

# **PENGEMBANGAN APLIKASI LAYANAN PENGADUAN MASYARAKAT UNTUK DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA PASURUAN BERBASIS ANDROID**

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Hammer Zoelfagar Mahdias  
NIM: 135150401111096



**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

PENGEMBANGAN APLIKASI LAYANAN PENGADUAN MASYARAKAT UNTUK DINAS  
KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA PASURUAN  
BERBASIS ANDROID

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Hammer Zoelfagar Mahdias  
NIM: 135150401111096

Skripsi ini telah diuji dan dinyatakan lulus pada  
1 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

*Handwritten signature*

Himawat Aryadita, S.T, M.Sc  
NIP: 19801018 200801 1 003

Dosen Pembimbing II

*Handwritten signature*

Satrio Agung Wicaksono, S.Kom, M.Kom  
NIP: 19860521 201212 1 001

Mengetahui

Ketua Jurusan Sistem Informasi



*Handwritten signature*  
Herman Tolle, Dr. Eng., S.T, M.T  
NIP: 19740823 200012 1 001



## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa skripsi yang berjudul "PENGEMBANGAN APLIKASI LAYANAN PENGADUAN MASYARAKAT UNTUK DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA PASURUAN BERBASIS ANDROID" 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 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 aturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 1 Agustus 2018



Hammer Zoelfagar Mahdias

NIM: 135150401111096

## KATA PENGANTAR

Dengan menyebut nama Allah SWT Yang Maha Pengasih dan Maha Penyayang karena berkat Rahmat, Hidayah dan Karunia-Nya kepada penulis sehingga dapat menyelesaikan penyusunan naskah skripsi ini dengan lancar. Shalawat serta salam semoga senantiasa terlimpah kepada junjungan Nabi Besar Muhammad SAW, kepada keluarga, serta para sahabatnya.

Penulisan naskah skripsi ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana. Judul yang penulis ajukan sebagai skripsi adalah "PENGEMBANGAN APLIKASI LAYANAN PENGADUAN MASYARAKAT UNTUK DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA PASURUAN BERBASIS ANDROID".

Dalam penyusunan dan penulisan naskah skripsi ini penulis mendapatkan banyak bantuan, bimbingan serta dukungan dari berbagai pihak baik secara moril maupun materil. Dalam kesempatan ini penulis ingin mengucapkan banyak terima kasih kepada yang terhormat:

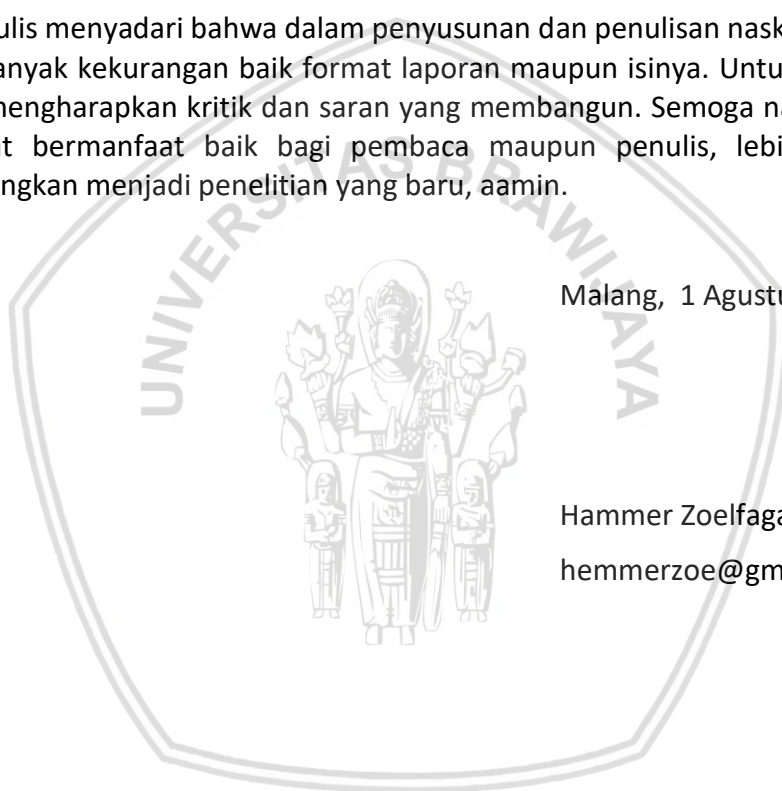
1. Bapak Himawat Aryadita, S.T, M.Sc selaku Dosen pembimbing pertama yang telah memberikan banyak arahan dan solusi dalam penyusunan skripsi ini, serta banyak hal baru yang penulis dapat selama menjadi bimbingan beliau.
2. Bapak Satrio Agung Wicaksono, S.Kom, M.Kom selaku pembimbing kedua yang telah memberikan banyak arahan dan solusi dalam penyusunan skripsi ini, serta banyak hal baru yang penulis dapat selama menjadi bimbingan beliau.
3. Bapak Ir. Suhari, MM selaku Kepala Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan yang telah memberikan izin untuk melakukan penelitian di instansinya.
4. Bapak Choirul Anam, S.Sos, MM selaku Kepala Bidang Pengelolaan Informasi Administrasi Kependudukan dan Pemanfaatan Data Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan selaku penanggung jawab yang telah memberikan banyak masukan selama penulis melakukan penelitian di instansinya.
5. Bapak Moch. Basori dan Ibu Djumariatini selaku orang tua penulis yang telah memberikan segenap dukungan yang sangat memotivasi bagi penulis baik berupa dukungan moral ataupun moril.

6. Sahabat kawan seperjuangan yang selalu memberikan semangat dan bantuan pikiran.
7. Nerissa Arviana Zagita dan R. Bagis Bagaskara yang telah memberikan waktu luang untuk menemani penulis dalam mengurus segala kepentingan skripsi
8. Teman – teman Sistem Informasi yang selalu memberikan semangat, dorongan, dan bantuan pikiran.
9. Seluruh pihak yang telah membantu kelancaran penulisan skripsi yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan dan penulisan naskah skripsi ini masih banyak kekurangan baik format laporan maupun isinya. Untuk itu penulis sangat mengharapkan kritik dan saran yang membangun. Semoga naskah skripsi ini dapat bermanfaat baik bagi pembaca maupun penulis, lebih-lebih bisa dikembangkan menjadi penelitian yang baru, aamin.

Malang, 1 Agustus 2018

Hammer Zoelfagar Mahdias  
hemmerzoe@gmail.com



## ABSTRAK

Dinas Kependudukan dan Pencatatan Sipil (Dispendukcapil) Kota Pasuruan merupakan unsur pelaksanaan Pemerintah Daerah Kota Pasuruan dalam memberikan pelayanan di bidang Kependudukan dan Catatan Sipil kepada Masyarakat Kota Pasuruan yang dituntut untuk meningkatkan pelayanan salah satunya memberikan kenyamanan kepada masyarakat dalam menyampaikan pendapat maupun keluhan yang dialami. Selama ini penanganan keluhan masyarakat di Dispendukcapil Kota Pasuruan masih menerapkan secara manual yang banyak memakan waktu, tenaga dan biaya karena masyarakat Kota Pasuruan diharuskan mendatangi Dispendukcapil Kota Pasuruan dalam menyampaikan keluhannya, sedangkan pengelolaan pengaduan oleh Petugas Administrasi di Dispendukcapil Kota Pasuruan harus mencatat setiap aduan kemudian dibuatkan laporan pengaduan yang dapat terjadinya *Human Error* karena kelalaiannya dapat menyebabkan pengaduan tersebut tidak dapat diproses. Berdasarkan permasalahan tersebut maka diperlukan aplikasi Layanan Pengaduan Masyarakat berbasis Android agar dalam menyampaikan keluhan tanpa harus datang ke Dispendukcapil Kota Pasuruan, maupun disediakannya sistem informasi pengelolaan pengaduan untuk Petugas Administrasi. Pengembangan sistem ini menggunakan metode *waterfall model* dimana sistem diuji menggunakan metode *Validation Testing* dan *User Acceptance Testing*. Dari hasil penelitian ini menghasilkan analisis kebutuhan, rancangan, implementasi dan pengujian sistem. Hasil pengujian pada sistem ini menunjukkan bahwa sistem berjalan sesuai dengan harapan dan mampu mempermudah laporan pengaduan bagi Masyarakat Kota Pasuruan ataupun pengelolaan pengaduan bagi Petugas Administrasi.

**Kata kunci:** *E-Complaint, Sistem Informasi, Waterfall, Pemrograman Android*

## ABSTRACT

*The Departement of Population and Civil Registration (Dispendukcapil) of Pasuruan City is an element of the implementation of the Regional Government of Pasuruan City in providing services in the field of Population and Civil Registry to the people of Pasuruan City which is required to improve the services, one of them is provide comfort to the community in expressing opinions or complaints experienced. So far, the handling of public complaints in Dispendukcapil Pasuruan still manually apply a lot of time, energy and cost because the people of Pasuruan City are required to visit Dispendukcapil Pasuruan in complaining, while the management complaints by the Administration Officer in Dispendukcapil Pasuruan should record each complaint and then made a complaint report that Human Error may cause because its negligence may cause the complaint to be unprocessable. Based on the problem, it is necessary an application of Complaints Service Society based on Android in order to submit a complaint without having to come to Dispendukcapil Pasuruan, as well as providing a complaint management information system for Administration Officer. The development of this system using waterfall model where the system is tested using Validation Testing method and User Acceptance Testing. From the results of this study produce needs analysis, design, implementation and testing system. Test results on this system indicate that the system is running in accordance with expectations and able to facilitate reports of complaints for Pasuruan City People or management of complaints for Administration Officers.*

**Keywords:** *E-Complaint, Information System, Waterfall, Android Programming*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvii
DAFTAR LAMPIRAN .....	xxi
<b>BAB 1</b> Pendahuluan.....	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	4
<b>BAB 2</b> LANDASAN KEPUSTAKAAN .....	<b>6</b>
2.1 Kajian Pustaka .....	6
2.2 Pengertian Sistem Informasi.....	7
2.3 Pengertian Pengaduan.....	7
2.4 Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.....	8
2.4.1 Visi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	8
2.4.2 Misi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	8
2.4.3 Struktur Organisasi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan .....	9
2.5 <i>Business Process Modeling Notation</i> (BPMN) .....	9
2.6 <i>Software Development Life Cycle</i> (SDLC) .....	14
2.6.1 Analisis Kebutuhan .....	15
2.6.1.1 Wawancara .....	16
2.6.1.2 Observasi .....	16
2.6.1.3 Kebutuhan Fungsional .....	16





2.6.1.4	Kebutuhan Non – Fungsional.....	17
2.6.1.5	Use Case Diagram .....	17
2.6.1.6	Use Case Description .....	19
2.6.1.7	Activity Diagram.....	20
2.6.2	Desain Sistem.....	22
2.6.2.1	Sequence Diagram .....	22
2.6.2.2	Class Diagram.....	24
2.6.3	Implementasi Sistem .....	26
2.6.4	Pengujian Sistem.....	26
2.6.4.1	Validation Testing .....	27
2.6.4.2	User Acceptance Testing .....	27
2.6.4.3	Likert Scale .....	28
2.7	Android .....	28
2.8	Web Service.....	29
2.9	RESTful Web Service.....	29
2.10	JSON .....	30
<b>BAB 3</b>	<b>METODOLOGI .....</b>	<b>32</b>
3.1	Identifikasi Masalah .....	32
3.2	Studi Literatur .....	33
3.3	Analisis Kebutuhan .....	33
3.4	Perancangan .....	33
3.5	Implementasi .....	34
3.6	Pengujian Dan Analisis Hasil Pengujian .....	34
3.7	Pengambilan Keputusan dan Saran .....	34
<b>BAB 4</b>	<b>ANALISIS KEBUTUHAN .....</b>	<b>36</b>
4.1	Pemodelan Proses Bisnis .....	36
4.1.1	Identifikasi Proses Bisnis AS-IS.....	36
4.1.1.1	Proses Masyarakat Kota Pasuruan membuat Pengaduan... 36	
4.1.1.2	Proses Memberikan Laporan Pengaduan oleh Petugas Administrasi .....	37
4.1.2	Analisis Permasalahan .....	37
4.1.3	Pemodelan Proses Layanan To-Be.....	38
4.1.3.1	Proses Masyarakat Kota Pasuruan membuat Pengaduan... 39	

4.1.3.2	Proses Memberikan Laporan Pengaduan oleh Petugas Administrasi .....	39
4.1.4	Analisis Proses Bisnis .....	40
4.1.5	Analisis Activity Proses Bisnis .....	41
4.2	Analisis Persyaratan .....	41
4.2.1	Identifikasi Pemangku Kepentingan .....	42
4.2.1.1	Tipe Pengguna dan Pemangku Kepentingan .....	42
4.2.1.2	Identifikasi Perwakilan Pemangku kepentingan.....	43
4.2.1.3	Kebutuhan Pemangku Kepentingan .....	43
4.2.2	Identifikasi Fitur .....	45
4.3	Persyaratan Deklaratif .....	48
4.3.1	Persyaratan Fungsional.....	48
4.3.2	Persyaratan Non-Fungsional.....	51
4.4	Pemodelan <i>Use Case</i> .....	52
4.5	<i>Use Case Description</i> .....	53
4.5.1	<i>Use case description</i> Login.....	53
4.5.2	<i>Use case description</i> Registrasi.....	54
4.5.3	<i>Use case description</i> Melihat Daftar Pengaduan.....	55
4.5.4	<i>Use case description</i> Melaporkan Pengaduan.....	56
4.5.5	<i>Use case description</i> Melihat Layanan.....	56
4.5.6	<i>Use case description</i> Mengelola Data Pengguna.....	57
4.5.7	<i>Use case description</i> Mengelola Data Pengaduan.....	60
4.6	<i>Activity Diagram</i> .....	62
4.6.1	<i>Activity Diagram</i> Login.....	62
4.6.2	<i>Activity Diagram</i> Registrasi.....	62
4.6.3	<i>Activity Diagram</i> Melihat Daftar Pengaduan.....	63
4.6.4	<i>Activity Diagram</i> Melaporkan Pengaduan.....	64
4.6.5	<i>Activity Diagram</i> Melihat Layanan .....	65
4.6.6	<i>Activity Diagram</i> Mengelola Data Pengguna.....	65
4.6.7	<i>Activity Diagram</i> Mengelola Data Pengaduan.....	67
BAB 5 PERANCANGAN.....		68
5.1	Perancangan <i>Sequence Diagram</i> .....	68
5.1.1	<i>Sequence Diagram</i> Login .....	68



5.1.2	<i>Sequence Diagram</i> Registrasi .....	69
5.1.3	<i>Sequence Diagram</i> Melihat Daftar Pengaduan .....	71
5.1.4	<i>Sequence Diagram</i> Melaporkan Pengaduan .....	73
5.1.5	<i>Sequence Diagram</i> Melihat Layanan .....	75
5.1.6	<i>Sequence Diagram</i> Mengelola Data Pengguna .....	76
5.1.7	<i>Sequence Diagram</i> Mengelola Data Pengaduan .....	80
5.2	Perancangan <i>Class Diagram</i> .....	84
5.2.1	<i>Class Diagram</i> Berbasis Android.....	84
5.2.2	<i>Class Diagram</i> Berbasis <i>Website</i> .....	87
5.3	Perancangan Komunikasi Data .....	88
5.3.1	Login.....	88
5.3.2	Registrasi.....	89
5.3.3	Ubah Password .....	89
5.3.4	Melaporkan Pengaduan.....	90
5.3.5	Melihat Daftar Pengaduan.....	91
5.4	Pemetaan <i>Class Diagram</i> ke <i>Relational Model</i> .....	92
5.5	Normalisasi <i>Relational Model</i> .....	93
5.6	Perancangan Antarmuka Sistem.....	94
5.6.1	Perancangan Antarmuka Sistem berbasis Android .....	94
5.6.1.1	Halaman Login .....	94
5.6.1.2	Halaman Registrasi .....	94
5.6.1.3	Halaman Beranda .....	95
5.6.1.4	Halaman Tentang Kami.....	95
5.6.1.5	Halaman Layanan.....	95
5.6.1.6	Halaman <i>NavigationView</i> .....	96
5.6.1.7	Halaman Pengaduan .....	96
5.6.1.8	Halaman Daftar Pengaduan.....	97
5.6.1.9	Halaman Profil .....	97
5.6.1.10	Halaman Ubah Password.....	98
5.6.2	Perancangan Antarmuka Sistem berbasis <i>Website</i> .....	99
5.6.2.1	Halaman Login .....	99
5.6.2.2	Halaman Beranda .....	99
5.6.2.3	Halaman Data Pengguna .....	100

5.6.2.4 Halaman Data Pengaduan .....	100
5.6.2.5 Halaman Laporan Pengaduan.....	101
<b>BAB 6 IMPLEMENTASI .....</b>	<b>102</b>
6.1 Batasan Implementasi .....	102
6.2 Implementasi Basis Data.....	102
6.2.1 Tabel BIODATA_WNI.....	102
6.2.2 Tabel PENGGUNA.....	103
6.2.3 Tabel TOPIK_PENGADUAN .....	103
6.2.4 Tabel TUJUAN_PENGADUAN .....	104
6.2.5 Tabel KATEGORI_PENGADUAN.....	104
6.2.6 Tabel KOTAK_PESAN.....	105
6.2.7 Tabel SEQUENCE_COUNTER .....	105
6.3 Implementasi Program .....	106
6.3.1 Program Berbasis Android .....	106
6.3.1.1 Impelementasi Prosedur Mengirim Pengaduan.....	106
6.3.1.2 Implementasi Prosedur Menampilkan Daftar Pengaduan	109
6.3.2 Program Berbasis <i>Website</i> .....	111
6.3.2.1 Kelas Controlpengaduan.....	111
6.3.2.2 Kelas Controladmin.....	115
6.4 Implementasi Antarmuka .....	123
6.4.1 Antarmuka Berbasis Android .....	123
6.4.1.1 Halaman Login .....	123
6.4.1.2 Halaman Registrasi .....	124
6.4.1.3 Halaman Beranda .....	124
6.4.1.4 Halaman Tentang Kami.....	125
6.4.1.5 Halaman Layanan.....	125
6.4.1.6 Halaman NavigationView.....	126
6.4.1.7 Halaman Pengaduan.....	126
6.4.1.8 Halaman Daftar Pengaduan.....	127
6.4.1.9 Halaman Profil .....	127
6.4.1.10 Halaman Ubah Password.....	128
6.4.2 Antarmuka Berbasis <i>Website</i> .....	129
6.4.2.1 Halaman Login .....	129



6.4.2.2 Halaman Beranda .....	129
6.4.2.3 Halaman Data Pengguna .....	130
6.4.2.4 Halaman Data Pengaduan .....	130
6.4.2.5 Halaman Laporan Pengaduan.....	131
<b>BAB 7 PENGUJIAN DAN ANALISIS HASIL PENGUJIAN.....</b>	<b>132</b>
7.1 Pengujian .....	132
7.1.1 Validation Testing .....	132
7.1.1.1 Validation Testing berbasis Android.....	132
7.1.1.1.1 Pengujian pada Fungsional Registrasi.....	132
7.1.1.1.2 Pengujian pada Fungsional Melaporkan Pengaduan ..	134
7.1.1.1.3 Pengujian pada Fungsional Melihat Layanan .....	135
7.1.1.2 Validation Testing Berbasis Website .....	136
7.1.1.2.1 Pengujian pada Fungsional Mengelola Data Pengguna .....	136
7.1.1.2.2 Pengujian pada Fungsional Mengelola Data Pengaduan .....	140
7.1.2 User Acceptance Testing .....	142
7.1.2.1 Hasil Pengujian User Acceptance Testing.....	146
7.2 Analisis Hasil Pengujian.....	149
<b>BAB 8 PENUTUP .....</b>	<b>151</b>
8.1 Kesimpulan.....	151
8.2 Saran .....	151
<b>DAFTAR PUSTAKA.....</b>	<b>153</b>



## DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi.....	9
Gambar 2.2 Contoh BPMN Diagram .....	14
Gambar 2.3 Waterfall Model .....	15
Gambar 2.4 Gambar Kebutuhan Fungsional.....	17
Gambar 2.5 Contoh Use Case Diagram.....	19
Gambar 2.6 Contoh Use Case Deskripsi.....	19
Gambar 2.7 Contoh Activity Diagram .....	21
Gambar 2.8 Contoh Sequence Diagram.....	24
Gambar 2.9 Contoh Class Diagram .....	26
Gambar 2.10 Struktur Objek JSON.....	30
Gambar 2.11 Struktur Array JSON .....	30
Gambar 2.12 Struktur <i>Value</i> JSON .....	31
Gambar 2.13 Struktur <i>String</i> JSON.....	31
Gambar 2.14 Struktur <i>Number</i> JSON .....	31
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	32
Gambar 4.1 BPMN Masyarakat Kota Pasuruan Membuat Pengaduan Sistem Lama .....	37
Gambar 4.2 BPMN Memberikan Laporan Pengaduan Oleh Petugas Administrasi Sistem Lama .....	37
Gambar 4.3 BPMN Masyarakat Kota Pasuruan membuat Pengaduan Sistem Baru .....	39
Gambar 4.4 BPMN Memberikan Laporan Pengaduan oleh Petugas Administrasi Sistem Baru .....	40
Gambar 4.5 <i>Use Case</i> Diagram Sistem Layanan Pengaduan Masyarakat.....	52
Gambar 4.6 <i>Activity Diagram</i> Login .....	62
Gambar 4.7 <i>Activity Diagram</i> Registrasi .....	63
Gambar 4.8 <i>Activity Diagram</i> Melihat Daftar Pengaduan .....	64
Gambar 4.9 <i>Activity Diagram</i> Melaporkan Pengaduan .....	64
Gambar 4.10 <i>Activity Diagram</i> Melihat Layanan .....	65
Gambar 4.11 <i>Activity Diagram</i> Mengelola Data Pengguna .....	66
Gambar 4.12 <i>Activity Diagram</i> Mengelola Data Pengaduan.....	67
Gambar 5.1 <i>Sequence Diagram</i> Login.....	68



Gambar 5.2 <i>Sequence Diagram</i> Registrasi .....	70
Gambar 5.3 <i>Sequence Diagram</i> Melihat Daftar Pengaduan.....	72
Gambar 5.4 <i>Sequence Diagram</i> Melihat Pesan Balasan Pengaduan.....	73
Gambar 5.5 <i>Sequence Diagram</i> Melaporkan Pengaduan.....	74
Gambar 5.6 <i>Sequence Diagram</i> Melihat Layanan.....	75
Gambar 5.7 <i>Sequence Diagram</i> Mengelola Data Pengguna .....	76
Gambar 5.8 <i>Sequence Diagram</i> Tambah Pengguna .....	77
Gambar 5.9 <i>Sequence Diagram</i> Hapus Pengguna .....	78
Gambar 5.10 <i>Sequence Diagram</i> Ubah Status Akun Pengguna.....	79
Gambar 5.11 <i>Sequence Diagram</i> Ubah <i>Password</i> Pengguna .....	80
Gambar 5.12 <i>Sequence Diagram</i> Mengelola Data Pengaduan.....	81
Gambar 5.13 <i>Sequence Diagram</i> Balas Aduan.....	81
Gambar 5.14 <i>Sequence Diagram</i> Hapus Aduan.....	82
Gambar 5.15 <i>Sequence Diagram</i> Ubah Status Aduan .....	83
Gambar 5.16 <i>Sequence Diagram</i> Cetak Aduan.....	83
Gambar 5.17 <i>Class Diagram</i> Berbasis Android.....	85
Gambar 5.18 <i>Class Diagram</i> Berbasis <i>Website</i> .....	87
Gambar 5.19 Relational Model .....	92
Gambar 5.20 Normalisasi Relational Model .....	93
Gambar 5.21 Perancangan Antarmuka <i>Login</i> .....	94
Gambar 5.22 Perancangan Antarmuka Registrasi .....	94
Gambar 5.23 Perancangan Antarmuka Beranda .....	95
Gambar 5.24 Perancangan Antarmuka Tentang Kami .....	95
Gambar 5.25 Perancangan Antarmuka Layanan .....	96
Gambar 5.26 Perancangan Antarmuka <i>NavigationView</i> .....	96
Gambar 5.27 Perancangan Antarmuka Pengaduan Berbasis Android .....	97
Gambar 5.28 Perancangan Antarmuka Daftar Pengaduan Berbasis Android .....	97
Gambar 5.29 Perancangan Antarmuka Profil Berbasis Android.....	98
Gambar 5.30 Perancangan Antarmuka Ubah <i>Password</i> Berbasis Android .....	98
Gambar 5.31 Perancangan Antarmuka <i>Login</i> Berbasis <i>Website</i> .....	99
Gambar 5.32 Perancangan Antarmuka Beranda Berbasis <i>Website</i> .....	99
Gambar 5.33 Perancangan Antarmuka Data Pengguna Berbasis <i>Website</i> .....	100

Gambar 5.34 Perancangan Antarmuka Data Pengaduan Berbasis <i>Website</i> .....	100
Gambar 5.35 Perancangan Antarmuka Laporan Pengaduan Berbasis <i>Website</i> .....	101
Gambar 6.1 Implementasi Antarmuka <i>Login</i> Berbasis Android .....	123
Gambar 6.2 Implementasi Antarmuka Registrasi Berbasis Android.....	124
Gambar 6.3 Implementasi Antarmuka Beranda Berbasis Android.....	124
Gambar 6.4 Implementasi Antarmuka Tentang Kami Berbasis Android.....	125
Gambar 6.5 Implementasi Antarmuka Layanan Berbasis Android.....	125
Gambar 6.6 Implementasi Antarmuka NavigationView Berbasis Android.....	126
Gambar 6.7 Implementasi Antarmuka Pengaduan Berbasis Android .....	126
Gambar 6.8 Implementasi Antarmuka Daftar Pengaduan Berbasis Android .....	127
Gambar 6.9 Implementasi Antarmuka Profil Berbasis Android.....	127
Gambar 6.10 Implementasi Antarmuka Ubah <i>Password</i> Berbasis Android .....	128
Gambar 6.11 Implementasi Antarmuka Login Berbasis <i>Website</i> .....	129
Gambar 6.12 Implementasi Antarmuka Beranda Berbasis <i>Website</i> .....	129
Gambar 6.13 Implementasi Antarmuka Data Pengguna Berbasis <i>Website</i> .....	130
Gambar 6.14 Implementasi Antarmuka Data Pengaduan Berbasis <i>Website</i> .....	130
Gambar 6.15 Implementasi Antarmuka Laporan Pengaduan Berbasis <i>Website</i> .....	131



## DAFTAR TABEL

Tabel 2.1 Notasi BPMN .....	10
Tabel 2.2 Notasi-Notasi <i>Use Case</i> .....	17
Tabel 2.3 Notasi-Notasi <i>Activity Diagram</i> .....	20
Tabel 2.4 Notasi-Notasi <i>Sequence Diagram</i> .....	22
Tabel 2.5 Notasi-Notasi <i>Class Diagram</i> .....	24
Tabel 4.1 Analisis Permasalahan .....	38
Tabel 4.2 Analisis Proses Bisnis .....	40
Tabel 4.3 Analisis Activity Proses Bisnis .....	41
Tabel 4.4 Tipe Pengguna dan Pemangku Kepentingan.....	42
Tabel 4.5 Identifikasi Perwakilan Pemangku Kepentingan.....	43
Tabel 4.6 Kebutuhan Pemangku Kepentingan.....	43
Tabel 4.7 Identifikasi Fitur.....	46
Tabel 4.8 Spesifikasi Kebutuhan Fungsional Masyarakat Kota Pasuruan.....	48
Tabel 4.9 Spesifikasi Persyaratan Fungsional Petugas Administrasi.....	50
Tabel 4.10 Spesifikasi Kebutuhan Non-Fungsional .....	52
Tabel 4.11 <i>Use Case description</i> Login .....	53
Tabel 4.12 <i>Use Case description</i> Registrasi.....	54
Tabel 4.13 <i>Use Case description</i> Melihat Daftar Pengaduan.....	55
Tabel 4.14 <i>Use Case description</i> Melaporkan Pengaduan.....	56
Tabel 4.15 <i>Use Case description</i> Melihat Layanan.....	56
Tabel 4.16 <i>Use Case description</i> Mengelola Data Pengguna.....	57
Tabel 4.17 <i>Use Case description</i> Mengelola Data Pengaduan.....	60
Tabel 5.1 <i>Web Service</i> untuk Login .....	88
Tabel 5.2 <i>Web Service</i> untuk Registrasi .....	89
Tabel 5.3 <i>Web Service</i> untuk Ubah Password.....	89
Tabel 5.4 <i>Web Service</i> untuk Melaporkan Pengaduan .....	90
Tabel 5.5 <i>Web Service</i> untuk Melihat Daftar Pengaduan .....	91
Tabel 6.1 Implementasi Basis Data Tabel BIODATA_WNI .....	102
Tabel 6.2 Implementasi Basis Data Tabel PENGGUNA .....	103
Tabel 6.3 Implementasi Basis Data Tabel TOPIK_PENGADUAN .....	103
Tabel 6.4 Implementasi Basis Data Tabel TUJUAN_PENGADUAN.....	104

Tabel 6.5 Implementasi Basis Data Tabel KATEGORI_PENGADUAN .....	104
Tabel 6.6 Implementasi Basis Data Tabel KOTAK_PESAN.....	105
Tabel 6.7 Implementasi Basis Data Tabel SEQUENCE_COUNTER.....	105
Tabel 6.8 Kode Program Fungsi KrimPesan .....	106
Tabel 6.9 Penjelasan Kode Program Fungsi krimPesan .....	107
Tabel 6.10 Kode Program Fungsi alert_pesanberhasil .....	108
Tabel 6.11 Penjelasan Kode Program Fungsi alert_pesanberhasil .....	109
Tabel 6.12 Kode Program Fungsi DataAduan .....	109
Tabel 6.13 Penjelasan Kode Program Fungsi DataAduan .....	110
Tabel 6.14 Kode Program Fungsi getHalamanProsesAduan.....	111
Tabel 6.15 Penjelasan Kode Program Fungsi getHalamanProsesAduan .....	112
Tabel 6.16 Kode Program Fungsi getUpdateResponPengaduan .....	113
Tabel 6.17 Penjelasan Kode Program Fungsi getUpdateResponPengaduan....	113
Tabel 6.18 Kode Program Fungsi getStatusAduan.....	114
Tabel 6.19 Penjelasan Kode Program Fungsi getStatusAduan .....	114
Tabel 6.20 Kode Program Fungsi getHapusAduan.....	114
Tabel 6.21 Penjelasan Kode Program Fungsi getHapusAduan .....	115
Tabel 6.22 Kode Program Fungsi index.....	115
Tabel 6.23 Penjelasan Kode Program Fungsi index .....	115
Tabel 6.24 Kode Program Fungsi getHalamanBeranda .....	116
Tabel 6.25 Penjelasan Kode Program Fungsi getHalamanBeranda.....	116
Tabel 6.26 Kode Program Fungsi getHalamanPegguna .....	117
Tabel 6.27 Penjelasan Kode Program Fungsi getHalamanPegguna .....	117
Tabel 6.28 Kode Program Fungsi getDataLogin .....	118
Tabel 6.29 Penjelasan Kode Program Fungsi getDataLogin.....	118
Tabel 6.30 Kode Program Fungsi getTambahPegguna .....	119
Tabel 6.31 Penjelasan Kode Program Fungsi getTambahPegguna.....	119
Tabel 6.32 Kode Program Fungsi getStatusPegguna .....	120
Tabel 6.33 Penjelasan Kode Program Fungsi getStatusPegguna.....	120
Tabel 6.34 Kode Program Fungsi getHapusPegguna .....	121
Tabel 6.35 Penjelasan Kode Program Fungsi getHapusPegguna.....	121
Tabel 6.36 Kode Program Fungsi getUpdatePasswordPegguna.....	122



Tabel 6.37 Penjelasan Kode Program Fungsi <code>getUpdatePasswordPengguna</code> ....	122
Tabel 6.38 Kode Program Fungsi <code>logout</code> .....	123
Tabel 6.39 Penjelasan Kode Program Fungsi <code>logout</code> .....	123
Tabel 7.1 Pengujian Registrasi .....	132
Tabel 7.2 Pengujian Registrasi <i>Alternative Flows</i> : Data Registrasi salah .....	133
Tabel 7.3 Pengujian Registrasi <i>Alternative Flows</i> : Data Registrasi tidak lengkap .....	134
Tabel 7.4 Pengujian Melaporkan Pengaduan .....	134
Tabel 7.5 Pengujian Melaporkan Pengaduan <i>Alternative Flows</i> : Data laporan pengaduan tidak lengkap .....	135
Tabel 7.6 Pengujian Melihat Layanan .....	135
Tabel 7.7 Pengujian Mengelola Data Pengguna .....	136
Tabel 7.8 Pengujian Mengelola Data Pengguna <i>Alternative Flows</i> : Data Diri Pengguna Salah .....	136
Tabel 7.9 Pengujian Mengelola Data Pengguna <i>Alternative Flows</i> : NIK / No. HP Pengguna Telah Terdaftar .....	136
Tabel 7.10 Pengujian Mengelola Data Pengguna <i>Alternative Flows</i> : Gagal Ubah <i>Password</i> Akun Pengguna .....	137
Tabel 7.11 Pengujian Mengelola Data Pengguna <i>Subflow</i> : Tambah Pengguna	138
Tabel 7.12 Pengujian Mengelola Data Pengguna <i>Subflow</i> : Hapus Pengguna ...	138
Tabel 7.13 Pengujian Mengelola Data Pengguna <i>Subflow</i> : Ubah Status Akun Pengguna.....	139
Tabel 7.14 Pengujian Mengelola Data Pengguna <i>Subflow</i> : Ubah <i>Password</i> Pengguna.....	138
Tabel 7.15 Pengujian Mengelola Data Pengaduan .....	140
Tabel 7.16 Pengujian Mengelola Data Pengaduan <i>Alternative Flows</i> : Gagal Hapus Aduan .....	140
Tabel 7.17 Pengujian Mengelola Data Pengaduan <i>Subflow</i> : Balas Aduan .....	140
Tabel 7.18 Pengujian Mengelola Data Pengaduan <i>Subflow</i> : Hapus Aduan.....	141
Tabel 7.19 Pengujian Mengelola Data Pengaduan <i>Subflow</i> : Ubah Status Aduan .....	141
Tabel 7.20 Pengujian Mengelola Data Pengaduan <i>Subflow</i> : Cetak Aduan.....	142
Tabel 7.21 <i>Stakeholder</i> Penguji <i>User Acceptance Testing</i> .....	143
Tabel 7.22 Skenario yang harus dikerjakan Petugas Administrasi .....	143
Tabel 7.23 Skenario yang harus dikerjakan Masyarakat Kota Pasuruan .....	144

Tabel 7.24 Tabel Pertanyaan UAT untuk Petugas Administrasi..... 145  
Tabel 7.25 Tabel Pertanyaan UAT untuk Masyarakat Kota Pasuruan ..... 146  
Tabel 7.26 Tabel Skor Penilaian Jawaban Pertanyaan Pengujian UAT ..... 146  
Tabel 7.27 Interpretasi *Skor Likert* Sumber: Bertram (2007)..... 147  
Tabel 7.28 Hasil Pengujian UAT untuk Petugas Administrasi ..... 147  
Tabel 7.29 Hasil Pengujian UAT untuk Masyarakat Kota Pasuruan..... 148



## DAFTAR LAMPIRAN

LAMPIRAN A.....	155
LAMPIRAN B.....	156
LAMPIRAN C.....	157
LAMPIRAN D.....	158
LAMPIRAN E.....	160
LAMPIRAN F.....	161
LAMPIRAN G.....	163
LAMPIRAN H.....	164



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Pesatnya perkembangan teknologi informasi pada saat ini sangat cepat dan hampir menyeluruh disemua kalangan dan semua bidang. Penggunaan sistem informasi untuk membantu kinerja organisasi semakin dibutuhkan dikarenakan informasi sebagai salah satu sumber daya yang penting dalam manajemen modern. Dengan didukung oleh kecanggihan teknologi informasi telah memungkinkan pengembangan sistem informasi yang semakin handal berbasis website ataupun mobile. Di era sekarang hampir segala bidang telah memanfaatkan produk dari teknologi informasi. Salah satunya Bidang Pelayanan Publik.

Kurang puasnya pelayanan publik yang diterima oleh masyarakat membuat masyarakat mulai mengeluh dan melakukan pengaduan. Aduan masyarakat muncul sebagai wujud pernyataan atau ungkapan rasa kurang puas terhadap satu produk atau layanan, baik secara lisan maupun tertulis dari pelanggan internal ataupun eksternal. Masyarakat memperoleh hak dalam diberinya ruang untuk mengeluarkan pikiran secara lisan ataupun tulisan, dalam hal ini adalah keluhan atas penyelenggaraan pelayanan publik yang diberikan oleh pemerintah.

Pengelolaan pengaduan (keluhan) dari masyarakat untuk menjamin penyediaan pelayanan publik yang sesuai dengan asas-asas umum pemerintahan yang baik, dan agar terjaminnya perlindungan hak setiap warga Negara dari kemungkinan-kemungkinan pengingkaran atau penyalahgunaan wewenang dalam menyelenggarakan pelayanan publik. Menurut Schelling and Chris Topher, Ed (dalam Rahmayanty, 2010) pentingnya mengorganisir pengaduan karena pelanggan yang mengeluh adalah teman baik kami, karena mereka memberikan kesempatan kepada kami untuk meningkatkan pelayanan. Pemerintah dalam hal ini adalah orang yang diberi mandat oleh masyarakat sebagai penyedia pelayanan untuk memperbaiki pelayanannya, perlu dilakukan berbagai upaya diantaranya adalah penanganan pengaduan (keluhan) dari masyarakat.

Dinas Kependudukan dan Pencatatan Sipil (Dispendukcapil) Kota Pasuruan merupakan unsur pelaksanaan pemerintah Daerah Kota Pasuruan dalam memberikan pelayanan kepada masyarakat Kota Pasuruan di bidang Kependudukan, Catatan Sipil dan Mobilitas Penduduk. Penyelenggaraan yang baik menuntut agar Aparatur Negara Bekerja lebih baik dalam suatu pemerintahan untuk meningkatkan pelayanan terhadap masyarakat salah satunya memberikan kenyamanan kepada masyarakat dalam menyampaikan pendapat maupun keluhan yang mereka alami.

Selama ini penanganan pengaduan (keluhan) masyarakat khususnya di Dinas Kependudukan dan Pencatatan sipil belum maksimal walaupun sudah menerapkan sistem pengaduan, namun sistem pengaduan masyarakat yang

diterapkan masih secara manual dengan mengharuskan masyarakat mendatangi dinas terkait. Dengan sistem pengaduan manual akan memakan banyak waktu, tenaga, dan biaya karena masyarakat Kota Pasuruan harus datang pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dengan menemui petugas administrasi kemudian akan dilakukan pendataan tentang pengaduan yang ingin disampaikan. Setelah menerima pengaduan, bagian petugas administrasi juga masih belum maksimal dalam mengelola pengaduan salah satunya pembuatan laporan pengaduan karena harus mencatat setiap aduan dan melakukan pendataan sebagai dokumentasi dalam pelaporan pengaduan dari masyarakat kemudian akan membuat laporan pengaduan yang akan diserahkan pada bidang yang dituju untuk di tindak lanjuti pengaduan tersebut. Sehingga dengan sistem pengaduan seperti itu hasil yang didapatkan memerlukan waktu yang lama dan dapat terjadinya kesalahan petugas (*Human Error*) administrasi karena kelalaiannya menyebabkan pengaduan tersebut tidak dapat diproses atau di tindak lanjuti.

Oleh Karena itu, dengan semakin maju dan canggihnya perkembangan teknologi saat ini, salah satu penerapan teknologi sistem informasi berupa aplikasi dapat dimanfaatkan sebagai sarana yang tepat untuk membantu pihak Dinas Kependudukan dan Pencatatan Sipil dalam menerapkan pelayanan pengaduan masyarakat menjadi lebih baik. Pemanfaatan aplikasi dengan dasar kecanggihan komputer, Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dapat menerapkan sebuah sistem informasi layanan pengaduan yang dapat diakses kapan dan dimana saja maupun dapat menampilkan informasi layanan-layanan yang tersedia pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan oleh Masyarakat Kota Pasuruan.

Teknologi yang sedang mendapatkan banyak perhatian dari masyarakat adalah android. Android merupakan OS (*Operating System*) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. Android adalah sistem operasi *mobile* berbasis linux yang mendukung era *open-source* (Hermawan, 2011). Dengan kata lain di jaman yang semakin canggih, android sangat dibutuhkan untuk mempermudah masyarakat menggunakan atau mengakses suatu sistem melalui *smartphone*. Aplikasi yang tersedia di Android sangat beragam, dan mudah diakses penggunaanya. Contoh pengguna Android untuk kegiatan sehari-hari, ketika seseorang ingin membaca buku, dia tidak perlu membawa buku tersebut, cukup menggunakan *e-book reader*.

Berdasarkan penjelasan tersebut, maka perlu dibuat sebuah sistem berbasis android yang dapat diakses dimana saja, kapan saja dan lebih praktis digunakan oleh masyarakat Kota Pasuruan, sedangkan Petugas Administrasi perlu disediakannya sistem informasi pengelolaan pengaduan salah satunya pembuatan laporan pengaduan Masyarakat Kota Pasuruan agar menjadi lebih mudah. Pengembangan sistem tersebut akan melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian (Sommerville, 2011). Sistem ini diharapkan mampu memberikan suatu media komunikasi antara masyarakat dan Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dalam

penginformasian layanan maupun adanya aduan atau keluhan yang dialami oleh Masyarakat Kota Pasuruan tanpa perlu harus datang ke Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang yang telah dijelaskan oleh penulis, maka diperoleh rumusan permasalahan sebagai berikut:

1. Bagaimana hasil analisis kebutuhan dan rancangan aplikasi Layanan Pengaduan Masyarakat pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan ?
2. Bagaimana hasil implementasi aplikasi Layanan Pengaduan Masyarakat pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan ?
3. Bagaimana hasil pengujian aplikasi Layanan Pengaduan Masyarakat pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan ?

## 1.3 Tujuan

Berdasarkan latar belakang dan permasalahan yang telah dijelaskan. Tujuan dari tugas akhir ini adalah menganalisis kebutuhan, merancang, mengimplementasikan dan melakukan pengujian sistem pada pengembangan aplikasi Layanan Pengaduan Masyarakat dengan teknologi berbasis Android yang digunakan oleh Masyarakat Kota Pasuruan agar dalam penyampaian pengaduan tanpa harus datang di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dan digunakan oleh Petugas Administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan agar membantu dalam melakukan pengelolaan pengaduan Masyarakat Kota Pasuruan menjadi lebih mudah.

## 1.4 Manfaat

Manfaat yang didapat dari penelitian yang dilakukan ini adalah :

1. Bagi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan mendapatkan aplikasi untuk meningkatkan pelayanan pengaduan kepada masyarakat Kota Pasuruan dan pengelolaan pengaduan yang dilakukan oleh Petugas Administrasi menjadi lebih mudah.
2. Bagi penulis yaitu mampu mengaplikasikan ilmu yang didapat selama mengikuti perkuliahan di Jurusan Sistem Informasi Universitas Brawijaya.

## 1.5 Batasan masalah

Penelitian tugas ini memiliki batasan – batasan yang digunakan sebagai pedoman dalam melakukan penelitian sebagai berikut :

1. Teknik pengujian lunak yang digunakan dalam penelitian ini menggunakan teknik *Validation Testing* dan *User Acceptance Testing*.



2. Pada aplikasi ini ditujukan untuk masyarakat Kota Pasuruan yang telah memiliki KTP Region Kota Pasuruan.
3. Aplikasi ini berfokus pada keluhan *Attitudinal Complaint* dan *Service Related Complaint* pada Masyarakat Kota Pasuruan di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.
4. Pada aplikasi ini pengelolaan aduan dikelola oleh Petugas Administrasi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan berbasis *Website*.
5. Aplikasi ini berjalan pada mode portrait di *smartphone*.

## 1.6 Sistematika pembahasan

Sistematika penulisan memberikan gambaran dan penjelasan secara garis besar dari isi penelitian yang terdiri dari beberapa bab sebagai berikut:

### I. BAB 1 PENDAHULUAN

Dalam bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika laporan tugas akhir.

### II. BAB 2 LANDASAN KEPUSTAKAAN

Dalam bab ini menjelaskan tentang kajian pustaka dari literatur, dasar teori berupa buku-buku yang memperkuat penulisan tugas akhir ini dan referensi dari metode yang digunakan dalam pengembangan aplikasi mobile.

### III. BAB 3 METODOLOGI PENELITIAN

Dalam bab ini menjelaskan tentang metode dan langkah kerja yang dilakukan dalam penelitian, yang terdiri dari beberapa proses yaitu studi literatur, rekayasa kebutuhan, perancangan dan implementasi perangkat lunak, serta pengujian dan analisis.

### IV. BAB 4 ANALISIS KEBUTUHAN

Dalam bab ini menjelaskan tentang analisa kebutuhan pada aplikasi mobile layanan pengaduan masyarakat meliputi *use case diagram*, *use case scenario* dan *activity diagram*

### V. BAB 5 PERANCANGAN

Dalam bab ini menjelaskan tentang perancangan aplikasi yang akan dibuat meliputi perancangan *sequence diagram*, *class diagram*, antarmuka sistem dan perancangan basis data.

### VI. BAB 6 IMPLEMENTASI

Dalam bab ini menjelaskan tentang proses implementasi sistem yang dilakukan berdasarkan hasil rancangan sistem.

## VII. BAB 7 PENGUJIAN DAN ANALISIS HASIL PENGUJIAN

Dalam bab ini menjelaskan tentang pengujian (*testing*) terhadap sistem dan analisa hasil pengujian dari sistem yang telah dibangun.

## VIII. Penutup

Dalam bab ini menjelaskan tentang kesimpulan yang dapat diambil dari penelitian dan saran-saran untuk perbaikan dan pengembangan penelitian.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Kajian pustaka penelitian ini membahas penelitian sebelumnya yang meneliti tentang pengembangan aplikasi pengaduan online. Penelitian yang berkaitan dengan pengembangan tersebut dilakukan oleh Anofrizen (2017). Penelitian ini menjelaskan bahwa dalam proses pengaduan pelaksanaan Program Keluarga Harapan (PKH) oleh Dinas Sosisal dan Pemakaman Kota Pekanbaru dilakukan secara manual. Salah satu kendalanya pengaduan tidak tercatat dengan sempurna maka dikembangkan sebuah sistem informasi pengaduan masyarakat PKH sehingga mampu membantu kinerja Dinas Sosisal dan Pemakaman Kota Pekanbaru dalam mengelola pengaduan yang masuk dan mampu mengatasi permasalahan yang terjadi pada PKH Kota Pekanbaru.

Penelitian yang juga berkaitan tentang pengembangan aplikasi pengaduan online. Penelitian yang dilakukan oleh Jumardi dan Solichin (2016). Penelitian ini menjelaskan bahwa Pemerintah Kota Makassar memiliki masalah terhadap lingkungan salah satunya masalah sampah. Sampah merupakan konsekuensi kehidupan manusia dimana semakin tinggi jumlah penduduk maka semakin banyak sampah yang dihasilkan. Pada tingkat tertentu, jumlah sampah menjadi persoalan sosial yang rumit karena aromanya busuk yang sangat menyengat. Salah satu upaya dalam mengatasi persoalan sampah dengan mendorong partisipasi masyarakat dalam melakukan pengaduan sampah yang ada disekitarnya. Maka perlu dikembangkan aplikasi layanan pengaduan masyarakat dalam penyampaian keluhan terhadap masalah lingkungan khususnya persoalan sampah menjadi lebih mudah.

Selain itu, terdapat penelitian yang dilakukan oleh Sarker, Faruque dan Rahman (2015). Penelitian ini menjelaskan perbandingan berbagai model siklus hidup pengembangan perangkat lunak (SDLC). Beberapa model untuk proses tersebut, masing-masing menggambarkan pendekatan untuk berbagai kegiatan yang berlangsung selama proses. Ada beberapa model yang dijelaskan seperti : Waterfall, V, Incremental, RAD, Iterative dan Spiral Model. Setiap model akan dijelaskan bagaimana proses pengembangan perangkat lunak termasuk keuntungan dan kerugian dari masing – masing model. Sehingga penelitian ini dapat mempermudah pengembang perangkat lunak (*developers*) mengetahui dalam pemilihan model mana yang akan digunakan dalam pengembangan perangkat lunak yang sesuai pada kondisi tertentu (permintaan pelanggan atau konsumen)

Dari beberapa kajian teori yang dijelaskan, nantinya akan menjadi acuan bagi penulis dalam penelitian terkait “ **Pengembangan Aplikasi Layanan Pengaduan Masyarakat Untuk Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan Berbasis Android** ”

## 2.2 Pengertian Sistem Informasi

Sistem adalah suatu proses yang saling berkaitan yang memiliki 3 aspek yakni input, proses dan output. Sebuah sistem dikatakan baik ketika sistem tersebut bisa memberikan output yang diharapkan dari input yang direncanakan. Sistem informasi terdiri dari gabungan teknologi, fasilitas, orang-orang, media, pengendalian dan prosedur yang bertujuan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberikan notifikasi kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan dasar informasi untuk pengambilan keputusan yang cerdas (Jogiyanto, 1999).

Sistem Informasi merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 1999). Dengan adanya sistem informasi, beberapa komponen dalam perusahaan bisa saling membagi informasi, data dan kebutuhan dari proses bisnis secara cepat dan terintegrasi. Dengan begitu antara satu proses bisnis dengan proses bisnis lain bisa dengan cepat untuk mendapatkan informasi.

## 2.3 Pengertian Pengaduan

Pengaduan merupakan bentuk timbal balik dari konsumen yang berisi informasi yang unik dan berharga, sehingga organisasi dapat memperhatikan peningkatan kualitas dan manajemen resiko (Gorton, 2005). Pengaduan berawal dari masyarakat yang mengadu kepada pemerintah berupa saran, pertanyaan atau keluhan. Menurut Endar Sugiarto (1999) dalam Sutopo dan Suryanto (2003), dijelaskan bahwa "sumber-sumber keluhan pelanggan antara lain : Pelanggan internal (pegawai suatu instansi / organisasi) dan pelanggan eksternal (masyarakat)". Menurut Endar Sugiarto, terdapat 4 kategori keluhan pelanggan, yaitu :

a. *Mechanical Complaint* (Keluhan Mekanikal)

Merupakan suatu keluhan yang disampaikan oleh pelanggan sehubungan dengan tidak berfungsinya peralatan yang dibeli/disampaikan kepada pelanggan tersebut.

b. *Attitudinal Complaint* (Keluhan akibat sikap petugas pelayanan)

Merupakan keluhan pelanggan yang timbul karena sikap *negative* petugas pelayanan pada saat melayani pelanggan.

c. *Service Related Complaint* (Keluhan yang berhubungan dengan Pelayanan)

Merupakan suatu keluhan pelanggan karena hal yang berhubungan dengan pelayanan itu sendiri

d. *Unusual Complaint* (Keluhan yang aneh)

Merupakan keluhan pelanggan yang bagi petugas merupakan keanehan (tidak wajar / tidak umum).

Sehingga, Pengaduan merupakan elemen penting bagi organisasi. Pengaduan dapat memberikan nilai positif bagi organisasi dimana organisasi dapat mengetahui apa yang menjadi kelemahannya ataupun kekurangannya dalam memberikan pelayanan kepada masyarakat.

## **2.4 Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan**

Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan merupakan unsur pelaksanaan pemerintahan Daerah Kota Pasuruan dalam memberikan pelayanan kepada masyarakat Kota Pasuruan di bidang Kependudukan, Catatan Sipil, dan Mobilitas yang mempunyai tugas pokok kepala daerah dalam melaksanakan sebagian urusan Pemerintah dan Pembangunan di bidang Kependudukan dan Catatan Sipil sesuai dengan ketentuan dan perundang-undangan yang berlaku.

Dinas Kependudukan dan Pencatatan Sipil, dipimpin oleh seorang Kepala Dinas yang berada di bawah dan bertanggung jawab kepada Walikota melalui Sekretaris Daerah ini. Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan terletak di Jalan Pahlawan No.22, Pasuruan Jawa Timur.

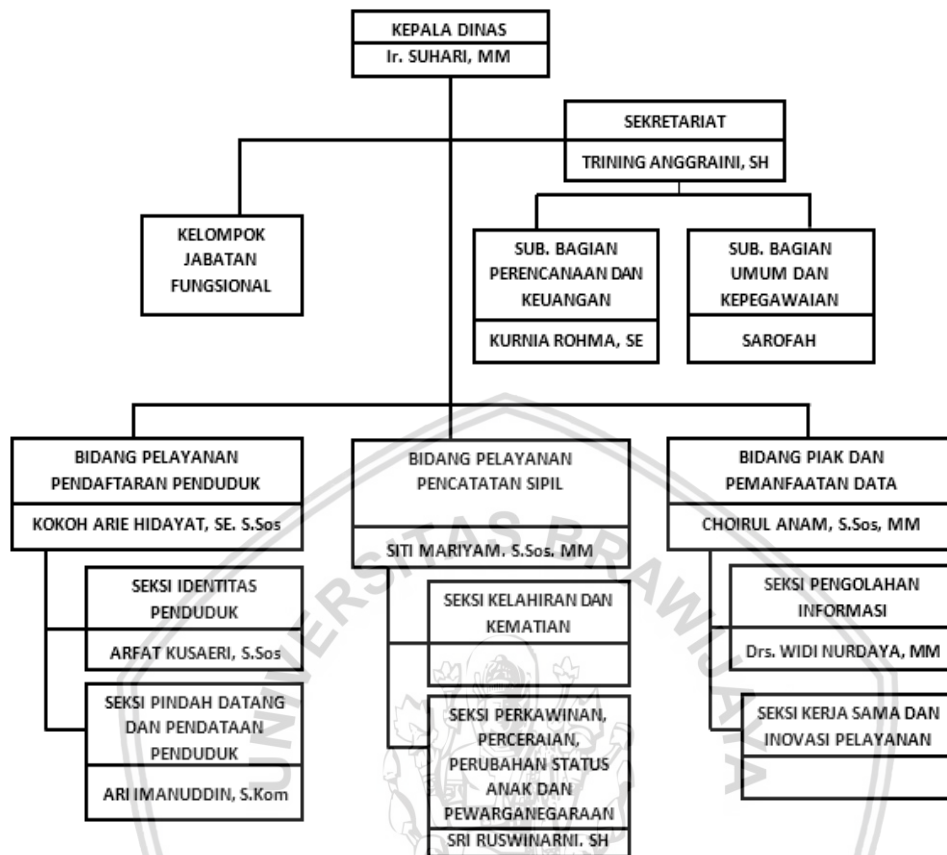
### **2.4.1 Visi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan**

Terwujudnya kualitas pelayanan informasi data, Administrasi Kependudukan dan Pencatatan Sipil yang akurat, tertib, mudah dan berkesinambungan.

### **2.4.2 Misi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan**

1. Meningkatkan profesionalisme aparatur dalam memberikan pelayanan prima administrasi kependudukan dan pencatatan sipil kepada masyarakat
2. Menyediakan saran prasana penunjang kegiatan pelayanan administrasi kependudukan dan pencatatan sipil
3. Meningkatkan tertib administrasi kependudukan dan pencatatan sipil guna perlindungan social dan penegakan hak-hak penduduk.

### 2.4.3 Struktur Organisasi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan



Gambar 2.1 Struktur Organisasi

### 2.5 Business Process Modeling Notation (BPMN)

BPMN adalah notasi grafis yang digunakan untuk memodelkan proses bisnis. BPMN biasa digunakan untuk menggambarkan diagram proses bisnis. Diagram proses bisnis merupakan diagram merepresentasikan aktivitas – aktivitas dan tugas – tugas dari suatu proses dan hubungan antar keduanya, Salah satu tujuan BPMN adalah untuk mengembangkan notasi yang dapat dimengerti oleh semua orang yang terlibat dalam proses bisnis. Pemodelan proses bisnis melibatkan berbagai kalangan, mulai dari pengguna bisnis, analiss bisnis, dan pemilik sampai dengan arsitek teknis dan pengembang / developer (Pant & Juric, 2008).

BPMN memiliki lima kategori elemen dasar yaitu (Object Management Group, 2011), yaitu dengan Flow Object, Data, Connecting Object, Swimlanes, Artifact.





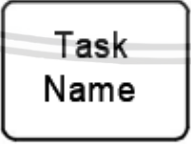

Terdapat empat cara untuk menghubungkan antar *Flow Object* atau informasi lainnya (Object Management Group, 2011) :

1. *Sequence Flows*
2. *Message Flows*

- 3. Association
- 4. Data Associations








Terdapat dua cara pengelompokan elemen pemodelan primer melalui "Swimlanes" (Object Management Group, 2011), yaitu dengan *Pool* dan *Lane*. Berikut adalah Tabel 2.1 yang menjelaskan tentang notasi BPMN.

**Tabel 2.1 Notasi BPMN**  
 Sumber: Object Management Group (2011)

Nama	Notasi	Deskripsi
<i>Start Event</i>		Memulai proses.
<i>Intermediate Events</i>		Menunjukkan terjadinya sesuatu (Event) di antara awal dan akhir proses. <i>Intermediate event</i> akan mempengaruhi aliran proses tetapi tidak memulai atau mengakhiri proses.
<i>End Events</i>		Mengakhiri proses.
<i>Activity</i>		<i>Activity</i> merupakan kegiatan atau pekerjaan yang dilakukan oleh perusahaan dalam sebuah proses.
<i>Task</i>		<i>Task</i> adalah bentuk <i>atomic</i> dari <i>Activity</i> yang disertakan dalam aliran proses. <i>Task</i> digunakan ketika pekerjaan dalam proses tidak dapat dipecah ke tingkat yang lebih detail.
<i>Service Task</i>		<i>Task</i> yang menggunakan beberapa jenis layanan, dapat berupa <i>web service</i> atau aplikasi otomatis.














Tabel 2.1 Notasi BPMN (Lanjutan)

Send Task		<p><i>Simple Task</i> yang dirancang untuk mengirim pesan kepada partisipan eksternal (relative terhadap proses). <i>Task</i> selesai ketika pesan selesai dikirim.</p>
Receive Task		<p><i>Simple Task</i> yang dirancang untuk menunggu pesan tiba dari partisipan eksternal (relative terhadap proses). <i>Task</i> selesai ketika pesan telah diterima.</p>
User Task		<p><i>Task</i> dimana manusia melakukan tugas dengan bantuan perangkat lunak aplikasi.</p>
Manual Task		<p><i>Task</i> yang dilakukan tanpa bantuan mesin eksekusi proses bisnis atau aplikasi lain.</p>
Business Rule Task		<p><i>Task</i> yang menyediakan mekanisme proses untuk memberikan masukan ke mesin aturan bisnis dan untuk mendapatkan output dari perhitungan yang disediakan oleh mesin aturan bisnis.</p>
Script Task		<p><i>Task</i> yang dijalankan oleh <i>business process engine</i>. Pemodel atau pelaksana mendefinisikan <i>script</i> dalam bahasa yang dapat diterjemahkan oleh mesin.</p>
Gateway		<p><i>Gateway</i> digunakan untuk mengontrol konvergensi dan divergensi dari <i>sequence flow</i> dalam proses dan dalam <i>Choreography</i>.</p>



Tabel 2.1 Notasi BPMN (Lanjutan)

<p><i>Gateway Control Types</i></p>	<p>Exclusive  or </p> <p>Event-Based  </p> <p>Parallel Event-Based </p> <p>Inclusive </p> <p>Complex </p> <p>Parallel </p>	<ul style="list-style-type: none"> <li>• <i>Exclusive decision</i> merging digunakan untuk membuat jalur alternative dalam aliran proses.</li> <li>• <i>Inclusive decision</i> digunakan untuk membuat jalur alternative parallel.</li> <li>• <i>Event-Based</i> merupakan titik percabangan dalam proses dimana jalur alternative yang mengikuti <i>gateway</i> didasarkan pada <i>event</i> yang terjadi.</li> <li>• <i>Parallel</i> digunakan untuk menggabungkan arus parallel dan membuat arus.</li> <li>• <i>Complex</i> digunakan untuk mensinkronisasi kejadian yang kompleks</li> </ul>
<p><i>Sequence Flow</i></p>		<p><i>Sequence Flow</i> digunakan untuk menunjukkan urutan <i>Flow elements</i> dalam sebuah proses atau <i>Choreography</i>. Setiap <i>Sequence Flow</i> hanya memiliki satu sumber dan satu target.</p>
<p><i>Conditional Flow</i></p>		<p><i>Sequence Flow</i> yang mendefinisikan ekspresi kondisi, aliran akan berjalan hanya jika kondisi bernilai <i>true</i>.</p>
<p><i>Default Flow</i></p>		<p><i>Sequence Flow</i> yang digunakan hanya ketika semua <i>conditional flow</i> / aliran kondisi yang keluar bernilai <i>false/not true</i> pada saat <i>runtime</i>.</p>

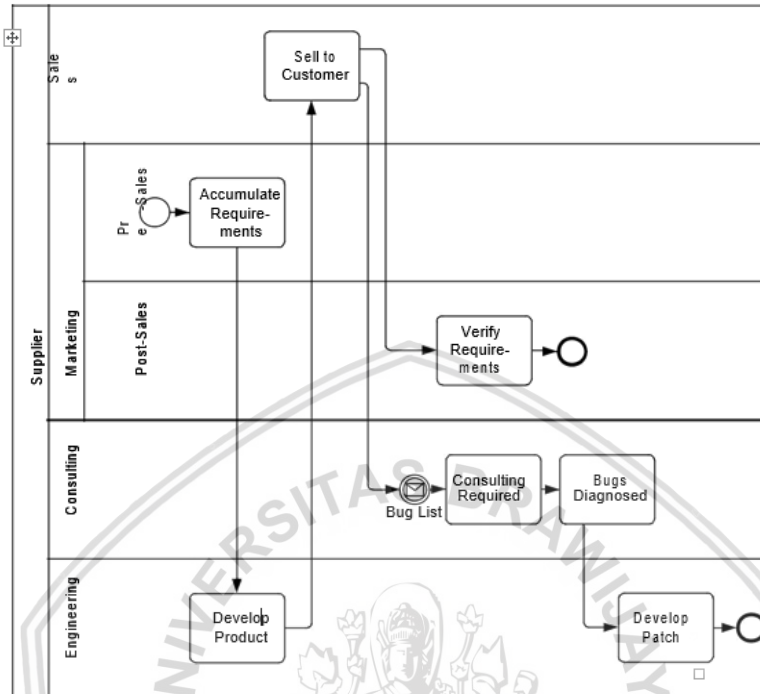


Tabel 2.1 Notasi BPMN (Lanjutan)

Message Flow		Digunakan untuk menunjukkan aliran pesan antara dua partisipan yang siap untuk mengirim dan menerima.
Association / Asosiasi		Digunakan untuk menghubungkan informasi dan artifact dengan flow object
Pool		Pool adalah representasi grafis dari partisipan dalam Collaboration.
Lane		Lane adalah sebuah sub-partisi dalam sebuah proses, kadang-kadang dalam Pool, dan akan memanjang sejauh panjang proses baik secara vertical maupun horizontal. Lane digunakan untuk mengatur dan mengkategorikan aktivitas.
Message		Digunakan untuk menggambarkan isi komunikasi antar dua partisipan.
Group		Digunakan untuk mengelompokkan elemen grafis yang berada dalam kategori yang sama. Jenis pengelompokkan tidak mempengaruhi sequence flow dalam grup. Nama kategori muncul dalam diagram sebagai label grup.
Text Annotation (dihubungkan dengan asosiasi)		Memberikan informasi teks tambahan untuk pembaca diagram BPMN



Berikut adalah gambar 2.1 merupakan contoh diagram BPMN yang membahas tentang hubungan antar departemen dan tugas-tugasnya yang menjelaskan tentang notasi BPMN.

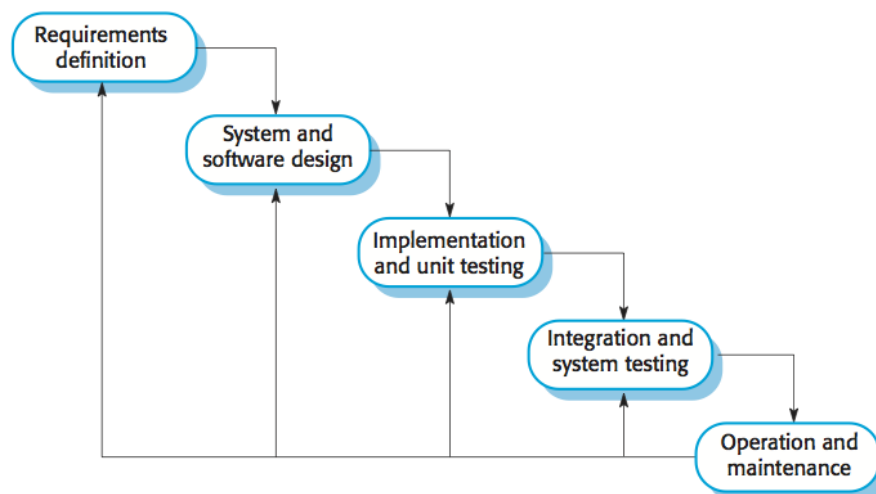


**Gambar 2.2 Contoh BPMN Diagram**  
 Sumber: Object Management Group (2011)

## 2.6 Software Development Life Cycle (SDLC)

*Software Development Life Cycle (SDLC)* merupakan metode pengembangan sistem tradisional yang digunakan sebagian besar organisasi saat ini. SDLC adalah kerangka kerja *framework* yang terstruktur yang berisi proses-proses berurutan dimana sistem informasi dikembangkan (Rainer & Turban, 2009).

*Waterfall Model* merupakan sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum di kerjakan. *Waterfall Model* memiliki tahapan-tahapan utama yang langsung mencerminkan dasar pengembangan sistem seperti analisis dan definisi persyaratan, desain sistem, perancangan perangkat lunak, implementasi dan pengujian sistem (Sommerville, 2011).



**Gambar 2.3 Waterfall Model**

Sumber : Sommerville (2011)

Tahap-tahap utama dari *waterfall model* pada gambar 2.3 yang memetakan kegiatan-kegiatan pengembangan dasar. Pada penelitian ini dilakukan dari tahap “Analisis dan definisi persyaratan” sampai kepada tahap “Integrasi dan pengujian sistem” tanpa melakukan tahap akhir “Operasi dan pemeliharaan”. Pada sub-bab berikut ini akan dijelaskan mengenai tahapan dari model *waterfall*.

### 2.6.1 Analisis Kebutuhan

Analisis Kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun melalui observasi dan wawancara pada pihak terkait. Analisis diperlukan untuk dapat membangun aplikasi yang sesuai kebutuhan dan terstruktur. Proses ini dapat dilakukan dengan mendapatkan semua kebutuhan pembuatan aplikasi agar dapat menentukan sejauh mana aplikasi dapat mencapai tujuan dari pembuatan aplikasi.

Dalam pembuatan sebuah sistem atau perangkat lunak awalnya akan dijelaskan cara kerja dan fungsi-fungsi yang ada pada sistem pada gambaran umum sistem. Dari gambaran umum akan didapatkan beberapa kebutuhan dari sistem yang akan dibuat digambarkan dalam *use case diagram*. Penggunaan *use case* sangat penting dalam mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas perangkat lunak dari sudut pandang user.

Pada tahap ini akan menghasilkan kebutuhan fungsional dan non-fungsional dari sistem, *use case diagram*, dan *activity diagram*. Berikut ini merupakan penjelasan dari konsep wawancara, observasi, kebutuhan fungsional, kebutuhan non-fungsional maupun *use case diagram* dan *activity diagram* :

### 2.6.1.1 Wawancara

Wawancara merupakan suatu proses untuk memperoleh keterangan atau data dari tujuan penelitian dengan cara percakapan tanya jawab. Percakapan dilakukan secara dua pihak, yaitu pewawancara yang akan memberikan pertanyaan dan yang diwawancarai yang memberikan jawaban atas pertanyaan dengan menggunakan alat yang dinamakan panduan wawancara (Siregar, 2013). Wawancara digunakan sebagai teknik pengumpulan data apabila ingin melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti, dan apabila peneliti ingin mengetahui dari responden yang lebih mendalam dan jumlah responden sedikit. Teknik pengumpulan data dengan wawancara dapat dilakukan secara terstruktur maupun tidak terstruktur dan dapat dilakukan melalui tatap muka maupun dengan telepon (Sugiyono, 2016). Metode teknik wawancara yang digunakan pada penelitian ini sebagai berikut :

#### 1. Wawancara Tak Terstruktur

Wawancara tak terstruktur merupakan sebuah teknik wawancara yang bebas dimana peneliti tidak menggunakan pedoman wawancara yang telah tersusun secara sistematis dan lengkap untuk pengambilan data. Pedoman yang digunakan dalam wawancara yaitu hanyalah berupa garis-garis besar permasalahan yang akan ditanyakan. Dalam wawancara tak terstruktur, peneliti belum mengetahui secara pasti data apa saja yang akan diperoleh, sehingga peneliti lebih banyak mendengarkan apa yang diceritakan oleh responden.

### 2.6.1.2 Observasi

Menurut Kartono (dalam Basuki, 2006) observasi merupakan studi yang disengaja dan sistematis tentang fenomena social dan gejala-gejala psikis dengan jalan pengamatan dan pencatatan. Observasi yang dilakukan pada penelitian ini menurut (Sugiyono, 2016).

#### 1. Observasi yang tidak berstruktur

Observasi tidak berstruktur yaitu observasi dimana pengamat dalam melaksanakan observasinya dan melakukan pengamatan secara bebas.

### 2.6.1.3 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem menggambarkan apa dan bagaimana sistem yang harus dilakukan oleh sebuah sistem. Kebutuhan-kebutuhan tersebut bergantung pada tipe aplikasi yang akan dikembangkan, apa yang diinginkan oleh pengguna dari aplikasi yang dikembangkan, dan bagaimana pendekatan dari pemangku kepentingan ketika menentukan kebutuhan aplikasi. Ketika menjelaskan kebutuhan dari pemangku kepentingan atau pengguna, kebutuhan fungsional adalah salah satu cara untuk menjelaskan fungsi yang diberikan kepada pengguna yang bisa dipahami (Sommerville, 2011).

Section/ Requirement ID	Requirement Definition
4.5.2	Provide Project Planning Capability
SR4.5.2.0*	The system shall provide a project planning capability
SR4.5.2.0.1	The system shall allow authorized users to enter project plans and timelines
SR4.5.2.0.2	The system shall allow authorized users to review, change, or update project plans and timelines
* For this example, "F" is used to designate a Functional requirement. Other designations could be "DF" for Design Feature or "STRQ" for Stakeholder Request.	

**Gambar 2.4 Gambar Kebutuhan Fungsional**

### 2.6.1.4 Kebutuhan Non – Fungsional

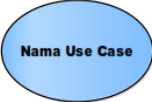
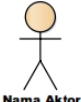

Kebutuhan non-fungsional adalah kebutuhan yang tidak langsung berkaitan dengan layanan yang diberikan oleh sistem kepada pengguna. Kebutuhan ini berkaitan dengan keandalan sistem, waktu proses dari sistem, dan keamanan sistem. Kebutuhan non-fungsional juga menjelaskan bagaimana sebuah sistem dikembangkan sesuai dengan keadaan atau kondisi eksternal dari sistem seperti kebutuhan sistem untuk merespon perangkat *input output*, bagaimana sistem bisa menampilkan data yang representative dan sesuai dengan keinginan pengguna, dan bagaimana sistem bisa menyesuaikan dengan sistem lain (Sommerville, 2011).

### 2.6.1.5 Use Case Diagram

Diagram *Use Case* adalah suatu pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. *Use Case* diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (S. & Shalahuddin, 2015). Berikut ini adalah Tabel 2.2 yang menjelaskan notasi-notasi yang ada pada Diagram *Use Case*:

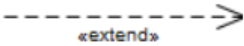

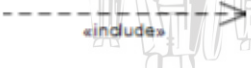
**Tabel 2.2 Notasi-Notasi Use Case**

Sumber: S. & Shalahuddin (2015)

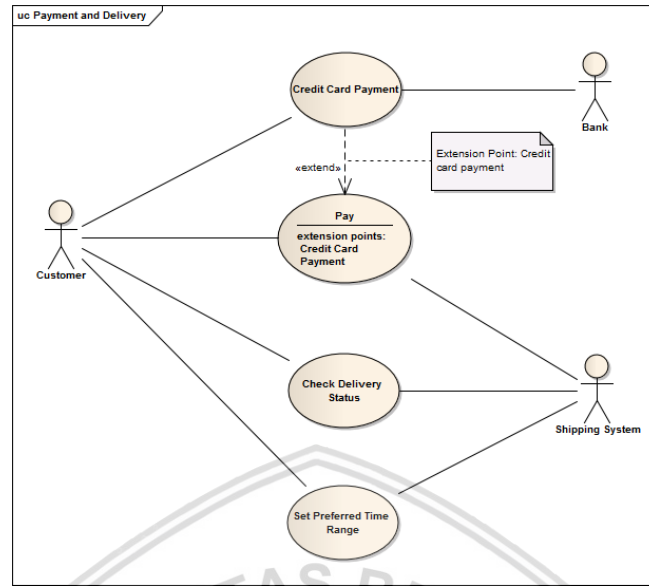
Nama	Notasi	Deskripsi
<i>Use Case</i>		Menggambarkan fungsionalitas yang dimiliki oleh sistem.
Aktor		Menggambarkan manusia, proses atau sistem lain yang berinteraksi dengan sistem.
<i>Association</i> / asosiasi		Menggambarkan komunikasi antara actor dan <i>use case</i> . Notasi digunakan ketika suatu actor ikut serta dalam



Tabel 2.2 Notasi-Notasi *Use Case* (Lanjutan)

		pelaksanaan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
<i>Extend</i> / Ekstensi		Menggambarkan relasi <i>use case</i> tambahan ke sebuah <i>use case</i> . <i>Use case</i> yang ditambahkan dapat berdiri sendiri meskipun tanpa <i>use case</i> tambahan tersebut. Pengarahan arah panah mengarah pada <i>use case</i> yang ditambahkan.
<i>Generalization</i> / Generalisasi		Menggambarkan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya. Pengarahan arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya.
<i>Include</i>		Menggambarkan relasi <i>use case</i> tambahan ke sebuah <i>use case</i> . <i>Use case</i> yang ditambahkan memerlukan <i>use case</i> tambahan tersebut untuk menjalankan fungsinya atau sebagai syarat dijalannkannya <i>use case</i> yang ditambahkan.

Berikut adalah gambar 2.4 merupakan contoh dari *Use Case Diagram*.



**Gambar 2.5 Contoh Use Case Diagram**

Sumber : Sommerville (2011)

### 2.6.1.6 Use Case Description

Menurut (Satzinger et al, 2010) *Use case description* merupakan deskripsi yang mencatat mengenai detail pemrosesan dari suatu *use case*. *Use case* memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan bisnis proses. Metode yang digunakan dalam mendeskripsikan *use case* yaitu *Fully developed description*.

*Fully developed description* merupakan metode yang paling formal mendokumentasikan sebuah *use case*. Meskipun memerlukan waktu lebih untuk mengerjakan, tetapi dapat meningkatkan kemungkinan akan pemahaman mengenai bisnis proses.

Use case Melihat Data Diri Mahasiswa	
Tujuan	Mengijinkan administrator untuk melihat data diri mahasiswa
Aktor	Administrator
Kondisi awal	Login tervalidasi dan valid
Skenario utama	<ol style="list-style-type: none"> <li>Administrator memilih menu data diri mahasiswa</li> <li>Sistem menampilkan daftar mahasiswa, untuk dilihat data dirinya</li> </ol>
Skenario alternatif	<ol style="list-style-type: none"> <li>Jika data diri mahasiswa yang dipilih masih kosong, maka sistem akan menunjukkan pesan "data diri masih kosong".</li> <li>Jika data diri mahasiswa yang dipilih masih belum lengkap, maka sistem akan menampilkan pesan "data diri masih belum lengkap".</li> </ol>
Kondisi akhir	Sistem menampilkan data diri mahasiswa sesuai dengan identitas mahasiswa yang dipilih.

**Gambar 2.6 Contoh Use Case Deskripsi**

Sumber: S. & Shalahuddin (2015)





### 2.6.1.7 Activity Diagram


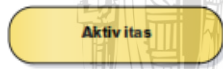

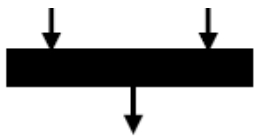
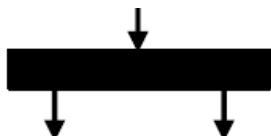
*Activity Diagram* atau aktifitas diagram berfungsi untuk menggambarkan *workflow* (aliran kerja) atau sebuah aktifitas sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity Diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor (S. & Shalahuddin, 2015). *Activity diagram* juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan sebuah proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan alur proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem (*User interface*) dimana setiap aktivitas memiliki sebuah rancangan tampilan antarmuka.
3. Rancangan pengujian dimana setiap aktifitas memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan antarmuka menu yang ditampilkan pada perangkat lunak.

Berikut ini merupakan notasi-notasi yang ada pada *Activity Diagram* :

**Tabel 2.3 Notasi-Notasi Activity Diagram**

Sumber: S. & Shalahuddin (2015)

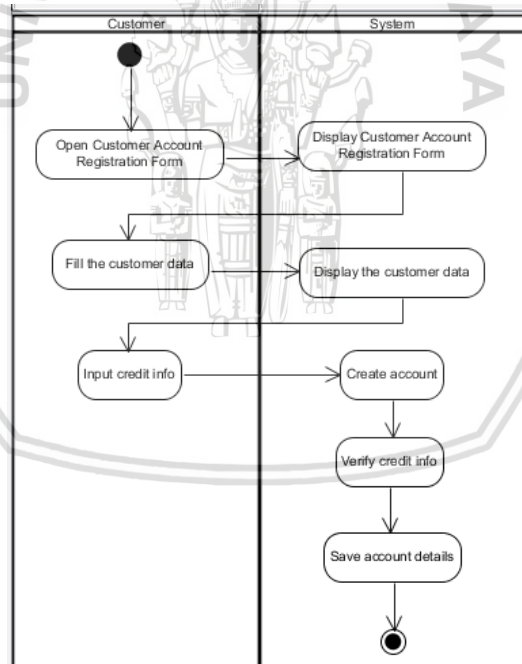
Nama	Notasi	Deskripsi
Status Awal		Menggambarkan status awal dari aktivitas sistem.
Aktivitas		Menggambarkan aktivitas yang dilakukan oleh sistem.
<i>Decision</i> / percabangan		Menggambarkan asosiasi percabangan dimana jika ada pilihan lebih dari satu pilihan aktivitas.
<i>Join</i> / penggabungan		Menggambarkan asosiasi penggabungan yang digunakan ketika ada lebih dari satu aktivitas digabungkan menjadi satu.
<i>Fork</i> / percabangan		Menggambarkan asosiasi percabangan yang digunakan ketika pemisah beberapa aliran konkuren dari satu aliran tunggal



**Tabel 2.3 Notasi-Notasi Activity Diagram (Lanjutan)**

Status Akhir		Menggambarkan status akhir yang dilakukan oleh sistem.
Swimlane		Menggambarkan dalam memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Berikut adalah gambar 2.7 merupakan contoh dari Activity Diagram.



**Gambar 2.7 Contoh Activity Diagram**

Sumber : Satzinger (2010)



## 2.6.2 Desain Sistem

Desain sistem atau perancangan sistem dilakukan setelah semua kebutuhan sistem telah didapatkan. Perancangan sistem menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Menurut (S. & Shalahuddin, 2015) UML (*Unified Modelling Language*) adalah salah satu standar bahasa visual yang banyak digunakan untuk mengidentifikasi *requirement*, membuat analisis desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak.





Dalam tahap ini akan menghasilkan *sequence diagram*, *class diagram*, perancangan antarmuka dan spesifikasi sistem. Berikut ini merupakan penjelasan dari *sequence diagram* dan *class diagram* :

### 2.6.2.1 Sequence Diagram

Diagram *Sequence* adalah menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan menerima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu maupun juga dibutuhkan untuk melihat skenario yang ada pada *use case* (S. & Shalahuddin, 2015). Berikut ini tabel 2.4 merupakan notasi-notasi yang ada pada Diagram *Sequence*:



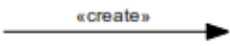
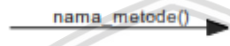
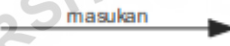
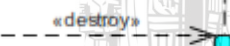
**Tabel 2.4 Notasi-Notasi *Sequence Diagram***

Sumber: S. & Shalahuddin (2015)

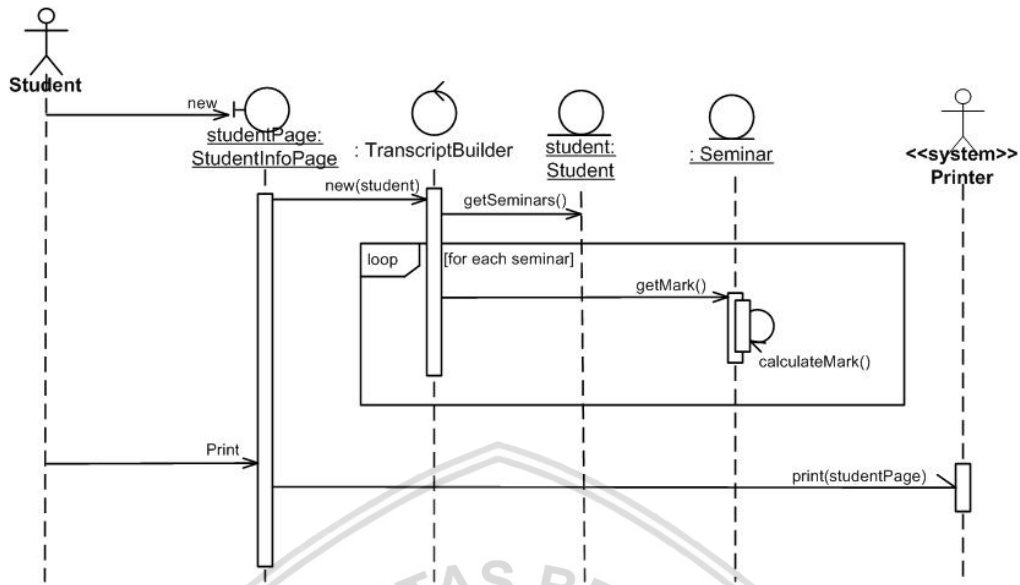
Nama	Notasi	Deskripsi
Aktor		Menggambarkan orang, proses, atau sistem lain yang berinteraksi dengan sistem.
<i>Lifeline</i>		Menggambarkan kehidupan suatu objek.
Boundary		Menggambarkan antarmuka dan interaksi satu atau lebih aktor dengan sistem
Controller		Menggambarkan perilaku mengatur, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani



Tabel 2.4 Notasi-Notasi *Sequence Diagram* (Lanjutan)

		Tugas utama dan mengontrol alur kerja suatu sistem
Entity		Menggambarkan informasi atau data yang harus disimpan oleh sistem
Waktu Aktif		Menggambarkan objek dalam keadaan aktif dan berinteraksi.
Pesan <i>create</i>		Menggambarkan suatu objek membuat objek lain.
Pesan <i>call</i>		Menggambarkan suatu objek memanggil operasi / method yang ada pada objek lain atau dirinya sendiri.
Pesan <i>send</i>		Menggambarkan suatu objek mengirimkan informasi ke objek lainnya.
Pesan <i>return</i>		Menggambarkan suatu objek telah menjalankan suatu operasi mengembalikan suatu kembalian objek tertentu.
Pesan <i>destroy</i>		Menggambarkan suatu objek mengakhiri hidup objek lain.

Berikut adalah gambar 2.8 merupakan contoh dari *Sequence Diagram*.



**Gambar 2.8 Contoh Sequence Diagram**


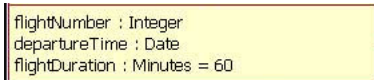
Sumber : Ambler (2004)

### 2.6.2.2 Class Diagram

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class Diagram* memiliki apa yang disebut atribut dan metode atau operasi (S. & Shalahuddin, 2015). Berikut ini tabel 2.5 merupakan notasi-notasi *Class Diagram* :

**Tabel 2.5 Notasi-Notasi Class Diagram**

Sumber: S. & Shalahuddin (2015)

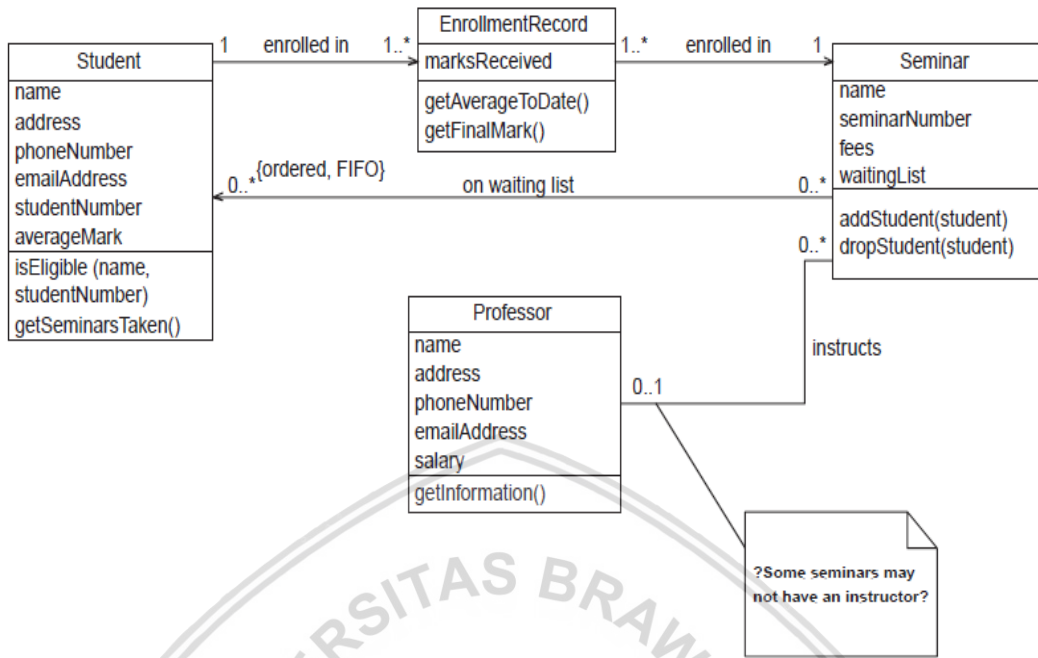
Nama	Notasi	Deskripsi
<i>Class / Kelas</i>		Menggambarkan kelas pada struktur sistem.
Atribut		Mendeskripsikan jangka nilai suatu hal yang dimiliki kelas tersebut dan sifat ini akan dimiliki semua objek hasil instansiasi dari kelas tersebut



Tabel 2.5 Notasi-Notasi *Class Diagram* (Lanjutan)

Operasi		Operasi yang dilakukan kelas tersebut dapat memiliki parameter dan nilai kembalian, bisa juga diatur tanpa keduanya.
Asosiasi tak berarah		Menggambarkan relasi antar kelas dengan makna umum yang tak berarah.
Asosiasi berarah		Menggambarkan relasi antar kelas dengan makna kelas tersebut digunakan oleh kelas lain.
Generalisasi		Menggambarkan relasi antar kelas dengan makna sebuah kelas merupakan kelas yang lebih umum daripada kelas yang menunjuk.
Kebergantungan		Menggambarkan relasi antar kelas dengan makna kebergantungan antar kelas.
<i>Aggregation</i> / Agregasi		Menggambarkan relasi antar kelas dengan makna hubungan yang menunjuk "bagian dari" yang dituju.
Komposisi		Menggambarkan relasi antar kelas dengan makna hubungan yang dituju "terdiri dari" yang menunjuk

Berikut adalah gambar 2.9 merupakan contoh dari *Class Diagram*.



**Gambar 2.9 Contoh Class Diagram**

Sumber : Ambler (2004)

### 2.6.3 Implementasi Sistem

Implementasi Sistem dilakukan setelah tahap perancangan selesai kemudian akan diimplementasikan ke dalam kode-kode bahasa pemrograman yang telah ditentukan. Tahap awal dari proses implementasi adalah melakukan penjabaran spesifikasi lingkungan penggunaan sistem atau perangkat lunak.

### 2.6.4 Pengujian Sistem

Pengujian sistem merupakan tahap terakhir dalam pengembangan sistem yang bertujuan sistem atau perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang diinginkan pada latar belakang pembuatan sistem atau perangkat lunak. Setiap unit dari program yang sudah ada kemudian diintegrasikan dan diuji sebagai satu kesatuan yang utuh untuk memastikan apakah kebutuhan sistem sudah terpenuhi. Proses pengujian memiliki dua spesifik, yaitu :

1. Untuk memastikan bahwa perangkat lunak sudah memenuhi seluruh persyaratan
2. Untuk mencari sebanyak mungkin kesalahan pada sistem atau hal – hal yang tidak sesuai dengan spesifikasi

Pada tahap ini dilakukan pengujian menggunakan pengujian *Black-Box Testing*. *Black-Box Testing* merupakan teknik pengujian yang hanya melibatkan

observasi output dari input nilai tertentu dan tidak ada percobaan untuk analisis kode program. *Black box testing* berfokus pada fungsi sebuah perangkat lunak (Agarwal, et al.,2010). Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkapkan kelas kesalahan daripada metode *white-box*. Pada penelitian ini akan menggunakan pengujian *validation testing* yang digunakan untuk menguji kebutuhan fungsional sedangkan *User Acceptance Testing* digunakan untuk memastikan apakah sistem telah memenuhi kriteria oleh pengguna atau customer.

#### **2.6.4.1 Validation Testing**

*Validation Testing* atau pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black Box*, karena tidak memerlukan untuk berkonsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan konformitas antara kinerja sistem dengan daftar kebutuhan (Indriati, 2010).

Cara Pengujian ini dengan melakukan, menjalankan dan mengeksekusi setiap modul kemudian dilakukan pengamatan pada hasil dari proses tersebut. Bisa juga dengan menyiapkan satu paket masukan valid dan tidak valid, kemudian memeriksa keluaran yang dihasilkan, apakah sesuai dengan apa yang diharapkan atau tidak. Apabila hasilnya sesuai, maka dapat dikatakan perangkat lunak dapat berjalan dengan baik.

#### **2.6.4.2 User Acceptance Testing**

*User Acceptance Testing* (UAT) merupakan pengujian yang menangani kebutuhan pengguna, *requirement*, dan *business process*. UAT dilakukan untuk mengetahui apakah suatu sistem telah memenuhi kriteria untuk diterima oleh pengguna atau *customer* (Hambling, 2013).

UAT biasanya dilakukan sebelum produk dihidupkan dan dilakukan setelah pengujian sistem. UAT dilakukan oleh *user* yang terlibat dengan membuat sistem yang dibuat. Terdapat beberapa poin yang perlu diperhatikan dalam membuat UAT (Softwaretestinghelp, 2007).

1. UAT bukan tentang halaman, *fields*, tombol. Asumsi tersebut dilakukan bahkan sebelum UAT dimulai. Semua hal mendasar itu seharusnya telah diuji dan bekerja dengan baik.



2. UAT adalah tentang entitas yang merupakan elemen utama dalam bisnis.
3. UAT juga merupakan bentuk pengujian pada inti sistem yang berarti ada kemungkinan bagus untuk mengidentifikasi beberapa bug pada fase ini juga.
4. UAT diklasifikasikan sebagai pengujian Alpha dan Beta, namun klasifikasi tersebut tidak begitu penting dalam konteks proyek pengembangan perangkat lunak biasa di industry berbasis layanan.
5. Sebagian besar waktu dalam proyek pengembangan perangkat lunak biasa, UAT dilakukan dilingkungan *quality assurance* jika tidak ada lingkungan pementasan atau UAT.

Dalam menentukan jumlah pengambilan berapa banyak *user* sebagai pengujian dalam penelitian ini, menggunakan teori menurut (Gay & Diehl, 1992). Bahwa sampel haruslah sebesar-besarnya. Pendapat ini mengasumsikan bahwa semakin banyak sampel yang diambil, maka akan semakin representatif dan hasilnya dapat digeneralisir. Penelitian deskriptif, minimum sampel 10% dari populasi.

#### 2.6.4.3 Likert Scale

Pada *Likert Scale* koresponden diminta untuk memberikan tingkatan skala ordinal. *Likert scale* ditunjukkan dalam 5 tingkatan dari "Sangat Tidak Setuju" sampai dengan "Sangat Setuju" seperti berikut : (Bertram, 2007)

1. Sangat Tidak Setuju, yang berarti aplikasi jauh diluar ekspektasi dari responden yang terkait pernyataan yang diberikan.
2. Tidak Setuju, yang berarti pernyataan yang diberikan tidak sesuai dengan pendapat responden.
3. Netral, yang berarti responden tidak memiliki pendapat atau pandangan terkait pernyataan yang diberikan.
4. Setuju, yang berarti pernyataan yang diberikan sesuai dengan pendapat dari responden.
5. Sangat Setuju, yang berarti aplikasi memberikan sesuatu yang melebihi ekspektasi dari responden terkait pernyataan yang diberikan.

## 2.7 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux, android menyediakan *platform* terbuka bagi para pengembang buat menciptakan

aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak (Wicaksono, 2013). Secara garis besar sistem operasi Android memiliki arsitektur :

1. *Applications dan Widget*
2. *Applications Frameworks*
3. *Libraries*
4. *Android Run Time*
5. *Linux Kernel*

## 2.8 Web Service

*Web Service* merupakan komponen perangkat lunak yang disimpan dalam suatu komputer yang dapat diakses oleh aplikasi, komponen perangkat lunak lain, atau komputer lain melalui suatu jaringan (Deitel & Deitel, 2012). Komunikasi *web service* menggunakan teknologi seperti XML, JSON, dan HTTP.

Perangkat yang menyediakan layanan disebut *web service host*. Aplikasi *client* mengirimkan *request* lewat suatu jaringan ke *web service host* dan memprosesnya lalu mengembalikannya berupa *response* kepada aplikasi *client*.

## 2.9 RESTful Web Service

REST merupakan salah satu standar komunikasi antara aplikasi yang heterogen. Arsitektur REST didasarkan pada *request* dan *response* yang ditransfer antara *client* dan *server* tanpa adanya partisipasi node yang mencatat *state* dari keadaan sebelumnya (Mehta, 2013). Menurut Mehta, prinsip utama pada REST antara lain:

- *Addressable resource* (sumber data memiliki alamat)
- Menggunakan standar HTTP
- Mampu mengirim berbagai format data
- *Statelessness* (tidak ada data client yang disimpan selama proses komunikasi)

RESTfull *web service* artinya suatu layanan yang memiliki *interface* dan mekanisme akses yang menggunakan prinsip dari REST. Beberapa perintah *request* yang digunakan dalam REST, antara lain:

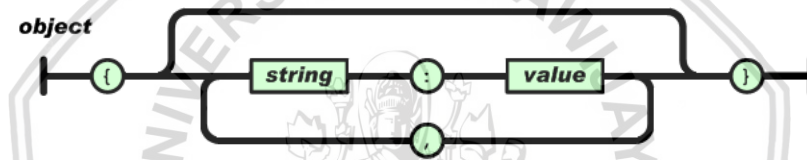
1. GET, biasanya digunakan untuk meminta sekumpulan data dari *server* ke *client*
2. POST, biasanya digunakan untuk membuat data baru ke *server*
3. PUT, biasanya untuk memperbarui data pada *server*
4. DELETE, biasanya digunakan untuk menghapus data *server*

## 2.10 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari *Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3-Desember 1999*. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON menggunakan bentuk sebagai berikut :

### 1. Objek

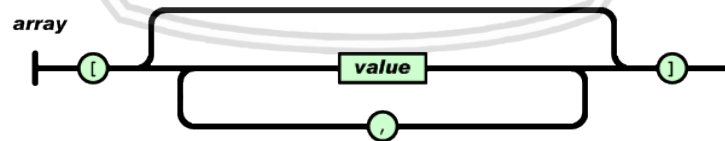
Objek merupakan sekumpulan pasangan *name/value* yang tidak teratur, sebuah objek dimulai dengan tanda awalan kurung kurawal ( { ) dan diakhiri dengan akhiran kurung kurawal ( } ). Masing – masing *name* diikuti dengan tanda titik dua ( : ) dan pasangan *name/value* dipisahkan oleh tanda koma ( , ) seperti yang terlihat pada gambar 2.10 berikut ini.



**Gambar 2.10 Struktur Objek JSON**  
 Sumber: Pengenalan JSON (2016)

### 2. Array

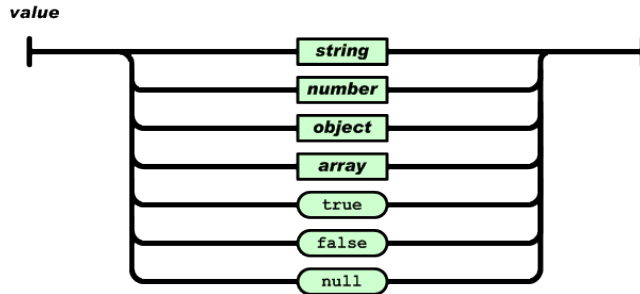
Array merupakan kumpulan dari *value* yang teratur. *Array* dimulai dengan awalan kurung siku ( [ ) dan diakhiri dengan akhiran kurung siku ( ] ). *Value* dipisahkan oleh koma ( , ), seperti yang terlihat pada gambar 2.11 berikut ini.



**Gambar 2.11 Struktur Array JSON**  
 Sumber: Pengenalan JSON (2016)

3. Value

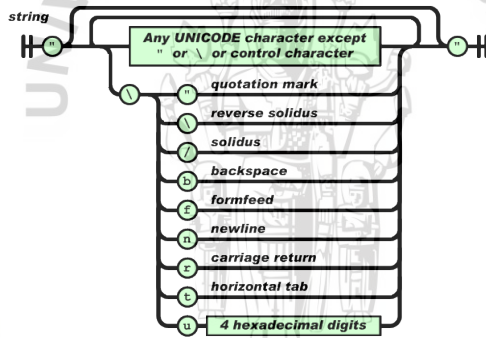
Value dapat berupa *string* di dalam *double quotes* ("), *number*, *true*, *false*, *null*, *object*, atau *Array*, seperti yang terlihat pada gambar 2.12 berikut ini.



Gambar 2.12 Struktur Value JSON  
 Sumber: Pengenalan JSON (2016)

4. String

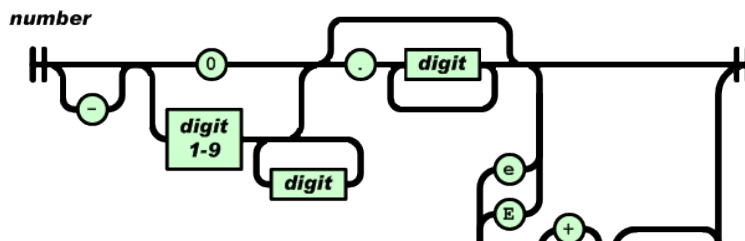
String merupakan rangkaian dari nol atau lebih karakter Unicode yang dibungkus dengan *double quotes* ("). *String* tunggal dinamakan karakter. *String* hampir sama seperti *string* yang ada pada bahasa C atau Java, seperti yang terlihat pada gambar 2.13 berikut ini.



Gambar 2.13 Struktur String JSON  
 Sumber: Pengenalan JSON (2016)

5. Number

Number hampir sama seperti yang ada pada bahasa C dan Java, terkecuali tidak digunakannya octal dan heksadesimal, seperti yang terlihat pada gambar 2.14 berikut ini.

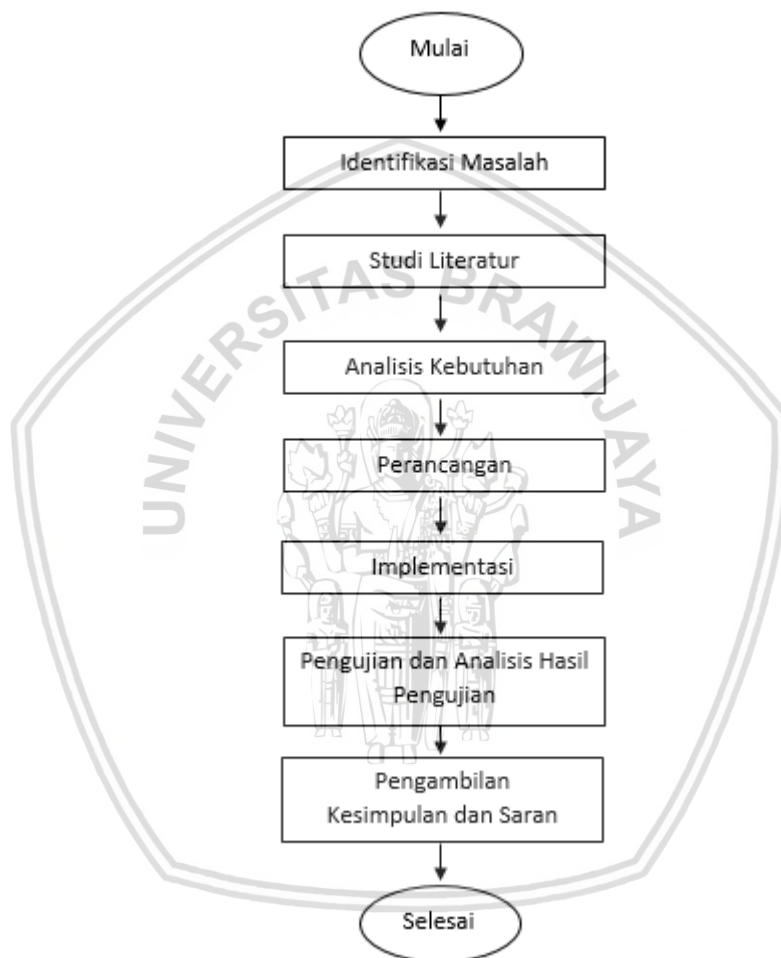


Gambar 2.14 Struktur Number JSON  
 Sumber: Pengenalan JSON (2016)



## BAB 3 METODOLOGI

Di dalam bab metodologi ini membahas langkah – langkah dan prosedur yang digunakan dalam proses penelitian ini dengan metode *Waterfall Model*. Di dalam metodologi penelitian yang digunakan penulis terdiri dari tujuh bagian yaitu identifikasi masalah, studi literatur, analisa kebutuhan, perancangan, implementasi, pengujian, dan pengambilan kesimpulan dan saran. Tahapan metodologi juga dapat dilihat pada gambar 3.1 dibawah ini.



Gambar 3.1 Diagram Alir Tahapan Penelitian

### 3.1 Identifikasi Masalah

Langkah awal yang dilakukan dalam penelitian ini adalah dengan melakukan identifikasi masalah yang ada pada objek penelitian untuk mengetahui apa saja dan sejauh apa penelitian yang dapat kita lakukan. Penelitian ini dilakukan dengan cara observasi dan wawancara untuk mengetahui permasalahan yang terjadi. Pada observasi dilakukan dengan cara pengamatan langsung pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan ataupun melakukan

wawancara terhadap Kepala Bidang Pengelolaan Informasi Administrasi Kependudukan dan Pemanfaatan Data pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan sesuai pedoman wawancara yang terlampir pada lampiran A dan perwakilan Masyarakat Kota Pasuruan yang pernah melakukan pengaduan berjumlah 10% dari rata-rata perbulan laporan pengaduan yang diterima di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan sesuai pedoman wawancara yang terlampir pada lampiran B.

### 3.2 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan sebagai pendukung dalam penulisan skripsi ini. Pengumpulan teori dilakukan dengan cara mengumpulkan literatur, jurnal, *browsing* internet dan sumber-sumber lainnya yang berhubungan erat dengan permasalahan yang diambil baik berupa buku ataupun *paper*.

### 3.3 Analisis Kebutuhan

Analisis kebutuhan diperlukan sebagai dasar dalam pembuatan aplikasi setelah tahap identifikasi masalah telah dilaksanakan. Analisis kebutuhan pada tahap ini dilakukan dengan cara wawancara pada Dinas kependudukan dan Pencatatan Sipil Kota Pasuruan yang diwakilkan oleh *Staff IT* sesuai pedoman wawancara yang tidak terstruktur yang terlampir pada lampiran C. Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh aplikasi kebutuhan yang diperoleh melalui identifikasi kebutuhan apa saja yang ingin didapatkan pada aplikasi (*Requirements*). Kemudian dilakukan pendekatan dengan cara memodelkan *requirement* yang telah didapatkan menggunakan Bahasa pemodelan UML (*Unified Modelling Language*).

Proses yang dilakukan dalam tahap analisis terhadap kebutuhan aplikasi ialah dengan membuat gambaran umum mengenai aplikasi, melakukan identifikasi *actor-actor* yang terlibat dalam aplikasi, melakukan penjabaran mengenai kebutuhan fungsional dan non-fungsional yang diperlukan dalam pembuatan aplikasi. Penjabaran kebutuhan fungsional yang dimodelkan ke dalam diagram *use case* dan *activity diagram*.

### 3.4 Perancangan

Perancangan aplikasi dilakukan setelah semua kebutuhan yang diperlukan untuk pembuatan aplikasi terpenuhi. Perancangan aplikasi berdasarkan *Object Oriented Analysis* dan *Object Oriented Design* menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Perancangan dimulai dengan pembuatan arsitektur aplikasi secara keseluruhan meliputi perancangan *sequence* diagram, perancangan *class* diagram, perancangan basis data dan perancangan desain antarmuka tampilan aplikasi.

### 3.5 Implementasi

Implementasi dilakukan dengan mengacu kepada perancangan aplikasi. Implementasi dilakukan dengan menggunakan tools *Android Studio* pada pembuatan platform *android* dengan menggunakan bahasa pemrograman java, sedangkan untuk platform *web*, menggunakan HTML 5, CCS 3.

Implementasi *user interface* didasarkan pada perancangan yang sudah dilakukan. Pada tahap akhir, dilakukan simulasi aplikasi yang telah dibuat pada *hardware virtual* pada aplikasi *android* yang telah terhubung dengan *Android Studio*, Sedangkan aplikasi *web* dilakukan pada *web browser*.

### 3.6 Pengujian Dan Analisis Hasil Pengujian

Pengujian perangkat lunak dibutuhkan untuk menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan (*requirement*) yang telah ditentukan. Agar pengujian pada perangkat lunak dapat berjalan dengan baik dan dapat mengetahui kekurangan yang ada pada aplikasi, maka dibutuhkan teknik pengujian perangkat. Teknik pengujian perangkat lunak yang digunakan dalam penelitian ini adalah teknik *Black Box Testing*.

Pengujian dibagi menjadi 2 bagian Pengujian berbasis *Website* akan dilakukan pada *web browser*, Sedangkan Pengujian berbasis *Android* dilakukan pada perangkat *smarthphone* berbasis *Android* dengan sistem operasi *Android 5.1 Lollipop* karena *Lollipop* sistem operasi *Android* yang banyak digunakan oleh pengguna.

Metode yang dipakai dalam pengujian yang akan dilakukan pada "Pengembangan Sistem Layanan Pengaduan Masyarakat Untuk Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan Berbasis *Android*" menggunakan *validation testing* dan *user acceptance testing*. Pengujian dilakukan untuk mengetahui kesiapan aplikasi untuk diimplementasikan.

*Validation Testing* merupakan salah satu metode yang ada dalam pengujian *black box testing*. Pengujian dengan metode *validation testing* mengacu pada pemeriksaan kepada sekumpulan aktifitas yang berbeda yang dilakukan untuk memastikan bahwa aplikasi telah mengimplementasikan fungsi sesuai dengan *requirements* yang telah didefinisikan sebelumnya.

Pada tahap pengujian *user acceptance testing*, dilakukan setelah pengujian *validation testing* dilakukan yakni suatu proses pengujian yang dilakukan oleh pengguna dengan hasil output dengan sebuah dokumen hasil uji yang dapat dijadikan bukti bahwa aplikasi tersebut sudah diterima dan sudah memenuhi kebutuhan yang diminta.

### 3.7 Pengambilan Keputusan dan Saran

Pada tahap ini Pengambilan Keputusan dilakukan setelah tahapan pengembang sistem dan pengujian telah selesai dilakukan. Kesimpulan diambil

dari analisis hasil pengujian dan analisis terhadap aplikasi yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memberikan pertimbangan atas pengembangan perangkat lunak lebih lanjut.





## BAB 4 ANALISIS KEBUTUHAN

Bab ini membahas mengenai analisis kebutuhan terhadap sistem yang akan dibangun. Analisis kebutuhan sistem dijelaskan dengan proses bisnis yang menggunakan BPMN, pemaparan permasalahan, Identifikasi Persyaratan berdasarkan pemangku kepentingan, deskripsi fitur dari sistem, persyaratan deskriptif, identifikasi aktor dan pemodelan *use case* beserta *activity diagram*.

### 4.1 Pemodelan Proses Bisnis

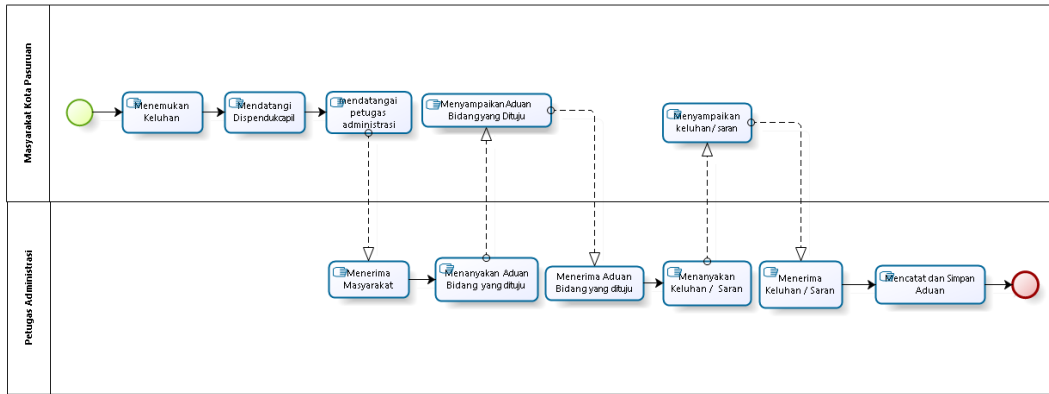
Pemodelan proses bisnis dilakukan untuk menggambarkan alur proses bisnis layanan pengaduan masyarakat pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan, baik proses bisnis yang sedang berjalan maupun proses bisnis yang dapat diwujudkan dalam bentuk perangkat lunak. Langkah – langkah yang dilakukan yaitu, mulai dari wawancara secara langsung kepada Kabid. PIAK dan Pemanfaatan Data yang terlampir pada lampiran D mengenai hasil wawancara yang telah dilakukan, kemudian dilanjutkan dengan melakukan analisis terhadap kegiatan pelayanan pengaduan yang telah ada sebelumnya di Dinas Kependudukan dan Pencatatan Sipil.

#### 4.1.1 Identifikasi Proses Bisnis AS-IS

Berdasarkan hasil wawancara dan observasi yang dilakukan, diperoleh bagaimana proses bisnis layanan pengaduan masyarakat kota pasuruan di Dinas Kependudukan dan Pencatatan Sipil berlangsung.

##### 4.1.1.1 Proses Masyarakat Kota Pasuruan membuat Pengaduan

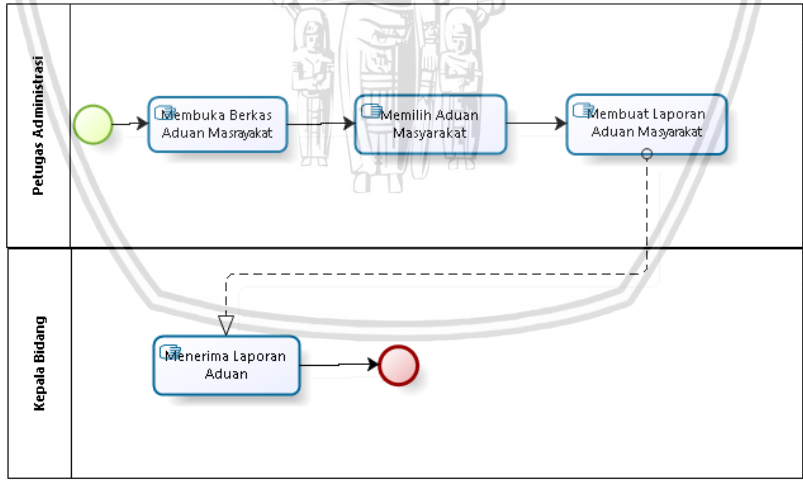
Setiap masyarakat Kota Pasuruan yang ingin melakukan pengaduan kepada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan harus mendatangi Dinas terkait. Kemudian, masyarakat tersebut akan menemui petugas registrasi untuk menyampaikan terkait keluhan atau aduan yang dialami sekaligus memberitahukan Bidang mana aduan tersebut ditujukan. Petugas Administrasi setelah menerima keluhan Masyarakat Kota Pasuruan kemudian mencatat dan menyimpan pengaduan tersebut untuk diproses. Proses pengaduan masyarakat sebelum ada sistem dapat digambarkan dengan BPMN dalam gambar 4.1.



Gambar 4.1 BPMN Masyarakat Kota Pasuruan Membuat Pengaduan Sistem Lama

4.1.1.2 Proses Memberikan Laporan Pengaduan oleh Petugas Administrasi

Petugas Administrasi akan melakukan pembuatan laporan pengaduan Masyarakat Kota Kota Pasuruan dengan membuka berkas seluruh pengaduan selanjutnya membuat laporan pengaduan masyarakat sesuai dengan pengaduan yang telah dipilih oleh Petugas Administrasi kemudian diberikan kepada Kepala Bidang sesuai bidang yang dituju pada laporan pengaduan masyarakat. Proses Memberikan Laporan Pengaduan oleh Petugas Administrasi sebelum ada sistem dapat digambarkan dengan BPMN dalam gambar 4.2.



Gambar 4.2 BPMN Memberikan Laporan Pengaduan Oleh Petugas Administrasi Sistem Lama

4.1.2 Analisis Permasalahan

Pada bagian ini akan dijelaskan lebih rinci tentang masalah yang sebelumnya dijelaskan dilatar belakang dalam bentuk tabel analisis permasalahan yang di tunjukkan pada Tabel 4.1.



**Tabel 4.1 Analisis Permasalahan**

Masalah	<ol style="list-style-type: none"> <li>1. Masyarakat diharuskan datang di Dinas Terkait</li> <li>2. Apabila banyaknya antrian masyarakat yang ingin menyampaikan pengaduan</li> <li>3. Apabila terjadi kelalaian Petugas Administrasi dalam proses pembuatan laporan Pengaduan Masyarakat</li> <li>4. Apabila Petugas Administrasi lagi berhalangan hadir</li> </ol>
Mempengaruhi	Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan
Dampak	<ol style="list-style-type: none"> <li>1. Membuang waktu tenaga dan biaya bagi Masyarakat Kota Pasuruan</li> <li>2. Petugas Admnistrasi tidak bisa mengerjakan pekerjaan yang lain apabila banyaknya masyarakat yang ingin menemuinya untuk penyampaian aduan</li> <li>3. Lobby Dinas terkait menjadi penuh dan sesak dengan antrian pengaduan yang banyak</li> <li>4. Laporan Pengaduan Masyarakat akan terhambat apabila terjadi kesalahan dalam proses pembuatan laporan pengaduan</li> <li>5. Kekecewaan Masyarakat Kota Pasuruan karena pengaduannya tidak bisa tersampaikan</li> </ol>
Solusi	Menyediakan sebuah sistem yang dapat digunakan oleh masyarakat Kota Pasuruan dalam penyampaian pengaduan dan dapat digunakan oleh Petugas Administrasi dalam melakukan pembuatan laporan pengaduan masyarakat secara terautomisasi

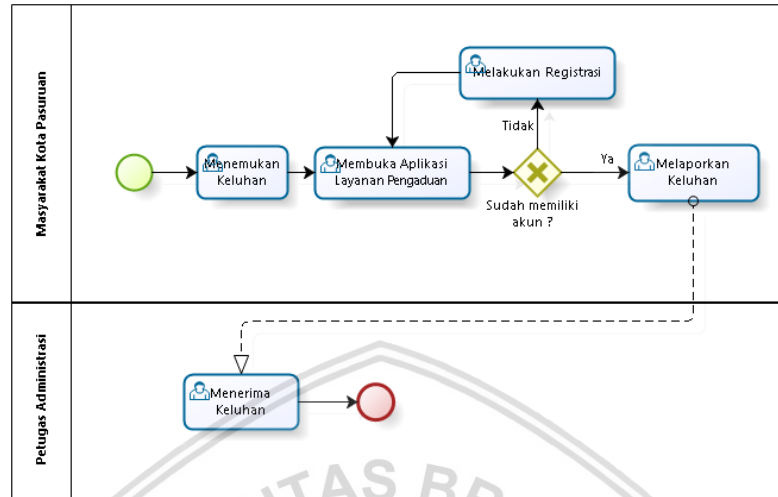
#### 4.1.3 Pemodelan Proses Layanan To-Be

Dari Hasil analisis permasalahan yang diperoleh, didapatkan solusi terhadap masing – masing proses bisnis yang dibuat dalam diagram BPMN



### 4.1.3.1 Proses Masyarakat Kota Pasuruan membuat Pengaduan

Pada proses Masyarakat Kota Pasuruan membuat Pengaduan yang sudah diperbarui memiliki alur proses yang digambarkan pada Gambar 4.3



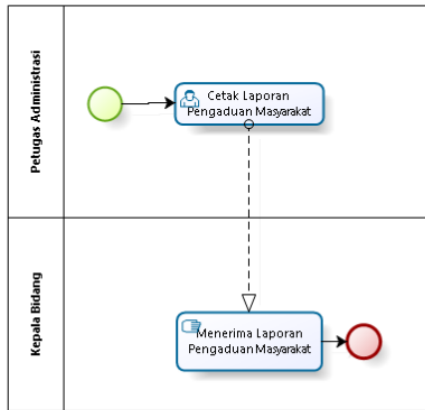
**Gambar 4.3 BPMN Masyarakat Kota Pasuruan membuat Pengaduan Sistem Baru**

Alur Proses Masyarakat Kota Pasuruan membuat Pengaduan dengan menemukan keluhan kemudian membuka Aplikasi Layanan Pengaduan. Apabila bila belum memiliki akun diwajibkan untuk melakukan registrasi dan kemudian Masyarakat Kota Pasuruan dapat melakukan pengaduan pada Aplikasi Layanan Pengaduan Masyarakat. Perbandingan Proses Masyarakat Kota Pasuruan membuat pengaduan sistem baru dengan sistem lama adalah mempermudah Masyarakat Kota Pasuruan dalam melaporkan pengaduan tanpa harus mendatangi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan tetapi dengan aplikasi Layanan Pengaduan Masyarakat maka pelaporan pengaduan dapat dilakukan dimana saja. Sedangkan bagi Petugas Administrasi tanpa harus mencatat pengaduan yang diterima karena akan tersimpan secara otomatis di sistem.

### 4.1.3.2 Proses Memberikan Laporan Pengaduan oleh Petugas Administrasi

Pada proses pendataan dan pelaporan pengaduan oleh Petugas Administrasi yang sudah diperbarui memiliki alur proses yang digambarkan pada Gambar 4.4





**Gambar 4.4 BPMN Memberikan Laporan Pengaduan oleh Petugas Administrasi Sistem Baru**

Alur proses memberikan laporan pengaduan oleh Petugas Administrasi dengan langsung melakukan fungsi cetak laporan pada sistem maka sistem otomatis mencetak laporan pengaduan sesuai dengan identitas masyarakat yang melakukan pengaduan tersebut, kemudian laporan pengaduan tersebut akan diberikan kepada Kepala Bidang sesuai pengaduan di bidang yang dituju dan menerima laporan pengaduan tersebut. Perbandingan proses memberikan laporan pengaduan oleh Petugas Administrasi sistem baru dengan sistem lama adalah mempermudah Petugas Administrasi dalam mencetak laporan pengaduan secara otomatis melalui sistem tanpa perlu membuka berkas pengaduan dan dibuahkan laporan pengaduan secara manual.

**4.1.4 Analisis Proses Bisnis**

Analisis Proses Bisnis dalam penelitian ini akan membahas mengenai Proses Bisnis yang diusulkan kepada Pihak Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan. Penjelasan analisis Proses Bisnis akan ditunjukkan pada tabel 4.2.

**Tabel 4.2 Analisis Proses Bisnis**

Kode BP	Nama BP	Deskripsi
BP01	Masyarakat Kota Pasuruan membuat Pengaduan	Dalam melakukan pengaduan masyarakat kota pasuruan dapat dilakukan dimana dan kapan saja dengan menggunakan sistem Layanan Pengaduan, tetapi diharuskan melakukan pendaftaran apabila belum memiliki akun pengguna di dalam sistem Layanan Pengaduan Masyarakat

**Tabel 4.3 Analisis Activity Proses Bisnis (Lanjutan)**

BP02	Memberikan Pengaduan oleh Administrasi	Laporan oleh Petugas	Petugas Administrasi akan lebih mudah dalam melakukan percetakan laporan pengaduan masyarakat Kota Pasuruan di dalam Sistem Layanan Pengaduan Masyarakat
------	--	----------------------	--

#### 4.1.5 Analisis Activity Proses Bisnis

Analisis Activity Proses Bisnis dalam penelitian ini akan membahas mengenai tiap aktivitas yang terdapat di dalam Bisnis Proses yang diusulkan kepada pihak Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan. Penjelasan Analisis Aktivitas Proses Bisnis akan di tunjukkan pada tabel 4.3.

**Tabel 4.3 Analisis Activity Proses Bisnis**

Kode BP	Kode ACT	Activity	Deskripsi
BP01	ACT01	Melakukan Registrasi	Masyarakat Kota Pasuruan sebelum melakukan pelaporan pengaduan diwajibkan melakukan Registrasi apabila tidak memiliki akun pada sistem layanan pengaduan
	ACT02	Melaporkan Keluhan	Masyarakat Kota Pasuruan dapat mengirimkan keluhan kepada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan melalui sistem layanan pengaduan
BP02	ACT03	Cetak Laporan Pengaduan	Petugas Administrasi dapat mencetak Laporan Pengaduan Masyarakat Kota Pasuruan secara otomatis melalui sistem layanan pengaduan

#### 4.2 Analisis Persyaratan

Analisis persyaratan dalam penelitian ini akan membahas mengenai analisis persyaratan yang dibutuhkan untuk merancang Sistem Layanan Pengaduan Masyarakat. Hal ini dilakukan agar sistem yang dikembangkan nantinya dapat memenuhi kebutuhan pengguna berdasarkan proses bisnis usulan yang telah di



modelkan sebelumnya. Terdapat beberapa bagian dalam melakukan analisis persyaratan, yaitu identifikasi tipe pemangku kepentingan, analisis persyaratan fungsional dan non-fungsional, serta pemodelan *usecase*.

#### 4.2.1 Identifikasi Pemangku Kepentingan

Identifikasi tipe pengguna dan pemangku kepentingan meliputi tipe pengguna dan pemangku kepentingan serta informasi pengguna sistem sesuai dengan proses bisnis usulan yang telah dilakukan sebelumnya pada tahap pemodelan proses bisnis.

##### 4.2.1.1 Tipe Pengguna dan Pemangku Kepentingan

Pada bagian ini dijelaskan mengenai tipe-tipe pemangku kepentingan sesuai *Pool / Lane* pada BPMN maupun yang berkaitan secara langsung dengan Sistem Layanan Pengaduan Masyarakat beserta deskripsi dan perwakilan masing-masing tipe pemangku kepentingan.

**Tabel 4.4 Tipe Pengguna dan Pemangku Kepentingan**

Tipe Pemangku Kepentingan	Deskripsi	Perwakilan Pemangku Kepentingan
Pengguna	Orang-orang yang berperan langsung dalam menggunakan sistem. Tipe pengguna ini akan dirumuskan sebagai aktor dalam pemodelan <i>usecase</i> .	Masyarakat Kota Pasuruan, Petugas Administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan
Pengembang	Orang yang menganalisis, merancang, dan mengembangkan atau membuat Sistem Layanan Pengaduan Masyarakat berdasarkan tahap-tahap tertentu.	Penulis
Pemegang wewenang	Orang yang memiliki wewenang untuk menindaklanjuti pengaduan masyarakat Kota Pasuruan pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.	Kepala Bidang di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan

Pada Tabel 4.4 Disebutkan bahwa terdapat 2 perwakilan kepentingan pada tipe pengguna, 1 perwakilan pada tipe pengembang, dan 1 perwakilan pada tipe pemegang wewenang. Masing – masing tipe pemangku kepentingan tersebut memiliki deskripsi peran yang berbeda – beda.

#### 4.2.1.2 Identifikasi Perwakilan Pemangku kepentingan

Pada bagian ini akan dijelaskan tentang tipe-tipe pemangku kepentingan yang berhubungan dengan sistem. Tipe –tipe tersebut dijelaskan pada Tabel 4.5.

**Tabel 4.5 Identifikasi Perwakilan Pemangku Kepentingan**

Perwakilan Pemangku Kepentingan	Deskripsi	Tanggung Jawab
Masyarakat Kota Pasuruan	Orang yang melakukan pengaduan pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	Memastikan bahwa sistem dapat memenuhi kebutuhan Masyarakat Kota Pasuruan

**Tabel 4.5 Identifikasi Perwakilan Pemangku Kepentingan (Lanjutan)**

Petugas Administrasi	Orang yang memiliki tanggung jawab dalam mengelola layanan pengaduan masyarakat dan pencatatan seluruh data pengaduan	Memastikan bahwa sistem dapat memenuhi kebutuhan Petugas Administrasi
----------------------	---	---

#### 4.2.1.3 Kebutuhan Pemangku Kepentingan

Pada bagian ini mendefinisikan kebutuhan masing-masing pemangku kepentingan. Masyarakat Kota Pasuruan diperoleh dari hasil analisis permasalahan yang telah dilakukan pada tahap pemodelan proses bisnis, sedangkan dari Petugas Administrasi diperoleh dengan melakukan wawancara kepada *Staff IT* yang terlampir pada lampiran F mengenai hasil wawancara yang telah dilakukan. Pendefinisian kebutuhan berikut bertujuan untuk menjelaskan alasan pemangku kepentingan memerlukan kebutuhan sistem.

**Tabel 4.6 Kebutuhan Pemangku Kepentingan**

Kebutuhan	Pemangku Kepentingan	Situasi Saat Ini	Solusi yang Ditawarkan
Integrasi dan Keamanan data	Masyarakat Kota Pasuruan dan Petugas Administrasi	-	Data tersimpan dalam satu database yang dilengkapi dengan proses otorisasi pada sistem





**Tabel 4.6 Kebutuhan Pemangku Kepentingan (Lanjutan)**

Akses Sistem dapat dilakukan 24 jam sehari 7 hari seminggu	Masyarakat Kota Pasuruan dan Petugas Administrasi	-	Sistem dapat diakses secara <i>online</i>
Kemudahan akses bagi pengguna sistem dimanapun dan kapanpun	Masyarakat Kota Pasuruan	-	Sistem dapat diakses selama terdapat koneksi internet dan memiliki aplikasi pada <i>smarthphone</i> yang digunakan
Kemudahan dalam melakukan pelaporan pengaduan	Masyarakat Kota Pasuruan	Masyarakat Kota Pasuruan harus datang secara	Pelaporan pengaduan dapat dilakukan secara
		langsung ke Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan untuk melakukan pelaporan pengaduan	<i>online</i> melalui sistem
Kemudahan dalam mendapatkan informasi administrasi kependudukan	Masyarakat Kota Pasuruan	Masyarakat Kota Pasuruan memperoleh informasi administrasi kependudukan melalui brosur yang disiapkan oleh Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	Informasi mengenai administrasi kependudukan Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dapat diakses dengan mudah melalui sistem
Kemudahan dalam merekapitulasi laporan pengaduan masyarakat	Petugas Administrasi	Petugas Administrasi harus mendata satu persatu status pengaduan yang sudah dan belum	Pendataan pengaduan berupa status pengaduan dan Kabid. Yang menerima aduan terbanyak dapat

**Tabel 4.6 Kebutuhan Pemangku Kepentingan (Lanjutan)**

		diproses sekaligus Kabid. Yang menerima pengaduan terbanyak secara manual	dilakukan melalui sistem
Kemudahan dalam mencetak laporan pengaduan masyarakat	Petugas Administrasi	Petugas Administrasi harus menulis laporan pengaduan pada kertas pengaduan	Pencetakan laporan pengaduan masyarakat dapat dilakukan melalui sistem
Kemudahan dalam mengubah status laproan pengaduan	Petugas Administrasi	Petugas Administrasi memilah laporan pengaduan yang sudah dan belum diproses	Pengubahan status laporan pengaduan masyarakat dapat dilakukan melalui sistem
Kemudahan dalam melihat informasi seluruh laporan pengaduan	Petugas Administrasi	Petugas Administrasi melihat satu persatu laporan pengaduan secara manual	Informasi terbaru seluruh laporan pengaduan tersedia secara otomatis melalui sistem
Keamanan sistem Layanan Pengaduan	Petugas Administrasi	-	Sistem dapat melakukan autentifikasi data registrasi yang khusus untuk pengguna masyarakat Kota Pasuruan

Tabel 4.6 tersebut memuat 10 kebutuhan pemangku kepentingan yang ditemukan pada penelitian ini yang disertai dengan penjelasan mengenai alasan mengapa persyaratan tersebut dibutuhkan oleh masing-masing pemangku kepentingan.

#### 4.2.2 Identifikasi Fitur

Fitur menjelaskan tentang kesimpulan dari kemampuan Sistem Layanan Pengaduan yang akan dibangun sesuai. Fitur yang dijelaskan pada Tabel 4.7 berikut ini didapatkan dari penjabaran kebutuhan pemangku kepentingan.



Tabel 4.7 Identifikasi Fitur

Kode ACT	Kode	Nama Fitur	Deskripsi
-	FITUR1	Login	Sistem dapat melakukan autentifikasi pengguna
-	FITUR2	Logout	Sistem dapat digunakan untuk keluar dari sistem
ACT01	FITUR3	Registrasi	Sistem dapat digunakan untuk pendaftaran oleh masyarakat Kota Pasuruan
-	FITUR4	Mengelola Akun	Sistem dapat digunakan oleh Masyarakat Kota Pasuruan melihat biodata profil akun miliknya dan merubah <i>password</i>
-	FITUR5	Melihat Informasi Layanan	Sistem dapat digunakan oleh Masyarakat Kota Pasuruan untuk melihat informasi layanan administrasi kependudukan
-	FITUR6	Melihat Daftar Pengaduan	Sistem dapat digunakan oleh Masyarakat Kota Pasuruan untuk melihat informasi pengaduan yang telah terkirim
ACT02	FITUR7	Melakukan pelaporan keluhan	Sistem dapat digunakan oleh Masyarakat Kota Pasuruan untuk melakukan pelaporan keluhan kepada Dinas Kependudukan dan Pencatatan Kota Pasuruan
-	FITUR8	Melihat Informasi Profil Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	Sistem dapat digunakan oleh Masyarakat Kota Pasuruan untuk melihat Profil maupun Alamat Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan
-	FITUR9	Melihat informasi agregat penguaduan	Sistem dapat digunakan oleh Petugas Administrasi untuk melihat informasi data pengaduan berupa berapa banyak aduan yang belum dan sudah diproses dan Bidang yang sering mendapat laporan aduan

Tabel 4.7 Identifikasi Fitur (Lanjutan)

-	FITUR10	Mengelola data pengaduan	Sistem dapat digunakan oleh Petugas Administrasi untuk mengetahui informasi seluruh laporan pengaduan serta dapat melakukan ubah status, mengirim pesan balasan dan hapus aduan dari masyarakat kota pasuruan yang dipilih
ACT03	FITUR11	Mencetak Laporan Pengaduan	Sistem dapat digunakan oleh Petugas Administasi untuk membuat berita acara laporan pengaduan masyarakat secara otomatis
-	FITUR12	Mengirim pesan balasan pengaduan	Sistem dapat digunakan oleh Petugas Administrasi untuk mengirimkan pesan balasan dari pengaduan yang telah selesai diproses kepada Masyarakat Kota Pasuruan
-	FITUR13	Melihat pesan balasan laporan pengaduan	Sistem dapat digunakan oleh Masyarakat Kota Pasuruan untuk mengetahui isi pesan balasan laporan pengaduan yang telah diproses
-	FITUR14	Mengelola data pengguna	Sistem dapat digunakan oleh Petugas Administrasi untuk melakukan tambah,habus,ubah status dan ubah password pengguna aplikasi layanan pengaduan
-	FITUR15	Keamanan Sistem Layanan Pengaduan	Sistem harus mampu melakukan autentifikasi data registrasi yang khusus digunakan untuk masyarakat Kota Pasuruan

Tabel 4.7 memuat fitur-fitur sistem yang diperoleh dari hasil analisis kebutuhan pengguna yang telah dilakukan pada Tabel 4.6 dan analisis activity proses bisnis pada Tabel 4.3.

### 4.3 Persyaratan Deklaratif

Pada tahap ini akan menjelaskan mengenai persyaratan-persyaratan yang harus dipenuhi oleh sistem secara deskriptif. Persyaratan-persyaratan tersebut dibagi menjadi dua macam, yakni persyaratan fungsional dan persyaratan non-fungsional

#### 4.3.1 Persyaratan Fungsional

Persyaratan fungsional merupakan pengelompokan persyaratan yang harus disediakan oleh sistem secara fungsional didalamnya. Persyaratan ini didasarkan pada identifikasi fitur yang telah didefinisikan pada Tabel 4.7. Persyaratan fungsional dikelompokkan berdasarkan pemangku kepentingan. Spesifikasi Persyaratan Fungsional Masyarakat Kota Pasuruan ditunjukkan pada Tabel 4.8.

**Tabel 4.8 Spesifikasi Kebutuhan Fungsional Masyarakat Kota Pasuruan**

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi	Use Case
FITUR1	SRS-F-EPKT-01	Login	Masyarakat Kota Pasuruan memasukkan No. HP dan password untuk dapat masuk dan menggunakan Sistem	Login
FITUR2	SRS-F-EPKT-02	Logout	Masyarakat Kota Pasuruan dapat melakukan akun miliknya keluar dari sistem	Logout
FITUR3	SRS-F-EPKT-03	Registrasi	Masyarakat Kota Pasuruan membuat akun baru pada sistem untuk dapat menggunakan sistem	Registrasi
FITUR4	SRS-F-EPKT-04	Melihat Profil Pengguna	Masyarakat Kota Pasuruan Melihat Biodata Akun Miliknya di Sistem	Melihat Profil
FITUR4	SRS-F-EPKT-05	Mengubah Password Pengguna	Masyarakat Kota Pasuruan dapat mengubah password akun miliknya di sistem	Ubah Password
FITUR5	SRS-F-EPKT-06	Melihat Layanan	Masyarakat Kota Pasuruan dapat melihat	Melihat Layanan

**Tabel 4.8 Spesifikasi Kebutuhan Fungsional Masyarakat Kota Pasuruan  
(Lanjutan)**

			layanan yang diberikan oleh Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	
FITUR6	SRS-F-EPKT-07	Melihat Daftar Pengaduan	Masyarakat Kota Pasuruan dapat melihat daftar pengaduan miliknya maupun mengetahui status pengaduannya di sistem	Melihat Daftar Pengaduan
FITUR7	SRS-F-EPKT-08	Melaporkan Aduan	Masyarakat Kota Pasuruan dapat melakukan aduan terkait keluhan/saran/kritik dengan mengisi form aduan yang disediakan oleh sistem	Melaporkan Pengaduan
FITUR8	SRS-F-EPKT-09	Melihat Informasi Dinas Terkait	Masyarakat Kota Pasuruan dapat melihat informasi alamat dan nomor telepon Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan	Melihat Tentang Kami
FITUR13	SRS-F-EPKT-10	Melihat Pesan Balasan Laporan Pengaduan	Masyarakat Kota Pasuruan dapat melihat isi pesan balasan pengaduan yang telah diproses	Melihat Pesan Balasan Pengaduan

Spesifikasi Persyaratan Fungsional Petugas Administrasi ditunjukkan pada tabel 4.9

Tabel 4.9 Spesifikasi Persyaratan Fungsional Petugas Administrasi

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi	Use Case
FITUR1	SRS-F-EPKT-11	Login	Petugas Administrasi memasukkan <i>username</i> dan <i>password</i> untuk dapat masuk dan menggunakan Sistem	Login
FITUR2	SRS-F-EPKT-12	Logout	Petugas Administrasi dapat melakukan akun miliknya keluar dari sistem	Logout
FITUR9	SRS-F-EPKT-13	Melihat Halaman Utama	Petugas Administrasi dapat melihat informasi data pengaduan berupa total pengaduan, status pengaduan dan Kabid. Yang sering menerima pengaduan	Melihat Beranda
FITUR10	SRS-F-EPKT-14	Mengelola Data Pengaduan	Petugas Administrasi dapat melihat informasi daftar pengaduan yang telah dikirim oleh masyarakat Kota Pasuruan di Sistem	Melihat Data Pengaduan
FITUR10	SRS-F-EPKT-15	Hapus Pengaduan	Petugas Administrasi dapat menghapus pesan pengaduan masyarakat di Sistem	Hapus Aduan
FITUR10	SRS-F-EPKT-16	Ubah Status Aduan	Petugas Administrasi dapat mengubah status pesan pengaduan masyarakat di Sistem	Ubah Status Aduan
FITUR11	SRS-F-EPKT-17	Cetak Aduan	Petugas Administrasi dapat mencetak pengaduan masyarakat sebagai laporan aduan di Sistem	Cetak Aduan
FITUR12	SRS-F-EPKT-18	Balas Aduan	Petugas Administrasi dapat mengirim pesan	Balas Aduan

**Tabel 4.9 Spesifikasi Persyaratan Fungsional Petugas Administrasi (Lanjutan)**

			konfirmasi pengaduan masyarakat di Sistem	
FITUR14	SRS-F-EPKT-19	Mengelola Data Pengguna	Petugas Administrasi dapat melihat informasi daftar seluruh pengguna yang telah terdaftar oleh Sistem	Mengelola Data Pengguna
FITUR14	SRS-F-EPKT-20	Tambah Pengguna	Petugas Administrasi dapat menambah daftar pengguna layanan pengaduan masyarakat di Sistem	Tambah Pengguna
FITUR14	SRS-F-EPKT-21	Ubah Status Akun Pengguna	Petugas Administrasi dapat mengubah status akun pengguna di Sistem	Ubah Status Akun Pengguna
FITUR14	SRS-F-EPKT-22	Hapus Pengguna	Petugas Administrasi dapat menghapus akun pengguna di Sistem	Hapus Pengguna
FITUR14	SRS-F-EPKT-23	Ubah Password Pengguna	Petugas Administrasi dapat mengubah password akun pengguna di Sistem	Ubah Password Pengguna

Tabel 4.8 dan Tabel 4.9 merupakan daftar persyaratan fungsional sistem yang akan digunakan sebagai dasar dalam pembuatan sistem. Persyaratan fungsional diperoleh dari pendefinisian fitur pada Tabel 4.7.

### 4.3.2 Persyaratan Non-Fungsional

Persyaratan non-fungsional merupakan pengelompokan persyaratan yang tidak terikat dengan fungsional tertentu dari sistem. Persyaratan ini berisi batasan kemampuan yang disediakan oleh sistem sebagai pendukung yang didasarkan pada identifikasi fitur pada Tabel 4.7. Spesifikasi Persyaratan non-fungsional sistem ditunjukkan pada tabel 4.10.

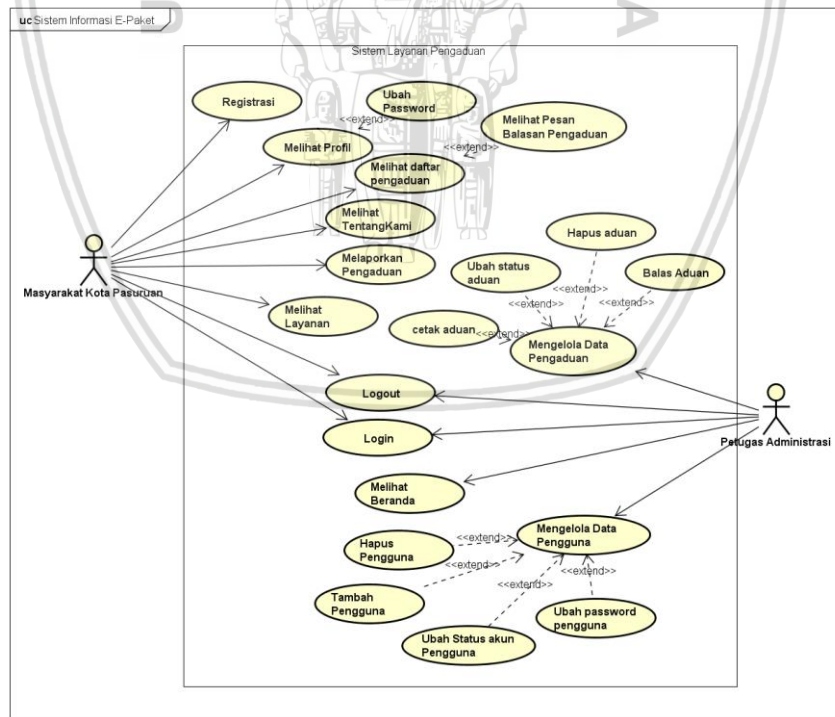


Tabel 4.10 Spesifikasi Kebutuhan Non-Fungsional

Kode Fitur	Kode Fungsi	Parameter	Deskripsi
FITUR15	SRS-NF-EPKT-1	Security	Sistem dapat melakukan penolakan registrasi pengguna apabila pengguna yang melakukan registrasi pada sistem layanan pengaduan masyarakat bukan masyarakat yang memiliki KTP kota pasuruan

#### 4.4 Pemodelan Use Case

Pemodelan *use case* dilakukan dengan menggunakan *use case* diagram yang dilengkapi dengan hubungan antar *use case* maupun hubungan masing-masing *use case* dengan aktor sistem. *Use Case* diagram ini melibatkan Masyarakat Kota Pasuruan dan Petugas Administrasi sebagai aktor. *Use Case* diagram ditunjukkan dalam gambar 4.5.



Gambar 4.5 Use Case Diagram Sistem Layanan Pengaduan Masyarakat

Gambar 4.5 merupakan pemodelan persyaratan fungsional pada Sistem Layanan Pengaduan Masyarakat untuk Dinas Kependudukan dan Pencatatan Sipil



Kota Pasuruan. *Use Case* dibuat berdasarkan hasil penjabaran persyaratan fungsional pada Tabel 4.8 dan 4.9, sedangkan aktor diperoleh dari penjabaran pengguna sistem pada Tabel 4.5.

#### 4.5 Use Case Description

*Use Case Description* berisi tentang penjelasan lebih lanjut mengenai *use case* Sistem Layanan Pengaduan Masyarakat yang telah digambarkan pada Gambar 4.5. Tabel spesifikasi *use case* berisi beberapa keterangan seperti Kode *use case*, deskripsi singkat *use case*, *actor*, *basic flows*, *subflows*, *alternative flow*, *pre-condition*, dan *post-condition*.

##### 4.5.1 Use case description Login

Tabel 4.11 Use Case description Login

<b>Flow of Events untuk use case login</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-01
<b>Brief Description</b>	<i>Use case</i> ini menjelaskan bagaimana User masuk ke dalam Sistem Informasi Layanan pengaduan Masyarakat
<b>Actor</b>	Masyarakat Kota Pasuruan
<b>Pre-Condition</b>	<ol style="list-style-type: none"> <li>1. Masyarakat Kota Pasuruan membuka aplikasi Sistem Informasi Layanan pengaduan Masyarakat</li> <li>2. Masyarakat Kota Pasuruan telah melakukan registrasi di Sistem Informasi Layanan pengaduan Masyarakat</li> </ol>
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan form <i>login</i> untuk masuk ke dalam sistem</li> <li>2. Masyarakat Kota Pasuruan memasukkan no.hp dan <i>password</i> , kemudian menekan tombol "LOGIN"</li> <li>3. Sistem melakukan autentifikasi akun Masyarakat Kota Pasuruan</li> <li>4. Sistem menampilkan halaman berada sitem Layanan pengaduan Masyarakat</li> </ol>
<b>Alternative Flows</b>	<p><b>a. {no.hp dan <i>password</i>} tidak terdaftar</b></p> <p>Jika, di <i>basic flow</i>, Masyarakat Kota Pasuruan memasukkan no.hp atau <i>password</i> yang tidak terdaftar dalam sistem,maka sistem menampilkan notofikasi "No.HP / Password anda salah"</p>
<b>Subflow</b>	Tidak ada



Tabel 4.11 *Use Case description Login (Lanjutan)*

<b>Post-Conditions</b>	Jika Berhasil, maka Masyarakat Kota Pasuruan dapat masuk ke dalam Sistem Informasi Layanan pengaduan Masyarakat
------------------------	---

4.5.2 *Use case description Registrasi*Tabel 4.12 *Use Case description Registrasi*

<b>Flow of Events untuk use case Registrasi</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-03
<b>Brief Description</b>	<i>Use Case</i> ini menjelaskan Masyarakat Kota Pasuruan dapat membuat akun ke dalam sistem dengan memasukkan data pribadi sesuai yang diminta oleh sistem
<b>Actor</b>	Masyarakat Kota Pasuruan
<b>Pre-Condition</b>	Masyarakat Kota Pasuruan berada di halaman login Sistem Informasi Layanan pengaduan Masyarakat
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Masyarakat Kota Pasuruan memilih menu Daftar</li> <li>2. Sistem menampilkan Form Register untuk melakukan pendaftaran akun</li> <li>3. Masyarakat Kota Pasuruan mengisi form dan memasukkan data pribadi</li> <li>4. Masyarakat Kota Pasuruan menekan tombol "Daftar"</li> <li>5. Sistem melakukan verifikasi data Form Registrasi</li> <li>6. Sistem menampilkan halaman login dan notifikasi bahwa "Registrasi Berhasil"</li> </ol>
<b>Alternative Flows</b>	<p><b>a. Data registrasi salah</b></p> <p>Jika, di <i>basic flow</i>, data registrasi yang dimasukkan oleh Masyarakat Kota Pasuruan tidak benar disebabkan tidak memiliki KTP Kota Pasuruan, maka sistem menampilkan notifikasi "Pendaftar harus KTP KOTA Pasuruan"</p> <p><b>b. Data Registrasi tidak lengkap</b></p> <p>Jika, di <i>basic flow</i>, Masyarakat Kota Pasuruan memasukkan data registrasi tidak lengkap atau tidak mengisi salah satu data, maka sistem menampilkan notifikasi bahwa terdapat salah satu data yang belum terisi</p>

Tabel 4.12 *Use Case description* Registrasi (Lanjutan)

<b>Subflow</b>	Tidak ada
<b>Post-Condition</b>	Masyarakat Kota Pasuruan berhasil melakukan registrasi atau pendaftaran akun ke sistem

#### 4.5.3 *Use case description* Melihat Daftar Pengaduan

Tabel 4.13 *Use Case description* Melihat Daftar Pengaduan

<b>Flow of Events untuk use case Melihat Daftar Pengaduan</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-07
<b>Brief Description</b>	<i>Use Case</i> ini menjelaskan Masyarakat Kota Pasuruan dapat mengetahui informasi daftar dan status pengaduan yang telah dikirim maupun pesan balasan dari pengaduan yang telah dirposes di sistem
<b>Actor</b>	Masyarakat Kota Pasuruan
<b>Pre-Condition</b>	Masyarakat Kota Pasuruan berada di halaman beranda sistem Layanan Pengaduan Masyarakat
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Masyarakat Kota Pasuruan memilih menu Daftar Pengaduan</li> <li>2. Sistem menampilkan informasi Daftar Pengaduan beserta status aduan</li> <li>3. Jika Masyarakat Kota Pasuruan memilih operasi Melihat Pesan Balasan Pengaduan maka sistem akan menjalankan <i>sub flow</i> <b>Melihat Pesan Balasan Pengaduan</b></li> </ol>
<b>Alternative Flows</b>	Tidak ada
<b>Subflow</b>	<b>Melihat Pesan Balasan Pengaduan</b> <ol style="list-style-type: none"> <li>1. Petugas Administrasi menekan tombol Status Pengaduan</li> <li>2. Sistem akan menampilkan notifikasi berupa isi pesan balasan pengaduan</li> </ol>
<b>Post-Condition</b>	Masyarakat Kota Pasuruan telah mengetahui informasi daftar dan status pengaduannya maupun pesan balasan dari pengaduan yang telah dirposes di sistem

#### 4.5.4 Use case description Melaporkan Pengaduan

Tabel 4.14 Use Case description Melaporkan Pengaduan

<b>Flow of Events untuk use case Melaporkan Pengaduan</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-08
<b>Brief Description</b>	Use Case ini menjelaskan Masyarakat Kota Pasuruan dapat melakukan pelaporan pengaduan kepada Bidang yang dituju terkait keluhan / saran dan kritik
<b>Actor</b>	Masyarakat Kota Pasuruan
<b>Pre-Condition</b>	Masyarakat Kota Pasuruan berada di halaman beranda sistem Layanan Pengaduan Masyarakat
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Masyarakat Kota Pasuruan memilih menu Pengaduan</li> <li>2. Sistem menampilkan form untuk melaporkan pengaduan</li> <li>3. Masyarakat Kota Pasuruan mengisi form dan memasukkan data pengaduan</li> <li>4. Masyarakat Kota Pasuruan menekan tombol "KIRIM"</li> <li>5. Sistem menampilkan notifikasi bahwa "Pengaduan anda Berhasil Terkirim"</li> </ol>
<b>Alternative Flows</b>	<p><b>a. Data laporan pengaduan tidak lengkap</b></p> <p>Jika, di basic flow, Masyarakat Kota Pasuruan memasukkan data laporan pengaduan tidak lengkap atau tidak mengisi salah satu data, maka sistem menampilkan notifikasi bahwa terdapat salah satu data yang belum terisi</p>
<b>Subflow</b>	Tidak ada
<b>Post-Condition</b>	Masyarakat Kota Pasuruan berhasil melakukan pelaporan pengaduan

#### 4.5.5 Use case description Melihat Layanan

Tabel 4.15 Use Case description Melihat Layanan

<b>Flow of Events untuk use case Melihat Layanan</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-06

Tabel 4.15 Use Case description Melihat Layanan (Lanjutan)

<b>Brief Description</b>	Use Case ini menjelaskan Masyarakat Kota Pasuruan dapat mengetahui informasi layanan administrasi kependudukan dan pencatatan sipil
<b>Actor</b>	Masyarakat Kota Pasuruan
<b>Pre-Condition</b>	Masyarakat Kota Pasuruan berada di halaman beranda sistem Layanan Pengaduan Masyarakat
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Masyarakat Kota Pasuruan memilih menu Layanan</li> <li>2. Sistem menampilkan informasi layanan administrasi kependudukan dan pencatatan sipil</li> </ol>
<b>Alternative Flows</b>	Tidak ada
<b>Subflow</b>	Tidak ada
<b>Post-Condition</b>	Masyarakat Kota Pasuruan telah mengetahui informasi layanan administrasi kependudukan dan pencatatan sipil yang tersedia oleh sistem

#### 4.5.6 Use case description Mengelola Data Pengguna

Tabel 4.16 Use Case description Mengelola Data Pengguna

<b>Flow of Events untuk use case Mengelola Data Pengguna</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-19
<b>Brief Description</b>	Use case ini menjelaskan bagaimana Petugas Administrasi mengelola data pengguna meliputi ubah password, tambah, hapus dan ubah status akun pengguna
<b>Actor</b>	Petugas Administrasi
<b>Pre-Condition</b>	Petugas Administrasi berada di halaman beranda Sistem Layanan Pengaduan Masyarakat
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Petugas Administrasi memilih menu Data Pengguna</li> <li>2. Sistem menampilkan informasi daftar pengguna sistem dan menampilkan kolom aksi untuk mengelola data pengguna {<b>memilih operasi pengelolaan data pengguna</b>}</li> <li>3. Jika Petugas Administrasi memilih operasi tambah pengguna maka sistem akan menjalankan <i>sub flow</i> <b>Tambah Pengguna</b></li> <li>4. Jika Petugas Administrasi memilih operasi hapus</li> </ol>



Tabel 4.16 Use Case description Mengelola Data Pengguna (Lanjutan)

	<p>pengguna maka sistem akan menjalankan <i>sub flow Hapus Pengguna</i></p> <p>5. Jika Petugas Administrasi memilih operasi ubah status akun pengguna maka sistem akan menjalankan <i>sub flow Ubah Status Akun Pengguna</i></p> <p>6. Jika Petugas Administrasi memilih operasi ubah password pengguna maka sistem akan menjalankan <i>sub flow Ubah Password Pengguna</i></p>
<b>Post-Conditions</b>	Petugas Administrasi berhasil mengelola Data Pengguna
<b>Alternative Flows</b>	<p><b>a. Data diri pengguna salah</b></p> <p>Pada subflow Tambah Pengguna apabila data diri pengguna yang dimasukkan oleh Petugas Administras tidak benar disebabkan bukan KTP Kota Pasuruan, maka sistem menampilkan notifikasi “Pendaftar harus KTP KOTA Pasuruan”.</p> <p><b>b. NIK / No. HP Pengguna telah terdaftar</b></p> <p>Pada subflow Tambah Pengguna, sistem akan melakukan pengecekan terhadap NIK / No.HP Pengguna yang baru dimasukkan. Jika data tersebut telah tersimpan dalam sistem sebelumnya, maka data tersebut tidak akan tersimpan dan menampilkan notifikasi bahwa “Pendaftaran Gagal: NIK / No.HP telah terdaftar di sistem e-Paket!”</p> <p><b>c. Gagal Ubah Password Akun Pengguna</b></p> <p>Pada <i>subflow</i> Ubah Password Akun Pengguna, sistem akan melakukan pengecekan apakah data password lama tidak sesuai dengan password pengguna di sistem. Jika benar tidak sesuai maka sistem akan menampilkan notifikasi “Update Gagal: inputan password lama anda salah!”</p>
<b>Subflow</b>	<p><b>Tambah Pengguna</b></p> <ol style="list-style-type: none"> <li>1. Petugas Administrasi menekan tombol Tambah Pengguna</li> <li>2. Sistem akan menampilkan form untuk menambahkan pengguna</li> <li>3. Petugas Administrasi mengisi form dan memasukkan data diri pengguna</li> </ol>



Tabel 4.16 Use Case description Mengelola Data Pengguna (Lanjutan)

	<p>4. Petugas Administrasi menekan tombol “Tambah”</p> <p>5. Sistem melakukan verifikasi data pribadi pengguna</p> <p>6. Sistem menampilkan notifikasi bahwa “Berhasil Terupdate : Akun berhasil terdaftar!”</p> <p><b>Hapus Pengguna</b></p> <p>1. Petugas Administrasi memilih icon “Hapus” pada kolom aksi</p> <p>2. Sistem menampilkan pemberitahuan ““Yakin ingin hapus Akun [nama_lengkap] ?”</p> <p>3. Petugas Administrasi menekan tombol “OKE”</p> <p>4. Sistem melakukan operasi penghapusan akun pengguna</p> <p>5. Sistem menampilkan notifikasi bahwa “Berhasil Terupdate : Akun berhasil dihapus!”</p> <p><b>Ubah Status Akun Pengguna</b></p> <p>1. Petugas Administrasi memilih icon “Ubah Status Akun” pada kolom aksi</p> <p>2. Sistem menampilkan pemberitahuan “Yakin ubah status Akun [nama_lengkap] ?”</p> <p>3. Petugas Administrasi menekan tombol “OKE”</p> <p>4. Sistem melakukan operasi pengubahan status akun pengguna</p> <p>5. Sistem menampilkan notifikasi bahwa “Berhasil Terupdate : Status telah berubah!”</p> <p><b>Ubah Password Pengguna</b></p> <p>1. Petugas Administrasi memilih icon “Ubah Password” pada kolom aksi</p> <p>2. Sistem akan menampilkan form untuk mengubah password pengguna</p> <p>3. Petugas Administrasi mengisi form dengan memasukkan data password lama kemudian password baru pengguna</p> <p>4. Petugas Administrasi menekan tombol “Simpan Perubahan”</p> <p>5. Sistem melakukan operasi dan verifikasi pengubahan password akun pengguna</p> <p>6. Sistem menampilkan notifikasi bahwa “Berhasil Update: password anda berhasil di ubah!”</p>
--	--



#### 4.5.7 Use case description Mengelola Data Pengaduan

Tabel 4.17 Use Case description Mengelola Data Pengaduan

<b>Flow of Events untuk use case Mengelola Data Pengaduan</b>	
<b>Kode Use Case</b>	SRS-F-EPKT-14
<b>Brief Description</b>	Use case ini menjelaskan bagaimana Petugas Administrasi mengelola data pengaduan meliputi ubah status dan hapus pengaduan maupun mencetak pengaduan yang digunakan sebagai pelaporan
<b>Actor</b>	Petugas Administrasi
<b>Pre-Condition</b>	Petugas Administrasi berada di halaman beranda Sistem Layanan Pengaduan Masyarakat
<b>Basic Flow of Events</b>	<ol style="list-style-type: none"> <li>1. Petugas Administrasi memilih menu Data Pengaduan</li> <li>2. Sistem menampilkan informasi daftar pengguna masyarakat dan menampilkan kolom aksi untuk mengelola data pengaduan {memilih operasi pengelolaan data pengaduan}</li> <li>3. Jika Petugas Administrasi memilih operasi balas pengaduan maka sistem akan menjalankan <i>sub flow</i> <b>Balas Aduan</b></li> <li>4. Jika Petugas Administrasi memilih operasi hapus pengaduan maka sistem akan menjalankan <i>sub flow</i> <b>Hapus Aduan</b></li> <li>5. Jika Petugas Administrasi memilih operasi ubah status pengaduan maka sistem akan menjalankan <i>sub flow</i> <b>Ubah Status Aduan</b></li> <li>6. Jika Petugas Administrasi memilih operasi cetak pengaduan maka sistem akan menjalankan <i>sub flow</i> <b>Cetak Aduan</b></li> </ol>
<b>Alternative Flows</b>	<p><b>a. Gagal Hapus Aduan</b></p> <p>Pada <i>subflow</i> Hapus Aduan, sistem akan melakukan pengecekan apakah status pengaduan telah berubah menjadi BELUM PROSES di sistem. Jika benar maka sistem akan menampilkan notifikasi "Gagal Terhapus : Aduan Pesan harus status Sudah Di Proses!"</p>
<b>Subflow</b>	<p><b>Balas Aduan</b></p> <ol style="list-style-type: none"> <li>1. Petugas Administrasi memilih icon "Balas Aduan" pada kolom aksi</li> </ol>

Tabel 4.17 Use Case description Mengelola Data Pengaduan (Lanjutan)

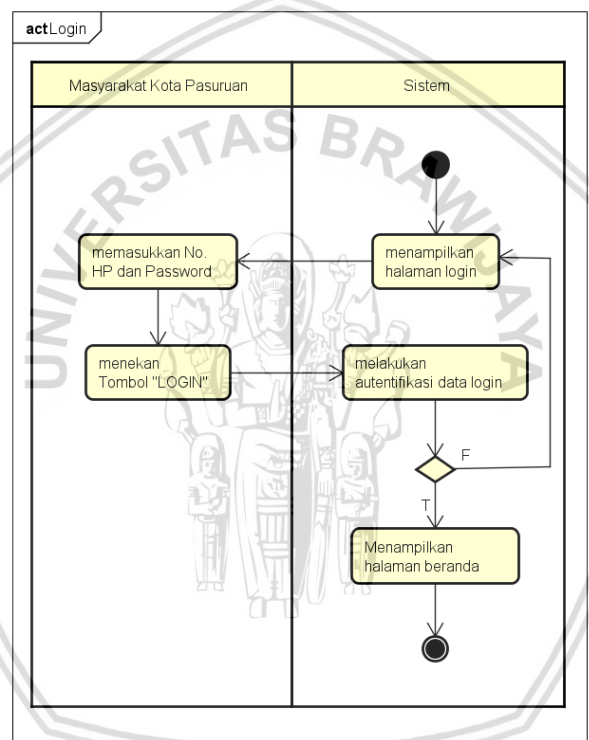
	<ol style="list-style-type: none"> <li>2. Sistem akan menampilkan form balas pesan pengaduan</li> <li>3. Petugas Administrasi mengisi pesan pada form balas pesan pengaduan</li> <li>4. Petugas Adminstrasi menekan tombol “Simpan Perubahan”</li> <li>5. Sistem menampilkan notifikasi bahwa “Pesan Balasan Pengaduan telah tersimpan!”</li> </ol> <p><b>Hapus Aduan</b></p> <ol style="list-style-type: none"> <li>1. Petugas Administrasi memilih icon “Hapus Aduan” pada kolom aksi</li> <li>2. Sistem menampilkan pemberitahuan “Yakin ingin hapus Aduan Pesan ID : [kode_pesan] ?”</li> <li>3. Petugas Adminstrasi menekan tombol “OKE”</li> <li>4. Sistem melakukan operasi penghapusan pesan pengaduan</li> <li>5. Sistem menampilkan notifikasi bahwa “Berhasil Terhapus : Pesan Pengaduan telah dihapus!”</li> </ol> <p><b>Ubah Status Aduan</b></p> <ol style="list-style-type: none"> <li>1. Petugas Administrasi memilih icon “Ubah Status Aduan” pada kolom aksi</li> <li>2. Sistem menampilkan pemberitahuan “Yakin ubah status Aduan [kode_pesan] ?”</li> <li>3. Petugas Adminstrasi menekan tombol “OKE”</li> <li>4. Sistem melakukan operasi pengubahan status pesan pengaduan</li> <li>5. Sistem menampilkan notifikasi bahwa “Berhasil Terupdate : Status Aduan telah berubah!”</li> </ol> <p><b>Cetak Aduan</b></p> <ol style="list-style-type: none"> <li>1. Petugas Administrasi memilih icon “Cetak PDF” pada kolom aksi</li> <li>2. Sistem menampilkan pelaporan pengaduan dalam format PDF</li> </ol>
<b>Post-Conditiions</b>	Petugas Administrasi berhasil mengelola Data Pengaduan

## 4.6 Activity Diagram

*Activity Diagram* merupakan diagram yang digunakan untuk menggambarkan alur kerja dari sebuah Spesifikasi *use case*.

### 4.6.1 Activity Diagram Login

*Activity Diagram* Login dimulai ketika Masyarakat Kota Pasuruan mengakses halaman login dan sistem menampilkan halaman *login*, kemudian Masyarakat Kota Pasuruan mengisi No.HP dan Password, kemudian sistem mengecek masukkan dari Masyarakat Kota Pasuruan. Jika data yang dimasukkan oleh Masyarakat Kota Pasuruan benar maka akan masuk ke dalam sistem, berikut ini merupakan *activity diagram* Login yang ditunjukkan pada gambar 4.6.



powered by Astah

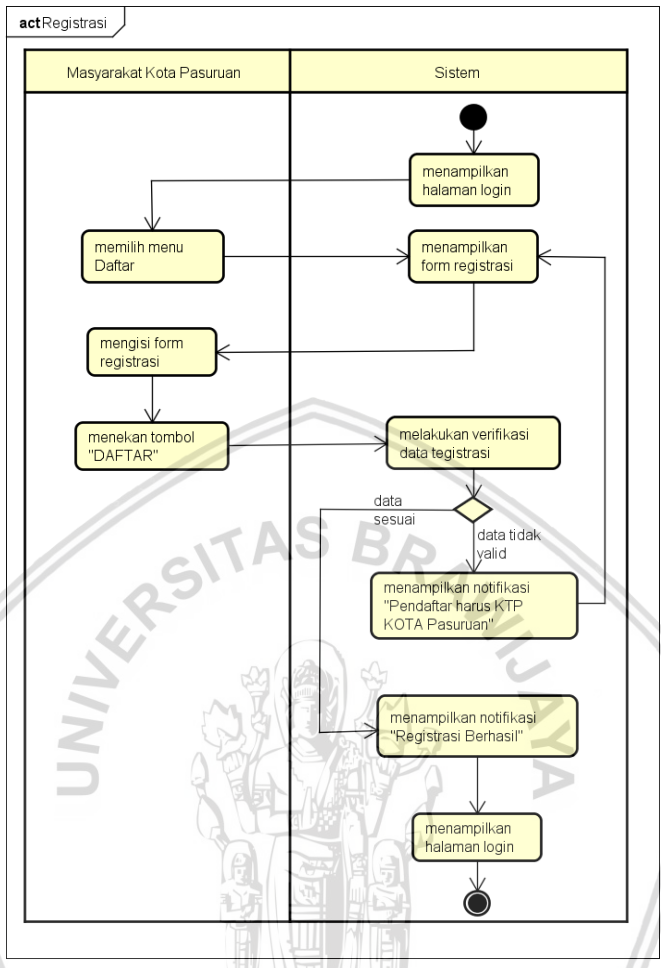
Gambar 4.6 Activity Diagram Login

### 4.6.2 Activity Diagram Registrasi

*Activity Diagram* Registrasi dimulai ketika Masyarakat Kota Pasuruan berada di halaman login, lalu Masyarakat Kota Pasuruan memilih menu Daftar. Dalam menu daftar, sistem akan menampilkan form Registrasi, kemudian Masyarakat Kota Pasuruan mengisi form registrasi sesuai data pribadi yang dimiliki. Sistem akan melakukan verifikasi data registrasi. Jika data yang diisi oleh Masyarakat Kota Pasuruan benar maka sistem akan menampilkan



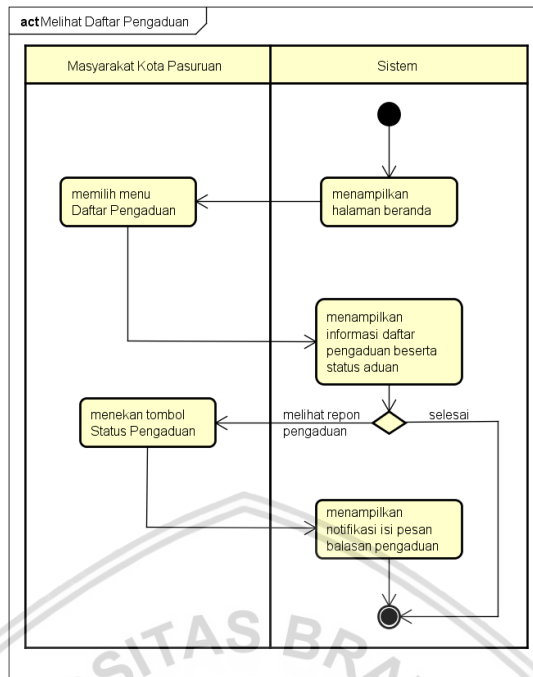
notifikasi. Berikut ini merupakan *activity diagram* Registrasi yang ditunjukkan pada gambar 4.7.



Gambar 4.7 Activity Diagram Registrasi

**4.6.3 Activity Diagram Melihat Daftar Pengaduan**

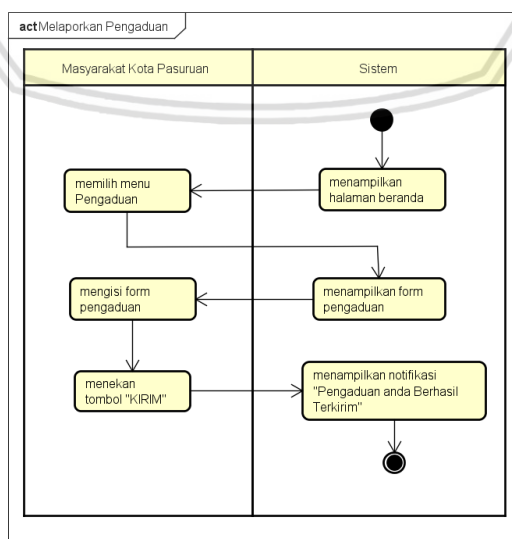
*Activity Diagram* Melihat Daftar Pengaduan dimulai ketika Masyarakat Kota Pasuruan berhasil masuk ke dalam sistem dan berada dalam menu beranda. Kemudian Masyarakat Kota Pasuruan memilih menu Daftar Pengaduan, maka sistem akan menampilkan informasi daftar data pengaduan beserta status pengaduannya, sedangkan untuk melihat pesan balasan pengaduan selanjutnya Masyarakat Kota Pasuruan menekan tombol status pengaduan dan sistem akan menampilkan notifikasi isi pesan balasan pengaduan. Berikut ini merupakan *Activity diagram* Melihat Daftar Pengaduan yang ditunjukkan pada gambar 4.8.



Gambar 4.8 Activity Diagram Melihat Daftar Pengaduan

#### 4.6.4 Activity Diagram Melaporkan Pengaduan

Activity Diagram Melaporkan Pengaduan dimulai ketika Masyarakat Kota Pasuruan berhasil masuk ke dalam sistem dan berada dalam menu beranda. Kemudian Masyarakat Kota Pasuruan memilih menu Pengaduan, sistem akan menampilkan form Pengaduan. Kemudian Masyarakat Kota Pasuruan mengisi form Pengaduan. Setelah mengisi form pengaduan maka Masyarakat Kota Pasuruan menekan tombol KIRIM selanjutnya sistem akan menampilkan notifikasi. Berikut ini merupakan *activity diagram* Melaporkan Pengaduan yang ditunjukkan pada gambar 4.9.

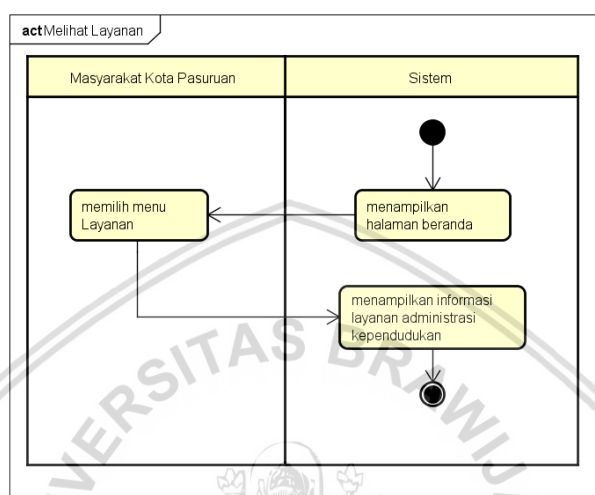


Gambar 4.9 Activity Diagram Melaporkan Pengaduan



#### 4.6.5 Activity Diagram Melihat Layanan

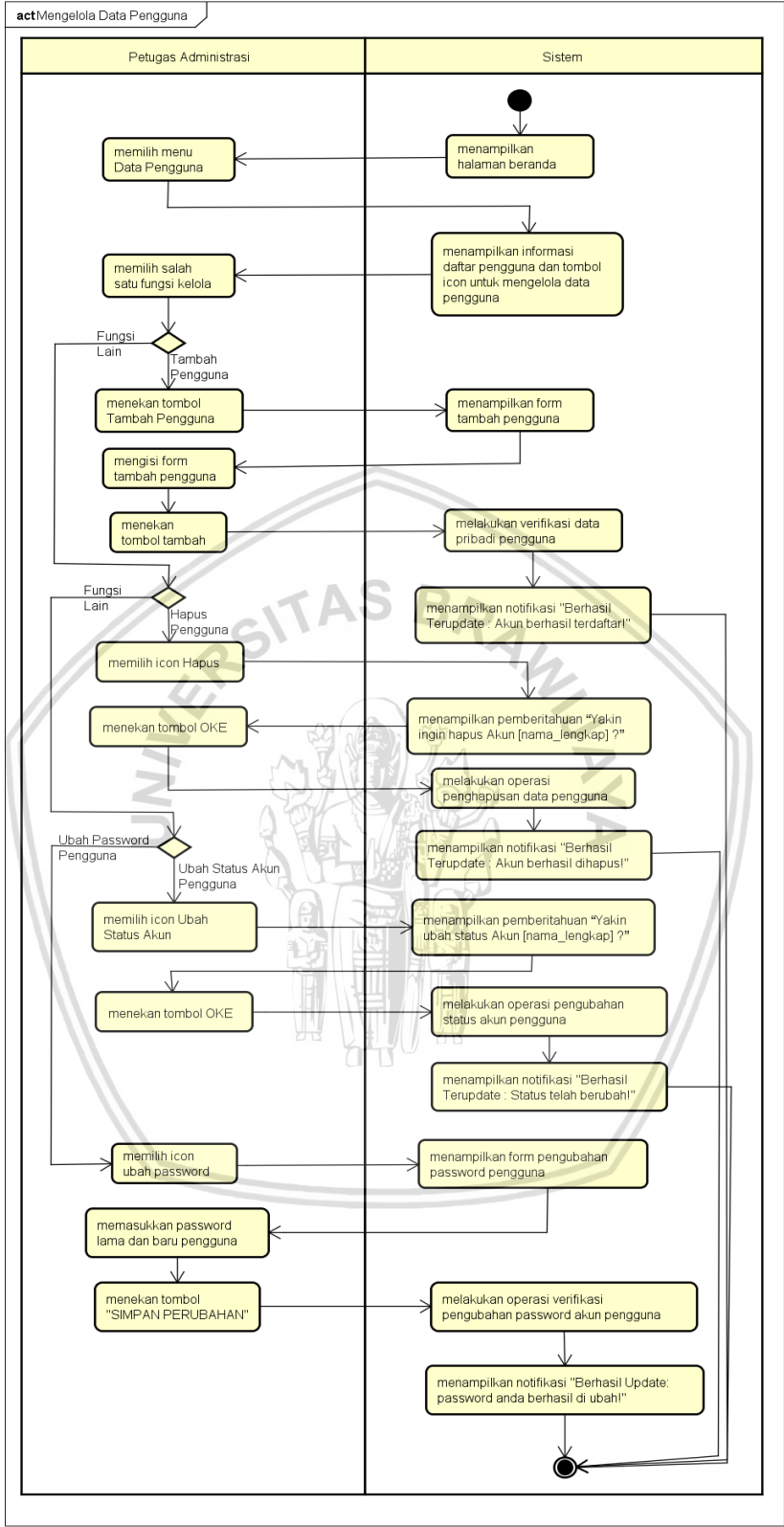
*Activity Diagram* Melihat Daftar Layanan dimulai ketika Masyarakat Kota Pasuruan berhasil masuk ke dalam sistem dan berada dalam menu beranda. Kemudian Masyarakat Kota Pasuruan memilih menu Layanan, maka sistem akan menampilkan informasi layanan administrasi kependudukan. Berikut ini merupakan *Activity Diagram* Melihat Layanan yang ditunjukkan pada gambar 4.10.



Gambar 4.10 Activity Diagram Melihat Layanan

#### 4.6.6 Activity Diagram Mengelola Data Pengguna

*Activity Diagram* Mengelola Data Pengguna dimulai ketika Petugas Administrasi berhasil masuk dalam sistem dan berada dalam menu beranda, lalu Petugas Administrasi memilih menu Data Pengguna. Sistem akan menampilkan informasi daftar pengguna yang terdaftar dalam sistem dan beberapa fungsi kelola yang dapat Petugas Administrasi pilih. Berikut ini merupakan *Activity diagram* Mengelola Data Pengguna yang ditunjukkan pada gambar 4.11.



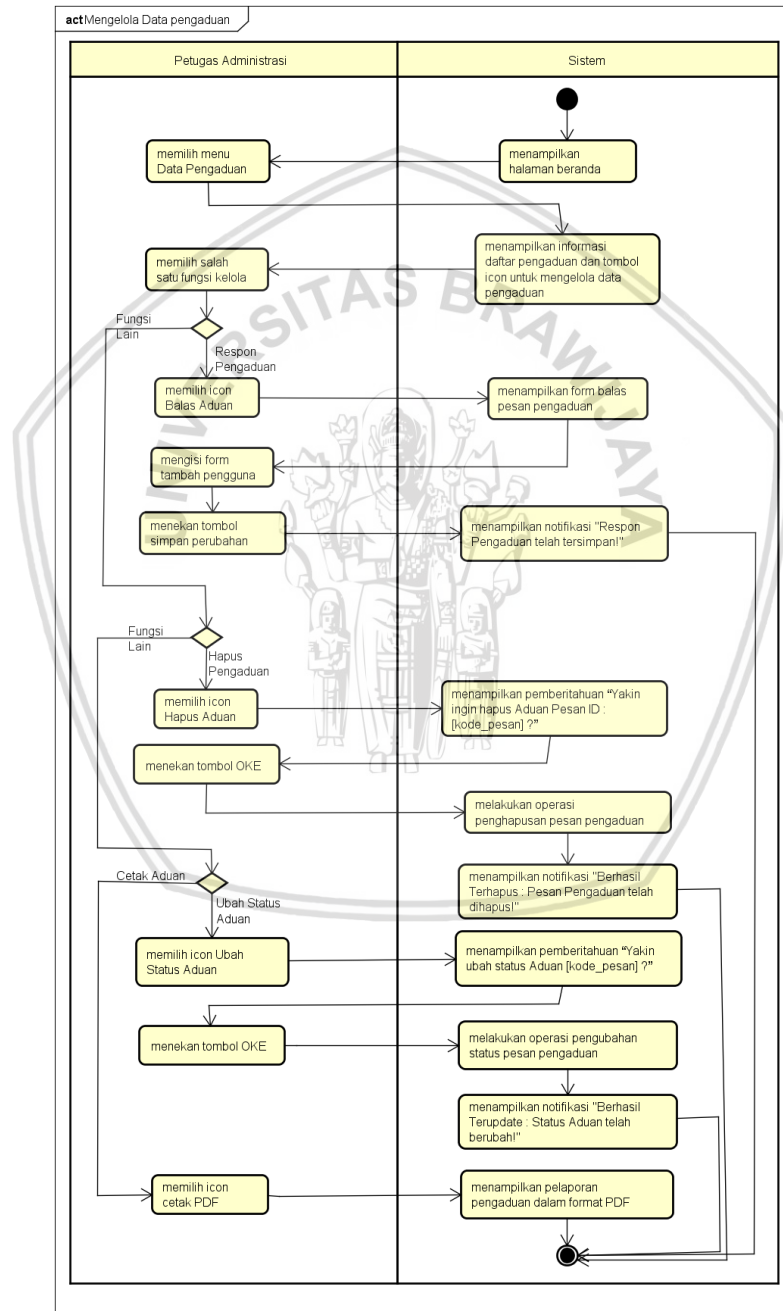
powered by Astah

Gambar 4.11 Activity Diagram Mengelola Data Pengguna



### 4.6.7 Activity Diagram Mengelola Data Pengaduan

Activity Diagram Mengelola Data Pengaduan dimulai ketika Petugas Administrasi berhasil masuk dalam sistem dan berada dalam menu beranda, lalu Petugas Administrasi memilih menu Data Pengaduan. Sistem akan menampilkan informasi daftar pengaduan beserta status pengaduannya dan beberapa fungsi kelola yang dapat Petugas Administrasi pilih. Berikut ini merupakan Activity diagram Mengelola Data Pengguna yang ditunjukkan pada gambar 4.12.



powered by Astah

Gambar 4.12 Activity Diagram Mengelola Data Pengaduan





## BAB 5 PERANCANGAN

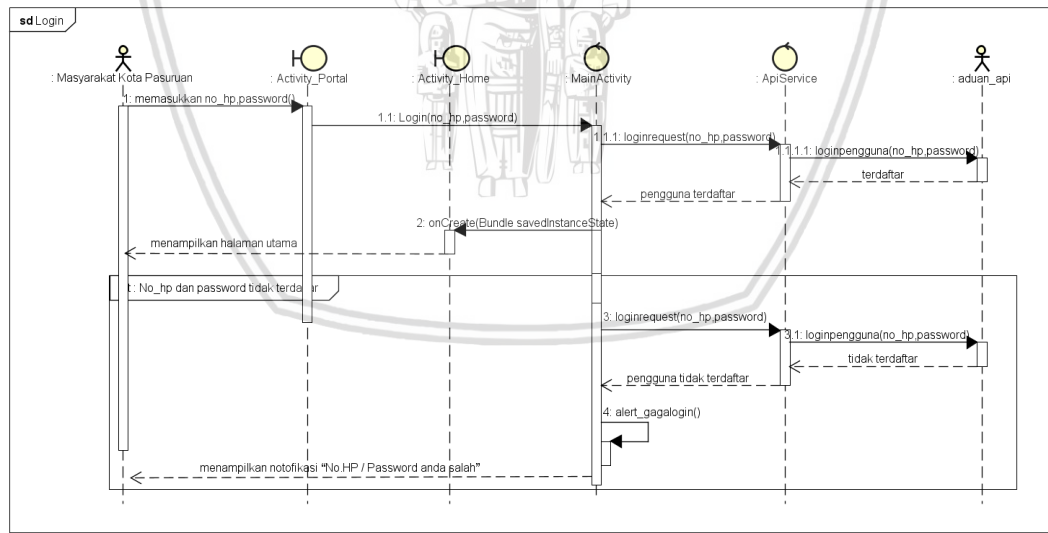
Bab ini membahas mengenai perancangan sistem pada penelitian ini dilakukan dengan pendekatan berorientasi objek, sehingga pemodelan sistem dibuat dengan menggunakan diagram UML (*Unified Modelling Language*). Sehingga perancangan dari sistem yang akan diimplementasikan terdiri dari *sequence diagram*, *class diagram*, perancangan komunikasi data, pemodelan data, serta perancangan antarmuka sistem. Hasil dari perancangan ini nantinya akan digunakan sebagai dasar untuk membangun sistem Layanan Pengaduan Masyarakat di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.

### 5.1 Perancangan *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek yang terjadi dalam kurun waktu tertentu yang dibuat berdasarkan *use case diagram* sebelumnya. *Sequence diagram* yang digambarkan tidak mencakup seluruh fungsi pada *use case*, hanya beberapa fungsi saja yang digambarkan. Berikut ini merupakan *sequence diagram* dari sistem Layanan Pengaduan Masyarakat.

#### 5.1.1 *Sequence Diagram Login*

*Sequence diagram Login* ini dibuat menggunakan *activity diagram* dari Gambar 4.6 sebagai acuan. *Sequence diagram* dari fungsi Login dapat dilihat pada Gambar 5.1.



Gambar 5.1 *Sequence Diagram Login*

*Sequence diagram Login* dimulai ketika Masyarakat Kota Pasuruan mengakses sistem dan sistem menampilkan form login kemudian Masyarakat Kota Pasuruan memasukkan no.hp dan *password*. Masukan dari Masyarakat Kota Pasuruan akan di lanjutkan ke dalam *controller MainActivity* dengan memanggil fungsi *Login()* dan

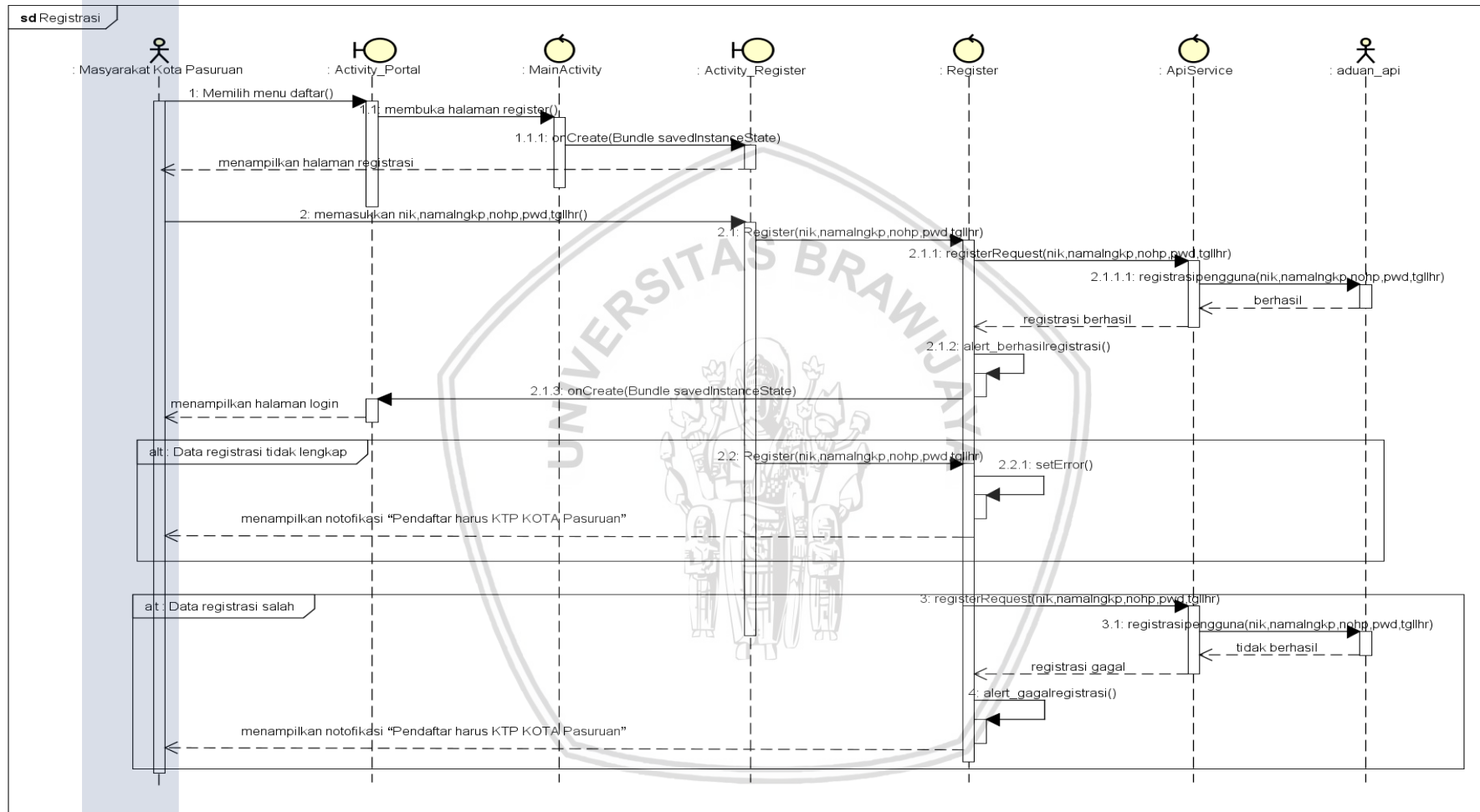


dilanjutkan kembali ke *controller* ApiService dengan memanggil fungsi `loginrequest()` untuk menampung sementara data pengguna. *Controller* ApiService memanggil *webservice* `aduan_api` dengan fungsi `member()` untuk dilakukan pengecekan dalam sistem data login yang dikirimkan. Jika hasil pengecekan pengguna terdaftar maka *controller* MainActivity akan memanggil fungsi `onCreate()` pada *boundary* Activity\_Home untuk menampilkan halaman utama sistem, Apabila hasil pengecekan pengguna tidak terdaftar maka *controller* MainActivity akan memanggil fungsi `alert_gagallogin()` untuk menampilkan pesan *error*.

### 5.1.2 Sequence Diagram Registrasi

*Sequence diagram* Registrasi ini dibuat menggunakan *activity diagram* dari Gambar 4.7 sebagai acuan. *Sequence diagram* dari fungsi Registrasi dapat dilihat pada Gambar 5.2



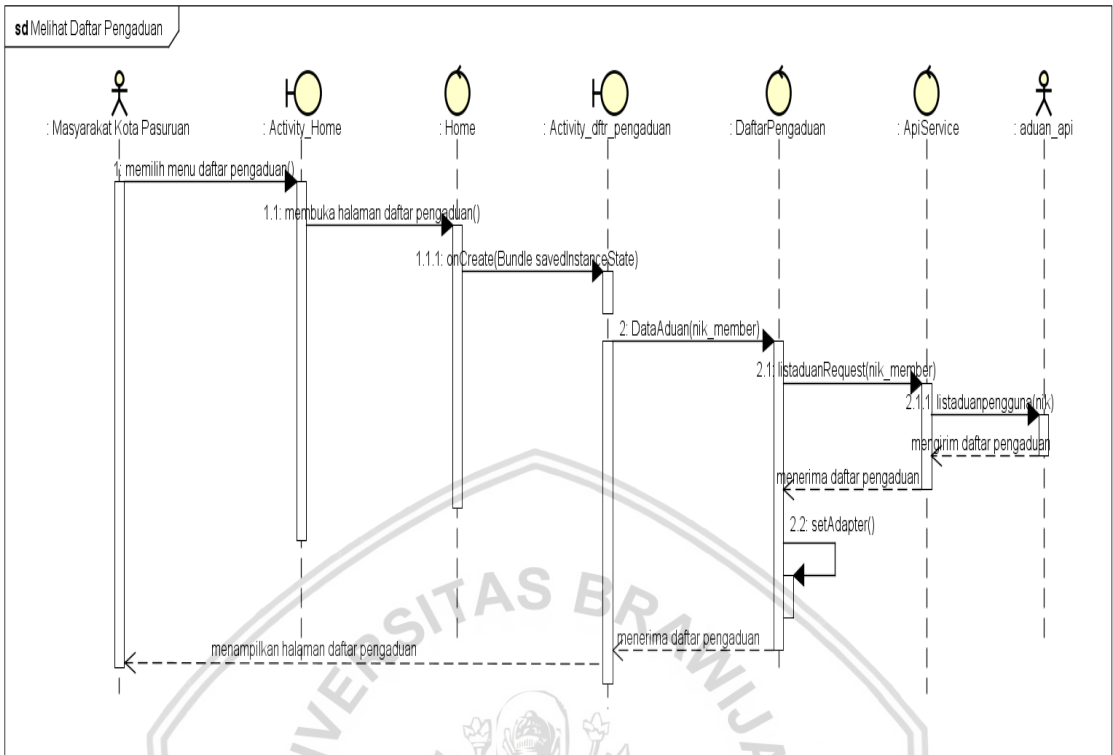


Gambar 5.2 Sequence Diagram Registrasi

*Sequence diagram* Registrasi dimulai ketika Masyarakat Kota Pasuruan memilih menu daftar pada *boundary* Activity\_Portal. Activity\_Portal memanggil *controller* MainActivity untuk membuka halaman registrasi. *Controller* MainActivity akan memanggil fungsi onCreate() pada *boundary* Activity\_Register sehingga akan menampilkan halaman registrasi, kemudian Masyarakat Kota Pasuruan mengisi form registrasi. Masukan data registrasi dari Masyarakat Kota Pasuruan akan dilanjutkan ke dalam *controller* Register dengan memanggil fungsi register() dan dilanjutkan kembali ke *controller* ApiService dengan memanggil fungsi registerRequest() untuk menampung sementara data registrasi. *Controller* ApiService memanggil *webservice* aduan\_api dengan fungsi daftarmember() untuk dilakukan verifikasi dan menyimpan data registrasi ke dalam sistem. Jika proses registrasi berhasil maka *controller* Register akan memanggil fungsi alert\_berhasilregistrasi() sehingga menampilkan pesan registrasi berhasil dan akan memanggil fungsi onCreate() pada *boundary* Activity\_Portal untuk menampilkan kembali halaman login, Apabila proses registrasi gagal maka *controller* Register akan memanggil fungsi alert\_gagalregistrasi() untuk menampilkan pesan error.

### 5.1.3 *Sequence Diagram* Melihat Daftar Pengaduan

*Sequence diagram* Melihat Daftar Pengaduan ini dibuat menggunakan *activity diagram* dari Gambar 4.8 sebagai acuan. *Sequence diagram* dari fungsi Melihat Daftar Pengaduan dapat dilihat pada Gambar 5.3.



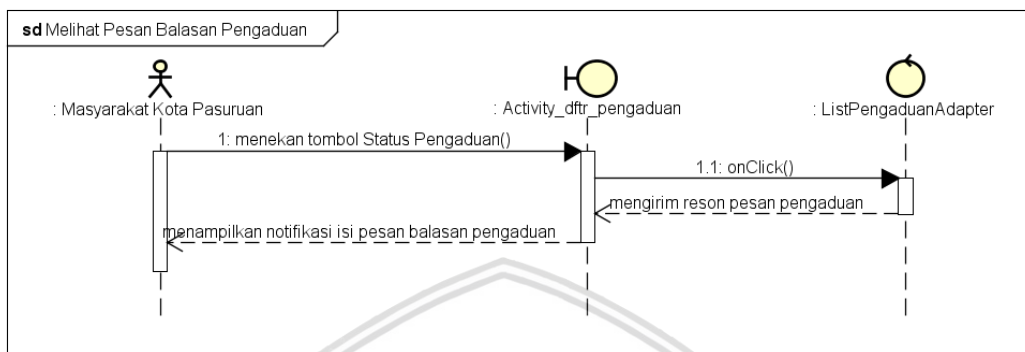
powered by Astah

**Gambar 5.3 Sequence Diagram Melihat Daftar Pengaduan**

*Sequence diagram* Melihat Daftar Pengaduan dimulai ketika Masyarakat Kota Pasuruan memilih menu daftar pengaduan pada *boundary* Activity\_Home. *Boundary* Activity\_Home memanggil *controller* Home untuk membuka halaman daftar pengaduan. *Controller* Home akan memanggil fungsi onCreate pada *boundary* Activity\_dfr\_pengaduan untuk menampilkan halaman daftar pengaduan, sebelum menampilkannya terlebih dahulu *boundary* Activity\_dfr\_pengaduan memanggil fungsi DataAduan() pada *controller* DaftarPengaduan dan dilanjutkan ke *controller* ApiService dengan memanggil fungsi listaduanRequest() untuk meminta data daftar pengaduan ke pada *webserive* aduan\_api dengan memanggil fungsi listaduanmember(). *Controller* DaftarPengaduan memanggil fungsi setAdapter() untuk menata daftar pengaduan menjadi daftar tersusun maka secara otomatis menampilkan halaman daftar pengaduan.

A. *Sequence Diagram* Melihat Pesan Balasan Pengaduan

*Sequence Diagram* Melihat Balasan Pengaduan merupakan *subflows* dari *Activity Diagram* Melihat Daftar Pengaduan pada Gambar 4.8. *Sequence diagram* dari fungsi Melihat Pesan Balasan Pengaduan dapat dilihat pada Gambar 5.4.



powered by Astah

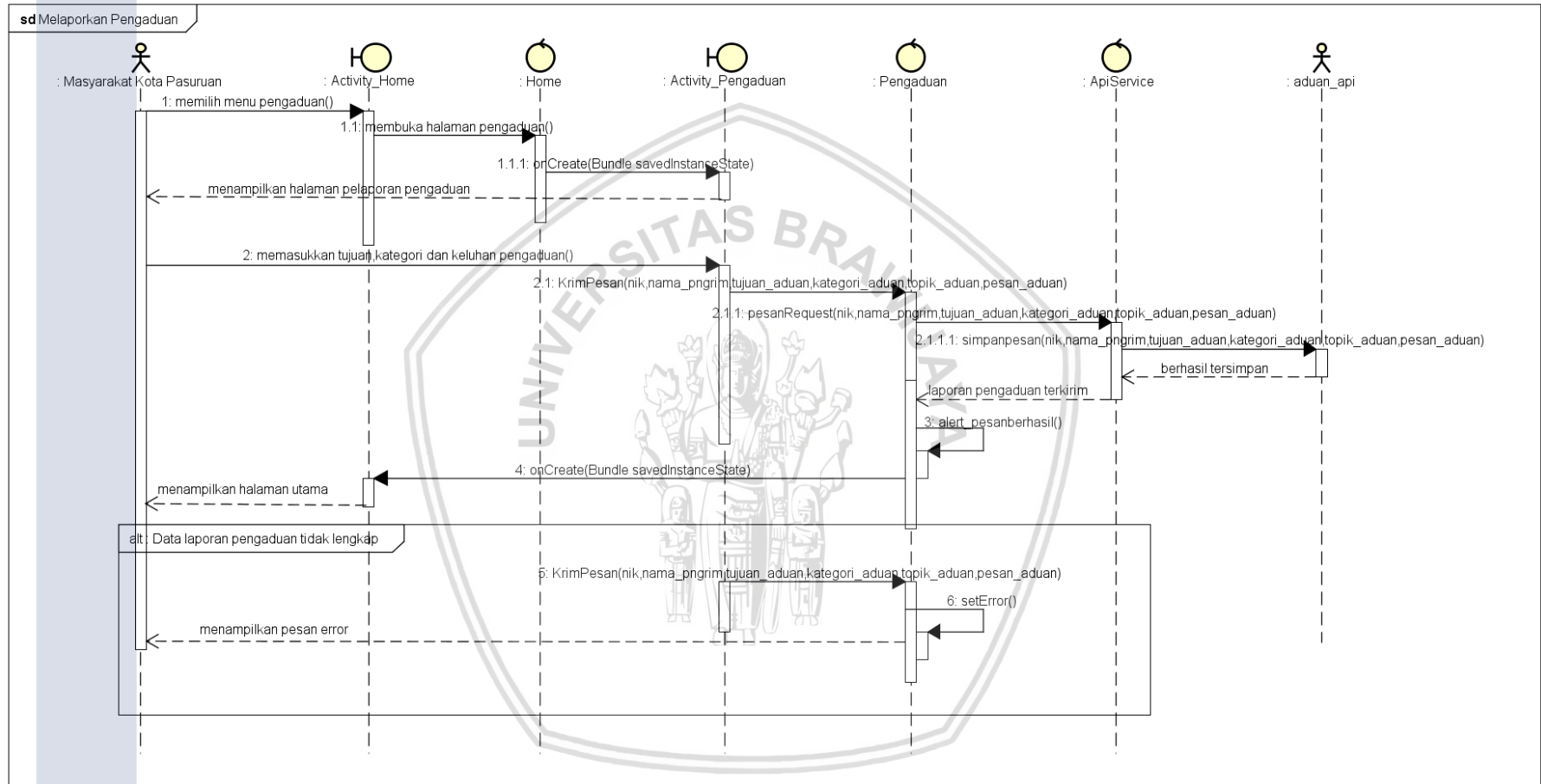
**Gambar 5.4 *Sequence Diagram* Melihat Pesan Balasan Pengaduan**

*Sequence diagram* Melihat Pesan Balasan Pengaduan dimulai ketika pada halaman daftar pengaduan, Masyarakat Kota Pasuruan menekan tombol status pengaduan pada *boundary* Activity\_dftr\_pengaduan, selanjutnya *boundary* Activity\_dftr\_pengaduan memanggil fungsi onClick() pada *controller* ListPengaduanAdapter maka akan menampilkan notifikasi berupa isi pesan balasan dari pengaduan tersebut.

**5.1.4 *Sequence Diagram* Melaporkan Pengaduan**

*Sequence diagram* Login ini dibuat menggunakan *activity diagram* dari Gambar 4.9 sebagai acuan. *Sequence diagram* dari fungsi Login dapat dilihat pada Gambar 5.5.



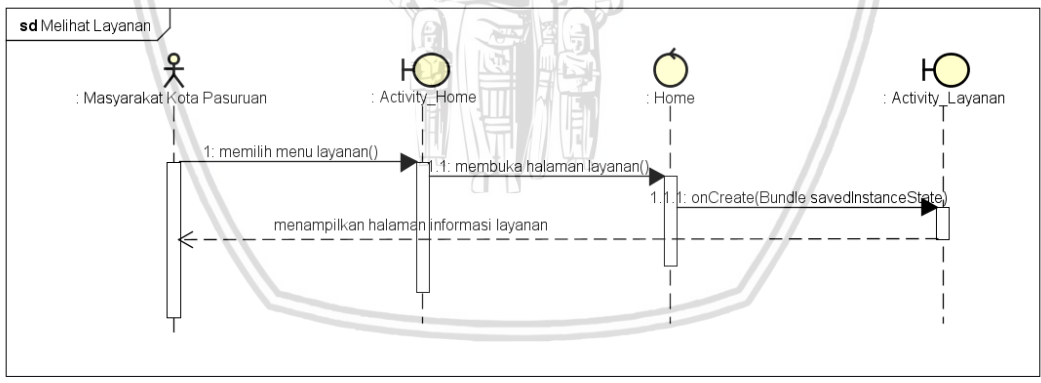


Gambar 5.5 Sequence Diagram Melaporkan Pengaduan

*Sequence diagram* Melaporkan Pengaduan dimulai ketika Masyarakat Kota Pasuruan memilih menu pengaduan pada *boundary* Activity\_Home. *Boundary* Activity\_Home memanggil *controller* Home untuk membuka halaman pengaduan. *Controller* Home akan memanggil fungsi onCreate() pada *boundary* Activity\_Pengaduan sehingga akan menampilkan halaman pengaduan, kemudian Masyarakat Kota Pasuruan mengisi form pengaduan. Masukan data pengaduan dari Masyarakat Kota Pasuruan akan dilanjutkan ke dalam *controller* Pengaduan dengan memanggil fungsi kirimPesan() dan dilakukan pengecekan masukan data pengaduan. Jika terdapat data pengaduan yang tidak lengkap maka *controller* Pengaduan akan memanggil fungsi setError() untuk menampilkan pesan *error*, sedangkan data pengaduan lengkap akan dilanjutkan ke *controller* ApiService dengan memanggil fungsi pesanRequest() untuk menampung sementara data pengaduan dan akan dilanjutkan memanggil *webservice* aduan\_api dengan fungsi simpanpesan() untuk menyimpan laporan pengaduan. *Controller* Pengaduan akan memanggil fungsi alert\_pesanberhasil() untuk menampilkan notifikasi pesan laporan pengaduan berhasil terkirim dan memanggil fungsi onCreate() pada *boundary* Activity\_Home untuk menampilkan kembali halaman utama sistem.

**5.1.5 Sequence Diagram Melihat Layanan**

*Sequence diagram* Login ini dibuat menggunakan *activity diagram* dari Gambar 4.10 sebagai acuan. *Sequence diagram* dari fungsi Login dapat dilihat pada Gambar 5.6



powered by Astah

**Gambar 5.6 Sequence Diagram Melihat Layanan**

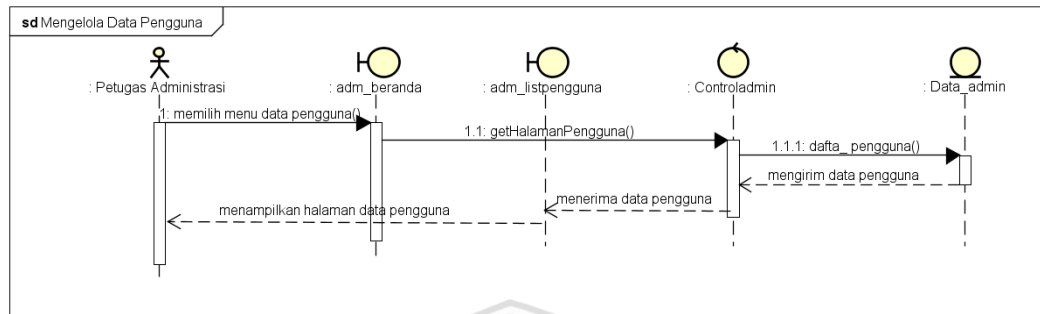
*Sequence diagram* Melihat Layanan dimulai ketika Masyarakat Kota Pasuruan memilih menu layanan pada *boundary* Activity\_Home. *Boundary* Activity\_Home memanggil *controller* Home untuk membuka halaman layanan. *Controller* Home akan memanggil fungsi onCreate() pada *boundary* Activity\_Layanan sehingga akan menampilkan halaman informasi layanan.





### 5.1.6 Sequence Diagram Mengelola Data Pengguna

Sequence diagram Login ini dibuat menggunakan activity diagram dari Gambar 4.11 pada bab sebelumnya sebagai acuan. Sequence diagram dari fungsi Login dapat dilihat pada Gambar 5.7.



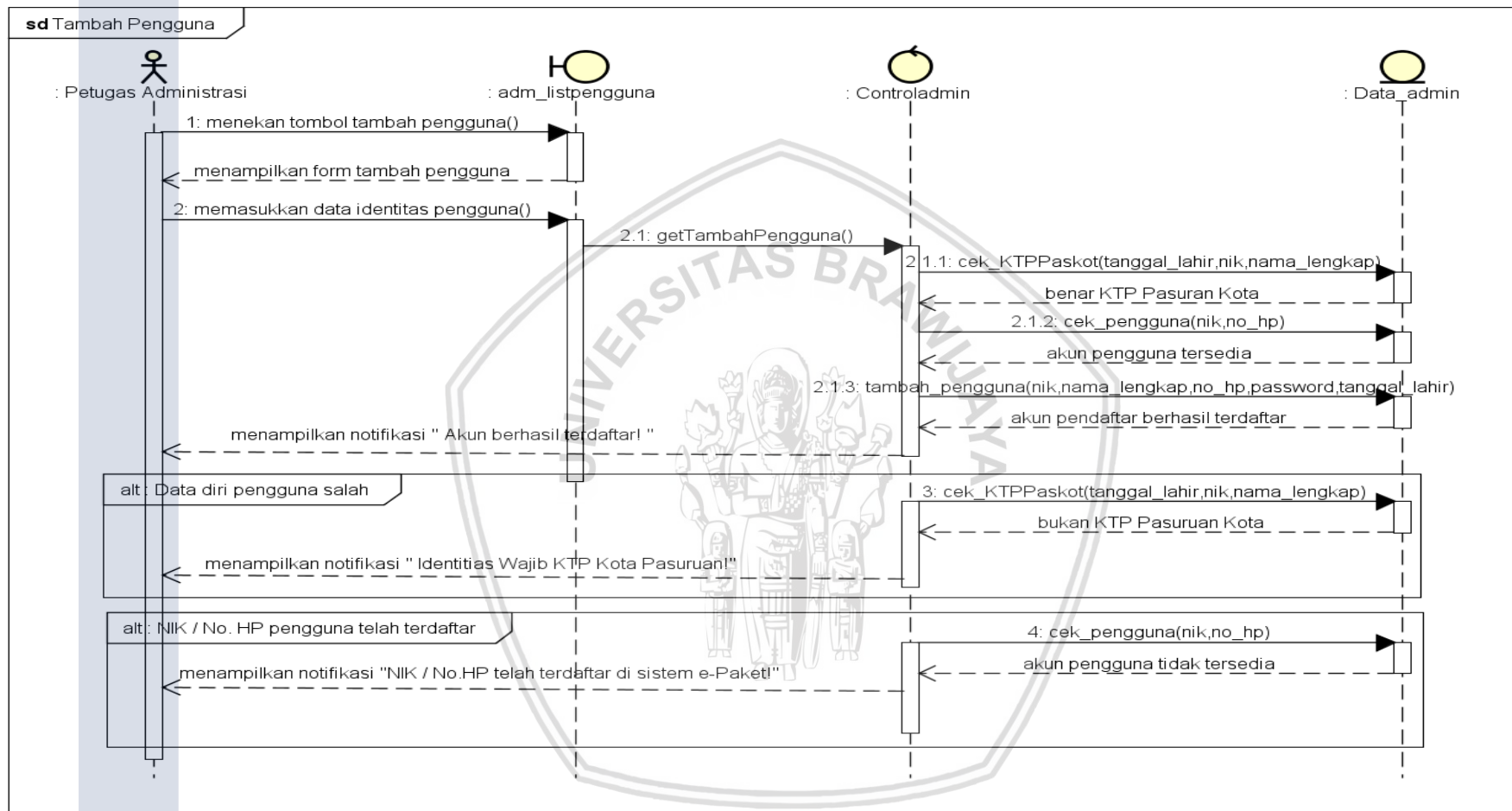
powered by Astah

Gambar 5.7 Sequence Diagram Mengelola Data Pengguna

Sequence diagram Mengelola Data Pengguna dimulai ketika Petugas Administrasi memilih menu data pengguna pada *boundary* adm\_beranda. *Boundary* adm\_beranda memanggil *controller* Controladmin untuk menampilkan halaman data pengguna, sebelum menampilkannya terlebih dahulu *controller* Controladmin memanggil fungsi all\_member() pada *entity* Data\_admin untuk mengambil seluruh daftar pengguna yang telah terdaftar. Sehingga akan menampilkan halaman dengan informasi daftar pengguna

#### A. Sequence Diagram Tambah Pengguna

Sequence Diagram Tambah Pengguna merupakan subflows dari Activity Diagram Mengelola Data Pengguna pada Gambar 4.11. Sequence diagram dari fungsi Melihat Pesan Balasan Pengaduan dapat dilihat pada Gambar 5.8.

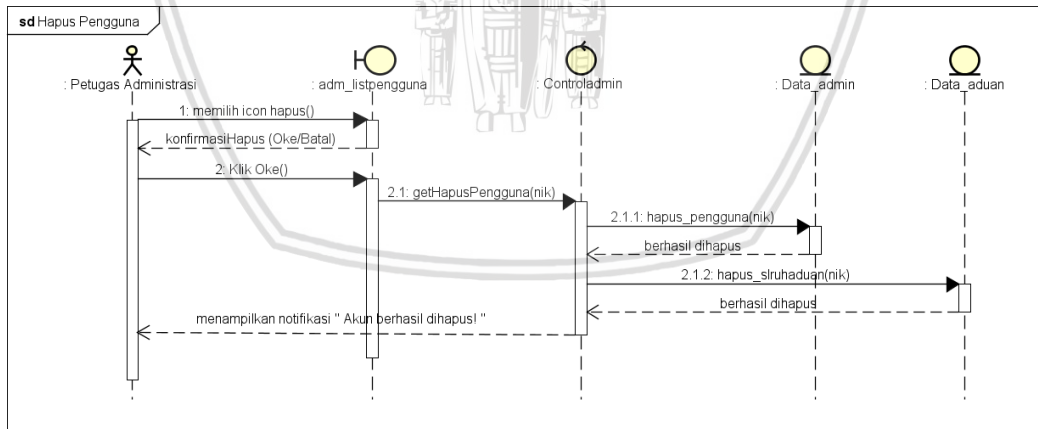


Gambar 5.8 Sequence Diagram Tambah Pengguna

*Sequence diagram* Tambah Pengguna dimulai ketika pada halaman daftar pengguna, Petugas Administrasi menekan tombol tambah pengguna pada *boundary* adm\_listpengguna, selanjutnya *boundary* adm\_listpengguna menampilkan form tambah pengguna. Petugas Administrasi mengisi identitas pengguna pada form tambah pengguna, kemudian data identitas pengguna masukan dari Petugas Administrasi akan dilanjutkan ke dalam *controller* Controladmin dengan memanggil fungsi getTambahMember() untuk menampung sementara data identitas pengguna. *Controller* Controladmin sebelum disimpan pada sistem akan memanggil *entity* Data\_admin untuk melakukan verifikasi dan pengecekan. *Entity* Data\_admin melakukan verifikasi data identitas pengguna ber-KTP Kota Pasuruan dengan memanggil fungsi cek\_KTPPaskot(). Jika verifikasi benar maka dilanjutkan dengan melakukan pengecekan apakah member tersedia atau tidak, *entity* Data\_admin akan memanggil fungsi cek\_member(). Jika pengecekan tersedia maka dilanjutkan tahap yang terakhir yaitu menyimpan data identitas pengguna pada sistem maka *entity* Data\_admin akan memanggil fungsi tambah\_member() kemudian akan menampilkan notifikasi “berhasil terdaftar”. Apabila saat melakukan verifikasi identitas pengguna ternyata bukan KTP Kota Pasuruan ataupun saat pengecekan ternyata tidak tersedia maka *controller* Controladmin akan menampilkan notifikasi gagal.

**B. Sequence Diagram Hapus Pengguna**

*Sequence Diagram* Hapus Pengguna merupakan *subflows* dari *Activity Diagram* Mengelola Data Pengguna pada Gambar 4.11. *Sequence diagram* dari fungsi Hapus Pengguna dapat dilihat pada Gambar 5.9.



powered by Astah

**Gambar 5.9 Sequence Diagram Hapus Pengguna**

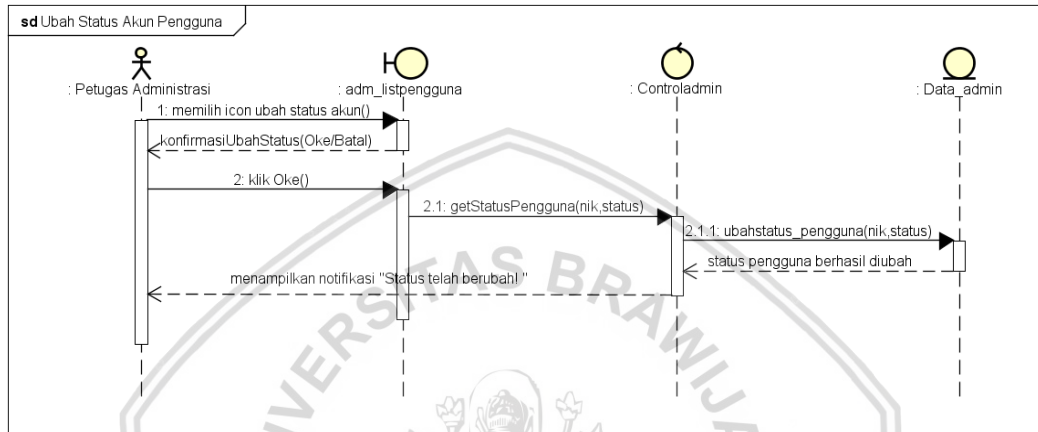
*Sequence diagram* Hapus Pengguna dimulai ketika pada halaman daftar pengguna, Petugas Administrasi memilih icon hapus pada *boundary* adm\_listpengguna. *Boundary* adm\_listpengguna akan menampilkan pop-up konfrimasihapus kemudian Petugas Administrasi mengklik oke, selanjutnya akan dilanjutkan *controller* Controladmin dengan memanggil fungsi getHapusMember() untuk menghapus akun pengguna. *Controller*



Controladmin akan memanggil *entity* Data\_admin dengan memanggil fungsi hapus\_member dan fungsi hapusall\_aduanmember() untuk menghapus dari daftar data pengguna maupun daftar pengaduan didalam sistem. Selanjutnya *controller* Controladmin akan menampilkan notifikasi berhasil dihapus.

C. *Sequence Diagram* Ubah Status Akun Pengguna

*Sequence Diagram* Tambah Pengguna merupakan *subflows* dari *Activity Diagram* Mengelola Data Pengguna pada Gambar 4.11. *Sequence diagram* dari fungsi Ubah Status Akun Pengguna dapat dilihat pada Gambar 5.10.

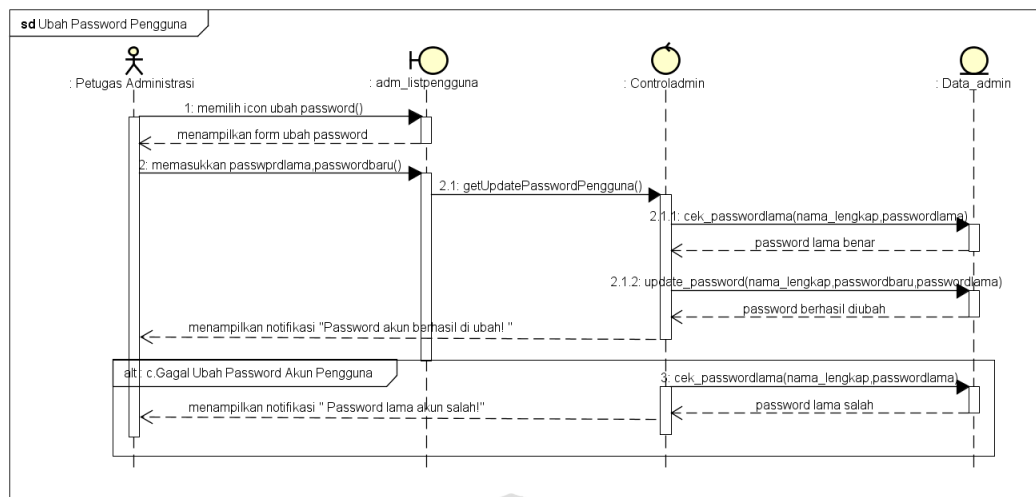


**Gambar 5.10 Sequence Diagram Ubah Status Akun Pengguna**

*Sequence diagram* Ubah Status Akun Pengguna dimulai ketika pada halaman daftar pengguna, Petugas Administrasi memilih icon ubah status akun pada *boundary* adm\_listpengguna. *Boundary* adm\_listpengguna akan menampilkan pop-up konfirmasiUbahStatus kemudian Petugas Administrasi mengklik oke, selanjutnya akan dilanjutkan *controller* Controladmin dengan memanggil fungsi getStatusMember() untuk mengubah status akun pengguna. *Controller* Controladmin akan memanggil *entity* Data\_admin dengan memanggil fungsi ubahstatus\_member() untuk mengubah status akun pengguna didalam sistem. Selanjutnya *controller* Controladmin akan menampilkan notifikasi status telah berubah.

D. *Sequence Diagram* Ubah Password Pengguna

*Sequence Diagram* Tambah Pengguna merupakan *subflows* dari *Activity Diagram* Mengelola Data Pengguna pada Gambar 4.11. *Sequence diagram* dari fungsi Ubah Password Pengguna Pengguna dapat dilihat pada Gambar 5.11.



powered by Astah

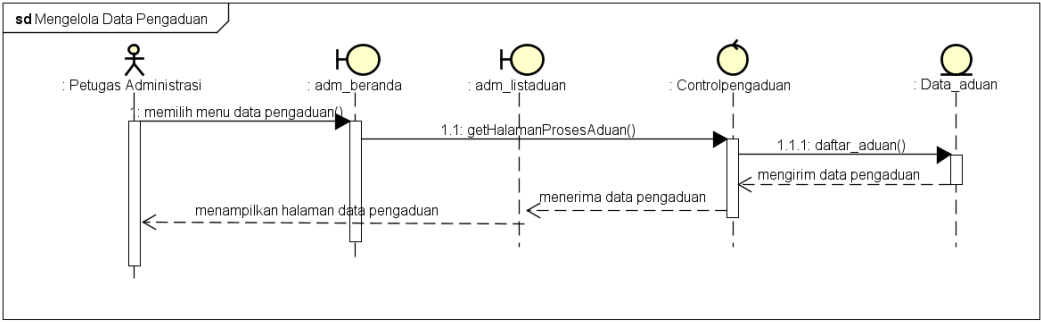
**Gambar 5.11 Sequence Diagram Ubah Password Pengguna**

*Sequence diagram* Ubah Password Pengguna dimulai ketika pada halaman daftar pengguna, Petugas Administrasi memilih icon ubah *password* pada *boundary* *adm\_listpengguna*. *Boundary* *adm\_listpengguna* akan menampilkan form ubah *password* kemudian Petugas Administrasi memasukkan data *password* lama dan baru. Masukan data *password* dari Petugas Administrasi akan dilanjutkan *controller* *Controladmin* dengan memanggil fungsi *getUpdatePasswordMember()* untuk mengubah *password* pengguna. *Controller* *Controladmin* akan memanggil *entity* *Data\_admin* untuk melakukan pengecekan *password* lama terlebih dahulu dengan memanggil fungsi *cek\_passwordlama()*. Jika pengecekan cocok akan dilanjutkan memanggil fungsi *update\_password()* maka *password* pengguna berhasil diubah dan *controller* *Controladmin* akan menampilkan notifikasi berhasil, sedangkan saat pengecekan tidak cocok maka *controller* *Controladmin* akan menampilkan notifikasi gagal.

### 5.1.7 Sequence Diagram Mengelola Data Pengaduan

*Sequence diagram* Mengelola Data Pengaduan ini dibuat menggunakan *activity diagram* dari Gambar 4.12 sebagai acuan. *Sequence diagram* dari fungsi Mengelola Data Pengaduan dapat dilihat pada Gambar 5.12.





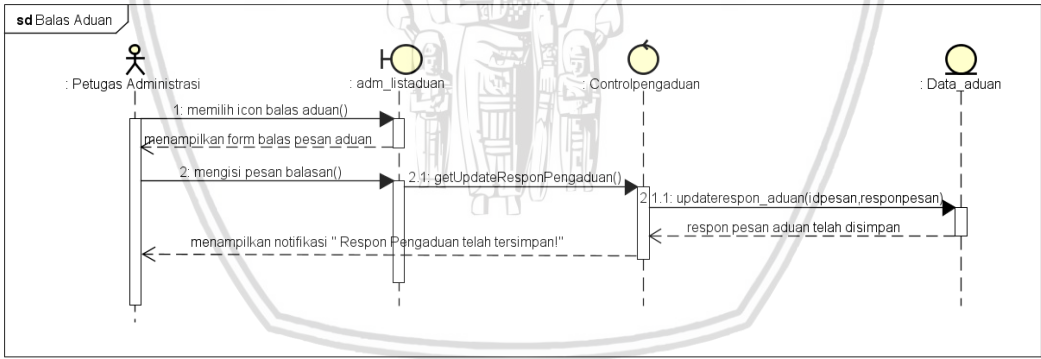
powered by Astah

**Gambar 5.12 Sequence Diagram Mengelola Data Pengaduan**

Sequence diagram Mengelola Data Pengaduan dimulai ketika Petugas Administrasi memilih menu data pengaduan pada *boundary* adm\_beranda. *Boundary* adm\_beranda memanggil *controller* Controlpengaduan untuk menampilkan halaman data pengaduan, sebelum menampilkannya terlebih dahulu *controller* Controlpengaduan memanggil fungsi all\_aduan() pada *entity* Data\_aduan untuk mengambil seluruh daftar pengaduan yang telah tersimpan. Sehingga akan menampilkan halaman dengan informasi daftar pengaduan.

**A. Sequence Diagram Balas Aduan**

Sequence Diagram Balas Aduan merupakan *subflows* dari Activity Diagram Mengelola Data Pengaduan pada Gambar 4.12. Sequence diagram dari fungsi Melihat Pesan Balasan Pengaduan dapat dilihat pada Gambar 5.13.



powered by Astah

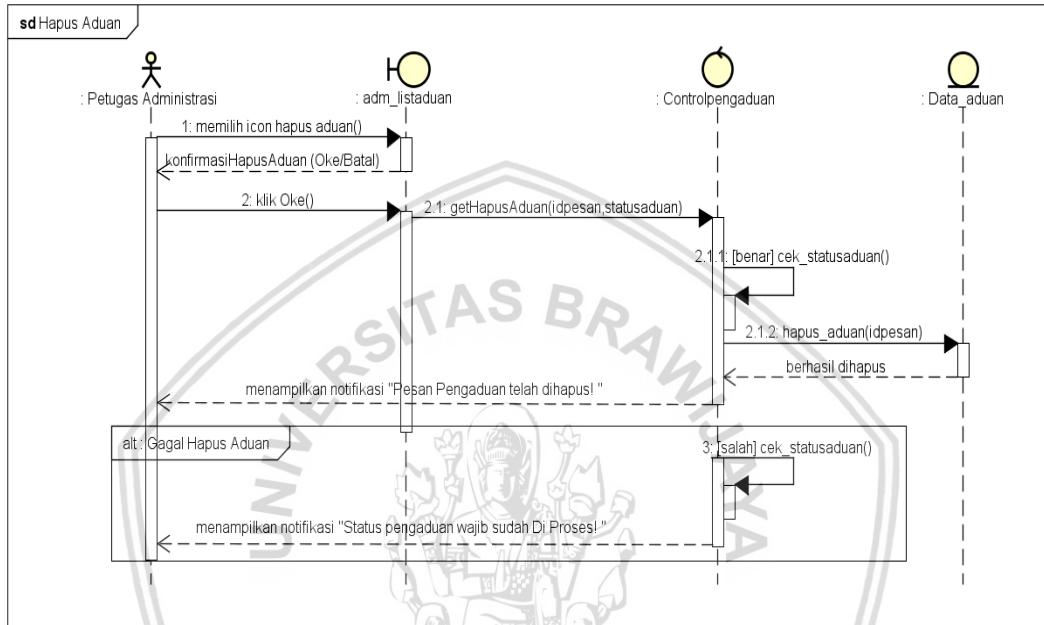
**Gambar 5.13 Sequence Diagram Balas Aduan**

Sequence diagram Balas Aduan dimulai ketika pada halaman daftar pengaduan, Petugas Administrasi memilih icon ubah balas aduan pada *boundary* adm\_listaduan. *Boundary* adm\_listaduan akan menampilkan form balas pesan aduan kemudian Petugas Administrasi mengisi pesan balasan. Masukan data pesan balasan dari Petugas Administrasi akan dilanjutkan *controller* Controlpengaduan dengan memanggil fungsi getUpdateresponPengaduan() untuk mengirim pesan balasan pengaduan terhadap laporan pengaduan Masyarakat Kota Pasuruan. *Controller* Controlpengaduan akan memanggil *entity* Data\_aduan untuk menyimpan

pesan balasan pengaduan pada sistem dengan memanggil fungsi `updateResponAdun()`. Selanjutnya *controller* `ControlPengaduan` akan menampilkan notifikasi berhasil.

**B. Sequence Diagram Hapus Aduan**

*Sequence Diagram* Hapus Aduan merupakan *subflows* dari *Activity Diagram* Mengelola Data Pengaduan pada Gambar 4.12. *Sequence diagram* dari fungsi Melihat Pesan Balasan Pengaduan dapat dilihat pada Gambar 5.14.



powered by Astah

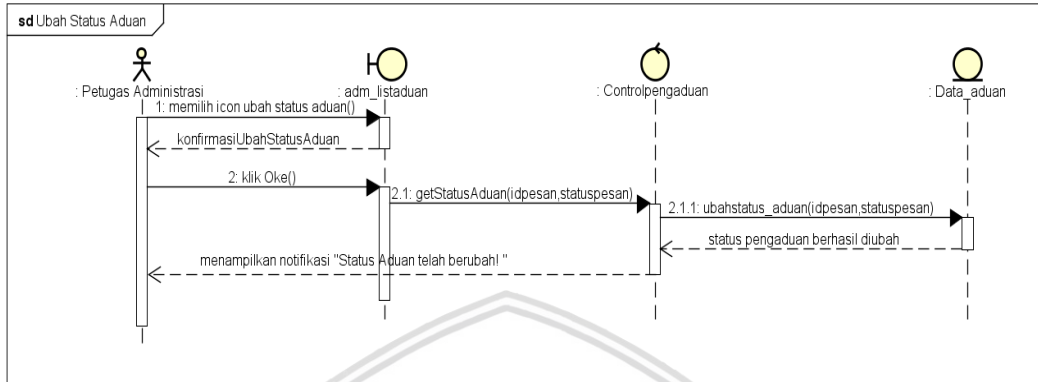
**Gambar 5.14 Sequence Diagram Hapus Aduan**

*Sequence diagram* Hapus Aduan dimulai ketika pada halaman daftar pengaduan, Petugas Administrasi memilih icon hapus aduan pada *boundary* `adm_listaduan`. *Boundary* `adm_listaduan` akan menampilkan pop-up `konfirmasiHapusAduan` kemudian Petugas Administrasi mengklik oke. *Controller* `Controlpengaduan` akan memanggil fungsi `getHapusAduan()` untuk menghapus laporan pengaduan tetapi terlebih dahulu melakukan pengecekan status pengaduan. Jika status pengaduan sudah diproses maka *controller* `Controlpengaduan` akan memanggil *Entity* `Data_aduan` dengan fungsi `hapus_aduan()` sehingga pengaduan tersebut akan terhapus dari sistem dan *controller* `Controlpengaduan` akan menampilkan notifikasi berhasil, Sedangkan status pengaduan belum diproses maka *controller* `Controlpengaduan` akan menampilkan notifikasi gagal.



C. Sequence Diagram Ubah Status Aduan

Sequence Diagram Ubah Status Aduan merupakan subflows dari Activity Diagram Mengelola Data Pengaduan pada Gambar 4.12. Sequence diagram dari fungsi Melihat Pesan Balasan Pengaduan dapat dilihat pada Gambar 5.15.



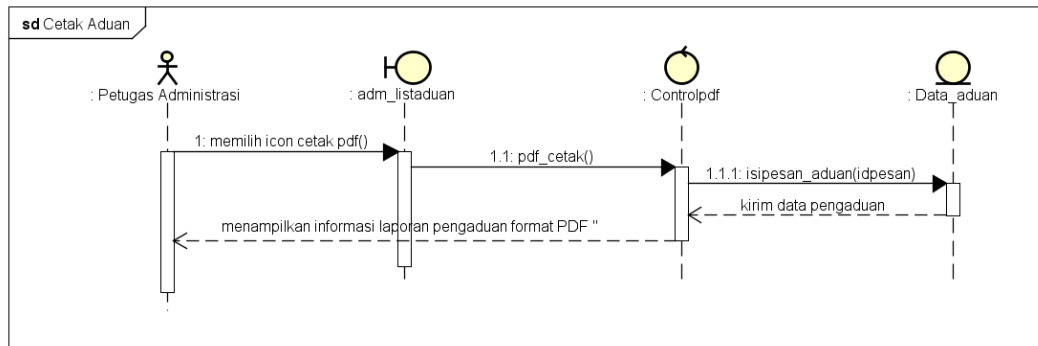
powered by Astah

Gambar 5.15 Sequence Diagram Ubah Status Aduan

Sequence diagram Ubah Status Aduan dimulai ketika pada halaman daftar pengaduan, Petugas Administrasi memilih icon ubah status aduan pada boundary adm\_listaduan. Boundary adm\_listaduan akan menampilkan pop-up konfirmasiUbahStatusAduan kemudian Petugas Administrasi mengklik oke, selanjutnya akan dilanjutkan controller Controlpengaduan dengan memanggil fungsi getStatusAduan() untuk mengubah status laporan pengaduan. Controller Controlpengguna akan memanggil entity Data\_aduan dengan memanggil fungsi ubahstatus\_aduan() untuk mengubah status laporan pengaduan didalam sistem. Selanjutnya controller Controlpengaduan akan menampilkan notifikasi status telah berubah.

D. Sequence Diagram Cetak Aduan

Sequence Diagram Cetak Aduan merupakan subflows dari Activity Diagram Mengelola Data Pengguna pada Gambar 4.12. Sequence diagram dari fungsi Melihat Pesan Balasan Pengaduan dapat dilihat pada Gambar 5.16.



powered by Astah

Gambar 5.16 Sequence Diagram Cetak Aduan



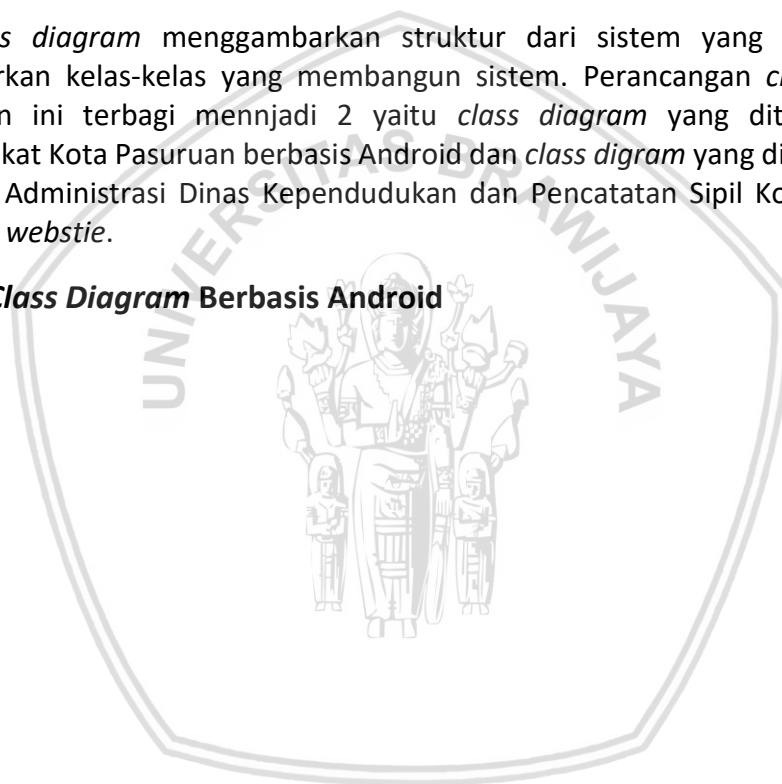


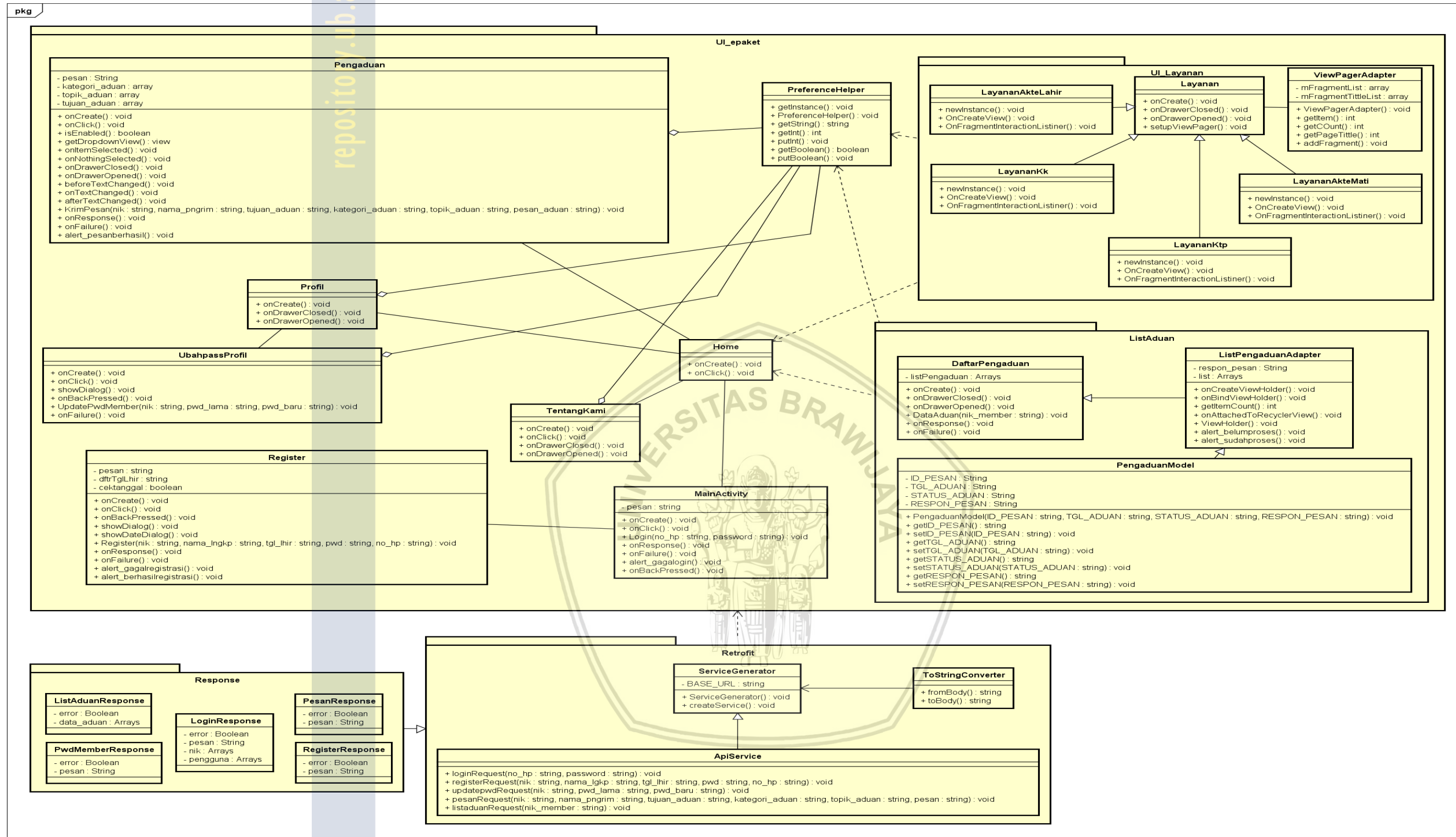
*Sequence diagram* Cetak Aduan dimulai ketika pada halaman daftar pengaduan, Petugas Administrasi memilih icon cetak pdf pada *boundary* adm\_listaduan. *Boundary* adm\_listaduan akan memanggil *controller* Controlpdf dengan memanggil fungsi pdf\_cetak() untuk menampilkan laporan pengaduan berformat PDF. *Controller* Controlpdf akan memanggil *entity* Data\_aduan dengan memanggil fungsi isipesan\_aduan() untuk mengirimkan data pengaduan yang diinginkan didalam sistem. Selanjutnya *controller* Controlpdf akan menampilkan informasi laporan pengaduan Masyarakat Kota Pasuruan berformat PDF.

## 5.2 Perancangan *Class Diagram*

*Class diagram* menggambarkan struktur dari sistem yang didefinisikan berdasarkan kelas-kelas yang membangun sistem. Perancangan *class diagram* penelitian ini terbagi menjadi 2 yaitu *class diagram* yang ditujukan oleh Masyarakat Kota Pasuruan berbasis Android dan *class diagram* yang ditujukan oleh Petugas Administrasi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan berbasis *webstie*.

### 5.2.1 *Class Diagram* Berbasis Android



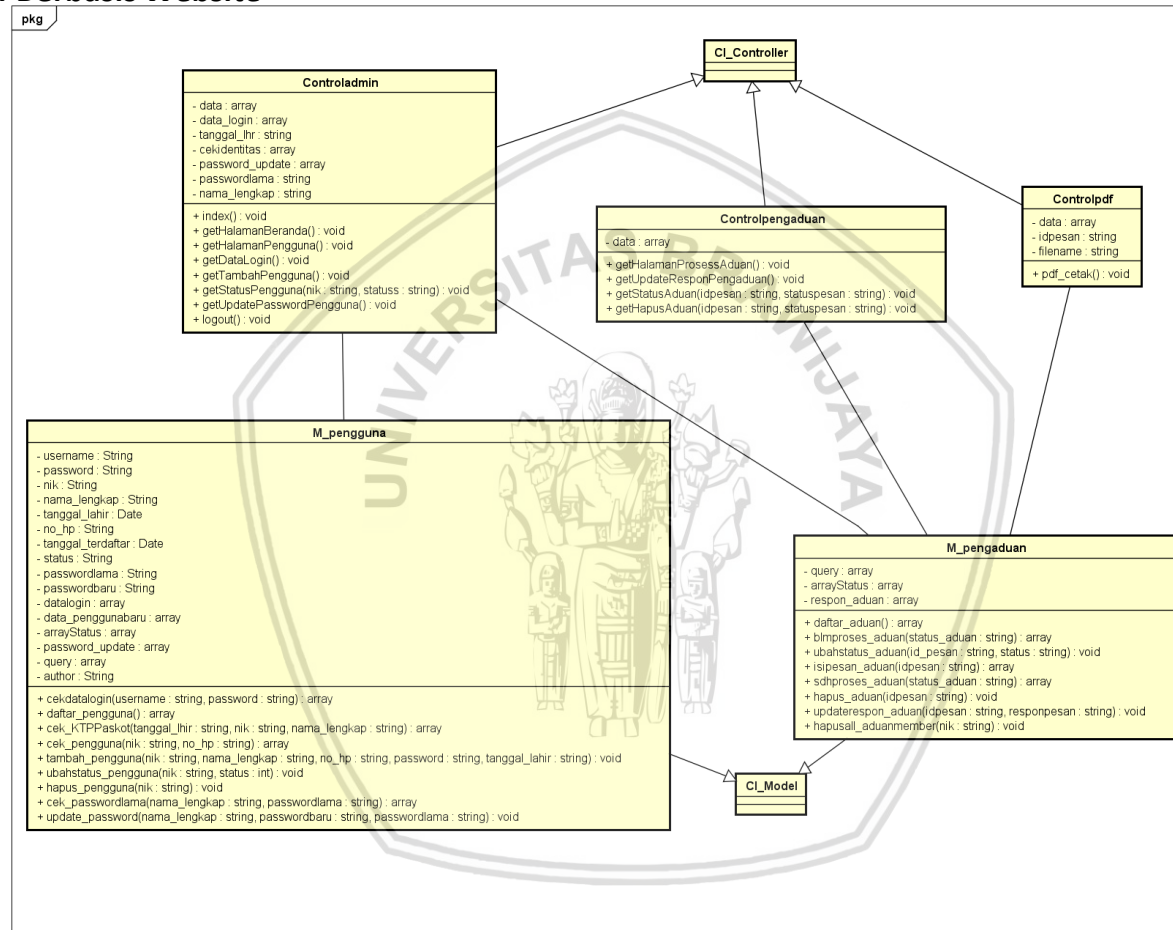


Gambar 5.17 Class Diagram Berbasis Android

Berdasarkan gambar 5.17, *Class Diagram* dari sistem layanan pengaduan masyarakat yang berbasis android yang ditujukan untuk Masyarakat Kota Pasuruan. Jumlah kelas yang dihasilkan sejumlah 25 kelas yang telah dikelompokkan menjadi 3 folder. Berikut ini daftar mengenai tiap folder beserta kelasnya dalam sistem layanan pengaduan :

1. UI\_epaket berisi kelas yang mengatur fungsi-fungsi untuk menampilkan informasi pada pengguna. Terdapat beberapa kelas, yaitu Pengaduan, Profil, UbahpassProfil, TentangKami, Register, MainActivity, Home dan PreferenceHelper. Tetapi dalam folder UI\_Paket terdapat beberapa folder didalamnya beserta kelasnya sebagai berikut:
  - a. UI\_layanan berisi kelas yang mengatur fungsi-fungsi dalam menampilkan informasi layanan yang tersedia. Beberapa kelas yang terdapat didalamnya, yaitu : Layanan, LayananAkteLahir, LayananAkteMati, LayananKK, LayananKtp dan ViewPagerAdapter.
  - b. ListAduan berisi kelas yang mengatur fungsi-fungsi tampilan mengenai informasi daftar pengaduan yang tersimpan dalam sistem. Beberapa kelas yang terdapat didalamnya, yaitu : DaftarPengaduan, ListPengaduanAdapter, PengaduanModel.
2. Retrofit berisi kelas yang mengatur fungsi-fungsi dalam menghubungkan terhadap *webservice* yang digunakan untuk pengambilan data sesuai permintaan yang diinginkan. Terdapat beberapa kelas yaitu : ServiceGenerator, ToStringConverter, ApiService.
3. Response berisi kelas yang mengatur ataupun menyimpan data sementara yang telah didapatkan dari *webservice* pada tiap kelas didalamnya sesuai dengan kegunaannya. Terdapat beberapa kelas yaitu : ListAduanResponse, PwdMemberResponse, LoginResponse, PesanResponse, RegisterResponse.

### 5.2.2 Class Diagram Berbasis Website



Gambar 5.18 Class Diagram Berbasis Website

id by Astah

Berdasarkan gambar 5.18, *Class Diagram* dari sistem layanan pengaduan masyarakat yang berbasis *website* yang ditujukan untuk Petugas Administrasi pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan. Sistem berbasis *website* ini dibangun menggunakan *framework Codeigniter* sehingga kelas yang akan diterapkan dibagi menjadi kelas *controller* dan kelas *model*. Kelas *controller* merupakan generalisasi atau Mewari kelas *CI\_Controller*. Kelas *controller* tersebut terdiri dari kelas *Controladmin*, *Controlpengaduan* dan *Controlpdf*. Kelas *Controladmin* secara umum berfungsi untuk mengelola data pengguna yang terdaftar pada sistem. Kelas *Controlpengaduan* berfungsi untuk mengelola pengaduan masyarakat kota pasuruan yang telah tersimpan. Kelas *Controlpdf* berfungsi untuk mencetak laporan pengaduan masyarakat yang diinginkan berformat *PDF* secara otomatis.

Kelas *model* merupakan kelas kelas generalisasi atau mewarisi fungsi dari kelas *CI\_Model*. Kelas *model* berfungsi untuk operasi data pada *database*, baik itu untuk operasi memasukkan data, *update* data, ataupun menghapus data. Kelas *model* terdiri dari kelas *M\_pengguna* dan kelas *M\_pengaduan*. Kelas *M\_pengguna* berfungsi untuk operasi data pengguna sistem. Kelas *M\_Pengaduan* berfungsi untuk operasi data pengaduan yang telah dikirimkan oleh Masyarakat Kota Pasuruan. Selanjutnya pada kelas *model* yang terdiri dari *M\_pengguna* dan *M\_pengaduan* digunakan sebagai acuan dalam perancangan basis data.

### 5.3 Perancangan Komunikasi Data

Bagian ini menjelaskan bagaimana proses pertukaran data antara aplikasi Layanan Pengaduan Masyarakat berbasis *Android* yang digunakan oleh Masyarakat Kota Pasuruan sebagai *front-end* dengan *web service* sebagai *back-end*. Aplikasi tidak dapat mengakses database di sever secara langsung. Tetapi membutuhkan *web service* sebagai perantara untuk melakukan komunikasi data. Data yang diberikan oleh *web service* dalam bentuk format *JSON*. Perancangan dilakukan dengan mendefinisikan input yang diperlukan dan output yang dihasilkan dari *web service*. *Variabel* output yang ditampilkan pada setiap fungsi di *webservice* digunakan sebagai salah satu acuan dalam perancangan basis data.

#### 5.3.1 Login

Tabel 5.1 *Web Service* untuk Login

Nama	loginpengguna
Tujuan	Meminta <i>web service</i> melakukan verifikasi dan pengambilan informasi pengguna sesuai inputan
Parameter <i>Input</i>	1. no_hp 2. password
<i>Method</i>	GET
<i>Output</i>	{

**Tabel 5.1 Web Service untuk Login (Lanjutan)**

	<pre> "pengguna": [   {     "NIK": " ",     "NAMA LENGKAP": "",     "NO_HP": " ",     "PASSWORD": " ",     "TANGGAL LAHIR": " ",     "TANGGAL TERDAFTAR": " ",     "STATUS": " "   } ], "error": false }         </pre>
--	---

**5.3.2 Registrasi**

**Tabel 5.2 Web Service untuk Registrasi**

<b>Nama</b>	registrasipengguna
<b>Tujuan</b>	Meminta <i>web service</i> melakukan penyimpanan data pengguna yang mendaftar ke dalam sistem
<b>Parameter Input</b>	<ol style="list-style-type: none"> <li>1. nik</li> <li>2. nama lengkap</li> <li>3. tanggal lahir</li> <li>4. password</li> <li>5. no. hp</li> </ol>
<b>Method</b>	POST
<b>Output</b>	<pre> {   "error": ,   "pesan": " " }         </pre>

**5.3.3 Ubah Password**

**Tabel 5.3 Web Service untuk Ubah Password**

<b>Nama</b>	updatepasspengguna
<b>Tujuan</b>	Meminta <i>web service</i> melakukan <i>update password</i> pengguna yang tersimpan di sistem

Tabel 5.3 *Web Service* untuk Ubah Password (Lanjutan)

Parameter <i>Input</i>	<ol style="list-style-type: none"> <li>1. nik</li> <li>2. password baru</li> <li>3. password lama</li> </ol>
<i>Method</i>	PUT
<i>Output</i>	<pre>{   "error": ,   "pesan": " " }</pre>

### 5.3.4 Melaporkan Pengaduan

Tabel 5.4 *Web Service* untuk Melaporkan Pengaduan

Nama	simpanpesan
Tujuan	Meminta <i>web service</i> melakukan penyimpanan data laporan pengaduan ke dalam sistem
Parameter <i>Input</i>	<ol style="list-style-type: none"> <li>1. nik</li> <li>2. nama pengirim</li> <li>3. tujuan aduan</li> <li>4. kategori aduan</li> <li>5. topik aduan</li> <li>6. pesan</li> </ol>
<i>Method</i>	POST
<i>Output</i>	<pre>{   "data_pesanan": {     "ID_PESAN": " ",     "NIK": " ",     "NAMA_PENGINRIM": " ",     "ID_TUJUAN": " ",     "ID_KATEGORI": " ",     "ID_TOPIK": " ",     "PESAN": " ",     "TGL_ADUAN": " ",     "STATUS_ADUAN": " ",     "RESPON_PESAN": " "   },   "error": ,   "pesan": " " }</pre>

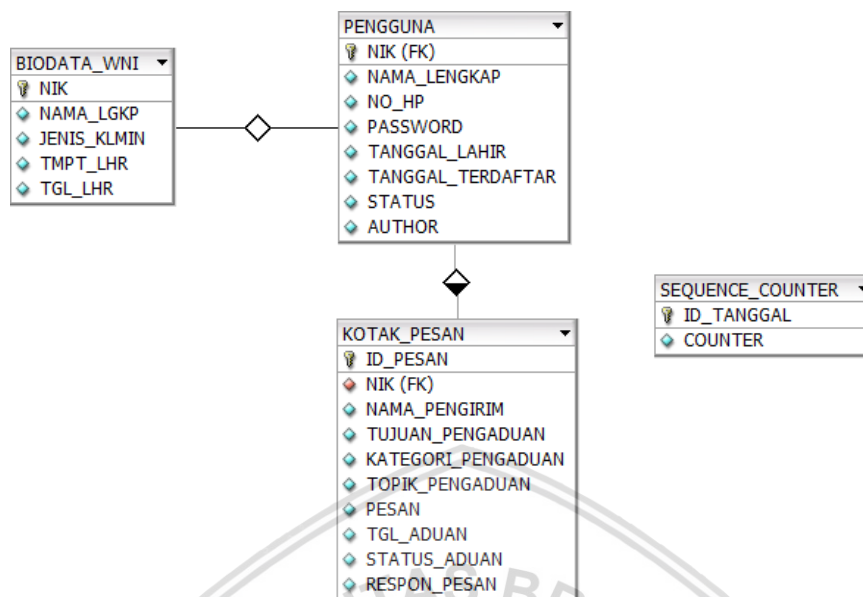
### 5.3.5 Melihat Daftar Pengaduan

Tabel 5.5 *Web Service* untuk Melihat Daftar Pengaduan

Nama	listaduanpengguna
Tujuan	Meminta <i>web service</i> melakukan pengambilan informasi laporan pengaduan sesuai data nik pengguna di dalam sistem
Parameter <i>Input</i>	1. Nik
<i>Method</i>	GET
<i>Output</i>	<pre>{   "data_aduan": [     {       "ID_PESAN": " ",       "NIK": " ",       "NAMA_PENKIRIM": " ",       "ID_TUJUAN": " ",       "ID_KATEGORI": " ",       "ID_TOPIK": " ",       "PESAN": " ",       "TGL_ADUAN": " ",       "STATUS_ADUAN": " ",       "RESPON_PESAN": " "     }   ],   "error": " " }</pre>



## 5.4 Pemetaan *Class Diagram* ke *Relational Model*



**Gambar 5.19 Relational Model**

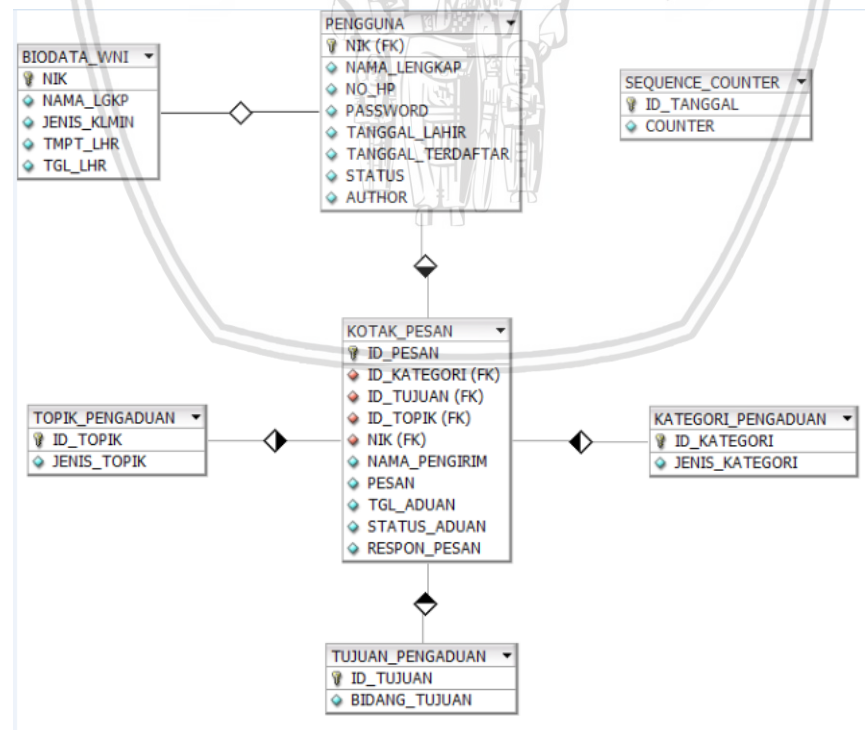
Gambar 5.19 merupakan rancangan pemetaan *class diagram* berbasis *website* yang terdiri dari M\_pengguna dan M\_pengaduan maupun dari hasil variabel output pada perancangan komunikasi data yang telah didefinisikan pada sub bab sebelumnya ke *relational model*. Perancangan *relational model* digambarkan dalam bentuk *physical data model* meliputi pendefinisian antar tabel-tabel dan hubungan antar tabel yang akan digunakan sebagai acuan untuk membangun basis data sistem dalam penelitian ini. Hal ini dilakukan agar basis data dapat berjalan sesuai dengan kebutuhan sistem. Jumlah tabel utama pada Sistem Layanan Pengaduan terdiri dari :

1. BIODATA\_WNI merupakan tabel yang berasal dari acuan M\_pengguna dimana tabel ini digunakan untuk menyimpan seluruh data identitas Masyarakat Kota Pasuruan yang memiliki KTP Kota Pasuruan.
2. PENGGUNA merupakan tabel yang berasal dari acuan M\_pengguna dimana tabel ini digunakan untuk melakukan penyimpanan identitas pengguna dari Masyarakat Kota Pasuruan maupun petugas administrasi dari Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan yang telah terdaftar pada Sistem Layanan Pengaduan. Tabel PENGGUNA memiliki relasi dengan tabel BIODATA\_WNI dimana Identitas pengguna sebelum terdaftar dalam Sistem Layanan Pengaduan akan dilakukan pencocokan identitas terlebih dahulu kepada tabel BIODATA\_WNI untuk mengetahui apakah pengguna beridentitas KTP Kota Pasuruan apabila cocok amaka akan tersimpan pada tabel PENGGUNA dan telah terdaftar kedalam Sistem Layanan Pengaduan.

3. KOTAK\_PESAN merupakan tabel yang berasal dari acuan output pada sub bab perancangan komunikasi data dimana tabel ini digunakan sebagai penyimpanan informasi laporan pengaduan Masyarakat Kota Pasuruan yang telah dikirimkan melalui Sistem Layanan Pengaduan. Tabel KOTAK\_PESAN memiliki relasi dengan tabel PENGGUNA dimana akan melakukan pencocokan identitas Masyarakat Kota Pasuruan yang melaporkan pengaduan Masyarakat Kota Pasuruan yang dapat melaporkan pengaduan harus atau telah terdaftar terlebih dahulu kedalam sistem dan identitas Masyarakat Kota Pasuruan yang telah tersimpan pada tabel PENGGUNA.
4. SEQUENCE\_COUNTER merupakan tabel yang dibutuhkan oleh *Staff Ahli IT* Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan sesuai pada Lampiran F Hasil Wawancara dimana tabel ini digunakan untuk penyimpanan data berupa berapa banyak laporan pengaduan masyarakat kota pasuruan yang masuk kedalam sistem setiap bulan.

### 5.5 Normalisasi *Relational Model*

Setelah memodelkan pemetaan ke *relational model*, langkah selanjutnya yaitu melakukan normalisasi *relational model*. Tujuan dari normalisasi adalah untuk membuat agar data yang ada tidak mengalami redundan dan memiliki integritas data yang kuat sehingga ketika kita melakukan relasi antar tabel akan dengan mudah kita menjaga integritas data.



Gambar 5.20 Normalisasi Relational Model

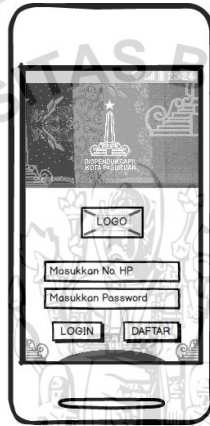
## 5.6 Perancangan Antarmuka Sistem

Bagian ini menjelaskan bagaimana antarmuka sistem dirancang untuk membantu dalam proses implementasi sistem. Perancangan antarmuka akan terbagi menjadi dua tampilan yaitu berbasis Android untuk Masyarakat Kota Pasuruan dan berbasis *Website* untuk Petugas Administrasi Dinas Kependudukan dan Pencatan Sipil Kota Pasuruan pada sistem layanan pengaduan masyarakat.

### 5.6.1 Perancangan Antarmuka Sistem berbasis Android

#### 5.6.1.1 Halaman *Login*

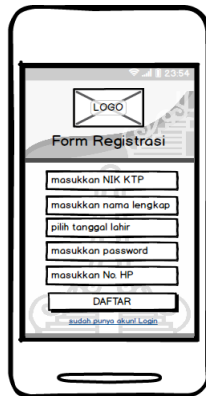
Halaman *login* berbasis Android merupakan halaman untuk masuk ke dalam sistem yang ditujukan kepada Masyarakat Kota Pasuruan, terlebih dahulu Masyarakat Kota Pasuruan harus memasukkan no. hp dan *password*. Berikut ini merupakan rancangan antarmuka *login* pada gambar 5.21.



Gambar 5.21 Perancangan Antarmuka *Login*

#### 5.6.1.2 Halaman Registrasi

Halaman Registrasi merupakan halaman untuk melakukan pendaftaran ke dalam sistem, terlebih dahulu Masyarakat Kota Pasuruan harus mengisi data identitas sesuai permintaan data yang dibutuhkan. Berikut ini merupakan rancangan antarmuka registrasi pada gambar 5.22.

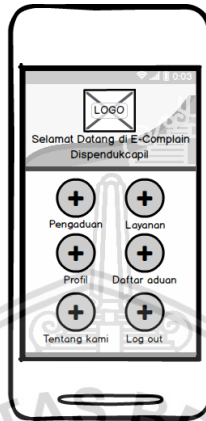


Gambar 5.22 Perancangan Antarmuka Registrasi



### 5.6.1.3 Halaman Beranda

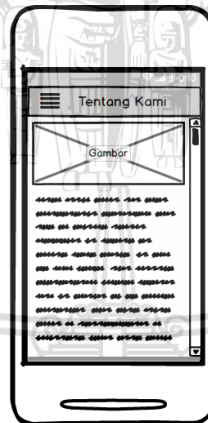
Halaman Beranda berbasis Android merupakan merupakan halaman utama dari sistem yang ditujukan kepada Masyarakat Kota Pasuruan, yang menampilkan semua menu atau fitur yang tersedia pada sistem. Berikut ini merupakan rancangan antarmuka beranda pada gambar 5.23.



Gambar 5.23 Perancangan Antarmuka Beranda

### 5.6.1.4 Halaman Tentang Kami

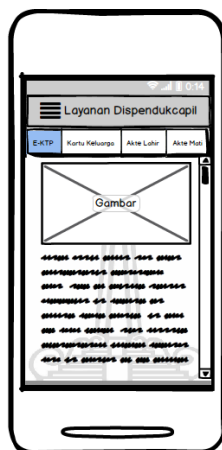
Halaman Tentang Kami merupakan halaman informasi mengenai profil singkat Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan yang terdapat pada sistem. Berikut ini merupakan rancangan antarmuka tentang kami pada gambar 5.24.



Gambar 5.24 Perancangan Antarmuka Tentang Kami

### 5.6.1.5 Halaman Layanan

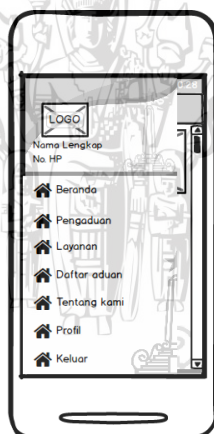
Halaman Layanan merupakan halaman informasi mengenai pelayanan yang tersedia pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan, misalnya E-KTP, Akte Lahir, Akte Mati dan Kartu Keluarga yang terdapat pada sistem. Berikut ini merupakan rancangan antarmuka layanan pada gambar 5.25.



Gambar 5.25 Perancangan Antarmuka Layanan

#### 5.6.1.6 Halaman *NavigationView*

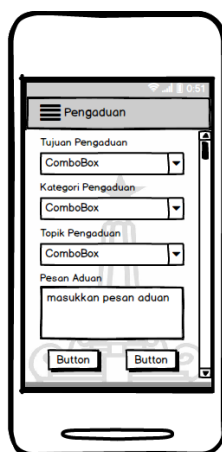
Halaman *NavigationView* merupakan jalan pintas yang menampilkan daftar menu yang tersedia pada sistem hanya dengan menggeser tampilan ke arah kanan atau menekan symbol *justify* tanpa perlu kembali ke halaman beranda tetapi halaman *NavigationView* tidak tersedia pada halaman beranda. Berikut ini merupakan rancangan antarmuka *NavigationView* pada gambar 5.26.



Gambar 5.26 Perancangan Antarmuka *NavigationView*

#### 5.6.1.7 Halaman Pengaduan

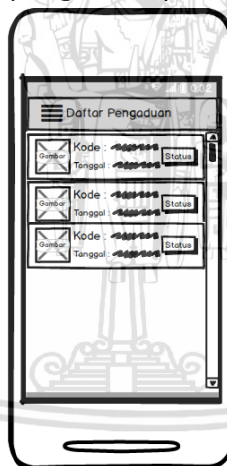
Halaman Pengaduan merupakan halaman untuk melakukan pelaporan pengaduan kepada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan melalui Sistem Layanan Pengaduan Masyarakat. Masyarakat Kota Pasuruan terlebih dahulu mengisi data laporan pengaduan berupa tujuan pengaduan, kategori pengaduan, topik pengaduan dan pesan aduan. Berikut ini merupakan rancangan antarmuka pengaduan pada gambar 5.27.



Gambar 5.27 Perancangan Antarmuka Pengaduan Berbasis Android

#### 5.6.1.8 Halaman Daftar Pengaduan

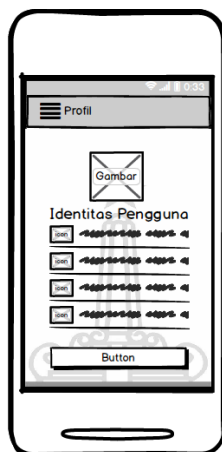
Halaman Daftar Pengaduan merupakan halaman informasi mengenai laporan pengaduan yang pernah dilakukan sesuai dengan identitas Masyarakat Kota Pasuruan yang telah *login* di Sistem Layanan Pengaduan Masyarakat. Masyarakat Kota Pasuruan juga disediakan tombol untuk melihat respon dari pengaduan yang dikirimkan dengan menekan tombol status pengaduan. Berikut ini merupakan rancangan antarmuka daftar pengaduan pada gambar 5.28.



Gambar 5.28 Perancangan Antarmuka Daftar Pengaduan Berbasis Android

#### 5.6.1.9 Halaman Profil

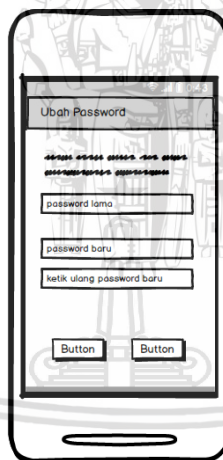
Halaman Profil merupakan halaman informasi mengenai identitas biodata diri pengguna Masyarakat Kota Pasuruan yang telah tersimpan di dalam sistem. Berikut ini merupakan rancangan antarmuka profil pada gambar 5.29.



Gambar 5.29 Perancangan Antarmuka Profil Berbasis Android

#### 5.6.1.10 Halaman Ubah *Password*

Halaman Ubah *Password* merupakan halaman untuk melakukan penggantian *password* lama menjadi *password* baru pada akun pengguna Masyarakat Kota Pasuruan apabila ingin melakukan perubahan *password* yang dimiliki. Masyarakat Kota Pasuruan diharuskan mengisi data berupa memasukkan *password* lama, *password* baru dan memasukkan ulang *password* baru. Berikut ini merupakan rancangan antarmuka ubah *password* pada gambar 5.30.

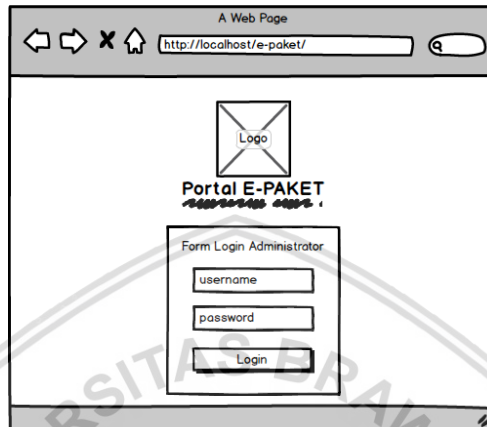


Gambar 5.30 Perancangan Antarmuka Ubah *Password* Berbasis Android

## 5.6.2 Perancangan Antarmuka Sistem berbasis Website

### 5.6.2.1 Halaman Login

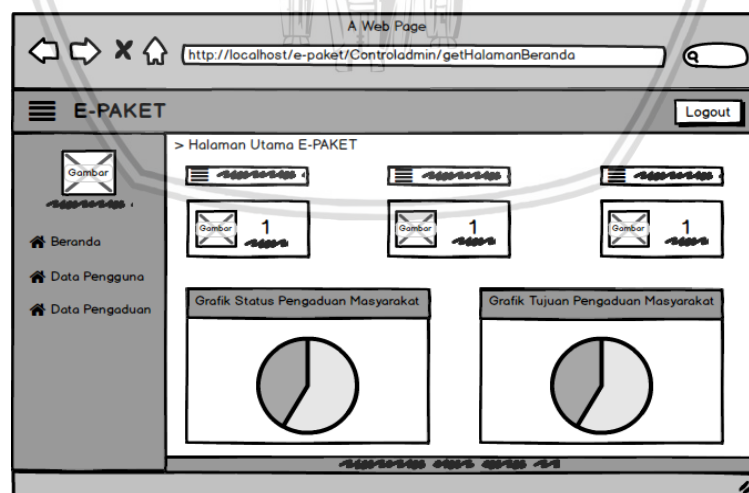
Halaman *login* berbasis *Website* merupakan halaman untuk masuk ke dalam sistem yang ditujukan kepada Petugas Administrasi, terlebih dahulu Petugas Administrasi harus memasukkan *username* dan *password*. Berikut ini merupakan rancangan antarmuka *login* pada gambar 5.31.



Gambar 5.31 Perancangan Antarmuka *Login* Berbasis *Website*

### 5.6.2.2 Halaman Beranda

Halaman Beranda berbasis *Android* merupakan merupakan halaman utama dari sistem yang ditujukan kepada Petugas Administrasi, yang menampilkan grafik pengaduan dan informasi mengenai staus pengaduan yang sudah atau belum diproses pada sistem. Berikut ini merupakan rancangan antarmuka Beranda pada gambar 5.32.

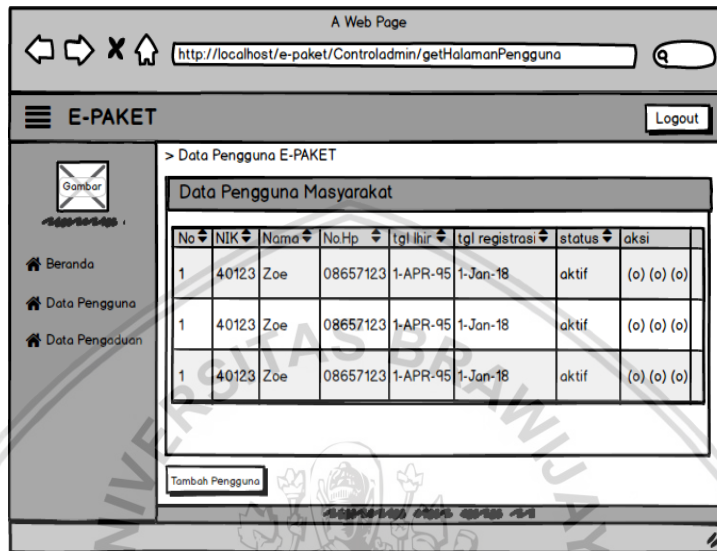


Gambar 5.32 Perancangan Antarmuka Beranda Berbasis *Website*



### 5.6.2.3 Halaman Data Pengguna

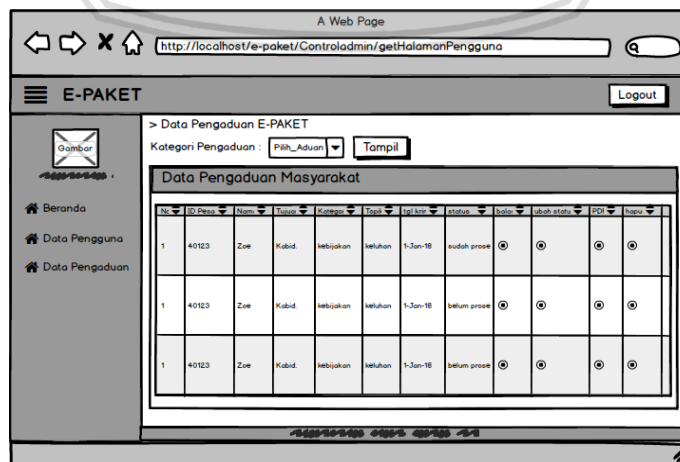
Halaman Data Pengguna merupakan halaman yang menampilkan informasi seluruh data pengguna yang terdaftar dan digunakan untuk mengelola pengguna dengan tersedianya fungsi yang terdapat pada kolom aksi di tabel data pengguna ataupun tombol tambah pengguna yang digunakan untuk mendaftar pengguna melalui Petugas Administrasi yang tersedia di dalam sistem. Berikut ini merupakan rancangan antarmuka Data Pengguna pada gambar 5.33.



Gambar 5.33 Perancangan Antarmuka Data Pengguna Berbasis Website

### 5.6.2.4 Halaman Data Pengaduan

Halaman Data Pengaduan merupakan halaman yang menampilkan informasi seluruh data pengaduan yang tersimpan maupun menampilkan status pengaduannya. Halaman Data Pengaduan juga digunakan untuk mengelola pengaduan Masyarakat Kota Pasuruan salah satunya mencetak laporan pengaduan secara otomatis melalui sistem. Berikut ini merupakan rancangan antarmuka Data Pengaduan pada gambar 5.34.

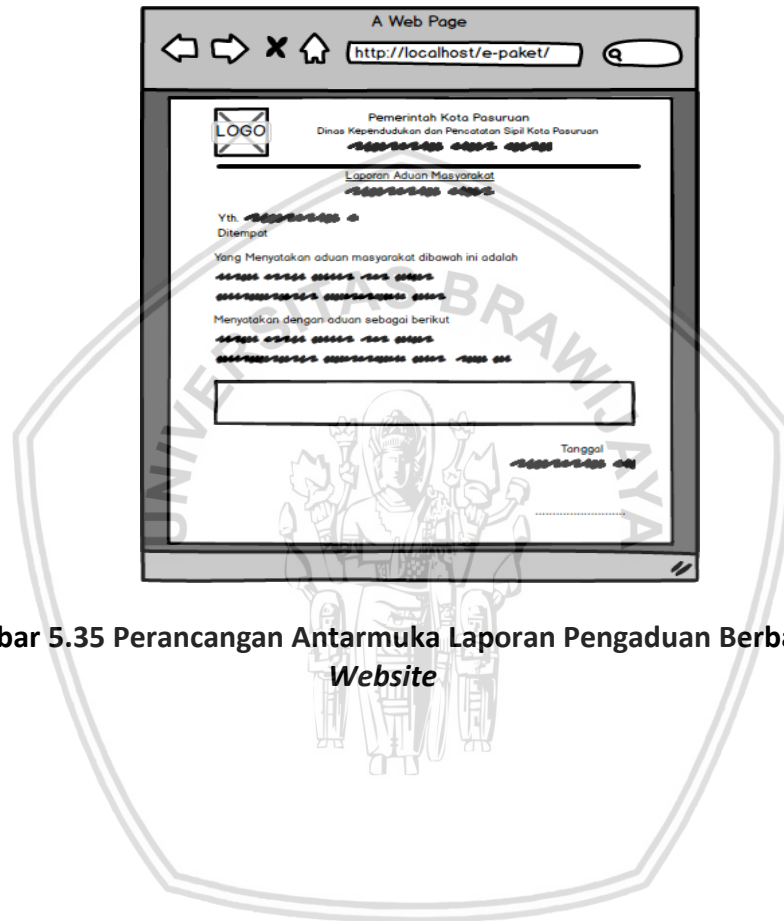


Gambar 5.34 Perancangan Antarmuka Data Pengaduan Berbasis Website



### 5.6.2.5 Halaman Laporan Pengaduan

Halaman Laporan Pengaduan merupakan halaman yang menampilkan laporan pengaduan masyarakat berformat *PDF* yang akan dicetak oleh petugas administrasi untuk diberikan kepada Bidang yang dituju sesuai pengaduan tersebut berisi identitas pengirim dan isi pengaduan kemudian disediakan kotak respon pengaduan untuk memberikan pernyataan atau balasan mengenai pengaduan tersebut. Berikut ini merupakan rancangan antarmuka Laporan Pengaduan pada gambar 5.35.



Gambar 5.35 Perancangan Antarmuka Laporan Pengaduan Berbasis Website



## BAB 6 IMPLEMENTASI

Bab ini membahas mengenai hasil implementasi Sistem Layanan Pengaduan Masyarakat di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan. Hasil dari implementasi berdasarkan analisis kebutuhan dan perancangan yang telah dibahas pada bab sebelumnya. Bab ini dibagi menjadi beberapa pembahasan, antara lain batasan implementasi, implementasi program dan implementasi antarmuka.

### 6.1 Batasan Implementasi

Berikut ini beberapa batasan dalam implementasi Sistem Layanan Pengaduan Masyarakat Kota Pasuruan di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan :

1. Pengembangan *webservice* menggunakan *RESTful Web Service*
2. Masyarakat Kota Pasuruan harus memiliki akun untuk mengakses Sistem Layanan Pengaduan Masyarakat
3. Oracle sebagai Manajemen Basis Data Sistem
4. Sistem memiliki fitur utama berupa mengirim pengaduan dan melihat balasan pesan pengaduan bagi Masyarakat Kota Pasuruan, serta kelola data pengaduan termasuk fitur cetak laporan pengaduan bagi Petugas Administrasi
5. Untuk mengakses sistem diperlukan koneksi internet, sehingga kecepatan akses untuk menampilkan data atau mengirim pengaduan bergantung pada kecepatan koneksi internet masing-masing perangkat yang digunakan

### 6.2 Implementasi Basis Data

Implementasi Basis Data pada penelitian ini mengacu pada gambar 5.20 sebagai struktur tabel yang akan dibangun. *Query SQL* yang digunakan dalam implementasi Sistem Layanan Pengaduan Masyarakat sebagai berikut :

#### 6.2.1 Tabel BIODATA\_WNI

**Tabel 6.1 Implementasi Basis Data Tabel BIODATA\_WNI**

1	CREATE TABLE	PENGADUAN.BIODATA_WNI
2	(	
3	NIK	NUMBER(16) NOT NULL,
4	NAMA_LGKP	VARCHAR2(60 BYTE) NOT NULL,
5	JENIS_KLMIN	NUMBER(1) NOT NULL,
6	TMPT_LHR	VARCHAR2(60 BYTE) NOT NULL,
7	TGL_LHR	DATE NOT NULL
8	)	
9		
10		

Tabel 6.1 Implementasi Basis Data Tabel BIODATA\_WNI (Lanjutan)

11	CREATE UNIQUE INDEX PENGADUAN.BIODATA_WNI_PK ON
12	PENGADUAN.BIODATA_WNI
13	(NIK);
14	
15	ALTER TABLE PENGADUAN.BIODATA_WNI ADD (
16	CONSTRAINT BIODATA_WNI_PK
17	PRIMARY KEY
18	(NIK));

### 6.2.2 Tabel PENGGUNA

Tabel 6.2 Implementasi Basis Data Tabel PENGGUNA

1	CREATE TABLE PENGADUAN.PENGGUNA
2	(
3	NIK                  NUMBER(16)                  NOT NULL,
4	NAMA LENGKAP      VARCHAR2(32 BYTE)          NOT NULL,
5	NO_HP              VARCHAR2(32 BYTE)          NOT NULL,
6	PASSWORD          VARCHAR2(32 BYTE)          NOT NULL,
7	TANGGAL LAHIR      DATE                      NOT NULL,
8	TANGGAL TERDAFTAR DATE                      NOT NULL,
9	STATUS            VARCHAR2(32 BYTE)          NOT NULL,
10	AUTHOR            VARCHAR2(32 BYTE)          NOT NULL
11	);
12	
13	
14	CREATE UNIQUE INDEX PENGADUAN.MEMBER_PK ON PENGADUAN.PENGGUNA
15	(NIK);
16	
17	CREATE INDEX PENGADUAN.PENGGUNA_IDX ON PENGADUAN.PENGGUNA
18	(NO_HP);
19	
20	ALTER TABLE PENGADUAN.PENGGUNA ADD (
21	CONSTRAINT MEMBER_PK
22	PRIMARY KEY
23	(NIK)
24	CONSTRAINT UNIQ_NO_HP
25	UNIQUE (NO_HP));
26	
27	
28	ALTER TABLE PENGADUAN.PENGGUNA ADD (
29	CONSTRAINT NIK_FK
30	FOREIGN KEY (NIK)
31	REFERENCES PENGADUAN.BIODATA_WNI (NIK));
32	

### 6.2.3 Tabel TOPIK\_PENGADUAN

Tabel 6.3 Implementasi Basis Data Tabel TOPIK\_PENGADUAN

1	CREATE TABLE PENGADUAN.TOPIK_PENGADUAN
2	(
3	ID_TOPIK          CHAR(2 BYTE)              NOT NULL,
4	JENIS_TOPIK      VARCHAR2(128 BYTE)          NOT NULL

**Tabel 6.3 Implementasi Basis Data Tabel TOPIK\_PENGADUAN (Lanjutan)**

```

5 )
6 CREATE UNIQUE INDEX PENGADUAN.TOPIK_PENGADUAN_PK ON
7 PENGADUAN.TOPIK_PENGADUAN
8 (ID_TOPIK);
9
10 ALTER TABLE PENGADUAN.TOPIK_PENGADUAN ADD (
11 CONSTRAINT TOPIK_PENGADUAN_PK
12 PRIMARY KEY
13 (ID_TOPIK));
14

```

**6.2.4 Tabel TUJUAN\_PENGADUAN****Tabel 6.4 Implementasi Basis Data Tabel TUJUAN\_PENGADUAN**

```

1 CREATE TABLE PENGADUAN.TUJUAN_PENGADUAN
2 (
3 ID_TUJUAN CHAR(2 BYTE) NOT NULL,
4 BIDANG_TUJUAN VARCHAR2(128 BYTE) NOT NULL
5 )
6 CREATE UNIQUE INDEX PENGADUAN.KATEGORI_PENGADUAN_PK ON
7 PENGADUAN.TUJUAN_PENGADUAN
8 (ID_TUJUAN);
9
10 ALTER TABLE PENGADUAN.TUJUAN_PENGADUAN ADD (
11 CONSTRAINT TUJUAN_PENGADUAN_PK
12 PRIMARY KEY
13 (ID_TUJUAN));
14
15

```

**6.2.5 Tabel KATEGORI\_PENGADUAN****Tabel 6.5 Implementasi Basis Data Tabel KATEGORI\_PENGADUAN**

```

1 CREATE TABLE PENGADUAN.KATEGORI_PENGADUAN
2 (
3 ID_KATEGORI CHAR(2 BYTE) NOT NULL,
4 JENIS_KATEGORI VARCHAR2(128 BYTE) NOT NULL
5 )
6
7 CREATE UNIQUE INDEX PENGADUAN.KATEGORI_PENGADUAN_PK ON
8 PENGADUAN.KATEGORI_PENGADUAN
9 (ID_KATEGORI);
10
11 ALTER TABLE PENGADUAN.KATEGORI_PENGADUAN ADD (
12 CONSTRAINT KATEGORI_PENGADUAN_PK
13 PRIMARY KEY
14 (ID_KATEGORI));
15

```

## 6.2.6 Tabel KOTAK\_PESAN

Tabel 6.6 Implementasi Basis Data Tabel KOTAK\_PESAN

```
1 CREATE TABLE PENGADUAN.KOTAK_PESAN
2 (
3   ID_PESAN    VARCHAR2(35 CHAR)    NOT NULL,
4   NIK         NUMBER(16)           NOT NULL,
5   NAMA_PENIRIM VARCHAR2(35 CHAR)    NOT NULL,
6   ID_TUJUAN   CHAR(2 CHAR)         NOT NULL,
7   ID_KATEGORI CHAR(2 CHAR)         NOT NULL,
8   ID_TOPIK    CHAR(2 CHAR)         NOT NULL,
9   PESAN       VARCHAR2(450 CHAR)    NOT NULL,
10  TGL_ADUAN   DATE                 NOT NULL,
11  STATUS_ADUAN VARCHAR2(35 CHAR)    NOT NULL,
12  RESPON_PESAN VARCHAR2(350 CHAR)    NOT NULL
13 );
14 CREATE UNIQUE INDEX PENGADUAN.KOTAK_PESAN_PK ON
15 PENGADUAN.KOTAK_PESAN
16 (ID_PESAN);
17
18 ALTER TABLE PENGADUAN.KOTAK_PESAN ADD (
19   CONSTRAINT KOTAK_PESAN_PK
20   PRIMARY KEY
21   (ID_PESAN);
22
23 ALTER TABLE PENGADUAN.KOTAK_PESAN ADD (
24   CONSTRAINT FK_PESAN_NIK
25   FOREIGN KEY (NIK)
26   REFERENCES PENGADUAN.PENGGUNA (NIK));
27
28 ALTER TABLE PENGADUAN.KOTAK_PESAN ADD (
29   CONSTRAINT FK_KATEGORI
30   FOREIGN KEY (ID_KATEGORI)
31   REFERENCES PENGADUAN.KATEGORI_PENGADUAN (ID_KATEGORI),
32   CONSTRAINT FK_PESAN_NIK
33   FOREIGN KEY (NIK)
34   REFERENCES PENGADUAN.PENGGUNA (NIK),
35   CONSTRAINT FK_TOPIK
36   FOREIGN KEY (ID_TOPIK)
37   REFERENCES PENGADUAN.TOPIK_PENGADUAN (ID_TOPIK),
38   CONSTRAINT FK_TUJUAN
39   FOREIGN KEY (ID_TUJUAN)
40   REFERENCES PENGADUAN.TUJUAN_PENGADUAN (ID_TUJUAN));
41
42
43
```

## 6.2.7 Tabel SEQUENCE\_COUNTER

Tabel 6.7 Implementasi Basis Data Tabel SEQUENCE\_COUNTER

```
1 CREATE TABLE PENGADUAN.SEQUENCE_COUNTER
2 (
3   ID_TANGGAL VARCHAR2(60 BYTE)    NOT NULL,
4   COUNTER    INTEGER              NOT NULL
5 )
```

**Tabel 6.7 Implementasi Basis Data Tabel SEQUENCE\_COUNTER (Lanjutan)**

6	)
7	
8	ALTER TABLE PENGADUAN.SEQUENCE_COUNTER ADD (
9	PRIMARY KEY
10	(ID_TANGGAL));

### 6.3 Implementasi Program

Dengan menerapkan pola perancangan *model-view-controller*, maka kelas – kelas yang diterapkan dibagi menjadi kelas *controller* dan kelas model. Pada implementasi program ini akan diuraikan beberapa program dari kelas *controller* pada sistem berbasis *Website*, sedangkan pada sistem berbasis Android akan menampilkan beberapa proses implementasi kode program yang ditampilkan adalah proses mengirim pengaduan dan menampilkan daftar pengaduan yang terdapat pada Sistem Layanan Pengaduan Masyarakat.

#### 6.3.1 Program Berbasis Android

##### 6.3.1.1 Implementasi Prosedur Mengirim Pengaduan

Implementasi yang digunakan pada Mengirim Pengaduan dalam Sistem Layanan Pengaduan Masyarakat dengan memiliki beberapa fungsi sebagai berikut.

##### 1. Fungsi KirimPesan

Fungsi KirimPesan berfungsi mengirimkan data laporan pengaduan dimana memiliki parameter berupa `nik`, `nama_pngrim`, `tujuan_aduan`, `kategori_aduan`, `topik_aduan` dan `pesan_aduan`. Selanjutnya akan dilakukan inialisasi sebagai penghubung kepada *webservice* dan memanggil *service* yang telah disediakan untuk mengirimkan data laporan pengaduan kedalam sistem melalui *webservice* dengan nama *service* `pesanRequest`. Didalamnya terdapat fungsi `onResponse` digunakan apabila berhasil terhubung kedalam *webservice* dan menampilkan pesan apakah data laporan pengaduan yang dikirimkan berhasil tersimpan atau gagal tersimpan, sedangkan `onFailure` digunakan untuk menampilkan pesan error apabila tidak terhubung dengan *webservice*. Kode Program Fungsi KirimPesan pada Proses Mengirim Pengaduan dapat dilihat pada Tabel 6.8 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.9 berikut ini.

**Tabel 6.8 Kode Program Fungsi KirimPesan**

1	<code>public void KirimPesan(String nik, String nama_pngrim,</code>
2	<code>int tujuan_aduan, int kategori_aduan, int topik_aduan,</code>
3	<code>String pesan_aduan) {</code>
4	<code>    Log.d("isi inputan", "nik: " + nik + ", nama : " +</code>
5	<code>nama_pngrim + ", tujuan : " + tujuan_aduan + ", kategori : "</code>
6	<code>+ kategori_aduan + ", topik : " + topik_aduan</code>



Tabel 6.8 Kode Program Fungsi krimPesan (Lanjutan)

7	+ " pesan : " + pesan_aduan);
8	<b>service</b> =
9	ServiceGenerator.createService(ApiService.class);
10	<b>CallBody</b> = <b>service</b> .pesanRequest(nik, nama_pngrim,
11	tujuan_aduan, kategori_aduan, topik_aduan, pesan_aduan);
12	<b>CallBody</b> .enqueue(new Callback<PesanResponse>() {
13	<b>public void</b> onResponse(Response<PesanResponse>
14	response) {
15	PesanResponse model = response.body();;
16	<b>if</b> (model.error) {
17	<b>pesan</b> = model.pesan;
18	<b>loading</b> .dismiss();
19	Toast.makeText(Pengaduan.this, <b>pesan</b> ,
20	Toast.LENGTH_SHORT).show();
21	Log.d("coba2 semoaga keluar", new
22	Gson().toJson(response.body()));
23	}
24	<b>if</b> (!model.error) {
25	<b>pesan</b> = model.pesan;
26	<b>loading</b> .dismiss();
27	Toast.makeText(Pengaduan.this, <b>pesan</b> ,
28	Toast.LENGTH_SHORT).show();
29	alert_pesanberhasil();
30	Log.d("coba2 semoaga keluar", new
31	Gson().toJson(response.body()));
32	}
33	}
34	
35	@Override
36	<b>public void</b> onFailure(Throwable t) {
37	<b>loading</b> .dismiss();
38	Toast.makeText(Pengaduan.this,
39	t.getMessage(), Toast.LENGTH_SHORT).show();
40	Log.d("coba2 semoaga keluar",
41	t.getMessage());
42	}
43	});
44	}

Tabel 6.9 Penjelasan Kode Program Fungsi krimPesan

Baris Kode	Penjelasan
1-3	Deklarasi nama fungsi bernama KrimPesan dengan memiliki parameter nik,nama_pngrim,tujuan_aduan,kategori_aduan,topik_aduan,pesan_aduan
4-7	Menampilkan kedalam konsola agar mengetahui masukkan parameter di fungsi KrimPesan
8-9	Inisialisasi objek service sebagai penghubung dengan <i>webservice</i> sekaligus memanggil <i>class</i> ApiService sebagai penyedia fungsi dari <i>webservice</i>



**Tabel 6.9 Penjelasan Kode Program Fungsi kirimPesan (Lanjutan)**

10-11	Inisialisasi objek Callbody untuk memanggil service sekaligus fungsi pesanRequest yang memiliki parameter
12	Callbody melakukan pemanggilan selanjutnya data yang didapat ditampung pada class PesanResponse
13	Deklarasi fungsi onResponse digunakan apabila pemanggilan berhasil dengan memanggil PesanResponse
15	Inisialisasi objek model dari class PesanResponse untuk menyimpan data JSON dari PesanResponse
16-23	Jika model index error <i>true</i> maka model akan mengambil data yang tersimpan di index pesan, selanjutnya akan ditampilkan berupa toast
24-31	Jika model index error <i>false</i> maka model akan mengambil data yang tersimpan di index pesan, selanjutnya memanggil fungsi alert_pesanberhasil
35-41	Deklarasi fungsi onFailure yang memiliki parameter digunakan apabila pemanggilan gagal dan menampilkan data parameter berupa toast

2. Fungsi alert\_pesanberhasil

Fungsi alert\_pesanberhasil berfungsi untuk menampilkan pop-up informasi apabila pengiriman laporan pengaduan yang dilakukan oleh Masyarakat Kota Pasuruan telah tersimpan dan halaman akan kembali pada halaman beranda. Kode Program Fungsi alert\_pesanberhasil pada Proses Mengirim Pengaduan dapat dilihat pada Tabel 6.10 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.11 berikut ini.

**Tabel 6.10 Kode Program Fungsi alert\_pesanberhasil**

```

1      private void alert_pesanberhasil() {
2          AlertDialog.Builder peringatan=new
3      AlertDialog.Builder(Pengaduan.this,AlertDialog.THEME_HOLO_L
4      IGH);
5          peringatan.setTitle("Berhasil");
6          peringatan.setIcon(R.drawable.ok);
7          peringatan.setMessage("Pengaduan Anda Telah
8      Terkirim");
9          peringatan.setNeutralButton("OK", new
10     DialogInterface.OnClickListener() {
11         public void onClick(DialogInterface dialog, int
12     id) {
13             Intent intent = new Intent(Pengaduan.this,
14     Home.class);
15             startActivity(intent);
16         }
17     });
18     peringatan.show();
19     }
    
```

**Tabel 6.11 Penjelasan Kode Program Fungsi alert\_pesanberhasil**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama alert_pesanberhasil
2-4	Inisialisasi objek peringatan untuk menampilkan boxdialog
5-8	Peringatan melakukan setting untuk boxdialog
9-15	Peringatan melakukan fungsi onClick saat memilih OK yang didalamnya untuk melakukan perpindahan halaman
18	Peringatan ditampilkan

### 6.3.1.2 Implementasi Prosedur Menampilkan Daftar Pengaduan

Implementasi yang digunakan pada Menampilkan Daftar Pengaduan yang telah dikirimkan oleh Masyarakat dengan memiliki fungsi sebagai berikut.

#### 1. Fungsi DataAduan

Fungsi DataAduan berfungsi untuk mengambil data pelaporan pengaduan masyarakat dengan parameter NIK Masyarakat Kota yang telah *login* kedalam sistem. Selanjutnya akan dilakukan inisialisasi sebagai penghubung kepada *webservice* dan memanggil *service* yang telah disediakan untuk mengambil data pelaporan pengaduan masyarakat sesuai NIK melalui *webservice* dengan nama *service* listaduanRequest. Didalamnya terdapat fungsi onResponse digunakan apabila berhasil terhubung kedalam *webservice* dan akan menyimpan seluruh data laporan pengaduan yang tersimpan pada sistem kedalam model kemudian akan ditampilkan pada halaman daftar pengaduan, sedangkan onFailure digunakan untuk menampilkan pesan error apabila tidak terhubung dengan *webservice*. Kode Program Fungsi DataAduan pada Proses Menampilkan Daftar Pengaduan dapat dilihat pada Tabel 6.12 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.13 berikut ini.

**Tabel 6.12 Kode Program Fungsi DataAduan**

1	<code>public void DataAduan(String nik_member)</code>
2	<code>{</code>
3	<code>Log.d("isi inputan", "nik: "+nik_member);</code>
4	<code>service=</code>
5	<code>ServiceGenerator.createService(ApiService.class);</code>
6	<code>CallBody=service.listaduanRequest(nik_member);</code>
7	<code>CallBody.enqueue(new Callback&lt;ListAduanResponse&gt;()</code>
8	<code>{</code>
9	<code>    @Override</code>
10	<code>    public void</code>
11	<code>onResponse(Response&lt;ListAduanResponse&gt; response) {</code>
12	<code>        model=response.body();</code>
13	<code>        if(model.error)</code>
14	<code>        {</code>
15	<code>            Toast.makeText(DaftarPengaduan.this,</code>
16	<code>"gagal", Toast.LENGTH_SHORT).show();</code>

**Tabel 6.12 Kode Program Fungsi DataAduan (Lanjutan)**

```

17         Log.d("coba2 semoaga keluar", new
18 Gson().toJson(response.body()) );
19     }
20     if(!model.error)
21     {
22         for (int
23 i=0;i<model.data_aduan.size();i++){
24             listPengaduan.add(new
25 PengaduanModel(model.data_aduan.get(i).ID_PESAN,
26
27 model.data_aduan.get(i).TGL_ADUAN,model.data_aduan.get(i).
28 STATUS_ADUAN,model.data_aduan.get(i).RESPON_PESAN));
29         }
30         adapter.notifyDataSetChanged();
31         loading.dismiss();
32     }
33     }
34
35     @Override
36     public void onFailure(Throwable t) {
37         Toast.makeText(DaftarPengaduan.this,
38 t.getMessage(), Toast.LENGTH_SHORT).show();
39         Log.d("coba2 semoaga keluar",
40 t.getMessage() );
41     }
42     });
43 }
44 }

```

**Tabel 6.13 Penjelasan Kode Program Fungsi DataAduan**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama DataAduan dengan memiliki parameter nik_nember
3	Menampilkan kedalam konsole agar mengetahui masukan parameter di fungsi DataAduan
4-5	Inialisasi objek service sebagai penghubung dengan <i>webservice</i> sekaligus memanggil <i>class ApiService</i> sebagai penyedia fungsi dari <i>webservice</i>
6	Inialisasi objek Callbody untuk memanggil service sekaligus fungsi listaduanRequest yang memiliki parameter
7	Callbody melakukan pemanggilan selanjutnya data yang didapat ditampung pada class ListAduanResponse
10-11	Deklarasi fungsi onResponse digunakan apabila pemanggilan berhasil dengan memanggil ListAduanResponse
12	Inialisasi objek model dari class PesanResponse untuk menyimpan data JSON dari ListAduanResponse



**Tabel 6.13 Penjelasan Kode Program Fungsi DataAduan (Lanjutan)**

13-18	Jika data model index error <i>true</i> maka akan ditampilkan berupa toast berisi “gagal”
20-30	Jika index error <i>false</i> maka model akan melakukan perulangan untuk mengambil seluruh model dengan index data_aduan untuk disimpan dalam objek PengaduanModel selanjutnya memanggil notifyDatasetChanged agar data yang akan ditampilkan berubah dengan data terbaru
35-39	Deklarasi fungsi onFailure yang memiliki parameter digunakan apabila pemanggilan gagal dan menampilkan data parameter berupa toast

### 6.3.2 Program Berbasis Website

#### 6.3.2.1 Kelas Controlpengaduan

Kelas Controlpengaduan merupakan kelas yang berfungsi untuk memproses data pengaduan pada Sistem Layanan Pengaduan Masyarakat yang digunakan oleh Petugas Administrasi berbasis website dengan memiliki beberapa fungsi sebagai berikut.

##### 1. Fungsi getHalamanProsesAduan

Fungsi getHalamanProsesAduan berfungsi untuk menampilkan informasi seluruh laporan pengaduan Masyarakat yang tersimpan pada sistem dan akan diberikan pilihan apabila memilih menampilkan laporan pengaduan berstatus sudah diproses ataupun belum diproses akan ditampilkan sesuai status laporan pengaduan yang diinginkan pada halaman data pengaduan. Kode Program Fungsi getHalamanProsesAduan pada kelas Controlpengaduan dapat dilihat pada Tabel 6.14 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.15 berikut ini.

**Tabel 6.14 Kode Program Fungsi getHalamanProsesAduan**

1	<code>public function getHalamanProsesAduan() {</code>
2	<code>    \$data['opsi'] = 0;</code>
3	<code>    \$data['list_aduan'] = NULL;</code>
4	<code>    if (isset(\$_POST['opsi_aduan'])) {</code>
5	<code>        \$data['opsi'] = (int) \$_POST['opsi_aduan'];</code>
6	<code>        if (\$data['opsi'] == 1) {</code>
7	<code>            \$this-&gt;load-&gt;model('M_pengaduan');</code>
8	<code>            \$status_aduan = 'BELUM PROSES';</code>
9	<code>            \$data['list_aduan'] =</code>
10	<code>\$this-&gt;M_pengaduan-&gt;blmproses_aduan(\$status_aduan);</code>
11	<code>            \$this-&gt;load-&gt;view('adm_listaduan', \$data);</code>
12	<code>        } elseif (\$data['opsi'] == 2) {</code>
13	<code>            \$this-&gt;load-&gt;model('M_pengaduan');</code>
14	<code>            \$status_aduan = 'SUDAH PROSES';</code>
15	<code>            \$data['list_aduan'] =</code>
16	<code>\$this-&gt;M_pengaduan-&gt;sdhproses_aduan(\$status_aduan);</code>
17	<code>            \$this-&gt;load-&gt;view('adm_listaduan', \$data);</code>

**Tabel 6.14 Kode Program Fungsi getHalamanProsesAduan (Lanjutan)**

```

18 } else {
19     $this->load->model('M_pengaduan');
20     $data['list_aduan'] = $this->M_pengaduan
21     ->all_aduan();
22     $this->load->view('adm_listaduan', $data);
23     }
24     } else {
25     $this->load->model('M_pengaduan');
26     $data['list_aduan'] =
27     $this->M_pengaduan->all_aduan();
28     $this->load->view('adm_listaduan', $data);
29     }
30     }
    
```

**Tabel 6.15 Penjelasan Kode Program Fungsi getHalamanProsesAduan**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getHalamanProsesAduan
2-3	Inisialisasi nilai objek data dengan tipe data array yang memiliki index opsi dan list_aduan
4	Pengecekan apakah inputan opsi_aduan memiliki nilai
5	Jika opsi_aduan memiliki nilai benar maka variabel data akan menyimpan inputan opsi_aduan ke index opsi
6-8	pengecekan nilai index opsi, jika bernilai 1 maka akan memanggil model M_pengaduan, kemudian inisialisasi variabel status_aduan dengan nilai Belum Proses
9-10	variabel data dengan index list_aduan akan menyimpan pengembalian nilai dari memanggil fungsi blomproses_aduan dengan parameter status_aduan pada model M_pengaduan
11	nilai variabel data kemudian dikirimkan pada halaman view adm_listaduan
12-14	pengecekan nilai index opsi, jika bernilai 2 maka akan memanggil model M_pengaduan, kemudian inisialisasi variabel status_aduan dengan nilai Sudah Proses
15-16	variabel data dengan index list_aduan akan menyimpan pengembalian nilai dari memanggil fungsi blomproses_aduan dengan parameter status_aduan pada model M_pengaduan
17	nilai variabel data kemudian dikirimkan pada halaman view adm_listaduan
18-19	pengecekan nilai index opsi, jika bukan bernilai 1 atau 2 maka akan memanggil model M_pengaduan
20-21	variabel data dengan index list_aduan akan menyimpan pengembalian nilai dari memanggil fungsi all_aduan pada model M_pengaduan



**Tabel 6.15 Penjelasan Kode Program Fungsi getHalamanProsesAduan (Lanjutan)**

22	nilai variabel data kemudian dikirimkan pada halaman view adm_listaduan
24-25	Jika opsi_aduan tidak memiliki nilai maka akan memanggil model M_pengaduan
26-27	variabel data dengan index list_aduan akan menyimpan pengembalian nilai dari memanggil fungsi all_aduan pada model M_pengaduan
28	nilai variabel data kemudian dikirimkan pada halaman view adm_listaduan

## 2. Fungsi getUpdateResponAduan

Fungsi getUpdateResponAduan berfungsi melakukan respon pengaduan yang telah dikirimkan oleh Masyarakat Kota Pasuruan sesuai pengaduan yang diinginkan. Kode Program Fungsi getUpdateResponPengaduan pada kelas Controlpengaduan dapat dilihat pada Tabel 6.16 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.17 berikut ini.

**Tabel 6.16 Kode Program Fungsi getUpdateResponPengaduan**

1	<code>public function getUpdateResponPengaduan() {</code>
2	<code>    \$this-&gt;load-&gt;model('M_pengaduan');</code>
3	<code>    \$this-&gt;M_pengaduan-&gt;updaterespon_aduan(\$_POST['idpesan'],</code>
4	<code>    \$_POST['responpesan']);</code>
5	<code>    redirect('Controlpengaduan/getHalamanProsesAduan?id=1',</code>
6	<code>    'refresh');</code>
7	<code>}</code>

**Tabel 6.17 Penejelasan Kode Program Fungsi getUpdateResponPengaduan**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getUpdateResponPengaduan
2	Memanggil model M_pengaduan
3-4	Memanggil fungsi updaterespon_aduan dengan parameter inputan idpesan dan responpesan pada model M_pengaduan
5	Memanggil fungsi getHalamanProsesAduan pada Controller Controlpengaduan

## 3. Fungsi getStatusAduan

Fungsi getStatusAduan berfungsi mengubah status laporan pengaduan Masyarakat Kota Pasuruan, seperti status yang dari belum diproses menjadi sudah diproses. Kode Program Fungsi getUpdateStatusAduan pada kelas

Controlpengaduan dapat dilihat pada Tabel 6.18 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.19 berikut ini.

**Tabel 6.18 Kode Program Fungsi getStatusAduan**

1	public function getStatusAduan(\$idpesan, \$statuspesan)
2	{
3	\$this->load->model('M_pengaduan');
4	\$this->M_pengaduan->ubahstatus_aduan(\$idpesan,
5	\$statuspesan);
6	redirect('Controlpengaduan/getHalamanProsesAduan?id=2',
7	'refresh');
8	}
9	}

**Tabel 6.19 Penjelasan Kode Program Fungsi getStatusAduan**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getStatusAduan dengan parameter idpesan dan statuspesan
2	Memanggil model M_pengaduan
3-4	Memanggil fungsi updaterespon_aduan dengan parameter inputan idpesan dan statuspesan pada model M_pengaduan
5	Memanggil fungsi getHalamanProsesAduan pada Controller Controlpengaduan

4. Fungsi getHapusAduan

Fungsi getHapusAduan berfungsi menghapus laporan pengaduan Masyarakat Kota Pasuruan yang diinginkan dengan syarat status pengaduannya telah sudah diproses apabila belum diproses akan gagal menghapus laporan pengaduan yang diinginkan. Kode Program Fungsi getUpdateStatusAduan pada kelas Controlpengaduan dapat dilihat pada Tabel 6.20 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.21 berikut ini.

**Tabel 6.20 Kode Program Fungsi getHapusAduan**

1	public function getHapusAduan(\$idpesan, \$statuspesan) {
2	if (\$statuspesan == 'BELUM%20PROSES') {
3	redirect('Controlpengaduan/getHalamanProsesAduan?id=4',
4	'refresh');
5	} else {
6	\$this->load->model('M_pengaduan');
7	\$this->M_pengaduan->hapus_aduan(\$idpesan);
8	redirect('Controlpengaduan/getHalamanProsesAduan?id=3',
9	'refresh');
10	}
11	}
12	

**Tabel 6.21 Penjelasan Kode Program Fungsi getHapusAduan**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getHapusAduan dengan parameter idpesan dan statuspesan
2-3	Pengecekan jika statuspesan bernilai BelumProses, maka akan memanggil fungsi getHalamanProsesAduan pada Controller Controlpengaduan
5-6	Jika statuspesan bukan bernilai BelumProses maka memanggil model M_pengaduan
7	Memanggil fungsi hapus_aduan dengan parameter idpesan pada model M_pengaduan
8	Memanggil fungsi getHalamanProsesAduan pada Controller Controlpengaduan

### 6.3.2.2 Kelas Controladmin

Kelas Controladmin merupakan kelas yang berfungsi untuk memproses data pengguna maupun informasi pada Sistem Layanan Pengaduan Masyarakat yang digunakan oleh Petugas Administrasi berbasis *website* dengan memiliki beberapa fungsi sebagai berikut.

#### 1. Fungsi index

Fungsi index berfungsi menampilkan halaman utama / *login* sebelum masuk halaman beranda sistem. Kode Program Fungsi index pada kelas Controladmin dapat dilihat pada Tabel 6.22 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.23 berikut ini.

**Tabel 6.22 Kode Program Fungsi index**

1	<code>public function index() {</code>
2	<code>    \$this-&gt;load-&gt;view('portal');</code>
3	<code>}</code>

**Tabel 6.23 Penjelasan Kode Program Fungsi index**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama index
2	Memanggil view dengan halaman portal



## 2. Fungsi getHalamanBeranda

Fungsi `getHalamanBeranda` berfungsi menampilkan informasi berupa agregat laporan pengaduan dan Kabid. Yang sering mendapatkan laporan pengaduan pada Halaman Beranda sistem. Kode Program Fungsi `getHalamanBeranda` pada kelas `Controladmin` dapat dilihat pada Tabel 6.24 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.25 berikut ini.

**Tabel 6.24 Kode Program Fungsi `getHalamanBeranda`**

1	<code>public function getHalamanBeranda() {</code>
2	<code>    \$data['blumproses'] = 0;</code>
3	<code>    \$data['sudahproses'] = 0;</code>
4	<code>    \$data['tjn_dafduk'] = 0;</code>
5	<code>    \$data['tjn_capil'] = 0;</code>
6	<code>    \$data['tjn_piak'] = 0;</code>
7	<code>    \$this-&gt;load-&gt;model('M_pengaduan');</code>
8	<code>    \$data['list_aduan'] =</code>
9	<code>    \$this-&gt;M_pengaduan-&gt;daftar_aduan();</code>
10	<code>    \$this-&gt;load-&gt;model('M_pengguna');</code>
11	<code>    \$data['list_member'] =</code>
12	<code>    \$this-&gt;M_pengguna-&gt;daftar_pengguna();</code>
13	
14	<code>    foreach (\$data['list_aduan'] as \$values) {</code>
15	<code>        if (\$values-&gt;BIDANG_TUJUAN == 'BID. PENCATATAN</code>
16	<code>    SIPIIL') {</code>
17	<code>            \$data['tjn_dafduk']+=1;</code>
18	<code>        } elseif (\$values-&gt;BIDANG_TUJUAN == 'BID.</code>
19	<code>    PENDAFTARAN PENDUDUK') {</code>
20	<code>            \$data['tjn_capil']+=1;</code>
21	<code>        } elseif (\$values-&gt;BIDANG_TUJUAN == 'BID. PIAK</code>
22	<code>    DAN PEMANFAATAN DATA') {</code>
23	<code>            \$data['tjn_piak']+=1;</code>
24	<code>        }</code>
25	<code>        if (\$values-&gt;STATUS_ADUAN == 'BELUM PROSES') {</code>
26	<code>            \$data['blumproses']+=1;</code>
27	<code>        } else {</code>
28	<code>            \$data['sudahproses']+=1;</code>
29	<code>        }</code>
30	<code>    }</code>
31	<code>    \$this-&gt;load-&gt;view('adm_beranda', \$data);</code>
32	<code>}</code>

**Tabel 6.25 Penjelasan Kode Program Fungsi `getHalamanBeranda`**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama <code>getHalamanBeranda</code>
2-6	Inisialisasi nilai objek data dengan tipe data array yang memiliki index <code>blumproses,sudahproses,tjn_dafduk,tjn_capil</code> dan <code>tjn_piak</code>



**Tabel 6.25 Penjelasan Kode Program Fungsi getHalamanBeranda (Lanjutan)**

7	Memanggil model M_pengaduan
8-9	Variabel data dengan index list_aduan akan menyimpan pengembalian nilai dari memanggil fungsi daftar_aduan pada model M_pengaduan
10	Memanggil model M_pengguna
11-12	Variabel data dengan index list_member akan menyimpan pengembalian nilai dari memanggil fungsi daftar_member pada model M_pengguna
14-28	Perulangan untuk melakukan penjumlahan setiap bidang pengaduan dan status pengaduannya yang akan disimpan di variabel data di index tjn_dafduk, tjn_capil, tjn_piak, blomproses dan sudahproses
31	Nilai variabel data dikirimkan pada halaman view adm_beranda

### 3. Fungsi getHalamanPegguna

Fungsi getHalamanPegguna berfungsi menampilkan informasi seluruh data pengguna yang terdaftar pada halaman data pengguna . Kode Program Fungsi getHalamanPegguna pada kelas Controladmin dapat dilihat pada Tabel 6.26 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.27 berikut ini.

**Tabel 6.26 Kode Program Fungsi getHalamanPegguna**

1	<code>public function getHalamanPegguna() {</code>
2	<code>    \$this-&gt;load-&gt;model('M_pengguna');</code>
3	<code>    \$data['daftar'] =</code>
4	<code>    \$this-&gt;M_pengguna-&gt;daftar_pengguna();</code>
5	<code>    \$this-&gt;load-&gt;view('adm_listpengguna', \$data);</code>
6	<code>}</code>

**Tabel 6.27 Penjelasan Kode Program Fungsi getHalamanPegguna**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getHalamanPegguna Memanggil model M_pengguna
2-3	Variabel data dengan index daftar akan menyimpan pengembalian nilai dari memanggil fungsi daftar_pengguna pada model M_pengguna
5-6	Nilai variabel data kemudian dikirimkan pada halaman view adm_listpengguna

#### 4. Fungsi getDataLogin

Fungsi `getDataLogin` berfungsi melakukan verifikasi *username* dan *password* yang dimasukkan oleh petugas administrasi sebelum masuk kedalam halaman beranda sistem. Kode Program Fungsi `getDataLogin` pada kelas `Controladmin` dapat dilihat pada Tabel 6.28 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.29 berikut ini.

**Tabel 6.28 Kode Program Fungsi `getDataLogin`**

1	<code>public function getDataLogin() {</code>
2	<code>    \$this-&gt;load-&gt;model('M_pengguna');</code>
3	<code>    \$data =</code>
4	<code>    \$this-&gt;M_pengguna-&gt;cekdatalogin(\$_POST['username'],</code>
5	<code>    \$_POST['password']);</code>
6	<code>        if (\$data-&gt;num_rows() == 1) {</code>
7	<code>            foreach (\$data-&gt;result() as \$value) {</code>
8	<code>                \$data_login['NIK'] = \$value-&gt;NIK;</code>
9	<code>                \$data_login['NAMA LENGKAP'] =</code>
10	<code>                \$value-&gt;NAMA LENGKAP;</code>
11	<code>                \$data_login['NO_HP'] = \$value-&gt;NO_HP;</code>
12	<code>                \$data_login['PASSWORD'] = \$value-&gt;PASSWORD;</code>
13	<code>                \$this-&gt;session-&gt;set_userdata(\$data_login);</code>
14	<code>            }</code>
15	<code>            redirect('Controladmin/getHalamanBeranda',</code>
16	<code>'refresh');</code>
17	<code>        } else {</code>
18	<code>            echo " &lt;script&gt;alert('Gagal Login: Cek</code>
19	<code>username , password!');history.go(-1);&lt;/script&gt;";</code>
20	<code>        }</code>
21	<code>    }</code>
22	<code>}</code>

**Tabel 6.29 Penjelasan Kode Program Fungsi `getDataLogin`**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama <code>getDataLogin</code>
2	Memanggil model <code>M_pengguna</code>
3-5	Variabel <code>data</code> akan menyimpan pengembalian nilai dari memanggil fungsi <code>cekdatalogin</code> dengan parameter inputan <code>username</code> dan <code>password</code> pada model <code>M_pengguna</code>
6-14	Pengecekan variabel <code>data</code> apabila jumlah baris bernilai 1 maka dilakukan pengambilan nilai dari <code>data</code> ke dalam variabel <code>data_login</code> bertipe array kemudian disimpan kedalam fungsi <code>set_userdata</code> pada <code>session</code>
15	Memanggil fungsi <code>getHalamanBeranda</code> pada Controller <code>Controladmin</code> Jika variabel <code>data</code> jumlah baris bukan bernilai 1, maka akan
17-19	menampilkan peringatan gagal login

5. Fungsi getTambahPegguna

Fungsi getTambahPegguna berfungsi melakukan pendaftaran Masyarakat Kota Pasuruan sebagai pengguna Sistem Layanan Pengaduan Masyarakat melalui petugas administrasi dengan dilakukan verifikasi identitas harus berKTP Kota Pasuruan dan verikasi selanjutnya apakah no\_hp telah terdaftar atau tidak jika tidak terdaftar maka berhasil terdaftar kedalam sistem. Kode Program Fungsi getTambahPegguna pada kelas Controladmin dapat dilihat pada Tabel 6.30 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.31 berikut ini.

**Tabel 6.30 Kode Program Fungsi getTambahPegguna**

```

1      public function getTambahPegguna() {
2          $this->load->model('M_pegguna');
3          $tanggal_lhr =
4      date('d/M/Y', (strtotime($_POST['tgllahir'])));
5          $cekidentitas =
6      $this->M_pegguna->cek_KTPPaskot($tanggal_lhr,$_POST['nik']
7      ,strtoupper($_POST['namalengkap']));
8          if (count($cekidentitas) == 0) {
9
10         redirect('Controladmin/getHalamanPegguna?id=2',
11         'refresh');
12         } else {
13             $result =
14         $this->M_pegguna->cek_pegguna($_POST['nik'],$_POST['no_hp
15         ']);
16             if ($result->num_rows() == 1) {
17
18         redirect('Controladmin/getHalamanPegguna?id=3',
19         'refresh');
20         } else {
21
22         $this->M_pegguna->tambah_pegguna($_POST['nik'],$_POST['na
23         malengkap'],$_POST['no_hp'],$_POST['password'],$tanggal_lhr
24         );
25
26         redirect('Controladmin/getHalamanPegguna?id=1',
27         'refresh');
28         }
29     }
30 }
    
```

**Tabel 6.31 Penjelasan Kode Program Fungsi getTambahPegguna**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getTambahPegguna
2	Memanggil model M_pegguna
3-4	Variabel tanggal_lhr akan menyimpan inputan tgllahir
5-7	Variabel cekidentitas akan menyimpan pengembalian nilai dari



**Tabel 6.31 Penjelasan Kode Program Fungsi getTambahPegguna (Lanjutan)**

8-10	memanggil fungsi cek_KTPPaskot dengan parameter tanggal_lhr, inputan nik dan namalengkap pada model M_pengguna Pengecekan cekidentitas apabila jumlah baris bernilai 0, maka memanggil fungsi getHalamanPegguna pada Controller Controladmin
12-15	Jika cekidentitas jumlah baris bukan bernilai 0, maka variabel result akan menyimpan pengembalian nilai dari memanggil fungsi cek_pengguna dengan parameter inputan nik dan no_hp pada model M_pengguna
16-19	Pengecekan result apabila jumlah baris bernilai 1, maka memanggil fungsi getHalamanPegguna pada Controller Controladmin
20-24	Jika result jumlah baris bukan bernilai 1, maka memanggil fungsi tambah_pengguna dengan parameter inputan nik, namalengkap, no_hp, password dan tanggal_lhr pada model M_pengguna
26	Memanggil fungsi getHalamanPegguna pada Controller Controladmin

6. Fungsi getStatusPegguna

Fungsi getStatusPegguna berfungsi mengubah status pengguna yang terdaftar dalam sistem yang digunakan untuk penonaktifan ataupun pengaktifan pengguna . Kode Program Fungsi getStatusPegguna pada kelas Controladmin dapat dilihat pada Tabel 6.32 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.33 berikut ini.

**Tabel 6.32 Kode Program Fungsi getStatusPegguna**

```

1 public function getStatusPegguna($nik, $status) {
2     $this->load->model('M_pengguna');
3     $this->M_pengguna->ubahstatus_pengguna($nik,
4     $status);
5     redirect('Controladmin/getHalamanPegguna?id=4',
6     'refresh');
7     }
    
```

**Tabel 6.33 Penjelasan Kode Program Fungsi getStatusPegguna**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getStatusPegguna dengan parameter nik dan status
2	Memanggil model M_pengguna
3-4	Memanggil fungsi ubahstatus_pengguna dengan parameter nik dan status pada model M_pengguna
5	Memanggil fungsi getHalamanPegguna pada Controller Controladmin



## 7. Fungsi getHapusPegguna

Fungsi `getHapusPegguna` berfungsi menghapus pengguna yang telah terdaftar di dalam sistem dan otomatis akan melakukan penghapusan laporan pengaduan yang pernah dilakukan oleh pengguna tersebut. Kode Program Fungsi `getHapusPegguna` pada kelas `Controladmin` dapat dilihat pada Tabel 6.34 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.35 berikut ini.

**Tabel 6.34 Kode Program Fungsi getHapusPegguna**

1	<code>public function getHapusPegguna(\$nik) {</code>
2	<code>    \$this-&gt;load-&gt;model('M_pegguna');</code>
3	<code>    \$this-&gt;M_pegguna-&gt;hapus_pegguna(\$nik);</code>
4	<code>    \$this-&gt;load-&gt;model('M_pengaduan');</code>
5	<code>    \$this-&gt;M_pengaduan-&gt;hapus_slruhaduan(\$nik);</code>
6	<code>    redirect('Controladmin/getHalamanPegguna?id=5',</code>
7	<code>    'refresh');</code>
8	<code>}</code>

**Tabel 6.35 Penjelasan Kode Program Fungsi getHapusPegguna**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama <code>getHapusPegguna</code> dengan parameter <code>nik</code>
2	Memanggil model <code>M_pegguna</code>
3	Memanggil fungsi <code>hapus_pegguna</code> dengan parameter <code>nik</code> pada model <code>M_pegguna</code>
4	Memanggil model <code>M_pengaduan</code>
5	Memanggil fungsi <code>hapus_slruhaduan</code> dengan parameter <code>nik</code> pada model <code>M_pengaduan</code>
6	Memanggil fungsi <code>getHalamanPegguna</code> pada Controller <code>Controladmin</code>

## 8. Fungsi getUpdatePasswordPegguna

Fungsi `getUpdatePasswordPegguna` berfungsi mengubah *password* pengguna melalui petugas administrasi dengan dilakukan verifikasi *password* lama apabila sesuai akan diubah dengan *password* baru. Kode Program Fungsi `getUpdatePasswordPegguna` pada kelas `Controladmin` dapat dilihat pada Tabel 6.36 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.37 berikut ini.



**Tabel 6.36 Kode Program Fungsi getUpdatePasswordPengguna**

```

1      public function getUpdatePasswordPengguna() {
2          $nama_lengkap = $_POST['id1'];
3          $password = md5(strtoupper($_POST['pwdbaru']));
4          $passwordlama = md5(strtoupper($_POST['pwdlama']));
5          $this->load->model('M_pengguna');
6          $result =
7      $this->M_pengguna->cek_passwordlama($nama_lengkap,
8      $passwordlama);
9          if ($result->num_rows() == 1) {
10     $this->M_pengguna->update_password($nama_lengkap,$password,
11     $passwordlama);
12
13     redirect('Controladmin/getHalamanPengguna?id=7',
14     'refresh');
15         } else {
16             redirect('Controladmin/getHalamanPengguna?id=6',
17     'refresh');
18         }
19     }

```

**Tabel 6.37 Penjelasan Kode Program Fungsi getUpdatePasswordPengguna**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama getUpdatePasswordPengguna
2-4	Inisialisasi nilai objek nama lengkap dengan inputan id1, password dengan inputan pwdbaru dengan enkripsi md5 dan passwordlama dengan inputan pwdlama dengan enkripsi md5
5	Memanggil model M_pengguna
6-8	Variabel result akan menyimpan pengembalian nilai dari memanggil fungsi cek_passwordlama dengan parameter nama_lengkap dan passwordlama pada model M_pengguna
9-11	Pengecekan result apabila jumlah baris bernilai 1, maka akan memanggil fungsi update_password dengan parameter nama_lengkap, password dan passwordlama pada model M_pengguna
13	Memanggil fungsi getHalamanPengguna pada Controller Controladmin
15-16	Jika result jumlah baris bukan bernilai 1, maka akan memanggil fungsi getHalamanPengguna pada Controller Controladmin

**9. Fungsi logout**

Fungsi logout berfungsi untuk keluar dari sistem dan kembali pada halaman utama atau *login* dengan menghapus data session terlebih dahulu. Kode Program Fungsi logout pada kelas Controladmin dapat dilihat pada Tabel 6.38 dan penjelasan dari setiap kode program dapat dilihat pada Tabel 6.39 berikut ini.

**Tabel 6.38 Kode Program Fungsi logout**

1	<code>public function logout() {</code>
2	<code>    \$this-&gt;session-&gt;sess_destroy();</code>
3	<code>    redirect();</code>
4	<code>}</code>

**Tabel 6.39 Penjelasan Kode Program Fungsi logout**

Baris Kode	Penjelasan
1	Deklarasi nama fungsi bernama logout
2-4	Memanggil fungsi sess_destroy pada Session
3	Memanggil fungsi index pada Conctroller Controladmin

## 6.4 Implementasi Antarmuka

Implementasi antarmuka Sistem Layanan Pengaduan Masyarakat telah dirancang pada sub bab 5.4. Hasil implementasi antarmuka dari perancangan ini akan dibagi menjadi 2 bagian yaitu berbasis *website* dan berbasis Android.

### 6.4.1 Antarmuka Berbasis Android

#### 6.4.1.1 Halaman Login

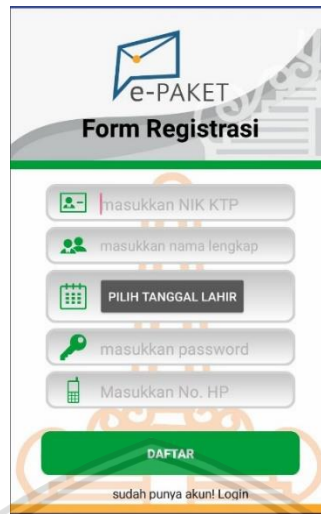


**Gambar 6.1 Implementasi Antarmuka Login Berbasis Android**

Pada Gambar 6.1 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.21. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan yang sebelum masuk kedalam Sistem Layanan Pengaduan Masyarakat diharuskan terlebih dahulu melakukan *login* dengan memasukkan No. HP dan *password*.



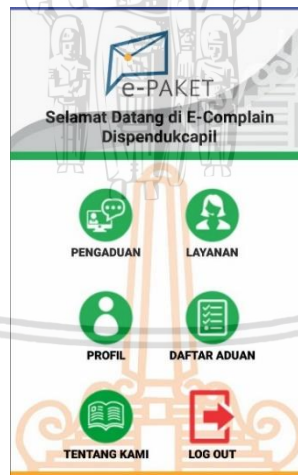
#### 6.4.1.2 Halaman Registrasi



**Gambar 6.2 Implementasi Antarmuka Registrasi Berbasis Android**

Pada Gambar 6.2 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.22. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan apabila belum memiliki akun maka diharuskan mendaftar terlebih dahulu dengan mengisi Form Registrasi yang memiliki syarat pendaftar harus ber KTP Kota Pasuruan.

#### 6.4.1.3 Halaman Beranda



**Gambar 6.3 Implementasi Antarmuka Beranda Berbasis Android**

Pada Gambar 6.3 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.23. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan yang menampilkan menu / fitur yang tersedia sekaligus menjadi halaman utama pada Sistem Layanan Pengaduan.

#### 6.4.1.4 Halaman Tentang Kami



**Gambar 6.4 Implementasi Antarmuka Tentang Kami Berbasis Android**

Pada Gambar 6.4 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.24. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan yang menampilkan informasi singkat mengenai Dinas Kependudukan Dan Pencatatan Sipil Kota Pasuruan.

#### 6.4.1.5 Halaman Layanan



**Gambar 6.5 Implementasi Antarmuka Layanan Berbasis Android**

Pada Gambar 6.5 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.25. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan yang menampilkan informasi mengenai pelayanan yang tersedia pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan, seperti : Layanan E-KTP, Layanan Kartu Keluarga, Layanan Akte Kelahiran dan Layanan Akte Kematian.

### 6.4.1.6 Halaman NavigationView



**Gambar 6.6 Implementasi Antarmuka NavigationView Berbasis Android**

Pada Gambar 6.6 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.26. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan yang menampilkan menu-menu yang tersedia tanpa perlu harus ke halaman utama Sistem Layanan Pengaduan hanya dengan menekan tombol *toggle bar* di pojok kanan atas atau hanya mengeser tampilan layar ke kanan.

### 6.4.1.7 Halaman Pengaduan



**Gambar 6.7 Implementasi Antarmuka Pengaduan Berbasis Android**



Pada Gambar 6.7 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.27. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan apabila akan melakukan pelaporan pengaduan diharuskan mengisi Form Pengaduan terlebih dahulu meliputi tujuan, kategori, topik dan pesan pengaduan.

#### 6.4.1.8 Halaman Daftar Pengaduan



Gambar 6.8 Implementasi Antarmuka Daftar Pengaduan Berbasis Android

Pada Gambar 6.8 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.28. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan apabila ingin melihat laporan pengaduan yang pernah dilakukan. Masyarakat Kota Pasuruan juga dapat melihat respon dari pengaduan yang telah dikirimkan dengan menekan tombol status pengaduan maka akan muncul *pop-up* pesan berisi isi respon dari laporan pengaduan tersebut.

#### 6.4.1.9 Halaman Profil



Gambar 6.9 Implementasi Antarmuka Profil Berbasis Android

Pada Gambar 6.9 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.29. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan yang menampilkan informasi mengenai identitas biodata diri pengguna yang telah *login* kedalam Sistem Layanan Pengaduan Masyarakat.

#### 6.4.1.10 Halaman Ubah *Password*

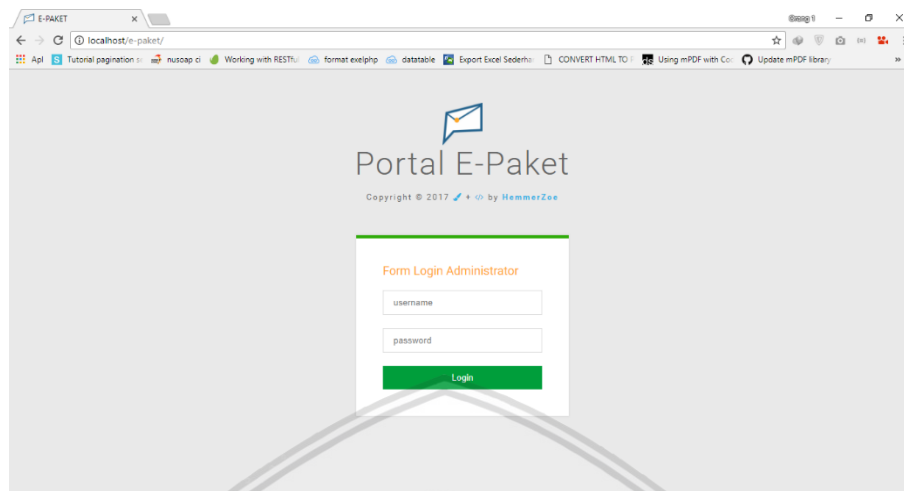


**Gambar 6.10 Implementasi Antarmuka Ubah *Password* Berbasis Android**

Pada Gambar 6.10 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.30. Pada halaman ini ditujukan kepada Masyarakat Kota Pasuruan apabila akan melakukan mengubah *password* dari akunnya yang telah tersimpan di sistem dengan mengisi form ubah *password* terlebih dahulu meliputi *password* lama, *password* terbaru dan konfirmasi ulang *password* terbaru.

## 6.4.2 Antarmuka Berbasis Website

### 6.4.2.1 Halaman Login



**Gambar 6.11 Implementasi Antarmuka Login Berbasis Website**

Pada Gambar 6.11 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.31. Pada halaman ini ditujukan kepada Petugas Administrasi yang sebelum masuk kedalam Sistem Layanan Pengaduan Masyarakat diharuskan terlebih dahulu melakukan *login* dengan memasukkan *username* dan *password*.

### 6.4.2.2 Halaman Beranda

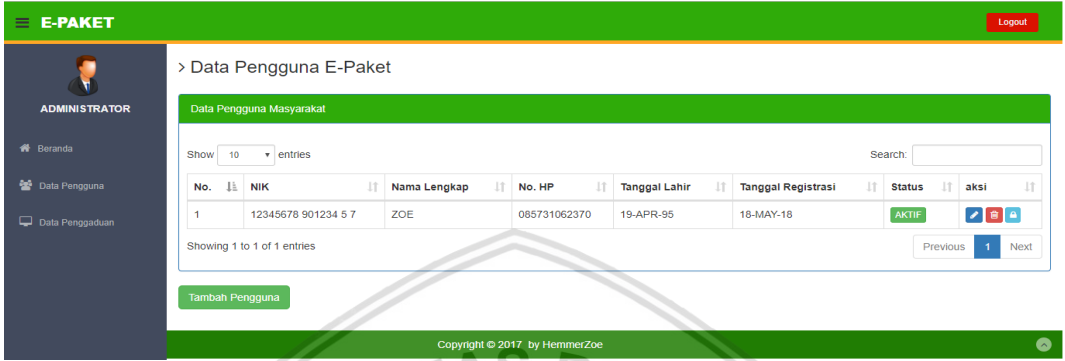


**Gambar 6.12 Implementasi Antarmuka Beranda Berbasis Website**

Pada Gambar 6.12 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.32. Pada halaman ini ditujukan kepada Petugas Administrasi yang

menampilkan informasi mengenai agregat pengaduan berupa total pengaduan dan status pengaduan yang telah diterima. Halaman ini juga menampilkan grafik dari status pengaduan yang telah diproses ataupun belum diproses dan grafik mengenai Kabid. Yang banyak menerima pengaduan oleh masyarakat Kota Pasuruan.

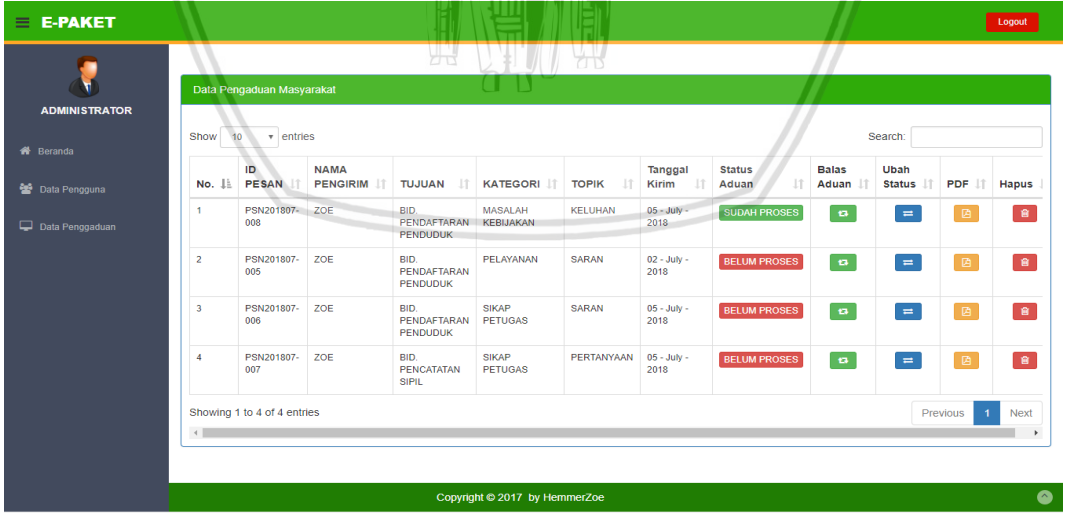
**6.4.2.3 Halaman Data Pengguna**



**Gambar 6.13 Implementasi Antarmuka Data Pengguna Berbasis Website**

Pada Gambar 6.13 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.33. Pada halaman ini ditujukan kepada Petugas Administrasi yang mengelola data pengguna oleh masyarakat Kota Pasuruan yang telah terdaftar kedalam sistem. Halaman ini disediakan fitur-fitur antara lain menambah daftar pengguna aplikasi melalui Petugas Administrasi, menon-aktifkan sementara akun, menghapus akun dan mengubah password akun Masyarakat Kota Pasuruan yang telah terdaftar kedalam sistem.

**6.4.2.4 Halaman Data Pengaduan**



**Gambar 6.14 Implementasi Antarmuka Data Pengaduan Berbasis Website**

Pada Gambar 6.14 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.34. Pada halaman ini ditujukan kepada Petugas Administrasi yang

repository.ub.ac.id

mengelola data pengaduan yang dikirimkan oleh masyarakat Kota Pasuruan. Halaman ini disediakan fitur-fitur antara lain menghapus pengaduan, mengubah status pengaduan, merespon pengaduan dan mencetak laporan pengaduan.

#### 6.4.2.5 Halaman Laporan Pengaduan

PEMERINTAH KOTA PASURUAN  
**DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL**  
Jalan Pahlawan No.22 Telp. (0343) 421717 Fax. (0343) 416883  
PASURUAN 67126

---

**Laporan Aduan Masyarakat**  
Kode Pesan : PSN201803-001  
Tanggal Cetak : 29-03-2018 / 00:38:30 WIB

Yth. KABID. PELAYANAN PENCATATAN SIPIL  
Ditempat

Yang menyampaikan aduan masyarakat di bawah ini adalah :

ID Pesan : : PSN201803-001  
NIK : : 3514016011890001  
Nama Lengkap : USNAENI  
Tanggal Aduan : 29-MAR-18

Menyatakan dengan aduan sebagai Berikut :

KATEGORI PENGADUAN : PELAYANAN  
TOPIK PENGADUAN : KELUHAN  
ISI PENGADUAN : saya sudah melakukan perekaman tapi belum dipanggil untuk melakukan pencetakan KTP di dispendukcapil , mohon ditanggapi terima kasih

Pernyataan Respon dari Pengaduan :

Pasuruan, 29 March 2018  
KABID. PELAYANAN PENCATATAN SIPIL

**Gambar 6.15 Implementasi Antarmuka Laporan Pengaduan Berbasis Website**

Pada Gambar 6.15 adalah hasil implementasi dari perancangan antarmuka pada Gambar 5.35. Pada halaman hasil laporan salah satu pengaduan yang telah dikirimkan oleh Masyarakat Kota pasuruan dan telah tercetak dalam format *PDF* yang ditujukan kepada Petugas Administrasi. Dalam laporan tersebut akan disediakan *box* kotak untuk memberikan pernyataan mengenai pengaduan yang disampaikan dan selanjutnya dari pernyataan tersebut akan dikirimkan kepada pengirim pengaduan tersebut sebagai balasan atau respon dari pengaduan yang telah diproses.



## BAB 7 PENGUJIAN DAN ANALISIS HASIL PENGUJIAN

Bab ini membahas mengenai dilakukannya pengujian dan analisis hasil pengujian terhadap Sistem Layanan Pengaduan Masyarakat yang telah dibangun. Hal ini dilakukan untuk menemukan kesalahan yang terdapat pada sistem dan memastikan bahwa sistem yang dikembangkan telah sesuai. Pengujian pada penelitian ini dilakukan dengan menggunakan *validation testing* dan *user acceptance testing*. Setelah pengujian selesai maka akan dilanjutkan dengan menganalisa hasil pengujian.

### 7.1 Pengujian

Pada pengujian pertama menggunakan *validation testing* merupakan pengujian yang mengabaikan mekanisme internal sistem, dan berfokus pada menguji semua kebutuhan fungsional sistem dengan menggunakan kasus uji yang berbeda-beda berupa input dan kondisi eksekusi yang dipilih. Pengujian *validation testing* akan membandingkan hasil yang diharapkan apakah sama dengan hasil yang didapat yang akan ditulis pada kolom status validasi.

Pengujian kedua menggunakan *User Acceptance Testing* dilakukan dengan cara menyebarkan sejumlah *task* dan pertanyaan pada *stakeholder* terkait, seperti Petugas Administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dan Masyarakat Kota Pasuruan. Hasil dari jawaban para *stakeholder* akan dihitung dan dirata-rata untuk menghasilkan presentase bahwa sistem dapat diterima atau tidak diterima oleh para *stakeholder*.

#### 7.1.1 Validation Testing

*Validation testing* bertujuan untuk memastikan sistem yang dikembangkan dapat berjalan sesuai dengan kebutuhan fungsional yang diharapkan. Pengujian *validation testing* akan terbagi menjadi dua pengujian yaitu pengujian berbasis Android untuk Masyarakat Kota Pasuruan dan berbasis *Website* untuk Petugas Administrasi Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan pada sistem Layanan Pengaduan Masyarakat. Pengujian tidak dilakukan di seluruh fungsional, tetapi pengujian berbasis android 3 fungsional dan pengujian berbasis *website* 2 fungsional yang akan didokumentasikan dalam penelitian ini.

##### 7.1.1.1 Validation Testing berbasis Android

###### 7.1.1.1.1 Pengujian pada Fungsional Registrasi

Tabel 7.1 Pengujian Registrasi

<b>Nama Kasus Pengujian</b>	Registrasi
<b>Objek Pengujian</b>	SRS-F-EPKT-03
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem

**Tabel 7.1 Pengujian Registrasi (Lanjutan)**

	dapat menambah pengguna baru dan menampilkan notifikasi berhasil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Daftar</li> <li>2. Aktor memasukkan identitas diri pada formulir registrasi (Identitas pengguna ber-KTP Kota Pasuruan)</li> <li>3. Aktor menekan tombol Daftar</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menyimpan data identitas pengguna baru ke dalam <i>database</i> sistem dan menampilkan notifikasi berhasil
<b>Hasil Pengujian</b>	Sistem menyimpan data identitas pengguna baru ke dalam <i>database</i> sistem dan menampilkan notifikasi berhasil
<b>Status Validasi</b>	Valid

**Tabel 7.2 Pengujian Registrasi *Alternative Flows* : Data Registrasi salah**

<b>Nama Kasus Pengujian</b>	Registrasi
<b>Objek Pengujian</b>	SRS-F-EPKT-03
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan haru ber-KTP Kota Pasuruan jika data identitas diri yang dimasukkan bukan ber-KTP Kota Pasuruan pada formulir tambah pengguna
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Daftar</li> <li>2. Aktor memasukkan identitas diri pada formulir registrasi (Identitas pengguna bukan ber-KTP Kota Pasuruan)</li> <li>3. Aktor menekan tombol Daftar</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika pendaftar harus ber-KTP Kota Pasuruan
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa pendaftar harus ber-KTP Kota Pasuruan
<b>Status Validasi</b>	Valid



Tabel 7.3 Pengujian Registrasi *Alternative Flows* : Data Registrasi tidak lengkap

<b>Nama Kasus Pengujian</b>	Registrasi
<b>Objek Pengujian</b>	SRS-F-EPKT-03
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan jika data identitas diri yang dimasukkan pada formulir registrasi tidak lengkap
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Daftar</li> <li>2. Aktor memasukkan identitas diri pada formulir registrasi tidak secara lengkap</li> <li>3. Aktor menekan tombol Daftar</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika terdapat kolom formulir registrasi yang kosong harus diisi
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa terdapat kolom formulir registrasi yang kosong harus diisi
<b>Status Validasi</b>	Valid

7.1.1.1.2 Pengujian pada *Fungsional Melaporkan Pengaduan*

Tabel 7.4 Pengujian Melaporkan Pengaduan

<b>Nama Kasus Pengujian</b>	Melaporkan Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-08
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menyimpan laporan pengaduan yang dikirimkan dan menampilkan notifikasi berhasil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Pengaduan</li> <li>2. Aktor memasukan data pengaduan pada formulir pengaduan</li> <li>3. Aktor menekan tombol KIRIM</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menyimpan laporan pengaduan yang dikirimkan ke dalam <i>database</i> dan menampilkan notifikasi berhasil
<b>Hasil Pengujian</b>	Sistem menyimpan laporan pengaduan yang dikirimkan ke dalam <i>database</i> dan menampilkan notifikasi berhasil
<b>Status Validasi</b>	Valid

**Tabel 7.5 Pengujian Melaporkan Pengaduan *Alternative Flows* : Data laporan pengaduan tidak lengkap**

<b>Nama Kasus Pengujian</b>	Melaporkan Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-08
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan jika data pengaduan yang dimasukkan pada formulir pengaduan tidak lengkap
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Pengaduan</li> <li>2. Aktor memasukan data pengaduan pada formulir pengaduan tidak secara lengkap</li> <li>3. Aktor menekan tombol KIRIM</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika terdapat kolom formulir pengaduan yang kosong harus diisi
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa terdapat kolom formulir pengaduan yang kosong harus diisi
<b>Status Validasi</b>	Valid

#### 7.1.1.1.3 Pengujian pada *Fungsional Melihat Layanan*

**Tabel 7.6 Pengujian Melihat Layanan**

<b>Nama Kasus Pengujian</b>	Melihat Layanan
<b>Objek Pengujian</b>	SRS-F-EPKT-06
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan informasi layanan administrasi kependudukan dan pencatatan sipil pada sistem
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Layanan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan informasi layanan administrasi kependudukan dan pencatatan sipil
<b>Hasil Pengujian</b>	Sistem menampilkan informasi seluruh daftar pengguna sistem dan pencatatan sipil
<b>Status Validasi</b>	Valid

### 7.1.1.2 Validation Testing Berbasis Website

#### 7.1.1.2.1 Pengujian pada Fungsional Mengelola Data Pengguna

Tabel 7.7 Pengujian Mengelola Data Pengguna

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan informasi seluruh daftar pengguna yang terdaftar pada sistem
<b>Prosedur Pengujian</b>	1. Aktor memilih menu Data Pengguna
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan informasi seluruh daftar pengguna sistem
<b>Hasil Pengujian</b>	Sistem menampilkan informasi seluruh daftar pengguna sistem
<b>Status Validasi</b>	Valid

Tabel 7.8 Pengujian Mengelola Data Pengguna *Alternative Flows* : Data Diri Pengguna Salah

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan pendaftaran gagal jika data diri identitas pengguna yang dimasukkan bukan ber-KTP Kota Pasuruan pada formulir tambah pengguna
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol Tambah Pengguna</li> <li>2. Aktor memasukkan data diri identitas pengguna yang bukan KTP Kota Pasuruan pada formulir tambah pengguna</li> <li>3. Aktor menekan tombol Tambah</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika pendaftaran gagal
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa pendaftaran gagal
<b>Status Validasi</b>	Valid

**Tabel 7.9 Pengujian Mengelola Data Pengguna *Alternative Flows* : NIK / No. HP Pengguna Telah Terdaftar**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan pendaftaran gagal jika data identitas yang dimasukkan pada formulir tambah pengguna berupa NIK / No. HP telah terdaftar sebelumnya didalam sistem
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol Tambah Pengguna</li> <li>2. Aktor memasukkan data diri identitas berupa NIK atau No. HP yang telah terdaftar pada formulir tambah pengguna</li> <li>3. Aktor menekan tombol Tambah</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika pendaftaran gagal
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa pendaftaran gagal
<b>Status Validasi</b>	Valid

**Tabel 7.10 Pengujian Mengelola Data Pengguna *Alternative Flows* : Gagal Ubah *Password* Akun Pengguna**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan update gagal jika password lama yang dimasukkan pada formulir ubah <i>password</i> tidak cocok dengan <i>password</i> yang tersimpan
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Ubah <i>Password</i></li> <li>2. Aktor memasukkan data <i>password</i> lama tidak sesuai pada formulir ubah <i>password</i></li> <li>3. Aktor menekan tombol Simpan Perubahan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika update gagal
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa update gagal

**Tabel 7.10 Pengujian Mengelola Data Pengguna *Alternative Flows* : Gagal Ubah Password Akun Pengguna (Lanjutan)**

<b>Status Validasi</b>	Valid
------------------------	-------

**Tabel 7.11 Pengujian Mengelola Data Pengguna *Subflow* : Tambah Pengguna**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menambah pengguna baru
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol Tambah Pengguna</li> <li>2. Aktor memasukkan data identitas pengguna baru pada formulir tambah pengguna</li> <li>3. Aktor menekan tombol Tambah</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menyimpan data identitas pengguna baru ke dalam <i>datatabase</i> sistem
<b>Hasil Pengujian</b>	Sistem menyimpan data identitas pengguna baru ke dalam <i>database</i> sistem
<b>Status Validasi</b>	Valid

**Tabel 7.12 Pengujian Mengelola Data Pengguna *Subflow* : Hapus Pengguna**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menghapus pengguna sistem yang diinginkan dan menampilkan notifikasi berhasil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Hapus (Pengguna sistem yang ingin dihapus)</li> <li>2. Aktor mengkonfirmasi penghapusan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menghapus pengguna dan menampilkan notifikasi berhasil
<b>Hasil Pengujian</b>	Sistem menghapus pengguna dan menampilkan notifikasi berhasil
<b>Status Validasi</b>	Valid

**Tabel 7.13 Pengujian Mengelola Data Pengguna *Subflow* : Ubah Status Akun Pengguna**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat mengubah status akun pengguna sistem yang diinginkan dan menampilkan notifikasi berhasil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Ubah Status Akun(Pengguna sistem yang ingin statusnya diubah)</li> <li>2. Aktor mengkonfirmasi perubahan status akun pengguna</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat mengubah status akun pengguna dan menampilkan notifikasi berhasil
<b>Hasil Pengujian</b>	Sistem mengubah status akun pengguna dan menampilkan notifikasi berhasil
<b>Status Validasi</b>	Valid

**Tabel 7.14 Pengujian Mengelola Data Pengguna *Subflow* : Ubah *Password* Pengguna**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengguna
<b>Objek Pengujian</b>	SRS-F-EPKT-19
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat mengubah <i>password</i> pengguna sistem yang diinginkan dan menampilkan notifikasi berhasil
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Ubah <i>Password</i> (Pengguna sistem yang ingin <i>password</i>nya diubah)</li> <li>2. Aktor memasukkan data ubah <i>password</i> pada formulir ubah <i>password</i></li> <li>3. Aktor menekan tombol Simpan Perubahan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat mengubah <i>password</i> pengguna dan menampilkan notifikasi berhasil
<b>Hasil Pengujian</b>	Sistem mengubah <i>password</i> pengguna dan menampilkan notifikasi berhasil
<b>Status Validasi</b>	valid



## 7.1.1.2.2 Pengujian pada Fungsional Mengelola Data Pengaduan

Tabel 7.15 Pengujian Mengelola Data Pengaduan

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-14
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan informasi seluruh daftar pengaduan yang tersimpan pada sistem
<b>Prosedur Pengujian</b>	1. Aktor memilih menu Data Pengaduan
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan informasi seluruh daftar pengaduan
<b>Hasil Pengujian</b>	Sistem menampilkan informasi seluruh daftar pengaduan
<b>Status Validasi</b>	Valid

Tabel 7.16 Pengujian Mengelola Data Pengaduan *Alternative Flows* : Gagal Hapus Aduan

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-14
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menampilkan notifikasi peringatan gagal terhapus jika menghapus pengaduan yang masih berstatus Belum Diproses
<b>Prosedur Pengujian</b>	1. Aktor memilih icon Hapus Aduan (Pengaduan yang ingin dihapus masih status Belum Diproses) 2. Aktor mengkonfirmasi penghapusan
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika gagal terhapus
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa gagal terhapus
<b>Status Validasi</b>	Valid

Tabel 7.17 Pengujian Mengelola Data Pengaduan *Subflow* : Balas Aduan

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-14

<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat meng- <i>update</i> respon balasan pengaduan yang diinginkan dan menampilkan notifikasi berhasil terupdate
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Balas Aduan (Pengaduan yang ingin di <i>update</i> respon pesannya)</li> <li>2. Aktor mengisi pesan pada formulir balas pesan pengaduan</li> <li>3. Aktor menekan tombol Simpan Perubahan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat meng- <i>update</i> pesan balas pengaduan ke dalam <i>database</i> sistem dan menampilkan notifikasi berhasil terupdate
<b>Hasil Pengujian</b>	Sistem meng- <i>update</i> pesan balas pengaduan ke dalam <i>database</i> sistem dan menampilkan notifikasi berhasil terupdate
<b>Status Validasi</b>	Valid

**Tabel 7.18 Pengujian Mengelola Data Pengaduan *Subflow* : Hapus Aduan**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-14
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat menghapus pengaduan yang diinginkan dan menampilkan notifikasi berhasil terhapus
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Hapus Aduan (Pengaduan yang ingin dihapus dengan status pengaduan Sudah Diproses)</li> <li>2. Aktor mengkonfirmasi penghapusan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat menampilkan notifikasi peringatan jika berhasil terhapus
<b>Hasil Pengujian</b>	Sistem menampilkan notifikasi peringatan bahwa berhasil terhapus
<b>Status Validasi</b>	Valid

**Tabel 7.19 Pengujian Mengelola Data Pengaduan *Subflow* : Ubah Status Aduan**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-14
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem

**Tabel 7.19 Pengujian Mengelola Data Pengaduan *Subflows* : Ubah Status Aduan (Lanjutan)**

	dapat mengubah status pengaduan yang diinginkan dan menampilkan notifikasi berhasil terupdate
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Ubah Status Aduan (Pengaduan yang ingin statusnya diubah)</li> <li>2. Aktor mengkonfirmasi perubahan status pengaduan</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat mengubah status pengaduan dan menampilkan notifikasi berhasil terupdate
<b>Hasil Pengujian</b>	Sistem mengubah status pengaduan dan menampilkan notifikasi berhasil terupdate
<b>Status Validasi</b>	Valid

**Tabel 7.20 Pengujian Mengelola Data Pengaduan *Subflow* : Cetak Aduan**

<b>Nama Kasus Pengujian</b>	Mengelola Data Pengaduan
<b>Objek Pengujian</b>	SRS-F-EPKT-14
<b>Tujuan Pengujian</b>	Pengujian dilakukan untuk memastikan sistem dapat membuka tab halaman baru pada <i>browser</i> dengan menampilkan laporan pengaduan berformat <i>pdf</i>
<b>Prosedur Pengujian</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih icon Cetak PDF</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem dapat membuka tab halaman baru dengan menampilkan laporan pengaduan berformat <i>pdf</i>
<b>Hasil Pengujian</b>	Sistem membuka tab halaman baru dengan menampilkan laporan pengaduan berformat <i>pdf</i>
<b>Status Validasi</b>	Valid

### 7.1.2 User Acceptance Testing

*User Acceptance Testing* (UAT) bertujuan untuk mengetahui apakah aplikasi yang dikembangkan sesuai dengan harapan pemangku kepentingan (*Stakeholders*). Pengujian ini dilakukan dengan tidak memperhatikan kodingan atau program didalam sistem, namun memperhatikan apakah Aplikasi Layanan Pengaduan Masyarakat ini dapat diterima oleh *stakeholder*.

Pada *user acceptance testing* ini akan melibatkan beberapa *stakeholders* dalam proses pengujiannya, yaitu Petugas Administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dan Masyarakat Kota Pasuruan. *Stakeholders* disini berperan sebagai tester (penguji). Tabel 7.21 akan dijelaskan mengenai *stakeholders*.

**Tabel 7.21 Stakeholder Penguji User Acceptance Testing**

No	Stakeholders	Alasan	Persyaratan
1	Petugas Administrasi	<ol style="list-style-type: none"> <li>1. Agar mendapatkan kemudahan dalam pengelolaan laporan pengaduan</li> <li>2. Agar dapat mengoperasikan sistem yang akan digunakan</li> </ol>	Mengetahui cara menggunakan komputer
2	Masyarakat Kota Pasuruan	Agar mendapatkan kemudahan dalam penyampaian laporan pengaduan secara online	<ol style="list-style-type: none"> <li>1. Mengetahui cara menggunakan <i>smarthphone</i></li> <li>2. Pernah melakukan pelaporan pengaduan</li> </ol>

Pelaksanaan pengujian *user acceptance* dilakukan dengan mendatangi satu per satu *tester* (penguji) untuk dimintai kesediaan menggunakan Sistem Layanan Pengaduan Masyarakat. Jumlah *tester* pada pengujian *user acceptance* digolongkan menjadi 2 kelompok yaitu berbasis *website* ditujukan kepada petugas administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan 2 orang dan berbasis Android ditujukan kepada Masyarakat Kota Pasuruan 10 orang (sesuai 10% dari total rata-rata perbulan laporan pengaduan yang diterima sebanyak 100 laporan pengaduan).

Pertama-tama *tester* diberikan penjelasan tentang Aplikasi Layanan Pengaduan Masyarakat beserta fungsionalitasnya. Selanjutnya *tester* diberikan lembar skenario untuk mencoba fungsi yang tersedia. *Tester* harus mengikuti setiap langkah yang sudah dituliskan dalam lembar skenario. Lembar skenario akan dibagi menjadi 2 lembar, yaitu skenario yang harus dikerjakan oleh Petugas Administrasi ditunjukkan pada Tabel 7.22, sedangkan skenario yang harus dikerjakan oleh Masyarakat Kota Pasuruan ditunjukkan pada Tabel 7.23.

**Tabel 7.22 Skenario yang harus dikerjakan Petugas Administrasi**

No.	Tugas yang dikerjakan
1	Berada di halaman <i>login</i>
2	Masukkan username dan password pada halaman <i>login</i> klik tombol <i>login</i>

**Tabel 7.22 Skenario yang harus dikerjakan Petugas Administrasi (Lanjutan)**

3	Berada di halaman beranda
4	Klik menu Data Pengguna
5	Masuk pada halaman data pengguna tekan tombol Tambah Pengguna untuk menambah daftar pengguna melalui <i>website</i>
6	Memasukkan identitas pengguna baru dan klik simpan atau kembali untuk membatalkan
7	Kembali pada halaman data pengguna, klik icon hapus untuk menghapus identitas pengguna dari sistem
8	Kembali pada halaman data pengguna, klik menu Data Pengaduan
9	Masuk pada halaman data pengaduan klik icon pdf untuk mencetak laporan pengaduan
10	Kembali pada halaman data pengguna, klik icon balas aduan untuk merespon laporan pengaduan
11	Kembali pada halaman data pengguna, klik icon ubah status untuk mengubah status pengaduan dari belum proses menjadi sudah proses atau sebaliknya
12	Kembali pada halaman data pengguna, klik icon hapus aduan untuk menghapus pengaduan
13	Klik tombol logout untuk keluar

**Tabel 7.23 Skenario yang harus dikerjakan Masyarakat Kota Pasuruan**

No.	Tugas yang dikerjakan
1	Berada di halaman <i>login</i>
2	klik tombol Daftar
3	Masuk pada halaman registrasi untuk melakukan pendaftaran bagi pengguna baru
4	Memasukkan identitas pengguna baru dan klik daftar atau kembali untuk membatalkan
5	Masukkan no_hp dan password pada halaman <i>login</i> klik tombol <i>login</i>
6	Berada di halaman utama
7	Tekan menu Layanan
8	Masuk halaman layanan, baca informasi layanan
9	Tekan tombol menu pada pojok kiri atas, tekan menu tentang kami
10	Masuk halaman tentang kami, baca informasi singkat tentang Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan
11	Tekan tombol menu pada pojok kiri atas, tekan menu daftar pengaduan

**Tabel 7.23 Skenario yang harus dikerjakan Masyarakat Kota Pasuruan (Lanjutan)**

12	Masuk halaman daftar pengaduan, tekan tombol status pengaduan untuk melihat respon pengaduan
13	Tekan tombol menu pada pojok kiri atas, tekan menu profil
14	Masuk pada halaman profil, tekan tombol ubah password untuk mengubah password lama
15	Mengubah password dan klik simpan atau kembali untuk membatalkan
16	Tekan tombol menu pada pojok kiri atas, tekan menu pengaduan
17	Masuk halaman pengaduan
18	Masukkan data laporan pengaduan yang diminta dan klik kirim atau kembali untuk membatalkan laporan
19	Klik tombol logout untuk keluar

Daftar pertanyaan akan dibagi menjadi 2 jenis, yaitu untuk petugas administrasi pertanyaan yang ditunjukkan sesuai pada Tabel 7.24 dan untuk Masyarakat Kota Pasuruan pertanyaan yang ditunjukkan sesuai pada Tabel 7.25. Seluruh pertanyaan pengujian UAT yang di ajukan menggunakan penilaian skala *likert* dari skala nilai 1 sampai 5. Skor penilaian pada setiap jawaban yang di berikan dengan menggunakan skala *likert* yang ditunjukkan pada Tabel 7.26.

**Tabel 7.24 Tabel Pertanyaan UAT untuk Petugas Administrasi**

No.	Pertanyaan	Penilaian				
		STS	TS	N	S	SS
1	Apakah fitur mengelola data pengguna yang terdaftar pada Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?					
2	Apakah fitur mengelola data pengaduan Masyarakat Kota Pasuruan pada Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?					
3	Apakah proses pencetakan pelaporan pengaduan melalui Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?					
4	Apakah Aplikasi Layanan Pengaduan Masyarakat ini berjalan sesuai dengan kebutuhan ?					
5	Apakah Aplikasi Layanan Pengaduan Masyarakat ini mudah dipahami ?					

**Tabel 7.25 Tabel Pertanyaan UAT untuk Masyarakat Kota Pasuruan**

No.	Pertanyaan	Penilaian				
		STS	TS	N	S	SS
1	Apakah Aplikasi Layanan Pengaduan Masyarakat ini mudah dipahami ?					
2	Apakah dengan adanya fitur pengaduan pada Aplikasi Layanan Pengaduan Masyarakat mengenai pelaporan pengaduan yang ingin disampaikan menjadi lebih mudah?					
3	Apakah proses pelaporan pengaduan melalui Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?					
4	Apakah dengan adanya fitur layanan pada Aplikasi Layanan Pengaduan Masyarakat informasi yang diberikan mudah dimengerti ?					
5	Apakah dengan adanya fitur daftar pengaduan pada Aplikasi Layanan Pengaduan Masyarakat membantu dalam memonitoring pengaduan yang dikirimkan berupa status dan respon dari pengaduan yang telah dikirimkan ?					

**Tabel 7.26 Tabel Skor Penilaian Jawaban Pertanyaan Pengujian UAT**

Skala Jawaban	Keterangan	Skor
STS	Sangat Tidak Setuju	1
TS	Tidak Setuju	2
N	Netral	3
S	Setuju	4
SS	Sangat Setuju	5

### 7.1.2.1 Hasil Pengujian User Acceptance Testing

Setelah mendapatkan hasil jawaban kuisisioner yang telah dibagikan kepada Petugas Administrasi berjumlah 2 responden di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan dan Masyarakat Kota Pasuruan berjumlah 10

responden. Kemudian berdasarkan data hasil kuisioner tersebut, dapat dicari presentase masing – masing jawaban dengan rumus :

$$\text{Skor} = (\text{STS} \times 1) + (\text{TS} \times 2) + (\text{N} \times 3) + (\text{S} \times 4) + (\text{SS} \times 5)$$

$$\text{Index (\%)} = (\text{Total Skor} / \text{Skor Ideal} \times 100)$$

Keterangan :

- STS : Penilaian Sangat Tidak Setuju
- TS : Penilaian Tidak Setuju
- N : Penilaian Netral
- S : Penilaian Setuju
- SS : Penilaian Sangat Setuju
- Total Skor : jumlah seluruh skor
- Skor Ideal : Nilai tertinggi dikalikan dengan jumlah sampel / responden

Setelah diperoleh index dari setiap jawaban, maka interpretasi atau skor dapat dilihat pada Tabel 7.27 yang digunakan sebagai acuan interpretasi skor yang didapatkan pada saat pengujian *user acceptance testing*.

**Tabel 7.27 Interpretasi Skor Likert**

Sumber: Bertram (2007)

Inpertasi Skor Dengan interval = 20	Keterangan
0%-19%	Sangat Tidak Memuaskan
20%-39%	Tidak Memuaskan
40%-59%	Biasa
60%-79%	Memuaskan
80%-100%	Sangat Memuaskan

Berikut ini adalah hasil perhitungan dari masing - masing jawaban kuisioner yang telah dibagi menjadi 2 tabel. Dapat dilihat pada Tabel 7.28 merupakan hasil dari pengujian *user acceptance testing* untuk Petugas Administrasi dan Tabel 7.29 merupakan hasil dari pengujian *user acceptance testing* untuk Masyarakat Kota Pasuruan.

**Tabel 7.28 Hasil Pengujian UAT untuk Petugas Administrasi**

No.	Pertanyaan	Penilaian					Skor
		1	2	3	4	5	
1	Apakah fitur mengelola data pengguna yang terdaftar					10	10





**Tabel 7.28 Hasil Pengujian UAT untuk Petugas Administrasi (Lanjutan)**

	pada Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?						
2	Apakah fitur mengelola data pengaduan Masyarakat Kota Pasuruan pada Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?				4	5	9
3	Apakah proses pencetakan pelaporan pengaduan melalui Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?				4	5	9
4	Apakah Aplikasi Layanan Pengaduan Masyarakat ini berjalan sesuai dengan kebutuhan ?					10	10
5	Apakah Aplikasi Layanan Pengaduan Masyarakat ini mudah dipahami ?				4	5	9
Total Skor							47
Rata – Rata Index (%)							94%

**Tabel 7.29 Hasil Pengujian UAT untuk Masyarakat Kota Pasuruan**

No.	Pertanyaan	Penilaian					Skor
		1	2	3	4	5	
1	Apakah Aplikasi Layanan Pengaduan Masyarakat ini mudah dipahami ?				28	15	43
2	Apakah dengan adanya fitur pengaduan pada Sistem Layanan Pengaduan Masyarakat mengenai pelaporan pengaduan yang ingin disampaikan menjadi lebih mudah?				12	35	47

Tabel 7.29 Hasil Pengujian UAT untuk Masyarakat Kota Pasuruan (Lanjutan)

3	Apakah proses pelaporan pengaduan melalui Aplikasi Layanan Pengaduan Masyarakat berjalan dengan baik ?			6	24	10	40
4	Apakah dengan adanya fitur layanan pada Aplikasi Layanan Pengaduan Masyarakat informasi yang diberikan mudah dimengerti ?			12	24		36
5	Apakah dengan adanya fitur daftar pengaduan pada Sistem Layanan Pengaduan membantu dalam memonitoring pengaduan yang dikirimkan berupa status dan respon dari pengaduan yang telah dikirimkan ?			6	24	10	40
Total Skor							206
Rata – Rata Index (%)							82%

## 7.2 Analisis Hasil Pengujian

Berdasarkan hasil pengujian dari Aplikasi Layanan Pengaduan Masyarakat dengan pengujian *validation testing* dan *user acceptance testing* yang telah dilakukan, maka didapatkan analisis hasil pengujian sebagai berikut:

1. Pengujian dengan menggunakan *validation testing* yang dilakukan dengan melihat kesesuaian antara kebutuhan fungsional sistem dengan kesesuaian kinerja sistem saat dijalankan. Sehingga berdasarkan hasil pengujian *validation testing* pada Tabel 7.1 sampai dengan Tabel 7.20 maka dapat disimpulkan bahwa semua kebutuhan fungsional Sistem Layanan Pengaduan Masyarakat Dinas Kependudukan dan Pencatatan Sipil kota Pasuruan yang telah diuji seluruhnya memiliki status valid atau sistem ini berjalan sesuai dengan kebutuhan fungsionalnya.
2. Pengujian dengan menggunakan *user acceptance testing* berdasarkan jawaban responden atau penguji yang terdiri dari Petugas Administrasi dan Masyarakat Kota Pasuruan terhadap seluruh pertanyaan yang diajukan. Maka dapat diketahui nilai presentase berdasarkan interpretasi Skor *Likert* yang

ditunjukkan pada Tabel 7.27. Dari hasil pengujian aplikasi Layanan Pengaduan Masyarakat berbasis *Website* yang ditujukan kepada Petugas Administrasi yang dapat dilihat pada Tabel 7.28 didapatkan tingkat presentase kepuasan 94% dengan status “Sangat Memuaskan” yang menandakan bahwa Aplikasi Layanan Pengaduan Masyarakat berbasis *Website* dapat diterima oleh Petugas Administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan, sedangkan pengujian aplikasi Layanan Pengaduan Masyarakat berbasis Android yang ditujukan kepada Masyarakat Kota Pasuruan yang dapat dilihat pada Tabel 7.29 didapatkan tingkat presentase kepuasan 82% dengan status “Sangat Memuaskan” yang menandakan bahwa Aplikasi Layanan Pengaduan Masyarakat berbasis Android dapat diterima oleh Masyarakat Kota Pasuruan.



## BAB 8 PENUTUP

### 8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat diambil kesimpulan dari hasil Pengembangan Aplikasi Layanan Pengaduan Masyarakat sebagai berikut:

1. Perancangan telah dilakukan berdasarkan hasil analisis kebutuhan. Dari analisis kebutuhan didapatkan informasi perwakilan pemangku kepentingan yang akan menggunakan sistem, analisis persyaratan fungsional, analisis persyaratan non-fungsional, pemodelan *use case* beserta penjelasan beberapa fungsi seperti terdapat 7 buah *description use case* dan 7 buah *activity diagram*. Sedangkan dari perancangan sistem menghasilkan 7 buah *sequence diagram*, *class diagram*, rancangan komunikasi data *webservice*, *relational model* dan rancangan antarmuka sistem.
2. Hasil dari perancangan sistem kemudian diimplementasikan kedalam bentuk sistem berbasis Android yang digunakan untuk Masyarakat Kota Pasuruan dan sistem berbasis *website* digunakan untuk Petugas Administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan. Sistem berbasis Android dikembangkan menggunakan bahasa pemrograman *Java*, dengan memanfaatkan *webservice* sebagai penghubung dengan *database*, sementara sistem berbasis *website* dikembangkan dengan bahasa pemrograman PHP dan Oracle sebagai *database*.
3. Proses pengujian dilakukan dengan menggunakan *validation testing* dan *user acceptance testing* (UAT). Berdasarkan hasil *validation testing*, fungsi atau fitur-fitur yang ada didalam sistem layanan pengaduan masyarakat berjalan sesuai harapan. Sementara itu, hasil dari *user acceptance testing*, diambil kesimpulan bahwa pengguna menerima sistem dan mampu membantu mempermudah laporan pengaduan bagi Masyarakat Kota Pasuruan maupun mempermudah pengelolaan pengaduan oleh petugas administrasi di Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.

### 8.2 Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian pada skripsi ini yaitu:

1. Sistem berbasis Android yang ditujukan Masyarakat Kota Pasuruan dapat dikembangkan dengan fitur *sms gateway* untuk mengirimkan kode verifikasi sebagai bukti bahwa nomor handphone yang didaftarkan oleh pengguna benar nomor milik sendiri dan masih aktif.
2. Untuk pengembangan selanjutnya, perlu dilakukan evaluasi sistem layanan pengaduan masyarakat dengan pengguna untuk mengetahui sejauh mana

sistem yang dikembangkan dapat memecahkan masalah pada Dinas Kependudukan dan Pencatatan Sipil Kota Pasuruan.



## DAFTAR PUSTAKA

- Agarwal, B. B., Tayal, S. P. & Gupta, M., 2010. *Software Engineering & Testing*. 1<sup>st</sup> ed. Massachusetts: Jones and Bartlett
- Anofrizen. 2017. *Sistem Informasi Pengaduan Masyarakat Program Keluarga Harapan Kota Pekanbaru*
- Ambler, Scoot W. 2004. *The Object Primer 3<sup>rd</sup> Edition: Agile Model Driven Development with UML 2*. Cambridge, UK
- Anonim. 2016. *Pengenalan JSON*. Tersedia di : <<http://www.json.org/json-id.html>> [Diakses tanggal 14 Februari 2018]
- Basuki, Heru. 2006. *Penelitian Kualitatif untuk Ilmu - Ilmu Kemanusiaan dan Budaya*. Jakarta
- Bertram, Dane. 2007. *Likert Scales*. Tersedia di : <<http://poincare.matf.bg.ac.rs/~kristina/topic-dane-likert.pdf>> [Diakses tanggal 19 Mei 2018]
- Deitel, Paul, Harvey Deitel. 2012. *Java for Programmers Second Edition*. Boston: Pearson Education
- Dennis, Wixom dan Roth., 2009. *System Analysis and Design, Fourth Edition*, John Wiley & Sons, Inc, Hoboken
- Gorton, Brous. 2005. *Health Service Review Council: Guide to Complaint Handling in Health Care Services*. Australia
- Hambling, B. & Goethem, P. V., 2013. *User Acceptance Testing: A Step-by-step Guide*. s.l.: BCS Learning & Development Limited
- Indriati. 2010. *Pengujian Validasi*. Tersedia di : <<http://indryz.lecture.ub.ac.id/files/2010/11/Pengujian-Validasi.docx>> [Diakses tanggal 14 Februari 2018]
- Jogiyanto, H. M., 1995. *Analisis & Disain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset
- Jumardi A., Solichin A., 2016. *Prototipe Aplikasi Layanan Pengaduan Masyarakat Berbasis Android dan Web Service*
- Kalali, Masoud, Bhakti Mehta. 2013. *Developing Restful Service with JAX-RS 2.0, WebSockets, and JSON*. Birmingham: Packt Publishing
- Object Management Group, 2011. *OMG Object Management Group*. Tersedia di : <<http://www.omg.org/spec/BPMN/2.0/PDF/>> [Diakses tanggal 14 Februari 2018]
- Rahmayanty, Nina. 2010. *Manajemen Pelayanan Prima*. Yogyakarta: Graha Ilmu

- Rainer, R.K. & Turban, E., 2009. *Introduction To Information Systems*. 2<sup>nd</sup> ed. New Jersey: John Wiley & Sons
- Gay, L.R., Diehl, P.L 1992. *Research Methods for Business Management*. New York: MacMilan Publishing Company
- S. & Shalahuddin, 2015. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Penerbit Informatika
- Stephanus, Hermawan, 2011. *Mudah Membuat Aplikasi Android*. Yogyakarta: Andi Offset
- Sarker H.I.,Faruque F., Rahman A.,2015. *A Survey of Software Development Process Models in Software Engineering*
- Satzinger, J. W., et al. 2010. *System Analysis and Design in a Changing World*. Boston: Course Technology
- Softwaretestinghelp, 2017. *what-is-user-acceptance-testing-uat*. Tersedia di <<http://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/>> [Diakses tanggal 18 Februari 2018]
- Sommerville, Ian. 2011. *Software Engineering, Ninth Edition*. New York: Addison-Wesley.
- Sugiyono, 2016. *Metode Penelitian Pendekatan Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta
- Sutopo dan Adi Suryanto. 2003. *Pelayanan Prima*. Jakarta: Lembaga Administrasi Negara
- Wicaksono, B. 2013. *Sistem Operasi Android Versi 4.0 (ICS: Ice Cream Sandwich) dan Versi 4.1 (Jelly Bean)*