

**PEMBANGUNAN SISTEM APLIKASI PELAYANAN  
ADMINISTRASI DESA BERBASIS WEB DENGAN  
FITUR NOTIFIKASI SMS GATEWAY  
(STUDI KASUS DESA PELANGWOT)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Anton Firdaus  
NIM: 145150207111083



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

# PENGESAHAN

PEMBANGUNAN SISTEM APLIKASI PELAYANAN  
ADMINISTRASI DESA BERBASIS WEB DENGAN  
FITUR NOTIFIKASI SMS GATEWAY  
(STUDI KASUS DESA PELANGWOT)

## SKRIPSI

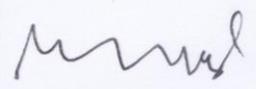
Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Anton Firdaus  
NIM: 145150207111083

Skripsi ini telah diuji dan dinyatakan lulus pada  
26 Oktober 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

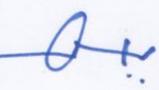
Dosen Pembimbing II



Nurudin Santoso, S.T., M.T  
NIP: 197409162000121001

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

Mengetahui  
Dekan Fakultas Teknik Informatika



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



## PERNYATAAN ORISINALITAS

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

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

Malang, 20 November 2018



Anton Firdaus

NIM: 145150207111083

## KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “PEMBANGUNAN SISTEM APLIKASI PELAYANAN ADMINISTRASI BERBASIS WEB DENGAN FITUR NOTIFIKASI SMS GATEWAY (STUDI KASUS DESA PELANGWOT)”.

Penulis menyadari bahwa dalam proses penyusunan skripsi ini tidak lepas dari adanya kerja sama dan bantuan dari berbagai pihak, sehingga pada kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak – pihak yang turut membantu selama penyusunan skripsi, diantaranya:

1. Bapak Nurudin Santoso, S.T., M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
2. Ibu Faizatul Amalia, S.Pd., M.Pd selaku dosen pembimbing II yang juga telah dengan teliti memberikan saran, perbaikan dan koreksi dalam proses penyelesaian skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T, Ph.D., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Drs. Mardji, M.T, dan Bapak Edy Santoso, S.Si., M.Kom. selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat.
6. Kedua orang tua penulis, Bapak Jupri dan Ibu Siti Khoiroh beserta keluarga besar Surahman yang selalu memberikan dukungan secara moril dan materiil.
7. Wahyu Wisnu, Galang Perdana, Juwanda, Shinta, dan Aulia. Sahabat yang telah menyertai jejak selama meniti ilmu di Malang.
8. Tafarrara, Ditya, Yudha, Clara, Fendra, Haida, Sastra, Yoga, dan Andriko. Manusia-Manusia yang selalu ada pada kilometer terakhir perjuangan skripsi ini.
9. Teman-teman Kopma Squad, Informatika, dan berbagai pihak pendukung yang namanya tidak dapat saya tuliskan satu-persatu.

Malang, 15 November 2018

Penulis

Antonfirdaus64@gmail.com

## ABSTRAK

**Anton Firdaus, Pembangunan Sistem Aplikasi Pelayanan Administrasi Desa Berbasis Web Dengan Fitur Notifikasi Sms Gateway (Studi Kasus Desa Pelangwot)**

**Pembimbing: Nurudin Santoso, S.T., M.T. dan Faizatul Amalia, S.Pd.,M.Pd.**

Pelayanan publik merupakan tugas pokok pemerintah dalam pengabdianya terhadap masyarakat. Tidak terkecuali pemerintahan desa. Sebagai tolok ukur keberhasilan, pelayanan administrasi dan persuratan sudah seharusnya dapat dilakukan dengan baik dan cepat agar dapat memberikan nilai kepuasan terhadap masyarakat. Akan tetapi pada penerapannya tentunya terdapat beberapa kondisi yang dapat menjadi kendala sehingga pelayanan tidak dapat diterapkan secara maksimal. Desa Pelangwot sendiri merupakan salah satu desa di kabupaten Lamongan yang memiliki luas sebesar 6.767,824 km<sup>2</sup>, jarak antara dusun dan kantor desa yang dapat mencapai 3,5 Km dengan kondisi jalan yang kurang baik tentunya menjadi faktor yang menyebabkan kendala bagi masyarakat untuk dapat mengurus kebutuhan administrasi di kantor desa. Melihat masalah tersebut tentunya dibutuhkan adanya media yang dapat menjembatani masyarakat dalam mencari sumber informasi desa sekaligus pengurusan administrasi. Dalam hal ini penulis mengembangkan media berbasis web yang dapat digunakan untuk melakukan pemesanan surat, dengan fitur sms gateway untuk memberikan informasi kepada masyarakat mengenai status surat yang mereka pesan. Proses pengembangan sistem dilakukan dengan menggunakan waterfall model yang menghasilkan empat jenis pengguna, delapan puluh lima kebutuhan fungsional, dan satu kebutuhan non fungsional. Pengujian dilakukan pada level unit dan validasi, serta terdapat pula pengujian compatibility dengan hasil pengujian 100% valid.

Kata kunci: administrasi, informasi, sms gateway

## ABSTRACT

**Anton Firdaus, Pembangunan Sistem Aplikasi Pelayanan Administrasi Desa Berbasis Web Dengan Fitur Notifikasi Sms Gateway (Studi Kasus Desa Pelangwot)**

**Pembimbing: Nurudin Santoso, S.T., M.T. dan Faizatul Amalia, S.Pd.,M.Pd.**

*The public service is the principal task of the Government in his devotion to the community. The village government is no exception. As success yardstick, service administration and correspondence was supposed to be done well and quickly in order to deliver satisfaction value to society. But on its application of course there are some conditions that can become a constraint so that the Ministry cannot be applied to the maximum. The village itself is one of the Pelangwot village in Lamongan which has an area of 6,767.824 km<sup>2</sup>, the distance between the village and the village office which can reach 3.5 km with the road conditions become unfavorable factors that lead to constraints for the community to be able to take care of the needs of administration in the Office of the village. See the problem is of course required the existence of a media that can bridge the communities in search of information sources at the same time maintaining the village administration. In this case the author developed a web-based media that can be used to make a booking letter, with the features of sms gateway to provide information to the public about the status of their mail messages. The system development process is done by using the waterfall model that produces four types of users, eighty-five functional needs, and one of the non-functional requirements. Testing is done at the level of the unit and validation, as well as there are also testing compatibility with 100% valid test results.*

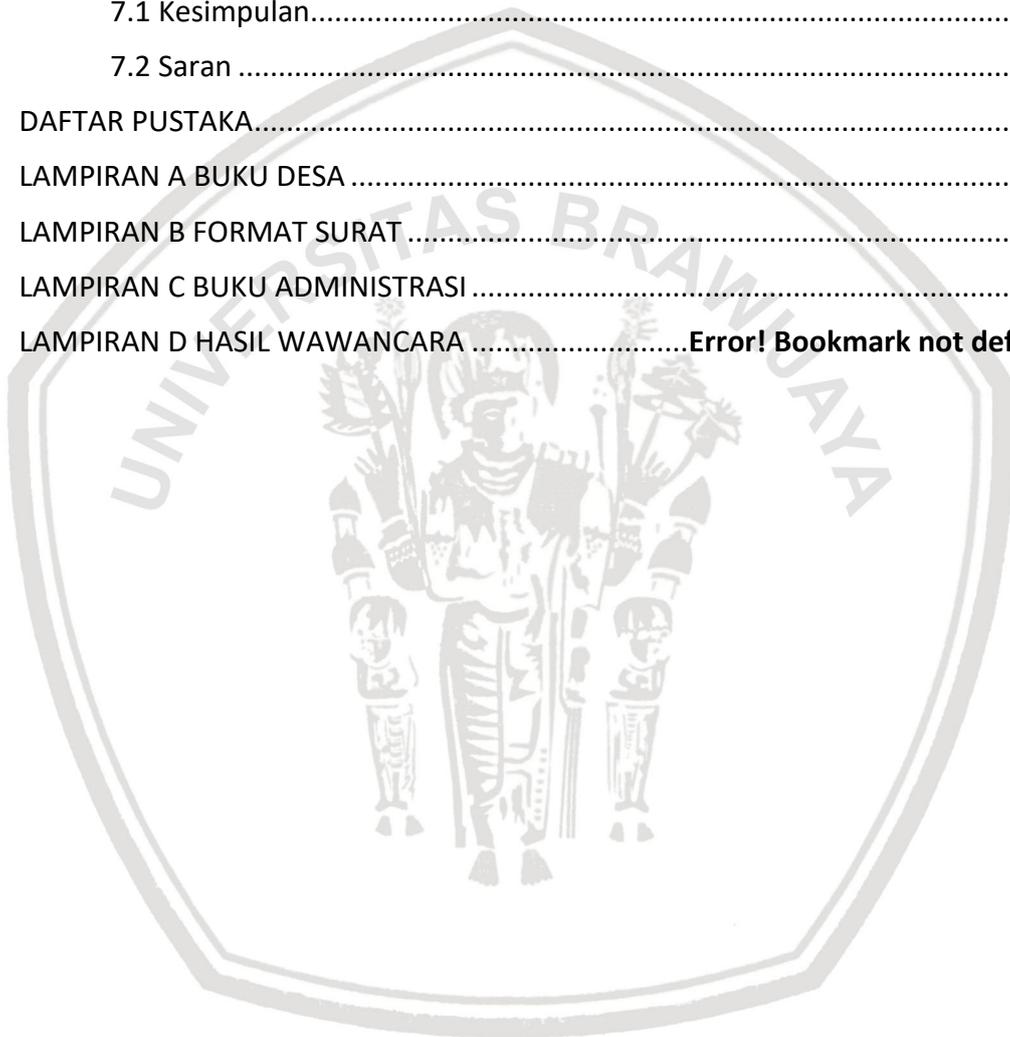
*Keywords: Administration, information, sms gateway.*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN .....	xviii
BAB 1 PENDAHULUAN.....	19
1.1 Latar belakang.....	19
1.2 Rumusan masalah.....	19
1.3 Tujuan .....	19
1.4 Manfaat.....	20
1.5 Batasan masalah .....	20
1.6 Sistematika pembahasan .....	20
BAB 2 LANDASAN KEPUSTAKAAN .....	22
2.1 Kajian Pustaka .....	22
2.2 <i>Software Application</i> .....	22
2.3 Administrasi Desa .....	22
2.4 <i>SMS Gateway</i> .....	23
2.5 Rekayasa Perangkat Lunak .....	25
2.5.1 Model <i>Waterfall</i> .....	25
2.6 Pemodelan sistem.....	26
2.6.1 <i>Use case Diagram</i> .....	26
2.6.2 <i>Sequence Diagram</i> .....	26
2.6.3 <i>Class Diagram</i> .....	27
2.7 Pengujian Perangkat Lunak.....	28
2.7.1 Pengujian Unit.....	28
2.7.2 Pengujian Validasi .....	29
2.7.3 Pengujian <i>Compatibility</i> .....	29

BAB 3 METODOLOGI .....	30
3.1 Tipe Penelitian .....	30
3.2 Strategi dan Rancangan Penelitian .....	30
3.2.1 Studi Literatur .....	31
3.2.2 Analisis Kebutuhan .....	31
3.2.3 Pengumpulan Data .....	31
3.2.4 Perancangan Sistem .....	31
3.2.5 Implementasi .....	32
3.2.6 Pengujian .....	32
3.2.7 Penarikan Kesimpulan .....	32
BAB 4 REKAYASA KEBUTUHAN .....	33
4.1 Deskripsi Sistem .....	33
4.2 Identifikasi Pengguna .....	33
4.3 Identifikasi Kebutuhan .....	34
4.3.1 Kebutuhan Fungsional .....	34
4.3.2 Kebutuhan Non Fungsional .....	40
4.4 Pemodelan <i>Use case</i> .....	40
4.4.1 Diagram <i>use case</i> .....	41
4.4.2 Skenario <i>use case</i> .....	42
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	74
5.1 Perancangan .....	74
5.1.1 <i>Sequence Diagram</i> .....	74
5.1.2 Perancangan Basis Data .....	76
5.1.1 <i>Class Diagram</i> .....	78
5.1.2 Perancangan Algoritme .....	79
5.1.3 Perancangan Antarmuka .....	80
5.2 Implementasi .....	84
5.2.1 Spesifikasi Sistem .....	84
5.2.2 Implementasi Basis Data .....	84
5.2.3 Implementasi Algoritme .....	85
5.2.4 Implementasi Antarmuka .....	87
BAB 6 PENGUJIAN .....	90

6.1 Pengujian Unit.....	90
6.1.1 Pengujian unit memesan surat .....	90
6.1.2 Pengujian unit edit BPD .....	92
6.1.3 Pengujian unit tambah APBD .....	96
6.2 Pengujian Validasi .....	98
6.3 Pengujian <i>Compatibility</i> .....	124
BAB 7 PENUTUP .....	126
7.1 Kesimpulan.....	126
7.2 Saran .....	126
DAFTAR PUSTAKA.....	127
LAMPIRAN A BUKU DESA .....	128
LAMPIRAN B FORMAT SURAT .....	135
LAMPIRAN C BUKU ADMINISTRASI .....	146
LAMPIRAN D HASIL WAWANCARA .....	<b>Error! Bookmark not defined.</b>



## DAFTAR TABEL

Tabel 4.1 Identifikasi pengguna (Lanjutan).....	33
Tabel 4.2 Kebutuhan Fungsional.....	34
Tabel 4.3 Kebutuhan non fungsional .....	40
Tabel 4.4 Skenario <i>use case register</i> .....	42
Tabel 4.5 Skenario <i>use case login</i> .....	42
Tabel 4.6 Skenario <i>use case logout</i> .....	43
Tabel 4.7 Skenario <i>use case</i> melihat profil pengguna.....	43
Tabel 4.8 Skenario <i>use case</i> mengubah profil pengguna.....	44
Tabel 4.9 Skenario <i>use case</i> melihat daftar artikel .....	44
Tabel 4.10 Skenario <i>use case</i> melihat isi artikel.....	45
Tabel 4.11 Skenario <i>use case</i> menambah artikel.....	45
Tabel 4.12 Skenario <i>use case</i> mengubah artikel.....	45
Tabel 4.13 Skenario <i>use case</i> melihat profil desa .....	46
Tabel 4.14 Skenario <i>use case</i> melihat visi misi .....	46
Tabel 4.15 Skenario <i>use case</i> mengubah profil desa .....	47
Tabel 4.16 Skenario <i>use case</i> mengubah visi misi .....	47
Tabel 4.17 Skenario <i>use case</i> melihat profil <i>kades</i> .....	48
Tabel 4.18 Skenario <i>use case</i> melihat profil <i>sekdes</i> .....	48
Tabel 4.19 Skenario <i>use case</i> melihat profil perangkat desa.....	49
Tabel 4.20 Skenario <i>use case</i> melihat BPD .....	49
Tabel 4.21 Skenario <i>use case</i> menambah profil <i>Kades</i> .....	49
Tabel 4.22 Skenario <i>use case</i> menambah profil <i>Sekdes</i> .....	50
Tabel 4.23 Skenario <i>use case</i> menambah profil anggota perangkat .....	50
Tabel 4.24 Skenario <i>use case</i> tambah BPD .....	51
Tabel 4.25 Skenario <i>use case</i> mengubah profil <i>kades</i> .....	51
Tabel 4.26 Skenario <i>use case</i> mengubah profil <i>sekdes</i> .....	52
Tabel 4.27 Skenario <i>use case</i> mengubah profil perangkat .....	52
Tabel 4.28 Skenario <i>use case</i> mengubah profil BPD.....	53
Tabel 4.29 Skenario <i>use case</i> menghapus profil <i>kades</i> .....	53
Tabel 4.30 Skenario <i>use case</i> menghapus profil <i>sekdes</i> .....	54
Tabel 4.31 Skenario <i>use case</i> menghapus profil perangkat.....	54

Tabel 4.32 Skenario <i>use case</i> menghapus profil BPD .....	54
Tabel 4.33 Skenario <i>use case</i> melihat anggota LPMD .....	55
Tabel 4.34 Skenario <i>use case</i> melihat anggota PKK.....	55
Tabel 4.35 Skenario <i>use case</i> melihat anggota Karang Taruna .....	56
Tabel 4.36 Skenario <i>use case</i> melihat anggota RTRW .....	56
Tabel 4.37 Skenario <i>use case</i> tambah LPMD .....	56
Tabel 4.38 Skenario <i>use case</i> tambah anggota PKK.....	57
Tabel 4.39 Skenario <i>use case</i> menambah profil anggota Kartar .....	57
Tabel 4.40 Skenario <i>use case</i> tambah RT/RW.....	58
Tabel 4.41 Skenario <i>use case</i> menghapus profil anggota LPMD .....	58
Tabel 4.42 Skenario <i>use case</i> menghapus profil anggota PKK.....	59
Tabel 4.43 Skenario <i>use case</i> menghapus profil anggota Kartar .....	59
Tabel 4.44 Skenario <i>use case</i> menghapus profil anggota RTRW .....	59
Tabel 4.45 Skenario <i>use case</i> mengubah profil anggota LPMD.....	60
Tabel 4.46 Skenario <i>use case</i> mengubah profil anggota PKK .....	60
Tabel 4.47 Skenario <i>use case</i> mengubah profil anggota kartar .....	61
Tabel 4.48 Skenario <i>use case</i> mengubah profil anggota RTRW.....	61
Tabel 4.49 Skenario <i>use case</i> melihat data APBD .....	62
Tabel 4.50 Skenario <i>use case</i> melihat data RKPD .....	62
Tabel 4.51 Skenario <i>use case</i> tambah data APBD.....	63
Tabel 4.52 Skenario <i>use case</i> tambah data RKPD .....	63
Tabel 4.53 Skenario <i>use case</i> mengubah data APBD .....	63
Tabel 4.54 Skenario <i>use case</i> mengubah data RKPD .....	64
Tabel 4.55 Skenario <i>use case</i> menghapus data APBD .....	64
Tabel 4.56 Skenario <i>use case</i> menghapus data RKPD.....	65
Tabel 4.57 Skenario <i>use case</i> memvalidasi data APBD .....	65
Tabel 4.58 Skenario <i>use case</i> memvalidasi data RKPD .....	66
Tabel 4.59 Skenario <i>use case</i> melihat pesan aspirasi .....	66
Tabel 4.60 Skenario <i>use case</i> mengirim aspirasi.....	67
Tabel 4.61 Skenario <i>use case</i> menanggapi aspirasi .....	67
Tabel 4.62 Skenario <i>use case</i> menghapus aspirasi .....	67
Tabel 4.63 Skenario <i>use case</i> menghapus tanggapan aspirasi .....	68

Tabel 4.64 Skenario <i>use case</i> memvalidasi pesan aspirasi .....	68
Tabel 4.65 Skenario <i>use case</i> melihat data penduduk.....	69
Tabel 4.66 Skenario <i>use case</i> tambah data penduduk .....	69
Tabel 4.67 Skenario <i>use case</i> menghapus data penduduk.....	70
Tabel 4.68 Skenario <i>use case</i> mengubah data penduduk.....	70
Tabel 4.69 Skenario <i>use case</i> memesan surat .....	71
Tabel 4.70 Skenario <i>use case</i> melihat status pesanan.....	71
Tabel 4.71 Skenario <i>use case</i> melihat daftar pesanan.....	71
Tabel 4.72 Skenario <i>use case</i> mencetak surat .....	72
Tabel 4.73 Skenario <i>use case</i> menyimpan surat.....	72
Tabel 4.74 Skenario <i>use case</i> mengubah status pesanan.....	73
Tabel 4.75 Skenario <i>use case</i> mengirim notifikasi .....	73
Tabel 5.1 Perancangan algoritme memesan surat .....	79
Tabel 5.2 Perancangan algoritme mengedit profil BPD.....	79
Tabel 5.3 Perancangan algoritme menambah data APBD.....	80
Tabel 5.4 Implementasi algoritme memesan surat .....	85
Tabel 5.5 Implementasi algoritme edit BPD .....	86
Tabel 5.6 Implementasi algoritme tambah APBD.....	86
Tabel 6.1 Kasus uji memesan surat.....	91
Tabel 6.2 Kasus uji <i>edit</i> BPD .....	94
Tabel 6.3 Kasus uji tambah APBD .....	97
Tabel 6.4 Kasus uji berhasil <i>login</i> sebagai petugas .....	99
Tabel 6.5 Kasus uji gagal <i>login</i> .....	99
Tabel 6.6 Kasus uji berhasil menambah artikel .....	100
Tabel 6.7 Kasus uji batal menambah artikel .....	100
Tabel 6.8 Kasus uji gagal menambah artikel.....	101
Tabel 6.9 Kasus menekan tombol edit.....	102
Tabel 6.10 Kasus uji batal mengubah isi artikel .....	102
Tabel 6.11 Kasus uji menekan tombol hapus .....	103
Tabel 6.12 Kasus uji menekan tombol Ya .....	104
Tabel 6.13 Kasus uji menekan tombol Tidak.....	104
Tabel 6.14 Kasus uji berhasil menambah profil perangkat.....	105

Tabel 6.15 Kasus uji memasukkan file MP3 ke dalam <i>field</i> foto.....	105
Tabel 6.16 Kasus uji memasukkan gambar dengan ukuran lebih dari 1 MB .....	106
Tabel 6.17 Kasus uji memasukkan gambar dengan ukuran lebih dari 1 MB .....	106
Tabel 6.18 Kasus uji <i>field</i> NIP dengan jumlah karakter kurang dari 16 .....	106
Tabel 6.19 Kasus uji berhasil mengubah profil .....	108
Tabel 6.20 Kasus uji menekan menekan tombol batal .....	108
Tabel 6.21 Kasus uji menekan tombol hapus .....	109
Tabel 6.22 Kasus uji menekan tombol Ya .....	109
Tabel 6.23 Kasus uji menekan tombol Tidak.....	110
Tabel 6.24 Kasus uji berhasil menambah data <i>RKPD</i> .....	111
Tabel 6.25 Kasus uji gagal menambah data <i>RKPD</i> .....	111
Tabel 6.26 Kasus uji menekan tombol hapus .....	112
Tabel 6.27 Kasus uji menekan tombol Ya .....	112
Tabel 6.28 Kasus uji menekan tombol Tidak.....	112
Tabel 6.29 Kasus uji menekan tombol csv .....	114
Tabel 6.30 Kasus uji menekan tombol excel .....	114
Tabel 6.31 Kasus uji menekan tombol PDF.....	114
Tabel 6.32 Kasus uji berhasil memesan surat .....	116
Tabel 6.33 Kasus uji gagal memesan surat .....	116
Tabel 6.34 Kasus uji menekan tombol ubah status .....	118
Tabel 6.35 Kasus uji menekan tombol <i>save</i> .....	118
Tabel 6.36 Kasus uji menekan tombol batal .....	118
Tabel 6.37 Kasus uji menekan tombol <i>print</i> .....	120
Tabel 6.38 Kasus uji menekan tombol cetak .....	120
Tabel 6.39 Kasus uji menekan tombol batal .....	120
Tabel 6.40 Kasus uji menekan tombol kirim notifikasi .....	122
Tabel 6.41 Kasus uji menekan tombol batal .....	122
Tabel 6.42 Kasus uji menekan tombol kirim .....	122

## DAFTAR GAMBAR

Gambar 2.1 Model Linier (Pressman dan Maxim, 2015) .....	25
Gambar 2.2 Contoh <i>Use case</i> (Sommerville, 2010) .....	26
Gambar 2.3 Contoh <i>sequence diagram</i> (Sommerville, 2010) .....	27
Gambar 2.4 Contoh <i>class diagram</i> (Sommerville, 2010) .....	27
Gambar 2.5 <i>Flow graph</i> fungsi dasar (Pressman dan Maxim, 2015).....	28
Gambar 2.6 Proses pengujian <i>black box</i> (Sommerville, 2010) .....	29
Gambar 3.1 Bagan Penelitian.....	30
Gambar 4.1 <i>Use case diagram</i> .....	41
Gambar 5.1 <i>Sequence diagram</i> memesan surat.....	74
Gambar 5.2 <i>Sequence diagram</i> tambah data APBD .....	75
Gambar 5.3 <i>Sequence diagram</i> edit profil bpd .....	76
Gambar 5.4 <i>Conceptual data model</i> .....	77
Gambar 5.5 <i>Class Diagram</i> .....	78
Gambar 5.6 Perancangan antarmuka halaman utama.....	80
Gambar 5.7 Perancangan antarmuka halaman profil.....	81
Gambar 5.8 Perancangan antarmuka halaman lembaga .....	82
Gambar 5.9 Perancangan antarmuka halaman layanan sambung warga .....	82
Gambar 5.10 Perancangan antarmuka halaman layanan persuratan.....	83
Gambar 5.11 <i>Physical data model</i> .....	84
Gambar 5.12 <i>Physical data model</i> .....	85
Gambar 5.13 Antarmuka halaman profil .....	87
Gambar 5.14 Antarmuka halaman utama .....	87
Gambar 5.15 Antarmuka halaman lembaga .....	88
Gambar 5.16 Antarmuka halaman sambung warga .....	88
Gambar 5.17 Antarmuka halaman persuratan .....	89
Gambar 6.1 <i>Flow graph</i> pengujian unit memesan surat belum menikah .....	90
Gambar 6.2 Hasil pengujian unit memesan surat.....	92
Gambar 6.3 Hasil pengujian unit memesan surat.....	92
Gambar 6.4 Hasil pengujian unit memesan surat.....	92
Gambar 6.5 <i>Flow graph</i> pengujian unit edit BPD .....	93
Gambar 6.6 Hasil pengujian unit edit BPD.....	95

Gambar 6.7 Hasil pengujian unit edit BPD .....	95
Gambar 6.8 Hasil pengujian unit edit BPD .....	95
Gambar 6.9 <i>Flow graph</i> pengujian unit tambah APBD .....	96
Gambar 6.10 Hasil pengujian unit tambah APBD .....	98
Gambar 6.11 Hasil pengujian unit tambah APBD .....	98
Gambar 6.12 Hasil pengujian unit tambah APBD .....	98
Gambar 6.13 Hasil pengujian kasus uji berhasil <i>login</i> .....	99
Gambar 6.14 Hasil pengujian kasus uji gagal <i>login</i> .....	100
Gambar 6.15 Hasil pengujian kasus uji berhasil menambah berita atau artikel	101
Gambar 6.16 Hasil pengujian kasus uji batal menambah berita atau artikel.....	101
Gambar 6.17 Hasil pengujian kasus uji gagal menambah berita atau artikel ....	102
Gambar 6.18 Hasil pengujian kasus uji menekan tombol edit .....	103
Gambar 6.19 Hasil pengujian kasus uji menekan tombol batal .....	103
Gambar 6.20 Hasil pengujian kasus uji menekan tombol hapus.....	104
Gambar 6.21 Hasil pengujian kasus uji menekan tombol OK.....	104
Gambar 6.22 Hasil pengujian kasus uji menekan tombol Batal .....	105
Gambar 6.23 Hasil pengujian kasus uji berhasil menambah perangkat.....	107
Gambar 6.24 Hasil pengujian kasus uji file dengan format mp3 .....	107
Gambar 6.25 Hasil pengujian kasus uji gambar dengan ukuran 2MB .....	107
Gambar 6.26 Hasil pengujian kasus uji mengkosongkan bidang nama.....	107
Gambar 6.27 Hasil pengujian kasus uji memberi 2 karakter pada bidang NIP...	108
Gambar 6.28 Hasil pengujian kasus uji berhasil mengubah profil perangkat ....	108
Gambar 6.29 Hasil pengujian kasus uji menekan tombol batal .....	109
Gambar 6.30 Hasil pengujian kasus uji menekan tombol hapus.....	110
Gambar 6.31 Hasil pengujian kasus uji menekan tombol YES.....	110
Gambar 6.32 Hasil pengujian kasus uji menekan tombol NO .....	110
Gambar 6.33 Hasil pengujian kasus uji berhasil menambah data <i>RKPD</i> .....	111
Gambar 6.34 Hasil pengujian kasus uji gagal menambah data <i>RKPD</i> .....	112
Gambar 6.35 Hasil pengujian validasi fungsional hapus <i>RKPD</i> .....	113
Gambar 6.36 Hasil pengujian validasi fungsional hapus <i>RKPD</i> .....	113
Gambar 6.37 Hasil pengujian validasi fungsional hapus <i>RKPD</i> .....	113
Gambar 6.38 Hasil pengujian validasi fungsional simpan <i>RKPD</i> .....	115

Gambar 6.39 Hasil pengujian validasi fungsional simpan <i>RKPD</i> .....	115
Gambar 6.40 Hasil pengujian validasi fungsional simpan <i>RKPD</i> .....	115
Gambar 6.41 Hasil pengujian validasi fungsional memesan surat pengantar....	117
Gambar 6.42 Hasil pengujian validasi fungsional memesan surat pengantar....	117
Gambar 6.43 Hasil pengujian validasi fungsional mengubah status pesanan....	119
Gambar 6.44 Hasil pengujian validasi fungsional mengubah status pesanan....	119
Gambar 6.45 Hasil pengujian validasi fungsional mengubah status pesanan....	119
Gambar 6.46 Hasil pengujian validasi fungsional mencetak pesanan surat.....	121
Gambar 6.47 Hasil pengujian validasi fungsional mencetak pesanan surat.....	121
Gambar 6.48 Hasil pengujian validasi fungsional mencetak pesanan surat.....	121
Gambar 6.49 Hasil pengujian validasi fungsional mengirim notifikasi .....	123
Gambar 6.50 Hasil pengujian validasi fungsional mengirim notifikasi .....	123
Gambar 6.51 Hasil pengujian validasi fungsional mengirim notifikasi .....	123
Gambar 6.52 Hasil Pengujian <i>Compatibility</i> menggunakan aplikasi <i>SortSite</i> .....	124
Gambar 6.53 Tampilan sistem dengan google chrome .....	124
Gambar 6.54 Tampilan sistem dengan mozilla firefox .....	125
Gambar 6.55 Tampilan dengan microsoft edge .....	125
Gambar 6.56 Tampilan <i>Website</i> pada perangkat <i>mobile</i> .....	125



## DAFTAR LAMPIRAN

LAMPIRAN A BUKU DESA .....	128
A.1 Profil Desa.....	128
A.2 Visi Misi.....	131
A.3 Profil Pemerintahan.....	133
A.4 Lembaga Kemasyarakatan.....	134
LAMPIRAN B FORMAT SURAT .....	135
B.1 Surat Keterangan Belum Menikah.....	135
B.2 Surat Keterangan Menikah .....	136
B.3 Surat Keterangan Cerai.....	137
B.4 Surat Keterangan Jual-Beli.....	138
B.5 Surat Keterangan Usaha .....	139
B.6 Surat Keterangan Penghasilan.....	140
B.7 Surat Keterangan Tidak Mampu.....	141
B.8 Surat Keterangan Kehilangan .....	142
B.9 Surat Pengantar .....	143
B.10 Surat Keterangan Pindah .....	144
B.11 Surat Pengajuan Kartu Keluarga.....	145
LAMPIRAN C BUKU ADMINISTRASI .....	146
C.1 Buku Aparat Pemerintahan Desa .....	146
C.2 Buku Induk Penduduk.....	146
C.3 Buku Anggaran Desa.....	146

## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Evolusi teknologi informasi banyak mengakibatkan adanya peralihan sistem kerja konvensional menjadi sistem digital. Tidak terkecuali pada institusi pemerintahan desa, sebagai tolok ukur keberhasilan dalam melayani masyarakat, sudah sepatutnya pemerintah memberikan fasilitas agar masyarakat dapat dengan mudah mendapat informasi serta melakukan kegiatan administrasi persuratan di kantor desa.

Desa Pelangwot merupakan salah satu desa pada Kecamatan Laren kabupaten Lamongan yang sampai saat ini masih menggunakan sistem kerja konvensional dalam proses pelayanan masyarakat dan mengharuskan masyarakat untuk datang ke kantor desa. Hal tersebut kurang efektif mengingat pejabat desa yang terkadang memiliki kepentingan di luar kantor atau tuntutan dinas, akibatnya masyarakat diharuskan untuk datang kembali pada lain waktu.

Jarak tempuh dusun menuju kantor desa yang mencapai 3,5 Km dengan kondisi jalan yang kurang baik menjadi faktor yang dapat mempersulit dan membuat masyarakat malas untuk melakukan pengurusan administrasi, terlebih bagi masyarakat yang bekerja di siang hari banyak yang merasa terganggu pekerjaannya jika harus berulang-ulang mengurus ke kantor desa. Di sisi lain penyimpanan data demografi menggunakan kertas, rawan terhadap adanya kerusakan serta susahya menata dan pencarian data tersebut.

Berangkat dari permasalahan tersebut dibutuhkan adanya media digital yang dapat digunakan sebagai sumber informasi dan pelayanan masyarakat secara daring, agar dapat memudahkan masyarakat serta pemerintah desa dalam menjalankan fungsi pemerintahan dengan lebih baik.

### 1.2 Rumusan masalah

Mengacu pada bahasan latar belakang yang ada, maka rumusan masalahnya adalah bagaimana proses pengembangan sistem aplikasi yang dapat digunakan pemerintah desa Pelangwot untuk melayani masyarakat dalam kegiatan administrasi?

### 1.3 Tujuan

Tujuan dari penelitian ini adalah untuk mengembangkan sistem yang dapat digunakan sebagai media pelayanan dan informasi masyarakat, serta sebagai penunjang kinerja aparat pemerintahan desa Pelangwot dalam kegiatan administrasi.

## 1.4 Manfaat

Adapun manfaat-manfaat yang didapat dari penelitian ini adalah:

1. Memajukan teknologi informasi desa Pelangwot.
2. Mempermudah pelayanan masyarakat di bidang administrasi dan surat-menyerurat.
3. Meningkatkan kemampuan dalam bidang teknologi informasi dan komunikasi bagi aparaturnya desa.

## 1.5 Batasan masalah

Untuk mempersempit kemungkinan penyebaran masalah dalam penulisan penelitian ini, maka ditetapkan batasan-batasan sebagai berikut:

1. Layanan Administrasi yang terdapat pada sistem mencakup pada Administrasi Umum, Administrasi Penduduk, dan Administrasi Keuangan.
2. Ruang lingkup administrasi umum terbatas hanya pada buku aparat pemerintahan desa dan kelurahan.
3. Ruang lingkup administrasi penduduk terbatas hanya pada buku induk penduduk.
4. Ruang lingkup administrasi keuangan terbatas hanya pada buku anggaran desa.

## 1.6 Sistematika pembahasan

### BAB I PENDAHULUAN

Bab ini berisi latar belakang yang menjadi dasar penelitian, rumusan masalah yang berkaitan dengan hal yang ingin dikaji dalam penelitian, tujuan yang ingin didapatkan dari penelitian, manfaat dari penelitian yang sudah dilaksanakan, batasan masalah, serta sistematika pembahasan.

### BAB II TINJAUAN PUSTAKA

Bagian tinjauan pustaka menguraikan teori penunjang yang mendasari proses penelitian.

### BAB III METODOLOGI PENELITIAN

Bab metode penelitian berisi penjelasan mengenai studi literatur sebagai acuan dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta penarikan kesimpulan dan saran.

### BAB IV REKAYASA KEBUTUHAN

Dalam bab ini diuraikan penjelasan mengenai analisis kebutuhan yang dibutuhkan untuk membangun aplikasi sesuai dengan apa yang dibutuhkan baik kebutuhan baik pengguna maupun kebutuhan bagi sistem yang akan dibangun.

## **BAB V PERANCANGAN DAN IMPLEMENTASI**

Bagian ini berisi deskripsi perancangan struktur, komponen, representasi data, serta implementasi dari fungsi yang menjadi fokus yang didasarkan pada apa yang telah dirancangan sebelumnya.

## **BAB VI PENGUJIAN**

Pada bab ini menjelaskan mengenai strategi dan teknik pengujian yang dilakukan serta penjelasan mengenai analisis yang telah dilakukan terhadap sistem yang dibangun, dan terdapat kasus uji serta hasil pengujian yang didapatkan.

## **BAB VII PENUTUP**

Dalam bagian penutup menjelaskan mengenai kesimpulan yang berisikan jawaban dari rumusan masalah yang telah dibuat sebelumnya dengan berdasarkan hasil penelitian, dan terdapat saran guna mengembangkan penelitian lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Kajian penelitian yang digunakan dalam penelitian ini merujuk pada penelitian sebelumnya yang dikerjakan oleh Hidayatulloh dan Mulyadi (2015) dengan judul Sistem Pelayanan Administrasi Kependudukan Desa Candigatak Berbasis web. Hidayatulloh dan Mulyadi membuat sebuah sistem yang dapat melakukan pelayanan persuratan desa seperti surat keterangan lahir, keterangan menikah hingga surat pengantar pembuatan catatan kepolisian. Dari penelitian ini didapatkan kesimpulan bahwasanya masyarakat serta aparat Desa Candigatak sangat dimudahkan dengan adanya sistem pelayanan administrasi kependudukan desa, terlebih sistem informasi ini berbasis web sehingga dapat diakses di berbagai tempat dan *platform*. Selain itu Wiharto (2011) dalam penelitiannya yang berjudul sistem informasi akademik berbasis sms *gateway* mendapatkan kesimpulan bahwa dengan adanya sistem yang telah ia bangun para siswa dapat dengan cepat dan mudah mengetahui informasi-informasi penting dari sekolah. Begitu juga dengan pihak sekolah, dengan adanya sistem informasi akademik berbasis sms *gateway* ini mereka merasa terbantu dalam hal pengumuman dan penyampaian informasi yang bersifat masal.

### 2.2 Software Application

*Software* atau perangkat lunak adalah program komputer yang ketika dijalankan memberikan fitur, fungsi, dan kinerja yang memungkinkan proses manipulasi informasi secara memadai. Sedangkan Perangkat lunak aplikasi adalah program yang berdiri sendiri yang memproses bisnis atau data teknis dengan cara yang memfasilitasi operasi bisnis atau manajemen pengambilan keputusan secara teknis (Pressman dan Maxim, 2015).

### 2.3 Administrasi Desa

Administrasi dalam arti yang luas adalah segenap proses kegiatan untuk mencapai suatu tujuan, sedangkan dalam arti yang sempit administrasi adalah proses pelayanan untuk mencapai suatu tujuan (Widjaja, 2002). Desa adalah suatu wilayah yang ditempati oleh sejumlah penduduk sebagai kesatuan masyarakat termasuk di dalamnya kesatuan masyarakat hukum yang mempunyai organisasi pemerintahan terendah di bawah camat dan berhak menyelenggarakan rumah tangganya sendiri dalam ikatan Negara Kesatuan Republik Indonesia (Widjaja, 2002).

Dari pengertian tersebut dapat ditarik kesimpulan bahwasanya administrasi desa merupakan segenap proses kegiatan yang dilakukan oleh badan desa dengan tujuan ikut serta dalam penyelenggaraan rumah tangga di Negara Kesatuan Republik Indonesia. Dalam Peraturan Kementrian Dalam Negeri (2016) No. 47 Tahun 2016 BAB 2 Pasal 2 disebutkan bahwa ruang lingkup administrasi meliputi

administrasi umum, administrasi penduduk, administrasi keuangan, dan administrasi pembangunan.

Administrasi umum merupakan kegiatan pencatatan data mengenai kegiatan-kegiatan pemerintahan desa dan pemerintahan kelurahan, administrasi tersebut meliputi beberapa buku seperti keputusan desa, keputusan kepala desa, kekayaan desa, dan buku aparat desa. Administrasi penduduk sendiri berisikan data kependudukan seperti perubahan penduduk, perkembangan penduduk, dan jumlah penduduk. Sementara administrasi keuangan dan administrasi pembangunan, masing-masing memuat data anggaran pendapatan belanja desa dan rencana kerja pembangunan desa.

## 2.4 SMS Gateway

*Short message service (SMS)* merupakan mekanisme yang digunakan dalam penyampaian pesan singkat dalam jaringan *mobile*, sedangkan *sms gateway* adalah layanan yang menawarkan mekanisme transmisi pesan, mengubah pesan pada jaringan seluler ke dalam bentuk media lain atau sebaliknya. Hal tersebut memungkinkan adanya mekanisme pengiriman atau penerimaan pesan SMS dengan atau tanpa menggunakan ponsel (Katankar dan Thakare, 2010).

Layanan *SMS Gateway* yang digunakan dalam pengerjaan sistem ini yaitu Gammu. Gammu merupakan aplikasi yang dapat digunakan untuk mengontrol ponsel, modem, dan perangkat sejenis lainnya (Wammu, 2018). Fungsi-fungsi yang tersedia dalam gammu antara lain: Buku telepon, daftar panggilan, *SMS retrieval*, *MMS retrieval*, dan lain-lain.

Beberapa kelebihan gammu dibanding aplikasi *SMS Gateway* lainnya adalah:

1. Dapat dijalankan baik pada platform windows maupun Linux.
2. Kompatibel dengan berbagai perangkat.
3. Menggunakan database MySql.
4. *Open Source* sehingga dapat digunakan secara gratis.
5. Gammu tidak memerlukan banyak hardware

Pengaturan yang harus dikerjakan selama pembentukan *SMS Gateway* dengan menggunakan Gammu adalah perubahan *device*, *connection*, dan *database* yang terdapat pada file *gammudrc* dan *smsdrc* pada folder bin.

### File gammudrc

```
[Gammu]
device = com5:
connection = at115200
```

**File smsdrc**

```
[gammu]
device = com6:
connection = at

[smsd]
Service = sql
Driver = odbc // driver koneksi database menggunakan ODBC
SQL = mysql
host = localmysql //ini adalah koneksi ODBC ke mysql yang
kita buat
PIN = 1234
logfile = smsdlog
phoneid = phone1
User = root //username mysql Anda
Password = // passwore mysql Anda
Database = sms // database gammu anda
```

*Device* dan *connection* dapat dilihat pada *device manager* masing-masing komputer pada *port COM & LPT PC UI Interface*. Beberapa perintah atau *command line* yang terdapat pada *Gammu SMS Gateway* antara lain:

**1. Identifikasi Perangkat**

```
gammu --identify
```

**2. Inisialisasi Service**

```
gammu-smsd -i -c smsdrc -n gammuSMSD
```

**3. Service Start**

```
gammu-smsd -s -c smsdrc -n gammuSMSD
```

**4. Uninstall Service**

```
gammu-smsd -u -c smsdrc -n gammuSMSD
```

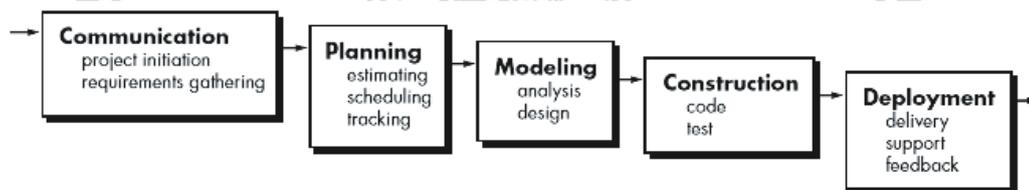
## 2.5 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan disiplin ilmu yang mempelajari segala aspek produksi dalam perangkat lunak yang dimulai dari tahap spesifikasi sistem hingga mempertahankan sistem setelah digunakan (Sommerville, 2010). Sedang Bjørner (2006) mendefinisikan bahwa rekayasa perangkat lunak (RPL) lebih mementingkan efisiensi pengerjaan untuk memuaskan pengguna dengan asumsi masalah yang dihadapi pengguna telah mampu terpecahkan. Dari definisi di atas dapat disimpulkan bahwasanya RPL tidak hanya berkuat pada perencanaan dan perancangan, RPL didefinisikan sebagai satuan proses yang saling terintegrasi mulai dari sebelum perangkat lunak itu dibuat, tahap perancangan, implementasi, serta pasca implementasi sehingga siklus hidup perangkat lunak dapat berlangsung secara efisien dan terukur.

Siklus hidup perangkat lunak merupakan tahapan yang harus dilalui dalam proses pengembangan sistem dari konsep awal hingga tahapan terakhir. Tahap-tahap pengembangan sistem tersebut dapat berisikan alur informasi, saat penentuan keputusan, dan lain sebagainya (Kadir, 2003). *The software life cycle is the sequence of different activities that take place during software development* (Gustafson, 2002). Hal tersebut menunjukkan bahwa siklus hidup perangkat lunak merupakan urutan kegiatan selama proses pengembangan berlangsung. Urutan tersebut tentunya tidak harus benar-benar urut, akan tetapi dapat didasarkan pada jenis siklus hidup yang digunakan dalam pengembangan perangkat lunak itu sendiri. Siklus hidup perangkat lunak sendiri memiliki beberapa jenis atau model di antaranya: model *waterfall*, *prototyping*, *incremental*, dan model spiral.

### 2.5.1 Model Waterfall

Model *waterfall* linier diciptakan pertama kali oleh Royce pada tahun 1997. Model *clasic life cycle* ini disebut sebagai model *waterfall* karena proses eksekusi yang selaras dengan deretan air terjun (Gustafson, 2002). Model ini diilustrasikan pada Gambar 2.1.



Gambar 2.1 Model Linier (Pressman dan Maxim, 2015)

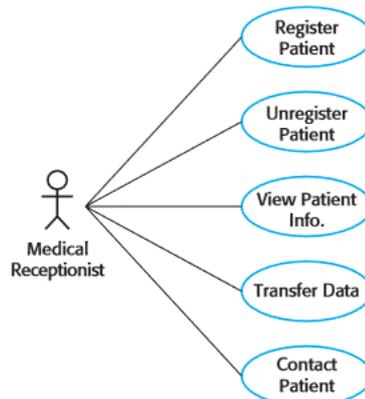
Model *waterfall* sendiri merupakan salah satu siklus hidup yang prosesnya berbasis pada rencana. *Developer* harus memiliki rencana dan menjadwalkan semua aktivitas sebelum melakukan pengerjaan pengembangan sistem. Hal tersebut dikarenakan proses eksekusi model linier mengharuskan masing-masing dari kelima tahapan dikerjakan secara berurutan. Dimulai dengan komunikasi, spesifikasi pengguna, desain, implementasi, hingga pada proses uji coba dan penyebaran sistem terhadap pengguna (Pressman dan Maxim, 2015).

## 2.6 Pemodelan sistem

Pemodelan sistem adalah proses pembuatan model abstrak dari suatu sistem guna merepresentasikan perspektif kebutuhan sistem itu sendiri. Pemodelan sistem berbasis objek pada umumnya digambarkan dalam notasi grafis yang didasarkan pada notasi dalam *Unified Modeling Language* (Sommerville, 2010). *Unified Modeling Language* (UML) sendiri merupakan bahasa standar yang digunakan untuk membuat *blueprint* dari perangkat lunak. UML digunakan untuk memvisualisasikan sistem, menentukan pembangunan sistem, serta mendokumentasikan artefak sistem perangkat lunak (Pressman dan Maxim, 2015).

### 2.6.1 Use case Diagram

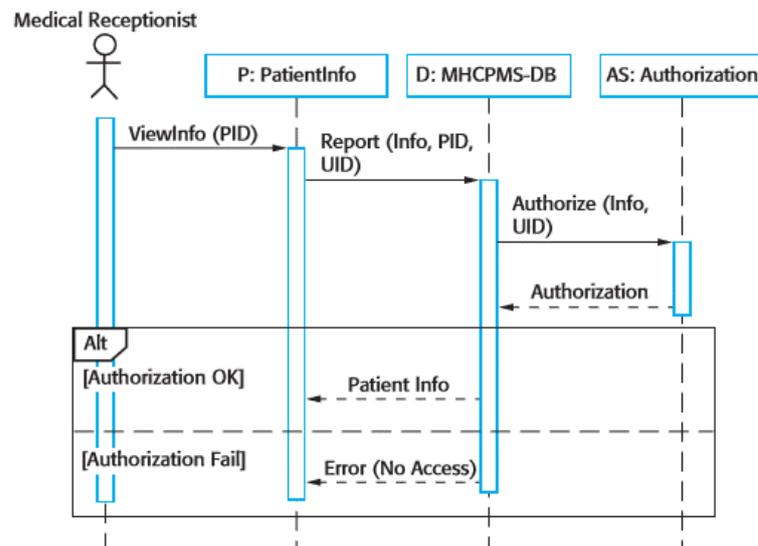
*Use case* diagram digunakan untuk merepresentasikan interaksi *external* dengan suatu sistem. Dalam bentuk yang paling sederhana, *Use case* diwujudkan dalam bentuk elips dengan aktor yang terlibat diwujudkan dalam bentuk *stick figures* (Sommerville, 2010). Dalam Gambar 2.2 berikut ini menggambarkan seorang *medical receptionist* (aktor) dapat menjalankan beberapa kebutuhan (*Use case*).



Gambar 2.2 Contoh *Use case* (Sommerville, 2010)

### 2.6.2 Sequence Diagram

*Sequence* diagram digunakan untuk merepresentasikan interaksi antara aktor dan objek serta interaksi antara objek itu sendiri dalam suatu sistem. Diagram ini menunjukkan urutan interaksi yang terjadi selama kasus penggunaan tertentu (Sommerville, 2010). Gambar 2.3 adalah contoh *sequence* diagram yang menggambarkan dasar-dasar notasi. Objek dan aktor yang terlibat terletak pada bagian atas diagram. Interaksi antara objek ditunjukkan oleh panah bernotasi. Persegi panjang pada garis putus-putus menunjukkan garis hidup dari objek yang bersangkutan. Anotasi pada panah menunjukkan panggilan ke objek, parameter, dan nilai kembali. Sebuah kotak bernama *alt* digunakan untuk menunjukkan alternatif dengan kondisi yang ditunjukkan dalam tanda kurung siku.

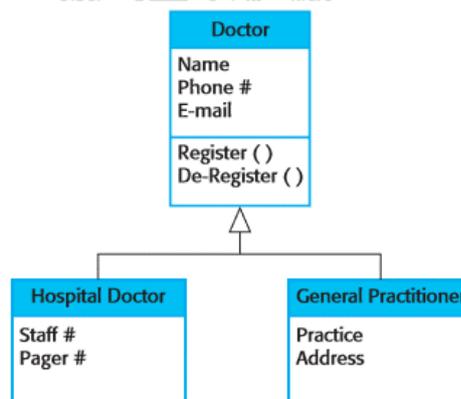


Gambar 2.3 Contoh *sequence* diagram (Sommerville, 2010)

### 2.6.3 Class Diagram

*Class* diagram digunakan ketika mengembangkan model sistem berorientasi objek untuk menunjukkan kelas dan hubungan antar kelas dalam suatu sistem. Elemen utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan antarmuka. Setiap kotak dibagi menjadi bagian horizontal. Bagian pertama (atas) berisi nama kelas. Bagian kedua (tengah) berisi atribut-atribut kelas. Bagian ketiga dari diagram kelas berisi operasi atau perilaku kelas. Operasi mengacu pada apa saja yang bisa dilakukan oleh kelas yang biasanya diimplementasikan sebagai metode (Sommerville, 2010).

Gambar 2.4 berikut ini merupakan contoh diagram kelas dengan hubungan generalisasi.



Gambar 2.4 Contoh *class* diagram (Sommerville, 2010)

## 2.7 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan kegiatan yang dilakukan dengan tujuan untuk menemukan adanya kesalahan pada suatu sistem yang sedang atau telah dikembangkan guna memastikan bahwasanya sistem dapat bekerja tanpa adanya kesalahan atau kegagalan. Selain itu pengujian perangkat lunak juga dilakukan untuk membuktikan jika kebutuhan pengguna telah terpenuhi. Dua Pendekatan dasar dalam pengujian perangkat lunak yaitu pengujian kotak putih dan pengujian kotak hitam (Khan, 2011b).

### 2.7.1 Pengujian Unit

Pengujian unit merupakan pengujian yang berfokus pada unit modul atau kelas-kelas terkecil pada sistem. Pengujian ini dilakukan dengan menggunakan teknik pengujian *white box*.

Khan (2011b) dalam jurnalnya yang berjudul *Different Approaches to White Box Testing Technique for Finding Errors*, *white box testing* merupakan teknik pengujian yang didasarkan pada analisis kerja internal sistem, dalam artian pengujian ini digunakan untuk menguji baris kode guna mengungkap kesalahan pada sebuah fungsi tertentu. Pengujian kotak putih dapat diterapkan pada tingkat pengujian rendah seperti pengujian unit.

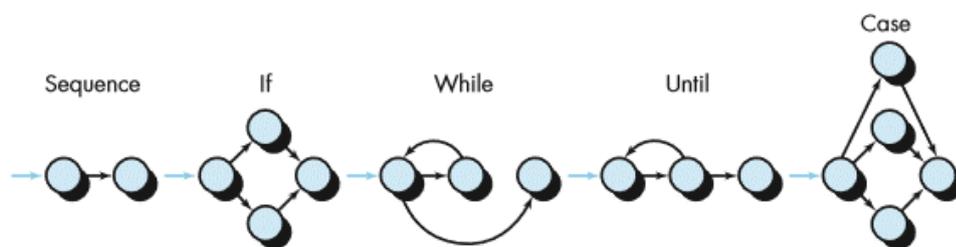
*White box* mencakup beberapa kriteria pengujian seperti *control flow*, *data flow*, *branch*, *looping*, dan *basis path*. Pengujian jalur dasar atau *basis path testing* merupakan salah satu teknik pengujian *white box* yang memastikan bahwa setiap jalur independen yang ada telah melalui jalur kode seperti apa yang telah dirancang (Khan, 2011b). Beberapa tahapan yang harus dikerjakan dalam pengujian *basis path* adalah:

1. Mendefinisikan *flow graph*.

*Flow graph* merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol program. Terdapat tiga jenis notasi yang digunakan dalam *flow graph* yaitu:

- Simbol berbentuk bulat disebut dengan *node*.
- Simbol panah disebut dengan *edge*.
- Area yang dibatasi oleh *node* dan *edge* disebut dengan region.

Gambar 2.5 berikut ini menunjukkan *Basic flow graph* yang digunakan sebagai dasar pengerjaan pengujian *basis path*:



Gambar 2.5 *Flow graph* fungsi dasar (Pressman dan Maxim, 2015)

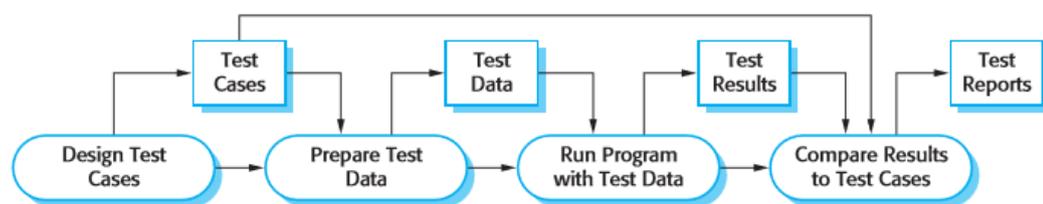
2. Menghitung ukuran kompleksitas (*cyclomatic complexity*).  
 Dalam kasus pengujian *basis path*, ukuran kompleksitas dapat digunakan untuk menentukan jumlah jalur independen. Beberapa rumus perhitungan ukuran kompleksitas adalah sebagai berikut:
  - $V(G) = P + 1$ . P adalah jumlah dari *node* predikat.
  - $V(G)$  = jumlah dari *region*.
  - $V(G) = E - N + 2$ . E adalah jumlah *edge* sedangkan N adalah jumlah *node*.
3. Menentukan kasus uji atau *test case*.

### 2.7.2 Pengujian Validasi

Pengujian validasi dikerjakan setelah semua komponen fungsi telah berhasil digabungkan. Level pengujian ini berfokus pada apa yang menjadi masukan *user* dan apa yang menjadi keluaran sistem dengan menggunakan teknik pengujian *black box*.

*Black Box Testing* merupakan pendekatan pengujian yang berfokus pada persyaratan fungsional sistem yang ada. Dalam pengujian ini seorang penguji tidak perlu mengetahui kerja internal suatu sistem, hanya perlu mengetahui apa yang menjadi masukan dan keluaran sistem untuk mengetahui apakah terdapat kesalahan dan apakah sistem tersebut telah memenuhi apa yang menjadi kriteria pengguna (Khan, 2011a).

Tahapan pertama yang harus dikerjakan dalam pengujian ini adalah penentuan *test case* dan *test data*. *Test case* merupakan spesifikasi pengujian seperti masukan untuk *test* serta keluaran yang diharapkan dari hasil pengujian. Sedangkan *test data* merupakan data yang menjadi masukan dari fungsi yang akan diuji. Diagram alir proses pengujian ini ditunjukkan dalam Gambar 2.6 berikut.



Gambar 2.6 Proses pengujian *black box* (Sommerville, 2010)

### 2.7.3 Pengujian *Compatibility*

Pengujian *compatibility* merupakan jenis pengujian *non-fungsional*, pengujian ini digunakan untuk memastikan kepuasan pengguna, kecocokan dengan berbagai *browser*, *database*, perangkat keras, sistem operasi, perangkat *mobile*, dan jaringan yang berbeda (Desyatnikov, 2018).

Pengujian ini dapat dilakukan dengan menggunakan aplikasi *PowerMapper* yang bernama *sortSite*. Aplikasi ini akan melakukan pemindaian terhadap semua halaman yang ada pada web. Mengidentifikasi setiap *thumbnail*, *metadata*, dan *URL* dari sistem yang akan diuji.

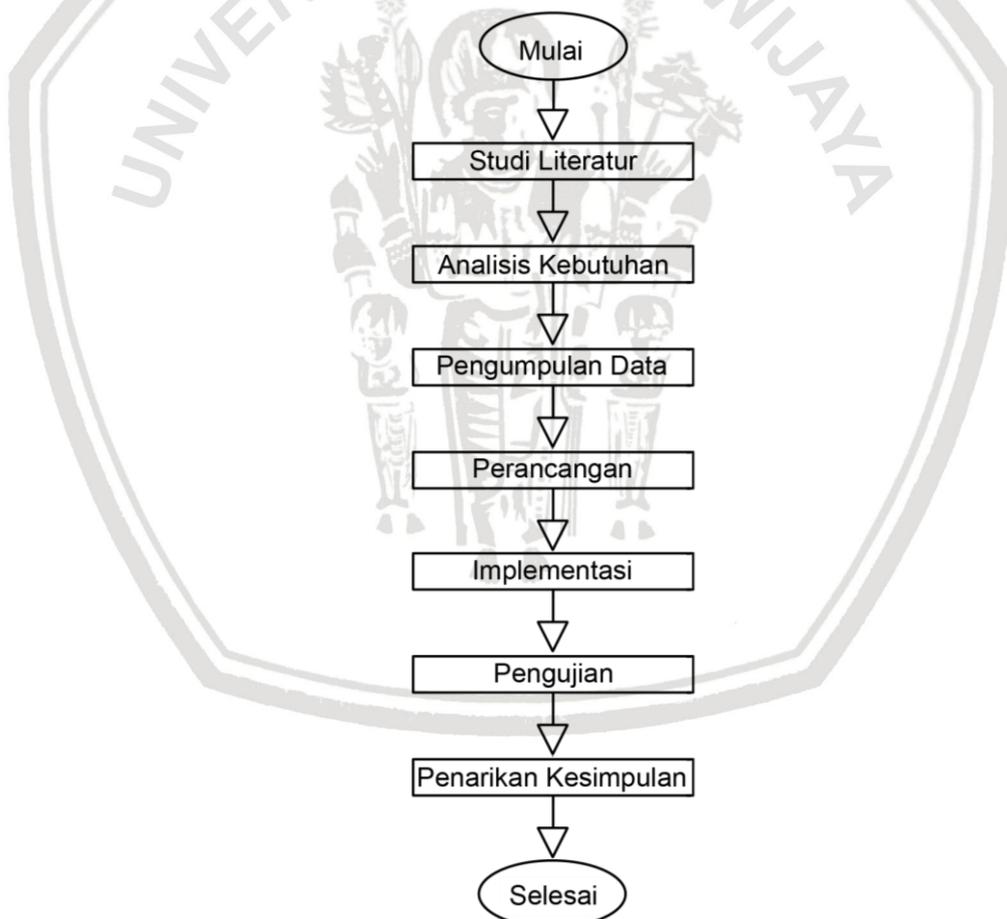
## BAB 3 METODOLOGI

### 3.1 Tipe Penelitian

Tipe penelitian yang digunakan dalam penelitian ini adalah tipe Implementatif - Pengembangan, tipe penelitian ini merupakan bagian dari kegiatan penelitian implementatif untuk membuat sebuah produk utuh dari sesuatu yang belum ada menjadi ada dengan menerapkan prinsip-prinsip rekayasa secara utuh, yang meliputi analisis, perancangan, implementasi dan pengujian.

### 3.2 Strategi dan Rancangan Penelitian

Dalam melakukan penelitian ini penulis membagi menjadi beberapa tahapan, dimulai dengan melakukan pencarian bahasan yang sesuai dengan topik penelitian penulis, menganalisis kebutuhan sistem, pengumpulan data berdasarkan hasil analisis kebutuhan, perancangan sistem dan implementasi ke dalam bahasa pemrograman untuk dilakukan pengujian dan penarikan kesimpulan. Bagan penelitian penulis terdapat pada Gambar 3.1 berikut ini.



Gambar 3.1 Bagan Penelitian

### 3.2.1 Studi Literatur

Pada tahap ini penulis melakukan proses penyusunan teori dasar serta pustaka yang dirasa relevan dengan penelitian yang akan dikerjakan, teori dan pustaka biasanya didapat dari buku atau hasil penelitian sebelumnya yang nantinya dari beberapa pustaka yang didapat akan dijadikan bahan acuan atau referensi. Beberapa teori dan pustaka tersebut antara lain:

1. Sistem informasi
2. Administrasi desa
3. SMS Gateway
4. Rekayasa perangkat lunak
5. Pemodelan sistem
6. Pengujian perangkat lunak

### 3.2.2 Analisis Kebutuhan

Setelah didapatkan literatur, tahapan selanjutnya ialah melakukan analisis kebutuhan, tahapan ini dilakukan guna menentukan fungsional yang akan dikerjakan dalam pembuatan sistem. Tahapan ini dilakukan dengan melakukan konsultasi dan wawancara terhadap pihak kantor desa Pelangwot untuk mendapatkan fungsional yang dibutuhkan oleh pengguna. Hasil analisis ini kemudian diwujudkan ke dalam diagram *use case* dan *use case scenario*.

### 3.2.3 Pengumpulan Data

Dalam melakukan pengumpulan data yang akan digunakan dalam pembangunan sistem, penulis menggunakan beberapa metode yaitu wawancara dan studi dokumen.

#### 1. Wawancara

Wawancara dilakukan dengan pihak kantor desa, dalam hal ini adalah petugas dan kepala desa yang nantinya akan menjadi operator sistem.

#### 2. Studi Dokumen.

Metode ini dilakukan guna mempelajari dokumen-dokumen yang berkaitan dengan kebutuhan sistem, seperti format persuratan yang mana telah mendapat izin dari pihak kantor desa Pelangwot.

### 3.2.4 Perancangan Sistem

Perancangan sistem bertujuan untuk memenuhi kebutuhan pengguna akan gambaran, alur kerja, dan aktivitas secara sistematis yang diwujudkan ke dalam diagram aktivitas serta eksemplar antarmuka aplikasi. Selain itu terdapat juga proses perancangan basis data, *class*, dan algoritme.

### 3.2.5 Implementasi

Implementasi merupakan proses pembuatan sistem. Tahapan ini mengacu pada tahap perancangan yang meliputi basis data dan implementasi antarmuka pengguna. Pada pembuatan sistem ini penulis menggunakan bahasa pemrograman php dan javascript, serta mysql sebagai sistem manajemen basis data.

### 3.2.6 Pengujian

Tahapan pengujian merupakan tahapan yang dilakukan guna mengetahui kesesuaian sistem dengan apa yang telah ditetapkan pada tahap perancangan. Selain itu pada tahap ini juga dilakukan analisis untuk menemukan kesalahan yang terdapat pada sistem yang sedang dikerjakan. Beberapa level pengujian yang penulis kerjakan yaitu pengujian unit, dan pengujian validasi.

Pengujian Unit dilakukan pada modul unit terkecil pada program. Pada penelitian ini penulis menggunakan *basis path testing* yang dilakukan pada tiga sampel uji yang berasal dari algoritme pada masing-masing kelas atau fungsi individual. Pengujian validasi dilakukan dengan menggunakan pengujian *black box* terhadap seluruh kebutuhan untuk memastikan kesesuaian hasil implementasi dengan apa yang telah dispesifikasikan sebelumnya. Sedangkan pengujian *compatibility* dilakukan dengan menggunakan tipe pengujian *browser compatibility* dan perangkat *mobile* terhadap beberapa *browser* seperti *Chrome*, *Firefox*, *Internet Explorer*, *Safari*, dan *Opera*.

### 3.2.7 Penarikan Kesimpulan

Penarikan kesimpulan dikerjakan setelah semua tahapan yang dimulai dari perancangan, implementasi, dan pengujian selesai dilakukan. Tahapan ini mengacu pada hasil pengujian dan analisis sistem yang telah dikerjakan. Tahapan ini berfungsi untuk mengevaluasi terjadinya kesalahan serta menyempurnakan penulisan sehingga dapat digunakan untuk pertimbangan atas pengembangan sistem di kemudian hari.

## BAB 4 REKAYASA KEBUTUHAN

### 4.1 Deskripsi Sistem

Sistem ini merupakan sebuah sistem informasi desa berbasis web yang ditujukan sebagai wadah transparansi data pemerintah terhadap masyarakat desa Pelangwot. Layanan yang disediakan pada sistem ini yakni layanan administrasi dan pemesanan surat.

Jenis administrasi yang disajikan pada sistem ini terbatas pada beberapa buku administrasi saja seperti buku penduduk, buku aparat pemerintah desa dan kelurahan, buku anggaran desa, serta buku rencana kerja. Sedangkan surat yang dapat dipesan melalui sistem ini adalah surat keterangan belum menikah, surat keterangan nikah, surat keterangan cerai, surat keterangan jual beli, surat keterangan usaha, surat keterangan kehilangan, surat keterangan penghasilan, surat keterangan tidak mampu, dan surat pengantar. Terdapat pula fitur notifikasi yang digunakan untuk mengirim pesan mengenai status pemesanan surat yang dibangun dengan menggunakan layanan *SMS Gateway*.

Pada sistem ini terdapat dua aktor yakni petugas dan masyarakat. Petugas akan bertindak sebagai Admin yang menangani permintaan surat serta mengelola isi dari sistem informasi ini sendiri, sementara aktor kedua yaitu masyarakat yang hanya dapat melakukan beberapa fungsional seperti melihat informasi, menyampaikan aspirasi dan memesan surat.

Sistem ini dibuat dengan menggunakan bahasa pemrograman PHP, Javascript dengan menggunakan beberapa *framework* seperti *Bootstrap*, *jQuery*, dan *codeigniter*. Sistem ini *mobile friendly*, dapat digunakan pada berbagai *browser*, dan Basis data yang digunakan adalah MySQL.

### 4.2 Identifikasi Pengguna

Mengacu pada hasil observasi dan wawancara dengan pihak kantor desa Pelangwot, didapatkan empat jenis atau klasifikasi pengguna yaitu *Guest*, petugas kantor desa dan masyarakat. Tabel 4.1 merupakan tabel yang mendeskripsikan pengguna sistem serta hal-hal yang dapat digunakan oleh pengguna tersebut.

**Tabel 4.1 Identifikasi pengguna**

No.	Klasifikasi Pengguna	Deskripsi Pengguna
1	<i>Guest</i>	Pengguna sistem yang berperan sebagai pengunjung. Dalam hal ini pengunjung merupakan masyarakat umum yang hanya dapat melihat informasi dan tidak dapat melakukan pemesanan surat atau kegiatan administrasi.
2	Warga	Pengguna sistem yang berperan sebagai masyarakat merupakan warga desa Pelangwot yang data kependudukannya tersimpan dalam arsip desa. Masyarakat dapat melihat informasi serta melakukan pemesanan surat.

**Tabel 4.1 Identifikasi pengguna (Lanjutan)**

No.	Klasifikasi Pengguna	Deskripsi Pengguna
3	Petugas	Petugas merupakan seseorang yang bekerja pada kantor desa Pelangwot, dalam hal ini petugas berkedudukan sebagai penyedia informasi dan operator sistem.
4	Kepala Desa	Kepala desa merupakan pengguna yang bertugas untuk menolak atau menyetujui apa yang menjadi masukan dari petugas.

### 4.3 Identifikasi Kebutuhan

Dengan didasarkan pada hasil identifikasi sistem dan pengguna, didapatkan beberapa kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional menggambarkan apa saja yang dapat dilakukan oleh pengguna, sedangkan kebutuhan non-fungsional merupakan kebutuhan yang secara tidak langsung harus ada pada sistem ini.

#### 4.3.1 Kebutuhan Fungsional

Kebutuhan pada sistem ini dikodekan dengan menggunakan format JAP\_F\_xxx. JAP merupakan singkatan dari nama sistem yang dibuat yaitu Jendela Administrasi Pelangwot, F merupakan jenis kebutuhan fungsional, sedangkan xxx merupakan nomor urut dari kebutuhan itu sendiri. Kebutuhan fungsional yang terdapat pada sistem termuat pada Tabel 4.2 berikut.

**Tabel 4.2 Kebutuhan Fungsional**

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_001	Register	Sistem harus dapat menyimpan data pengguna.	Guest
JAP_F_002	Login	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat masuk dengan otoritas tertentu.	
JAP_F_003	Logout	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melepas otoritas yang dimiliki.	Warga, Petugas, Kepala Desa
JAP_F_004	Melihat Profil Pengguna	Sistem harus dapat menampilkan profil atau data diri pengguna dengan otoritas warga.	Warga
JAP_F_005	Mengubah Profil Pengguna	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap data diri seperti email, nama nomor kartu keluarga dan nomor induk penduduk.	
JAP_F_006	Melihat Daftar Artikel	Sistem harus dapat menampilkan daftar artikel.	Guest, Warga, Petugas, Kepala Desa

**Tabel 4.2 Kebutuhan Fungsional (Lanjutan)**

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_007	Melihat Isi Artikel	Sistem harus dapat menampilkan salah satu isi artikel dari daftar artikel yang dipilih aktor	Guest, Warga, Petugas, Kepala Desa
JAP_F_008	Menambah Artikel	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah artikel baru.	Petugas
JAP_F_009	Menghapus Artikel	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu berita atau artikel.	
JAP_F_010	Mengubah Isi Artikel	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap isi suatu artikel.	
JAP_F_011	Melihat Profil Desa	Sistem harus dapat menampilkan profil desa berupa sejarah desa, geografi, sumber daya alam, sumber daya manusia, sumber daya pembangunan, dan sosial budaya.	Guest, Warga, Petugas, Kepala Desa
JAP_F_012	Melihat Visi Misi	Sistem harus dapat menampilkan visi misi desa.	
JAP_F_013	Mengubah Profil Desa	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap profil desa.	Petugas
JAP_F_014	Mengubah Visi Misi	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap visi misi desa.	
JAP_F_015	Melihat Profil Kades	Sistem harus dapat menampilkan daftar kepala desa berserta profilnya.	Guest, Warga, Petugas, Kepala Desa
JAP_F_016	Melihat Profil Sekdes	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melihat profil sekretaris desa.	
JAP_F_017	Melihat Profil Perangkat	Sistem harus dapat menampilkan daftar anggota perangkat desa.	
JAP_F_018	Melihat Profil BPD	Sistem harus dapat menampilkan daftar anggota badan permusyawaratan desa.	
JAP_F_019	Menambah Profil Kades	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil baru ke dalam daftar kepala desa.	Petugas

**Tabel 4.2 Kebutuhan Fungsional (Lanjutan)**

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_020	Menambah Profil Sekdes	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil baru ke dalam daftar sekretaris desa.	Petugas
JAP_F_021	Menambah Profil Perangkat	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil baru ke dalam daftar Perangkat desa.	
JAP_F_022	Menambah Profil BPD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil baru ke dalam daftar badan permusyawaratan desa.	
JAP_F_023	Menghapus Profil Kades	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil kepala desa.	
JAP_F_024	Menghapus Profil Sekdes	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil sekretaris desa.	
JAP_F_025	Menghapus Profil Perangkat	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil perangkat desa.	
JAP_F_026	Menghapus Profil BPD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil badan permusyawaratan desa.	
JAP_F_027	Mengubah Profil Kades	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap profil kepada desa.	
JAP_F_028	Mengubah Profil Sekdes	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap profil sekretaris desa.	
JAP_F_029	Mengubah Profil Perangkat	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melakukan perubahan terhadap profil perangkat desa.	

**Tabel 4.2 Kebutuhan Fungsional (Lanjutan)**

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_030	Mengubah Profil BPD	Sistem harus dapat menyediakan fasilitas perubahan data profil badan permusyawaratan desa	Petugas
JAP_F_031	Melihat Profil Kartar	Sistem harus dapat menampilkan daftar anggota karang taruna yang ada.	Guest, Warga, Petugas, Kepala Desa
JAP_F_032	Melihat Profil LPMD	Sistem harus dapat menampilkan daftar anggota lembaga permusyawaratan desa.	
JAP_F_033	Melihat Profil PKK	Sistem harus dapat menampilkan daftar anggota lembaga pembinaan kesejahteraan keluarga.	
JAP_F_034	Melihat Profil RTRW	Sistem harus dapat menampilkan daftar anggota RTRW.	
JAP_F_035	Menambah Profil Kartar	Sistem harus dapat menampilkan daftar anggota karang taruna.	
JAP_F_036	Menambah Profil LPMD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil anggota LPMD.	Petugas
JAP_F_037	Menambah Profil PKK	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil anggota PKK.	
JAP_F_038	Menambah Profil RTRW	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah profil anggota RTRW.	
JAP_F_039	Menghapus Profil KARTAR	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil anggota karang taruna.	
JAP_F_040	Menghapus Profil LPMD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil anggota LPMD.	
JAP_F_041	Menghapus Profil PKK	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil anggota PKK.	
JAP_F_042	Menghapus Profil RTRW	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu profil anggota RTRW.	

Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_043	Mengubah Profil Kartar	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu profil anggota karang taruna.	Petugas
JAP_F_044	Mengubah Profil LPMD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu profil anggota LPMD.	
JAP_F_045	Mengubah Profil PKK	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu profil anggota PKK.	
JAP_F_046	Mengubah Profil RTRW	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu profil anggota RTRW.	
JAP_F_047	Melihat Data APBD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melihat data APBD.	Guest, Warga, Petugas, Kepala Desa
JAP_F_048	Melihat Data RKPD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat melihat data RKPD.	
JAP_F_049	Menambah Data APBD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah data APBD	Petugas
JAP_F_050	Menambah Data RKPD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah data RKPD.	
JAP_F_051	Menghapus Data APBD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu data APBD.	
JAP_F_052	Menghapus Data RKPD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu data RKPD.	
JAP_F_053	Mengubah Data APBD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu data APBD	
JAP_F_054	Mengubah Data RKPD	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu data RKPD.	

**Tabel 4.2 Kebutuhan Fungsional (Lanjutan)**

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_055	Memvalidasi Data APBD	Sistem harus menyediakan fasilitas untuk memvalidasi data APBD yang telah dimasukkan oleh petugas.	Kepala Desa
JAP_F_056	Memvalidasi Data RKPD	Sistem harus menyediakan fasilitas untuk memvalidasi data <i>RKPD</i> yang telah dimasukkan oleh petugas.	
JAP_F_057	Melihat Pesan Aspirasi	Sistem harus dapat menampilkan daftar pesan aspirasi.	Guest, Warga, Petugas, Kepala Desa
JAP_F_058	Mengirim Aspirasi	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengirim pesan aspirasi.	Warga, Petugas, Kepala Desa
JAP_F_059	Menanggapi Aspirasi	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengirim balasan atau menanggapi salah satu pesan aspirasi.	
JAP_F_060	Menghapus Aspirasi	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu pesan aspirasi.	
JAP_F_061	Memvalidasi Pesan Aspirasi	Sistem harus menyediakan fasilitas bagi pengguna untuk memvalidasi pesan aspirasi yang telah dikirim oleh masyarakat.	Petugas, Kepala Desa
JAP_F_062	Menghapus Tanggapan Aspirasi	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu tanggapan aspirasi.	Warga, Petugas, Kepala Desa
JAP_F_063	Melihat Data Penduduk	Sistem harus dapat menampilkan data kependudukan.	Petugas, Kepala Desa
JAP_F_064	Menambah Data Penduduk	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menambah data penduduk.	Petugas
JAP_F_065	Menghapus Data Penduduk	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menghapus salah satu data penduduk.	
JAP_F_066	Mengubah Data Penduduk	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah salah satu data penduduk.	
JAP_F_067	Memesan Surat	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat memesan surat dalam berbagai jenis.	Warga

**Tabel 4.2 Kebutuhan Fungsional (Lanjutan)**

Kode	Use Case	Deskripsi Kebutuhan	Aktor
JAP_F_068	Melihat Status Pesanan	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengetahui status dari pesanan yang ia miliki dengan menggunakan nomor surat atau nomor pemesanan.	Warga
JAP_F_069	Melihat Daftar pesanan	Sistem harus dapat menampilkan daftar pesanan surat yang telah dipesan warga.	Petugas, Kepala Desa
JAP_F_070	Mengubah Status Pesanan	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengubah status salah satu pesanan surat.	Petugas
JAP_F_071	Mencetak surat	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mencetak surat dalam bentuk <i>Hard Copy</i> .	Petugas
JAP_F_072	Menyimpan surat	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat menyimpan surat dalam bentuk file <i>PDF</i> .	Petugas
JAP_F_073	Mengirim pesan notifikasi	Sistem harus dapat menyediakan fasilitas bagi pengguna agar dapat mengirim pesan notifikasi kepada pemesan surat dengan menggunakan layanan <i>SMS gateway</i> .	Petugas

#### 4.3.2 Kebutuhan Non Fungsional

Tabel 4.3 Berikut ini merupakan kebutuhan yang secara tidak langsung dibutuhkan sebagai pendukung sistem.

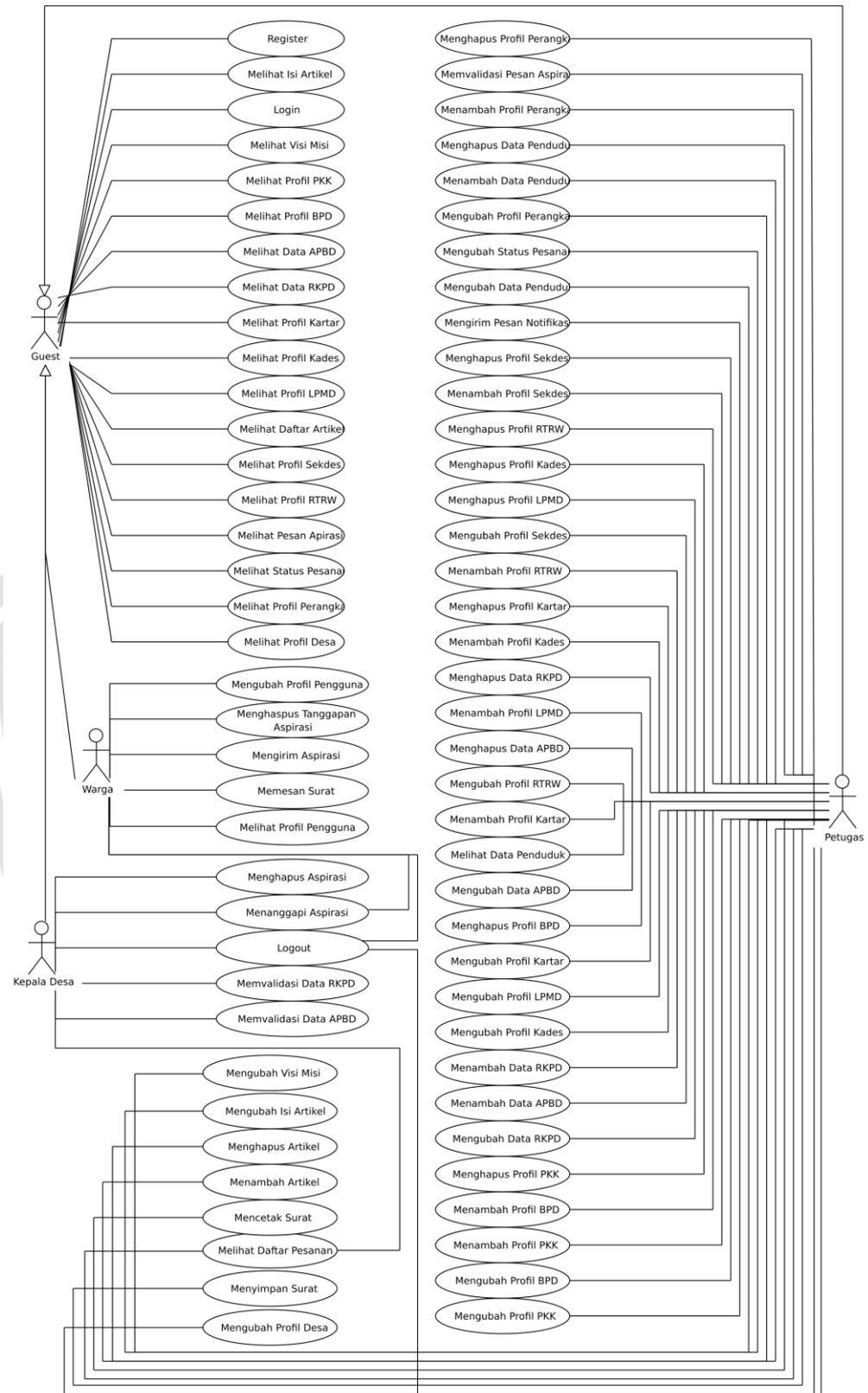
**Tabel 4.3 Kebutuhan non fungsional**

Kode	Nama Fungsi	Deskripsi Kebutuhan
JAP_NF_01	<i>Compatibility</i>	Sistem dapat berjalan di web <i>browser</i> seperti Mozilla Firefox, Google Chrome, Microsoft Edge, serta dapat berjalan pada perangkat <i>mobile</i> .

#### 4.4 Pemodelan Use case

Pemodelan *Use case* dibuat dengan mengacu pada kebutuhan yang telah dispesifikasikan sebelumnya, pemodelan dilakukan dengan membuat *Use case* diagram dan *Use case* skenario. *Use case* dimaksudkan untuk menggambarkan sistem, lingkungan sistem, serta interaksi yang terjadi antara aktor dengan sistem itu sendiri, diagram ini ditampilkan pada Gambar 4.1.

### 4.4.1 Diagram use case



Gambar 4.1 Use case diagram



#### 4.4.2 Skenario *use case*

Tabel 4.4 berikut merupakan skenario untuk mendaftarkan diri sebagai pengguna sistem dengan otoritas warga, sedangkan Tabel 4.5 merupakan skenario untuk masuk ke dalam sistem dengan otoritas tertentu yaitu warga, petugas, atau kepala desa.

**Tabel 4.4 Skenario *use case register***

<b>Register</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk mendaftarkan akun baru yang agar dapat menggunakan sistem dengan otoritas masyarakat
Aktor	<i>Guest</i> .
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu register yang ada sisi kanan atas <i>navigation bar</i>.</li> <li>2. Sistem menampilkan halaman <i>register</i>.</li> <li>3. Aktor memasukkan <i>username</i>, email, dan <i>password</i>.</li> <li>4. Aktor menekan tombol <i>register</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika aktor memasukkan <i>username</i> dengan jumlah karakter kurang dari 8 karakter, maka sistem menampilkan pesan kesalahan bahwa bidang <i>username</i> setidaknya harus berjumlah sebanyak 8 karakter.</li> <li>2. Jika aktor memasukkan email yang telah didaftarkan sebelumnya, maka sistem menampilkan pesan bahwa email telah digunakan.</li> <li>3. Jika aktor tidak mengisi bidang <i>password</i>, maka sistem menampilkan pesan bahwa bidang <i>password</i> tidak boleh kosong.</li> </ol>
<i>Post-Condition</i>	Sistem menyimpan data pendaftaran, dan mengirim kode aktiva si melalui email.

**Tabel 4.5 Skenario *use case login***

<b>Login</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk masuk ke dalam sistem dengan otoritas tertentu.
Aktor	<i>Guest</i> .
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu <i>login</i> yang ada sisi kanan atas <i>navigation bar</i>.</li> <li>2. Sistem menampilkan halaman <i>login</i>.</li> <li>3. Aktor memasukkan <i>username</i>, dan <i>password</i>.</li> <li>4. Aktor menekan tombol login</li> <li>5. Sistem melakukan pencocokan masukan <i>user</i> dengan data yang ada pada database.</li> </ol>

**Tabel 4.5 Skenario use case login (lanjutan)**

<b>Login</b>	
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika user memasukkan <i>username</i> dengan jumlah karakter kurang dari 8, maka sistem menampilkan pesan bahwa bidang <i>username</i> setidaknya harus berjumlah 8 karakter.</li> <li>2. Jika kombinasi <i>username</i> dan <i>password</i> tidak ditemukan dalam database, maka sistem menampilkan pesan bahwa kombinasi <i>username</i> dan <i>password</i> kurang tepat.</li> <li>3. Jika berhasil <i>login</i> dengan <i>level user</i> 0, maka sistem menampilkan halaman profil pengguna atau masyarakat.</li> <li>4. Jika berhasil <i>login</i> dengan <i>level user</i> 1, maka sistem menampilkan halaman daftar artikel sebagai petugas.</li> <li>5. Jika berhasil <i>login</i> dengan <i>level user</i> 2, maka sistem menampilkan halaman daftar artikel sebagai kepala desa.</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman sesuai dengan otoritas yang dimiliki aktor.

Tabel 4.6 merupakan skenario untuk melepas otoritas pengguna, sedangkan Tabel 4.7 merupakan skenario untuk melihat profil pengguna.

**Tabel 4.6 Skenario use case logout**

<b>logout</b>	
<i>Deskripsi</i>	<i>Use case</i> ini mendeskripsikan proses untuk melepas otoritas pengguna.
<i>Aktor</i>	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dengan otoritas yang dimiliki.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu <i>logout</i> yang terdapat pada sisi kanan <i>navigation bar</i>.</li> <li>2. Sistem menghapus sesi otoritas dari aktor.</li> </ol>
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan halaman utama <i>website</i> .

**Tabel 4.7 Skenario use case melihat profil pengguna**

<b>Melihat profil pengguna</b>	
<i>Deskripsi</i>	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan detail profil pengguna.
<i>Aktor</i>	Warga.
<i>Pre-Condition</i>	Aktor telah berhasil <i>login</i> dengan otoritas masyarakat.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil pengguna yang terdapat pada sisi kanan <i>navigation bar</i>.</li> <li>2. Sistem mengambil data dari <i>database</i> berdasarkan <i>username</i> aktor.</li> </ol>

**Tabel 4.7 Skenario *use case* melihat profil pengguna (Lanjutan)**

Melihat profil pengguna	
<i>Alternative Flow</i>	1. Jika aktor belum mengisi data seperti nama, nomor kartu keluarga dan nomor induk penduduk, maka sistem menampilkan halaman profil pengguna dalam bentuk <i>form</i> .
<i>Post-Condition</i>	Sistem menampilkan halaman profil pengguna.

Tabel 4.8 merupakan skenario untuk melakukan perubahan terhadap data diri pengguna, sedangkan Tabel 4.9 merupakan skenario untuk melihat daftar artikel.

**Tabel 4.8 Skenario *use case* mengubah profil pengguna**

Melihat profil pengguna	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk mengubah data diri atau profil pengguna
Aktor	Warga.
<i>Pre-Condition</i>	Aktor berada pada halaman profil pengguna.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu <i>edit</i> profil yang terdapat pada sisi kanan atas halaman.</li> <li>2. Sistem mengambil data dari <i>database</i> berdasarkan <i>username</i> aktor.</li> <li>3. Sistem menampilkan halaman <i>edit</i> profil pengguna,</li> <li>4. Aktor melakukan perubahan pada <i>form</i> yang diri dari bidang nama, nomor kartu keluarga, dan nomor induk penduduk.</li> <li>5. Aktor menekan tombol simpan.</li> </ol>
<i>Alternative Flow</i>	1. Aktor menekan tombol simpan tanpa melakukan perubahan pada <i>form</i> , sistem menampilkan halaman profil pengguna tanpa melakukan perubahan data yang ada pada <i>database</i> .
<i>Post-Condition</i>	Sistem melakukan perubahan data pada <i>database</i> dan menampilkan halaman profil pengguna.

**Tabel 4.9 Skenario *use case* melihat daftar artikel**

Melihat daftar artikel	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman daftar berita.
Aktor	<i>Guest</i> , Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	1. Sistem mengambil data artikel dari <i>database</i> .
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan 404 <i>error</i>.</li> <li>2. Jika data tidak ditemukan, sistem menampilkan halaman dengan pesan "<i>No data available</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan daftar artikel

Tabel 4.10 merupakan skenario *use case* untuk melihat detail atau isi dari artikel, Tabel 4.11 adalah skenario untuk menambah artikel baru, sedangkan Tabel 4.12 merupakan skenario untuk melakukan perubahan terhadap isi dari suatu artikel.

**Tabel 4.10 Skenario *use case* melihat isi artikel**

Melihat isi artikel	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan detail atau isi dari suatu artikel yang ada pada daftar artikel.
Aktor	Guest, Warga, Petugas, Kepala Desa.
Pre-Condition	Aktor berada pada halaman daftar berita atau artikel
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>read more</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
Post-Condition	Sistem menampilkan halaman detail artikel

**Tabel 4.11 Skenario *use case* menambah artikel**

Menambah artikel	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses <i>posting</i> berita atau artikel baru.
Aktor	Petugas.
Pre-Condition	Aktor telah login dan berada pada halaman daftar artikel.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>New posting</i>.</li> <li>2. Sistem menampilkan <i>form</i> dengan <i>field</i> tertentu.</li> <li>3. Aktor <i>mengisi</i> form.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. <i>Form</i> kosong, sistem menampilkan pesan peringatan.</li> <li>2. Aktor menekan tombol batal, artikel tidak tersimpan, kembali ke halaman daftar artikel.</li> </ol>
Post-Condition	Artikel tersimpan ke dalam <i>database</i> , menampilkan pesan berhasil dan halaman daftar artikel.

**Tabel 4.12 Skenario *use case* mengubah artikel**

Mengubah artikel	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan artikel mengenai berita atau kegiatan warga
Aktor	Petugas

**Tabel 4.12 Skenario *use case* mengubah artikel (lanjutan)**

Mengubah artikel	
<i>Pre-Condition</i>	Petugas telah <i>login</i> dan berada pada halaman daftar artikel
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Petugas menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Petugas melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "<i>No. data available in tabel</i>".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman detail artikel.

Tabel 4.13 merupakan skenario untuk melihat profil desa, sedangkan Tabel 4.14 merupakan skenario untuk melihat visi misi desa.

**Tabel 4.13 Skenario *use case* melihat profil desa**

Melihat profil desa	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan artikel mengenai profil desa Pelangwot.
Aktor	<i>Guest</i> , Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil desa.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman profil desa.

**Tabel 4.14 Skenario *use case* melihat visi misi**

Melihat visi misi	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan visi misi.
Aktor	<i>Guest</i> , Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu visi misi.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> </ol>

**Tabel 4.14 Skenario use case melihat visi misi**

Melihat visi misi	
	2. Jika data tidak ditemukan dalam <i>database</i> , sistem menampilkan halaman dengan pesan “No data available in table”.
<i>Post-Condition</i>	Sistem menampilkan halaman visi misi.

Tabel 4.15 merupakan skenario untuk mengubah isi dari profil desa, sedangkan Tabel 4.16 merupakan skenario untuk melakukan perubahan terhadap visi misi desa.

**Tabel 4.15 Skenario use case mengubah profil desa**

Mengubah profil desa	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan artikel profil desa
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil desa.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan “No. data available in tabel”.</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman profil desa.

**Tabel 4.16 Skenario use case mengubah visi misi**

Mengubah visi misi	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan data visi misi.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman visi misi.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan “No. data available in tabel”.</li> </ol>

**Tabel 4.16 Skenario *use case* mengubah visi misi (lanjutan)**

Mengubah visi misi	
	2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman visi misi.

Tabel 4.17 merupakan skenario untuk melihat profil kepala desa, sedangkan Tabel 4.18 merupakan skenario untuk melihat profil sekretaris desa.

**Tabel 4.17 Skenario *use case* melihat profil *kades***

Melihat profil <i>kades</i>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman profil kepala desa.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil <i>kades</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan 404 <i>error</i>.</li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "No data available in table".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan profil <i>kades</i> .

**Tabel 4.18 Skenario *use case* melihat profil *sekdes***

Melihat profil <i>sekdes</i>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman profil sekretaris desa.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil <i>sekdes</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan 404 <i>error</i>.</li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "No data available in table".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman profil <i>sekdes</i> .

Tabel 4.19 merupakan skenario untuk melihat profil perangkat desa, Tabel 4.20 adalah skenario untuk melihat profil badan permusyawaratan desa, sedangkan Tabel 4.21 merupakan skenario untuk menambah profil kepala desa baru.

**Tabel 4.19 Skenario *use case* melihat profil perangkat desa**

Melihat profil perangkat desa	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman profil perangkat desa.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil perangkat desa.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman profil perangkat desa

**Tabel 4.20 Skenario *use case* melihat BPD**

Melihat BPD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman profil anggota BPD.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu profil BPD.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman profil BPD.

**Tabel 4.21 Skenario *use case* menambah profil Kades**

Menambah profil Kades	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam kepala desa
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil kepala desa.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> </ol>

**Tabel 4.21 Skenario *use case* menambah profil *Kades***

Menambah profil <i>Kades</i>	
	<ol style="list-style-type: none"> <li>3. Aktor <i>memasukkan</i> data profil.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang telah dimasukkan.</li> </ol>
<i>Alternative Flow</i>	1. Aktor memasukkan data dengan tidak sesuai format, sistem akan gagal melakukan validasi dan menampilkan pesan kesalahan pada form tambah.
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

Tabel 4.22 merupakan skenario untuk menambah profil sekretaris desa, sedangkan Tabel 4.23 merupakan skenario untuk menambah profil perangkat desa.

**Tabel 4.22 Skenario *use case* menambah profil *Sekdes***

Menambah profil <i>Sekdes</i>	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam sekretaris desa
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah login dan berada pada halaman profil sekretaris desa.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> <li>3. Aktor memasukkan data profil.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang telah dimasukkan.</li> </ol>
<i>Alternative Flow</i>	1. Aktor memasukkan data dengan tidak sesuai format, sistem akan gagal melakukan validasi dan menampilkan pesan kesalahan pada form tambah.
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

**Tabel 4.23 Skenario *use case* menambah profil anggota perangkat**

Menambah profil anggota perangkat	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam anggota perangkat desa.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah login dan berada pada halaman daftar anggota perangkat desa.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> </ol>

**Tabel 4.23 Skenario *use case* menambah profil anggota perangkat (Lanjutan)**

Menambah profil anggota perangkat	
	<ol style="list-style-type: none"> <li>3. Aktor memasukkan data profil perangkat.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>2. Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

Tabel 4.24 merupakan skenario untuk menambah profil BPD baru, sedangkan Tabel 4.25 merupakan skenario untuk melakukan perubahan terhadap profil kepala desa.

**Tabel 4.24 Skenario *use case* tambah BPD**

Tambah BPD	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam anggota BPD.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil BPD
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> <li>3. Aktor memasukkan data profil BPD.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>2. Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

**Tabel 4.25 Skenario *use case* mengubah profil *kades***

Mengubah profil <i>kades</i>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil kepala desa
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil <i>kades</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> </ol>

**Tabel 4.25 Skenario use case mengubah profil kades (Lanjutan)**

Mengubah profil kades	
	5. Sistem memvalidasi hasil perubahan data.
<i>Alternative Flow</i>	1. Data tidak ditemukan dalam <i>database</i> , sistem menampilkan pesan "No. data available in tabel". 2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman profil kades

Tabel 4.26 merupakan skenario untuk mengubah profil sekretaris desa, sedangkan Tabel 4.27 merupakan skenario untuk mengubah profil perangkat.

**Tabel 4.26 Skenario use case mengubah profil sekdes**

Mengubah Profil sekdes	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil sekretaris desa
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil sekdes
<i>Main Flow</i>	1. Aktor menekan tombol <i>edit</i> . 2. Sistem mengambil data dari <i>database</i> . 3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i> . 4. Aktor melakukan perubahan. 5. Sistem memvalidasi hasil perubahan data.
<i>Alternative Flow</i>	1. Data tidak ditemukan dalam <i>database</i> , sistem menampilkan pesan "No. data available in tabel". 2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman profil sekdes.

**Tabel 4.27 Skenario use case mengubah profil perangkat**

Mengubah profil perangkat	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan salah satu profil dari anggota perangkat desa
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar perangkat desa.
<i>Main Flow</i>	1. Aktor menekan tombol <i>edit</i> pada salah satu profil perangkat desa. 2. Sistem mengambil data dari <i>database</i> . 3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i> . 4. Aktor melakukan perubahan.

**Tabel 4.27 Skenario *use case* mengubah profil perangkat (Lanjutan)**

Mengubah profil perangkat	
	5. Sistem memvalidasi hasil perubahan data.
<i>Alternative Flow</i>	1. Data tidak ditemukan dalam <i>database</i> , sistem menampilkan pesan “No. data available in tabel”. 2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil.

Tabel 4.28 merupakan skenario untuk mengubah profil anggota badan permusyawaratan desa, sedangkan Tabel 4.29 merupakan skenario untuk menghapus salah satu profil kepada desa.

**Tabel 4.28 Skenario *use case* mengubah profil BPD**

Mengubah profil BPD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil pada salah satu anggota BPD
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil BPD
<i>Main Flow</i>	1. Aktor menekan tombol edit pada salah satu profil anggota BPD. 2. Sistem mengambil data dari <i>database</i> . 3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i> 4. Aktor melakukan perubahan. 5. Sistem memvalidasi hasil perubahan data.
<i>Alternative Flow</i>	1. Data tidak ditemukan dalam <i>database</i> , sistem menampilkan pesan “No. data available in tabel”. 2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil

**Tabel 4.29 Skenario *use case* menghapus profil *kades***

Menghapus profil <i>kades</i>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan profil kepala desa yang memiliki status non aktif.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman profil <i>kades</i> desa.
<i>Main Flow</i>	1. Aktor menekan tombol hapus pada salah satu profil. 2. Sistem menampilkan pesan konfirmasi.
<i>Alternative Flow</i>	1. Aktor memilih tombol Ya, data berhasil dihapus. 2. Aktor memilih tombol Tidak, data tidak dihapus

**Tabel 4.29 Skenario use case menghapus profil kades (Lanjutan)**

Menghapus profil kades	
Post-Condition	Data terhapus, sistem menampilkan pesan berhasil

Tabel 4.30 merupakan skenario untuk menghapus salah satu profil anggota sekretaris desa, Tabel 4.31 untuk menghapus salah satu anggota perangkat, Sedangkan Tabel 4.32 merupakan skenario untuk menghapus profil anggota BPD.

**Tabel 4.30 Skenario use case menghapus profil sekdes**

Menghapus profil sekdes	
Deskripsi	Use case ini mendeskripsikan proses penghapusan profil sekretaris desa yang memiliki status non aktif.
Aktor	Petugas
Pre-Condition	Aktor telah <i>login</i> dan berada pada halaman profil kades desa.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu profil.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus</li> </ol>
Post-Condition	Data terhapus, sistem menampilkan pesan berhasil

**Tabel 4.31 Skenario use case menghapus profil perangkat**

Menghapus profil perangkat	
Deskripsi	Use case ini mendeskripsikan proses penghapusan profil dari anggota perangkat desa.
Aktor	Petugas
Pre-Condition	Aktor telah <i>login</i> dan berada pada halaman profil perangkat desa.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu profil.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>3. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>4. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
Post-Condition	Data terhapus, sistem menampilkan pesan berhasil

**Tabel 4.32 Skenario use case menghapus profil BPD**

Menghapus profil BPD	
Deskripsi	Use case ini mendeskripsikan proses penghapusan profil anggota BPD.
Aktor	Petugas
Pre-Condition	Aktor telah <i>login</i> dan berada pada halaman profil BPD.

**Tabel 4.32 Skenario *use case* menghapus profil BPD (Lanjutan)**

Menghapus profil BPD	
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu anggota PBD.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

Tabel 4.33 merupakan skenario untuk melihat daftar anggota LPMD, sedangkan Tabel 4.34 merupakan skenario untuk melihat daftar anggota PKK.

**Tabel 4.33 Skenario *use case* melihat anggota LPMD**

Melihat anggota LPMD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman daftar anggota lembaga pemberdayaan masyarakat desa.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lembaga LPMD.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error..</i></li> <li>2. jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman LPMD.

**Tabel 4.34 Skenario *use case* melihat anggota PKK**

Melihat anggota PKK	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman daftar anggota pembinaan kesejahteraan keluarga.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lembaga PKK.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error..</i></li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman yang berisi daftar anggota PKK.



Tabel 4.35 merupakan skenario untuk melihat daftar anggota karang taruna, Tabel 4.36 untuk melihat daftar anggota RTRW, sedangkan Tabel 4.37 merupakan skenario untuk menambah profil baru anggota LPMD.

**Tabel 4.35 Skenario use case melihat anggota Karang Taruna**

Melihat anggota Karang Taruna	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman daftar anggota karang taruna.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lembaga Karang taruna.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman yang berisi daftar karang taruna.

**Tabel 4.36 Skenario use case melihat anggota RTRW**

Melihat anggota RTRW.	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan lembaga RTRW.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu lembaga RTRW.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan dalam <i>database</i>, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman lembaga RTRW.

**Tabel 4.37 Skenario use case tambah LPMD**

Tambah LPMD	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam anggota LPMD
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar anggota lembaga pemberdayaan masyarakat desa.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> </ol>

**Tabel 4.37 Skenario *use case* tambah LPMD (Lanjutan)**

Tambah LPMD	
	<ol style="list-style-type: none"> <li>3. Aktor memasukkan data profil LPMD.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>2. Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

Tabel 4.38 merupakan skenario untuk menambah profil baru anggota PKK, sedangkan Tabel 4.39 merupakan skenario untuk menambah profil baru anggota karang taruna.

**Tabel 4.38 Skenario *use case* tambah anggota PKK**

Tambah anggota PKK	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam anggota PKK
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar anggota pembinaan kesejahteraan keluarga.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> <li>3. Aktor memasukkan data profil PKK.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>2. Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

**Tabel 4.39 Skenario *use case* menambah profil anggota Kartar**

Menambah profil anggota Kartar	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam anggota karang taruna
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar anggota Karang taruna.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> </ol>

**Tabel 4.39 Skenario *use case* menambah profil anggota Kartar (Lanjutan)**

Menambah profil anggota Kartar	
	<ol style="list-style-type: none"> <li>Aktor memasukkan data profil Karang taruna.</li> <li>Aktor menekan tombol <i>save</i>.</li> <li>Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

Tabel 4.40 merupakan skenario untuk menambah profil baru anggota RTRW, sedangkan Tabel 4.41 merupakan skenario untuk menghapus salah satu anggota LPMD.

**Tabel 4.40 Skenario *use case* tambah RT/RW**

Tambah RT/RW	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam anggota lembaga PKK
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar anggota PKK.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>Aktor menekan tombol tambah.</li> <li>Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> <li>Aktor memasukkan data profil PKK.</li> <li>Aktor menekan tombol <i>save</i>.</li> <li>Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

**Tabel 4.41 Skenario *use case* menghapus profil anggota LPMD**

Menghapus profil anggota LPMD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan profil anggota Lembaga pemberdayaan masyarakat desa
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman LPMD
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>Aktor menekan tombol hapus pada salah satu anggota LPMD</li> <li>Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>Aktor memilih tombol Ya, data berhasil dihapus.</li> </ol>

**Tabel 4.41 Skenario *use case* menghapus profil anggota LPMD (lanjutan)**

Menghapus profil anggota LPMD	
	2. Aktor memilih tombol Tidak, data tidak dihapus.
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

Tabel 4.42 merupakan skenario untuk menghapus salah satu anggota PKK, Tabel 4.43 untuk menghapus salah satu anggota karang taruna, sedangkan Tabel 4.44 merupakan skenario untuk menghapus salah satu anggota RTRW.

**Tabel 4.42 Skenario *use case* menghapus profil anggota PKK**

Menghapus profil anggota PKK	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan profil anggota PKK.
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman PKK
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu anggota PKK.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

**Tabel 4.43 Skenario *use case* menghapus profil anggota Kartar**

Menghapus profil anggota kartar	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan profil anggota Karang Taruna.
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman Karang taruna.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu anggota Karang taruna.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

**Tabel 4.44 Skenario *use case* menghapus profil anggota RTRW**

Menghapus profil anggota RTRW	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan profil anggota RTRW
<i>Aktor</i>	Petugas

**Tabel 4.44 Skenario *use case* menghapus profil anggota RTRW (lanjutan)**

Menghapus profil anggota RTRW	
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar lembaga PKK
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu profil anggota PKK</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

Tabel 4.45 merupakan skenario untuk melakukan perubahan terhadap salah satu profil anggota LPMD, sedangkan Tabel 4.46 untuk mengubah salah satu profil anggota PKK.

**Tabel 4.45 Skenario *use case* mengubah profil anggota LPMD**

Mengubah profil LPMD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil salah satu anggota LPMD
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman LPMD
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i> pada salah satu anggota lembaga pemberdayaan masyarakat desa.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "No. data available in tabel".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil

**Tabel 4.46 Skenario *use case* mengubah profil anggota PKK**

Mengubah profil LPMD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil salah satu anggota PKK
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman PKK
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol edit pada salah satu profil anggota PKK.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>



**Tabel 4.46 Skenario *use case* mengubah profil anggota PKK (lanjutan)**

Mengubah profil LPMD	
	<ol style="list-style-type: none"> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "<i>No. data available in tabel</i>".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil.

Tabel 4.47 merupakan skenario untuk melakukan perubahan terhadap salah satu profil anggota karang taruna, sedangkan Tabel 4.48 untuk anggota RTRW.

**Tabel 4.47 Skenario *use case* mengubah profil anggota kartar**

Mengubah profil anggota kartar	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil salah satu anggota karang taruna.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman Karang taruna.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol edit pada salah satu profil anggota Karang taruna.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "<i>No. data available in tabel</i>".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil.

**Tabel 4.48 Skenario *use case* mengubah profil anggota RTRW**

Mengubah profil anggota RTRW	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan pada salah satu profil anggota RTRW
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman lembaga RTRW
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i> pada salah satu profil anggota RTRW.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>

**Tabel 4.48 Skenario *use case* mengubah profil anggota RTRW (lanjutan)**

<b>Mengubah profil anggota RTRW</b>	
	<ol style="list-style-type: none"> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "<i>No. data available in tabel</i>".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil

Tabel 4.49 merupakan skenario untuk melihat data APBD, sedangkan Tabel 4.50 merupakan skenario untuk melihat data RKPD.

**Tabel 4.49 Skenario *use case* melihat data APBD**

<b>Melihat data APBD</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman berisi data APBD
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu Data APBD.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan 404 <i>error</i>.</li> <li>2. Jika data tidak ditemukan, sistem menampilkan halaman dengan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman data APBD

**Tabel 4.50 Skenario *use case* melihat data RKPD**

<b>Melihat data RKPD</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman yang berisi data RKPD.
Aktor	Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu data RKPD.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan 404 <i>error</i>.</li> <li>2. Jika data tidak ditemukan, sistem menampilkan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman data RKPD

Tabel 4.51 merupakan skenario untuk menambah data APBD, Tabel 4.52 untuk menambah data RKPD, sedangkan Tabel 4.53 merupakan skenario untuk melakukan perubahan terhadap salah satu data APBD.

**Tabel 4.51 Skenario *use case* tambah data APBD**

Tambah data APBD	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan data APBD
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman data APBD.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman dengan <i>field</i> tertentu.</li> <li>3. Aktor memasukkan data APBD baru.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>2. Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

**Tabel 4.52 Skenario *use case* tambah data RKPD**

Tambah data RKPD	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan data rupa
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman data rupa.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman dengan <i>field</i> tertentu.</li> <li>3. Aktor memasukkan data rupa baru.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> <li>2. Terdapat data yang sama pada <i>database</i>, penyimpanan gagal.</li> </ol>
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

**Tabel 4.53 Skenario *use case* mengubah data APBD**

Mengubah data APBD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan data APBD
Aktor	Petugas

**Tabel 4.53 Skenario *use case* mengubah data APBD (Lanjutan)**

Mengubah data APBD	
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman APBD
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan</li> <li>5. Sistem memvalidasi hasil perubahan data</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "<i>No. data available in tabel</i>"</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman data APBD

Tabel 4.54 merupakan skenario untuk melakukan perubahan terhadap salah satu data RKPD, sedangkan Tabel 4.55 merupakan skenario untuk menghapus salah satu data APBD.

**Tabel 4.54 Skenario *use case* mengubah data RKPD**

Mengubah data RKPD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan data RKPD
Aktor	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman RKPD
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di edit.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "<i>No. data available in tabel</i>".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil dan halaman data RKPD.

**Tabel 4.55 Skenario *use case* menghapus data APBD**

Menghapus data APBD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menghapus salah satu bagian data APBD

**Tabel 4.55 Skenario *use case* menghapus data APBD (Lanjutan)**

<b>Menghapus data APBD</b>	
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman data APBD.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu data APBD.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

Tabel 4.56 merupakan skenario untuk menghapus salah satu data RKPd, sedangkan Tabel 4.57 merupakan skenario untuk memvalidasi data APBD baru yang telah dimasukkan oleh petugas.

**Tabel 4.56 Skenario *use case* menghapus data RKPd**

<b>Menghapus data RKPd</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menghapus salah satu bagian data RKPd
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman data RKPd.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu data RKPd</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

**Tabel 4.57 Skenario *use case* memvalidasi data APBD**

<b>Memvalidasi data APBD</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses memvalidasi data APBD yang telah diajukan oleh petugas.
<i>Aktor</i>	Kepala Desa
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman data APBD
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu validasi</li> <li>2. Sistem menampilkan data APBD yang belum divalidasi.</li> <li>3. Aktor menekan tombol validasi.</li> <li>4. Aktor memilih untuk mengizinkan atau menolak.</li> <li>5. Aktor menekan tombol <i>save</i>.</li> </ol>

**Tabel 4.57** Skenario *use case* memvalidasi data APBD (Lanjutan)

Memvalidasi data APBD	
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih untuk mengizinkan pesan aspirasi, maka sistem mengubah status pesan menjadi “diterima”</li> <li>2. Aktor memilih untuk menolak pesan, maka sistem mengubah status pesanan menjadi “ditolak”</li> </ol>
<i>Post-Condition</i>	Status data berubah, sistem menampilkan halaman validasi data APBD.

Tabel 4.58 merupakan skenario untuk memvalidasi data RKPD, sedangkan Tabel 4.59 merupakan skenario untuk melihat daftar pesan aspirasi.

**Tabel 4.58** Skenario *use case* memvalidasi data RKPD

Memvalidasi data RKPD	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses memvalidasi data <i>RKPD</i> yang telah diajukan oleh petugas.
Aktor	Kepala Desa
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman data <i>RKPD</i>
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu validasi</li> <li>2. Sistem menampilkan data <i>RKPD</i> yang belum divalidasi.</li> <li>3. Aktor menekan tombol validasi</li> <li>4. Aktor memilih untuk mengizinkan atau menolak.</li> <li>5. Aktor menekan tombol <i>save</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih untuk mengizinkan pesan aspirasi, maka sistem mengubah status pesan menjadi “diterima”</li> <li>2. Aktor memilih untuk menolak pesan, maka sistem mengubah status pesanan menjadi “ditolak”</li> </ol>
<i>Post-Condition</i>	Status data berubah, sistem menampilkan halaman validasi data <i>RKPD</i> .

**Tabel 4.59** Skenario *use case* melihat pesan aspirasi

Melihat pesan aspirasi	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan pesan aspirasi yang disampaikan masyarakat.
Aktor	<i>Guest</i> , Warga, Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah Membuka <i>website</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu sambung warga.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan, sistem menampilkan halaman dengan pesan “<i>No data available in table</i>”.</li> </ol>

**Tabel 4.59 Skenario *use case* melihat pesan aspirasi (Lanjutan)**

<b>Melihat pesan aspirasi</b>	
<i>Post-Condition</i>	Sistem menampilkan halaman daftar aspirasi

Tabel 4.60 merupakan skenario untuk mengirim pesan aspirasi, Tabel 4.61 untuk menanggapi, sedangkan Tabel 4.62 merupakan skenario untuk menghapus pesan aspirasi.

**Tabel 4.60 Skenario *use case* mengirim aspirasi**

<b>Mengirim aspirasi</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk mengirim pesan aspirasi
Aktor	Warga
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman aspirasi
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memasukkan pesan pada form yang berada pada kiri halaman.</li> <li>2. Aktor menekan tombol kirim.</li> <li>3. Sistem memvalidasi masukan <i>user</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Masukan Kosong, sistem menampilkan peringatan</li> </ol>
<i>Post-Condition</i>	Pesan aspirasi berhasil disimpan ke dalam <i>database</i>

**Tabel 4.61 Skenario *use case* menanggapi aspirasi**

<b>Menanggapi aspirasi</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses membalas pesan aspirasi yang disampaikan oleh masyarakat
Aktor	Warga, Petugas, Kepala Desa
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar aspirasi
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol balas.</li> <li>2. Sistem menampilkan halaman baru dengan <i>form</i> balasan.</li> <li>3. Aktor memasukkan balasan.</li> <li>4. Aktor menekan tombol kirim.</li> <li>5. Sistem memvalidasi masukan <i>user</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Masukan Kosong, sistem menampilkan peringatan</li> </ol>
<i>Post-Condition</i>	Balasan berhasil disimpan ke dalam <i>database</i>

**Tabel 4.62 Skenario *use case* menghapus aspirasi**

<b>Menghapus aspirasi</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan pesan aspirasi yang disampaikan masyarakat

**Tabel 4.62 Skenario use case menghapus aspirasi (Lanjutan)**

<b>Menghapus aspirasi</b>	
<i>Aktor</i>	Warga, Petugas, Kepala Desa
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar aspirasi.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu pesan aspirasi.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

Tabel 4.63 merupakan skenario untuk menghapus salah satu pesan tanggapan aspirasi, sedangkan Tabel 4.64 merupakan skenario untuk memvalidasi pesan aspirasi.

**Tabel 4.63 Skenario use case menghapus tanggapan aspirasi**

<b>Menghapus tanggapan aspirasi</b>	
<i>Deskripsi</i>	<i>Use case</i> ini mendeskripsikan proses penghapusan tanggapan aspirasi yang disampaikan masyarakat
<i>Aktor</i>	Warga, Petugas, Masyarakat
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar aspirasi.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu balasan aspirasi yang telah ia kirim.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

**Tabel 4.64 Skenario use case memvalidasi pesan aspirasi**

<b>Memvalidasi pesan aspirasi</b>	
<i>Deskripsi</i>	<i>Use case</i> ini mendeskripsikan proses memvalidasi pesan aspirasi yang telah dikirim oleh masyarakat sebelum ditampilkan secara umum.
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada aspirasi
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu <i>validasi</i></li> <li>2. Sistem menampilkan daftar pesan yang belum divalidasi.</li> <li>3. Aktor menekan tombol <i>validasi</i>.</li> <li>4. Aktor memilih untuk mengizinkan atau menolak.</li> </ol>

**Tabel 4.64 Skenario *use case* memvalidasi pesan aspirasi (Lanjutan)**

Memvalidasi pesan aspirasi	
	5. Aktor menekan tombol <i>save</i> .
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih untuk mengizinkan pesan aspirasi, maka sistem mengubah status pesan menjadi 1.</li> <li>2. Aktor memilih untuk menolak pesan, maka sistem akan menghapus pesan tersebut dari sistem.</li> </ol>
<i>Post-Condition</i>	Status pesan aspirasi berubah, sistem menampilkan halaman validasi aspirasi.

Tabel 4.65 merupakan skenario untuk melihat data penduduk, sedangkan Tabel 4.66 merupakan skenario untuk menambah data penduduk baru.

**Tabel 4.65 Skenario *use case* melihat data penduduk**

Melihat data penduduk	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan halaman daftar anggota lembaga pemberdayaan masyarakat desa.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah membuka <i>website</i> dengan otoritas petugas.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu data penduduk.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i>.</li> <li>2. Jika data tidak ditemukan, sistem menampilkan pesan "<i>No data available in table</i>".</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan halaman lembaga pemberdayaan masyarakat desa

**Tabel 4.66 Skenario *use case* tambah data penduduk**

Tambah Data Penduduk	
Deskripsi	<i>Use case</i> mendeskripsikan proses penambahan profil baru ke dalam tabel kependudukan.
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah login dan berada pada halaman daftar penduduk.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol tambah.</li> <li>2. Sistem menampilkan halaman <i>form</i> dengan <i>field</i> tertentu.</li> <li>3. Aktor memasukkan data profil penduduk.</li> <li>4. Aktor menekan tombol <i>save</i>.</li> <li>5. Sistem memvalidasi data yang dimasukkan petugas.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Terdapat <i>field</i> yang kosong, sistem menampilkan peringatan.</li> </ol>

**Tabel 4.66 Skenario *use case* tambah data penduduk (Lanjutan)**

Tambah Data Penduduk	
	2. Terdapat data yang sama pada <i>database</i> , penyimpanan gagal.
<i>Post-Condition</i>	Data berhasil disimpan ke dalam <i>database</i> .

Tabel 4.67 merupakan skenario untuk menghapus salah satu data penduduk, sedangkan Tabel 4.68 merupakan skenario untuk melakukan perubahan terhadap salah satu data penduduk.

**Tabel 4.67 Skenario *use case* menghapus data penduduk**

Menghapus data Penduduk	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses penghapusan profil dari <i>database</i> kependudukan.
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar penduduk.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol hapus pada salah satu baris tabel.</li> <li>2. Sistem menampilkan pesan konfirmasi.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih tombol Ya, data berhasil dihapus.</li> <li>2. Aktor memilih tombol Tidak, data tidak dihapus.</li> </ol>
<i>Post-Condition</i>	Data terhapus, sistem menampilkan pesan berhasil.

**Tabel 4.68 Skenario *use case* mengubah data penduduk**

Mengubah data penduduk	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses perubahan profil penduduk
<i>Aktor</i>	Petugas
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman penduduk.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol <i>edit</i>.</li> <li>2. Sistem mengambil data dari <i>database</i>.</li> <li>3. Sistem menampilkan <i>form</i> dengan nilai dari data yang akan di <i>edit</i>.</li> <li>4. Aktor melakukan perubahan.</li> <li>5. Sistem memvalidasi hasil perubahan data.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Data tidak ditemukan dalam <i>database</i>, sistem menampilkan pesan "No. data available in tabel".</li> <li>2. Terdapat <i>field</i> yang kosong, sistem menampilkan pesan peringatan.</li> </ol>
<i>Post-Condition</i>	Data berhasil diubah, sistem menampilkan pesan berhasil

Tabel 4.69 merupakan skenario bagi warga untuk melakukan pemesanan surat, Tabel 6.70 untuk melihat status pesanan, sedangkan tabel 6.71 merupakan skenario untuk melihat daftar pesanan surat.

**Tabel 4.69 Skenario *use case* memesan surat**

<b>Memesan surat</b>	
Deskripsi	<i>Use case</i> mendeskripsikan proses untuk memesan surat
Aktor	Warga.
<i>Pre-Condition</i>	Aktor berada pada halaman daftar menu layanan persuratan
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih jenis surat yang akan dipesan.</li> <li>2. Sistem menampilkan halaman pemesanan.</li> <li>3. Aktor memasukkan data yang diperlukan.</li> <li>4. Sistem memvalidasi masukan.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Masukan kosong, sistem menampilkan peringatan.</li> </ol>
<i>Post-Condition</i>	Sistem menyimpan pesanan ke dalam <i>database</i> , dan menampilkan nomor surat atau nomor pemesanan.

**Tabel 4.70 Skenario *use case* melihat status pesanan**

<b>Melihat status pesanan</b>	
Deskripsi	<i>Use case</i> mendeskripsikan proses untuk melihat status pesanan
Aktor	Warga.
<i>Pre-Condition</i>	Aktor berada pada halaman layanan persuratan.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memasukkan nomor pemesanan.</li> <li>2. Aktor menekan tombol cek.</li> <li>3. Sistem memvalidasi masukan.</li> <li>4. Sistem mencari data pada <i>database</i>.</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Masukan kosong, sistem menampilkan peringatan.</li> <li>2. Data tidak ditemukan, sistem menampilkan pesan.</li> </ol>
<i>Post-Condition</i>	Sistem menampilkan status pesanan surat.

**Tabel 4.71 Skenario *use case* melihat daftar pesanan**

<b>Melihat daftar pesanan</b>	
Deskripsi	<i>Use case</i> ini mendeskripsikan proses untuk menampilkan daftar pesanan surat yang diajukan masyarakat.
Aktor	Petugas, Kepala Desa.
<i>Pre-Condition</i>	Aktor telah melakukan <i>autentikasi</i> .
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor memilih menu persuratan.</li> </ol>

**Tabel 4.71 Skenario use case melihat daftar pesanan (Lanjutan)**

Melihat daftar pesanan	
	2. Sistem mengambil data dari <i>database</i> .
<i>Alternative Flow</i>	1. Jika halaman tidak ditemukan, sistem menampilkan pesan <i>404 error</i> . 2. Jika data tidak ditemukan, sistem menampilkan pesan " <i>No. data available in table</i> "
<i>Post-Condition</i>	Sistem menampilkan daftar pesanan surat.

Tabel 4.72 berikut ini merupakan skenario untuk mencetak surat dengan menampilkannya ke dalam *windows* baru serta pilihan jenis cetak, sedangkan tabel 4.73 digunakan untuk menyimpan surat ke dalam *file* dengan *extensi pdf*.

**Tabel 4.72 Skenario use case mencetak surat**

Mencetak surat	
Deskripsi	<i>Use case</i> mendeskripsikan proses pencetakan pesanan surat yang telah diajukan masyarakat
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar pesanan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>print</i> . 2. Sistem menampilkan <i>layout printing</i> . 3. Aktor menekan tombol <i>print</i> . 4. Sistem mengubah status pesanan.
<i>Alternative Flow</i>	1. Aktor menekan tombol batal, kembali ke halaman <i>list_pesanan</i>
<i>Post-Condition</i>	Sistem mencetak dokumen surat, kembali ke halaman <i>list_pesanan</i>

**Tabel 4.73 Skenario use case menyimpan surat**

Menyimpan surat	
Deskripsi	<i>Use case</i> mendeskripsikan proses pencetakan menyimpan surat ke dalam bentuk <i>PDF</i> .
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar pesanan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>PDF</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menyimpan file dengan format pdf dan menggunakan nomor surat sebagai nama <i>file</i> .



Tabel 4.74 merupakan skenario untuk mengubah status dari pesan surat, sedangkan Tabel 4.75 merupakan skenario untuk pengiriman pesan notifikasi dengan menggunakan layanan *SMS Gateway*.

**Tabel 4.74 Skenario *use case* mengubah status pesanan**

Mengubah status pesanan	
Deskripsi	<i>Use case</i> mendeskripsikan proses untuk mengubah status pesanan
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor berada pada halaman persuratan.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol ubah status pada salah satu pesanan pada daftar pesanan.</li> <li>2. Sistem menampilkan <i>pop up</i> perubahan status</li> <li>3. aktor melakukan perubahan status</li> <li>4. aktor menekan tombol simpan</li> </ol>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol batal, sistem kembali menampilkan daftar pesanan tanpa melakukan perubahan data.</li> </ol>
<i>Post-Condition</i>	Sistem mengubah status pesanan pada <i>database</i> , dan menampilkannya pada tabel yang sesuai dengan status yang dimiliki saat ini.

**Tabel 4.75 Skenario *use case* mengirim notifikasi**

Mengirim notifikasi	
Deskripsi	<i>Use case</i> mendeskripsikan proses pengiriman notifikasi kepada pemesan surat bahwa pesannya telah selesai dengan menggunakan <i>sms gateway</i>
Aktor	Petugas.
<i>Pre-Condition</i>	Aktor telah <i>login</i> dan berada pada halaman daftar pesanan.
<i>Main Flow</i>	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol notifikasi pada salah satu pesan pada daftar pesanan.</li> <li>2. Sistem menampilkan halaman notifikasi dengan format pesan yang sesuai dengan jenis surat.</li> <li>3. Aktor menekan tombol kirim.</li> </ol>
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menyimpan pesan ke dalam <i>database</i> dan mengirimkannya dengan menggunakan <i>sms gateway</i> .

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

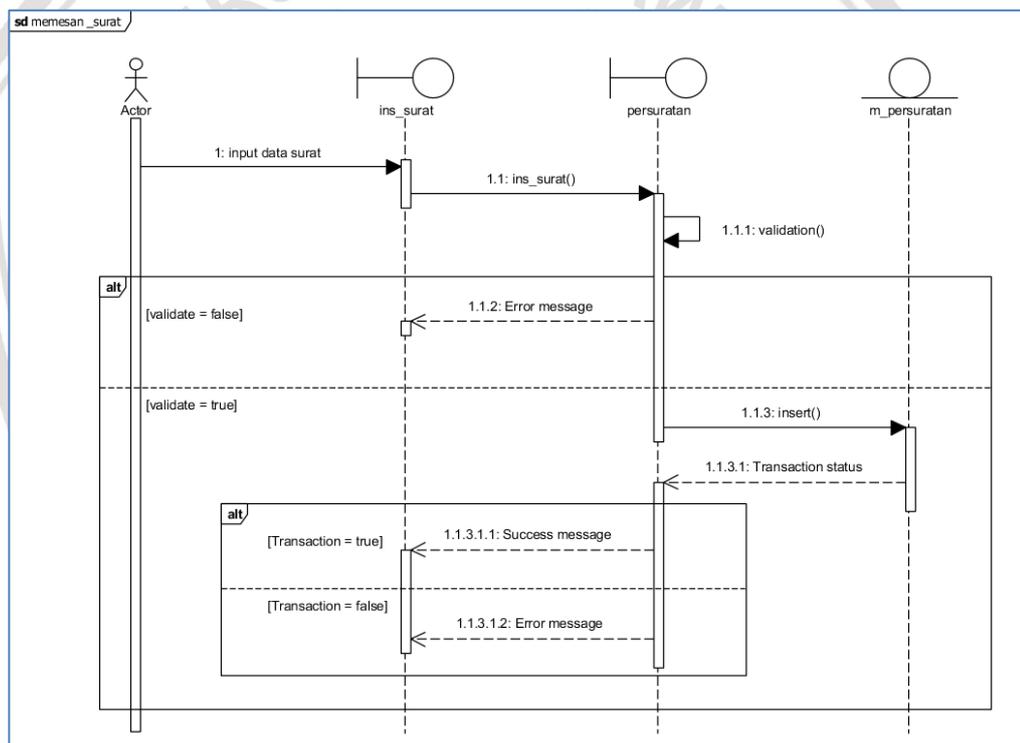
### 5.1 Perancangan

Perancangan sistem merupakan tahapan yang cukup penting, mengingat perancangan digunakan sebagai acuan implementasi selama pengembangan sistem berlangsung. Dalam pengembangan sistem ini, penulis membuat beberapa representasi perancangan antara lain *sequence diagram*, *physical data model*, *class diagram*, perancangan algoritme, dan perancangan antarmuka.

#### 5.1.1 Sequence Diagram

Pada tahap ini tiga sampel *use case* akan direpresentasikan ke dalam diagram *sequence*. Tiga sampel *use case* tersebut adalah *use case* memesan surat, *use case* melihat pesanan surat, dan edit visi misi. Diagram *sequence* sendiri digunakan untuk menunjukkan rangkaian pesan atau interaksi yang ada ada sistem.

##### 1. Memesan surat

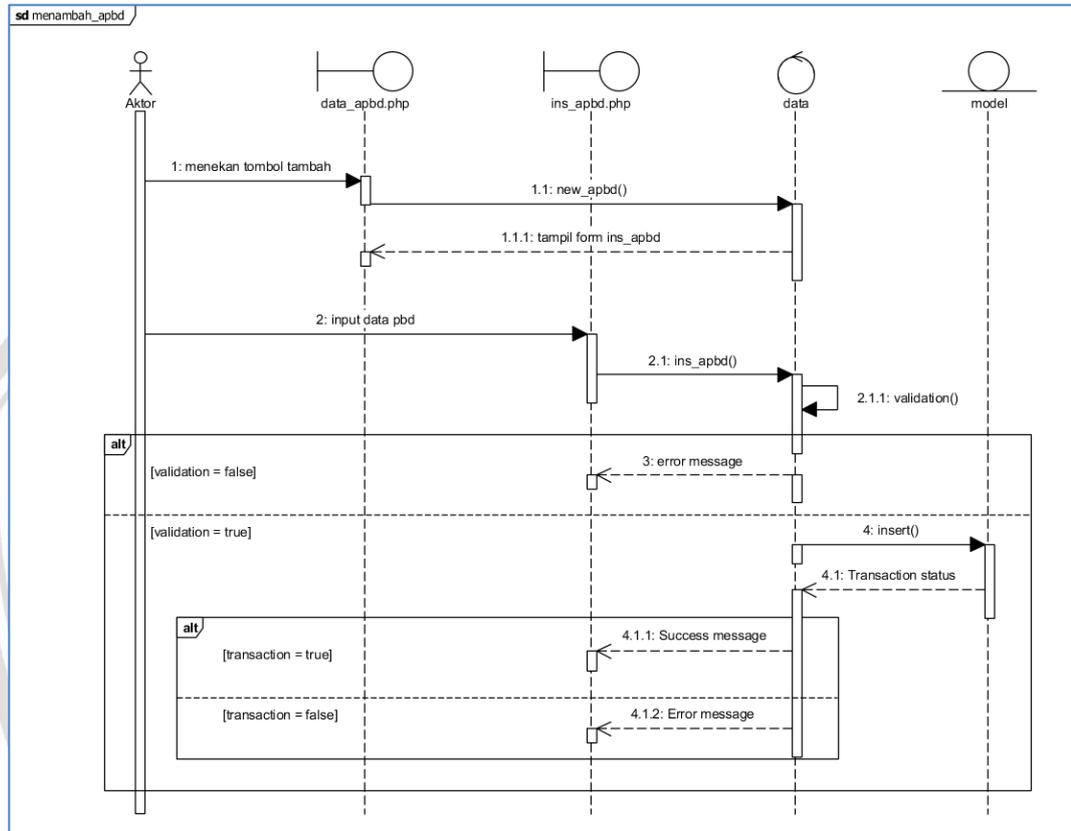


Gambar 5.1 Sequence diagram memesan surat

Gambar 5.1 menunjukkan *sequence* diagram untuk *Use case* memesan surat. Proses pemesanan surat dapat dilakukan ketika aktor telah melakukan autentikasi dan memilih jenis surat. Setelah aktor memasukkan data pada *field* yang tersedia, akan dilakukan verifikasi terhadap format data tersebut pada *method ins\_surat()* yang terdapat ada *controller class* persuratan.

Method *ins\_surat()* yang terdapat pada *controller* akan melakukan pemanggilan terhadap *method insert()* yang ada pada model jika ditemukan kesalahan data pada proses validasi maka sistem akan menampilkan pesan kesalahan pada form, jika tidak maka data disimpan ke dalam *database* . Jika data berhasil disimpan maka sistem akan menampilkan pesan berhasil dan nomor pesanan.

## 2. Menambah data APBD

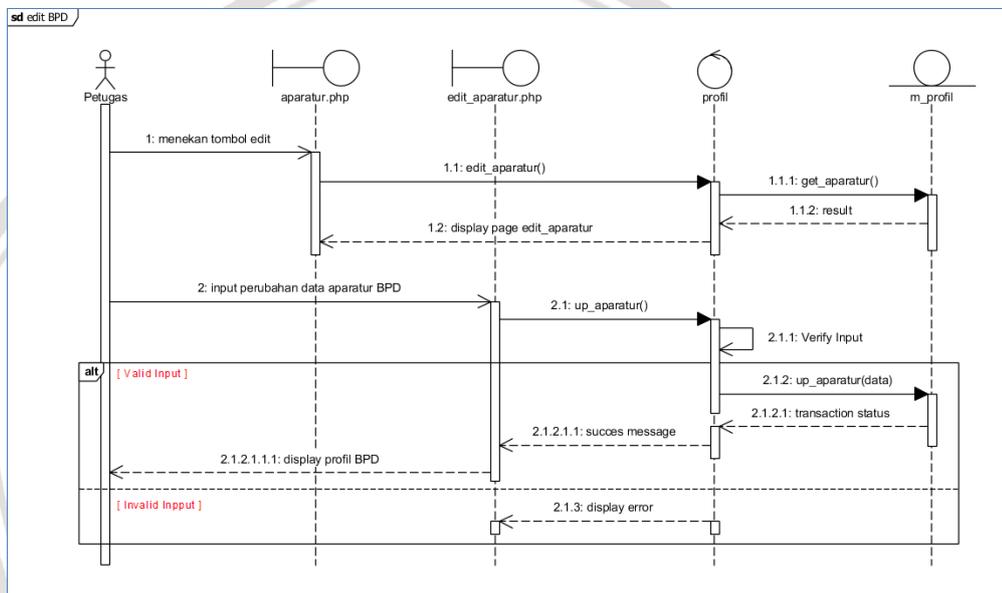


Gambar 5.2 Sequence diagram tambah data APBD

Gambar 5.2 menunjukkan *sequence diagram* untuk use case menambah data APBD. Aktor dalam use case ini adalah petugas, interaksi dimulai ketika petugas menekan tombol tambah dengan simbol *plus* pada halaman APBD. Sistem akan memanggil *method new\_apbd()* pada *controller* data yang digunakan untuk menampilkan *form* APBD. Setelah aktor memasukkan data APBD dan mengirimnya sistem akan melakukan pemanggil terhadap fungsi *ins\_apbd()* pada *controller* yang didalam-Nya terdapat proses validasi. Jika data telah benar maka sistem akan menyimpannya ke dalam *database* dengan memanggil fungsi *insert()* yang ada pada model, namun jika salah maka sistem akan kembali menampilkan *form* APBD dengan pesan kesalahan yang ada.

### 3. Mengubah profil BPD

Aktor dari *Use case* edit profil BPD adalah petugas, interaksi pada *use case* ini dimulai ketika petugas menekan tombol edit pada salah satu profil anggota BPD. Sistem akan memanggil fungsi *edit\_aparatur()* pada *controller* profil yang digunakan untuk menampilkan *form* edit, setelah aktor melakukan perubahan dan menekan tombol simpan maka sistem akan melakukan pemanggilan terhadap fungsi *up\_aparatur* yang di dalamnya terdapat proses validasi. Jika proses validasi berhasil sistem akan menyimpannya ke dalam *database* melalui fungsi *up\_aparatur()* pada model *m\_profil*. Namun jika pada saat validasi didapati kesalahan maka sistem akan kembali menampilkan *form* edit dengan pesan kesalahan yang ada. Sequence diagram *Use case* ini ditunjukkan pada Gambar 5.3 berikut

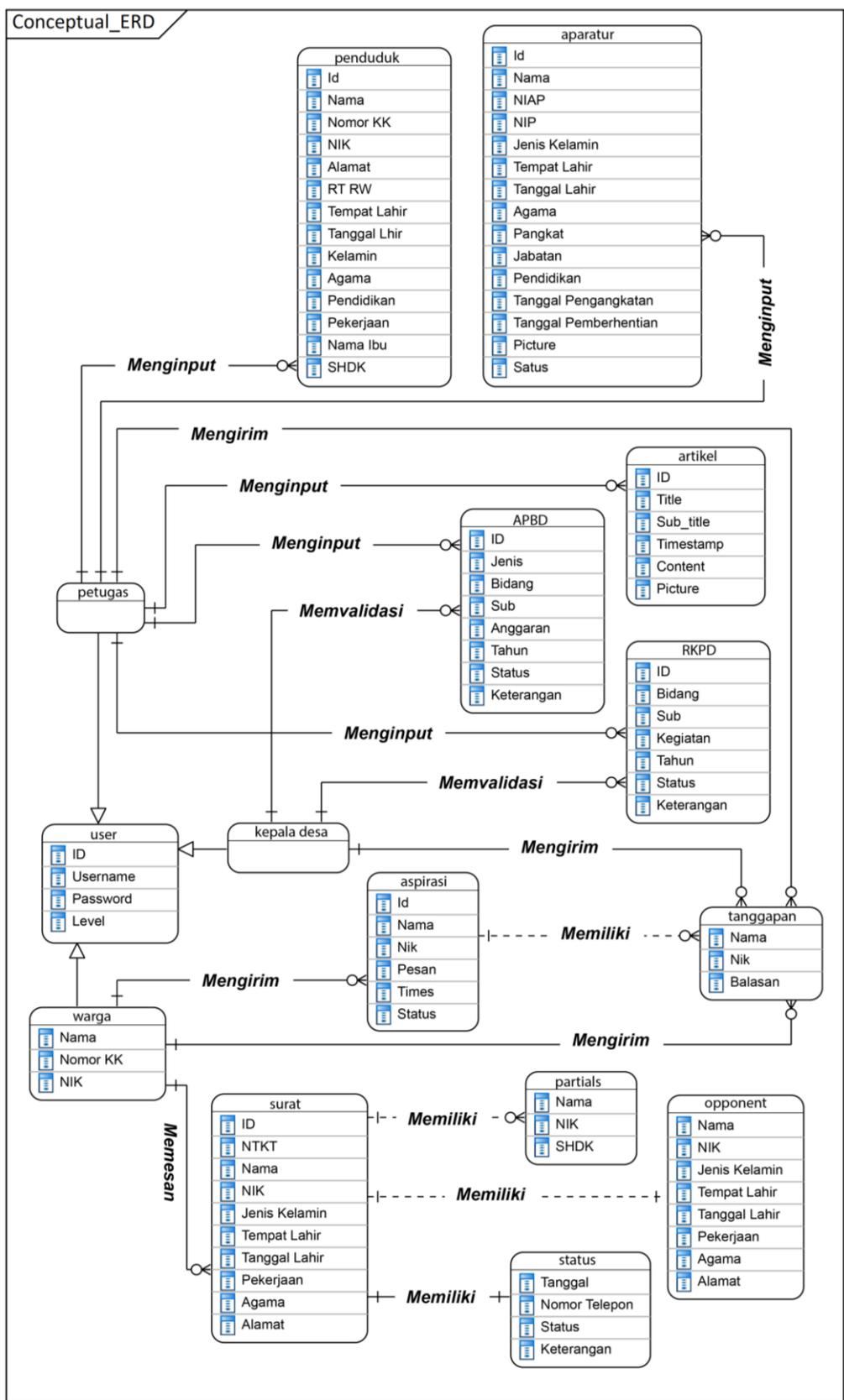


Gambar 5.3 Sequence diagram edit profil bpd

#### 5.1.2 Perancangan Basis Data

Perancangan basis data dikerjakan dengan menggunakan *conceptual data model* yang terdiri dari entitas, atribut, dan relasi. Entitas dari sistem ini terdiri dari *user*, *warga*, *petugas*, *kepala desa*, *surat*, *opponent*, *partials*, *status*, *aspirasi*, *tanggapan*, *APBD*, *RKDP*, *artikel*, *penduduk*, dan *aparatur*.

Entitas *warga*, *petugas*, dan *kepala desa* merupakan turunan dari entitas *user* yang memiliki atribut *Id*, *username*, *password*, dan *level*. Entitas *surat* memiliki relasi terhadap entitas *opponent*, *partials*, dan *status*. Sedangkan Entitas *aspirasi* memiliki relasi terhadap entitas *tanggapan*. Perancangan basis data pada pengembangan sistem ini ditunjukkan pada Gambar 5.4.



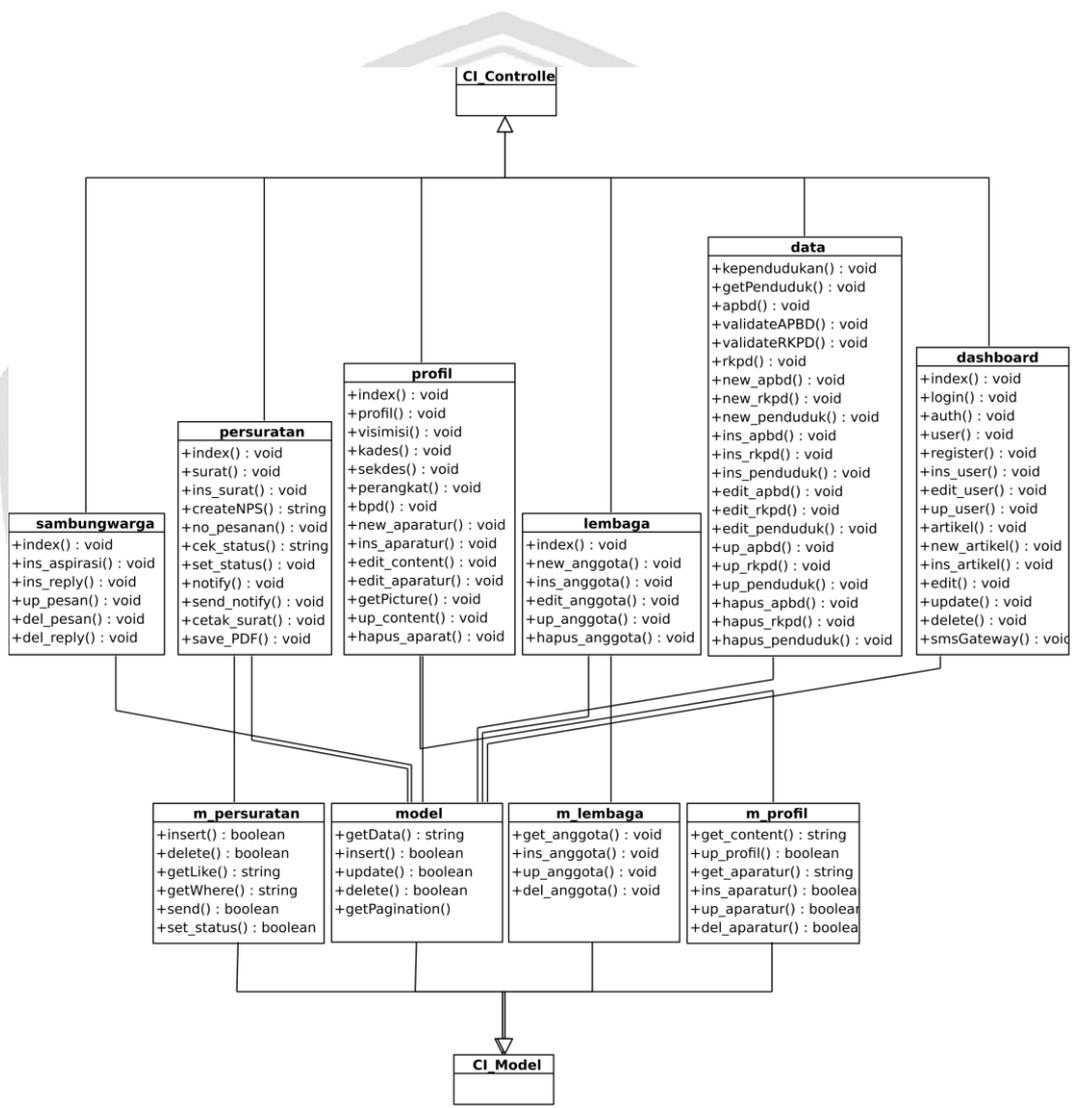
Gambar 5.4 Conceptual data model



### 5.1.1 Class Diagram

Pada perancangan sistem informasi administrasi ini terdiri dari beberapa *class* yang memiliki hubungan asosiasi. Masing-masing *class* merupakan generalisasi dari *class CI\_Controller* dan *class CI\_Model*, *class* generalisasi *CI\_Controller* antara lain *class dashboard*, *profil*, *lembaga*, *transparansi*, *sambungwarga* dan *persuratan*, sedangkan *class* generalisasi dari *CI\_Model* terdiri dari *class m\_dashboard*, *m\_profil*, *m\_lembaga*, *m\_transparansi*, dan *m\_persuratan*.

Gambar 5.5 berikut ini merupakan diagram *class* yang menggambarkan struktur *class* yang ada pada sistem ini.



Gambar 5.5 Class Diagram



### 5.1.2 Perancangan Algoritme

Tabel 5.1 berikut ini merupakan perancangan algoritme yang digunakan untuk memesan surat. Sedangkan tabel 5.2 merupakan perancangan algoritme yang digunakan untuk melakukan perubahan terhadap salah satu profil anggota BPD.

**Tabel 5.1 Perancangan algoritme memesan surat**

1	Mulai
2	..Input data
3	..Validate data
4	..IF (validate = false)
5	.....Tampil <i>form</i>
6	.....Tampil error message
7	..ELSE
8	.....Inisialisasi user input
9	.....Inisialisasi No. pemesanan
10	.....Inisialisasi data_input from user input + No. pemesanan
11	.....Insert data_input into <i>database</i>
12	.....IF (insert = true)
13	.....Message ('Pesan Berhasil Dikirim')
14	.....Tampil nomor pemesanan
15	.....ELSE
16	.....Message ('Pesan Tidak Berhasil Dikirim')
17	.....Kembali kehalaman persuratan
18	..END IF
19	END IF
20	Selesai

**Tabel 5.2 Perancangan algoritme mengedit profil BPD**

1	Mulai
2	..Input perubahan data BPD
3	..Validate data
4	..IF (validate = false)
5	.....Tampil <i>form</i> edit BPD
6	.....Tampil error message
7	..ELSE
8	.....Inisialisasi data_input from user input
9	.....Insert data_input into <i>database</i>
10	.....IF (insert = true)
11	.....Message ('Pesan Berhasil Dikirim')
12	.....ELSE
13	.....Message ('Pesan Tidak Berhasil Dikirim')
14	.....END IF
15	.....Tampil halaman profil BPD
16	..END IF
17	Selesai

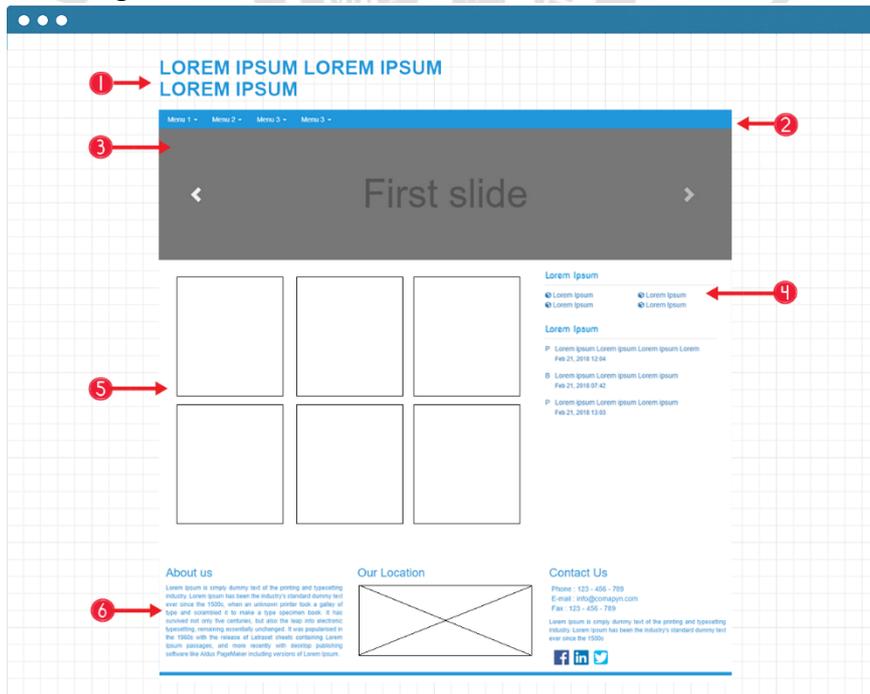
Tabel 5.3 berikut ini merupakan perancangan algoritme yang digunakan untuk menambah data ke dalam tabel APBD.

**Tabel 5.3 Perancangan algoritme menambah data APBD**

1	Mulai
2	..Input data APBD
3	..Validate data
4	..IF (validate = false)
5	.....Tampil <i>form</i> input APBD
6	.....Tampil error message
7	..ELSE
8	.....Inisialisasi <i>user</i> input
9	.....Insert data_input into <i>database</i>
10	.....IF (insert = true)
11	.....Message ('Pesan Berhasil Dikirim')
12	.....ELSE
13	.....Message ('Pesan Tidak Berhasil Dikirim')
14	.....END IF
15	.....Tampil halaman daftar APBD
16	..END IF
17	Selesai

### 5.1.3 Perancangan Antarmuka

#### 1. Perancangan antarmuka halaman utama



**Gambar 5.6 Perancangan antarmuka halaman utama**

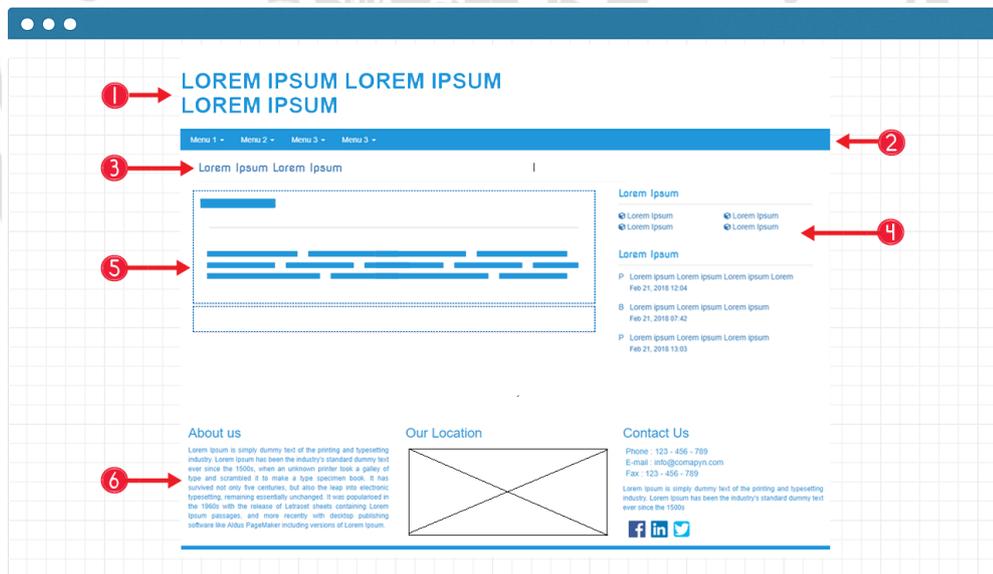
Gambar 5.6 merupakan perancangan dari halaman utama, halaman ini terdiri dari beberapa bagian antara lain:

1. Header.
2. Menu navigasi.
3. Carousel image slider.
4. Sidebar.
5. Main area, berisi thumbnail artikel.
6. Footer.

## 2. Perancangan antarmuka halaman profil

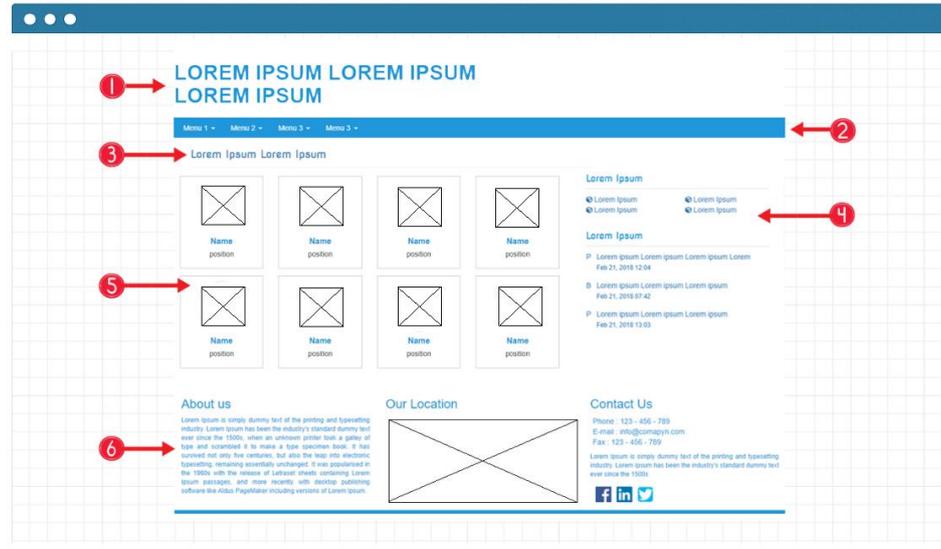
Gambar 5.7 merupakan perancangan dari halaman profil yang digunakan untuk menampilkan profil desa, visi misi, profil kepala desa, sekretaris desa, perangkat dan badan permusyawaratan desa. halaman ini terdiri dari beberapa bagian antara lain:

1. Header.
2. Menu navigasi.
3. Judul halaman.
4. Sidebar.
5. Main area, berisi desa, visi misi, profil kepala desa, sekretaris desa, perangkat dan badan permusyawaratan desa.
6. Footer.



Gambar 5.7 Perancangan antarmuka halaman profil

### 3. Perancangan antarmuka halaman menu lembaga

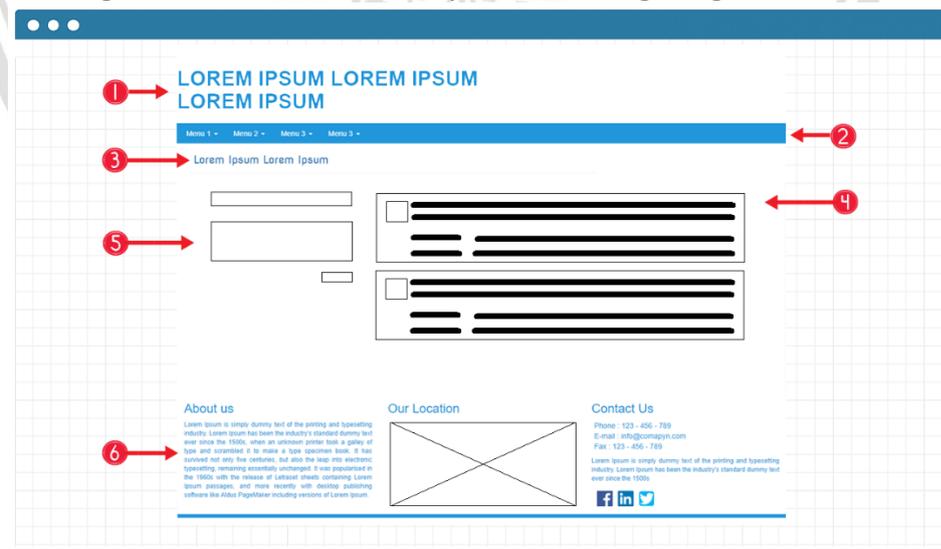


**Gambar 5.8 Perancangan antarmuka halaman lembaga**

Gambar 5.8 merupakan perancangan dari halaman lembaga. halaman ini terdiri dari beberapa bagian antara lain:

1. *Header.*
2. Menu navigasi.
3. Judul halaman.
4. *Sidebar.*
5. *Main area, berisi* daftar anggota lembaga pemberdayaan masyarakat desa, karang taruna, pembinaan kesejahteraan keluarga, dan PKK.
6. *Footer.*

### 4. Perancangan antarmuka halaman layanan sambung warga



**Gambar 5.9 Perancangan antarmuka halaman layanan sambung warga**



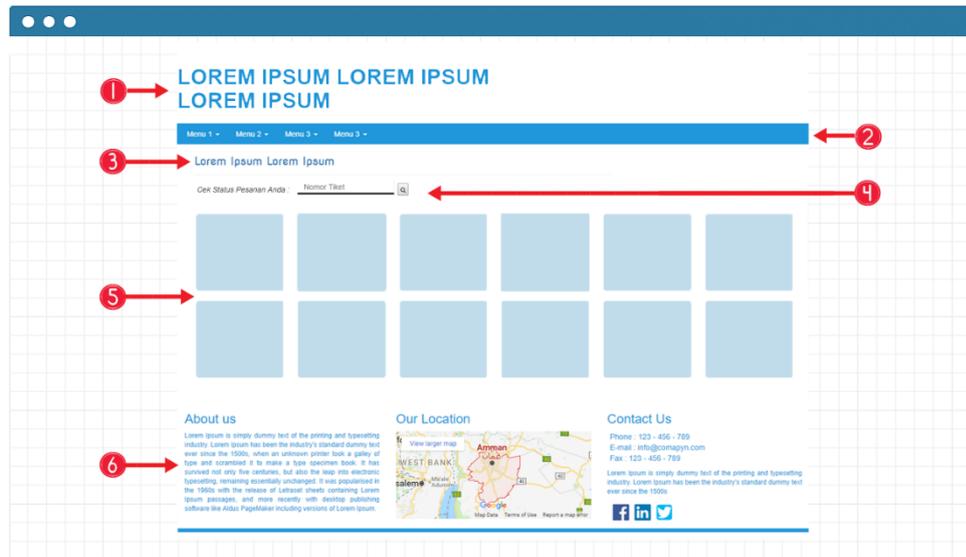
Gambar 5.9 merupakan perancangan dari halaman sabung warga. halaman ini terdiri dari beberapa bagian antara lain:

1. *Header.*
2. Menu navigasi.
3. Judul halaman.
4. *Main area*, berisi pesan aspirasi yang telah dikirim oleh warga.
5. *Form* yang digunakan untuk proses *otentikasi* dan pengiriman pesan aspirasi.
6. *Footer.*

5. Perancangan antarmuka halaman layanan persuratan

Gambar 5.10 merupakan perancangan dari halaman persuratan. halaman ini terdiri dari beberapa bagian antara lain:

1. *Header.*
2. Menu navigasi.
3. Judul halaman.
4. *Form* yang digunakan untuk mengecek status pesanan surat.
5. *Main area*, berisi menu dari beberapa jenis surat yang dapat dipesan.
6. *Footer.*



Gambar 5.10 Perancangan antarmuka halaman layanan persuratan

## 5.2 Implementasi

### 5.2.1 Spesifikasi Sistem

Sistem informasi pelayanan administrasi desa ini merupakan sistem yang berbasis *website*, sehingga dibutuhkan adanya koneksi internet untuk mengakses sistem. Pada pengembangan sistem ini digunakan spesifikasi perangkat keras dan perangkat lunak sebagai berikut:

1. Spesifikasi perangkat Keras

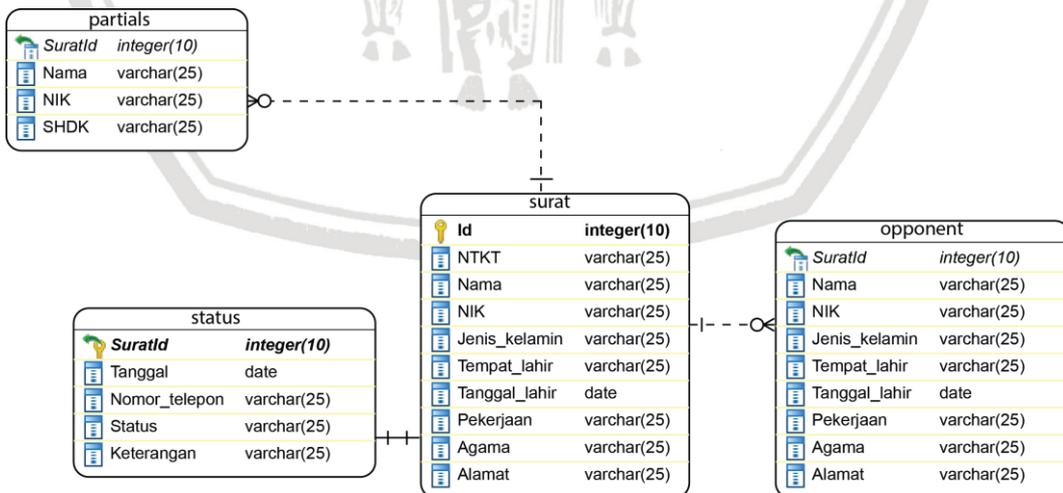
- Processor* : AMD FX-7600P *Radeon R7*
- Hard Disk Drive* : 1TB
- RAM* : 8GB DDR3L
- System Model* : ASUS N551ZU

2. Spesifikasi perangkat Lunak

- Sistem Operasi* : Windows 10 Home Single Language 64-bit
- Bahasa Pemrograman* : PHP versi 7.1.4
- Text Editor* : *Sublime Text 3*
- Server* : *Apache Web Server* versi 2.4.25
- Basis Data* : *Maria DB* versi 10.1.22

### 5.2.2 Implementasi Basis Data

Hasil perancangan basis data diterapkan ke dalam *physical data model* yang didalamnya terdapat nama tabel, kolom, tipe data, serta hubungan antar tabel itu sendiri. Hasil implementasi didapatkan beberapa tabel yaitu tabel *user*, *surat*, *opponent*, *partials*, *status*, *aspirasi*, *tanggapan*, *data\_apbd*, *data\_rkpd*, *artikel*, *penduduk*, dan *aparatur*. *Physical data model* pada pengembangan sistem ini ditunjukkan pada Gambar 5.11 dan Gambar 5.12 berikut.



Gambar 5.11 *Physical data model*





## 2. Algoritme edit BPD

Fungsi yang digunakan untuk melakukan perubahan pada salah satu profil anggota BPD adalah fungsi *up\_aparatur()* yang terletak pada *controller* profil. Pada fungsi ini setelah aktor melakukan perubahan data, data tersebut akan divalidasi, jika terdapat format yang salah maka sistem akan kembali menampilkan *form* dengan pesan kesalahan yang ada. Namun jika format masukan telah benar maka sistem akan menyimpannya ke dalam *database* untuk kemudian menampilkan pesan berhasil dan halaman profil BPD. Fungsi ini termuat pada tabel 5.5 berikut.

**Tabel 5.5 Implementasi algoritme edit BPD**

```

1 function up_aparatur(){
2   ..if ($this->form_validation->run('aparatur') == FALSE){
3     ...$this->load->view('backend/aparatur');
4   }else{
5     ...$data = $this->input->post();
6     ...$result = $this->m_profil->up_aparat($data);
7     ..if ($result == true) {
8       ....print "<script alert('Perubahan Disimpan');</script>";
9     }else{
10      ....print "<script>alert('Perubahan Gagal');</script>";
11    }
12    ..redirect('admin/profil/BPD', 'refresh');
13  ..}
14 }

```

## 3. Algoritme tambah APBD

Fungsi yang digunakan untuk menambah data APBD baru adalah fungsi *ins\_APB*D yang terletak pada *controller* data. Pada fungsi ini terdapat proses validasi atas perubahan yang dimasukkan oleh aktor, saat proses validasi berhasil maka data akan ditambahkan ke dalam *database* dengan memanggil fungsi *insert()* pada model, namun jika pada proses validasi terdapat kekeliruan maka sistem akan kembali menampilkan *form* dengan pesan kesalahan yang ada tanpa menyimpan data tersebut ke dalam *database*. Fungsi ini termuat pada tabel 5.6 berikut.

**Tabel 5.6 Implementasi algoritme tambah APBD**

```

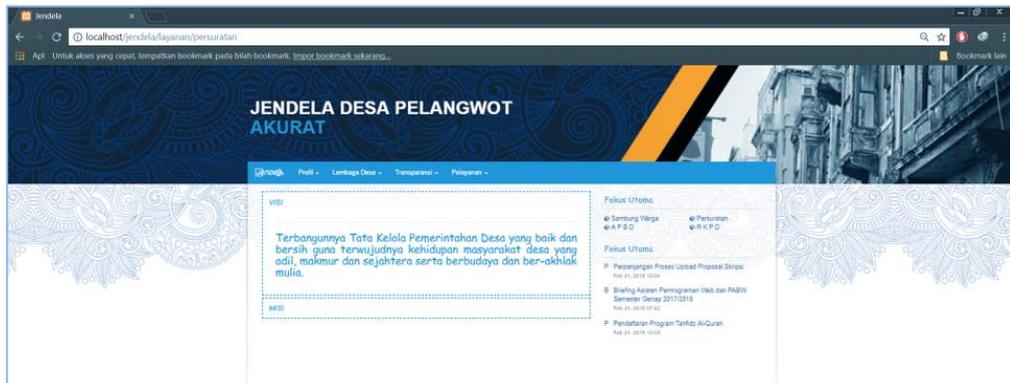
1 public function ins_APB() {
2   ..if ($this->form_validation->run('APBD') == FALSE){
3     ...$this->load->view('backend/ins_APB_'.$jenis, $data);
4   }else{
5     ..$insert = $this->model->insert($data);
6     ..if ($insert) {
7       ....print "<script>alert('Berhasil Disimpan');</script>";
8     }else{
9       ....print "<script>alert('Data Gagal Disimpan');</script>";
10    }
11    ..redirect('admin/data/APBD', 'refresh');
12  ..}
13 }

```

## 5.2.4 Implementasi Antarmuka

### 1. Implementasi antarmuka halaman profil

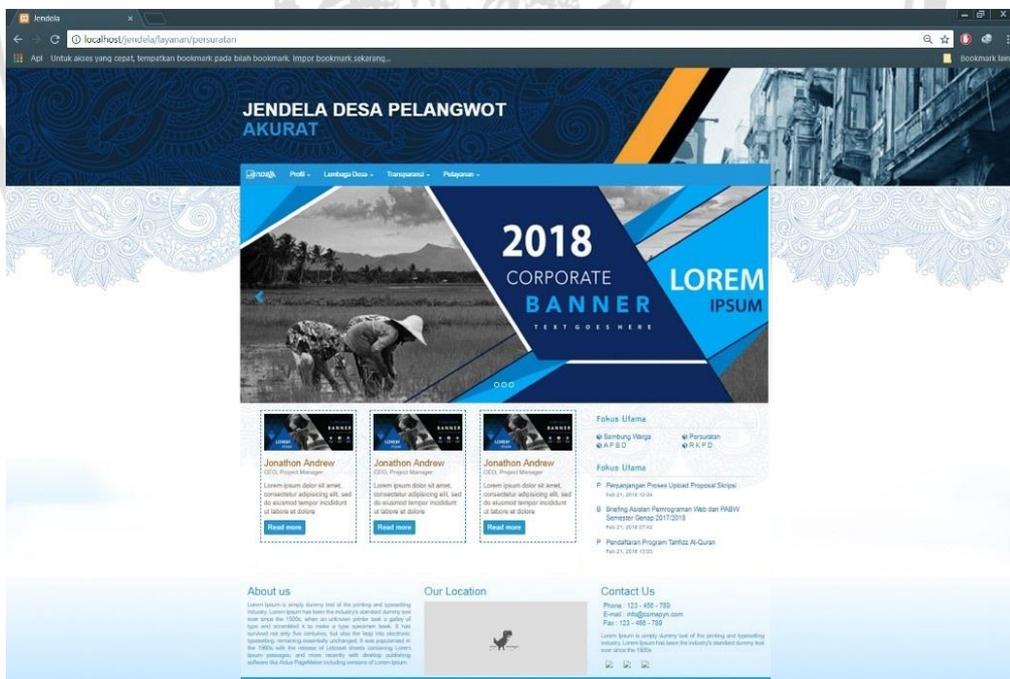
Gambar 5.11 merupakan hasil implementasi halaman profil. Halaman ini akan digunakan untuk menampilkan profil desa, visi misi, profil kepala desa, profil sekretaris desa, profil perangkat desa dan profil badan permusyawaratan desa.



Gambar 5.13 Antarmuka halaman profil

### 2. Implementasi antarmuka halaman utama

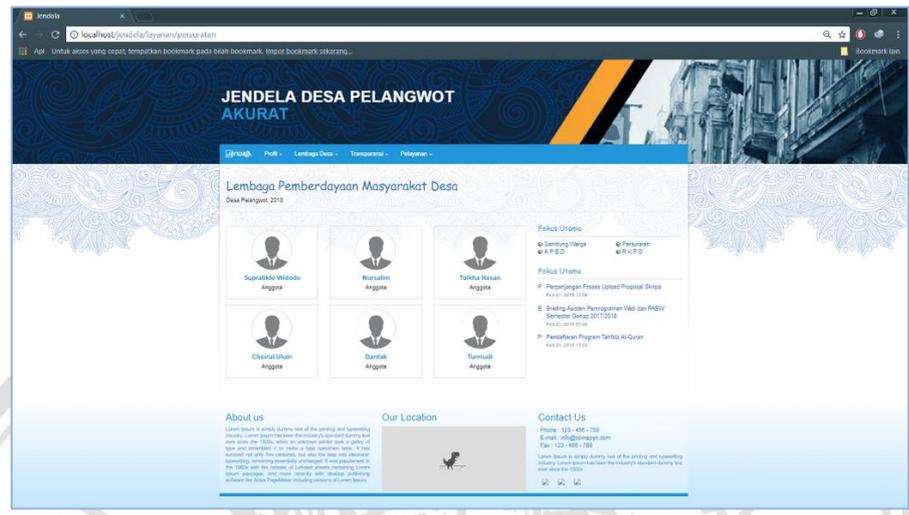
Gambar 5.12 berikut ini merupakan hasil implementasi dari perancangan halaman utama. Halaman ini berisi *thumbnail* dari artikel.



Gambar 5.14 Antarmuka halaman utama

### 3. Implementasi antarmuka halaman lembaga

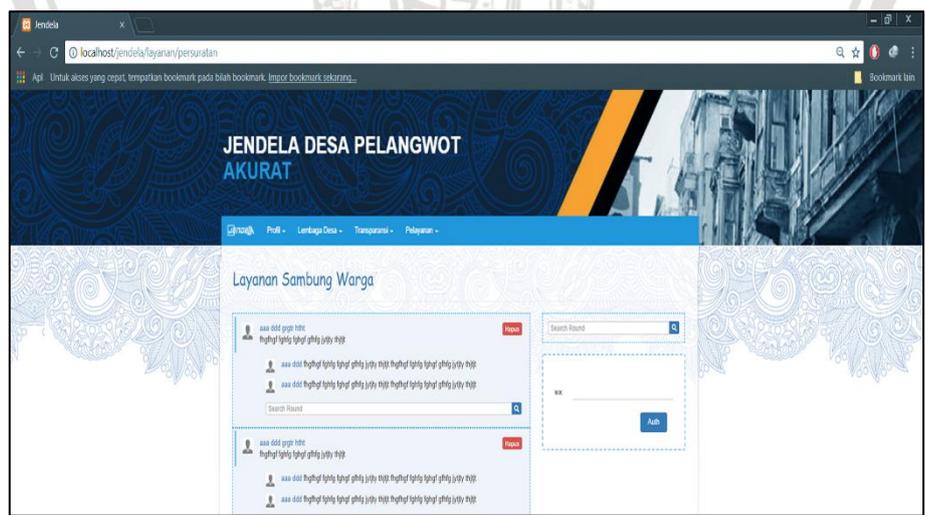
Gambar 5.13 merupakan hasil implementasi dari perancangan halaman lembaga. Halaman ini akan digunakan untuk menampilkan daftar profil anggota lembaga pemberdayaan masyarakat, anggota pembinaan kesejahteraan keluarga, anggota karang taruna, PKK dan anggota satuan perlindungan masyarakat.



Gambar 5.15 Antarmuka halaman lembaga

### 4. Implementasi antarmuka halaman sambung warga

hasil implementasi dari perancangan halaman sambung warga terdapat pada Gambar 5.14. Halaman ini akan digunakan untuk menampilkan daftar pesan aspirasi. Pada halaman ini juga terdapat form yang digunakan untuk mengirim pesan aspirasi.



Gambar 5.16 Antarmuka halaman sambung warga



5. Implementasi antarmuka halaman persuratan

Gambar 5.15 berikut ini merupakan hasil implementasi dari perancangan halaman sambung persuratan. Halaman ini akan digunakan untuk menampilkan menu atau jenis surat yang dapat dipesan oleh masyarakat.



Gambar 5.17 Antarmuka halaman persuratan

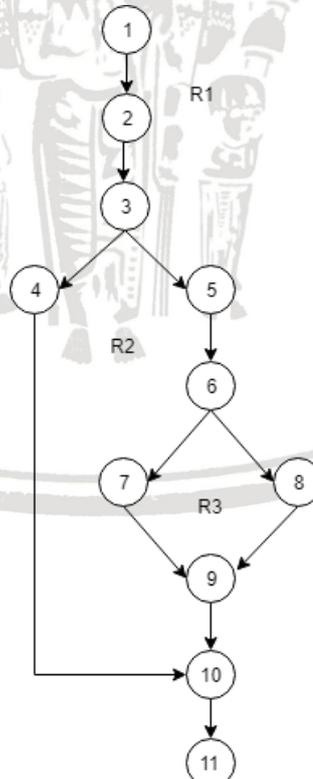
## BAB 6 PENGUJIAN

### 6.1 Pengujian Unit

#### 6.1.1 Pengujian unit memesan surat

Mulai	1
Input data pemesanan	2
Validate data	3
IF (validate = false)	4
Tampil form	4
Tampil error message	4
ELSE	
Inisialisasi user input	5
Inisialisasi No. pemesanan	5
Insert data_input into database	5
IF (insert = true)	6
Message ('Pesan Berhasil Dikirim')	7
Tampil nomor pemesanan	7
ELSE	
Message ('Pesan Tidak Berhasil Dikirim')	8
Kembali kehalaman form keterangan nikah	8
END IF	9
END IF	10
Selesai	11

Berdasarkan algoritme yang ada maka didapatkan bentuk *flow graph* pada Gambar 6.1 sebagai berikut:



Gambar 6.1 *Flow graph* pengujian unit memesan surat belum menikah

Berdasarkan jumlah *flow graph* yang telah diperoleh, maka ditentukan tiga jalur independen dan jumlah *cyclomatic complexity*, seperti berikut:

- Jalur *independen*  
 Jalur 1 : 1-2-3-4-10-11  
 Jalur 2 : 1-2-3-5-6-7-9-10-11  
 Jalur 3 : 1-2-3-5-6-7-8-9-10-11
- Perhitungan *cyclomatic complexity*:  

$$V(G) = Edge - Node + 2$$

$$= 12 - 11 + 2$$

$$= 3$$

$$V(G) = Predicate Node + 1$$

$$= 2 + 1$$

$$= 3$$

$$V(G) = Jumlah Region$$

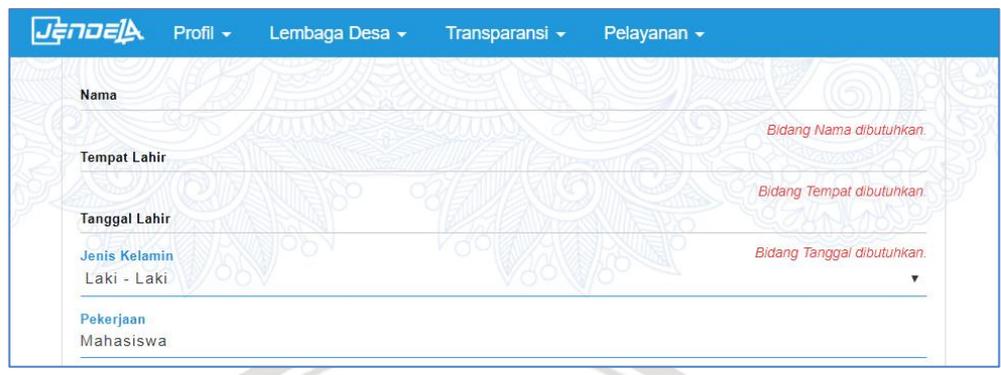
$$= 3$$

Berdasarkan jalur independen yang ada maka didapat kasus uji pada tabel 6.1.

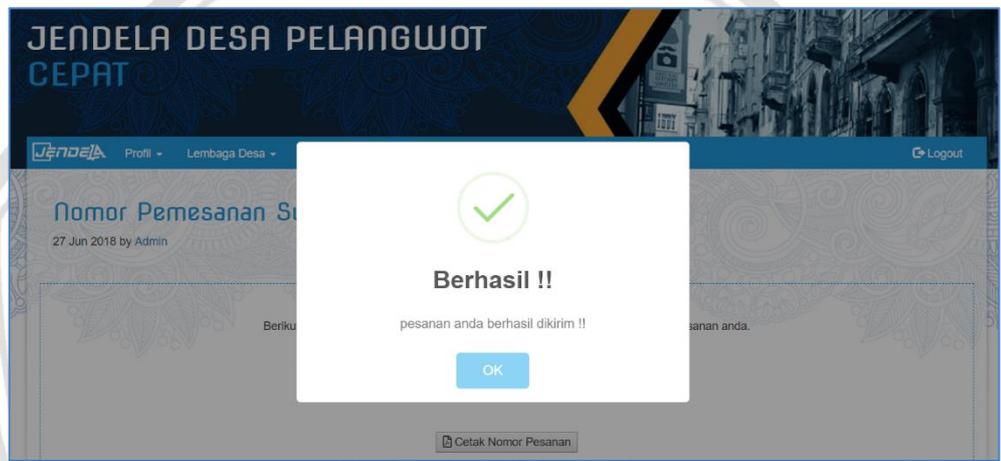
**Tabel 6.1 Kasus uji memesan surat**

Jalur	Prosedur Uji	Keluaran Yang Diharapkan	Keluaran Yang Didapatkan	Status
1	Memanggil operasi <i>insert()</i> dengan mengisi bidang NIK = 3524081112470000 Nama = nul Tempat Lahir = nul Tanggal Lahir = nul Jenis Kelamin = Laki - Laki Agama = Islam Pekerjaan = Mahasiswa Alamat = Pelangwot	Kembali ke <i>form</i> menampilkan pesan bahwa nama, tempat dan tanggal lahir harus diisi.	Kembali ke <i>form</i> menampilkan pesan bahwa nama, tempat dan tanggal lahir harus diisi.	valid
2	Memanggil operasi <i>insert()</i> dengan variabel NIK = 3524081112470000 Nama = Anton Firdaus Tempat Lahir = Lamongan Tanggal Lahir = 31-05-1996 Jenis Kelamin = Laki- Laki Agama = Islam Pekerjaan = Mahasiswa Alamat = Pelangwot	Data tersimpan ke dalam <i>database</i> , sistem menampilkan pesan berhasil	Data tersimpan ke dalam <i>database</i> , sistem menampilkan pesan berhasil	valid
3	Memanggil operasi <i>insert()</i> dengan Mematikan SQL Server ketika hendak mengirim data	Sistem menampilkan pesan kesalahan koneksi <i>database</i>	Sistem menampilkan pesan kesalahan koneksi <i>database</i>	valid

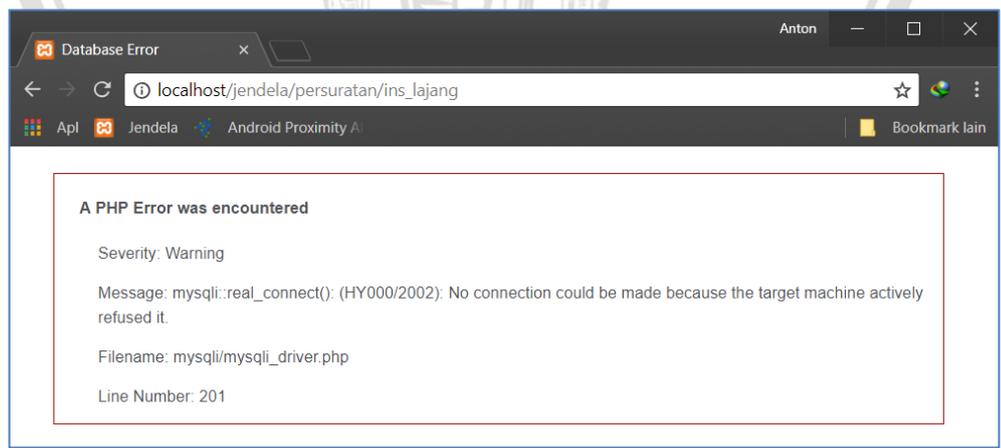
Gambar 6.2, 6.3, dan 6.4 berikut ini adalah hasil yang didapatkan pada pengujian fungsional memesan surat. Berdasarkan hasil ini dapat disimpulkan bahwa seluruh kasus uji memiliki hasil yang sesuai dengan yang diharapkan.



Gambar 6.2 Hasil pengujian unit memesan surat



Gambar 6.3 Hasil pengujian unit memesan surat



Gambar 6.4 Hasil pengujian unit memesan surat

### 6.1.2 Pengujian unit edit BPD

Mulai

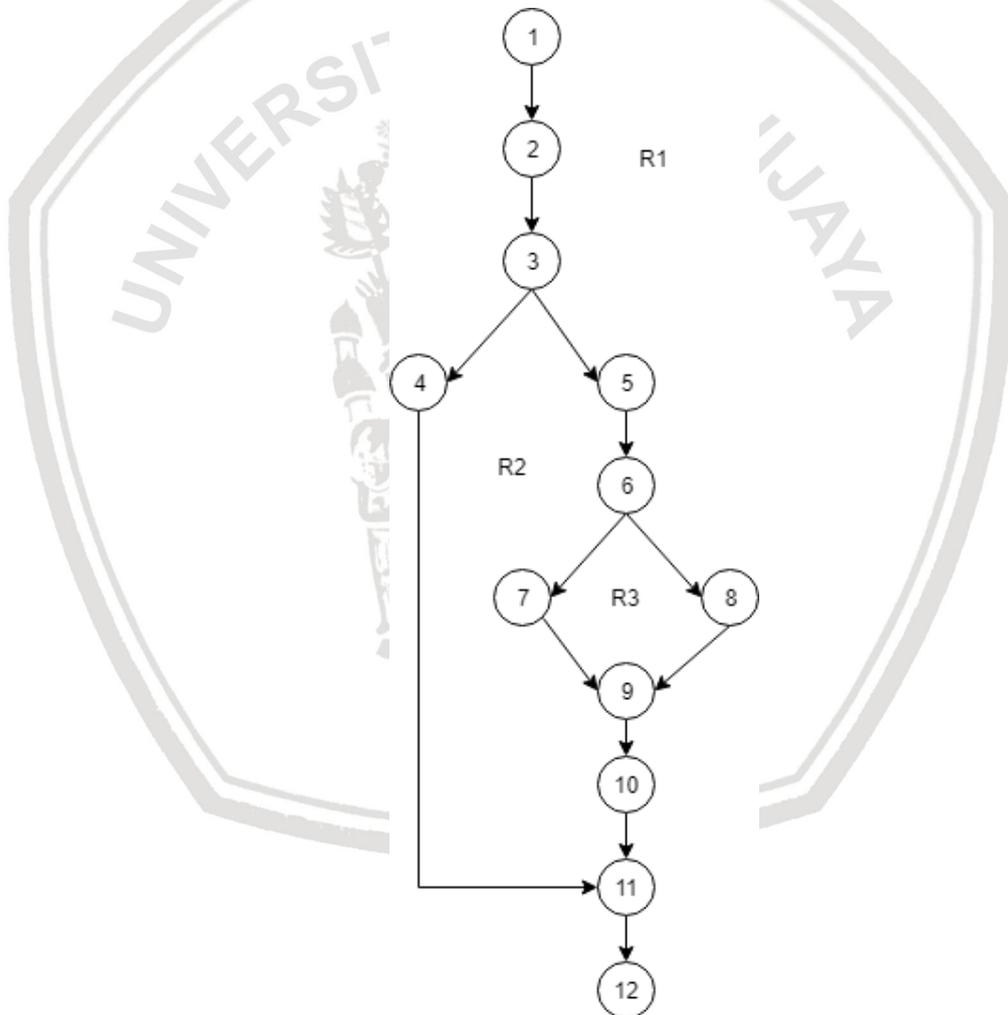
1



```

Input perubahan data BPD                                     2
Validate data                                              3
IF (validate = false)                                       3
    Tampil form edit BPD                                    4
    Tampil error message                                    4
ELSE
    Inisialisasi data_input from user input                5
    Insert data_input into database                        6
    IF (insert = true)                                     6
        Message ('Data Berhasil Disimpan')                7
    ELSE                                                    8
        Message ('Data Tidak Tersimpan')                  8
    END IF                                                 9
    Tampil halaman profil BPD                              10
END IF                                                    11
Selesai                                                    12
    
```

Berdasarkan algoritme yang ada maka didapatkan bentuk *flow graph* pada Gambar 6.5, jalur independen, serta *cyclomatic complexity* berikut:



**Gambar 6.5 Flow graph pengujian unit edit BPD**

Berdasarkan jumlah *flow graph* yang telah diperoleh, maka ditentukan tiga jalur independen dan jumlah *cyclomatic complexity*, seperti berikut:



- Jalur independen  
 Jalur 1 : 1-2-3-4-11-12  
 Jalur 2 : 1-2-3-5-6-7-9-10-11-12  
 Jalur 3 : 1-2-3-5-6-7-8-9-10-11-12
- Perhitungan *cyclomatic complexity*:  

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

$$= 13 - 12 + 2$$

$$= 3$$

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

$$= 2 + 1$$

$$= 3$$

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

$$= 3$$

Berdasarkan jalur independen yang ada maka didapat kasus uji pada tabel 6.2.

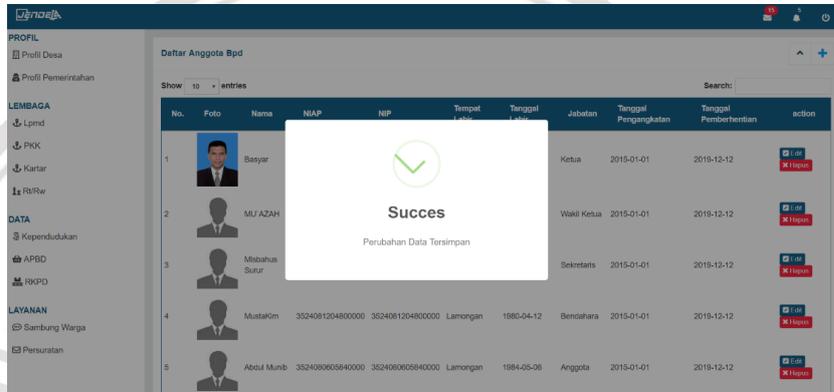
**Tabel 6.2 Kasus uji *edit* BPD**

Jalur	Prosedur Uji	Keluaran Yang Diharapkan	Keluaran Yang Didapatkan	Status
1	Memanggil operasi up_aparatur dengan variabel Nama : Basyar NIAP : 3524081710530000 NIP : 3524081710530000 Tempat Lahir : Lamongan Tanggal Lahir : 17/10/1953 Jenis Kelamin : Laki – Laki Jabatan : Ketua Tanggal Pengangkatan : 1/1/2015 Tanggal Pemberhentian 12/12/2019	Data tersimpan ke dalam <i>database</i> , sistem menampilkan pesan bahwa perubahan telah berhasil dilakukan	Data tersimpan ke dalam <i>database</i> , sistem menampilkan pesan bahwa perubahan telah berhasil dilakukan	valid
2	Memanggil operasi up_aparatur dengan variabel Nama : Basyar NIAP : 12345 NIP : 12345 Tempat Lahir : Lamongan Tanggal Lahir : 17/10/1953 Jenis Kelamin : Laki – Laki Jabatan : Ketua Pendidikan : STRATA I Tanggal Pengangkatan : 1/1/2015 Tanggal Pemberhentian 12/12/2019	Kembali ke <i>form</i> menampilkan pesan NIAP dan NIK harus 16 karakter	Kembali ke <i>form</i> menampilkan pesan NIAP dan NIK harus 16 karakter	valid

Tabel 6.2 Kasus uji *edit* BPD (lanjutan)

Jalur	Prosedur Uji	Keluaran Yang Diharapkan	Keluaran Yang Didapatkan	Status
3	Memanggil operasi up_APBD dengan variabel dengan mematikan SQL Server selama proses perubahan data hingga tombol simpan di tekan	Data tidak tersimpan sistem menampilkan pesan kesalahan	Data tidak tersimpan sistem menampilkan pesan kesalahan	valid

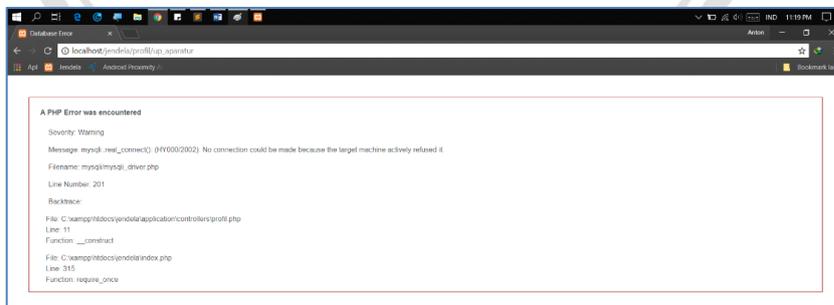
Gambar 6.6, 6.7, dan 6.8 berikut ini adalah hasil yang didapatkan pada pengujian fungsional memesan surat. Berdasarkan hasil tersebut dapat disimpulkan bahwa kasus uji perubahan data APBD berjalan sesuai dengan yang diharapkan.



Gambar 6.6 Hasil pengujian unit edit BPD



Gambar 6.7 Hasil pengujian unit edit BPD



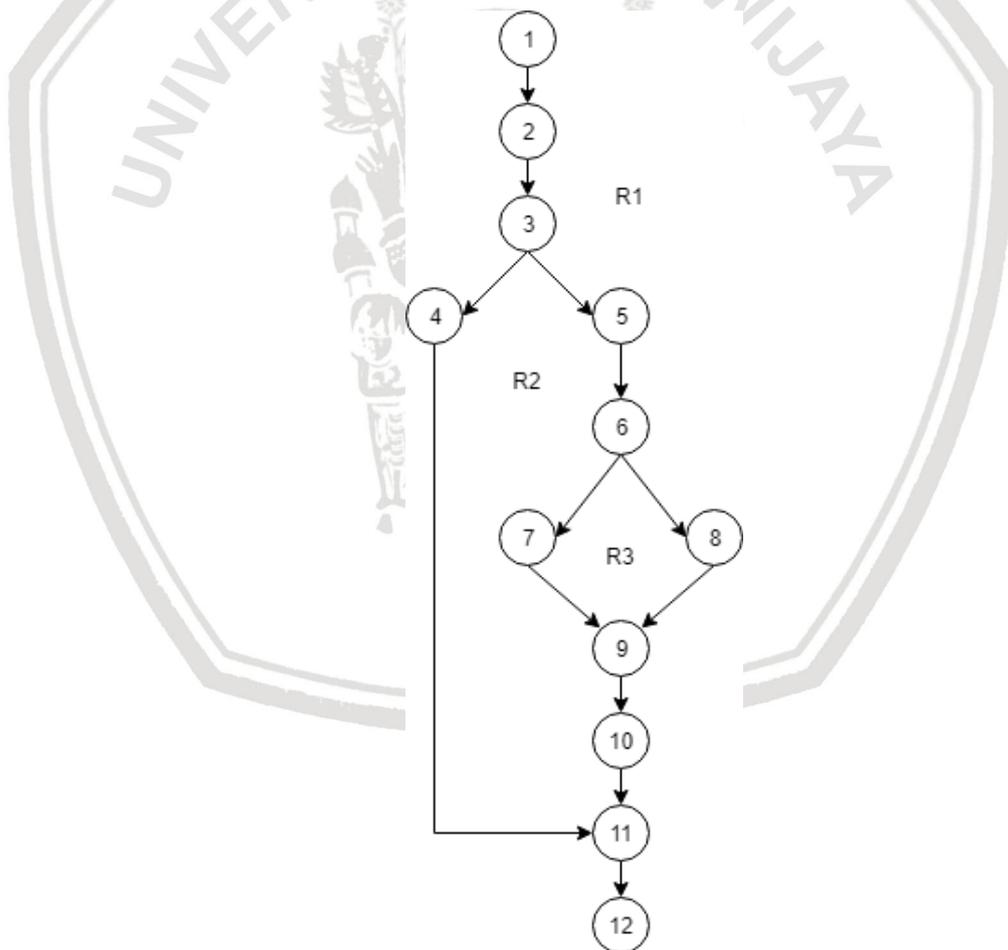
Gambar 6.8 Hasil pengujian unit edit BPD



### 6.1.3 Pengujian unit tambah APBD

Mulai	1
Input data APBD	2
Validate data	3
IF (validate = false)	3
Tampil form APBD	4
Tampil error message	4
ELSE	
Inisialisasi data_input from user input	5
Insert data_input into database	
IF (insert = true)	6
Message ('Data Berhasil Disimpan')	7
ELSE	
Message ('Data Tidak Tersimpan')	8
END IF	9
Tampil halaman APBD	10
END IF	11
Selesai	12

Berdasarkan algoritme yang ada maka didapatkan bentuk *flow graph* pada Gambar 6.5, jalur independen serta *cyclomatic complexity* berikut:



Gambar 6.9 *Flow graph* pengujian unit tambah APBD

Berdasarkan jumlah *flow graph* yang telah diperoleh, maka ditentukan tiga jalur independen dan jumlah *cyclomatic complexity*, seperti berikut:

- Jalur independen  
 Jalur 1 : 1-2-3-4-11-12  
 Jalur 2 : 1-2-3-5-6-7-9-10-11-12  
 Jalur 3 : 1-2-3-5-6-7-8-9-10-11-12
- Perhitungan *cyclomatic complexity*:  

$$V(G) = Edge - Node + 2$$

$$= 13 - 12 + 2$$

$$= 3$$

$$V(G) = Predicate Node + 1$$

$$= 2 + 1$$

$$= 3$$

$$V(G) = Jumlah Region$$

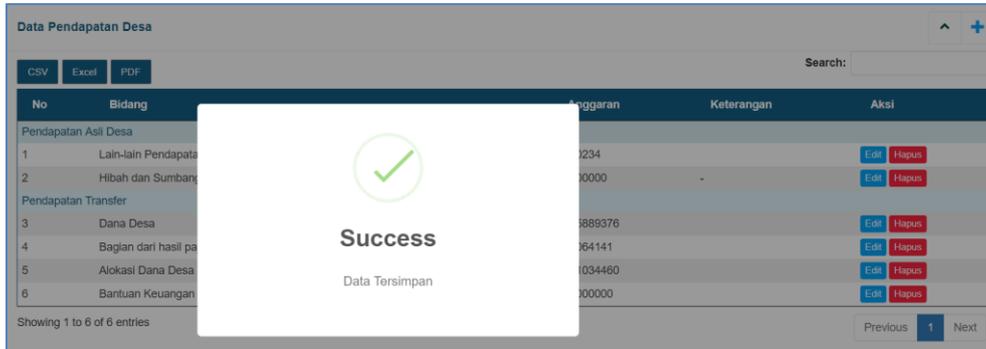
$$= 3$$

Berdasarkan jalur independen yang ada maka didapat kasus uji pada tabel 6.3.

**Tabel 6.3 Kasus uji tambah APBD**

Jalur	Prosedur Uji	Keluaran Yang Diharapkan	Keluaran Yang Didapatkan	Status
1	Memanggil prosedur ins_APBDD() dengan variabel Jenis : Pendapatan Lain Lain Bidang : Hibah dan Sumbangan dari pihak ke- 3 Jumlah pendapatan : 75.000.000 Keterangan : -	Data tersimpan ke dalam <i>database</i> , sistem menampilkan pesan berhasil	Data tersimpan ke dalam <i>database</i> , sistem menampilkan pesan berhasil	valid
2	Memanggil prosedur ins_APBDD() dengan variabel Jenis : Pendapatan Lain Lain Bidang : Jumlah pendapatan : Keterangan : -	Menampilkan pesan bahwa <i>field</i> bidang dan jumlah pendapatan harus diisi	Menampilkan pesan bahwa <i>field</i> bidang dan jumlah pendapatan harus diisi	valid
4	Memanggil prosedur ins_APBDD() dengan Mematikan SQL Server selama proses perubahan data hingga tombol simpan di tekan	Data tidak tersimpan sistem menampilkan pesan kesalahan	Data tidak tersimpan sistem menampilkan pesan kesalahan	valid

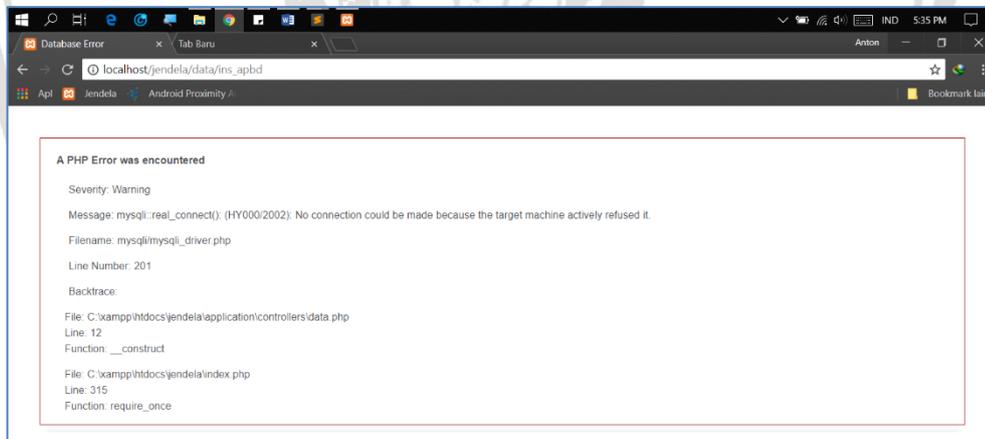
Berdasarkan hasil pada tabel 6.3 dapat dikatakan bahwa keluaran yang didapatkan dalam fungsional tambah data APBD telah sesuai dengan apa yang diharapkan. Hasil pengujian ini ditunjukkan pada Gambar 6.10, 6.11, dan 6.12.



Gambar 6.10 Hasil pengujian unit tambah APBD



Gambar 6.11 Hasil pengujian unit tambah APBD



Gambar 6.12 Hasil pengujian unit tambah APBD

## 6.2 Pengujian Validasi

Pengujian *black-box* pada penelitian ini menggunakan pengujian validasi yang dilakukan dengan berbagai prosedur uji pada tiap-tiap fungsional dengan membandingkan hasil keluaran dengan apa yang telah ditetapkan sebelumnya.

1. Pengujian *login*

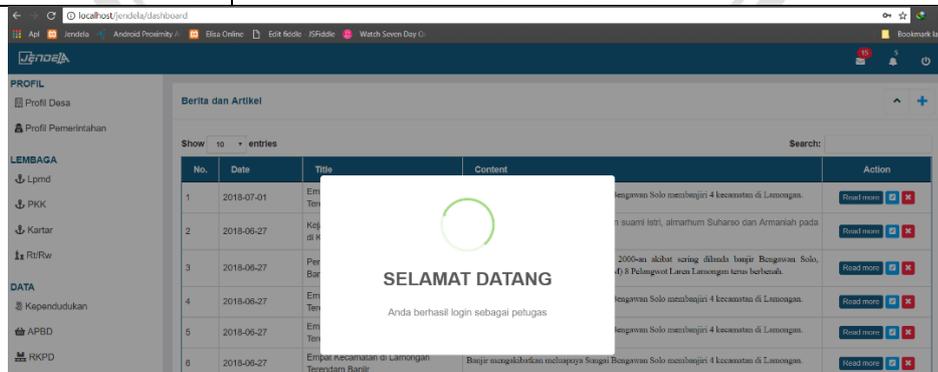
Beberapa kasus uji yang terdapat pada fungsional *login* termuat pada tabel 6.4 dan 6.5. Sedangkan hasil uji ditunjukkan pada Gambar 6.13 dan 6.14 berikut ini.

**Tabel 6.4 Kasus uji berhasil *login* sebagai petugas**

<b>Kode Kasus Uji</b>	PV_001
<b>Nama Kasus Uji</b>	Kasus uji berhasil login sebagai petugas
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Membuka halaman <i>login</i>.</li> <li>2. Memilih tab petugas.</li> <li>3. Memasukkan <i>username</i> = grahapraja dan <i>password</i> = Graha21792.</li> <li>4. Menekan tombol <i>login</i>.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan selamat datang pada halaman utama petugas.
<b>Hasil</b>	Sistem menampilkan pesan selamat datang pada halaman utama petugas.
<b>Status</b>	valid

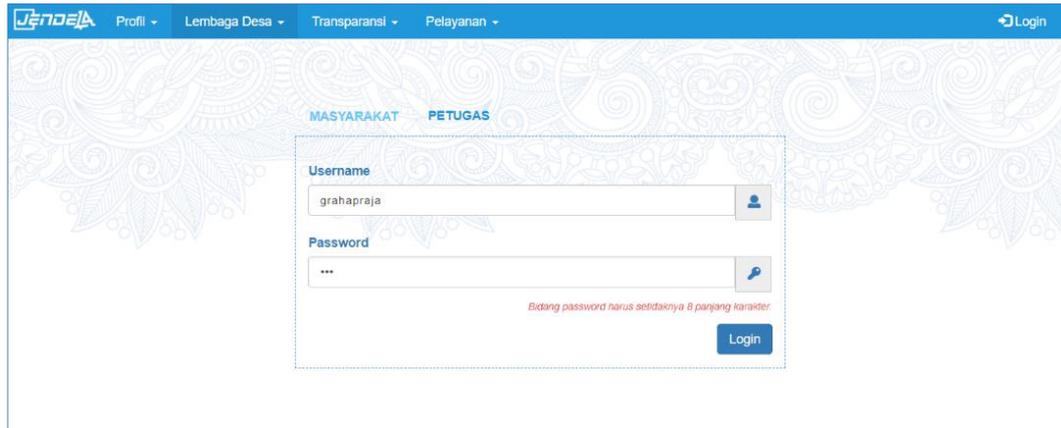
**Tabel 6.5 Kasus uji gagal *login***

<b>Kode Kasus Uji</b>	PV_002
<b>Nama Kasus Uji</b>	Kasus uji gagal <i>login</i>
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Membuka halaman <i>login</i>.</li> <li>2. Memilih tab petugas.</li> <li>3. Memasukkan <i>username</i> grahapraja dan <i>password</i> = 123.</li> <li>4. Menekan tombol <i>login</i>.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan bahwa <i>field password</i> haruslah diisi dengan karakter minimal 8 digit pada halaman login.
<b>Hasil</b>	Sistem menampilkan pesan bahwa <i>field password</i> haruslah diisi dengan karakter minimal 8 digit pada halaman <i>login</i> .
<b>Status</b>	Valid



**Gambar 6.13 Hasil pengujian kasus uji berhasil *login***





Gambar 6.14 Hasil pengujian kasus uji gagal login

2. Pengujian menambah artikel

Kasus uji yang terdapat pada fungsional menambah artikel termuat pada tabel 6.6, 6.7 dan 6.8. Sedangkan hasil uji ditunjukkan pada Gambar 6.15, 6.16 dan 6.17 berikut ini.

Tabel 6.6 Kasus uji berhasil menambah artikel

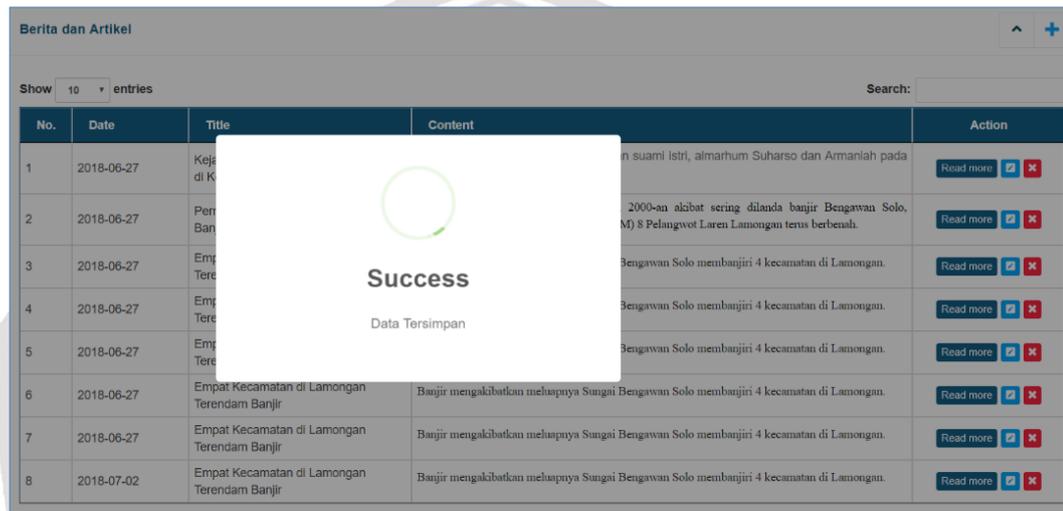
<b>Kode Kasus Uji</b>	PV_003
<b>Nama Kasus Uji</b>	Kasus uji berhasil menambah berita atau artikel
<b>Prosedur</b>	1. Mengisi field judul dan isi artikel. 2. Menekan tombol simpan.
<b>Hasil yang diharapkan</b>	Sistem menyimpan menyimpan data ke dalam <i>database</i> dan menampilkan pesan berhasil serta artikel tersebut pada daftar artikel pada halaman utama petugas.
<b>Hasil</b>	Sistem menyimpan menyimpan data ke dalam <i>database</i> dan menampilkan pesan berhasil serta artikel tersebut pada daftar artikel pada halaman utama petugas.
<b>Status</b>	Valid

Tabel 6.7 Kasus uji batal menambah artikel

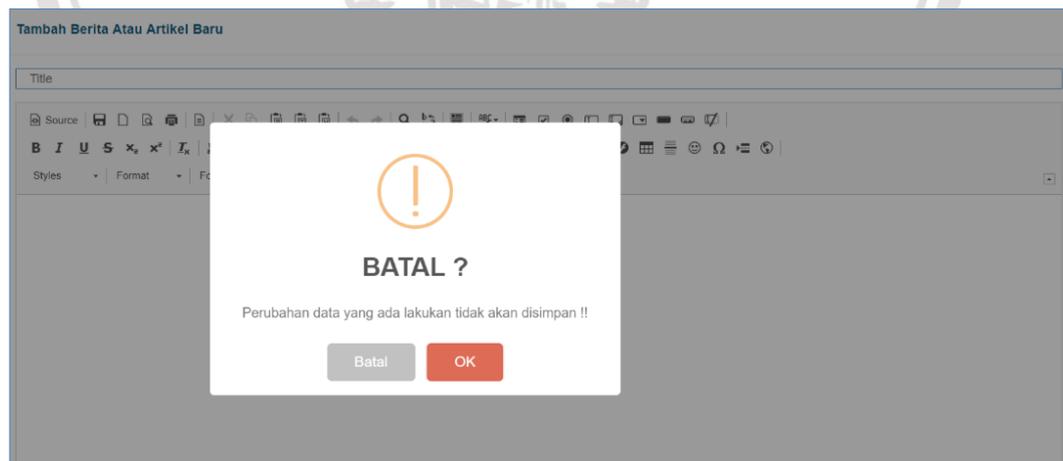
<b>Kode Kasus Uji</b>	PV_004
<b>Nama Kasus Uji</b>	Kasus uji berhasil menambah berita atau artikel.
<b>Prosedur</b>	1. Menekan tombol batal pada halaman <i>input</i> berita atau artikel baru.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan konfirmasi bahwa data tidak akan disimpan.
<b>Hasil</b>	Sistem menampilkan pesan konfirmasi bahwa data tidak akan disimpan.
<b>Status</b>	Valid

Tabel 6.8 Kasus uji gagal menambah artikel

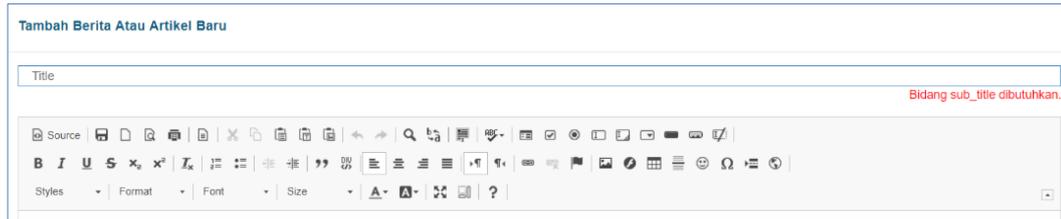
Kode Kasus Uji	PV_005
Nama Kasus Uji	Kasus uji gagal menambah berita atau artikel.
Prosedur	1. <i>User</i> menekan tombol simpan tanpa <i>mengisi</i> judul berita atau artikel.
Hasil yang diharapkan	Sistem menampilkan pesan bahwa judul artikel harus diisi.
Hasil	Sistem menampilkan pesan bahwa judul artikel harus diisi.
Status	Valid



Gambar 6.15 Hasil pengujian kasus uji berhasil menambah berita atau artikel



Gambar 6.16 Hasil pengujian kasus uji batal menambah berita atau artikel



**Gambar 6.17 Hasil pengujian kasus uji gagal menambah berita atau artikel**

3. Pengujian mengubah isi artikel

Beberapa kasus uji yang terdapat pada fungsional mengubah isi berita atau artikel terbuat pada tabel 6.9 dan 6.10. Sedangkan hasil uji ditunjukkan pada Gambar 6.18 dan 6.19 berikut ini.

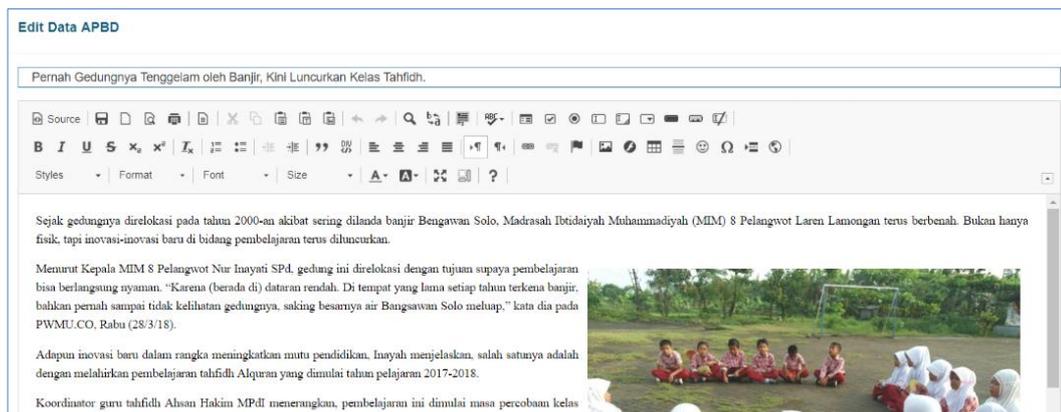
**Tabel 6.9 Kasus menekan tombol edit**

<b>Kode Kasus Uji</b>	PV_006
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol <i>edit</i> .
<b>Prosedur</b>	1. <i>User</i> menekan <i>edit</i> pada salah satu artikel pada halaman dashboard atau daftar artikel.
<b>Hasil yang diharapkan</b>	Sistem menampilkan judul serta isi artikel pada halaman <i>edit</i> artikel.
<b>Hasil</b>	Sistem menampilkan judul serta isi artikel pada halaman <i>edit</i> artikel.
<b>Status</b>	Valid

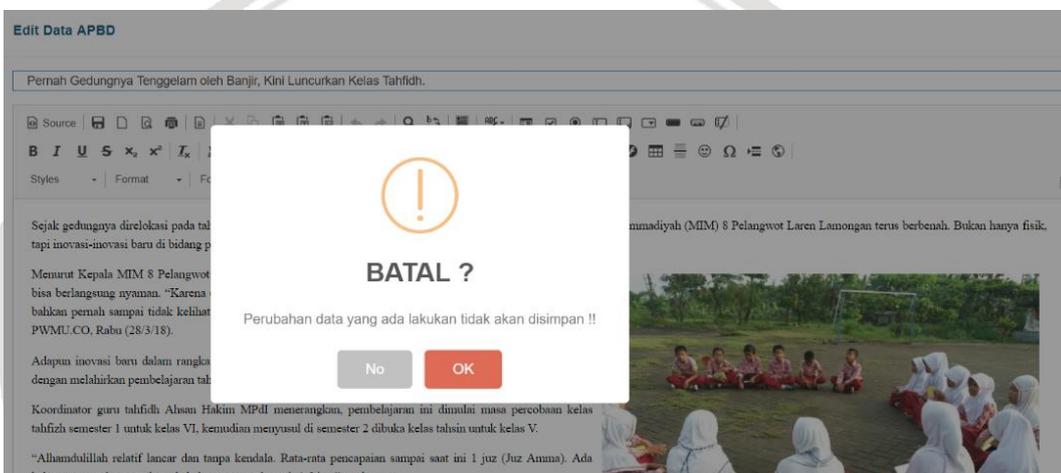
**Tabel 6.10 Kasus uji batal mengubah isi artikel**

<b>Kode Kasus Uji</b>	PV_007
<b>Nama Kasus Uji</b>	Kasus uji batal mengubah isi berita atau artikel.
<b>Prosedur</b>	1. <i>User</i> menekan tombol batal pada halaman <i>edit</i> artikel
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan konfirmasi bahwa perubahan data tidak akan disimpan.
<b>Hasil</b>	Sistem menampilkan pesan konfirmasi bahwa perubahan data tidak akan disimpan.
<b>Status</b>	Valid





Gambar 6.18 Hasil pengujian kasus uji menekan tombol edit



Gambar 6.19 Hasil pengujian kasus uji menekan tombol batal

#### 4. Pengujian menghapus artikel

Beberapa kasus uji pada fungsional menghapus berita atau artikel termuat pada tabel 6.11, 6.12 dan 6.13. Sedangkan hasil uji ditunjukkan pada Gambar 6.20, 6.21 dan 6.22 berikut ini.

Tabel 6.11 Kasus uji menekan tombol hapus

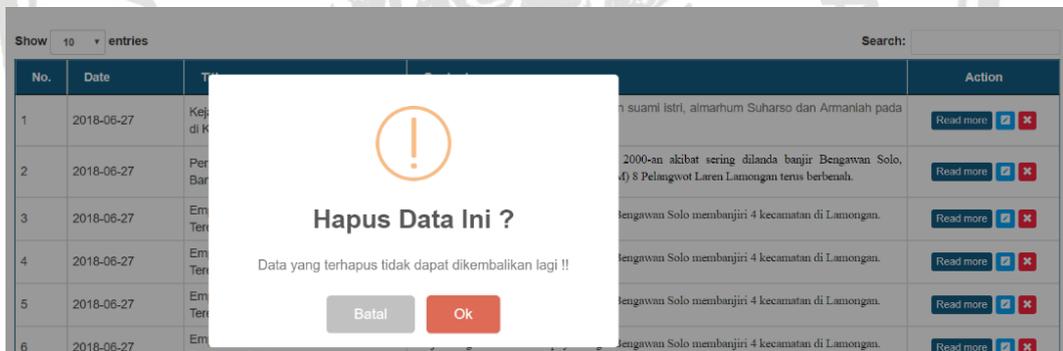
<b>Kode Kasus Uji</b>	PV_008
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol hapus.
<b>Prosedur</b>	1. <i>User</i> menekan tombol hapus pada salah satu artikel pada daftar artikel.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan konfirmasi bahwa <i>user</i> akan melakukan penghapusan data.
<b>Hasil</b>	Sistem menampilkan pesan konfirmasi bahwa <i>user</i> akan melakukan penghapusan data.
<b>Status</b>	Valid

**Tabel 6.12 Kasus uji menekan tombol Ya**

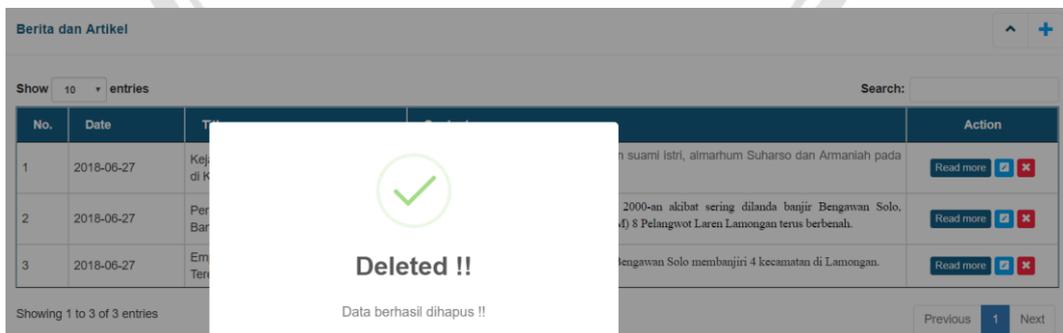
<b>Kode Kasus Uji</b>	PV_009
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol Ya.
<b>Prosedur</b>	1. <i>User</i> menekan tombol Ya pada pesan konfirmasi hapus.
<b>Hasil yang diharapkan</b>	Sistem menghapus data dan menampilkan pesan berhasil.
<b>Hasil</b>	Sistem menghapus data dan menampilkan pesan berhasil.
<b>Status</b>	Valid

**Tabel 6.13 Kasus uji menekan tombol Tidak**

<b>Kode Kasus Uji</b>	PV_010
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol Tidak.
<b>Prosedur</b>	1. <i>User</i> menekan tombol Tidak pada pesan konfirmasi hapus.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan batal tanpa menghapus data.
<b>Hasil</b>	Sistem menampilkan pesan batal tanpa menghapus data.
<b>Status</b>	Valid

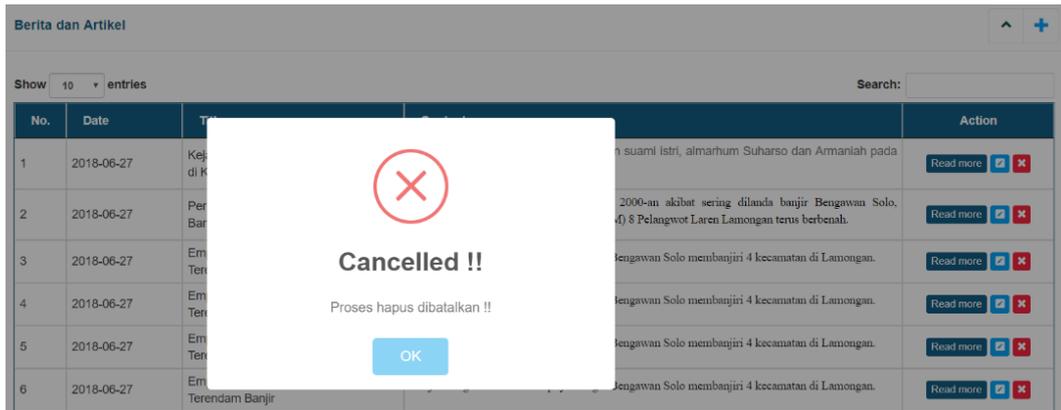


**Gambar 6.20 Hasil pengujian kasus uji menekan tombol hapus**



**Gambar 6.21 Hasil pengujian kasus uji menekan tombol OK**





Gambar 6.22 Hasil pengujian kasus uji menekan tombol Batal

5. Pengujian menambah profil perangkat

Beberapa kasus yang ada pada fungsional menambah profil perangkat termuat pada tabel 6.14, 6.15, 6.16, 6.17 dan 6.18. Sedangkan hasil uji ditunjukkan pada Gambar 6.23, 6.24, 6.25, 6.26 dan 6.27 berikut ini.

Tabel 6.14 Kasus uji berhasil menambah profil perangkat

Kode Kasus Uji	PV_011
Nama Kasus Uji	Kasus uji berhasil menambah profil anggota perangkat.
Prosedur	<ol style="list-style-type: none"> <li>1. <i>User</i> memasukkan data diri secara lengkap dan tepat.</li> <li>2. <i>User</i> memasukkan foto dengan format png dan ukuran kurang dari 1MB.</li> <li>3. <i>User</i> menekan tombol simpan</li> </ol>
Hasil yang diharapkan	Sistem menyimpan data ke dalam <i>database</i> dan menampilkan pesan berhasil.
Hasil	Sistem menyimpan data ke dalam <i>database</i> dan menampilkan pesan berhasil.
Status	Valid

Tabel 6.15 Kasus uji memasukkan file MP3 ke dalam *field* foto

Kode Kasus Uji	PV_012
Nama Kasus Uji	Kasus uji memasukkan file <i>mp3</i> ke dalam <i>field</i> foto.
Prosedur	<ol style="list-style-type: none"> <li>1. <i>User</i> memasukkan foto dengan format <i>mp3</i> ke dalam bidang foto.</li> </ol>
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa format file yang didukung hanya <i>jpeg, jpg, png</i> dan <i>gif</i> .
Hasil	Sistem menampilkan pesan kesalahan bahwa format file yang didukung hanya <i>jpeg, jpg, png</i> dan <i>gif</i> .
Status	Valid



**Tabel 6.16 Kasus uji memasukkan gambar dengan ukuran lebih dari 1 MB**

Kode Kasus Uji	PV_013
Nama Kasus Uji	Kasus uji memasukkan gambar dengan ukuran lebih dari 1MB.
Prosedur	1. <i>User</i> memasukkan foto dengan dengan ukuran file 2MB.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa <i>file</i> yang dipilih terlalu besar.
Hasil	Sistem menampilkan pesan kesalahan bahwa <i>file</i> yang dipilih terlalu besar.
Status	Valid

**Tabel 6.17 Kasus uji memasukkan gambar dengan ukuran lebih dari 1 MB**

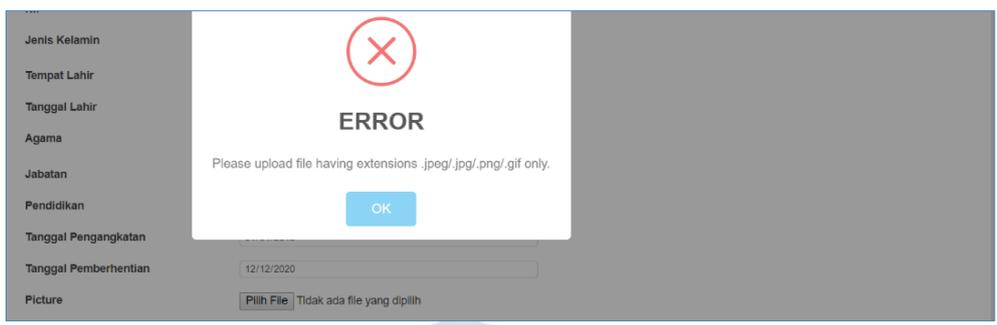
Kode Kasus Uji	PV_014
Nama Kasus Uji	Kasus uji <i>field</i> nama kosong.
Prosedur	1. <i>User</i> menekan tombol simpan tanpa <i>mengisi field</i> nama.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bidang nama harus diisi.
Hasil	Sistem menampilkan pesan kesalahan bidang nama harus diisi.
Status	Valid

**Tabel 6.18 Kasus uji *field* NIP dengan jumlah karakter kurang dari 16**

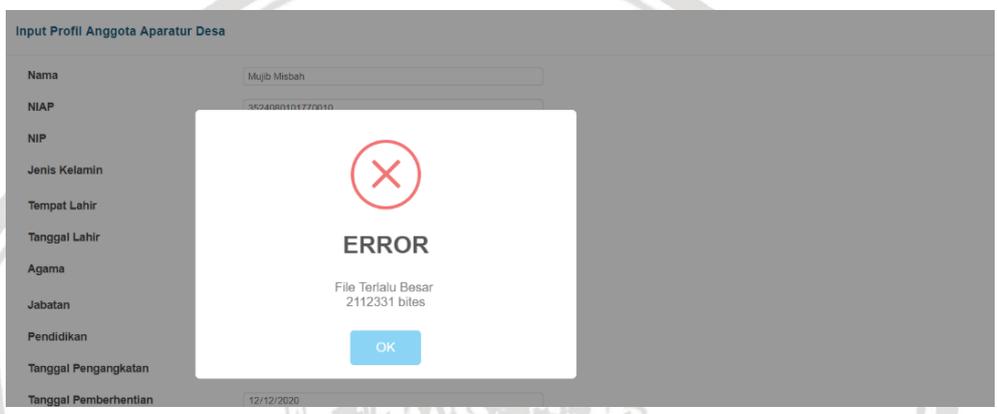
Kode Kasus Uji	PV_015
Nama Kasus Uji	Kasus uji <i>field</i> NIP dengan jumlah karakter kurang dari 16.
Prosedur	1. <i>User</i> menekan tombol simpan dengan memasukkan angka 35 ke dalam bidang NIP.
Hasil yang diharapkan	Sistem menampilkan pesan kesalahan bahwa masukan pada bidang NIP sekurang-kurangnya harus 16 karakter.
Hasil	Sistem menampilkan pesan kesalahan bahwa masukan pada bidang NIP sekurang-kurangnya harus 16 karakter.
Status	Valid



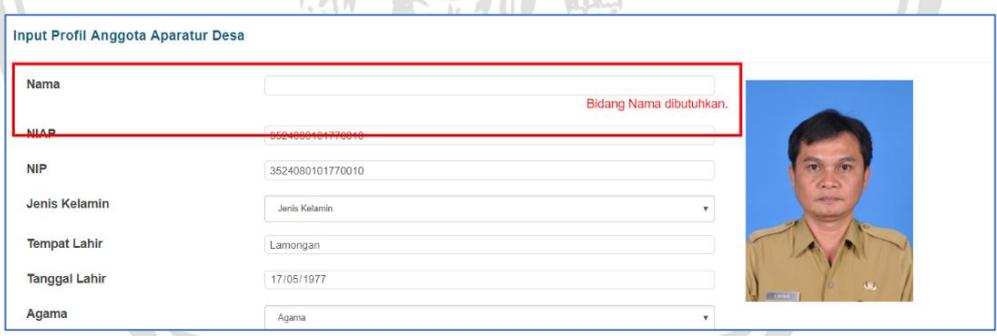
Gambar 6.23 Hasil pengujian kasus uji berhasil menambah perangkat



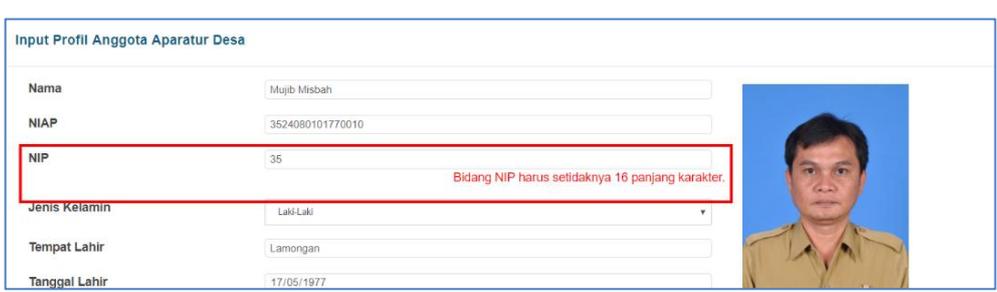
Gambar 6.24 Hasil pengujian kasus uji file dengan format mp3



Gambar 6.25 Hasil pengujian kasus uji gambar dengan ukuran 2MB



Gambar 6.26 Hasil pengujian kasus uji mengkosongkan bidang nama



**Gambar 6.27 Hasil pengujian kasus uji memberi 2 karakter pada bidang NIP**

6. Pengujian mengubah profil perangkat

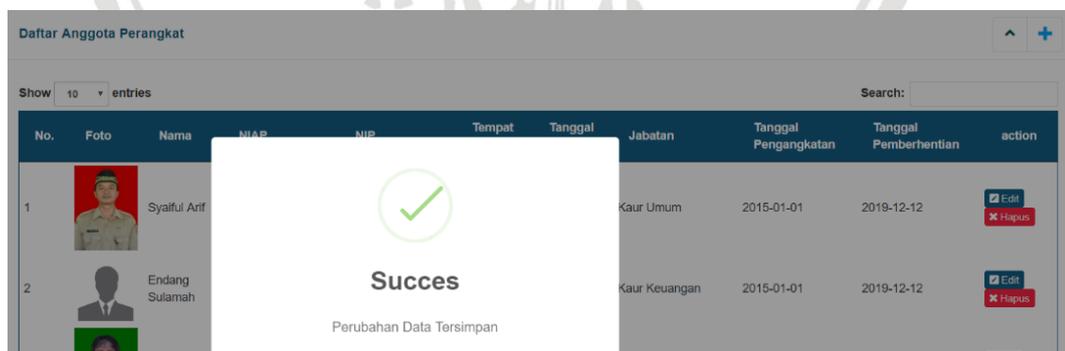
Beberapa kasus uji pada fungsional mengubah profil perangkat termuat pada tabel 6.19 dan 6.20. Sedangkan hasil uji ditunjukkan pada Gambar 6.28 dan 6.29 berikut ini.

**Tabel 6.19 Kasus uji berhasil mengubah profil**

<b>Kode Kasus Uji</b>	PV_016
<b>Nama Kasus Uji</b>	Kasus uji berhasil mengubah profil anggota perangkat desa
<b>Prosedur</b>	1. <i>User</i> melakukan perubahan dengan tepat dan menekan tombol simpan.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan bahwa perubahan data berhasil disimpan.
<b>Hasil</b>	Sistem menampilkan pesan bahwa perubahan data berhasil disimpan.
<b>Status</b>	Valid

**Tabel 6.20 Kasus uji menekan menekan tombol batal**

<b>Kode Kasus Uji</b>	PV_017
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol batal.
<b>Prosedur</b>	1. <i>User</i> menekan tombol batal pada halaman <i>edit</i> profil anggota perangkat.
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan konfirmasi dengan pilihan OK dan No.
<b>Hasil</b>	Sistem menampilkan pesan konfirmasi dengan pilihan OK dan No.
<b>Status</b>	Valid



**Gambar 6.28 Hasil pengujian kasus uji berhasil mengubah profil perangkat**



Gambar 6.29 Hasil pengujian kasus uji menekan tombol batal

### 7. Pengujian menghapus profil perangkat desa

Beberapa kasus uji pada fungsional menghapus profil perangkat desa termuat pada tabel 6.21, 6.22 dan 6.23. Sedangkan hasil uji ditunjukkan pada Gambar 6.30, 6.31 dan 6.32 berikut ini.

Tabel 6.21 Kasus uji menekan tombol hapus

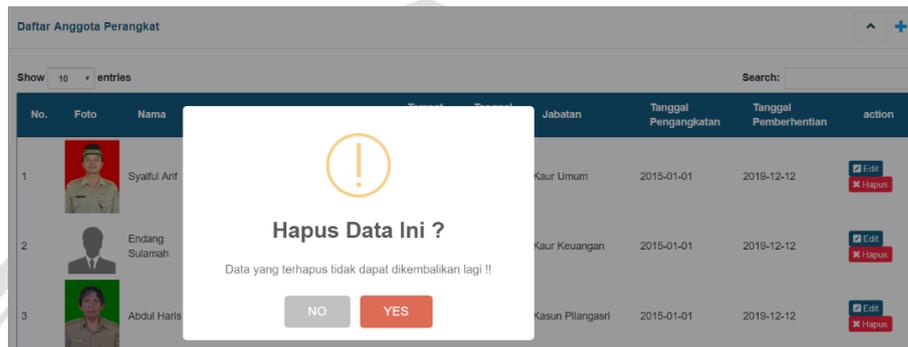
<b>Kode Kasus Uji</b>	PV_018
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol hapus
<b>Prosedur</b>	User menekan tombol hapus pada salah satu anggota pada daftar anggota perangkat
<b>Hasil yang diharapkan</b>	Sistem menampilkan pesan konfirmasi penghapusan dengan tombol Ya yang digunakan untuk eksekusi serta tombol Tidak untuk pembatalan proses hapus.
<b>Hasil</b>	Sistem menampilkan pesan konfirmasi penghapusan dengan tombol Ya yang digunakan untuk eksekusi serta tombol Tidak untuk pembatalan proses hapus.
<b>Status</b>	Valid

Tabel 6.22 Kasus uji menekan tombol Ya

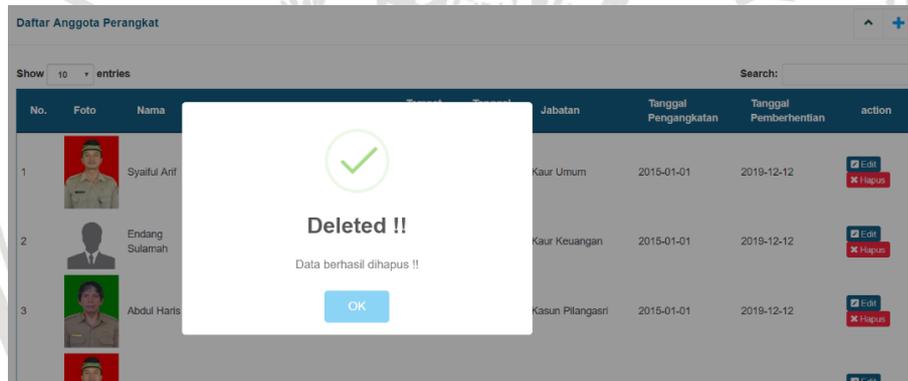
<b>Kode Kasus Uji</b>	PV_019
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol Ya
<b>Prosedur</b>	User menekan tombol Ya pada pesan konfirmasi hapus
<b>Hasil yang diharapkan</b>	Sistem menghapus data serta menampilkan pesan berhasil pada halaman daftar anggota perangkat
<b>Hasil</b>	Sistem menghapus data serta menampilkan pesan berhasil pada halaman daftar anggota perangkat
<b>Status</b>	Valid

Tabel 6.23 Kasus uji menekan tombol Tidak

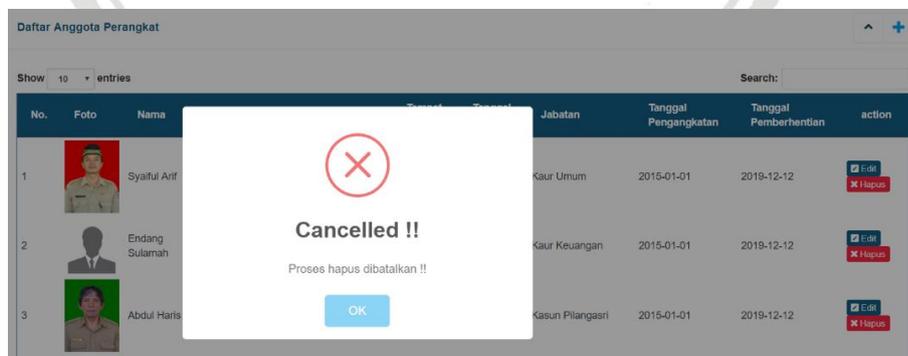
Kode Kasus Uji	PV_020
Nama Kasus Uji	Kasus uji menekan tombol hapus
Prosedur	User menekan tombol Tidak
Hasil yang diharapkan	Sistem menampilkan pesan bahwa proses penghapusan dibatal-kan
Hasil	Sistem menampilkan pesan bahwa proses penghapusan dibatal-kan
Status	Valid



Gambar 6.30 Hasil pengujian kasus uji menekan tombol hapus



Gambar 6.31 Hasil pengujian kasus uji menekan tombol YES



Gambar 6.32 Hasil pengujian kasus uji menekan tombol NO

8. Pengujian menambah data *RKPD*

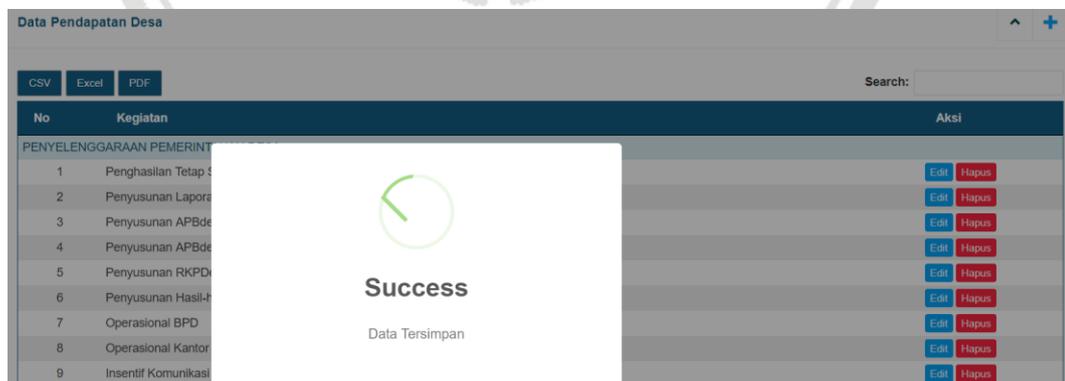
Beberapa kasus uji yang terdapat pada fungsional menambah data *RKPD* termuat pada tabel 6.24 dan 6.25. Sedangkan hasil uji ditunjukkan pada Gambar 6.33 dan 6.34 berikut ini.

**Tabel 6.24 Kasus uji berhasil menambah data *RKPD***

<b>Kode Kasus Uji</b>	PV_021
<b>Nama Kasus Uji</b>	Kasus uji berhasil menambah data <i>RKPD</i> .
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> memasukkan data profil dengan tepat dan benar.</li> <li>2. <i>User</i> memasukkan gambar dengan format jpg dan ukuran 500kb.</li> <li>3. <i>User</i> menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem menyimpan data ke dalam <i>database</i> dan menampilkan pesan berhasil pada halaman <i>RKPD</i> .
<b>Hasil</b>	Sistem menyimpan data ke dalam <i>database</i> dan menampilkan pesan berhasil pada halaman <i>RKPD</i> .
<b>Status</b>	Valid

**Tabel 6.25 Kasus uji gagal menambah data *RKPD***

<b>Kode Kasus Uji</b>	PV_022
<b>Nama Kasus Uji</b>	Kasus uji gagal menyimpan data <i>RKPD</i>
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> tidak memberikan masukan terhadap bidang kegiatan.</li> <li>2. <i>User</i> menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem kembali menampilkan <i>form</i> dengan pesan kesalahan bahwa bidang kegiatan harus diisi.
<b>Hasil</b>	Sistem kembali menampilkan <i>form</i> dengan pesan kesalahan bahwa bidang kegiatan harus diisi.
<b>Status</b>	Valid



**Gambar 6.33 Hasil pengujian kasus uji berhasil menambah data *RKPD***

**Gambar 6.34 Hasil pengujian kasus uji gagal menambah data *RKPD***

9. Pengujian menghapus data *RKPD*

Beberapa kasus uji yang terdapat pada fungsional hapus data *RKPD* termuat pada tabel 6.26, 6.27 dan 6.28. Sedangkan hasil uji ditunjukkan pada Gambar 6.35, 6.36 dan 6.37 berikut ini.

**Tabel 6.26 Kasus uji menekan tombol hapus**

<b>Kode Kasus Uji</b>	PV_023
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol hapus
<b>Prosedur</b>	User menekan tombol hapus pada salah satu data pada daftar <i>RKPD</i>
<b>Hasil yang diharapkan</b>	Sistem kembali menampilkan pesan konfirmasi dengan tombol Ya untuk proses eksekusi, serta tombol Tidak untuk membatalkan proses hapus.
<b>Hasil</b>	Sistem kembali menampilkan pesan konfirmasi dengan tombol Ya untuk proses eksekusi, serta tombol Tidak untuk membatalkan proses hapus.
<b>Status</b>	Valid

**Tabel 6.27 Kasus uji menekan tombol Ya**

<b>Kode Kasus Uji</b>	PV_024
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol Ya
<b>Prosedur</b>	User menekan tombol ya pada pesan konfirmasi hapus data <i>RKPD</i>
<b>Hasil yang diharapkan</b>	Sistem mengeksekusi proses hapus data dan menampilkan pesan berhasil pada halaman daftar <i>RKPD</i> .
<b>Hasil</b>	Sistem mengeksekusi proses hapus data dan menampilkan pesan berhasil pada halaman daftar <i>RKPD</i> .
<b>Status</b>	Valid

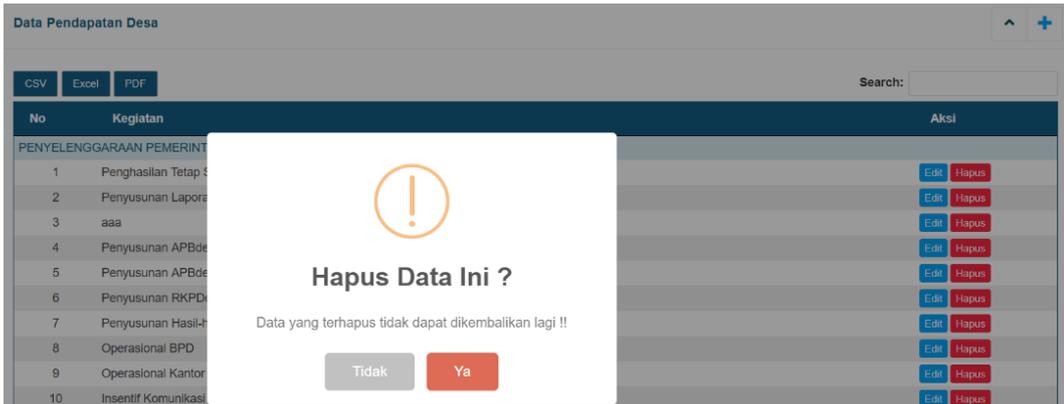
**Tabel 6.28 Kasus uji menekan tombol Tidak**

<b>Kode Kasus Uji</b>	PV_025
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol Tidak
<b>Prosedur</b>	User menekan tombol Tidak pada pesan konfirmasi hapus data <i>RKPD</i>

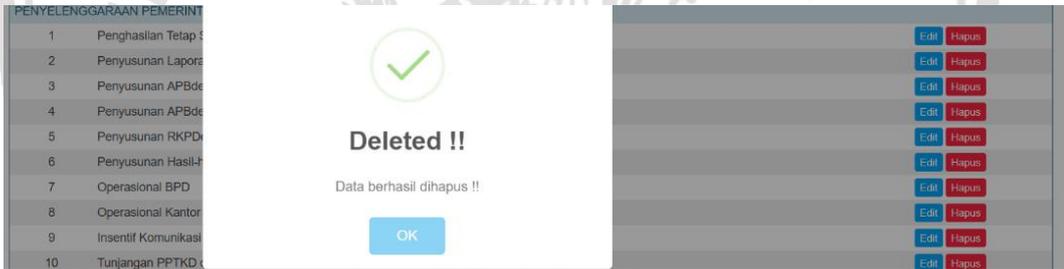


Tabel 6.28 Kasus uji menekan tombol Tidak (lanjutan)

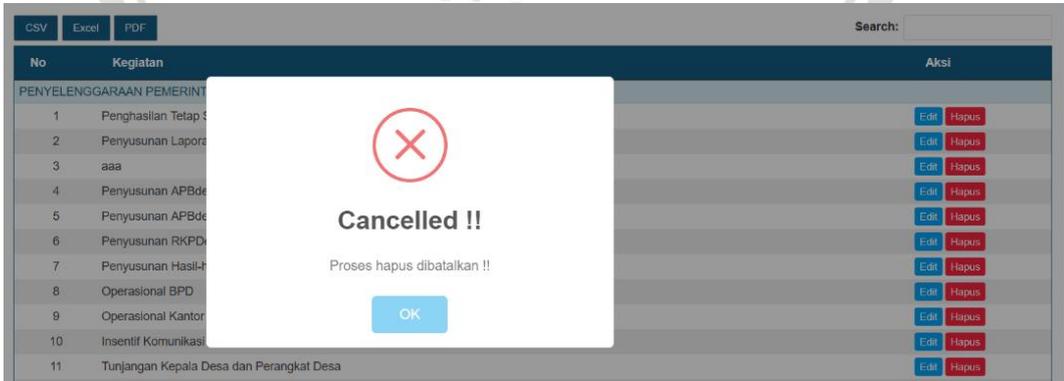
<b>Hasil yang diharapkan</b>	Sistem tidak menghapus data dan menampilkan pesan batal pada halaman daftar <i>RKPD</i> .
<b>Hasil</b>	Sistem tidak menghapus data dan menampilkan pesan batal pada halaman daftar <i>RKPD</i> .
<b>Status</b>	Valid



Gambar 6.35 Hasil pengujian validasi fungsional hapus *RKPD*



Gambar 6.36 Hasil pengujian validasi fungsional hapus *RKPD*



Gambar 6.37 Hasil pengujian validasi fungsional hapus *RKPD*

10. Pengujian menyimpan data *RKPD*

Beberapa kasus uji yang terdapat pada fungsional menyimpan data *RKPD* termuat pada tabel 6.29, 6.30, dan 6.31. Sedangkan hasil uji ditunjukkan pada Gambar 6.38, 6.39 dan 6.40 berikut ini.

**Tabel 6.29 Kasus uji menekan tombol csv**

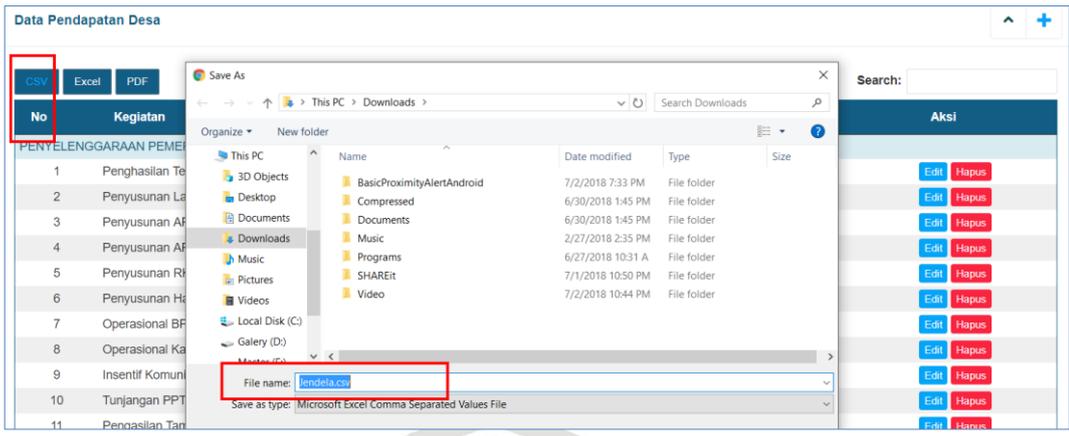
<b>Kode Kasus Uji</b>	PV_026
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol csv
<b>Prosedur</b>	User menekan tombol csv pada halaman daftar <i>RKPD</i>
<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>file</i> dialog untuk menyimpan <i>file RKPD</i> dengan format csv.
<b>Hasil</b>	Sistem menampilkan <i>file</i> dialog untuk menyimpan <i>file RKPD</i> dengan format csv.
<b>Status</b>	Valid

**Tabel 6.30 Kasus uji menekan tombol excel**

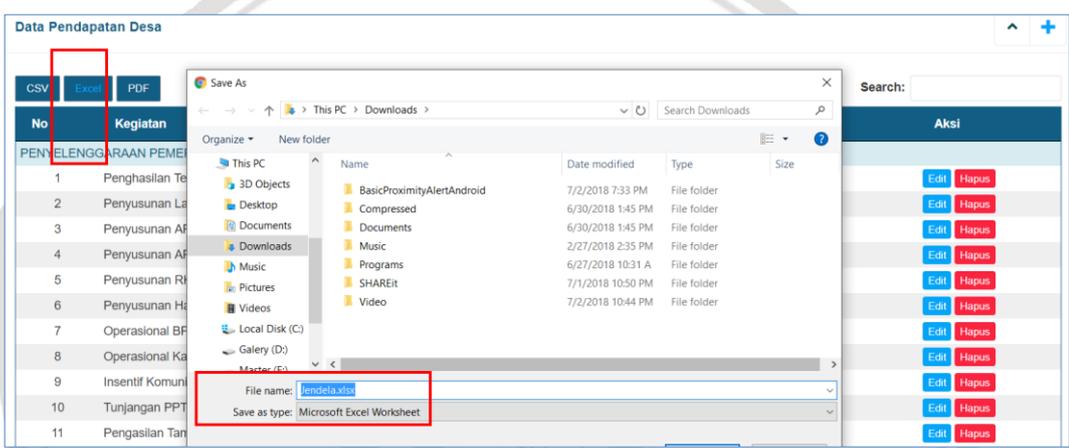
<b>Kode Kasus Uji</b>	PV_027
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol excel
<b>Prosedur</b>	User menekan tombol excel pada halaman daftar <i>RKPD</i>
<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>file</i> dialog untuk menyimpan <i>file RKPD</i> dengan format xlsx.
<b>Hasil</b>	Sistem menampilkan <i>file</i> dialog untuk menyimpan <i>file RKPD</i> dengan format xlsx.
<b>Status</b>	Valid

**Tabel 6.31 Kasus uji menekan tombol PDF**

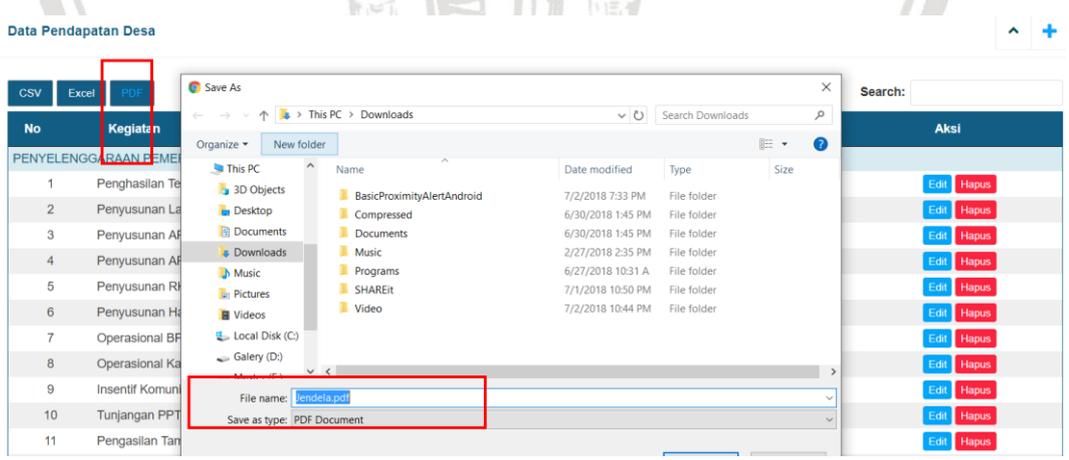
<b>Kode Kasus Uji</b>	PV_028
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol PDF
<b>Prosedur</b>	User menekan tombol PDF pada halaman daftar <i>RKPD</i>
<b>Hasil yang diharapkan</b>	Sistem menampilkan <i>file</i> dialog untuk menyimpan <i>file RKPD</i> dengan format PDF.
<b>Hasil</b>	Sistem menampilkan <i>file</i> dialog untuk menyimpan <i>file RKPD</i> dengan format PDF.
<b>Status</b>	Valid



Gambar 6.38 Hasil pengujian validasi fungsional simpan RKPD



Gambar 6.39 Hasil pengujian validasi fungsional simpan RKPD



Gambar 6.40 Hasil pengujian validasi fungsional simpan RKPD



## 11. Pengujian memesan surat

Beberapa kasus uji yang terdapat pada fungsional memesan surat termuat pada tabel 6.32 dan 6.33. Sedangkan hasil uji ditunjukkan pada Gambar 6.41 dan 6.42 berikut ini.

**Tabel 6.32 Kasus uji berhasil memesan surat**

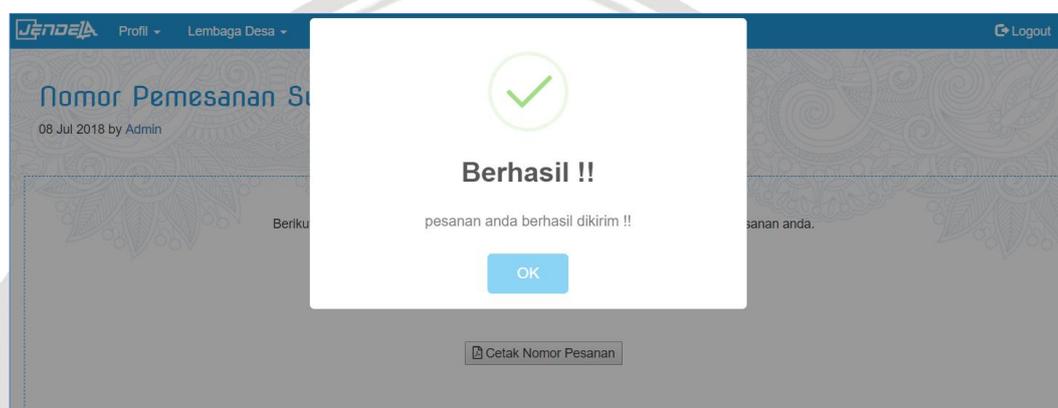
<b>Kode Kasus Uji</b>	PV_029
<b>Nama Kasus Uji</b>	Kasus uji berhasil memesan surat pengantar
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> memasukkan nomor telepon : 081336498565</li> <li>2. <i>User</i> memasukkan NIK : 3524080809720000</li> <li>3. <i>User</i> memasukkan nama : Jupri</li> <li>4. <i>User</i> memasukkan tempat lahir : tuban</li> <li>5. <i>User</i> memasukkan tanggal lahir : 1972-09-08</li> <li>6. <i>User</i> memilih jenis kelamin : Laki Laki</li> <li>7. <i>User</i> memasukkan pekerjaan : Wiraswasta</li> <li>8. <i>User</i> memasukkan agama : Islam</li> <li>9. <i>User</i> memasukkan RT dan RW : 02/03</li> <li>10. <i>User</i> memasukkan keterangan : untuk membuat SKCK</li> </ol> <p><i>User</i> menekan tombol pesan</p>
<b>Hasil yang diharapkan</b>	Sistem memasukkan pesanan ke dalam <i>database</i> dan menampilkan nomor pesanan serta pesan berhasil
<b>Hasil</b>	Sistem memasukkan pesanan ke dalam <i>database</i> dan menampilkan nomor pesanan serta pesan berhasil
<b>Status</b>	Valid

**Tabel 6.33 Kasus uji gagal memesan surat**

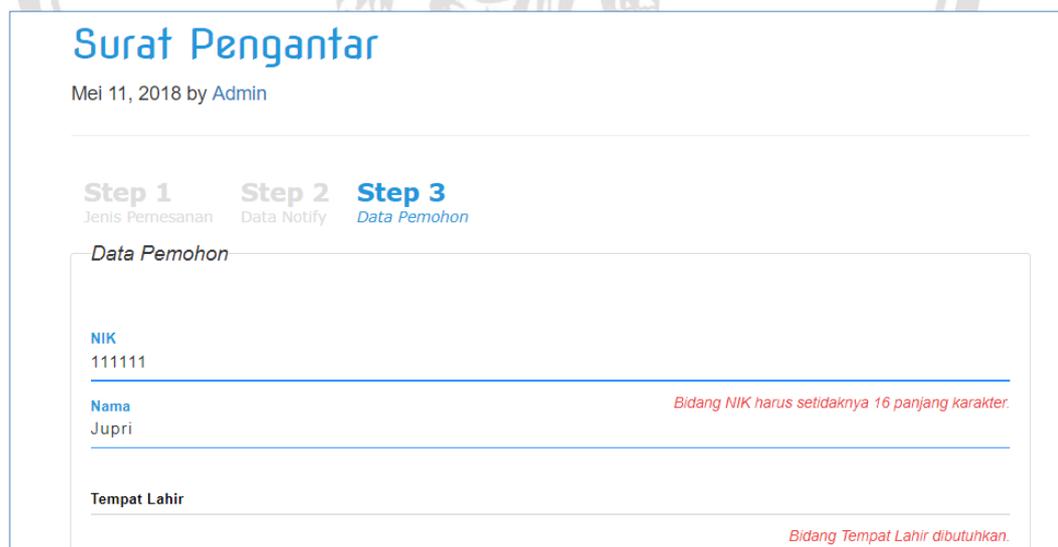
<b>Kode Kasus Uji</b>	PV_030
<b>Nama Kasus Uji</b>	Kasus uji berhasil memesan surat pengantar
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>User</i> memasukkan nomor telepon : 081336498565</li> <li>2. <i>User</i> memasukkan NIK : 111111</li> <li>3. <i>User</i> memasukkan nama : Jupri</li> <li>4. <i>User</i> tidak memberi nilai pada bidang tempat lahir</li> <li>5. <i>User</i> memasukkan tanggal lahir : 1972-09-08</li> <li>6. <i>User</i> memilih jenis kelamin : Laki Laki</li> <li>7. <i>User</i> memasukkan pekerjaan : Wiraswasta</li> <li>8. <i>User</i> memasukkan agama : Islam</li> <li>9. <i>User</i> memasukkan RT dan RW : 02/03</li> </ol>

**Tabel 6.33 Kasus uji gagal memesan surat (lanjutan)**

	<p>10. <i>User</i> memasukkan keterangan : untuk membuat SKCK</p> <p>11. <i>User</i> menekan tombol pesan</p>
<b>Hasil yang diharapkan</b>	Sistem menampilkan halaman pemesanan dengan pesan kesalahan bahwa bidang NIK setidaknya 16 karakter serta bidang tempat lahir tidak boleh kosong.
<b>Hasil</b>	Sistem menampilkan halaman pemesanan dengan pesan kesalahan bahwa bidang NIK setidaknya 16 karakter serta bidang tempat lahir tidak boleh kosong.
<b>Status</b>	Valid



**Gambar 6.41 Hasil pengujian validasi fungsional memesan surat pengantar**



**Gambar 6.42 Hasil pengujian validasi fungsional memesan surat pengantar**

## 12. Pengujian mengubah status pesanan

Beberapa kasus uji yang terdapat pada fungsional mengubah status pesanan termuat pada tabel 6.34, 6.35, dan 6.36. Sedangkan hasil uji ditunjukkan pada Gambar 6.43, 6.44 dan 6.45 berikut ini.

**Tabel 6.34 Kasus uji menekan tombol ubah status**

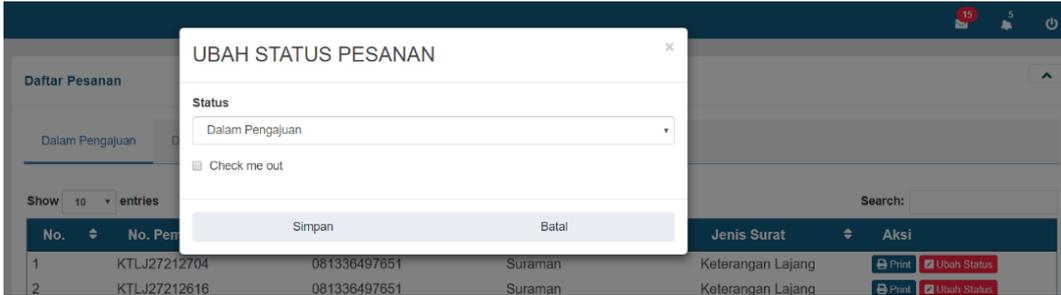
<b>Kode Kasus Uji</b>	PV_031
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol ubah status
<b>Prosedur</b>	<i>User</i> menekan tombol ubah status pada salah satu pesanan pada daftar pesanan surat
<b>Hasil yang diharapkan</b>	Sistem menampilkan pop up <i>form</i> perubahan status.
<b>Hasil</b>	Sistem menampilkan pop up <i>form</i> perubahan status.
<b>Status</b>	Valid

**Tabel 6.35 Kasus uji menekan tombol save**

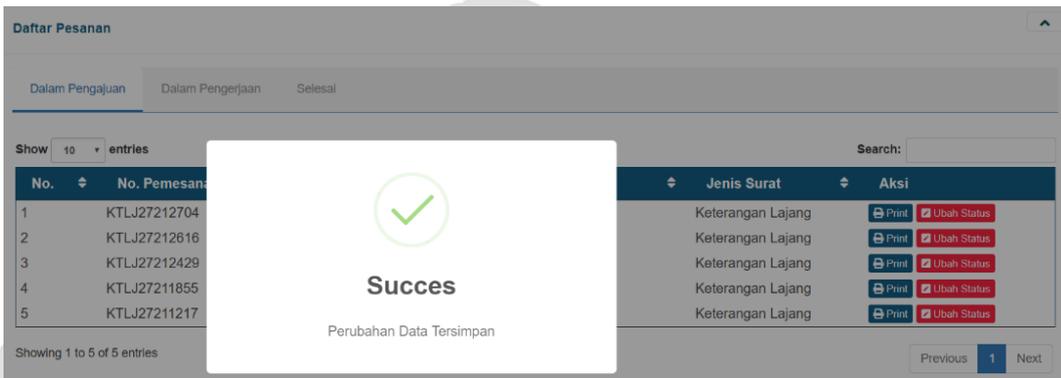
<b>Kode Kasus Uji</b>	PV_032
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol save
<b>Prosedur</b>	1. <i>User</i> memilih status 2. <i>User</i> mencentang tombol bersetujuan <i>User</i> menekan tombol save
<b>Hasil yang diharapkan</b>	Sistem mengubah status sesuai dengan yang dipilih <i>user</i> dan menampilkan pesan berhasil.
<b>Hasil</b>	Sistem mengubah status sesuai dengan yang dipilih <i>user</i> dan menampilkan pesan berhasil.
<b>Status</b>	Valid

**Tabel 6.36 Kasus uji menekan tombol batal**

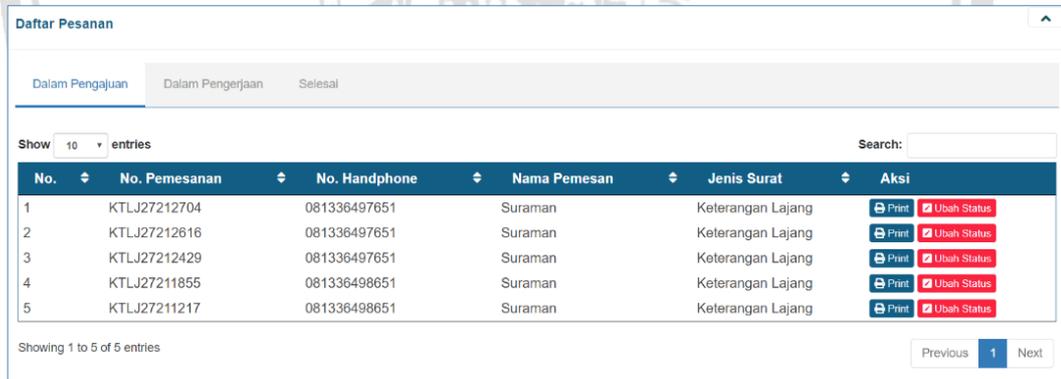
<b>Kode Kasus Uji</b>	PV_033
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol batal
<b>Prosedur</b>	1. <i>User</i> memilih status 2. <i>User</i> mencentang tombol bersetujuan <i>User</i> menekan tombol batal
<b>Hasil yang diharapkan</b>	Sistem tidak melakukan perubahan terhadap status pesanan.
<b>Hasil</b>	Sistem tidak melakukan perubahan terhadap status pesanan.
<b>Status</b>	Valid



Gambar 6.43 Hasil pengujian validasi fungsional mengubah status pesanan



Gambar 6.44 Hasil pengujian validasi fungsional mengubah status pesanan



Gambar 6.45 Hasil pengujian validasi fungsional mengubah status pesanan

### 13. Pengujian mencetak surat

Beberapa kasus uji yang terdapat pada fungsional mencetak surat termuat pada tabel 6.37, 6.38, dan 6.39. Sedangkan hasil uji ditunjukkan pada Gambar 6.46, 6.47, dan 6.48 berikut ini.

**Tabel 6.37 Kasus uji menekan tombol *print***

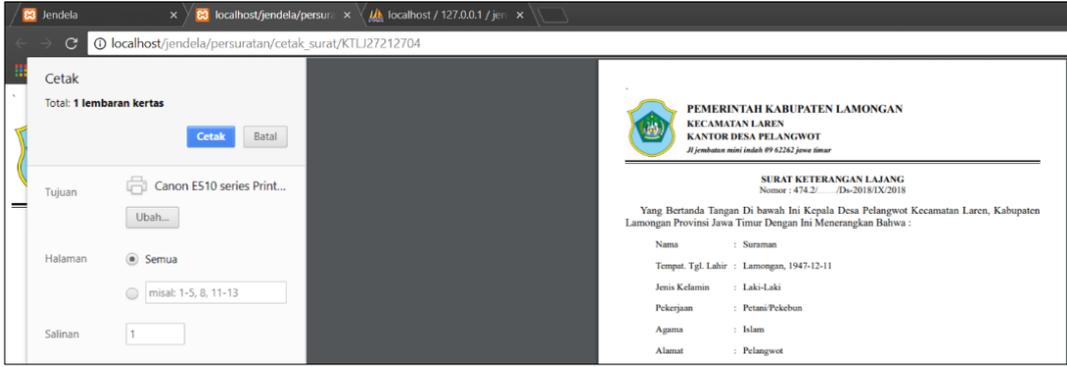
<b>Kode Kasus Uji</b>	PV_034
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol <i>print</i>
<b>Prosedur</b>	<i>User</i> menekan tombol <i>print</i> pada salah satu pesanan pada daftar pesanan surat.
<b>Hasil yang diharapkan</b>	Sistem tidak melakukan perubahan terhadap status pesanan.
<b>Hasil</b>	Sistem tidak melakukan perubahan terhadap status pesanan.
<b>Status</b>	Valid.

**Tabel 6.38 Kasus uji menekan tombol cetak**

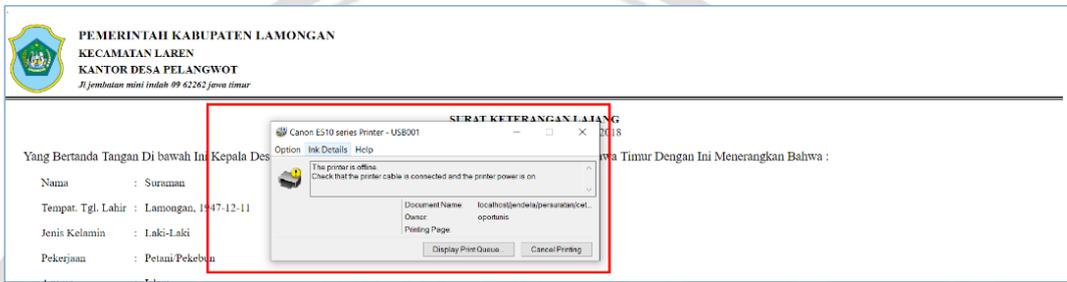
<b>Kode Kasus Uji</b>	PV_035
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol cetak
<b>Prosedur</b>	<i>User</i> menekan tombol cetak pada halaman <i>print</i> .
<b>Hasil yang diharapkan</b>	Sistem memasukkan dokumen pada antrean <i>print</i> .
<b>Hasil</b>	Sistem memasukkan dokumen pada antrean <i>print</i> .
<b>Status</b>	Valid.

**Tabel 6.39 Kasus uji menekan tombol batal**

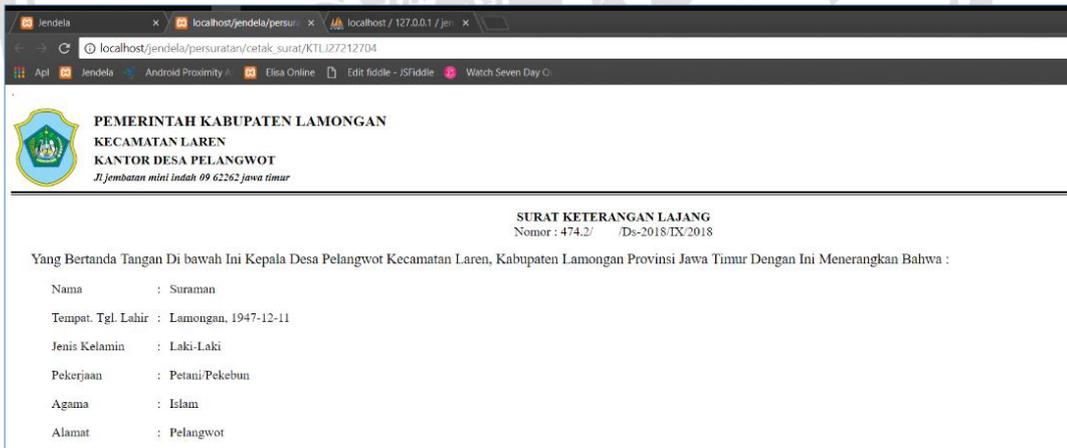
<b>Kode Kasus Uji</b>	PV_036
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol batal.
<b>Prosedur</b>	<i>User</i> menekan tombol batal pada halaman <i>print</i> .
<b>Hasil yang diharapkan</b>	Sistem menutup halaman <i>print</i> .
<b>Hasil</b>	Sistem menutup halaman <i>print</i> .
<b>Status</b>	Valid.



Gambar 6.46 Hasil pengujian validasi fungsional mencetak pesanan surat



Gambar 6.47 Hasil pengujian validasi fungsional mencetak pesanan surat



Gambar 6.48 Hasil pengujian validasi fungsional mencetak pesanan surat

#### 14. Pengujian mengirim pesan notifikasi

Beberapa kasus uji yang terdapat pada fungsional mengirim pesan notifikasi termuat pada tabel 6.40, 6.41 dan 6.42. Sedangkan hasil uji ditunjukkan pada Gambar 6.49, 6.50 dan 6.51 berikut ini.

**Tabel 6.40 Kasus uji menekan tombol kirim notifikasi**

<b>Kode Kasus Uji</b>	PV_037
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol kirim notifikasi
<b>Prosedur</b>	User menekan tombol kirim notifikasi pada salah satu pesan pada daftar pesanan surat
<b>Hasil yang diharapkan</b>	Sistem membuka halaman pesan notifikasi dengan nomor HP dan format pesan sesuai dengan jenis suratnya.
<b>Hasil</b>	Sistem membuka halaman pesan notifikasi dengan nomor HP dan format pesan sesuai dengan jenis suratnya.
<b>Status</b>	Valid

**Tabel 6.41 Kasus uji menekan tombol batal**

<b>Kode Kasus Uji</b>	PV_038
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol batal
<b>Prosedur</b>	User menekan tombol batal pada halaman pesan
<b>Hasil yang diharapkan</b>	Sistem kembali ke halaman daftar pesanan tanpa mengirim pesan notifikasi.
<b>Hasil</b>	Sistem kembali ke halaman daftar pesanan tanpa mengirim pesan notifikasi.
<b>Status</b>	Valid

**Tabel 6.42 Kasus uji menekan tombol kirim**

<b>Kode Kasus Uji</b>	PV_039
<b>Nama Kasus Uji</b>	Kasus uji menekan tombol kirim
<b>Prosedur</b>	User menekan tombol kirim pada halaman pesan
<b>Hasil yang diharapkan</b>	Sistem mengirim pesan notifikasi dan menampilkan pesan berhasil pada halaman daftar pesanan surat.
<b>Hasil</b>	Sistem mengirim pesan notifikasi dan menampilkan pesan berhasil pada halaman daftar pesanan surat.
<b>Status</b>	Valid

### Send Notify

**No Hp**

**Pesan**

Pesanan surat Keterangan Belum Menikah anda sudah selesai, dan dapat diambil dengan membawa KTP.

Kirim
Batal

Gambar 6.49 Hasil pengujian validasi fungsional mengirim notifikasi

Daftar Pesanan

Dalam Pengajuan   Dalam Pengerjaan   Selesai

Show 10 entries   Search:

No.	No. Pemesanan	No. Handphone	Nama Pemesan	Jenis Surat	Aksi
1	KTLJ27212704	081336497651	Suraman	Keterangan Lajang	<a href="#">Print</a> <a href="#">Ubah Status</a>
2	KTLJ27212616	081336497651	Suraman	Keterangan Lajang	<a href="#">Print</a> <a href="#">Ubah Status</a>
3	KTLJ27212429	081336497651	Suraman	Keterangan Lajang	<a href="#">Print</a> <a href="#">Ubah Status</a>
4	KTLJ27211855	081336498651	Suraman	Keterangan Lajang	<a href="#">Print</a> <a href="#">Ubah Status</a>
5	KTLJ27211217	081336498651	Suraman	Keterangan Lajang	<a href="#">Print</a> <a href="#">Ubah Status</a>

Showing 1 to 5 of 5 entries   Previous 1 Next

Gambar 6.50 Hasil pengujian validasi fungsional mengirim notifikasi

Berita dan Artikel

Pengajuan   Pengerjaan   Selesai

Show 10 entries

No.	Jenis Surat	Waktu Pemesanan
1	Surat Pengantar	2018-07-06 00:00:00
2	Keterangan Peng	2018-07-06 00:00:00
3	Keterangan Tidak	2018-07-06 00:00:00
4	Keterangan Pinda	2018-07-06 00:00:00
5	Keterangan Usaha	2018-07-06 00:00:00
6	Pengajuan Kartu	2018-07-08 15:07:00
7	Surat Pengantar	2018-07-08 20:07:00



**Success**

Pesan Berhasil Dikirim

Showing 1 to 7 of 7 entries

Gambar 6.51 Hasil pengujian validasi fungsional mengirim notifikasi



### 6.3 Pengujian *Compatibility*

Pengujian *compatibility* dilakukan dengan menggunakan *tools* buatan powerMapper bernama SortSite, alat ini dapat digunakan untuk memeriksa format HTML, *script*, dan gambar pada berbagai *browser*. Gambar 6.52 berikut ini merupakan hasil pengujian *compatibility* dengan menggunakan *tool* SortSite.

Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS	Android	BlackBerry
Version	11	16	60	11	51	66	11	4*	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓

\* Most Android devices from 4.1 onwards use Chrome as the default browser

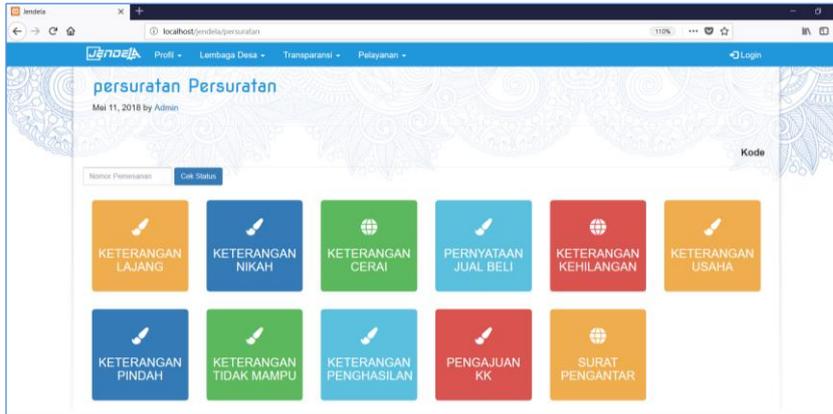
Gambar 6.52 Hasil Pengujian *Compatibility* menggunakan aplikasi *SortSite*

Berdasarkan Gambar 6.52 maka dapat dikatakan *system* berjalan dengan baik pada berbagai *browser* seperti Edge, Firefox, Opera, Chrome hingga perangkat mobile. Gambar 6.53 hingga 6.57 berikut ini tampilan sistem ketika dibuka dengan menggunakan *browser* yang berbeda.

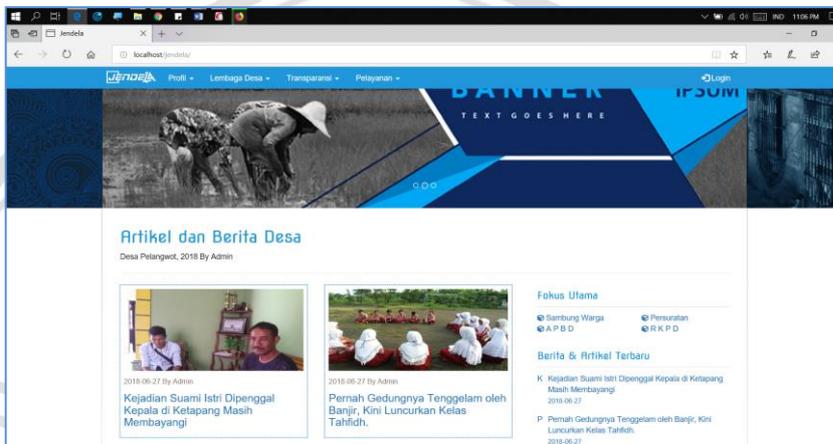


Gambar 6.53 Tampilan sistem dengan google chrome





Gambar 6.54 Tampilan sistem dengan mozilla firefox



Gambar 6.55 Tampilan dengan microsoft edge



Gambar 6.56 Tampilan Website pada perangkat mobile



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan Rumusan masalah pada penelitian ini, maka didapatkan kesimpulan bahwa proses pengembangan sistem dilakukan dengan menggunakan model *waterfall* yang secara bertahap meliputi proses analisis, perancangan, implementasi dan pengujian.

Pada tahap analisis didapatkan empat jenis pengguna yaitu *guest*, masyarakat, petugas dan kepala desa. Sedangkan kebutuhan yang didapatkan yaitu 73 kebutuhan fungsional dan 1 kebutuhan non fungsional. Proses perancangan dilakukan dengan memodelkan sistem ke dalam *sequence diagram*, *conceptual data model*, *class diagram*, dan *High Fidelity Mockup*. Sedangkan proses implementasi dilakukan dengan menggunakan bahasa pemrograman php dan *framework* Codeigniter. Implementasi antarmuka dikerjakan dengan menggunakan HTML dengan *framework* bootstrap serta beberapa *plugin* pendukung javascript.

Pengujian dilakukan pada level unit dan validasi. Selain itu terdapat pula pengujian *compatibility* yang ditujukan untuk menguji kebutuhan non fungsional. Pengujian unit dilakukan pada tiga fungsional dan didapatkan nilai kompleksitas sebanyak tiga pada masing-masing sampel uji. Pengujian validasi dilakukan terhadap 14 fungsional dengan kasus uji berjumlah 38. Sedangkan pengujian *compatibility* dilakukan dengan menggunakan aplikasi bernama sortSite dan didapatkan hasil bahwa tidak ada permasalahan sistem pada browser versi terakhir dari google chrome, mozilla, internet explore, microsoft edge, hingga *browser* pada perangkat mobile.

### 7.2 Saran

Saran yang diberikan penulis untuk penelitian mengenai sistem informasi administrasi desa selanjutnya yaitu pengembangan sistem dapat dilakukan terhadap buku administrasi secara menyeluruh serta adanya fitur pemesanan yang dapat dilakukan melalui *sms gateway*.

## DAFTAR PUSTAKA

- Bjørner, D., 2006. *Software Engineering 3: Domains, Requirements, and Software Design*. German: Springer.
- Desyatnikov, R., 2018. *What is Software Compatibility Testing?* [daring] Tersedia pada: <<https://www.softwaretestinghelp.com/software-compatibility-testing/>> [Diakses 7 Agu 2018].
- Gustafson, D., 2002. *Schaum's outline software engineering*. Manhattan: McGRAW-HILL.
- Hidayatulloh, S. dan Mulyadi, C., 2015. Sistem Pelayanan Administrasi Kependudukan Desa Candigatak Berbasis Web. *Jurnal IT CIDA*, 1(1), hal.42–55.
- Katankar, V.K. dan Thakare, V.M., 2010. Short Message Service using SMS Gateway. *International Journal*, 02(04), hal.1487–1491.
- Kementrian Dalam Negeri, 2016. *Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 47 Tahun 2016*. Tersedia pada: <<http://www.kemendagri.go.id/produk-hukum/2016/08/23/administrasi-pemerintahan-desa>>.
- Khan, M.E., 2011a. Different Approaches To Black Box Testing Techniques For Finding Errors. *International Journal of Softwar Engineering & Applications*, 2(4), hal.31–40.
- Khan, M.E., 2011b. Different Approaches To White Box Testing Techniques For Finding Errors. *International Journal of Software Engineering and its Applications*, 5(3), hal.1–14.
- Pressman, R.S. dan Maxim, B.R., 2015. *Software engineering: A Practitioner' S Approach*. Eight ed. New York: Raghu Srinivasan.
- Sommerville, I., 2010. *Software Engineering*. 9th ed. United States of America: Addison Wesley.
- Wammu, 2018. *Gammu and Wammu*. [daring] Tersedia pada: <<https://wammu.eu/gammu/>> [Diakses 22 Jul 2018].
- Widjaja, H.A., 2002. *Pemerintahan Desa dan Administrasi Desa*. Jakarta: PT RajaGrafindo Persada.
- Wiharto, Y., 2011. Sistem Indormasi Akademik Berbasis SMS Gateway. *Jurnal Teknologi dan Informatika*, 1(1), hal.1–28.