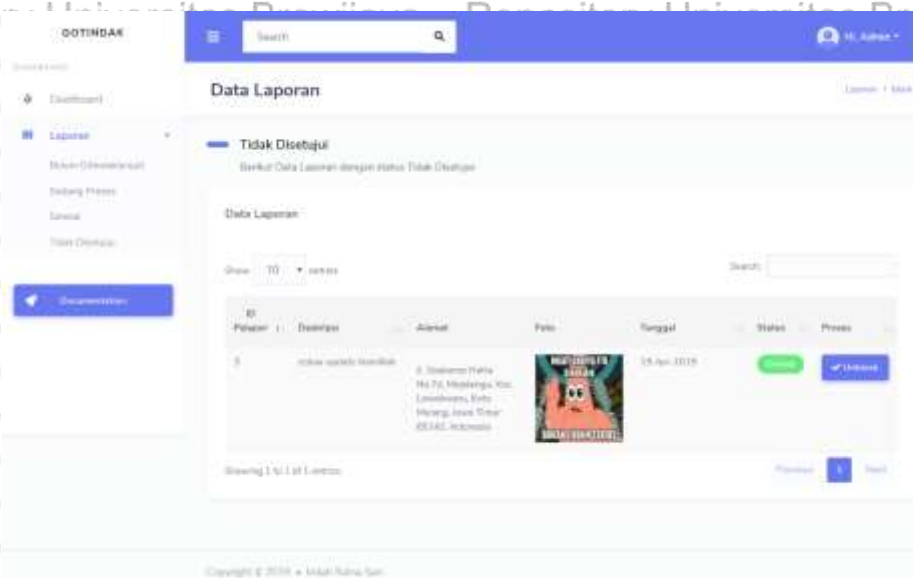
Gambar 5.55 Implementasi Antarmuka Website Halaman Status Selesai





### 5.2.5.12 Implementasi Antarmuka Website Halaman Status Tidak Disetujui

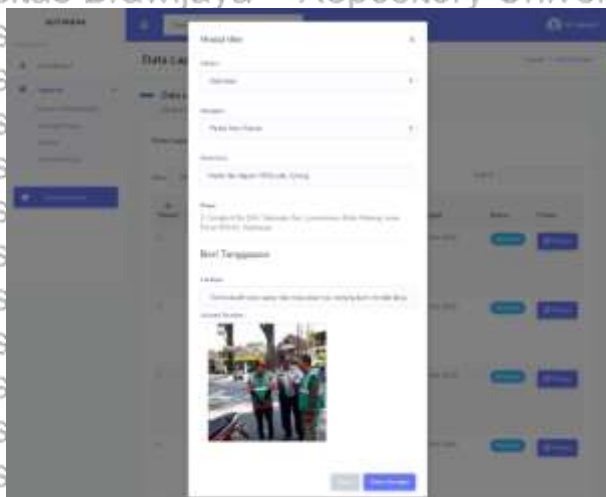
Berikut merupakan implementasi halaman laporan dengan status tidak disetujui merupakan halaman yang menampilkan seluruh laporan yang memiliki status tidak disetujui. Berikut hasil dari implementasi halaman status tidak disetujui. Implementasi *user interface* halaman status tidak disetujui ditunjukkan dalam Gambar 5.56.



Gambar 5.56 Implementasi Antarmuka Website Halaman Status Tidak Disetujui

### 5.2.5.13 Implementasi Antarmuka Website Halaman Detail Laporan

Berikut merupakan implementasi halaman detail laporan pada status belum ditinjau, sedang diproses, selesai dan tidak disetujui. Pengguna *website* dapat mengubah status pada setiap laporan. Implementasi *user interface* halaman status tidak disetujui ditunjukkan dalam Gambar 5.57.



Gambar 5.57 Implementasi Antarmuka Website Halaman Detail Laporan



## BAB 6 PENGUJIAN

Pengujian sistem merupakan tahapan yang dilakukan setelah sistem dibangun sepenuhnya. Pada pengujian sistem ini bertujuan untuk mengetahui apakah sistem yang telah dibangun sudah sesuai dengan analisis dan perancangan sistem, serta untuk menghindari adanya kesalahan (*bug*) yang terdapat pada sistem. Tahapan pengujian pada aplikasi *mobile* identifikasi parkir non resmi berdasarkan konsep *crowdsourcing* ini terdapat dua tahap yakni pengujian fungsional dan pengujian non fungsional. Teknik pengujian fungsional yang dilakukan pada sistem terdapat dua jenis pengujian yaitu *white box* dengan melakukan pengujian dari segi pengujian unit dan integrasi, sedangkan untuk pengujian *blackbox* dari segi pengujian validasi. Pada tahapan pengujian non fungsional menggunakan pengujian *usability* dan *compatibility*.

### 6.1 Pengujian Unit

Pengujian unit mendeskripsikan pengujian perangkat lunak pada unit atau *individual components* pada sistem aplikasi yang telah diimplementasi. Tujuan dilakukan pengujian unit untuk mengetahui unit atau komponen pada pada sistem dapat berjalan dengan benar berdasarkan fungsionalitas. Teknik pengujian unit kali ini menggunakan *basis path testing*. Dalam hal ini, *basis path testing* berfungsi untuk mengetahui pengujian dalam setiap jalur algoritme yang telah diimplementasikan.

#### 6.1.1 Pengujian Unit *Method post()*

*Method post()* merupakan *method* untuk membuat laporan yang nantinya data laporan disimpan dalam *database*. *Method post()* berada dalam kelas *TambahLaporanActivity* yang menghubungkan antara *view* dan *model* pada *database*. *Method post* yang akan dilakukan pegujian sesuai dengan perancangan algoritme yang telah diimplementasikan dalam kode program. Selanjutnya memodelkan algoritme dalam bentuk *flow graph* yang nantinya akan dilakukan perhitungan *basis path testing* yang terdapat dalam gambar 6.1. Setelah itu, melakukan perhitungan *cyclomatic complexity* hasil pengujian tiap jalurnya. Tabel 6.1 merupakan penjelasan dari *method post()*.

Tabel 6.1 Algoritme *Method post()*

No	Kode Program
1	Start
1	Inisialisasi
2	if (memproses validasi inputan)
3	Meginisialisasi <code>stringRequest (Mengirim.Method.POST,</code>
3	dengan service pada <code>Constants.URL_UPLOAD_LAPORAN)</code> {
4	Deklarasi <code>onResponse {</code>
5	if (server merespon)) {
6	menampilkan toast dengan
6	text, "Uploadsucceeded".show();
7	}
8	} else {

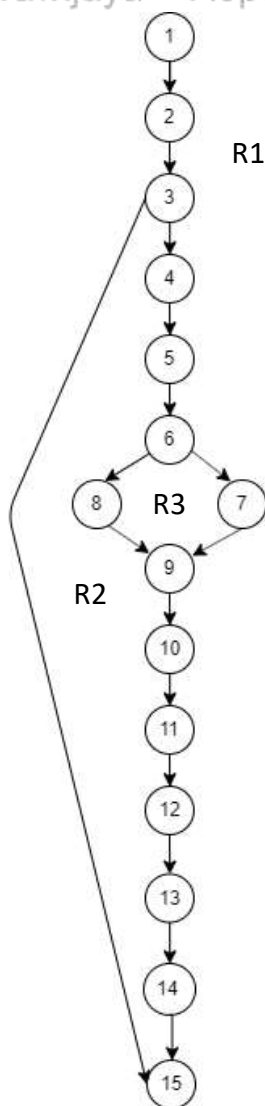




```

8      menampilkan toast."Upload tidak sukses" dengan
8      text.show();
9      }
10     progressDialog.dismiss();}}
11     Deklarasi onResponse {progressDialog.dismiss();}
11     Memambahkan parameter getParams()
11     Mengirim params nama ke server
11     Mengirim params deskripsi ke server
11     Mengirim params pengunggah ke server
11     Mengirim params tanggal ke server
11     Mengirim params status ke server
11     Mengirim params kategori ke server
11     Mengirim params alamat ke server
12     Mengirim params image ke server
13     return params;}
14     Menambahkan the request ke RequestQueue.
15     Selesai
    
```

1. **Basis path testing**  
 a. **Flow Graph**



Gambar 6.1 Flow Graph Method post()





### b. Cyclomatic Complexity ( $V(G)$ )

$$1. V(G) = \text{Jumlah Region}$$

$$= 3$$

$$2. V(G) = \text{Jumlah Edge} - \text{Jumlah Node} + 2$$

$$= 16 - 15 + 2$$

$$= 3$$

$$3. V(G) = \text{Predicate Node} + 1$$

$$= 2 + 1$$

$$= 3$$

### c. Independent Path

1. Jalur 1: 1-2-3-15

2. Jalur 2: 1-2-3-4-5-6-7-9-10-11-12-13-14-15

3. Jalur 3: 1-2-3-4-5-6-8-9-10-11-12-13-14-15

### d. Desain Kasus Uji

Tabel 6.2 Kasus Uji Pengujian Unit *Method post()*

No. Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil yang didapatkan	Status
1	Ketika melakukan validasi masukan	Berhasil melakukan request pada queue	Berhasil melakukan request pada queue	Valid
2	Ketika masukan berhasil dikirim pada server	Sistem akan menampilkan pesan "Upload Sukses"	Sistem akan menampilkan pesan "Upload Sukses"	Valid
3	Ketika masukan belum berhasil dikirim pada server	Sistem akan menampilkan pesan "Upload tidak sukses"	Sistem akan menampilkan pesan "Upload Sukses"	Valid

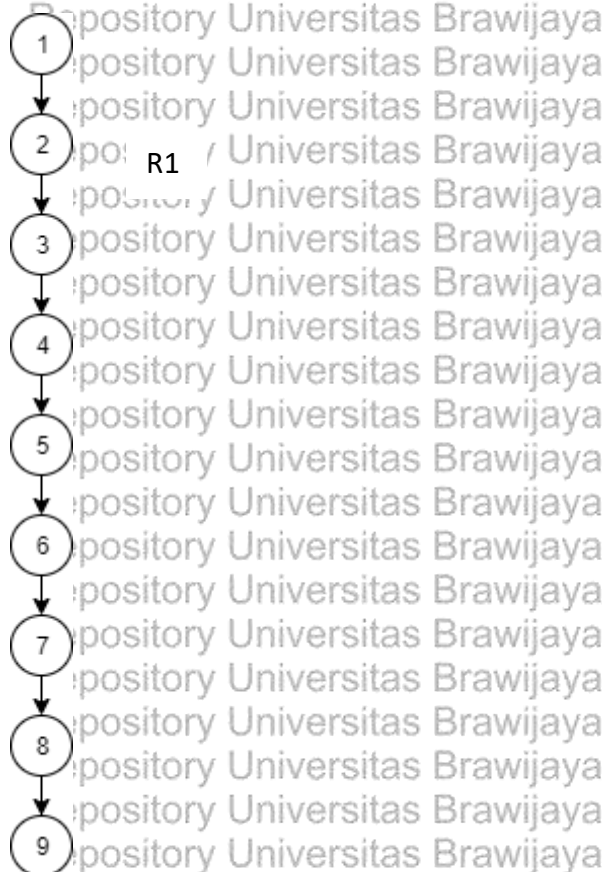
### 6.1.2 Pengujian Unit *Method onCreateView()*

*Method onCreateView()* merupakan method untuk melihat daftar laporan. *Method onCreateView()* berada dalam kelas Beranda Fragment yang menghubungkan antara *view* dan *model* pada *database*. *Method* yang akan dilakukan pengujian sesuai dengan perancangan algoritme yang telah diimplementasikan dalam kode program. Selanjutnya memodelkan algoritme dalam bentuk *flow graph* yang nantinya akan dilakukan perhitungan *basis path testing*. Setelah itu, melakukan perhitungan *cyclomatic complexity* hasil pengujian tiap jalurnya. Tabel 6.3 merupakan penjelasan dari *method onCreateView()*.



Tabel 6.3 Algoritme *Method onCreateView()*

No	Kode Program
1	Start
2	Inisialisasi <code>onCreateView</code>
3	Menginflate <code>R.layout.fragment_beranda</code>
3	Menambahkan <code>recyclerView</code> dengan <code>findViewById</code> <code>berandaRecyclerView</code>
3	Menambahkan <code>recyclerView</code> dengan <code>findViewById</code> <code>progressBarBeranda</code> );
3	Mengatur <code>layout manager</code> menjadi <code>LinearLayoutManager</code>
4	Melakukan <code>Request.Method.GET, url,</code>
5	Mengirim <code>Response.Listener&lt;String&gt;</code>
5	Inisialisai <code>onResponse</code> ( <code>String response</code> )
6	Memproses <code>ArrayList</code> laporan
7	Mengirim <code>Response.ErrorListener()</code>
7	Inisialisai <code>onErrorResponse</code> ( <code>VolleyError error</code> )
8	Menambahkan permintaan pada <code>request to queue</code>
9	End

1. **Basis path testing**a. **Flow Graph**Gambar 6.2 **Flow Graph Method onCreateView()**





### b. Cyclomatic Complexity (V(G))

$$1. V(G) = \text{Jumlah Region}$$

$$= 1$$

$$2. V(G) = \text{Jumlah Edge} - \text{Jumlah Node} + 2$$

$$= 8 - 9 + 2$$

$$= 1$$

$$3. V(G) = \text{Predicate Node} + 1$$

$$= 0 + 1$$

$$= 1$$

### c. Independent Path

1. Jalur 1: 1-2-3-4-5-6-7-8-9

### d. Desain Kasus Uji

Tabel 6.4 Kasus Uji Pengujian Unit *Method onCreateView()*

No. Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil yang didapatkan	Status
1	Melakukan proses permintaan data untuk menampilkan informasi laporan	Sistem berhasil menampilkan informasi laporan parkir pada halaman beranda	Sistem berhasil menampilkan informasi laporan parkir pada halaman beranda	Valid

### 6.1.3 Pengujian Unit *Method getAllProses()*

*Method getAllProses()* merupakan method untuk melihat daftar laporan. *Method getAllProses()* berada dalam class *m\_sedangproses* yang menghubungkan antara view dan model pada database. *Method getAllProses()* akan dilakukan pengujian sesuai dengan perancangan algoritma yang telah diimplementasikan dalam kode program. Selanjutnya menerapkan algoritma dalam bentuk *flow graph* yang akan dilakukan perhitungan *basis path testing* yang terdapat dalam gambar 6.3. Setelah itu, melakukan perhitungan *cyclomatic complexity* hasil pengujian tiap jalurnya. Tabel 6.5 merupakan penjelasan dari *method getAllProses()*.

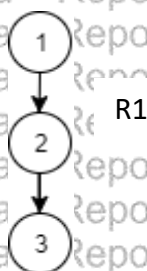
Tabel 6.5 Algoritme *Method getAllProses()*

No	Uraian
1	Start
2	Memanggil client = client Maka response = client memanggil request dengan method GET dan URL ke service getLaporanstatus.php dengan status=1 Mengambil result = response <i>JSON_decode</i> jadi array Mengembalikan fungsi result
3	End





1. **Basis path testing**  
a. **Flow Graph**



Gambar 6.3 Flow Graph Method *getAllProses()*

b. **Cyclomatic Complexity (V(G))**

1.  $V(G) = \text{Jumlah Region} - 1$

$= 1$

2.  $V(G) = \text{Jumlah Edge} - \text{Jumlah Node} + 2$

$= 2 - 3 + 2$

$= 1$

3.  $V(G) = \text{Predicate Node} + 1$

$= 0 + 1$

$= 1$

c. **Independent Path**

1. Jalur 1 : 1-2-3

d. **Desain Kasus Uji**

Tabel 6.6 Kasus Uji Penguujian Unit *Method getAllProses()*

No. Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil yang didapatkan	Status
1	Melakukan proses permintaan data untuk menampilkan informasi laporan dengan status sedang proses	Sistem berhasil menampilkan informasi laporan parkir dengan status sedang proses pada halaman sedang proses	Sistem berhasil menampilkan informasi laporan parkir dengan status sedang proses pada halaman sedang proses	Valid





#### 6.1.4 Pengujian Unit *Method* ubah()

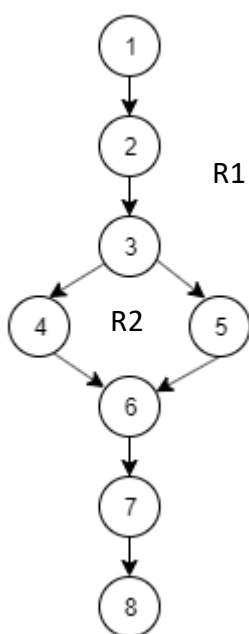
*Method* ubah() merupakan method untuk melihat daftar laporan. *Method* ubah() berada dalam kelas *m\_sedangproses* yang menghubungkan antara view dan model pada *database*. *Method* yang akan dilakukan pengujian sesuai dengan perancangan algoritme yang telah diimplementasikan dalam kode program. Selanjutnya memodelkan algoritme dalam bentuk *flow graph* yang nantinya akan dilakukan perhitungan *basis path testing* yang terdapat dalam gambar 6.4 itu, melakukan perhitungan *cyclomatic complexity* hasil pengujian tiap jalurnya. Tabel 6.7 merupakan penjelasan dari *method* ubah()

**Tabel 6.7 Algoritme *Method* ubah()**

No	Uraian
1	Start
2	Memanggil client = client Menampilkan variabel image
3	IF (image = null) (
4	Maka response = client memanggil request dengan method GET dan URL ke service ubah.php tanpa image)
5	Else (
	Maka response = client memanggil request dengan method GET dan URL ke service ubah.php dengan image)
6	END IF
7	Menampilkan url service ubah.php Mengambil result = response <i>JSON_decode</i> jadi array Mengembalikan fungsi result
8	End

#### 1. *Basis path testing*

##### a. *Flow Graph*



**Gambar 6.4 Algoritme *Method* ubah()**





### b. Cyclomatic Complexity (V(G))

$$1. V(G) = \text{Jumlah Region}$$

$$= 2$$

$$2. V(G) = \text{Jumlah Edge} - \text{Jumlah Node} + 2$$

$$= 8 - 8 + 2$$

$$= 2$$

$$3. V(G) = \text{Predicate Node} + 1$$

$$= 1 + 1$$

$$= 2$$

### c. Independent Path

1. Jalur 1 : 1-2-3-4-6-7-8

2. Jalur 2 : 1-2-3-5-6-7-8

### d. Desain Kasus Uji

Tabel 6.8 Kasus Uji Pengujian Unit *Method getAllProses()*

No. Jalur	Kasus Uji	Hasil Yang Diharapkan	Hasil yang didapatkan	Status
1	Ketika melakukan masukan <i>image</i> == null	Sistem berhasil memproses server dengan memanggil service ubah.php tanpa image	Sistem berhasil memproses server dengan memanggil service ubah.php tanpa image	Valid
2	Ketika melakukan masukan <i>image</i> tidak null	Sistem berhasil memproses server dengan memanggil service ubah.php dengan image	Sistem berhasil memproses server dengan memanggil service ubah.php dengan image	Valid

## 6.2 Pengujian Validasi

Pengujian validasi mendeskripsikan pengujian pada perangkat lunak menggunakan metode *blackbox testing*. Dalam hal ini, pengujian validasi digunakan untuk mengetahui apakah setiap fungsional pada sistem yang dibangun sudah sesuai dengan analisis kebutuhan pada *use case scenario* yang telah didefinisikan sebelumnya.





## 6.2.1 Pengujian Validasi Sistem Berbasis *Mobile*

### 6.2.1.1 Pengujian Validasi *Login*

Pengujian validasi *login* digunakan untuk mengetahui apakah kebutuhan fungsionalitas *login* dapat berjalan dengan baik, sehingga pengguna dapat masuk dalam sistem pada aplikasi GoTindak.

#### 1. Kasus uji coba ketika akun *email* terdaftar

Tabel 6.9 Kasus Uji Pengujian Validasi Akun *Email* Terdaftar

<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> menggunakan <i>email</i> terdaftar
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa <i>user</i> dapat masuk dalam sistem aplikasi ketika <i>login</i> menggunakan <i>email</i> yang sudah terdaftar.
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman <i>login</i> pada sistem aplikasi GoTindak</li> <li>2. Pengguna menambahkan <i>email</i> yang telah terdaftar pada <i>field</i> yang tersedia</li> <li>3. Menekan tombol <i>login</i></li> </ol>
<b>Hasil yang diharapkan</b>	Aktor dapat masuk dalam halaman beranda sistem aplikasi GoTindak dan menampilkan " <i>LoginSukses</i> "

#### 2. Kasus uji coba ketika akun *email* belum terdaftar

Tabel 6.10 Kasus Uji Pengujian Validasi Akun Belum *Email* Terdaftar

<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> menggunakan <i>email</i> belum terdaftar
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa <i>user</i> tidak dapat masuk dalam sistem aplikasi ketika akan <i>login</i> menggunakan <i>email</i> yang belum terdaftar.
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman <i>login</i> pada sistem aplikasi GoTindak</li> <li>2. Pengguna menambahkan <i>email</i> yang belum terdaftar pada <i>field</i> yang tersedia</li> <li>3. Menekan tombol <i>login</i></li> </ol>
<b>Hasil yang diharapkan</b>	Aktor tidak dapat masuk dalam halaman beranda sistem aplikasi GoTindak dan menampilkan " <i>Invalid Email Or Password</i> "





### 3. Kasus uji coba ketika *email* atau *password* tidak terisi

Tabel 6.11 Kasus Uji Pengujian Validasi *Email* atau *Password* Tidak Terisi

<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> ketika salah satu <i>field email</i> atau <i>password</i> tidak terisi
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa <i>user</i> tidak dapat masuk dalam sistem aplikasi ketika salah satu <i>field email</i> atau <i>password</i> belum terisi
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman <i>login</i> pada sistem aplikasi GoTindak</li> <li>2. Pengguna tidak mengisi salah satu <i>field email</i> atau <i>password</i> yang tersedia</li> <li>3. Menekan tombol <i>login</i></li> </ol>
<b>Hasil yang diharapkan</b>	Aktor tidak dapat masuk dalam sistem aplikasi GoTindak dan menampilkan pesan “isikan <i>email</i> anda” atau “isikan <i>password</i> anda” dan Pengguna tidak dapat masuk dalam sistem

#### 6.2.1.2 Pengujian Validasi Register

Pengujian validasi register digunakan untuk mengetahui apakah kebutuhan fungsionalitas register dapat berjalan dengan baik, sehingga pengguna dapat terdaftar dalam aplikasi GoTindak.

##### 1. Kasus uji coba ketika register akun berhasil

Tabel 6.12 Kasus Uji Pengujian Validasi Register Akun Berhasil

<b>Nama Kasus Uji</b>	Kasus uji berhasil melakukan register
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melakukan pendaftaran akun pada sistem.
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman register pada sistem aplikasi GoTindak</li> <li>2. Pengguna mengisi semua <i>field</i> pada <i>form</i> register yang telah disediakan</li> <li>3. Menekan tombol register</li> </ol>
<b>Hasil yang diharapkan</b>	Aktor dapat melakukan pendaftaran akun pada sistem aplikasi GoTindak dan berhasil masuk pada beranda sistem.





## 2. Kasus uji coba ketika *form* register tidak terisi lengkap

Tabel 6.13 Kasus Uji Pengujian Validasi *Form* Register Tidak Terisi Lengkap

<b>Nama Kasus Uji</b>	Kasus uji <i>form</i> register tidak terisi lengkap
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna tidak dapat melakukan pendaftaran akun pada sistem ketika <i>form</i> register tidak terisi dengan lengkap
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman register pada sistem aplikasi GoTindak</li> <li>2. Pengguna tidak mengisi salah satu <i>field</i> pada <i>form</i> register yang telah disediakan</li> <li>3. Menekan tombol register</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan pada salah satu <i>field</i> bahwa harus diisi lengkap dan pengguna tidak dapat masuk dalam sistem

### 6.2.1.3 Membuat laporan parkir

Pengujian validasi membuat laporan digunakan untuk mengetahui apakah kebutuhan fungsionalitas membuat laporan dapat berjalan dengan baik, sehingga pengguna dapat membuat laporan parkir pada aplikasi GoTindak.

Tabel 6.14 Kasus Uji Pengujian Validasi Membuat Laporan Parkir

<b>Nama Kasus Uji</b>	Membuat Laporan
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat membuat laporan parkir pada aplikasi GoTindak
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman tambah pada sistem aplikasi GoTindak</li> <li>2. Pengguna mengisi semua <i>field</i> pada <i>form</i> tambah laporan yang meliputi foto, deskripsi, lokasi dan kategori yang telah disediakan</li> <li>3. Menekan tombol Post</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menambahkan data laporan dan tersimpan pada <i>database</i> serta dapat menampilkan laporan yang telah ditambahkan pada halaman beranda. Aktor berhasil menambahkan laporan dan kembali pada halaman laporan





#### 6.2.1.4 Melihat Daftar Laporan

Pengujian validasi melihat laporan digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat laporan dapat berjalan dengan baik, sehingga pengguna dapat melihat daftar laporan parkir pada aplikasi GoTindak.

**Tabel 6.15 Kasus Uji Pengujian Validasi Melihat Laporan**

<b>Nama Kasus Uji</b>	Melihat Laporan Parkir
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat daftar laporan parkir
<b>Prosedur Pengujian</b>	1. Pengguna masuk dalam halaman beranda pada sistem aplikasi GoTindak
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan halaman beranda yang berisi daftar laporan parkir pengguna

#### 6.2.1.5 Melihat Detail Laporan

Pengujian validasi melihat detail laporan digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat detail laporan dapat berjalan dengan baik, sehingga pengguna dapat melihat detail laporan parkir pada aplikasi GoTindak.

**Tabel 6.16 Kasus Uji Pengujian Validasi Melihat Detail Laporan**

<b>Nama Kasus Uji</b>	Melihat Detail Laporan Parkir
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat detail daftar laporan parkir
<b>Prosedur Pengujian</b>	1. Pengguna masuk dalam halaman beranda pada sistem aplikasi GoTindak 2. Pengguna memilih salah satu laporan parkir
<b>Hasil yang diharapkan</b>	Sistem menampilkan data halaman detail yang berisi daftar laporan parkir pengguna. Aktor berhasil melihat halaman detail laporan

#### 6.2.1.6 Memberi Dukungan Laporan

Pengujian validasi memberi dukungan laporan digunakan untuk mengetahui apakah kebutuhan fungsionalitas memberi dukungan laporan dapat berjalan dengan baik, sehingga pengguna dapat memberikan dukungan laporan parkir pada aplikasi GoTindak.





**Tabel 6.17 Kasus Uji Pengujian Validasi Memberi Dukungan Laporan**

<b>Nama Kasus Uji</b>	Memberi dukungan laporan parkir
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat memberi dukungan pada laporan parkir
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman beranda pada sistem aplikasi GoTindak</li> <li>2. Pengguna memilih salah satu laporan parkir dan menekan <i>icon</i> dukungan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem data menambahkan jumlah dukungan pada laporan yang telah mendapatkan <i>like</i>

#### 6.2.1.7 Melihat Riwayat Laporan

Pengujian validasi melihat riwayat laporan digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat riwayat laporan dapat berjalan dengan baik, sehingga pengguna dapat melihat riwayat laporan parkir pada aplikasi GoTindak.

**Tabel 6.18 Kasus Uji Pengujian Validasi Melihat Riwayat Laporan**

<b>Nama Kasus Uji</b>	Melihat riwayat laporan
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat riwayat laporan yang telah dikirimkan sebelumnya
<b>Prosedur Pengujian</b>	1. Pengguna masuk dalam halaman riwayat pada sistem aplikasi GoTindak
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan data laporan pengguna atau riwayat laporan pengguna yang sudah dikirimkan sebelumnya

#### 6.2.1.8 Melihat Profil

Pengujian validasi melihat profil digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat profil dapat berjalan dengan baik, sehingga pengguna dapat melihat profil pada aplikasi GoTindak.





Tabel 6.19 Kasus Uji Pengujian Validasi Melihat Profil

<b>Nama Kasus Uji</b>	Melihat profil laporan
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat profil
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu profil pada tab menu profil
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan informasi profil pengguna dan aktor dapat melihat profil

#### 6.2.1.9 Mengedit Profil

Pengujian validasi mengedit profil digunakan untuk mengetahui apakah kebutuhan fungsionalitas mengedit profil dapat berjalan dengan baik, sehingga pengguna dapat mengedit profil pada aplikasi GoTindak.

Tabel 6.20 Kasus Uji Pengujian Validasi Mengedit Profil

<b>Nama Kasus Uji</b>	Mengedit Profil
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melakukan edit profil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu profil pada tab menu profil</li> <li>2. Pengguna mengubah nama lengkap, nomor hp, <i>username</i>, serta <i>email</i></li> <li>3. Pengguna menekan tombol simpan profil</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil melakukan proses edit profil pengguna dan dapat tersimpan dalam <i>database</i> . Selain itu sistem menampilkan informasi profil pengguna dan aktor dapat melihat profil yang telah diperbarui

#### 6.2.1.10 Mengupload Foto Profil

Pengujian validasi upload foto profil digunakan untuk mengetahui apakah kebutuhan fungsionalitas upload foto profil dapat berjalan dengan baik, sehingga pengguna dapat melakukan upload foto profil pada aplikasi GoTindak.

Tabel 6.21 Kasus Uji Pengujian Validasi Mengupload Foto Profil

<b>Nama Kasus Uji</b>	Mengupload Foto Profil
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melakukan edit profil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu profil pada tab menu profil</li> <li>2. Pengguna menekan <i>icon photo</i> profil</li> <li>3. Pengguna upload photo profil</li> </ol>





<b>Hasil yang diharapkan</b>	Sistem berhasil melakukan proses upload photo profil pengguna dan dapat tersimpan dalam <i>database</i> .
------------------------------	---

#### 6.2.1.11 Mengedit *Password*

Pengujian validasi mengedit *password* digunakan untuk mengetahui apakah kebutuhan fungsionalitas mengedit *password* dapat berjalan dengan baik, sehingga pengguna dapat mengedit *password* pada aplikasi GoTindak.

**Tabel 6.22 Kasus Uji Pengujian Validasi Mengedit *Password***

<b>Nama Kasus Uji</b>	Mengedit <i>Password</i>
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melakukan <i>edit password</i>
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu profil pada tab menu profil</li> <li>2. Pengguna mengubah <i>password</i> pada <i>form</i> ubah <i>password</i></li> <li>3. Pengguna menekan unah <i>password</i> dan mengeklik tombol simpan profil</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil melakukan proses ubah <i>password</i> pengguna dan tersimpan dalam <i>database</i> . Aktor dapat masuk dalam sistem dengan menggunakan <i>password</i> yang telah diperbarui.

#### 6.2.1.12 Menambah Komentar

Pengujian validasi menambah komentar digunakan untuk mengetahui apakah kebutuhan fungsionalitas menambah komentar dapat berjalan dengan baik, sehingga pengguna dapat menambahkan komentar pada aplikasi GoTindak.

**Tabel 6.23 Kasus Uji Pengujian Validasi Menambah Komentar**

<b>Nama Kasus Uji</b>	Menambah komentar
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat menambahkan komentar
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman beranda pada sistem aplikasi GoTindak</li> <li>2. Pengguna memilih salah satu laporan parkir</li> <li>3. Pengguna masuk pada halaman detail laporan parkir</li> <li>4. Pengguna menambahkan komentar pada <i>field</i> komentar</li> <li>5. Pengguna menekan tombol kirim komentar</li> </ol>





<b>Hasil yang diharapkan</b>	Sistem berhasil menambahkan komentar pengguna dan tersimpan dalam <i>database</i> . Aktor berhasil menambahkan komentar pada laporan.
------------------------------	---

#### 6.2.1.13 Melihat Komentar

Pengujian validasi melihat komentar digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat komentar dapat berjalan dengan baik, sehingga pengguna dapat melihat komentar pada aplikasi GoTindak.

**Tabel 6.24 Kasus Uji Pengujian Validasi Melihat Komentar**

<b>Nama Kasus Uji</b>	Melihat Komentar
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat komentar
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman beranda pada sistem aplikasi GoTindak</li> <li>2. Pengguna memilih salah satu laporan parkir</li> <li>3. Pengguna masuk pada halaman detail laporan parkir</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan komentar pengguna pada detail laporan dan aktor dapat melihat komentar pengguna

#### 6.2.1.14 Logout

Pengujian validasi *logout* digunakan untuk mengetahui apakah kebutuhan fungsionalitas *logout* dapat berjalan dengan baik, sehingga pengguna dapat *logout* pada aplikasi GoTindak.

**Tabel 6.25 Kasus Uji Pengujian Validasi Logout**

<b>Nama Kasus Uji</b>	<i>Logout</i>
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat <i>logout</i> dari sistem
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman profil pengguna</li> <li>2. Pengguna menekan <i>icon logout</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan halaman awal pada aplikasi dan pengguna berhasil <i>logout</i> dari sistem.

#### 6.2.1.15 Memilih Fitur *Camera*

Pengujian validasi memilih fitur *camera* digunakan untuk mengetahui apakah kebutuhan fungsionalitas fitur *camera* dapat berjalan dengan baik, sehingga pengguna dapat memilih fitur *camera* pada aplikasi GoTindak.



Tabel 6.26 Kasus Uji Pengujian Memilih Fitur *Camera*

<b>Nama Kasus Uji</b>	Memilih Fitur <i>Camera</i>
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat memilih fitur <i>camera</i> ketika ingin menambahkan photo laporan
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman tambah pada sistem aplikasi GoTindak</li> <li>2. Pengguna menekan <i>field</i> tambah foto</li> <li>3. Pengguna memilih fitur <i>camera</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan fitur <i>camera</i> dan aktor dapat melakukan pengambilan photo menggunakan <i>camera</i> smartphone

#### 6.2.1.16 Memilih Fitur *Gallery*

Pengujian validasi memilih fitur *gallery* digunakan untuk mengetahui apakah kebutuhan fungsionalitas fitur *gallery* dapat berjalan dengan baik, sehingga pengguna dapat memilih fitur *gallery* pada aplikasi GoTindak.

Tabel 6.27 Kasus Uji Pengujian Memilih Fitur *Gallery*

<b>Nama Kasus Uji</b>	Memilih Fitur <i>Gallery</i>
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat memilih fitur <i>gallery</i> ketika ingin menambahkan photo laporan
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman tambah pada sistem aplikasi GoTindak</li> <li>2. Pengguna menekan <i>field</i> tambah foto</li> <li>3. Pengguna memilih fitur <i>gallery</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan fitur <i>gallery</i> dan aktor dapat melakukan pengambilan photo menggunakan <i>gallery</i> smartphone

#### 6.2.1.17 Melihat Artikel Identifikasi Parkir

Pengujian validasi melihat artikel identifikasi parkir digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat artikel identifikasi parkir dapat berjalan dengan baik, sehingga pengguna dapat melihat artikel identifikasi parkir pada aplikasi GoTindak.

Tabel 6.28 Kasus Uji Pengujian Validasi Melihat Artikel Identifikasi Parkir

<b>Nama Kasus Uji</b>	Melihat artikel identifikasi parkir
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat artikel identifikasi parkir





<b>Prosedur Pengujian</b>	1. Pengguna memilih menu tab artikel pada sistem aplikasi
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan halaman artikel identifikasi parkir dan aktor dapat melihat artikel identifikasi parkir

### 6.2.1.18 Melihat Detail Artikel Identifikasi

Pengujian validasi melihat detail artikel identifikasi parkir digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat detail artikel identifikasi parkir dapat berjalan dengan baik, sehingga pengguna dapat melihat detail artikel identifikasi parkir pada aplikasi GoTindak.

Tabel 6.29 Kasus Uji Pengujian Validasi Melihat Detail Artikel Identifikasi Parkir

<b>Nama Kasus Uji</b>	Melihat artikel identifikasi parkir
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat melihat artikel identifikasi parkir
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu tab artikel pada sistem aplikasi 2. Pengguna memilih salah satu artikel
<b>Hasil yang diharapkan</b>	Sistem berhasil menampilkan halaman detail artikel identifikasi parkir dan aktor dapat melihat detail artikel identifikasi parkir

## 6.2.2 Pengujian Validasi Sistem Berbasis Website

### 6.2.2.1 Pengujian Validasi Login Website

Pengujian validasi *login* digunakan untuk mengetahui apakah kebutuhan fungsionalitas *login* pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat masuk dalam sistem pada *website*.

#### 1. Kasus uji coba ketika akun *email* terdaftar

Tabel 6.30 Kasus Uji Pengujian Validasi Akun *Email* Terdaftar

<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> menggunakan <i>email</i> terdaftar
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa <i>user</i> dapat masuk dalam sistem aplikasi ketika <i>login</i> menggunakan <i>email</i> yang sudah terdaftar
<b>Prosedur Pengujian</b>	1. Pengguna masuk dalam halaman <i>login</i> pada sistem <i>website</i> 2. Pengguna menambahkan <i>email</i> yang telah terdaftar pada <i>field</i> yang tersedia





	3. Menekan tombol <i>login</i>
<b>Hasil yang diharapkan</b>	Aktor dapat masuk dalam halaman <i>dashboard</i> sistem <i>website</i>

## 2. Kasus uji coba ketika akun *email* belum terdaftar

Tabel 6.31 Kasus Uji Pengujian Validasi Akun Belum *Email* Terdaftar

<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> menggunakan <i>email</i> belum terdaftar
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa <i>user</i> tidak dapat masuk dalam sistem <i>website</i> ketika akan <i>login</i> menggunakan <i>email</i> yang belum terdaftar.
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman <i>login</i> pada sistem <i>website</i></li> <li>2. Pengguna memasukan <i>email</i> yang belum terdaftar pada <i>field</i> yang tersedia</li> <li>3. Menekan tombol <i>login</i></li> </ol>
<b>Hasil yang diharapkan</b>	Aktor tidak dapat masuk dalam halaman <i>dashboard</i> sistem <i>website</i> dan menampilkan " <i>Email Salah</i> "

## 3. Kasus uji coba ketika *email* atau *password* tidak terisi

Tabel 6.32 Kasus Uji Pengujian Validasi *Email* atau *Password* Tidak Terisi

<b>Nama Kasus Uji</b>	Kasus uji <i>login</i> ketika salah satu <i>field email</i> atau <i>password</i> tidak terisi
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa <i>user</i> tidak dapat masuk dalam sistem <i>website</i> ketika salah satu <i>field email</i> atau <i>password</i> belum terisi
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman <i>login</i> pada sistem <i>website</i></li> <li>2. Pengguna tidak mengisi salah satu <i>field email</i> atau <i>password</i> yang tersedia</li> <li>3. Menekan tombol <i>login</i></li> </ol>
<b>Hasil yang diharapkan</b>	Aktor tidak dapat masuk dalam <i>website</i> dan menampilkan pesan " <i>isikan email</i> anda" atau " <i>isikan password</i> anda" dan Admin tidak dapat masuk dalam sistem





### 6.2.2.2 Melihat Jumlah Laporan

Pengujian validasi melihat jumlah laporan digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat jumlah laporan pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat jumlah laporan dalam sistem *website*.

**Tabel 6.33 Kasus Uji Pengujian Validasi Melihat Jumlah Laporan**

<b>Nama Kasus Uji</b>	Melihat jumlah laporan
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat jumlah laporan pada sistem <i>website</i>
<b>Prosedur Pengujian</b>	1. Pengguna masuk dalam halaman <i>dashboard</i> pada sistem <i>website</i>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan jumlah laporan pada halaman <i>dashboard</i> dan aktor dapat melihat jumlah laporan

### 6.2.2.3 Melihat Laporan Belum Ditindaklanjuti

Pengujian validasi melihat laporan belum ditindaklanjuti digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat laporan belum ditindaklanjuti pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat masuk dalam sistem *website*.

**Tabel 6.34 Kasus Uji Pengujian Validasi Melihat Laporan Belum Ditindaklanjuti**

<b>Nama Kasus Uji</b>	Melihat laporan belum ditindaklanjuti
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat laporan belum ditindaklanjuti
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu belum ditindaklanjuti pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan laporan dengan status belum ditindaklanjuti dan aktor dapat melihat laporan belum ditindaklanjuti

### 6.2.2.4 Melihat Detail Laporan Belum Ditindaklanjuti

Pengujian validasi melihat laporan belum ditindaklanjuti digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat detail laporan belum ditindaklanjuti pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat detail laporan belum ditindaklanjuti dalam sistem *website*.





**Tabel 6.35 Kasus Uji Pengujian Validasi Melihat Detail Laporan Belum Ditindaklanjuti**

<b>Nama Kasus Uji</b>	Melihat detail laporan belum ditindaklanjuti
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat laporan belum ditindaklanjuti
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu belum ditindaklanjuti pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i></li> <li>2. Pengguna menekan tombol proses</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan detail laporan dengan status belum ditindaklanjuti dan aktor dapat melihat detail laporan belum ditindaklanjuti

#### 6.2.2.5 Mengubah Status Laporan Belum Ditindaklanjuti

Pengujian validasi mengubah status laporan belum ditindaklanjuti digunakan untuk mengetahui apakah kebutuhan fungsionalitas mengubah status laporan belum ditindaklanjuti pada sistem *website* dapat berjalan dengan baik, sehingga pengguna mengubah status laporan belum ditindaklanjuti dalam sistem *website*.

**Tabel 6.36 Kasus Uji Pengujian Validasi Mengubah Laporan Belum Ditindaklanjuti**

<b>Nama Kasus Uji</b>	Mengubah status laporan belum ditindaklanjuti
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mengubah status laporan belum ditindaklanjuti
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu belum ditindaklanjuti pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i></li> <li>2. Pengguna menekan tombol proses</li> <li>3. Pengguna memilih status pada dropdown yang disediakan oleh <i>form</i> detail laporan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat melakukan proses untuk mengubah status dan menyimpan data laporan pada <i>database</i> . Aktor dapat melihat status yang telah diubah

#### 6.2.2.6 Mencari Status Laporan Belum Ditindaklanjuti

Pengujian validasi mencari status laporan belum ditindaklanjuti digunakan untuk mengetahui apakah kebutuhan fungsionalitas mencari status laporan belum ditindaklanjuti pada sistem *website* dapat berjalan dengan baik, sehingga pengguna mencari status laporan belum ditindaklanjuti dalam sistem *website*.





**Tabel 6.37 Kasus Uji Pengujian Validasi Mencari Status Laporan Belum Ditindaklanjuti**

<b>Nama Kasus Uji</b>	Mencari status laporan belum ditindaklanjuti
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mencari status laporan belum ditindaklanjuti
<b>Prosedur Pengujian</b>	1. Pengguna memasukan kata kunci pada <i>field</i> pencarian yang telah disediakan pada halaman laporan belum ditindaklanjuti
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan hasil pencarian sesuai kata kunci yang dimasukkan oleh pengguna dan aktor dapat melihat hasil laporan sesuai dengan kata kunci pencarian

#### 6.2.2.7 Melihat Laporan Sedang Proses

Pengujian validasi melihat laporan sedang proses digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat laporan sedang proses pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat laporan sedang proses dalam sistem *website*.

**Tabel 6.38 Kasus Uji Pengujian Validasi Melihat Laporan Sedang Proses**

<b>Nama Kasus Uji</b>	Melihat laporan sedang proses
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat laporan sedang proses
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu sedang proses pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan laporan dengan status sedang proses dan aktor dapat melihat laporan sedang proses

#### 6.2.2.8 Melihat Detail Laporan Sedang Proses

Pengujian validasi melihat detail laporan sedang proses digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat detail laporan sedang proses pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat detail laporan sedang proses dalam sistem *website*.

**Tabel 6.39 Kasus Uji Pengujian Validasi Melihat Detail Laporan Sedang Proses**

<b>Nama Kasus Uji</b>	Melihat detail laporan sedang proses
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat laporan sedang proses





<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu sedang proses bagian <i>sidebar</i> menu navigasi dalam <i>website</i></li> <li>2. Pengguna menekan tombol proses</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan detail laporan dengan status sedang proses dan aktor dapat melihat detail laporan sedang proses

#### 6.2.2.9 Mengubah Status Laporan Sedang Proses

Pengujian validasi mengubah status laporan sedang proses digunakan untuk mengetahui apakah kebutuhan fungsionalitas mengubah status laporan sedang proses pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat mengubah status laporan sedang proses dalam sistem *website*.

**Tabel 6.40 Kasus Uji Pengujian Validasi Mengubah Status Laporan Sedang Proses**

<b>Nama Kasus Uji</b>	Mengubah status laporan sedang proses
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mengubah status laporan sedang proses
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu sedang proses pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i></li> <li>2. Pengguna menekan tombol proses</li> <li>3. Pengguna memilih status pada dropdown yang disediakan oleh <i>form</i> detail laporan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat melakukan proses untuk megubah status dan mneyimpan data laporan pada <i>database</i> . Aktor dapat melihat status yang telah diubah

#### 6.2.2.10 Mencari Status Laporan Sedang Proses

Pengujian validasi mencari status laporan sedang proses digunakan untuk mengetahui apakah kebutuhan fungsionalitas mencari status laporan sedang proses pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat mencari status laporan sedang diproses dalam sistem *website*.

**Tabel 6.41 Kasus Uji Pengujian Validasi Mencari Status Laporan Sedang Proses**

<b>Nama Kasus Uji</b>	Mencari status laporan sedang proses
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mencari status laporan sedang proses
<b>Prosedur Pengujian</b>	1. Pengguna menambahkan kata kunci pada <i>field</i> pencarian yang telah disediakan pada halaman laporan sedang proses





<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan hasil pencarian sesuai kata kunci yang dimasukkan oleh pengguna dan aktor dapat melihat hasil laporan sesuai dengan kata kunci pencarian
------------------------------	---

#### 6.2.2.11 Melihat Laporan Selesai

Pengujian validasi melihat laporan selesai digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat laporan selesai pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat laporan selesai dalam sistem *website*.

**Tabel 6.42 Kasus Uji Pengujian Validasi Melihat Laporan Selesai**

<b>Nama Kasus Uji</b>	Melihat laporan selesai
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat laporan selesai
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu selesai pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan laporan dengan status selesai dan aktor dapat melihat laporan selesai

#### 6.2.2.12 Melihat Detail Laporan Selesai

Pengujian validasi melihat detail laporan selesai digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat detail laporan selesai pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat detail laporan selesai dalam sistem *website*.

**Tabel 6.43 Kasus Uji Pengujian Validasi Melihat Detail Laporan Selesai**

<b>Nama Kasus Uji</b>	Melihat detail laporan selesai
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat detail laporan selesai
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu sedang proses bagian <i>sidebar</i> menu navigasi dalam <i>website</i> 2. Pengguna menekan tombol proses
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan detail laporan dengan status sedang proses dan aktor dapat melihat detail laporan sedang proses





### 6.2.2.13 Mencari Status Laporan Selesai

Pengujian validasi mencari status laporan selesai digunakan untuk mengetahui apakah kebutuhan fungsionalitas mencari status laporan selesai pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat mencari status laporan selesai dalam sistem *website*.

**Tabel 6.44 Kasus Uji Pengujian Validasi Mencari Status Laporan Selesai**

<b>Nama Kasus Uji</b>	Mencari status laporan selesai
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mencari status laporan sedang proses
<b>Prosedur Pengujian</b>	1. Pengguna memasukkan kata kunci pada <i>field</i> pencarian yang telah disediakan pada halaman laporan selesai
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan hasil pencarian sesuai kata kunci yang dimasukkan oleh pengguna dan aktor dapat melihat hasil laporan sesuai dengan kata kunci pencarian

### 6.2.2.14 Melihat Laporan Tidak Disetujui

Pengujian validasi melihat laporan tidak disetujui digunakan untuk mengetahui apakah kebutuhan fungsionalitas melihat laporan tidak disetujui pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat melihat laporan tidak disetujui dalam sistem *website*.

**Tabel 6.45 Kasus Uji Pengujian Validasi Melihat Laporan Tidak disetujui**

<b>Nama Kasus Uji</b>	Melihat laporan tidak disetujui
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat melihat laporan tidak disetujui
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu tidak disetujui pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan laporan dengan status tidak disetujui dan aktor dapat melihat laporan selesai

### 6.2.2.15 Mencari Laporan Tidak disetujui

Pengujian validasi mencari status laporan tidak disetujui digunakan untuk mengetahui apakah kebutuhan fungsionalitas mencari status laporan tidak disetujui pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat mencari status laporan tidak disetujui dalam sistem *website*.





Tabel 6.46 Kasus Uji Pengujian Validasi Mencari Laporan Tidak disetujui

<b>Nama Kasus Uji</b>	Mencari status laporan tidak disetujui
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mencari status laporan tidak disetujui
<b>Prosedur Pengujian</b>	1. Pengguna memasukkan kata kunci pada <i>field</i> pencarian yang telah disediakan pada halaman laporan selesai
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan hasil pencarian sesuai kata kunci yang dimasukkan oleh pengguna dan aktor dapat melihat hasil laporan sesuai dengan kata kunci pencarian

#### 6.2.2.16 Mengubah Status Laporan Tidak disetujui

Pengujian validasi mengubah status laporan sedang proses digunakan untuk mengetahui apakah kebutuhan fungsionalitas mengubah status laporan sedang proses pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat mengubah status laporan sedang proses dalam sistem *website*.

Tabel 6.47 Kasus Uji Pengujian Validasi Mengubah Status Laporan Tidak disetujui

<b>Nama Kasus Uji</b>	Mengubah status laporan tidak disetujui
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mengubah status laporan tidak disetujui
<b>Prosedur Pengujian</b>	1. Pengguna memilih menu sedang proses pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i> 2. Pengguna menekan tombol tidak disetujui 3. Pengguna memilih status pada dropdown yang disediakan oleh <i>form</i> detail laporan
<b>Hasil yang diharapkan</b>	Sistem dapat melakukan proses untuk mengubah status dan menyimpan data laporan pada <i>database</i> . Aktor dapat melihat status yang telah diubah

#### 6.2.2.17 Logout

Pengujian validasi *logout* digunakan untuk mengetahui apakah kebutuhan fungsionalitas *logout* dapat berjalan dengan baik, sehingga pengguna dapat *logout* pada sistem *website*.



Tabel 6.48 Kasus Uji Pengujian Validasi *Logout*

<b>Nama Kasus Uji</b>	<i>Logout</i>
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa pengguna dapat <i>logout</i> dari sistem
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna masuk dalam halaman profil admin</li> <li>2. Pengguna menekan <i>icon logout</i></li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan halaman awal pada aplikasi dan pengguna berhasil <i>logout</i> dari sistem.

### 6.2.2.18 Mengupload Foto Sedang Proses

Pengujian validasi mengupload foto sedang proses digunakan untuk mengetahui apakah kebutuhan fungsionalitas mengunggah foto sedang proses pada sistem *website* dapat berjalan dengan baik, sehingga pengguna dapat mengupload foto sedang proses dalam sistem *website*.

Tabel 6.49 Kasus Uji Pengujian Validasi Mengupload Foto Sedang Proses

<b>Nama Kasus Uji</b>	Mengupload Foto Sedang Proses
<b>Tujuan Pengujian</b>	Untuk membuktikan bahwa aktor dapat mengupload foto sedang proses
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu sedang proses pada bagian <i>sidebar</i> menu navigasi dalam <i>website</i></li> <li>2. Pengguna menekan tombol proses</li> <li>3. Pengguna menekan file upload gambar</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat melakukan proses upload gambar dan menyimpan file gambar pada <i>database</i> . Aktor berhasil melakukan proses upload gambar.

## 6.3 Pengujian Usabilitas

Pengujian usabilitas pada aplikasi GoTindak dilakukan dengan tujuan untuk mengetahui kepuasan dan tanggapan pengguna pada saat menjalankan aplikasi yang telah dikembangkan. Metode *System Usability Scale* (SUS) digunakan untuk pengujian usabilitas pada aplikasi GoTindak. Di samping itu, tujuan menerapkan metode *System Usability Scale* (SUS) dikarenakan sangat efektif dan handal ketika pengujian ini diterapkan dalam produk dan aplikasi.

### 6.3.1 Prosedur Pengujian

Metode Pengujian SUS memiliki sepuluh kriteria pernyataan, yang terdiri dari pernyataan positif ditandai dengan nomor ganjil sedangkan untuk pernyataan negatif dengan nomor genap. Alur dari pengujian ini diawali dengan memberi





penjelasan kepada responden mengenai aplikasi GoTindak, selanjutnya responden menjalankan aplikasi yang dikembangkan dan responden memberikan tanggapan berupa penilaian pada kuesioner yang diberikan. Pernyataan pada kuesioner mempunyai skor 1 sampai dengan 5. Selanjutnya melakukan normalisasi pada skor dengan aturan, jika skor pada pernyataan ganjil terdapat pengurangan nilai 1 sedangkan pada pernyataan genap terdapat pengurangan nilai 5. Sesudah itu untuk mendapatkan skor usabilitas dari aplikasi, dengan cara mengakumulasi skor setiap pernyataan yang telah dinormalisasi dengan mengalikan 2,5. Tabel 6.50 dibawah ini merupakan *grade ranking* dari skor usabilitas menggunakan metode pengujian SUS.

**Tabel 6.50 Parameter Bobot Skor Usabilitas (SUS)**

Skor	Kualifikasi	Hasil
>80	Sangat Baik (SB)	Berhasil
68-80	Baik (B)	Berhasil
68	Cukup (C)	Berhasil
51-68	Buruk (D)	Tidak Berhasil
<51	Kurang (K)	Tidak Berhasil

Berkaitan dengan itu, pada pengujian SUS ini akan dilakukan skenario bagaimana cara penggunaan aplikasi, karena pada aplikasi yang dikembangkan terdapat dua *platform* dengan dua kategori. Maka dengan melakukan pengujian pada *platform mobile* dengan aktor pelapor dan *platform website* pada aktor admin, sehingga terdapat 2 kategori pengujian. Setelah selesai menjalankan skenario penggunaan pada aplikasi GoTindak, selanjutnya responden uji diminta untuk mengisi kuesioner yang memuat 10 pernyataan. Tabel 6.51 menunjukkan daftar pernyataan kuesioner SUS pada responden uji pelapor dan pada tabel 6.51 menunjukkan daftar pernyataan pada responden uji admin.

**Tabel 6.51 Daftar Pernyataan Kuesioner Pengujian Usability Pelapor**

No	Pernyataan
1	Saya akan sering menggunakan aplikasi <i>mobile</i> identifikasi parkir non resmi atau aplikasi "GoTindak" ini untuk melaporkan adanya parkir non resmi sebagai alternatif media pelaporan
2	Saya merasa aplikasi "GoTindak" ini memiliki alur yang rumit saat melakukan aktivitas pelaporan
3	Saya merasa dapat menjalankan aplikasi "GoTindak" ini dengan mudah



Tabel 6.51 Daftar Pernyataan Kuesioner Pengujian *Usability* Pelapor (Lanjutan)

No	Pernyataan
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktivitas aplikasi "GoTindak" ini
5	Saya merasa aplikasi "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik
6	Saya menilai bahwa sistem aplikasi "GoTindak" ini banyak hal yang tidak konsisten
7	Saya menilai aplikasi "GoTindak" ini akan sangat mudah digunakan oleh banyak orang
8	Saya menilai aplikasi "GoTindak" ini sangat rumit ketika digunakan
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi "GoTindak" ini dengan baik
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi "GoTindak" ini

Tabel 6.52 Daftar Pernyataan Kuesioner Pengujian *Usability* Admin

No	Pernyataan
1	Saya akan sering menggunakan aplikasi <i>website</i> identifikasi parkir non resmi atau aplikasi "GoTindak" ini
2	Saya merasa aplikasi <i>website</i> "GoTindak" ini memiliki alur yang rumit saat melakukan aktivitas pelaporan
3	Saya merasa dapat menjalankan aplikasi <i>website</i> "GoTindak" ini dengan mudah
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktivitas aplikasi <i>website</i> "GoTindak" ini
5	Saya merasa aplikasi <i>website</i> "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik
6	Saya menilai bahwa sistem aplikasi <i>website</i> "GoTindak" ini banyak hal yang tidak konsisten
7	Saya menilai aplikasi <i>website</i> "GoTindak" ini akan sangat mudah digunakan oleh banyak orang



Tabel 6.52 Daftar Pernyataan Kuesioner Pengujian *Usability Admin* (Lanjutan)

No	Pernyataan
8	Saya menilai aplikasi <i>website</i> "GoTindak" ini sangat rumit ketika digunakan
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi <i>website</i> "Gotindak" ini dengan baik
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi <i>website</i> "GoTindak" ini

### 6.3.2 Hasil Pengujian

Setelah dilakukan pengujian menggunakan metode SUS kepada aktor dari sisi pelapor ataupun dari sisi admin, langkah selanjutnya yaitu melakukan perhitungan nilai skor pada data sesuai dengan prosedur pada metode SUS. Tujuan dari perhitungan tersebut untuk mengukur tingkat kepuasan pengguna serta kemudahan saat menjalankan aktivitas pada aplikasi yang dikembangkan. Pada tabel 6.53 akan dijelaskan hasil dari perhitungan skor pengujian usability pada sisi aktor pelapor dan tabel 6.54 akan dijelaskan hasil dari perhitungan skor pengujian usability pada sisi admin. Setelah mendapatkan data pengujian usability, selanjutnya data tersebut akan dinormalisasi menggunakan prosedur perhitungan SUS.

Tabel 6.53 Data Hasil Pengujian Usabilitas Aplikasi *Mobile Pelapor*

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
L1	3	1	4	2	4	2	4	2	4	2
L2	4	2	5	1	4	1	5	2	5	1
L3	4	2	4	1	5	1	4	2	4	2
L4	3	2	4	2	5	1	4	2	4	1
L5	4	2	4	1	4	1	5	1	4	1

Tabel 6.54 Data Hasil Pengujian Usabilitas *Website Administrator*

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
A1	4	2	4	2	4	3	4	2	4	2
A2	4	2	3	2	3	2	4	2	4	2

Pada Tabel 6.55 di bawah ini merupakan hasil rekapitulasi perhitungan kuesioner SUS pada responden uji pelapor. Untuk mendapatkan skor usability dari aplikasi, dengan cara mengakumulasi skor setiap pernyataan yang telah





dinormalisasi dengan mengalikan 2,5. Setelah itu, hasil total SUS yang telah didapatkan, dibagi dengan jumlah responden uji untuk menghasilkan nilai rata-rata skor SUS sebesar 80,5.

**Tabel 6.55 Data Hasil Normalisasi dan Skor SUS Aplikasi *Mobile* Pelapor**

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Jumlah	X2,5
L1	2	4	3	2	3	3	2	3	2	3	27	67,5
L2	3	3	4	4	3	4	1	3	4	4	33	82,5
L3	3	3	3	5	4	4	3	3	3	3	34	85
L4	2	3	3	3	4	4	3	3	3	4	32	80
L5	3	3	3	4	3	4	4	4	3	4	35	87,5
SKOR SUS												80,5

Pada Tabel 6.56 di bawah ini merupakan hasil rekapitulasi perhitungan kuesioner SUS pada responden uji admin. Untuk mendapatkan skor usability dari sistem *website*, dengan cara mengakumulasikan skor setiap pernyataan yang telah dinormalisasi dengan mengalikan 2,5. Setelah itu, hasil total SUS yang telah didapatkan dibagi dengan jumlah responden uji untuk menghasilkan nilai rata-rata skor SUS sebesar 71,25.

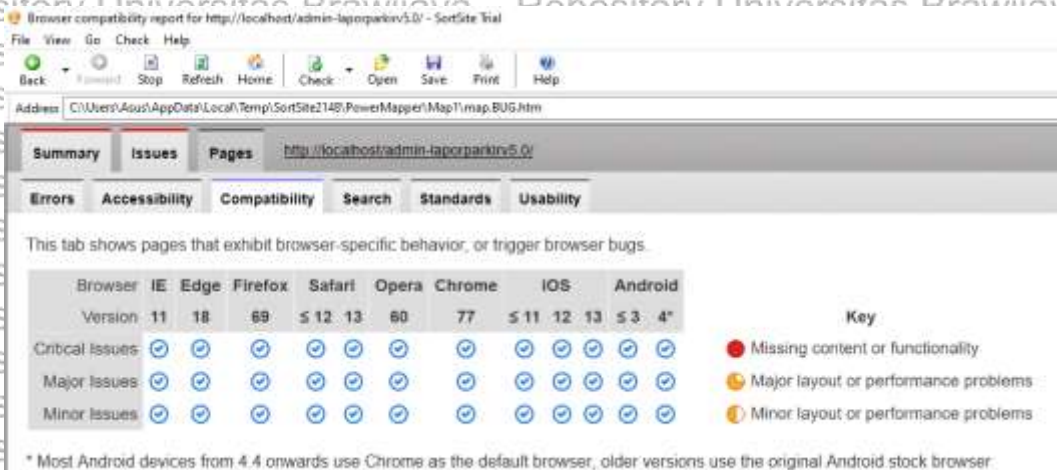
**Tabel 6.56 Data Hasil Normalisasi dan Skor SUS Website Administrator**

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Jumlah	X2,5
A1	3	3	3	3	3	2	3	3	3	3	29	72,5
A2	3	3	2	3	2	3	3	3	3	3	28	70
SKOR SUS												71,25

## 6.4 Pengujian *Compatibility*

Pengujian *compatibility* adalah teknik pengujian non fungsional yang dilakukan untuk menguji kompatibilitas pada sistem, apakah dapat digunakan pada *browser* dengan lingkungan berbeda-beda. Alat bantu untuk melakukan *compatibility testing* yaitu versi 5.31. *Tools* ini berfungsi untuk mengungkap kesalahan atau masalah saat mengeksekusi yang dapat diketahui melalui perbedaan-perbedaan konfigurasi. Berikut adalah hasil pengujian *compatibility* menggunakan *tools* seperti gambar 6.5.





**Gambar 6.5 Hasil Pengujian Compatibilty dengan Tool SortSite**

*Expected result* dari pengujian *compatibility* yaitu sistem harus dapat dijalankan pada lingkungan web browser yang berbeda-beda seperti Internet Explorer 11, Edge 18, Firefox 66, Safari 12, Opera 60, Chrome 74. Gambar yang telah dilampirkan di atas, dapat diketahui bahwa sistem dapat berjalan dengan baik pada 6 jenis web browser yang telah ditentukan tanpa terdapat sedikitpun kesalahan mayor maupun minor.

## 6.5 Analisis Hasil Pengujian

Tahapan selanjutnya yaitu menganalisis hasil pengujian yang dilakukan sebelumnya untuk memperoleh kesimpulan hasil uji. Pengujian yang dilakukan pada penelitian ini meliputi pengujian unit, pengujian validasi, pengujian usabilitas, dan pengujian *compatibility*.

### 6.5.1 Analisis Hasil Pengujian Unit

Pengujian unit dilakukan untuk mengetahui unit atau komponen pada sistem dapat berjalan dengan benar berdasarkan fungsionalitas pada masing-masing fitur. Pengujian unit dilakukan pada empat *sample method* yaitu 2 *sample method* pada aplikasi *mobile* dan 2 *sample method* pada sistem *website*. 2 *sample method* aplikasi *mobile* yaitu *method post* pada kelas *TambahLaporanActivity* dan *method onCreateView* pada kelas *BerandaFragment*. Pada pengujian *method post* terdapat 3 (tiga) kasus uji dengan hasil status valid, sedangkan pada *method onCreateView* terdapat 1 (satu) kasus uji dengan hasil uji status valid. Selanjutnya, untuk 2 *sample method* pada *website* yaitu *method getAllProses* pada kelas *m\_sedangproses*, dan *method ubah* pada kelas *m\_sedangproses*. Untuk pengujian *method getAllProses* terdapat 1 (satu) kasus uji dengan hasil status valid, sedangkan pada *method ubah* terdapat 2 (dua) kasus uji dengan hasil status valid.

### 6.5.2 Analisis Hasil Pengujian Validasi

Pegujian validasi dilakukan pada perangkat lunak menggunakan metode *blackbox testing*. Pengujian validasi digunakan untuk mengetahui apakah setiap fungsional pada sistem yang dibangun sudah sesuai dengan analisis kebutuhan





pada *use case scenario* yang telah didefinisikan sebelumnya. semua *scenario* yang terdefinisi pada kebutuhan fungsionalitas sistem, dilakukan pengujian validasi pada masing-masing *scenario* dengan total keseluruhan 36 kasus uji validasi dengan 18 kasus uji pada sistem *mobile* dan 18 kasus uji pada sistem *website*. Hasil dari 3 kasus uji tersebut menghasilkan kasus uji dengan status valid. Status valid di sini diartikan bahwa sistem dapat berjalan dengan baik sesuai dengan kebutuhan.

### 6.5.3 Analisis Hasil Pegujian Usabilitas

Pengujian usabilitas pada aplikasi GoTindak dilakukan dengan tujuan untuk mengetahui kepuasan dan tanggapan pengguna pada saat menjalankan aplikasi. Metode System *Usability Scale* (SUS) digunakan untuk pengujian usabilitas pada aplikasi GoTindak ini. Skor rata-rata yang dihasilkan pada pengujian usabilitas pada pengguna aplikasi *mobile* yaitu dengan nilai 80,5 dari 5 responden. Selain itu, untuk pengujian *usability* sistem *website* menghasilkan rata-rata 71,75. Nilai rata-rata yang dihasilkan pada menunjukkan kualifikasi baik untuk aplikasi GoTindak aktor pelapor dana sistem *website* GoTindak aktor administrator. Dalam hal ini, dapat dikatakan bahwa aplikasi GoTindak dapat digunakan dan sesuai dengan kebutuhan pengguna.

### 6.5.4 Analisis Hasil Pengujian *Compatibility*

Pengujian *compatibility* adalah dapat dijalankan pada lingkungan *web browser* yang berbeda-beda seperti Internet Explorer 11, Edge 18, Firefox 66, Safari 12, Opera 60, Chrome 74, iOS  $\leq 10.11.12$ , dan *Android*  $\leq 3.4$ . Berdasarkan gambar yang telah dilampirkan di atas, dapat diketahui bahwa sistem dapat berjalan dengan baik pada 11 jenis *web browser*.





## BAB 7 PENUTUP

### 7.1 Kesimpulan

Setelah menyelesaikan seluruh tahapan penelitian mulai dari proses analisis kebutuhan, perancangan, implementasi, serta pengujian. Maka akan didapatkan kesimpulan di antaranya:

1. Pembangunan aplikasi GoTindak sebagai aplikasi informasi laporan untuk mengidentifikasi parkir non resmi terdiri dari 3 kategori pemangku kepentingan yaitu pengguna, pelapor dan administrator. Proses iterasi pertama pada pengguna menghasilkan 2 kebutuhan fungsional *login* dan *register*. Untuk kategori pelapor menghasilkan 16 kebutuhan fungsional pada iterasi pertama yang terdiri dari *register*, *login*, membuat laporan, melihat detail laporan, memberi dukungan laporan, melihat riwayat laporan, melihat profil, mengedit profil, mengupload profil, mengedit *password*, menambah komentar, melihat komentar, *logout*, memilih fitur *camera*, dan memilih fitur *gallery* dan untuk iterasi kedua terdapat penambahan 2 kebutuhan fungsional yaitu melihat artikel identifikasi parkir dan melihat detail artikel identifikasi parkir. Sedangkan pada kategori administrator pada sistem *website* menghasilkan 17 kebutuhan fungsional pada iterasi pertama yang terdiri dari *login*, melihat jumlah laporan, melihat laporan belum ditindaklanjuti, melihat detail laporan belum ditindaklanjuti, mengubah status laporan belum ditindaklanjuti, mencari status laporan belum ditindaklanjuti, melihat laporan sedang proses, melihat detail laporan sedang proses, mengubah status laporan sedang proses, mencari status laporan sedang proses, melihat laporan selesai, melihat detail laporan selesai, melihat laporan tidak disetujui, mencari laporan tidak disetujui, mengubah status laporan tidak disetujui, dan *logout* dan untuk iterasi kedua terdapat 1 penambahan kebutuhan fungsional yaitu mengupload foto sedang proses. Total keseluruhan kebutuhan fungsional yang diperoleh yaitu 36 dengan 2 kali proses iterasi pada aplikasi *mobile* GoTindak dan 2 kali iterasi pada *website* admin GoTindak. Kebutuhan fungsional tersebut didapatkan dari proses wawancara *expert* pada pihak DISHUB Kota Malang bidang perparkiran, dan menyebarkan kuesioner terkait identifikasi parkir non resmi pada masyarakat umum.
2. Pembangunan aplikasi GoTindak sebagai aplikasi informasi laporan untuk mengidentifikasi parkir non resmi dalam melakukan pembangunan pada aplikasi perangkat lunak menggunakan metode *prototyping*. Dalam melakukan pembangunan sistem dihasilkan perancangan sistem sesuai dengan analisis kebutuhan pada iterasi 1 dan iterasi 2 di antaranya perancangan *sequence diagram*, *class diagram*, perancangan basis data, perancangan komponen, dan perancangan antarmuka. Untuk implementasi pada aplikasi GoTindak didasarkan pada perancangan yang telah dilakukan sebelumnya berupa perancangan komponen dan perancangan antarmuka. Untuk mengimplementasikan kode program pada aplikasi *mobile* GoTindak





menggunakan bahasa *java* untuk berbasis android, *framework codeigniter* untuk implementasi bagian *website* admin, juga menggunakan *web service* berfungsi untuk mengolah data pada *website* dan *mobile*. Pada saat implementasi basis data yang berisi *database* menggunakan MySQL sebagai layanan untuk menyimpan data.

3. Pembangunan aplikasi GoTindak sebagai aplikasi informasi laporan untuk mengidentifikasi parkir non resmi telah melakukan 2 pengujian fungsional yaitu pengujian unit, pengujian validasi, serta 2 pengujian non fungsional yaitu pengujian usabilitas dan pengujian *compatibility*. Untuk pengujian unit dilakukan pada empat *sample method* dengan hasil status valid. Selain itu pada pengujian validasi terdapat 18 kasus uji pelapor dan 18 kasus uji pada administrator dari total keseluruhan mendapatkan status valid.
4. Pembangunan aplikasi GoTindak sebagai aplikasi informasi laporan untuk mengidentifikasi parkir non resmi telah melakukan pengujian usabilitas menggunakan metode SUS menggunakan kuesioner menghasilkan rata-rata nilai 80,5 dan untuk pengujian *usability* sistem *website* menghasilkan rata-rata 71,75. Nilai 80,5 dapat dikualifikasikan sangat baik dan dikatakan berhasil sedangkan pada *website* mendapatkan skor 71,75 dikatakan baik. Aplikasi GoTindak dapat digunakan dan telah sesuai kebutuhan karena mendapatkan skor yang baik untuk pelapor maupun administrator.

## 7.2 Saran

Untuk proses pengembangan lebih lanjut pada aplikasi “GoTindak” maka diberikan saran sebagai penelitian selanjutnya yaitu:

1. Untuk pengembangan lebih lanjut dapat ditambahkan fitur untuk melihat titik lokasi parkir resmi.
2. Aplikasi *mobile* disarankan menambahkan fungsi untuk mengubah laporan pengguna.
3. Aplikasi *mobile* disarankan menambahkan fitur filter untuk mengetahui status parkir yang telah diproses oleh DISHUB.
4. Aplikasi *mobile* disarankan menambahkan fitur notifikasi untuk pelapor agar mengetahui laporan status laporan apakah telah ditindaklanjuti atau belum secara *real time*.
5. Aplikasi *mobile* disarankan menambahkan fitur gambar *zoom full screen* pada poting laporan.
6. Aplikasi *mobile* GoTindak juga dapat ditambahkan fungsi untuk memfasilitasi antara admin atau penanggung jawab proses pelaporan parkir dengan petugas lapangan yang menindaklanjuti laporan untuk saling berkomunikasi dan memonitoring mengenai laporan yang masuk.
7. Aplikasi GoTindak dapat ditambahkan fitur untuk memasukkan foto laporan lebih dari satu foto agar lebih memperjelas pengaduan yang ingin dilaporkan.





8. Aplikasi *mobile* GoTindak dapat ditambahkan like pada fitur komentar sehingga user dapat melihat top komen.



## DAFTAR REFERENSI

- Aldehyd, A., 2013. *Tip Trik Android dan Blackberry*. Jakarta: Andi Offset.
- Alfredo, I., 2017. Pembangunan Aplikasi Pelaporan Jalan Rusak Menggunakan Teknologi Geotagging Pada Platform Android. *Repository Jurnal Sistem Komputer UNIKOM*.
- Ambler, S. W., 2001-2008. *An Introduction to Agile Modeling*. s.l.:s.n.
- APJII, 2018. Survei APJII Penetrasi Internet di Indonesia Capai 143 Juta Jiwa. *Buketin APJII*, pp. 3-7.
- Bachtiar, A. M. & F. F. N., 2018. *Pemrograman Berorientasi Objek Menggunakan Java*. Bandung: Informatika.
- Blanco, J. U. D., 2009. *Codeigniter 1.7*. s.l.:Packet Publishing.
- Booch, G. J. R. & Ivar, J., 1999. *The Unified Modeling Language User Guide*. Germany: Longman Ltd.
- Defitria, U., 2018. Pembangunan Aplikasi Social Geotagging Destinasi Wisata Berbasis Android. *Jurnal Pengembangan Teknologi Informasi*, Volume 2, pp. 6610-6617.
- Developers, A., 2012. *Pengantar Android*. [Online] Available at: <https://developer.android.com/guide/?hl=id> [Accessed 29 April 2018].
- Developers, G., 2017. *Google Maps JavaScript API*. [Online] Available at: <https://developers.google.com/maps/documentation/android-api/> [Accessed 29 April 2018].
- Dharwiyanti, S., 2003. *Pengantar Unified Modeling Language (UML)*. s.l.:s.n.
- Dishub, 2016. *Dinas Perhubungan Kota Malang*. [Online] Available at: <https://dishub.malangkota.go.id/profil/struktur-organisasi/> [Accessed 21 February 2019].
- Dishub, 2017. *Dinas Perhubungan Kota Malang*. [Online] Available at: <https://dishub.malangkota.go.id/wp-content/uploads/sites/16/2017/05/PERSYARATAN-DAN-KRITERIA-PROSEDUR-PENYELENGGARAAN-PERPARKIRAN.pdf> [Accessed 21 February 2019].
- Dishub, K. M., 2015. Perubahan Rencana Strategis (RENSTRA). In: *Perubahan Rencana Strategis 2014-2018*. Malang: Dinas Perhubungan Kota Malang, p. 1.
- EMS, T., 2013. *Pemrograman Mobile Dengan Phonegap*. Jakarta: PT. Elex Media Komputindo.
- Estellés-arolas, E. a. G.-I.-d.-g. F., 2012. Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2), pp. 189-200.





H., N. S., 2015. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Revisi Kedua*. Bandung: Informatika Bandung.

Hariyanto, B., 2014. *Esensi-Esensi Bahasa Pemrograman Java*. Bandung: Informatika.

Hermawan, S., 2011. *Mudah Membuat Aplikasi Android*. Yogyakarta: ANDI.

Howe, 2006. *The Rise of Crowdsourcing*. [Online] Available at: <https://www.wired.com/2006/06/crowds/?pg=1&topic=crowds&topi> [Accessed 31 December 2019].

Howe, J., 2006. *The Rise of Crowdsourcing*. [Online] Available at: <https://www.wired.com/2006/06/crowds/?pg=1&topic=crowds&topi> [Accessed 31 December 2019].

Hunter, D., 2012. *eBook Geotagging: Linking Literature and Location*. [Online] Available at: <http://ir.obihiro.ac.id/dspace/handle/10322/3933> [Accessed Juni 4 2019].

JSON, 2019. *Introducing JSON*. [Online] Available at: <https://www.json.org/json-en.html> [Accessed 019 Juni 28].

JSON, 2019. *Introducing JSON*. [Online] Available at: <https://www.json.org> [Accessed Mei 11 2020].

Kurniaji, R., 2015. *Penerapan Wen Service Menggunakan JSON Untuk Mendukung Rancangan Aplikasi Pada Perguruan Tinggi*. s.l.:Akademi Manajemen Informatika dan Komputer Amik Raharja.

Mochamad, H. M. N., Achmad, C. R. & Ratna, N. T. S., 2017. *Aplikasi Rekomendasi Spot Area Wisata Berbasis Android dengan Teknik Geotag*. *Jurnal INFORM*, Volume 2.

Nandipati, A., 2011. *Assesment of metadata associated*. s.l: Institute for Geoinformatics University of Muenster.

Nazruddin, S. H., 2015. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Revisi Kedua*. Bandung: Informatika.

Nielsen, J., 2012. *Usability 101: Introduction to Usability*. [Online] Available at: <https://www.nngroup.com/articles/usability-101-introduction-tousability> [Accessed 19 Juli 2019].

Nunu, N., Asep, R. & Dian, R. N. H. S., 2019. *Penerapan Konsep Geotagging Pada Aplikasi Tanggap Darurat Bencana Berbasis Android*. *Jurnal Sistem Informasi*, Volume 6, pp. 43-47.





Nurzaman, Z., 2018. Pembangunan Aplikasi Photo Tangging Untuk Pertolongan Kecelakaan Lalu Lintas Di Kota Bandung Menggunakan Teknologi Geotagging Pada Platform Android.. *Repositori Jurnal Sistem Komputer UNIKOM*.

PowerMapper, 2019. <https://www.powermapper.com/products/sortsite/>. [Online] [Accessed 08 Agustust 2020].

Pressman, R. S., 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. 7 ed. Yogyakarta: Andi.

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th ed. America: Higher Education.

Puspa, M. N. S. A. B., Herman, T. & Fatwa, R., 2017. Designing Module E-Complaint System Based on Geotagging and Geofencing. *International Journal of Interactive Mobile Technologies (IJIM)*, Volume 11, pp. 113-129.

Rossa & Shalahuddin, M., 2014. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.

Sari, A. N. & Sunaryo, D., 2012. Perancangan dan pembanguna Perangkat Lunak Photo Uploader pada Facebook dengan Fitur Geotagging. *Jurnal Teknik Pomits*, Volume 1, pp. 1-6.

Shalahuddin, M. & R. A. S., 2018. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika

Shalahuddin, M. & Rossa, 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Obyek*. Bandung: Informatika Bandung.

Sidik, B., 2018. *Framework Codeigniter 3*. Bandung: Informatika.

Simarmata, J., 2010. *Rekayasa Perangkat Lunak*. 1 ed. Yogyakarta: s.n.

Sommerville, I., 2011. *SOFTWARE ENGINEERING*. s.l.:Pearson.

Sommerville, I., 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.

Statistik, B. P., 2018. Kota Malang Dalam Angka 2018. In: H. Prasetyo, ed. *Kota Malang Dalam Angka 2018*. Malang: BPS Kota Malang, p. 308.

Ulfani, D., 2018. Pembangunan Aplikasi Social Geotagging Destinasi Wisata Berbasis Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 6610-6617.

Wisnu, W., 2018. Rancang Bangun Aplikasi Geotagging Social Report Bencana Banjir. *Repositori Jurnal Mahasiswa PTIIK UB*, Volume 2, pp. 2817-2824.

Yasin, V., 2012. *Rekayasa Perangkat Lunak Berorientasi Objek*. Jakarta: Mitra Wacana Media.

Yudhanto, A. W. & Y., 2017. *Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio*. Jakarta: PT. Elex Media Komputindo.





## LAMPIRAN A HASIL WAWANCARA DAN KUESIONER

Berikut ini merupakan lampiran hasil wawancara secara tidak terstruktur yang dilakukan pada:

Hari/Tanggal : Rabu, 20 Februari 2019

Narasumber : Hary Dwi Yuniyanto, S.Psi

Jabatan : Seksi Pendataan

Lokasi : Dinas Perhubungan Kota Malang, Jalan Raden Intan No.1  
Malang

Tujuan dari wawancara ini untuk mengetahui persyaratan terkait parkir non resmi dan analisis kebutuhan dalam melakukan Pembangunan Aplikasi *Mobile* Identifikasi Parkir Non Resmi Berdasarkan Konsep *Crowdsourcing* Menggunakan *Geotagging*.

### B.1 Hasil Wawancara

Pertanyaan	Jawaban
Selamat siang, pak. Mohon maaf mengganggu waktu bapak.	Oh iya mbak, saya dengan pak Hary Dwi Yuniyanto
Perkenalkan saya Indah Ratna Sari dari Fakultas Ilmu Komputer Universitas Brawijaya. Saya ingin melakukan penelitian untuk skripsi saya Pembangunan Aplikasi <i>Mobile</i> Identifikasi Parkir Non Resmi Berdasarkan Konsep <i>Crowdsourcing</i> Menggunakan <i>Geotagging</i> . Maaf sebelumnya dengan bapak siapa saya melakukan wawancara ini?	
Bagaimana proses pelayanan laporan masyarakat mengenai parkir non resmi pada Dinas Perhubungan Kota Malang ?	Proses pelayanan pada pengwasan parkir kita selalu membackup baik di medsos ataupun whatsapp, tidak ada jangka waktu tertentu, laporan semuanya kita respon. Untuk proses pelaporan tersebut pihak Dishub menggunakan <i>website</i> , twitter, contact center maupun <i>email</i>





Bagaimana kriteria perpajakan antara parkir resmi dan parkir non resmi ?	Untuk parkir resmi ada dua yaitu parkir tetap dan parkir isidental yang sudah memenuhi ijin sebelumnya oleh dinas perhubungan sedangkan parkir non resmi karcis yang digunakan tidak sesuai dengan ketentuan.
Bagaimana batasan lokasi dalam parkir non resmi ?	Batasan lokasi dalam parkir non resmi meliputi tempat parkir di trotoar
Berapa jumlah titik parkir yang ada di Kota Malang ?	Jumah titik parkir di Kota Malang Kurang lebih 765 titik
Bagaimana ciri – ciri karcis parkir resmi dan parkir non resmi ?	Pada Dinas Perhubungan Kota Malang tidak pernah mencetak karcis non resmi, untuk karcis non remi tergantung ada individu masing-masing, jika beda berarti mereka melangar.
Bagaimana atribut juru parkir resmi di Kota Malang ?	Atribut juru parkir resmi memiliki Kartu Tanda Anggota, karcis remi dari dinas perhubungan, dan memakai rompi berwarna hijau.
Berapa banyak rata-rata perbulan Dinas Perhubungan Kota Malang menerima laporan mengenai parkir non resmi dari masyarakat Kota Malang ?	Masih ada rata-rata dalam 1 hari ada 5 laporan
Banyaknya laporan yang telah diterima bagaimana pihak Dinas Perhubungan dalam mengelola laporan parkir non resmi dari masyarakat Kota Malang ?	Setiap laporan langsung kita tindaklanjuti berdasarkan laoproan masyarakat melalui whatsapp dan karena keterbatasan pengawas kadang-kadang masih ada laporan yang belum kita tindak lanjuti.
Lalu apakah ada laporan parkir non resmi dari masyarakat Kota Malang	Laporan pada saat ini masih ada, setiap hari minimal ada 5 laporan





<p>selama proses laporan yang berlangsung sampai saat ini ?</p> <p>Selanjutnya dari pihak Dinas Perhubungan Kota Malang, apakah terdapat kendala dalam melakukan pengelolaan informasi laporan masyarakat mengenai parkir non resmi yang berlangsung sampai saat ini ?</p>	<p>Kendalanya keterbatasan pengawas dan kalau laporan lagi banyak agak kesusahan untuk menanganinya. Kan media yang digunakan untuk pelaporan tidak hanya satu tapi ada <i>website</i>, <i>twitter</i>, <i>email</i> dan lainnya. Itu yang menjadikan laporan-laporan tersebut menjadi tidak terurus.</p>
<p>Oh gitu pak, kalau untuk media yang biasa digunakan masyarakat untuk melaporkan pengaduan itu via apa ya pak?</p>	<p>Biasanya sih social media mbak itu paling banyak. Cuma ya gitu kalo lagi banyak yang masuk agak kesusahan. Apalagi disini tidak ada staff khusus yang menangani pengaduan yang masuk ini.</p>
<p>Kalau untuk petugas yang menindaklanjuti laporan yang masuk bagaimana pak?</p>	<p>Kalau itu jika ada laporan yang masuk mengenai pengaduan saya langsung cek permasalahannya apa, lokasinya dimana dan saya langsung menghubungi petugas yang sedang bertugas di lapangan</p>
<p>Via apa bapak menghubungi petugas yang dilapangan?</p>	<p>Biasanya saya lewat whatsapp grup</p>
<p>Apakah setiap laporan yang masuk langsung ditindaklanjuti pak?</p>	<p>Tidak mbak, yang susah itu kadang masyarakat melaporkan lokasi yang tidak akurat membuat petugas lapangan harus mencari-cari terlebih dahulu</p>
<p>Apakah sebelumnya sudah ada sistem untuk melaporkan keluhan parkir</p>	<p>Sudah ada sistem untuk melaporkan keluhan yang tertuang dalam karcis,</p>
<p>Bagaimana menyajikan konten atau informasi yang baik terkait Identifikasi Parkir Non Resmi?</p>	<p>Berdasarkan laporan masyarakat mengenai Parkir Non Resmi, biasanya informasi yang disampaikan idealnya adalah lokasi,</p>





<p>Informasi apa saja yang harus ada didalamnya?</p>	<p>terus penjelasan keluhan dan terdapat gambar keluhan dari masyarakat yang melaporkan</p>
<p>Oh begitu pak, nah penelitian skripsi saya ini berhubungan dengan laporan mengenai parkir non resmi dan yang saya tangkap dari wawancara dengan bapak, sepertinya diperlukan sebuah aplikasi yang digunakan untuk membantu masyarakat dan Dishub dalam proses pelaporan untuk mengidentifikasi laporan masyarakat terkait parkir non resmi ini pak. Nantinya aplikasi yang saya ingin bangun ini pada aplikasi <i>mobile</i> untuk masyarakat jadi masyarakat bisa melaporkan laporan dengan mengambil gambar kejadian yang sesuai, deskripsi, dan lokasi yang akurat agar mudah dilacak oleh petugas Dishub serta laporan yang masuk tersebut akan diterima oleh admin Dishub dan akan meneruskannya kepada petugas yang menangani sesuai dengan permasalahan yang terjadi dan lokasi yang akurat. Bagaimana pak?</p>	<p>Bagus tuh mbak agar masyarakat melaporkan laporan hanya pada satu media saja ya pada aplikasi itu dan pihak Dishub menjadi lebih mudah menanggapi informasi laporan yang masuk sehingga mudah untuk diidentifikasi.</p>
<p>Baik pak kalau bapak setuju. Saya akan memulai membangun aplikasi ini. Jika ada kekurangan atau sudah selesai nanti saya ke sini lagi menemui bapak.</p>	<p>Iya siap mbak. Sukses ya mbak, mudah-mudahan cepet selesai.</p>
<p>Aamiin, terimakasih pak. Kalau begitu wawancaranya sudah selesai pak, terimakasih banyak ya pak atas waktunya.</p>	<p>Iya mbak, sama-sama.</p>



## A.1 Lampiran Hasil Wawancara Dengan Kepala Bidang Pengelolaan Perparkiran

### HASIL WAWANCARA 1

Berikut ini merupakan lampiran hasil wawancara secara tidak terstruktur yang dilakukan pada:

Hari/Tanggal : Rabu, 20 Februari 2019  
Narasumber : Hary Dwi Yudianto, S.Psi  
Jabatan : Seksi Pendataan  
Lokasi : Dinas Perhubungan Kota Malang, Jalan Raden Intan No.1 Malang

Tujuan dari wawancara ini untuk mengetahui persyaratan terkait parkir non resmi dan analisis kebutuhan dalam melakukan Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging.

#### B.1 Hasil Wawancara

Pertanyaan	Jawaban
Selamat siang, pak. Mohon maaf mengganggu waktu bapak. Perkenalkan saya Indah Ratna Sari dari Fakultas Ilmu Komputer Universitas Brawijaya. Saya ingin melakukan penelitian untuk skripsi saya Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging. Maaf sebelumnya dengan bapak siapa saya melakukan wawancara ini?	Oh iya mbak, saya dengan pak Hary Dwi Yudianto
Bagaimana proses pelayanan laporan masyarakat mengenai parkir non resmi pada Dinas Perhubungan Kota Malang ?	Proses pelayanan pada pengawasan parkir kita selalu membackup baik di medsos ataupun whatsapp, tidak ada jangka waktu tertentu, laporan semuanya kita respon. Untuk proses pelaporan tersebut pihak Dishub menggunakan website, twitter, contact center maupun email
Bagaimana kriteria perparkiran antara parkir resmi dan parkir non resmi ?	Untuk parkir resmi ada dua yaitu parkir tetap dan parkir isidental yang sudah memenuhi ijin sebelumnya oleh dinas perhubungan sedangkan parkir non resmi karcis yang digunakan tidak sesuai dengan ketentuan.





Bagaimana batasan lokasi dalam parkir non resmi ?	Batasan lokasi dalam parkir non resmi meliputi tempat parkir di trotoar
Berapa jumlah titik parkir yang ada di Kota Malang ?	Jumlah titik parkir di Kota Malang Kurang lebih 765 titik
Bagaimana ciri – ciri karcis parkir resmi dan parkir non resmi ?	Pada Dinas Perhubungan Kota Malang tidak pernah mencetak karcis non resmi, untuk karcis non resmi tergantung ada individu masing-masing, jika beda berarti mereka melangar.
Bagaimana atribut juru parkir resmi di Kota Malang ?	Atribut juru parkir resmi memiliki Kartu Tanda Anggota, karcis remi dari dinas perhubungan, dan memakai rompi berwarna hijau.
Berapa banyak rata-rata perbulan Dinas Perhubungan Kota Malang menerima laporan mengenai parkir non resmi dari masyarakat Kota Malang ?	Masih ada rata-rata dalam 1 hari ada 5 laporan
Banyaknya laporan yang telah diterima bagaimana pihak Dinas Perhubungan dalam mengelola laporan parkir non resmi dari masyarakat Kota Malang ?	Setiap laporan langsung kita tindaklanjuti berdasarkan laporan masyarakat melalui whatsapp dan karena keterbatasan pengawas kadang-kadang masih ada laporan yang belum kita tindak lanjuti.
Lalu apakah ada laporan parkir non resmi dari masyarakat Kota Malang selama proses laporan yang berlangsung sampai saat ini ?	Laporan pada saat ini masih ada, setiap hari minimal ada 5 laporan
Selanjutnya dari pihak Dinas Perhubungan Kota Malang, apakah terdapat kendala dalam melakukan pengelolaan laporan sosial masyarakat mengenai parkir non resmi yang berlangsung sampai saat ini ?	Kendalanya keterbatasan pengawas dan kalau laporan lagi banyak agak kesusahan untuk menanganinya. Kan media yang digunakan untuk pelaporan tidak hanya satu tapi ada website, twitter, email dan lainnya. Itu yang menjadikan laporan-laporan tersebut menjadi tidak terurus.
Oh gitu pak, kalau untuk media yang biasa digunakan masyarakat untuk	Biasanya sih social media mbak itu paling banyak. Cuma ya gitu kalo lagi





melaporkan pengaduan itu via apa ya pak?	banyak yang masuk agak kesusahan. Apalagi disini tidak ada staff khusus yang menangani pengaduan yang masuk ini.
Kalau untuk petugas yang menindaklanjuti laporan yang masuk bagaimana pak?	Kalau itu jika ada laporan yang masuk mengenai pengaduan saya langsung cek permasalahannya apa, lokasinya dimana dan saya langsung menghubungi petugas yang sedang bertugas di lapangan.
Via apa bapak menghubungi petugas yang dilapangan?	Biasanya saya lewat whatsapp grup
Apakah setiap laporan yang masuk langsung ditindaklanjuti pak?	Tidak mbak, yang susah itu kadang masyarakat melaporkan lokasi yang tidak akurat membuat petugas lapangan harus mencari-cari terlebih dahulu
Apakah sebelumnya sudah ada sistem untuk melaporkan keluhan parkir	Sudah ada sistem untuk melaporkan keluhan yang tertuang dalam karcis.
Bagaimana menyajikan konten atau informasi yang baik terkait Identifikasi Parkir Non Resmi? Informasi apa saja yang harus ada didalamnya?	Berdasarkan laporan masyarakat mengenai Parkir Non Resmi, biasanya informasi yang disampaikan idealnya adalah lokasi, terus penjelasan keluhan dan terdapat gambar keluhan dari masyarakat yang melaporkan
Oh begitu pak, nah penelitian skripsi saya ini berhubungan dengan laporan mengenai parkir non resmi dan yang saya tangkap dari wawancara dengan bapak, sepertinya diperlukan sebuah aplikasi yang digunakan untuk membantu masyarakat dan Dishub dalam proses pelaporan untuk mengidentifikasi laporan masyarakat terkait parkir non resmi ini pak. Nantinya aplikasi yang saya ingin bangun ini pada aplikasi mobile untuk masyarakat jadi masyarakat bisa melaporkan laporan dengan mengambil gambar kejadian yang sesuai, deskripsi, dan lokasi yang akurat agar mudah	Bagus tuh mbak agar masyarakat melaporkan laporan hanya pada satu media saja ya pada aplikasi itu dan pihak Dishub menjadi lebih mudah menangani laporan sosial yang masuk sehingga mudah untuk di identifikasi.





dilacak oleh petugas Dishub serta laporan yang masuk tersebut akan diterima oleh admin Dishub dan akan meneruskannya kepada petugas yang menangani sesuai dengan permasalahan yang terjadi dan lokasi yang akurat. Bagaimana pak?	
Baik pak kalau bapak setuju. Saya akan memulai membangun aplikasi ini. Jika ada kekurangan atau sudah selesai nanti saya ke sini lagi menemui bapak.	Iya siap mbak. Sukses ya mbak, mudah-mudahan cepet selesai.
Aamiin, terimakasih pak. Kalau begitu wawancaranya sudah selesai pak, terimakasih banyak ya pak atas waktunya.	Iya mbak, sama-sama.

Penulis

Malang, 17/12/2019

Narasumber

Indah Ratna Sari

Hary Dwi Yudianto, S.Psi



## A.2 Lampiran Lembar Validasi Kuesioner Pengguna (Pelapor)

### LEMBAR VALIDASI KUISIONER

Judul Penelitian : Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging

Penyusun : Indah Ratna Sari

Prodi : Teknik Informatika

Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd

2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Nama Validator :

Ahli Bidang :

Petunjuk:

Untuk setiap pernyataan berikut berilah tanda cek (✓), silahkan anda memilih jawaban sesuai dengan pengalaman Anda terhadap penggunaan layanan parkir kendaraan pribadi di Kota Malang dengan skala penilaian sebagai berikut:

- 1 : Sangat Tidak Setuju
- 2 : Tidak Setuju
- 3 : Netral
- 4 : Setuju
- 5 : Sangat Setuju

Note: Semakin tinggi nilai yang diberikan maka Anda semakin setuju dengan pernyataan yang diajukan.

No.	Pertanyaan	Tingkat Persetujuan					Kitik/Saran
		STS	TS	N	S	ST	
1	Saya adalah pengguna kendaraan pribadi (motor/mobil)				✓		
2	Saya pernah melakukan parkir dengan kendaraan pribadi				✓		
3	Saya tidak mengetahui informasi mengenai penetapan lokasi parkir resmi ataupun non resmi				✓		
4	Saya tidak mengetahui informasi tarif parkir resmi yang ditetapkan oleh Dinas Perhubungan				✓		





No.	Pertanyaan	Tingkat Persetujuan					Kitik/Saran
		STS	TS	N	S	ST	
5	Saya pernah mengalami penarikan tarif parkir yang tidak sesuai atau tarif parkir mahal				✓		
6	Saya merasa pentingnya informasi untuk mengetahui ciri-ciri parkir resmi dan parkir non resmi				✓		
7	Saya merasa pentingnya informasi untuk mengetahui perbedaan parkir tetap dan parkir tidak tetap				✓		
8	Saya merasa pentingnya untuk mengetahui informasi tarif parkir tetap				✓		
9	Saya merasa pentingnya untuk mengetahui informasi tarif parkir tidak tetap				✓		
10	Saya dapat mengetahui perbedaan karcis parkir resmi dan parkir tidak resmi				✓		
11	Saya dapat mengetahui atribut juru parkir resmi				✓		
12	Saya mendapat pengalaman keluhan pada saat melakukan parkir kendaraan pribadi				✓		
13	Saya tidak mengetahui wadah untuk menyampaikan laporan keluhan parkir				✓		
14	Saya sangat mungkin mempunyai kendala dalam melakukan penyampaian keluhan parkir kendaraan pribadi terutama parkir non resmi				✓		
15	Saya merasa terbantu dengan adanya wadah untuk menyampaikan laporan sosial parkir resmi berupa aplikasi				✓		









### LEMBAR VALIDASI KUISIONER

Judul Penelitian : Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging

Penyusun : Indah Ratna Sari

Prodi : Teknik Informatika

Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd.  
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Nama Validator :

Ahli Bidang :

Petunjuk:

Untuk setiap pernyataan berikut berilah tanda cek (v), silahkan anda memilih jawaban sesuai dengan pengalaman Anda terhadap penggunaan layanan parkir kendaraan pribadi di Kota Malang dengan skala penilaian sebagai berikut:

- 1 : Sangat Tidak Setuju
- 2 : Tidak Setuju
- 3 : Netral
- 4 : Setuju
- 5 : Sangat Setuju

*rus*

Note: Semakin tinggi nilai yang diberikan maka Anda semakin setuju dengan pernyataan yang diajukan.

No.	Pertanyaan	Tingkat Persetujuan					Kitik/Saran
		STS	TS	N	S	ST	
1	Saya adalah pengguna kendaraan pribadi (motor/mobil)						
2	Saya pernah melakukan parkir dengan kendaraan pribadi						
3	Saya tidak mengetahui informasi mengenai penetapan lokasi parkir resmi ataupun non resmi						
4	Saya tidak mengetahui informasi tarif parkir resmi yang ditetapkan oleh Dinas Perhubungan						





No.	Pertanyaan	Tingkat Persetujuan					Kritik/Saran
		STS	TS	N	S	ST	
5	Saya pernah mengalami penarikan tarif parkir yang tidak sesuai atau tarif parkir mahal						
6	Saya merasa pentingnya informasi untuk mengetahui ciri-ciri parkir resmi dan parkir non resmi						
7	Saya merasa pentingnya informasi untuk mengetahui perbedaan parkir tetap dan parkir tidak tetap						
8	Saya merasa pentingnya untuk mengetahui informasi tarif parkir tetap						
9	Saya merasa pentingnya untuk mengetahui informasi tarif parkir tidak tetap						
10	Saya dapat mengetahui perbedaan karcis parkir resmi dan parkir tidak resmi						
11	Saya dapat mengetahui atribut juru parkir resmi						
12	Saya mendapat pengalaman keluhan pada saat melakukan parkir kendaraan pribadi						
13	Saya tidak mengetahui wadah untuk menyampaikan laporan keluhan parkir						
14	Saya sangat mungkin mempunyai kendala dalam melakukan penyampaian keluhan parkir kendaraan pribadi terutama parkir non resmi						
15	Saya merasa terbantu dengan adanya wadah untuk menyampaikan laporan sosial parkir resmi berupa aplikasi						





16	Saya merasa perlu adanya tindakan lanjut dari pelaporan parkir non resmi						
17	Saya sangat mungkin untuk menggunakan aplikasi laporan identifikasi parkir non resmi						

Simpulan Validator/Penilai

Lingkari jawaban berikut ini sesuai dengan kesimpulan anda:

A. Lembar Validasi Kuisisioner ini:

- 1. Sangat Tidak Setuju
- 2. Tidak Setuju
- 3. Cukup Setuju
- 4. Setuju
- 5. Sangat setuju

B. Lembar Validasi Kuisisioner:

- 1. Dapat digunakan tanpa revisi
- 2. Dapat digunakan dengan sedikit revisi
- 3. Dapat digunakan dengan banyak revisi
- 4. Belum dapat digunakan.

Saran:

referensi terkait pembuatan pertanyaan

---



---



---



---

Malang, 5 Mei 2019

Validator/Penilai,

(Lutfi Fauzan)





**LEMBAR VALIDASI KUISIONER**

Judul Penelitian : Pembangunan Aplikasi Mobile identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging

Penyusun : Indah Ratna Sari

Prodi : Teknik Informatika

Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd  
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Nama Validator :

Ahli Bidang :

Petunjuk:

Untuk setiap pernyataan berikut berilah tanda cek (v), silahkan anda memilih jawaban sesuai dengan pengalaman Anda terhadap penggunaan layanan parkir kendaraan pribadi di Kota Malang dengan skala penilaian sebagai berikut:

- 1 : Sangat Tidak Setuju
- 2 : Tidak Setuju
- 3 : Netral
- 4 : Setuju
- 5 : Sangat Setuju

Note: Semakin tinggi nilai yang diberikan maka Anda semakin setuju dengan pernyataan yang diajukan.

**DAFTAR PERTANYAAN**

No.	Pertanyaan	Tingkat Persetujuan					Kritik/Saran
		STS	TS	N	S	ST	
1	Saya adalah pengguna kendaraan pribadi (motor/mobil)						Sudah digunakan & Dulu Dir.
2	Saya pernah melakukan parkir dengan kendaraan pribadi						
3	Saya tidak mengetahui informasi mengenai penetapan lokasi parkir resmi ataupun non resmi						
4	Saya tidak mengetahui informasi tarif parkir resmi yang ditetapkan oleh Dinas Perhubungan						





No.	Pertanyaan	Tingkat Persetujuan					Kritik/Saran
		STS	TS	N	S	ST	
5	Saya pernah mengalami penarikan tarif parkir yang tidak sesuai atau tarif parkir mahal						
6	Saya merasa pentingnya informasi untuk mengetahui ciri-ciri parkir resmi dan parkir non resmi						
7	Saya merasa pentingnya informasi untuk mengetahui perbedaan parkir tetap dan parkir tidak tetap						resmi vs non resmi dan tetap vs tidak tetap
8	Saya merasa pentingnya untuk mengetahui informasi tarif parkir tetap						
9	Saya merasa pentingnya untuk mengetahui informasi tarif parkir tidak tetap						
10	Saya dapat mengetahui perbedaan karcis parkir resmi dan parkir tidak resmi						
11	Saya dapat mengetahui atribut juru parkir resmi						
12	Saya mendapat pengalaman keluhan pada saat melakukan parkir kendaraan pribadi						
13	Saya tidak mengetahui wadah untuk menyampaikan laporan keluhan parkir						
14	Saya sangat mungkin mempunyai kendala dalam melakukan penyampaian keluhan parkir kendaraan pribadi terutama parkir non resmi						
15	Saya merasa terbantu dengan adanya wadah untuk menyampaikan laporan sosial parkir resmi berupa aplikasi						





16	Saya merasa perlu adanya tindaklanjut dari pelaporan parkir non resmi								
17	Saya sangat mungkin untuk menggunakan aplikasi laporan identifikasi parkir non resmi								

Simpulan Validator/Penilai

Lingkari jawaban berikut ini sesuai dengan kesimpulan anda:

A. Lembar Validasi Kuisisioner ini:

1. Sangat Tidak Setuju
2. Tidak Setuju
3. Cukup Setuju
4. Setuju
5. Sangat setuju

B. Lembar Validasi Kuisisioner:


1. Dapat digunakan tanpa revisi
2. Dapat digunakan dengan sedikit revisi
3. Dapat digunakan dengan banyak revisi
4. Belum dapat digunakan.

Saran:

*Sewa tabel dan ketarangan lainnya*

Malang 6 Maret 2019

Validator/Penilai,



(Asy. Agus Khairama)



### A.3 Lampiran Hasil Kuesioner Pengguna (Pelapor)

**Data Diri Responden**

Nama Lengkap : **ROBI ANI SETIAWAN**  
 Kota tempat tinggal saat ini : **MALANG**  
 Jenis Kelamin : **LAKI-LAKI**  
 Laki-laki  
 Perempuan  
 Umur : **22 TAHUN**  
 Pekerjaan : **MAHASISWA**  
 PNS  
 Karyawan  
 Mahasiswa  
 Lainnya:  
 ID line / Whatsapp : **0812127**  
 (\*jika berkenan)  
 Alamat e-mail : **robi-ani.setiawan@yqf.co.com**  
 (\*jika berkenan)

#### DAFTAR PERTANYAAN

No.	Pertanyaan	Tingkat Persetujuan				
		1	2	3	4	5
		STS	TS	N	S	SS
1.	Saya pernah mendapati adanya keluhan pada saat melakukan parkir kendaraan pribadi.					✓
2.	Saya tidak pernah melakukan suatu laporan parkir non resmi kepada Dinas Perhubungan Kota Malang melalui media elektronik ataupun langsung.				✓	
3.	Saya tidak mengetahui prosedur untuk melaporkan parkir non resmi				✓	





NO.	Pertanyaan	Pertanyaan				
		1	2	3	4	5
		STS	TS	N	S	SS
4.	Saya merasa kesulitan dalam melakukan suatu laporan parkir non resmi kepada pihak terkait				✓	
5.	Saya pernah mengalami penarikan tarif parkir yang tidak sesuai atau tarif parkir mahal.		✓			
6.	Menurut saya, perlu adanya informasi untuk mengetahui ciri-ciri parkir resmi dan parkir non resmi.					✓
7.	Saya tidak mengetahui wadah untuk menyampaikan suatu laporan parkir non resmi kepada pihak terkait.				✓	
8.	Saya merasa terbantu dengan adanya wadah untuk menyampaikan laporan sosial parkir resmi berupa aplikasi mobile				✓	
9.	Saya merasa pentingnya untuk melihat laporan yang telah orang lain kirim.			✓		
10.	Saya sangat mungkin untuk melihat laporan yang telah saya kirim sebelumnya.				✓	
11.	Saya merasa sangat perlu untuk memberikan dukungan pada laporan yang telah orang lain kirim.				✓	
12.	Saya sangat mungkin untuk memberikan komentar pada laporan yang telah orang lain kirim.				✓	
13.	Saya merasa perlu adanya tindak lanjut dari laporan parkir non resmi.					✓
14.	Saya sangat mungkin untuk menggunakan aplikasi laporan identifikasi parkir non resmi.				✓	





**Data Diri Responden**

Nama Lengkap : Made Widya Anyari  
 Kota tempat tinggal saat ini : Malang  
 Jenis Kelamin :  
 Laki-laki  
 Perempuan  
 Umur : 22 thn  
 Pekerjaan :  
 PNS  
 Karyawan  
 Mahasiswa  
 Lainnya:  
 ID line / Whatsapp : widyanjani  
 (\*jika berkenan)  
 Alamat e-mail : widyanjani3@gmail.com  
 (\*jika berkenan)

**DAFTAR PERTANYAAN**

No.	Pertanyaan	Tingkat Persetujuan				
		1	2	3	4	5
		STS	TS	N	S	SS
1.	Saya pernah mendapati adanya keluhan pada saat melakukan parkir kendaraan pribadi.				✓	
2.	Saya tidak pernah melakukan suatu laporan parkir non resmi kepada Dinas Perhubungan Kota Malang melalui media elektronik ataupun langsung.				✓	
3.	Saya tidak mengetahui prosedur untuk melaporkan parkir non resmi				✓	





NO.	Pertanyaan	Pertanyaan				
		1	2	3	4	5
		STS	TS	N	S	SS
4.	Saya merasa kesulitan dalam melakukan suatu laporan parkir non resmi kepada pihak terkait				✓	
5.	Saya pernah mengalami penarikan tarif parkir yang tidak sesuai atau tarif parkir mahal.			✓		
6.	Menurut saya, perlu adanya informasi untuk mengetahui ciri-ciri parkir resmi dan parkir non resmi.					✓
7.	Saya tidak mengetahui wadah untuk menyampaikan suatu laporan parkir non resmi kepada pihak terkait.					✓
8.	Saya merasa terbantu dengan adanya wadah untuk menyampaikan laporan sosial parkir resmi berupa aplikasi mobile					✓
9.	Saya merasa pentingnya untuk melihat laporan yang telah orang lain kirim.			✓		
10.	Saya sangat mungkin untuk melihat laporan yang telah saya kirim sebelumnya.				✓	
11.	Saya merasa sangat perlu untuk memberikan dukungan pada laporan yang telah orang lain kirim.				✓	
12.	Saya sangat mungkin untuk memberikan komentar pada laporan yang telah orang lain kirim.				✓	
13.	Saya merasa perlu adanya tindak lanjut dari laporan parkir non resmi.					✓
14.	Saya sangat mungkin untuk menggunakan aplikasi laporan identifikasi parkir non resmi.				✓	





## LAMPIRAN B EVALUASI PENGGUNA TERHADAP PROTOTYPE

### B.1 Lampiran Lembar Evaluasi Pengguna (Pelapor)

#### LEMBAR VALIDASI PENELITIAN

Judul Penelitian : Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging

Penyusun : Indah Ratna Sari

Prodi : Teknik Informatika

Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd  
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Nama Validator : Agi Putra Kharisma

Ahli Bidang : Rekayasa Perangkat Lunak

Petunjuk Pengisian :

Lembar validasi penelitian ini bertujuan untuk mengetahui pendapat Bapak/Ibu mengenai penelitian yang dilakukan oleh peneliti untuk mengumpulkan data penelitian. Tujuan dilakukannya evaluasi ini mengetahui instrumen pertanyaan yang valid dan reliabel. Sehubungan dengan hal tersebut, saya mohon Bapak/Ibu untuk memberikan respon berupa saran dan kritik pada setiap pertanyaan yang telah tersedia pada kolom di bawah ini:

#### DAFTAR PERTANYAAN

No	Pertanyaan	Jawaban
1	Bagaimana menurut pendapat anda mengenai desain UI dan UX pada aplikasi ini?	
	<ul style="list-style-type: none"> <li>• Bagaimana penempatan skema icon yang diterapkan pada navigation bottom dan menyunya?</li> </ul>	Sudah baik + skema warna
	<ul style="list-style-type: none"> <li>• Bagaimana dengan icon yang digunakan, apakah tampak familiar?</li> </ul>	Sudah familiar
	<ul style="list-style-type: none"> <li>• Bagaimana cara mengoperasikan aplikasi, apakah cukup memudahkan pengguna?</li> </ul>	Cukup mudah
	<ul style="list-style-type: none"> <li>• Apakah letak icon dukungan dan share sudah sesuai?</li> </ul>	Sudah
2	Apakah penempatan logout pada menu setting sudah sesuai?	Ya.





	• Jika tidak, ditempatkan dimana yang bagus?	
3	Apakah informasi pada menu beranda yang ditampilkan sudah sesuai atau belum?	Sudah cukup.
	• Jika belum, informasi apa yang perlu ditambahkan?	
4	Apakah status laporan parkir non resmi untuk mengetahui tindakan lanjut oleh pihak dinas perhubungan diperlukan untuk anda?	Perlu.
5	Apakah informasi yang diberikan relevan dengan apa yang diinginkan pengguna?	Ya, untuk parkir non resmi yang sudah ditindaklanjuti disamakan (b/c/d).
6	Apakah fitur share laporan informasi parkir non resmi ke social media membantu anda?	Ya
7	Fungsi apa yang menurut anda perlu ditampilkan lagi?	Jumlah rekening, Tanggungan Foto

Simpulan Validator/Penilai

Lingkari jawaban berikut ini sesuai dengan kesimpulan anda:

Mengacu pada validasi penelitian yang dilakukan, dapat dinyatakan bahwa :

A. Lembar Validasi Kuisisioner:

1. Sangat Tidak Setuju
2. Tidak Setuju
3. Cukup Setuju
4. Setuju
5. Sangat setuju

B. Lembar Validasi Kuisisioner:

1. Dapat digunakan tanpa revisi
2. Dapat digunakan dengan sedikit revisi
3. Dapat digunakan dengan banyak revisi
4. Belum dapat digunakan.









### LEMBAR VALIDASI PENELITIAN

Judul Penelitian : Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging

Penyusun : Indah Ratna Sari

Prodi : Teknik Informatika

Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd  
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Nama Validator :

Ahli Bidang :

Petunjuk Pengisian :

Lembar validasi penelitian ini bertujuan untuk mengetahui pendapat Bapak/Ibu mengenai penelitian yang dilakukan oleh peneliti untuk mengumpulkan data penelitian. Tujuan dilakukannya evaluasi ini mengetahui instrumen pertanyaan yang valid dan reliabel. Sehubungan dengan hal tersebut, saya mohon Bapak/Ibu untuk memberikan respon berupa saran dan kritik pada setiap pertanyaan yang telah tersedia pada kolom di bawah ini:

### DAFTAR PERTANYAAN

No	Pertanyaan	Jawaban
1	Bagaimana menurut pendapat anda mengenai desain UI dan UX pada aplikasi ini?	
	<ul style="list-style-type: none"> <li>• Bagaimana penempatan skema icon yang diterapkan pada navigation bottom dan menyunya?</li> </ul>	Ya.
	<ul style="list-style-type: none"> <li>• Bagaimana dengan icon yang digunakan, apakah tampak familiar?</li> </ul>	Ya.
	<ul style="list-style-type: none"> <li>• Bagaimana cara mengoperasikan aplikasi, apakah cukup memudahkan pengguna?</li> </ul>	Ya.
	<ul style="list-style-type: none"> <li>• Apakah letak icon <del>dukungan dan</del> share sudah sesuai? <sup>like</sup></li> </ul>	Ya.
2	Apakah penempatan logout pada menu setting sudah sesuai?	Ya





	• Jika tidak, ditempatkan dimana yang bagus?	
3	Apakah informasi pada menu beranda yang ditampilkan sudah sesuai atau belum?	Ya.
	• Jika belum, informasi apa yang perlu ditambahkan?	Saran: ditambah keterangan yg tampil apa parkir lar / bukan
4	Apakah status laporan parkir non resmi untuk mengetahui tindakan lanjut oleh pihak dinas perhubungan diperlukan untuk anda?	Ya.
5	Apakah informasi yang diberikan relevan dengan apa yang diinginkan pengguna?	Ya.
6	Apakah fitur share laporan informasi parkir non resmi ke social media membantu anda?	Ya.
7	Fungsi apa yang menurut anda perlu ditampilkan lagi?	-

#### Simpulan Validator/Penilai

Lingkari jawaban berikut ini sesuai dengan kesimpulan anda:

Mengacu pada validasi penelitian yang dilakukan, dapat dinyatakan bahwa :

#### A. Lembar Validasi Kuisisioner:

1. Sangat Tidak Setuju
2. Tidak Setuju
3. Cukup Setuju
4. Setuju
5. Sangat setuju

#### B. Lembar Validasi Kuisisioner:

1. Dapat digunakan tanpa revisi
2. Dapat digunakan dengan sedikit revisi
3. Dapat digunakan dengan banyak revisi
4. Belum dapat digunakan.









### LEMBAR VALIDASI PENELITIAN

Judul Penelitian : Pembangunan Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging

Penyusun : Indah Ratna Sari

Prodi : Teknik Informatika

Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd  
2. Adam Hendra Brata, S.Kom., M.T., M.Sc.

Nama Validator : Niken Hendekusma W., S.Kom., M.Kom.

Ahli Bidang : Manajemen Data & Informasi

Petunjuk Pengisian :

Lembar validasi penelitian ini bertujuan untuk mengetahui pendapat Bapak/Ibu mengenai penelitian yang dilakukan oleh peneliti untuk mengumpulkan data penelitian. Tujuan dilakukannya evaluasi ini mengetahui instrumen pertanyaan yang valid dan reliabel. Sehubungan dengan hal tersebut, saya mohon Bapak/Ibu untuk memberikan respon berupa saran dan kritik pada setiap pertanyaan yang telah tersedia pada kolom di bawah ini:

### DAFTAR PERTANYAAN

No	Pertanyaan	Jawaban
1	Bagaimana menurut pendapat anda mengenai desain UI dan UX pada aplikasi ini?	
	<ul style="list-style-type: none"> <li>• Bagaimana penempatan skema icon yang diterapkan pada navigation bottom dan menunya?</li> </ul>	seharusnya tidak perlu scroll ketika navigasi menu. Di bawah
	<ul style="list-style-type: none"> <li>• Bagaimana dengan icon yang digunakan, apakah tampak familiar?</li> </ul>	setting → akan lebih mudah untuk info → ya
	<ul style="list-style-type: none"> <li>• Bagaimana cara mengoperasikan aplikasi, apakah cukup memudahkan pengguna?</li> </ul>	mudah
	<ul style="list-style-type: none"> <li>• Apakah letak icon dukungan dan share sudah sesuai?</li> </ul>	ya
2	Apakah penempatan logout pada menu setting sudah sesuai?	
		cari guideline terkait penempatan logout pada aplikasi mobile





	<ul style="list-style-type: none"> <li>Jika tidak, ditempatkan dimana yang bagus?</li> </ul>	
3	Apakah informasi pada menu beranda yang ditampilkan sudah sesuai atau belum?	sesuai, & tidak show
	<ul style="list-style-type: none"> <li>Jika belum, informasi apa yang perlu ditambahkan?</li> </ul>	judul
4	Apakah status laporan parkir non resmi untuk mengetahui tindak lanjut oleh pihak dinas perhubungan diperlukan untuk anda?	tidak
5	Apakah informasi yang diberikan relevan dengan apa yang diinginkan pengguna?	iya
6	Apakah fitur share laporan informasi parkir non resmi ke social media membantu anda?	ya
7	Fungsi apa yang menurut anda perlu ditampilkan lagi?	status / history & tampilan akrasian reangit

Simpulan Validator/Penilai

Lingkari jawaban berikut ini sesuai dengan kesimpulan anda:

Mengacu pada validasi penelitian yang dilakukan, dapat dinyatakan bahwa :

A. Lembar Validasi Kuisioner:

- Sangat Tidak Setuju
- Tidak Setuju
- Cukup Setuju
4. Setuju
- Sangat setuju

B. Lembar Validasi Kuisioner:

- Dapat digunakan tanpa revisi
2. Dapat digunakan dengan sedikit revisi
- Dapat digunakan dengan banyak revisi
- Belum dapat digunakan.









## LAMPIRAN C DOKUMEN PENDUKUNG

### C.1 Kartu Identitas Petugas Parkir







### C.2 Bukti Fisik Karcis Parkir Resmi









### C.3 Surat Permohonan Data Skripsi

**KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI**  
**UNIVERSITAS BRAWIJAYA**  
**FAKULTAS ILMU KOMPUTER**  
Gedung A FILKOM Lt. 1, Jl. Veteran No.8, Malang, 65145, Indonesia  
Telp. : +62-341-577911; Fax : +62-341-577911  
http://filkom.ub.ac.id E-mail : filkom@ub.ac.id

04 FEB 2019

Nomor : 319 /UN10.F15.05/PP/2019  
Perihal : *Permohonan Data Skripsi*

Yth. Kepala Badan Kesatuan Bangsa dan Politik Kota Malang  
Jl. A. Yani No. 93  
Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama : Indah Ratna Sari  
NIM : 175150209111029  
Judul Skripsi : Pembangunan Aplikasi Mobile Geotagging Parkir Non Resmi Berbasis Laporan Sosial Pada Platform Android  
Dosen Pembimbing : 1. Faizatul Amalia, S.Pd., M.Pd  
2. Andi Reza Perdanakusuma, S.Kom., M.MT.  
Prodi : Teknik Informatika

Guna melakukan survey atau pengambilan data untuk skripsi mahasiswa tersebut di Dinas Perhubungan Kota Malang, jenis survey atau data yang diperlukan dan rencana waktu pelaksanaan adalah :

Data : Jumlah Kendaraan Di Kota Malang  
Waktu : 29 Januari 2019 - 29 Juni 2019

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

Mengetahui,  
Dosen Pembimbing I

a.n. Dekan  
Kampus Tata Usaha,

  
Faizatul Amalia, S.Pd., M.Pd  
NIK/ NIP 2013096608212001

  
Syarif Luthur Utami, S.E  
NIK/ NIP 196110041994032001

**Tembusan Kepada Yth:**

1. Ketua Program Studi Teknik Informatika
2. Kepala Dinas Perhubungan Kota Malang
3. Mahasiswa yang bersangkutan





KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI  
UNIVERSITAS BRAWIJAYA  
FAKULTAS ILMU KOMPUTER  
Gedung A FILKOM LL 1, Jl. Veteran No.8, Malang, 65145, Indonesia  
Telp. : +62-341-577911; Fax : +62-341-577911  
<http://filkom.ub.ac.id/e-mail> - [filkom@ub.ac.id](mailto:filkom@ub.ac.id)

Nomor : 390 /UN10.F15.05/PP/2019  
Perihal : *Permohonan Data Skripsi*

11 FEB 2019

Yth. Kepala Dinas Perhubungan Kota Malang  
Jl. Raden Intan No.1  
Malang

Untuk mendukung penyelesaian skripsi mahasiswa berikut :

Nama	: Indah Ratna Sari
NIM	: 175150209111029
Judul Skripsi	: <i>Pembangunan Aplikasi Mobile Geotagging Parkir Non Resmi Berbasis Laporan Sosial Pada Platform Android</i>
Dosen Pembimbing	: Faizatul Amalia, S.Pd.,M.Pd
Prodi	: Teknik Informatika

Guna melakukan survey atau pengambilan data untuk skripsi mahasiswa tersebut di Instansi Bapak / Ibu, jenis survey atau data yang diperlukan dan rencana waktu pelaksanaan adalah :

Data	: Pengambilan Laporan Sosial Masyarakat Mengenai Parkir Liar Di Kota Malang dan Wawancara Mengenai Prosedur Perparkiran
Waktu	: 29 Januari 2019 - 29 Juni 2019

Atas perhatian dan kerjasamanya kami ucapkan terima kasih

Mengetahui,  
Dosen Pembimbing

Faizatul Amalia, S.Pd.,M.Pd  
NIK/ NIP 2013098608212001

a.n. Dekan  
Kepala Tata Usaha,

Sri Suyati Luhur Utami, S.E  
NIP 196110041994032001

**Tembusan Kepada Yth:**

1. Ketua Program Studi Teknik Informatika
2. Mahasiswa yang bersangkutan






## LAMPIRAN D PENGUJIAN

### D.1 Kuesioner Pengujian Usability Pengguna (Pelapor)

#### Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging

Nama Lengkap : ANA KUSUMA A  
 Umur : 22 TAHUN  
 Pekerjaan : MAHASISWA  
 Tanda Tangan : 

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi mobile identifikasi parkir non resmi atau aplikasi "GoTindak" ini untuk melaporkan adanya parkir non resmi sebagai alternatif media pelaporan			✓		
2	Saya merasa aplikasi "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan	✓				
3	Saya merasa dapat menjalankan aplikasi "GoTindak" ini dengan mudah				✓	
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi "GoTindak" ini		✓			
5	Saya merasa aplikasi "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik				✓	
6	Saya menilai bahwa sistem aplikasi "GoTindak" ini banyak hal yang tidak konsisten		✓			





No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi "GoTindak" ini akan sangat mudah digunakan oleh banyak orang				✓	
8	Saya menilai aplikasi "GoTindak" ini sangat rumit ketika digunakan		✓			
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi "Gotindak" ini dengan baik				✓	
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi "GoTindak" ini		✓			





**Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging**

Nama Lengkap : *Muhammad Zulfri*  
 Umur : *22 Tahun*  
 Pekerjaan : *Mahasiswa*  
 Tanda Tangan : *[Signature]*

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi mobile identifikasi parkir non resmi atau aplikasi "GoTindak" ini untuk melaporkan adanya parkir non resmi sebagai alternatif media pelaporan				✓	
2	Saya merasa aplikasi "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan		✓			
3	Saya merasa dapat menjalankan aplikasi "GoTindak" ini dengan mudah					✓
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi "GoTindak" ini	✓				
5	Saya merasa aplikasi "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik				✓	
6	Saya menilai bahwa sistem aplikasi "GoTindak" ini banyak hal yang tidak konsisten	✓				





No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi "GoTindak" ini akan sangat mudah digunakan oleh banyak orang					✓
8	Saya menilai aplikasi "GoTindak" ini sangat rumit ketika digunakan		✓			
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi "Gotindak" ini dengan baik					✓
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi "GoTindak" ini	✓				





### Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging

Nama Lengkap : Shinta Aprilisa  
Umur : 23  
Pekerjaan : Mahasiswa

Tanda Tangan : 

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi mobile identifikasi parkir non resmi atau aplikasi "GoTindak" ini untuk melaporkan adanya parkir non resmi sebagai alternatif media pelaporan				✓	
2	Saya merasa aplikasi "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan		✓			
3	Saya merasa dapat menjalankan aplikasi "GoTindak" ini dengan mudah				✓	
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi "GoTindak" ini	✓				
5	Saya merasa aplikasi "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik				✓	
6	Saya menilai bahwa sistem aplikasi "GoTindak" ini banyak hal yang tidak konsisten	✓				





No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi "GoTindak" ini akan sangat mudah digunakan oleh banyak orang					✓
8	Saya menilai aplikasi "GoTindak" ini sangat rumit ketika digunakan	✓				
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi "Gotindak" ini dengan baik				✓	
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi "GoTindak" ini	✓				





### Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging

Nama Lengkap : RIZNIN AYUSTIA  
 Umur : 22 th  
 Pekerjaan : MAHASISWA  
 Tanda Tangan :

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi mobile identifikasi parkir non resmi atau aplikasi "GoTindak" ini untuk melaporkan adanya parkir non resmi sebagai alternatif media pelaporan				✓	
2	Saya merasa aplikasi "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan		✓			
3	Saya merasa dapat menjalankan aplikasi "GoTindak" ini dengan mudah				✓	
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi "GoTindak" ini	✓				
5	Saya merasa aplikasi "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik					✓
6	Saya menilai bahwa sistem aplikasi "GoTindak" ini banyak hal yang tidak konsisten	✓				






No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi "GoTindak" ini akan sangat mudah digunakan oleh banyak orang				✓	
8	Saya menilai aplikasi "GoTindak" ini sangat rumit ketika digunakan		✓			
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi "Gotindak" ini dengan baik				✓	
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi "GoTindak" ini		✓			





### Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging

Nama Lengkap : Riska Octaviani  
 Umur : 23 Tahun  
 Pekerjaan : Mahasiswa  
 Tanda Tangan : 

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi mobile identifikasi parkir non resmi atau aplikasi "GoTindak" ini untuk melaporkan adanya parkir non resmi sebagai alternatif media pelaporan			✓		
2	Saya merasa aplikasi "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan		✓			
3	Saya merasa dapat menjalankan aplikasi "GoTindak" ini dengan mudah				✓	
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi "GoTindak" ini		✓			
5	Saya merasa aplikasi "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik					✓
6	Saya menilai bahwa sistem aplikasi "GoTindak" ini banyak hal yang tidak konsisten	✓				





No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi "GoTindak" ini akan sangat mudah digunakan oleh banyak orang				✓	
8	Saya menilai aplikasi "GoTindak" ini sangat rumit ketika digunakan		✓			
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi "Gotindak" ini dengan baik				✓	
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi "GoTindak" ini	✓				



## D.2 Kuesioner Pengujian Usability Administrator

### Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi Berdasarkan Laporan Sosial Menggunakan Geotagging

Nama Lengkap : *Dani Anggraeni*  
Umur : *40th*  
Pekerjaan : *PAS*  
Tanda Tangan : *Dani*

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi website identifikasi parkir non resmi atau aplikasi "GoTindak" ini				✓	
2	Saya merasa aplikasi website "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan		✓			
3	Saya merasa dapat menjalankan aplikasi website "GoTindak" ini dengan mudah			✓		
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi website "GoTindak" ini		✓			
5	Saya merasa aplikasi website "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik				✓	
6	Saya menilai bahwa sistem aplikasi website "GoTindak" ini banyak hal yang tidak konsisten			✓		





No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi website "GoTindak" ini akan sangat mudah digunakan oleh banyak orang				✓	
8	Saya menilai aplikasi website "GoTindak" ini sangat rumit ketika digunakan		✓			
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi website "GoTindak" ini dengan baik				✓	
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi website "GoTindak" ini		✓			





**Kuesioner Usability Aplikasi Mobile Identifikasi Parkir Non Resmi  
Berdasarkan Laporan Sosial Menggunakan Geotagging**

Nama Lengkap : *Suci Satriyanti*  
 Umur : *30th*  
 Pekerjaan : *PNS*  
 Tanda Tangan : *[Signature]*

No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi website identifikasi parkir non resmi atau aplikasi "GoTindak" ini				✓	
2	Saya merasa aplikasi website "GoTindak" ini memiliki alur yang rumit saat melakukan aktifitas pelaporan		✓			
3	Saya merasa dapat menjalankan aplikasi website "GoTindak" ini dengan mudah				✓	
4	Saya merasa membutuhkan bantuan orang lain pada saat menjalankan aktifitas aplikasi website "GoTindak" ini		✓			
5	Saya merasa aplikasi website "GoTindak" ini memiliki fitur dan fungsi dalam sistem yang terintegrasi dengan baik				✓	
6	Saya menilai bahwa sistem aplikasi website "GoTindak" ini banyak hal yang tidak konsisten			✓		





No.	Pernyataan	Tingkat Persetujuan				
		Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
		1	2	3	4	5
7	Saya menilai aplikasi website "GoTindak" ini akan sangat mudah digunakan oleh banyak orang				✓	
8	Saya menilai aplikasi website "GoTindak" ini sangat rumit ketika digunakan		✓			
9	Saya merasa sangat percaya diri dapat menggunakan aplikasi website "Gotindak" ini dengan baik		✓		✓	
10	Saya perlu belajar banyak hal untuk beradaptasi terlebih dahulu, sebelum saya menggunakan aplikasi website "GoTindak" ini		✓			





## LAMPIRAN E DOKUMENTASI PENELITIAN

### E.1 Dokumentasi Wawancara



**Gambar E.1 Wawancara terhadap Kepala Bidang Pengelolaan Perparkiran**

### E.2 Dokumentasi Validasi Hasil Wawancara



**Gambar E.2 Wawancara terhadap Kepala Bidang Pengelolaan Perparkiran**





### E.3 Dokumentasi Pengujian



**Gambar E.3 Pengisian Form Lembar Kuisiner Pengujian Usability**



**Gambar E.4 Pengisian Form Lembar Kuesioner Pengujian Usability**





## E.4 Dokumentasi Pendukung



**Gambar E.5 Dokumentasi Pendukung Bukti Karcis Resmi**



**Gambar E.6 Dokumentasi Pendukung Bukti Karcis Resmi**