

**Sistem Informasi Penyuluhan Pertanian di Dinas Pertanian
dan Ketahanan Pangan Kabupaten Padang Pariaman**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Surya Daren Hafizh
NIM: 155150201111251



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

PENGESAHAN

SISTEM INFORMASI PENYULUHAN PERTANIAN DI DINAS PERTANIAN DAN
KETAHANAN PANGAN KABUPATEN PADANG PARIAMAN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Surya Daren Hafizh
NIM: 155150201111251

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

Fajar Pradana, S.ST, M.Eng
NIP: 198711212015041004

Adam Hendra Brata, S.T, M.T, M.Sc
NIK: 2016079001051001

Mengetahui
Ketua Jurusan Teknik Informatika

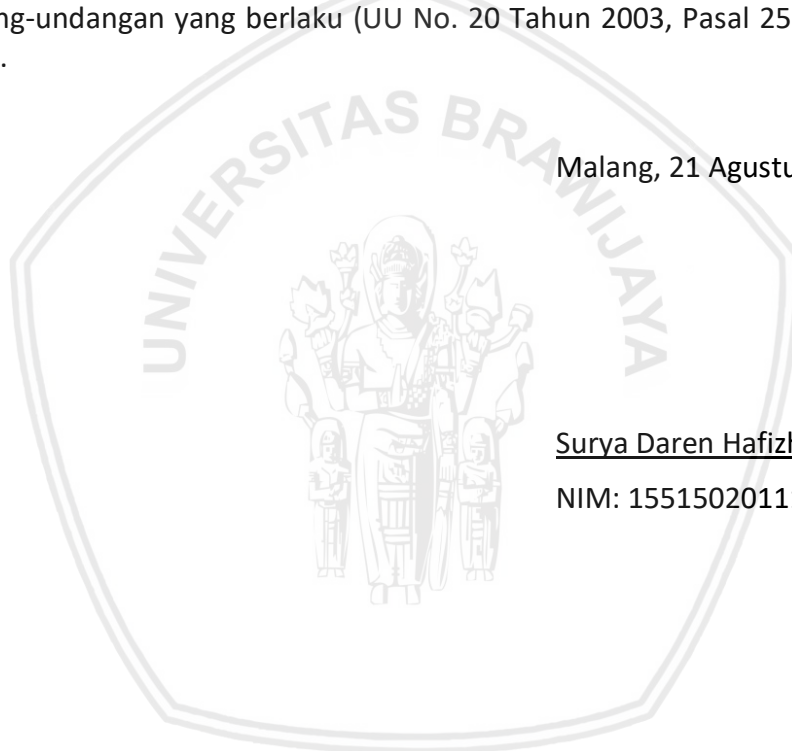
Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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, 21 Agustus 2019



Surya Daren Hafizh

NIM: 155150201111251

PRAKATA

Puji dan syukur penulis panjatkan bagi Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “SISTEM INFORMASI PENYULUHAN PERTANIAN DI DINAS PERTANIAN DAN KETAHANAN PANGAN KABUPATEN PADANG PARIAMAN” .

Penulis turut menyampaikan rasa terima kasih kepada pihak - pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Kepada kedua orang tua penulis, yaitu bapak Musadar, dan ibu Ernawati selalu memberikan segala masukan, do'a, motivasi dan semangat yang tidak terputus.
2. Bapak Fajar Pradana, S.ST,M.Eng selaku dosen pembimbing I dan Bapak Adam Hendra Barata, S.T, M.T, M.Sc selaku dosen pembimbing II yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya Malang.
4. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
5. Kepada saudara-saudari tercinta, Ayang ira, uni ari, uda lucky, uni meiri, uni ranti, dan refan. Yang senantiasa memberikan doa, dukungan, motivasi kepada penulis.
6. Kepada Keluarga besar yaitu, nenek, paman, bibi, ipar, dan ponakan yang juga telah senantiasa memberikan doa dan bantuan berupa moril maupun materi.
7. Kepada Rahmi Lutari, S.Ds yang telah senantiasa mebantu, mendoakan, memotivasi dan semangat yang tidak putus.
8. Kepada teman dan sahabat penulis yang juga telah memberikan doa dan bantuan kepada penulis, terutama CL Family selaku teman seperjuangan.
9. Semua pihak yang tidak semuanya bisa dituliskan di sini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi maupun sebagai pemberi semangat dan motivasi.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan penelitian skripsi ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 8 Juli 2019

Penulis

Suryadaren7@gmail.com



ABSTRAK

Dinas pertanian dan ketahanan pangan Kabupaten Padang Pariaman merupakan unsur pelaksana urusan pemerintahan di bidang pertanian yang mempunyai tanggung jawab dalam meningkatkan kualitas pertanian di Padang Pariaman. Untuk itu diadakannya sebuah kegiatan penyuluhan untuk meningkatkan kualitas pertanian. Kegiatan penyuluhan ini melibatkan penyuluh lapangan, kepala BPP, pegawai dinas dan kepala dinas. Sebelum kegiatan di laksanakan maka diperlukan sebuah draft programa yang berisi rancangan kegiatan satu tahun kedepan. Setiap kegiatan penyuluh lapangan mempunyai kewajiban untuk mengumpulkan laporan kegiatan untuk mempertanggung jawabkan hasil kerja dan pemakaian dana. Dalam membuat draft programa dan laporan kegiatan sampai pengesahan membutuhkan waktu yang lama. Dari permasalahan tersebut dibuatlah solusi dengan membangun sebuah sistem informasi penyuluhan pertanian dengan metode *SDLC waterfall*. Pada tahap analisis dan perancangan dibangun model yang dapat mempermudah komunikasi antara pengembang dan *stakeholder* demi menemukan solusi terbaik. Model yang dibangun pada tahap analisis dan perancangan yaitu *use case diagram*, *use case scenario*, *sequence diagram*, dan *class diagram*. Sistem yang telah dibangun akan diuji dengan *metode whitebox testing* untuk pengujian yang ditekankan pada kode program dan *blackbox testing* untuk pengujian pad fungsionalitas. Hasil dari pengujian mendekati 100% valid menggambarkan bahwa sistem dapat berjalan dengan baik dan dapat membantu mempermudah kegiatan penyuluhan pertanian.

Kata Kunci : *penyuluhan pertanian, sistem informasi, SDLC waterfall*

ABSTRACT

The Department of Agriculture and Food Security of the District of Padang Pariaman is an element of the government that focuses on implementing agricultural matters whose responsibilities encompasses increasing the quality of agriculture in Padang Pariaman. Hence, training activities were planned encompasses increasing the quality of agriculture. The training activities involve field counselor, the chairman of BPP, department staff and head of the department. Before the activity is carried out, a program draft must be formed containing the plan and the design of the activities for next year. For every activity, the field counselors have an obligation to submit the field report containing the results of the activity and the usage of available funds. The process of creating the program draft and report starting from planning and approval of the documents requiring a huge amount of time. To make the steps more efficient, a solution is proposed which is in the form of creating an information system for agricultural training events using an SDLC waterfall method. During the analysis and designing phase, a model is built to ease the communication between developers and developers in hopes of finding the best solution. The models built in the analysis and designing phase are the use case diagram, use case scenario, sequence diagram, and class diagram. The system that was built will then be tested using the whitebox testing and blackbox testing. The results from the testing showed was near 100% valid which described the system has worked well and can help improve agricultural training activities in the district.

Keywords: agricultural training, information system, SDLC waterfall

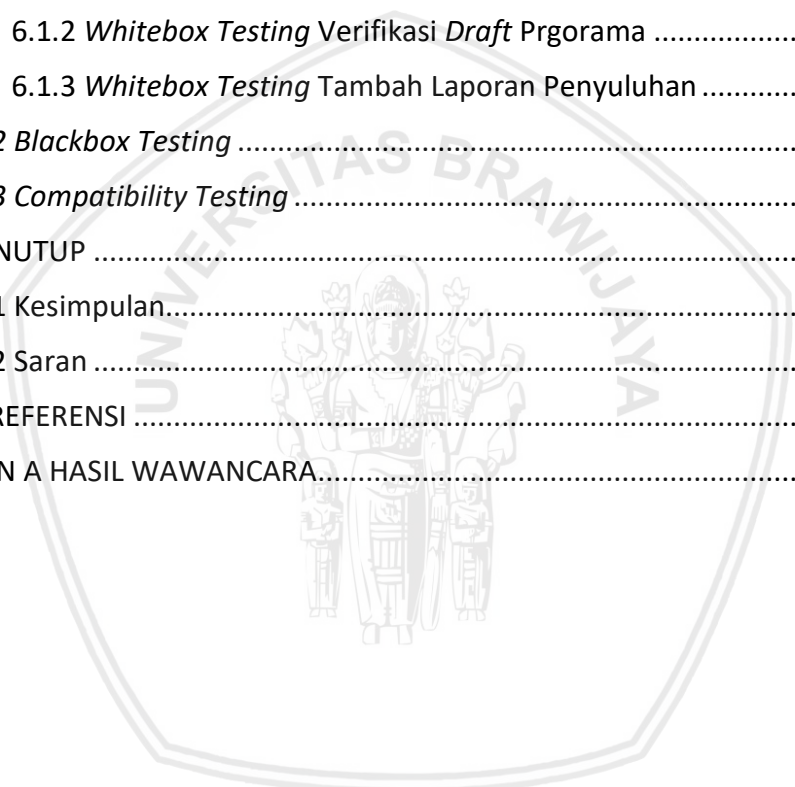
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Sistem Informasi	7
2.3 Penyuluhan Pertanian.....	7
2.4 SDLC	8
2.4.1 <i>SDLC Waterfall</i>	8
2.5 UML.....	9
2.5.1 <i>Use Case Diagram</i>	10
2.5.2 <i>Sequence Diagram</i>	10
2.5.3 <i>Class Diagram</i>	11
2.6 ERD	12
2.7 <i>Laravel</i>	13
2.8 Pengujian Perangkat Lunak.....	14
2.8.1 <i>Black-Box Testing</i>	14
2.8.2 <i>White-Box Testing</i>	14



2.8.3 <i>Compability Testing</i>	15
BAB 3 METODOLOGI	16
3.1 Studi Literatur	16
3.2 Pengumpulan Data	17
3.3 Analisis Kebutuhan Sistem.....	17
3.4 Perancangan Sistem.....	17
3.5 Implementasi Sistem	17
3.6 Pengujian	18
3.7 Kesimpulan dan Saran	18
BAB 4 ANALISIS KEBUTUHAN	19
4.1 Gambaran Umum Sistem.....	19
4.2 Proses Bisnis.....	19
4.2.1 Proses Bisnis Pembuatan Draft Programa	19
4.2.2 Proses Bisnis Pembuatan laporan Penyuluhan	21
4.2.3 Proses Bisnis Pengiriman Dana	22
4.3 Elisitasi Kebutuhan.....	23
4.4 Analisis Kebutuhan	23
4.4.1 Aturan Penomoran.....	23
4.4.2 Identifikasi Aktor	24
4.4.3 Analisis Kebutuhan Fungsional	25
4.4.4 Pembuatan <i>use case diagram</i>	39
4.4.5 Pembuatan <i>use case scenario</i>	40
4.4.6 Analisis kebutuhan <i>Non Fungsional</i>	65
BAB 5 PERANCANGAN & IMPLEMENTASI	67
5.1 Perancangan Perangkat Lunak	67
5.1.1 Perancangan <i>Sequence Diagram</i>	67
5.1.2 Perancangan <i>Class Diagram</i>	71
5.1.3 Perancangan Basis Data	75
5.1.4 Perancangan Komponen	75
5.1.5 Perancangan Antarmuka.....	79
5.2 Spesifikasi Sistem	81
5.2.1 Spesifikasi <i>Hardware</i>	82

5.2.2 Spesifikasi <i>Software</i>	82
5.3 Implementasi Sistem Informasi Penyuluhan	82
5.3.1 Implementasi Basis Data	83
5.3.2 Implementasi Algoritme.....	84
5.3.3 Implementasi Antar Muka	90
BAB 6 PENGUJIAN	95
6.1 <i>Whitebox Testing</i>	95
6.1.1 <i>Whitebox Testing</i> Tambah Draft Programa	95
6.1.2 <i>Whitebox Testing</i> Verifikasi Draft Prgorama	97
6.1.3 <i>Whitebox Testing</i> Tambah Laporan Penyuluhan	101
6.2 <i>Blackbox Testing</i>	104
6.3 <i>Compatibility Testing</i>	124
BAB 7 PENUTUP	126
7.1 Kesimpulan.....	126
7.2 Saran	127
DAFTAR REFERENSI	128
LAMPIRAN A HASIL WAWANCARA.....	130



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	6
Tabel 2.2 Notasi <i>Use Case</i> Diagram.....	10
Tabel 2.3 Komponen <i>Use Case Diagram</i>	11
Tabel 2.4 Komponen <i>Class Diagram</i>	12
Tabel 2.5 Komponen ERD.....	12
Tabel 2.6 Komponen <i>Flowgraph</i>	15
Tabel 4.1 Identifikasi Aktor	24
Tabel 4.2 Kebutuhan Fungsional.....	25
Tabel 4.3 <i>Use Case Login</i>	41
Tabel 4.4 <i>Use Case</i> Lihat Profil.....	42
Tabel 4.5 <i>Use Case</i> Ubah Data profil.....	42
Tabel 4.6 <i>Use Case</i> Lihat Daftar Notifikasi	43
Tabel 4.7 <i>Use Case</i> Lihat Detail Notifikasi	44
Tabel 4.8 <i>Use Case</i> Buat Draft Programa	45
Tabel 4.9 <i>Use Case</i> Masukan Jadwal Kegiatan.....	46
Tabel 4.10 <i>Use Case</i> Lihat Daftar <i>Draft</i> Programa	47
Tabel 4.11 <i>Use Case</i> Lihat Detail <i>Draft</i> Programa	48
Tabel 4.12 <i>Use Case</i> Verifikasi <i>Draft</i> Programa	49
Tabel 4.13 <i>Use Case</i> Lihat Jadwal dan Laporan	50
Tabel 4.14 <i>Use Case</i> Membuat Laporan Penyuluhan	50
Tabel 4.15 <i>Use Case</i> Lihat Detail Laporan penyuluhan.....	52
Tabel 4.16 <i>Use Case</i> Verifikasi Laporan Penyuluhan	52
Tabel 4.17 <i>Use Case</i> Ajukan Surat Tugas	53
Tabel 4.18 <i>Use Case</i> Lihat Daftar Surat Tugas.....	54
Tabel 4.19 <i>Use Case</i> Verifikasi Surat Tugas.....	55
Tabel 4.20 <i>Use Case</i> Menerbitkan Surat Tugas.....	56
Tabel 4.21 <i>Use Case</i> Ajukan Proposal Dana	57
Tabel 4.22 <i>Use Case</i> Lihat Daftar Proposal Dana	57
Tabel 4.23 <i>Use Case</i> Verifikas Proposal Dana	58
Tabel 4.24 <i>Use Case</i> Lihat Daftar Pengiriman Dana.....	59

Tabel 4.25 <i>Use Case</i> Lihat Detail Pengiriman Dana	59
Tabel 4.26 <i>Use Case</i> Lihat Konfirmasi Penerimaan Dana	60
Tabel 4.27 <i>Use Case</i> Lihat Daftar Pengguna	61
Tabel 4.28 <i>Use Case</i> Tambah Pengguna	62
Tabel 4.29 <i>Use Case</i> Lihat Data pengguna	63
Tabel 4.30 <i>Use Case</i> Edit Pengguna	63
Tabel 4.31 <i>Use Case</i> Hapus Data Pengguna	64
Tabel 4.32 <i>Use Case Logout</i>	65
Tabel 4.33 Kebutuhan Non Fungsional	65
Tabel 5.1 <i>Pseudocode</i> tambah <i>draft</i> programaTabel	76
Tabel 5.2 <i>Pseudocode</i> Verifikasi <i>Draft</i> Program.....	76
Tabel 5.3 <i>Pseudocode</i> Tambah Laporan Penyuluhan.....	78
Tabel 5.4 Spesifikasi <i>Hardware</i>	82
Tabel 5.5 Spesifikasi <i>Software</i>	82
Tabel 5.6 Implementasi algoritme tambah <i>draft</i> programa	84
Tabel 5.7 Implementasi algoritme varifikasi <i>draft</i> programa	86
Tabel 5.8 Implementasi algoritme tambah laporan penyuluhan	89
Tabel 6.1 Algoritme Tambah Draft Programa	95
Tabel 6.2 <i>Test Case</i> Tambah <i>Draft</i> Program.....	96
Tabel 6.3 Algoritma Verifikasi <i>Draft</i> Program.....	97
Tabel 6.4 <i>Test Case</i> Verifikasi Draft Program.....	100
Tabel 6.5 Algoritma Tambah Laporan Penyuluhan.....	101
Tabel 6.6 <i>Test Case</i> Tambah Laporan Penyuluhan	102
Tabel 6.7 Tabel Kasus Uji Login	104
Tabel 6.8 Tabel Kasus Uji <i>login</i> alternatif : Jika <i>email</i> dan <i>password</i> tidak di isi. 105	
Tabel 6.9 Tabel Kasus Uji <i>email</i> atau <i>password</i> salah.....	105
Tabel 6.10 Tabel Kasus Uji Lihat Profil	105
Tabel 6.11 Tabel Kasus Uji Ubah Data profil	106
Tabel 6.12 Tabel Kasus Uji <i>form</i> nama, <i>username</i> , <i>email</i> , <i>nip</i> kosong	106
Tabel 6.13 Tabel Kasus Uji Lihat Profil	107
Tabel 6.14 Tabel Kasus Uji Lihat Detail Notifikasi	107
Tabel 6.15 Tabel Kasus Uji Buat Draft Program.....	108

Tabel 6.16 Tabel Kasus Uji Buat <i>Draft</i> Programa : mengisi lebih dari 1000 kata	108
Tabel 6.17 Tabel Kasus Uji Buat <i>Draft</i> Programa : Mengosongkan salah satu <i>form</i>	109
Tabel 6.18 Tabel Kasus Uji Masukan Jadwal Kegiatan	109
Tabel 6.19 Tabel Kasus Uji Masukan Jadwal Kegiatan dengan mengosongkan salah satu form	110
Tabel 6.20 Tabel Kasus Uji Lihat <i>Draft</i> Programa.....	110
Tabel 6.21 Tabel Kasus Uji Lihat Detail <i>Draft</i> Programa	111
Tabel 6.22 Tabel Kasus Uji Verifikasi <i>Draft</i> Programa.....	111
Tabel 6.23 Tabel Kasus Uji satu <i>form</i> penjelasan kelayakan di kosongkan	112
Tabel 6.24 Tabel Kasus Uji Lihat Jadwal dan Laporan Penyuluhan.....	112
Tabel 6.25 Tabel Kasus Uji Membuat Laporan Penyuluhan	112
Tabel 6.26 Tabel Kasus Uji Status Belum Dikirim.....	113
Tabel 6.27 Tabel Kasus Uji <i>Form</i> Kosong.....	114
Tabel 6.28 Tabel Kasus Uji Lihat Detail Laporan penyuluhan	114
Tabel 6.29 Kasus Uji Verifikasi Laporan Penyuluhan	115
Tabel 6.30 Tabel Kasus Uji <i>Form</i> Kosong.....	115
Tabel 6.31 Tabel Kasus Uji Ajukan Surat Tugas.....	116
Tabel 6.32 Tabel Kasus Uji Lihat Daftar Surat Tugas.....	116
Tabel 6.33 Tabel Kasus Uji Verifikasi Surat Tugas	117
Tabel 6.34 Tabel Kasus Uji Menerbitkan Surat Tugas	117
Tabel 6.35 Tabel Kasus Uji Ajukan Proposal Dana	118
Tabel 6.36 Tabel Kasus Uji Lihat Daftar Proposal Dana	118
Tabel 6.37 Tabel Kasus Uji Verifikas Proposal Dana	119
Tabel 6.38 Tabel Kasus Uji Lihat Daftar Pengiriman Dana	119
Tabel 6.39 Tabel Kasus Uji Lihat Detail Pengiriman Dana.....	119
Tabel 6.40 Tabel Kasus Uji Konfirmasi Penerimaan Dana	120
Tabel 6.41 Tabel Kasus Uji Lihat Daftar pengguna.....	120
Tabel 6.42 Tabel Kasus Uji Tambah Pengguna.....	121
Tabel 6.43 Kasus Uji Tambah Pengguna <i>alternative</i> : <i>Form</i> tidak lengkap	121
Tabel 6.44 Tabel Kasus Uji Lihat Data Pengguna	122
Tabel 6.45 Tabel Kasus Uji <i>Edit</i> pengguna	122



Tabel 6.46 Tabel Kasus Uji Edit Pengguna *Alternative* : Form Tidak Lengkap 123

Tabel 6.47 Tabel Kasus Uji Hapus Pengguna..... 123

Tabel 6.48 Tabel Kasus Uji *Logout* 124

Tabel 6.49 *Compatibility Testing*..... 124



DAFTAR GAMBAR

Gambar 2.1 Model <i>Waterfall</i>	8
Gambar 2.2 Ilustrasi MVC <i>Laravel</i>	13
Gambar 3.1 Diagram Blok Metodologi	16
Gambar 4.1 Proses Bisnis Pembuatan Draft Programa dan Penerbitan Surat Tugas	20
Gambar 4.2 Proses Bisnis Pembuatan Laporan Kegiatan Penyuluhan	21
Gambar 4.3 Proses Bisnis Pengiriman Dana	22
Gambar 4.1 <i>Use Case Diagram</i>	40
Gambar 5.1 <i>Sequence Diagram</i> Tambah Draft Programa	67
Gambar 5.2 <i>Sequence Diagram</i> Verifikasi Draft Programa.....	68
Gambar 5.3 <i>Sequence Diagram</i> Buat Laporan Penyuluhan.....	70
Gambar 5.4 <i>Class Diagram</i>	71
Gambar 5.5 <i>Class Diagram Controller</i>	72
Gambar 5.6 <i>Class Diagram Model</i>	74
Gambar 5.7 Perancangan <i>Entity Relationship Diagram</i>	75
Gambar 5.8 Perancangan Antarmuka Tambah <i>Draft</i> programa.....	79
Gambar 5.9 Perancangan Antarmuka Verifikasi <i>Draft</i> Programa.....	80
Gambar 5.10 Perancangan Antarmuka Tambah Laporan Penyuluhan	81
Gambar 5.11 Implementasi Basis Data	83
Gambar 5.12 Halaman Tambah Draft Programa	91
Gambar 5.13 Halaman Verifikasi <i>Draft</i> Progama.....	92
Gambar 5.14 Halaman Tambah Laporan penyuluhan	93
Gambar 6.1 <i>Flow Graph</i> method tambah data programa	96
Gambar 6.2 <i>Flow Graph</i> method verifikasi <i>draft</i> prgorama.....	99
Gambar 6.3 <i>Flow Graph</i> Tambah laporan Penyuluhan.....	102
Gambar 6.4 Hasil <i>Compatibility Testing</i>	125

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Di Provinsi Sumatera Barat khususnya Kabupaten Padang Pariaman mempunyai potensi pertanian terbesar. Dalam hal ini yang memiliki peran penting dalam pembangunan ekonomi adalah sektor pertanian, kehutanan, perikanan dan Pendapatan Daerah Regional Bruto (PDRB) yaitu sebanyak 21.20%. Subsektor yang berperan penting dalam sektor pertanian, perikanan dan kehutanan yaitu subsektor tanaman pangan berkontribusi sebanyak 34,01%, subsektor perkebunan sebanyak 16,61%, dan subsektor hortikultura sebanyak 15,35% (Siyoksih, Nur & Tou 2018).

Bagian pelaksana urusan yang terkait dengan pemerintahan di bidang pertanian ialah Dinas Pertanian dan ketahanan Pangan Kabupaten Padang Pariaman. Kedinasan ini memiliki tanggung jawab dalam meningkatkan kualitas pertanian. Bentuk usaha yang digunakan untuk meningkatkan kualitas pertanian adalah Dinas Pertanian dan ketahanan Pangan Kabupaten Padang Pariaman diadakannya sebuah kegiatan penyuluhan pertanian dengan memberi pengarahan kepada petani agar mau merubah cara masyarakat dalam bersikap, berfikir dan berperilaku terhadap perkembangan teknologi. Kegiatan penyuluhan ini melibatkan penyuluh sebagai pemberi arahan tentang pertanian kepada kelompok tani, kepala BPP (*Badan Penyuluhan Pertanian*) sebagai koordinator penyuluh dalam sebuah kecamatan, pegawai dinas sebagai pengelola data pertanian dan kepala dinas sebagai kepala dari sebuah dinas pertanian.

Berdasarkan wawancara dengan Kepala Seksi Penyuluhan “Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman” terdapat beberapa permasalahan yaitu pertama proses pembuatan draft program, pengesahan draft program, dan penerbitan surat tugas kegiatan penyuluhan yang masih dilakukan secara manual. Sehingga membutuhkan waktu yang lama kurang lebih 5 hingga 7 hari dan memakan biaya untuk perjalanan kurang lebih 20 km dari wilayah kerja penyuluh menuju kantor “Dinas Pertanian Dan Ketahanan Pangan Kabupaten Padang Pariaman”. Dalam pemeriksaan draft program akan ada kemungkinan draft program ditolak, sehingga penyuluh harus mengambil kembali draft program untuk direvisi, tentu nya akan memakan waktu bagi penyuluh. Permasalahan selanjutnya Laporan hasil kegiatan penyuluhan pertanian tidak segera diberikan setelah penyuluhan melainkan diberikan satu kali setiap bulan, hal ini membuat laporan tidak relevan dengan waktu pelaksanaan dan membuat laporan terakumulasi dalam waktu yang lama. Sehingga kurang transparansinya penyuluh lapangan dalam membuat laporan. Dan juga proses pengumpulan yang dilakukan secara manual akan memakan waktu lama karena harus melalui beberapa tahap yaitu pemeriksaan oleh kepala BPP dan pemeriksaan oleh pegawai dinas.

Berdasarkan masalah yang telah dipaparkan sebelumnya, “Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman” membutuhkan sebuah

sistem aplikasi berbasis web yang menyediakan sarana untuk pembuatan, pengumpulan, dan pemeriksaan draft program secara *online*. Sarana untuk pengajuan, pengesahan, dan penerbitan surat tugas secara *online* dan pembuatan, pengumpulan, dan pemeriksaan laporan kegiatan kegiatan penyuluhan secara *online*. Sehingga disarankan agar membangun sistem informasi untuk penelitian skripsi dengan judul “SISTEM INFORMASI PENYULUHAN PERTANIAN DI DINAS PERTANIAN DAN KETAHANAN PANGAN KABUPATEN PADANG PARIAMAN”.

Di harapkan dengan adanya sistem ini dapat mempermudah “Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman” dalam mengelola draft program, laporan kegiatan penyuluhan, pengelolaan pengiriman dana di Kabupaten Padang Pariaman yang dibangun menggunakan Bahasa PHP dengan framework Laravel dan menyimpan data menggunakan database MYSQL. Pemilihan sebagai framework *Laravel* dan *database MYSQL*. Pemilihan *Laravel* sebagai framework yang digunakan dalam pengembangan aplikasi ini disebabkan karena *Laravel* memakai konsep MVC, penjelasan dari MVC sendiri ialah *model*, *view* serta *control*. *Laravel* ialah framework *PHP* yang menjelaskan pada fleksibilitas serta kesederhanaan. Sama halnya dengan framework *PHP* lain, *Laravel* dikembangkan memakai konsep MVC.

1.2 Rumusan Masalah

Berpedoman dari permasalahan yang telah di jelaskan pada latar belakang maka rumusan masalah untuk penelitian skripsi adalah sebagai berikut:

1. Bagaimana hasil dari analisis kebutuhan Sistem Informasi Penyuluhan Pertanian di Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman?
2. Bagaimanakah hasil dari perancangan Sistem Informasi Penyuluhan Pertanian di Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman dari analisi kebutuhan yang di lakukan?
3. Bagaimanakah hasil dari implementasi Sistem Informasi Penyuluhan Pertanian di Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman dari perancangan yang di buat?
4. Bagaimana hasil dari pengujian Sistem Informasi Penyuluhan Pertanian di Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman ?

1.3 Tujuan

Berpedoman dari pertanyaan rumusan masalah maka penelitian skripsi ini memiliki tujuan sebagai berikut:

1. Menghasilkan artefak spesifikasi kebutuhan sistem informasi penyuluhan pertanian.
2. Menghasilkan artefak perancangan sistem informasi penyuluhan pertanian.
3. Menghasilkan implementasi produk sistem informasi penyuluhan pertanian.

4. Menghasilkan sistem informasi yang sudah teruji informasi penyuluhan pertanian.

1.4 Manfaat

Berpedoman dari tujuan yang telah dijelaskan maka penelitian skripsi ini memiliki manfaat sebagai berikut:

1. Menambah pengetahuan dalam pembuatan sistem informasi yang berkaitan dengan pertanian.
2. Penelitian ini bermanfaat bagi penyuluh lapangan membuat laporan penyuluhan secara online dan membuat draft program secara online serta mengkonfirmasi penerimaan dana untuk transparansi dana
3. Bermanfaat bagi kepala BPP dalam mengelola data laporan dan draft program kegiatan penyuluhan
4. Penelitian ini bermanfaat bagi pegawai Dinas Pertanian dan Ketahanan Pangan dalam mengelola data kegiatan penyuluhan pertanian Kabupaten Padang Pariaman.

1.5 Batasan Masalah

Beberapa Batasan masalah pada penelitian ini ialah sebagai berikut:

1. Pengambilan data berasal dari Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman.
2. Sistem ini hanya dapat dikembangkan di Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman.
3. Struktur dari kebutuhan fungsional dan perancangan implementasi hanya di sesuaikan dengan kebutuhan pihak Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman.
4. Penggunaan Bahasa untuk sistem penelitian ini adalah Bahasa Indonesia.

1.6 Sistematika Pembahasan

Dalam Sistematika Pembahasan akan dijelaskan terkait apa saja isi dari beberapa bab, yaitu:

BAB I Pendahuluan

Pada Bab ini akan membahas hal-hal yang menjadi dasar pembuatan proposal skripsi dan ruang lingkup dari Dinas Pertanian dan Ketahanan Pangan sebagai tumpuan bab berikutnya. Pada tahapan ini akan menjadi acuan pertama dalam menentukan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika pembahasan.

BAB II Landasan Kepustakaan

Pada bab ini akan memberi penjelasan terkait dasar-dasar teori mengenai teknologi yang digunakan dalam penelitian skripsi.

BAB III Metodologi

Pada bab ini akan dijelaskan terkait tahapan metodologi yang dilakukan pada penelitian skripsi ini. Tahapan tersebut tersusun dari studi literatur, analisis dan perancangan, implementasi, pengujian, dan kesimpulan.

BAB IV Analisis dan Perancangan

Pada bab ini dibuat sebagai landasan tahap implementasi dan pengujian. Terdiri dari dua bagian yaitu analisi kebutuhan dan perancangan sistem. Pada analisis kebutuhan akan digambarkan memakai *use case diagram* dan *use case scenario*. Pada perancangan akan dilakukan berdasarkan hasil analisis dari spesifikasi kebutuhan.

BAB V Implementasi

Pada bab ini akan dijelaskan terkait pembahasan implementasi dari perancangan yang telah dilakukan pada tahap sebelumnya. Implementasi memiliki tujuan menghasilkan website yang bisa menjadi suatu solusi dari permasalahan pada latar belakang.

BAB VI Pengujian

Pada bab ini akan menjelaskan terkait metode yang akan digunakan untuk menguji hasil implementasi. Pengujian pada penelitian skripsi ini menggunakan black box, white box serta compability.

BAB VII Penutup

Pada bab ini menjelaskan kesimpulan hasil dari proses analisis kebutuhan, perancangan, implementasi serta pengujian. Hasil dari skripsi ini termasuk saran yang mempunyai tujuan agar bisa meningkatkan pengembangan lebih lanjut bagi penelitian berikutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab 2 akan dijelaskan mengenai aspek-aspek pengetahuan yang berhubungan dengan penelitian ini. Pada tahap awal akan menjelaskan kajian pustaka, selanjutnya akan menjelaskan tentang sistem informasi, penyuluhan pertanian, SDLC, UML, ERD, *Laravel*, dan pengujian perangkat lunak.

2.1 Kajian Pustaka

Kajian pustaka membahas penelitian terdahulu untuk dijadikan referensi penulis dalam membuat penelitian mengenai sistem penyuluhan pertanian pada dinas pertanian dan ketahanan pangan kabupaten Padang Pariaman. Penelitian-penelitian sebelumnya memiliki beberapa kesamaan dengan penelitian tentang Sistem Informasi Penyuluhan Pertanian ini. Pada penelitian sebelumnya terdapat fakta pendukung bahwa pentingnya sebuah perangkat lunak pada pertanian untuk mempermudah dalam pekerjaan.

Penelitian pertama yang dilakukan oleh (Hajar 2017) mengenai kegiatan kelompok tani di kecamatan patalassang kabupaten Gowa. Permasalahan yang ditemui yaitu kurangnya transparansi pembagian dana kepada kelompok tani, masalah selanjutnya yaitu tidak efektif dan efisien dalam penjadwalan kegiatan penyuluhan. Untuk mengatasi masalah tersebut (Hajar 2017) membangun sistem *monitoring* kelompok tani menggunakan teknik pengumpulan data yaitu observasi serta wawancara kepada kelompok tani daerah setempat, melakukan perancangan menggunakan metode SDLC *waterfall*, dan metode pengujian menggunakan *Whitebox* dan *Blackbox*. Setelah melakukan penelitian didapat kesimpulan bahwa sistem yang dibangun dapat membantu dalam transparansi pengiriman dana dan penjadwalan yang lebih efektif dan efisien setelah melewati pengujian.

Penelitian kedua yang dilakukan oleh (Shopyan 2017) pada Dinas Pertanian Tanaman Pangan dan Perkebunan di Kabupaten Baru. Masalah yang menjadi dasar penelitiannya yaitu masalah pemberdayaan kelompok tani yang masih kurang efektif sehingga kurang patuhnya kelompok tani. Petani-petani masih melakukan cara yang tradisional karena menurut para petani cara tradisional memiliki hasil yang banyak walaupun hasilnya kurang berkualitas. Untuk mengatasi masalah ini maka (Shopyan 2017) membangun sebuah sistem informasi yang dapat membantu kelompok tani sehingga lebih mudah dalam mengelola strategi organisasi dan informasi. Teknik pengumpulan data yang digunakan yaitu dengan observasi langsung ke tempat dan melakukan proses wawancara dengan penduduk, selanjutnya untuk proses perancangan menggunakan metode UML dan proses pengujian dengan menggunakan pengujian *Blackbox*. Setelah melakukan penelitian ini didapatkan hasil bahwa sistem ini dapat membantu kelompok tani untuk mengoperasikan bantuan serta keluhan dan dapat memudahkan petani dalam mendapatkan penyuluhan seputar pertanian dalam bentuk video untuk media pembelajaran.

Penelitian-penelitian diatas dapat menjadi pedoman untuk melakukan penelitian ini. Tabel 2.1 menjelaskan perbedaan serta persamaan dari tiap penelitian yang telah dipaparkan sebelumnya.

Tabel 2.1 Kajian Pustaka

No	Nama Penulis, Tahun terbit serta judul penelitian	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	St Hajar, 2017, Rancang Bangun Sistem <i>Monitoring</i> Kegiatan Kelompok Tani berbasis <i>web</i> di Kecamatan Pattallassang Kabupaten Gowa, Sulawesi Selatan.	Sistem informasi kegiatan kelompok tani. Untuk memudahkan mengelola data kelompok tani, kegiatan, dan bantuan. Menggunakan metode wawancara dalam mengumpulkan data, metode SDLC <i>waterfall</i> , dan metode <i>whitebox</i> dan <i>blackbox</i> dalam pengujian	Studi kasus nya di kabupaten Gowa. Sistem hanya mencakup transparansi dana dan penjadwalan kegiatan kelompok tani.	Studi kasus di Kabupaten Padang pariaman. Sistem mencakup pengelolaan draft programa rancangan kegiatan, mengelola laporan kegiatan, dan transparansi dana.
2	Nurzakina Sophyan, 2017, Rancang Bangun Sistem Informasi Pemberdayaan Kelompok Tani pada Dinas Pertanian Tanaman Pangan dan Perkebunan Kabupaten Barru Berbasis <i>web</i> .	Sistem infomasi untuk meningkatkan kualitas pertanian. Untuk memudahkan pengelolaan kegiatan kelompok tani. Menggunakan metode wawancara.	Studi kasus di Kabupaten barru. Sistem berguna untuk memudahkan kelompok tani dalam mendapatkan pengetahuan secara langsung dari sistem dalam bentuk video dan ada nya fitur pengoperasian	Studi kasus di Kabupaten Padang pariaman. Sistem mencakup pengelolaan draft programa rancangan kegiatan, mengelola laporan kegiatan, dan transparansi dana. Menggunakan

No	Nama Penulis, Tahun terbit serta judul penelitian	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
			bantuan dan keluhan. Hanya menggunakan metode <i>blackbox</i> dalam pengujian	metode <i>whitebox</i> dan <i>blackbox</i> dalam pengujian.

2.2 Sistem Informasi

Sistem informasi adalah sistem yang mempertemukan kebutuhan transaksi harian yang bersifat manajerial untuk dapat mengambil keputusan dengan informasi yang diperlukan (Iswandy 2015). Menurut (Aisyah & Falgenti 2017) sistem informasi merupakan pengolahan data-data sehingga menjadi sebuah informasi berguna dalam pengambilan keputusan di sebuah manajemen. Kesimpulan yang dapat diambil dari pengertian sistem informasi adalah sebuah sistem yang dapat mengolah data berupa *inputan* menjadi sebuah informasi. Sistem informasi sangat berguna dalam pertumbuhan sebuah organisasi atau perusahaan karena dapat menjadi acuan mengambil keputusan dalam menentukan target.

2.3 Penyuluhan Pertanian

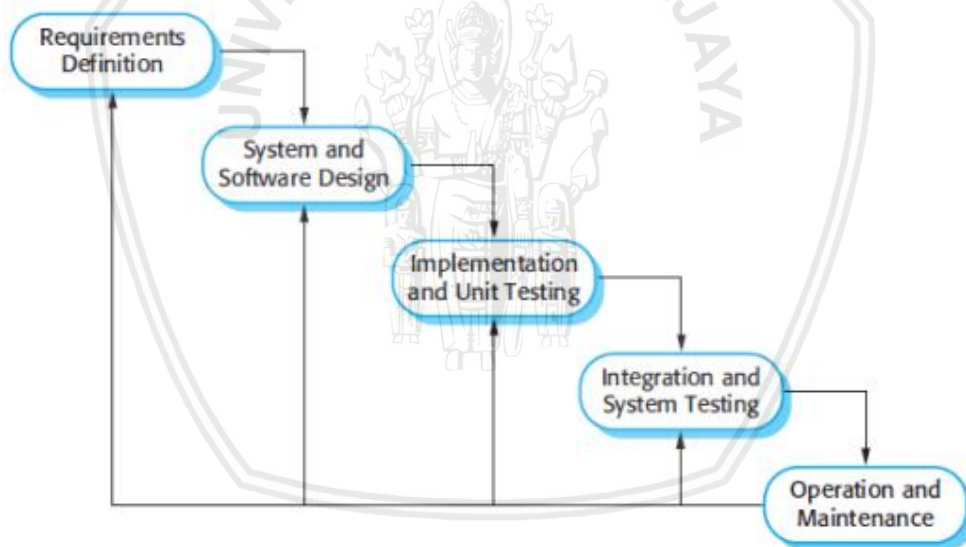
Penyuluhan pertanian merupakan sebuah program dengan tujuan untuk pembangunan pertanian di sebuah wilayah yang telah disusun dan disesuaikan dengan kebutuhan masyarakat dengan mengikutsertakan masyarakat dalam kegiatan tersebut agar masyarakat lebih memahami tentang pertanian (Adjid 2001). Menurut (Hasibuan 2016) penyuluhan pertanian adalah sebuah Pendidikan non formal atau luar sekolah yang bertujuan agar para petani dapat berusaha tani lebih baik, dapat menjaga kelestarian lingkungan lebih baik, dapat bertani lebih baik, dapat hidup lebih sejahtera, dan dapat bermasyarakat lebih baik. Dapat disimpulkan bahwa penyuluhan pertanian yaitu sebuah kegiatan yang telah tersusun sesuai kebutuhan masyarakat yang bertujuan untuk meningkatkan kualitas pertanian masyarakat. Dengan adanya penyuluhan pertanian maka petani dapat mendapatkan ilmu dalam pertanian yang lebih modern, sehingga dapat meningkatkan pertumbuhan pertanian dengan memadukan pengalaman yang ada dengan ilmu yang diberikan.

2.4 SDLC

Metode SDLC atau *System Development Life Cycle* merupakan proses dalam membangun atau mengembangkan perangkat lunak dengan menggunakan metodologi-metologi yang digunakan dalam mengembangkan sistem perangkat lunak sebelumnya (Rosa & M. Salahuddin 2014). Menurut (Hermawan, Hidayat & Utomo 2016) SDLC adalah pengembangan yang berguna untuk menjadi sebuah mekanisme dalam mengidentifikasi perangkat lunak. Dapat disimpulkan bahwa SDLC yaitu sebuah metode yang berbentuk langkah-langkah proses dalam mengembangkan sistem. Ada beberapa model SDLC yang sering digunakan, yaitu SDLC *waterfall*, *prototyping*, *incremental*, *build & fix*, *rapid*, *spiral*, *synchronize & stabilize* (Isnardi 2016). Pada penelitian ini menggunakan metode *waterfall*.

2.4.1 SDLC Waterfall

Dalam pembangunan sistem informasi penyuluhan ini terdapat siklus hidup yang berisi rangkaian prosedur pengembangan perangkat lunak dengan menggunakan *Waterfall*. Menurut (Pressman 2010) SDLC *waterfall* merupakan alur hidup klasik atau juga biasa disebut dengan model sekuensial linier dalam membangun sebuah perangkat lunak.



Gambar 2.1 Model Waterfall

Sumber : (Sommerville 2011)

Berdasarkan Gambar 2.1 model *Waterfall* memiliki beberapa tahapan dalam pengembangan perangkat lunak, yaitu :

1. Definisi dan analisis kebutuhan

Pada tahap ini untuk mendapatkan kebutuhan-kebutuhan dengan cara melakukan wawancara dengan kasi penyuluhan dan kepala BPP, hasil yang didapat nanti akan berupa analisis permasalahan, pemodelan bisnis, dan mengumpulkan data yang dibutuhkan

2. Desain

Tahap ini merupakan tahapan perancangan dari analisis kebutuhan yang sudah didapat agar dapat menjadikan dasar dalam implementasi program pada tahap selanjutnya. Pada penelitian skripsi ini perancangan terdiri dari *sequence diagram*, *basis data*, *class diagram*, komponen, dan antarmuka.

3. Implementasi dan pengujian unit

Tahap implementasi merupakan proses mengimplementasikan hasil perancangan sebelumnya. Hasil implementasi yaitu berupa basis data, kode program, dan antarmuka. Dilakukan pemeriksaan untuk memastikan bahwa setiap unit telah memenuhi spesifikasi sistem.

4. Integrasi dan sistem

Pada tahap ini seluruh di satukan kemudian diuji sebagai sebuah sistem yang lengkap untuk memastikan bahwa sistem sudah memenuhi syarat. Setelah melakukan pengujian sistem dapat diberikan kepada klien.

5. Operasi dan pemeliharaan

Pemeliharaan perangkat lunak dilakukan bertujuan agar sistem dapat berjalan dan berkembang sesuai kebutuhan. Tahap ini bertujuan untuk meningkatkan layanan dengan mengoreksi kesalahan yang tidak ditemukan sebelumnya.

Pada penelitian ini tidak dilakukan sampai tahap operasi dan pemeliharaan tetapi tahapan yang dilakukan hanya dari tahap definisi dan analisis kebutuhan hingga implementasi dan pengujian unit.


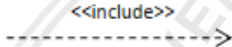
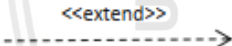



2.5 UML

Undifine Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan dalam dunia industry dalam mendefinisikan kebutuhan, merancang kebutuhan, dan menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & M. Salahuddin 2014). UML merupakan sebuah bagian dari notasi grafis yang menggunakan *meta-model* tunggal, digunakan pada saat mendesain sebuah perangkat lunak. Khususnya sebuah sistem yang dibangun dengan pemrograman berorientasi objek (Tofik & Hartawan 2017). Dapat disimpulkan UML merupakan sebuah alat yang dapat digunakan untuk membantu membangun sebuah sistem yang berorientasi pada objek. Alat-alat yang dapat digunakan untuk membantu membuat UML terdiri dari *usecase diagram*, *class diagram*, *sequence diagram*.

2.5.1 Use Case Diagram

Use case diagram mendeskripsikan interaksi aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case diagram* digunakan untuk melihat fungsi apa saja yang ada di dalam sistem dan siapa saja yang melakukan fungsi tersebut (Rosa & M. Salahuddin 2014). *Use case diagram* memvisualiasi kegiatan yang dapat dilakukan oleh aktor terhadap sistem. Dengan adanya *usecase diagram* akan mempermudah dalam memahami bagaimana seorang aktor dan sistem berinteraksi. Tabel 2.2 menjelaskan beberapa komponen yang terdapat pada *use case diagram*.

Tabel 2.2 Notasi *Use Case Diagram*

NO	SIMBOL	PENJELASAN	KETERANGAN
1		<i>Actor</i>	Aktor ialah pengguna yang mampu menjalankan <i>fungsi</i> di dalam sistem.
2		<i>Include</i>	Menspesifikasikan <i>base use case</i> yang memperluas fungsi nya pada <i>supplier use case</i> setiap saat.
3		<i>Extend</i>	Menspesifikasikan <i>base use case</i> yang memperluas fungsi nya pada <i>supplier use case</i> jika dibutuhkan.
4		<i>Association</i>	Yang menghubungkan aktor dengan <i>fungsi</i> atau <i>usecase</i>
5		<i>Use Case</i>	Menggambarkan <i>fungsi</i> dari sistem yang berhubungan dengan <i>actor</i> , <i>use case</i> harus dibuat dengan kata kerja
6		<i>Generalization</i>	Menggambarkan hubungan generalisasi antara dua objek dan spesifikasi nya





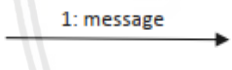



Sumber : (Rosa & M. Salahuddin 2014)

2.5.2 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan serta yang diterima oleh objek (Rosa & M. Salahuddin 2014). Diagram sekuensial atau *sequence diagram* menjelaskan interaksi antar objek dalam urutan waktu (Tofik & Hartawan 2017). *Sequence diagram* berisi tahapan-tahapan aktifitas yang dimulai oleh aktor. Urutan interaksi ini berasal dari *usecase scenario* yang sudah dibuat

sebelumnya, setiap *usecase* akan dibuatkan dalam bentuk *usecase diagram* satu persatu. *Use case diagram* memiliki beberapa komponen yang akan dijelaskan pada Tabel 2.3

Tabel 2.3 Komponen Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	<i>Aktor</i> merupakan seorang pengguna yang terlibat dengan <i>fungsiionalitas</i> atau pemicu dari <i>fungsiionalitas</i> ini terjadi.
2		<i>Entity Class</i>	Merupakan sebuah kelas yang menghubungkan dengan basis data.
3		<i>Boundary Class</i>	Merupakan sebuah kelas yang menampilkan antarmuka (<i>interface</i>) untuk pengguna dapat berinteraksi dengan sebuah sistem.
4		<i>Control Class</i>	Merupakan kelas untuk jembatan penghubung antara <i>entity class</i> dan <i>boundary class</i> .
5		<i>Message tipe send</i>	Merupakan simbol untuk mengirim pesan dari satu kelas ke kelas yang lainnya
6		<i>Message tipe return</i>	Merupakan simbol untuk mengirim pesan dari satu kelas ke kelas yang lainnya
7		<i>Activation</i>	Menggambarkan durasi dalam sebuah eksekusi perintah terhadap objek.
8		<i>Lifeline</i>	Merupakan garis yang terhubung dengan sebuah <i>object</i> .

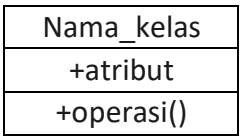
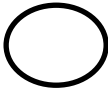



Sumber : (Rosa & M. Salahuddin 2014)

2.5.3 Class Diagram

Class Diagram merupakan sebuah penggambaran dari hubungan antar satu kelas dengan kelas yang lainnya dan dilengkapi dengan detail tiap kelas (Hendini 2016). *Class Diagram* terdiri dari *class*, atribut, operasi, dan *visibility* dari operasi dan atribut. *Class diagram* memiliki relasi tiga jenis, yaitu relasi asosiasi, relasi agregasi, dan relasi relasi generalisasi. Relasi asosiasi merupakan antar kelas saling menggunakan atribut masing-masing. Relasi generalisasi merupakan sebuah kelas

yang menjelaskan sifat khas maupun umum dari kelas lain. *Class diagram* memiliki beberapa komponen yang akan di jelaskan pada Tabel 2.4

Tabel 2.4 Komponen Class Diagram


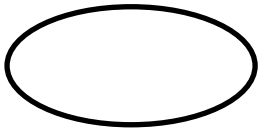
NO	SIMBOL	PENJELASAN	KETERANGAN
1		Kelas	Kelas pada struktur sistem.
2		Antarmuka	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek.
3		Asosiasi	Relasi antarkelas dengan makna umum, satu kelas saling menggunakan dengan kelas lainnya.
4		Generalisasi	Relasi antar kelas dengan makna umum-khusus.
5		Agregasi	Relasi antar kelas dengan makna semua-bagian.

Sumber : (Rosa & M. Salahuddin 2014)

2.6 ERD

Entity Realtionship Diagram (ERD) adalah metode yang digunakan untuk menggambarkan hubungan antar objek atau data yang memiliki notasi-notasi dalam hubungannya (Doro & Betshani 2009). ERD memiliki beberapa komponen dan notasi-notasi relasi yang menghubungkan beberapa objek. Untuk keterangan yang lebih jelas dapat dilihat pada Tabel 2.5

Tabel 2.5 Komponen ERD

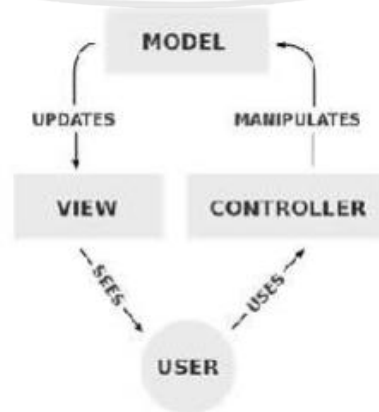
Notasi	Komponen	Keterangan
	<i>Entity</i>	Menggambarkan satu objek yang berbeda dengan objek lain.
	<i>Attribute</i>	Propoerti yang dimiliki sebuah objek dan menjelaskan karakteristik dari objek tersebut.

Notasi	Komponen	Keterangan
	<i>Relation</i>	Menjelaskan hubungan antar objek.
	<i>Relation 1:1</i>	Relasi ini menjelaskan jika satu entitas hanya berhubungan dengan satu entitas.
	<i>Relation 1:N</i>	Relasi yang menggambarkan satu objek bisa berhubungan dengan satu objek lainnya atau lebih.
	<i>Relation N:N</i>	Relasi yang menggambarkan bahwa tiap entitas dapat memiliki hubungan dengan satu entitas atau lebih.

Sumber: (Rosa & M. Salahuddin 2014)

2.7 Laravel

Laravel merupakan sebuah *framework* yang digunakan dalam membuat website dan berbasis PHP yang *open source*, *Laravel* dibangun dengan menggunakan pola *model view controller* (MVC). *Laravel* berbeda dengan *framework* berbasis PHP lainnya karena *Laravel* memiliki sebuah *routes* yang berguna untuk menghubungkan *request* pengguna dengan *controller* sehingga *controller* tidak langsung merespon *request* dari pengguna. Pada penelitian ini menggunakan *Laravel* versi 5. Gambar 2.2 merupakan ilustrasi dari *Laravel*.



Gambar 2.2 Ilustrasi MVC *Laravel*

Sumber : (Mualim & Putra 2017)

Pada Gambar 2.2 dapat dilihat ketika pengguna melakukan *request* maka akan di *handle* oleh *routes*, kemudian *routes* akan mengarahkan ke *controller*, *controller* diibaratkan sebuah jembatan antara *view* dan *model*, maka *controller* akan mengambil data ke *model*, dan mengirimkan data tersebut ke *view*, dan *view* menampilkan data yang dikirimkan *controller* untuk ditampilkan kepada pengguna.

2.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan sebuah elemen penting yang digunakan dalam membangun sebuah perangkat lunak, karena dengan pengujian perangkat lunak dapat menjamin kualitas sebuah perangkat lunak. Tanpa adanya pengujian ini maka akan besar kemungkinan ditemukannya kesalahan atau *bug* pada sistem yang dibangun. Menurut (Agus 2016) Pengujian perangkat lunak adalah suatu proses dalam pembangunan sistem yang digunakan untuk menentukan mutu, kelengkapan dan ketepatan suatu perangkat lunak. Pengujian perangkat lunak pada penelitian ini menggunakan 3 pengujian yaitu pengujian *Black Box* yang akan dijelaskan pada subbab 2.9.1, *White Box* yang akan dijelaskan pada subbab 2.9.2, dan *Compatibility* yang akan dijelaskan pada subbab 2.9.3

2.8.1 Black-Box Testing

Menurut (Astuti 2018) pengujian *black box* merupakan pengujian yang hanya dilakukan pada fungsionalitas dan hanya mengamati *output* dari sistem ketika sistem dimasukan data uji. Pengujian *black box* hanya menguji dari luar atau *interface* dan *fungsionalitas* saja tanpa mengetahui apa yang ada pada proses detilnya. Pengujian *black box* ada beberapa teknik yaitu *Boundary Value Analysis*, *all-pairs testing*, *state transition table*, *decision table*, dan *equivalence partitioning*.

Pada penelitian ini menggunakan teknik *Boundary Value Analysis* (BVA). BVA yaitu salah satu teknik pengujian *black box* dengan menguji sebuah sistem pada batas bawah dan batas atas saat memasukan data kedalam sistem (Mustaqbal, Firdaus. & Rahmadi 2015). Teknik ini dengan melakukan pengujian pada masing-masing fungsionalitas yaitu memasukan data-data pada batas atas dan batas bawah masukan data, dan *output* sistem akan menentukan hasil uji sistem ini.




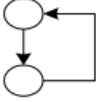
2.8.2 White-Box Testing

White Box Testing adalah sebuah pengujian pada sistem yang bertujuan untuk menemukan kesalahan pada kode program yang dibuat, jika *output* yang dihasilkan tidak sesuai maka dapat melakukan pemeriksaan terhadap kode-kode dan memperbaikinya sehingga menghasilkan *output* yang sesuai (Mustaqbal, Firdaus. & Rahmadi 2015).

Pada penelitian ini Teknik *white box* yang dipakai ialah *basis path testing*. *Basis path testing* ialah sebuah metode yang digunakan untuk mengukur kompleksitas sebuah logika. Pada penelitian ini terdapat beberapa tahap pengujian *white box*, yaitu yang pertama pembuatan *pseudocode* dari program, selanjutnya

menentukan *flowgraph* yang menggambarkan alur program, selanjutnya dilakukan pengujian *cyclo complexity* untuk mendapatkan jumlah jalur indepent dan terakhir pembuatan uji kasus pada tiap jalur yang sudah ditentukan. *Flowgraph* memiliki beberapa komponen yang akan digambarkan pada Tabel 2.6

Tabel 2.6 Komponen *Flowgraph*

Simbol	Penjelasan
	Node : simbol untuk proses
	Edge : penghubung antar node
	Predicate : digunakan jika ada pemilihan kondisi
	Region : wilayah yang dibatasi oleh node dan edege

Sumber : (Pare 2013)

Rumus untuk menghitung *cyclo complexity* ada tiga rumus yaitu :

1. $VG = \text{jumlah area atau region}$
2. $VG = \text{Edge} - \text{Node} + 2$
3. $VG = \text{Predicate node} + 1$

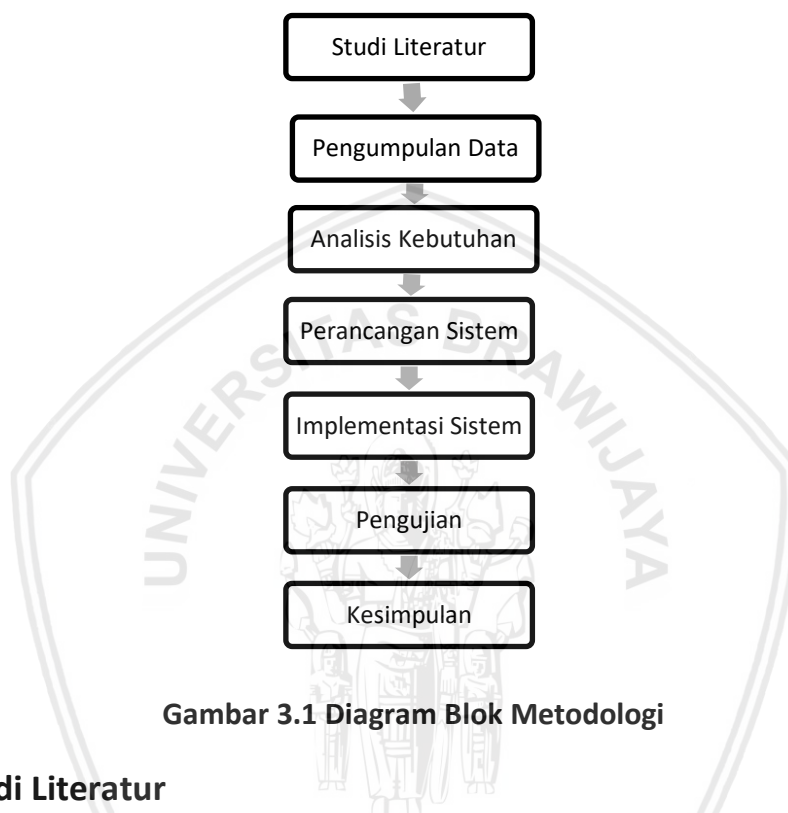
2.8.3 *Compability Testing*

Compatibility testing ialah pengujian pada perangkat lunak yang dipakai untuk memeriksa apakah suatu perangkat lunak dapat membagi informasi dan dapat berinteraksi secara benar dengan perangkat lunak lainnya. Pengujian ini sangat cocok untuk sebuah web *pages*, karena web *pages* mengandung data yang tidak berasal dari komputer itu sendiri melainkan dari perangkat lain (Tjandra & Pickerling 2015).

Pada penelitian ini menggunakan aplikasi *sortsite* untuk melakukan *compatibility testing*. Menurut (Saputra, Pradana & Priyambadha 2017) *sortsite* merupakan sebuah *software* yang dapat melakukan pemeriksaan kualitas sebuah web, setiap halaman web akan melakukan pemeriksaan pada 450 tempat pemeriksaan.

BAB 3 METODOLOGI

Bab ini berisi tahapan penelitian yaitu tahapan pertama studi literatur, tahap selanjutnya analisis kebutuhan, tahap selanjutnya yaitu perancangan sistem, tahap implementasi, tahap pengujian dan kesimpulan. Untuk alur yang lebih jelas terdapat pada Gambar 3.1



Gambar 3.1 Diagram Blok Metodologi

3.1 Studi Literatur

Studi literatur bertujuan mendalami konsep yang memiliki berhubungan dengan penelitian, dengan cara mencari referensi untuk mendukung penelitian ini. Berbagai sumber digunakan dalam penelitian ini, sumber-sumber yang digunakan berasal dari buku, paper, laporan ilmiah, dan jurnal. Daftar-daftar sumber yang digunakan dapat dilihat pada daftar pustaka. Pada penelitian terdapat beberapa komponen yang digunakan untuk penelitian ini, yaitu :

1. Sistem Informasi
2. Penyuluhan pertanian
3. *Swimlanes diagram*
4. SDLC
 - *SDLC waterfall*
5. ERD
6. UML
 - Usecase Diagram
 - Sequence Diagram
 - Class Diagram

7. Laravel
8. Pengujian Perangkat Lunak
 - *Black-box testing*
 - *White-box testing*
 - *Compatibility testing*

3.2 Pengumpulan Data

Tahap ini akan dilakukan pengumpulan kebutuhan sistem agar lebih paham terhadap masalah dan solusi yang akan dihadapi serta alur sistem seperti apa yang akan dibuat, wawancara akan dilakukan dengan pengumpulan data terlebih dahulu dengan kepala seksi penyuluhan dan penyuluh lapangan. Wawancara bertujuan untuk mendapatkan data meliputi alur proses bisnis kegiatan penyuluhan yang sedang berjalan, siapa saja yang terlibat dalam kegiatan penyuluhan yang sedang berjalan, apa kendala yang ditemui selama kegiatan berjalan. Setelah semua data di dapat maka akan dilakukan tahap selanjutnya yaitu analisis data.

3.3 Analisis Kebutuhan Sistem

Setelah data-data yang dibutuhkan sudah terkumpul melalui wawancara dengan kepala seksi penyuluh dan penyuluh lapangan, maka akan dilakukan analisis terhadap data yang sudah terkumpul. Terdapat beberapa tahap dalam melakukan analisis kebutuhan sistem. Untuk yang pertama ialah tahap dalam mendefinisikan aktor yang terlibat pada sistem ini, tahap selanjutnya ialah membuat *usecase* diagram untuk menentukan keterlibatan aktor dengan sebuah sistem serta tahap berikutnya ialah *usecase scenario* yang digunakan untuk mengetahui kegiatan yang dilakukan *actor* secara rinci.

3.4 Perancangan Sistem

Setelah semua kebutuhan sudah didefinisikan dengan benar, maka akan dilakukan perancangan sistem yang bertujuan untuk tumpuan dalam pembuatan implementasi sistem yang akan dibangun. UML adalah model yang akan digunakan dalam perancangan sistem ini. Perancangan sistem menggunakan *class diagram* digunakan untuk perancangan terhadap kelas, perancangan *sequence diagram* untuk menggambarkan objek, dan perancangan basis data dengan menggunakan ERD dan terakhir perancangan antarmuka sistem.

3.5 Implementasi Sistem

Setelah melakukan perancangan maka akan dilakukan implementasi terhadap hasil perancangan sistem. Implementasi sistem akan dibuat dalam bentuk website, terdapat tiga tahap implementasi, yaitu tahap implementasi *frontend* atau antarmuka, untuk implementasi *frontend* menggunakan Bahasa

pemograman web yaitu HTML5, CSS, dan javascript. Tahap selanjutnya akan dilakukan implementasi basis data dengan menggunakan Mysql sebagai *server database*. Dan terakhir implementasi *backend* sistem memakai Bahasa PHP dan memakai *framework* PHP yaitu *Laravel*.

3.6 Pengujian

Tahap pengujian bertujuan mencari kecacatan atau *bug* pada suatu sistem dan juga untuk menentukan kualitas sistem berjalan baik atau tidak. Pengujian dilakukan dengan tiga tahap yaitu tahap pertama dilakukan pengujian memakai metode *blackbox*, tahap selanjutnya pengujian dengan menggunakan *whitebox* dan terakhir pengujian *compatibility* untuk pengujian nonfungsional. Pengujian ini nanti berguna untuk perbaikan sistem jika terdapat kesalahan pada sistem.

3.7 Kesimpulan dan Saran

Tahap ini akan dilakukan penarikan kesimpulan dan saran setelah semua tahap dilakukan selesai. Kesimpulan didapat dari pertanyaan-pertanyaan dari rumusan masalah, sedangkan saran memiliki tujuan untuk memutuskan hal yang dilakukan terhadap site ini pada tahap selanjutnya untuk tahap pengembangan sistem ini.



BAB 4 ANALISIS KEBUTUHAN

Tahap analisis kebutuhan akan menjabarkan segala kebutuhan pengembangan sistem informasi penyuluhan pertanian. Langkah-langkah dalam analisis kebutuhan ini, antara lain: identifikasi aktor, analisis kebutuhan fungsional, pembuatan *use case* diagram, pembuatan *use case* scenario, dan analisis kebutuhan non fungsional.

4.1 Gambaran Umum Sistem

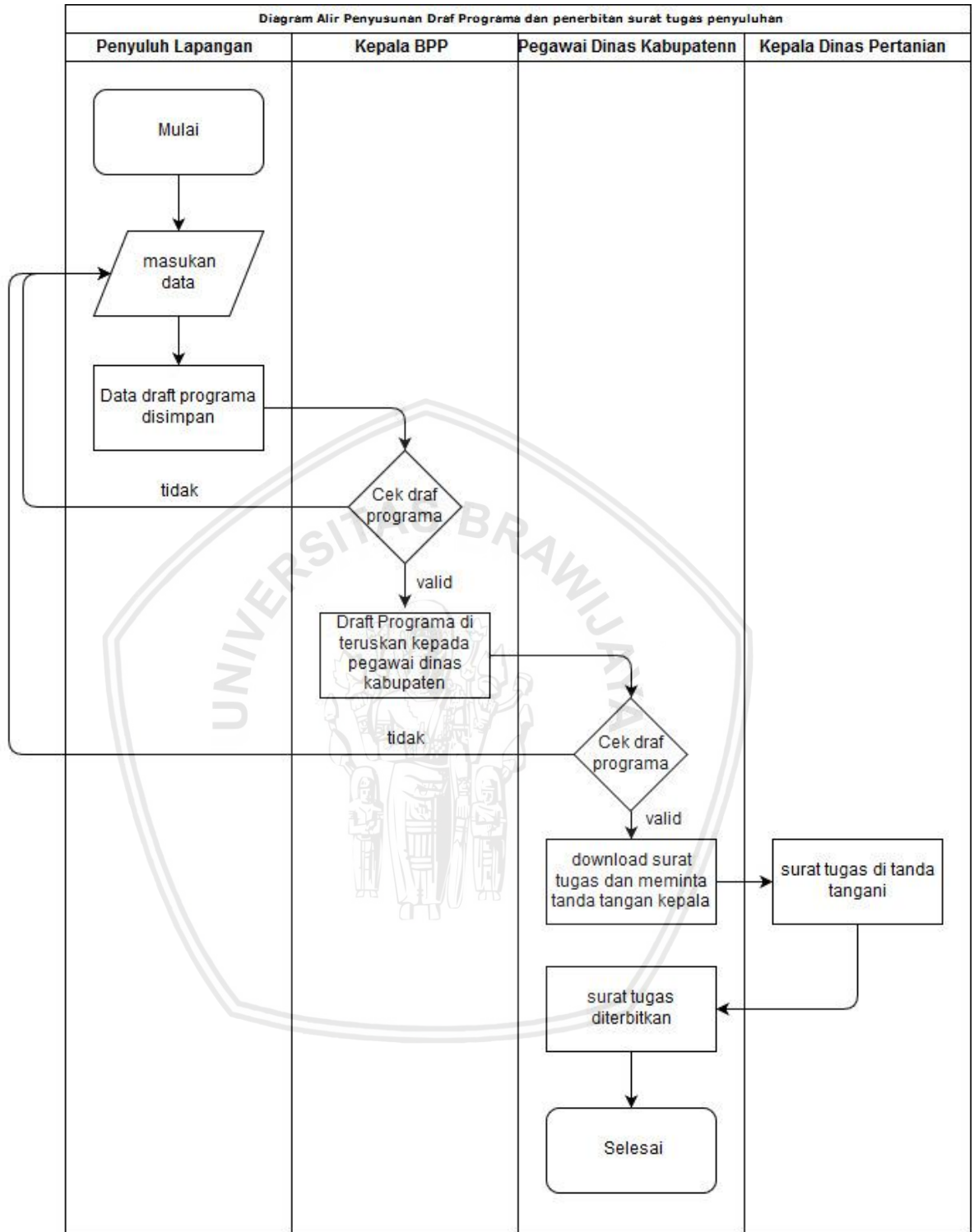
Sistem informasi pertanian di dinas pertanian dan ketahanan pangan kabupaten padang pariaman digambarkan menyediakan beberapa fungsi untuk pengguna. Fungsi yang pertama yaitu menjadi sarana dari proses pengumpulan draft program, pengesahan draft program, sampai penerbitan surat tugas setelah disetujui kepala dinas. Sistem informasi penyuluhan berfungsi dalam pengumpulan laporan penyuluhan sampai pemeriksaan laporan penyuluhan. Sistem informasi penyuluhan berfungsi dalam transparansi pengiriman dana oleh pegawai dinas dari pengumpulan proposal dana, konfirmasi pengiriman dana sampai penerimaan dana oleh penyuluh. Untuk alur yang lebih rinci akan dijelaskan pada subbab proses bisnis masing-masing fungsi kompleks sistem ini.

4.2 Proses Bisnis

Pemodelan proses berguna untuk menggambarkan alur proses sistem informasi penyuluhan ini. Proses bisnis usulan merupakan gambaran dari solusi dalam menyelesaikan masalah. Proses bisnis meliputi alur proses pembuatan draft program, proses bisnis pengumpulan laporan, dan proses bisnis pengiriman dana dari pegawai dinas kepada penyuluh.

4.2.1 Proses Bisnis Pembuatan Draft Program

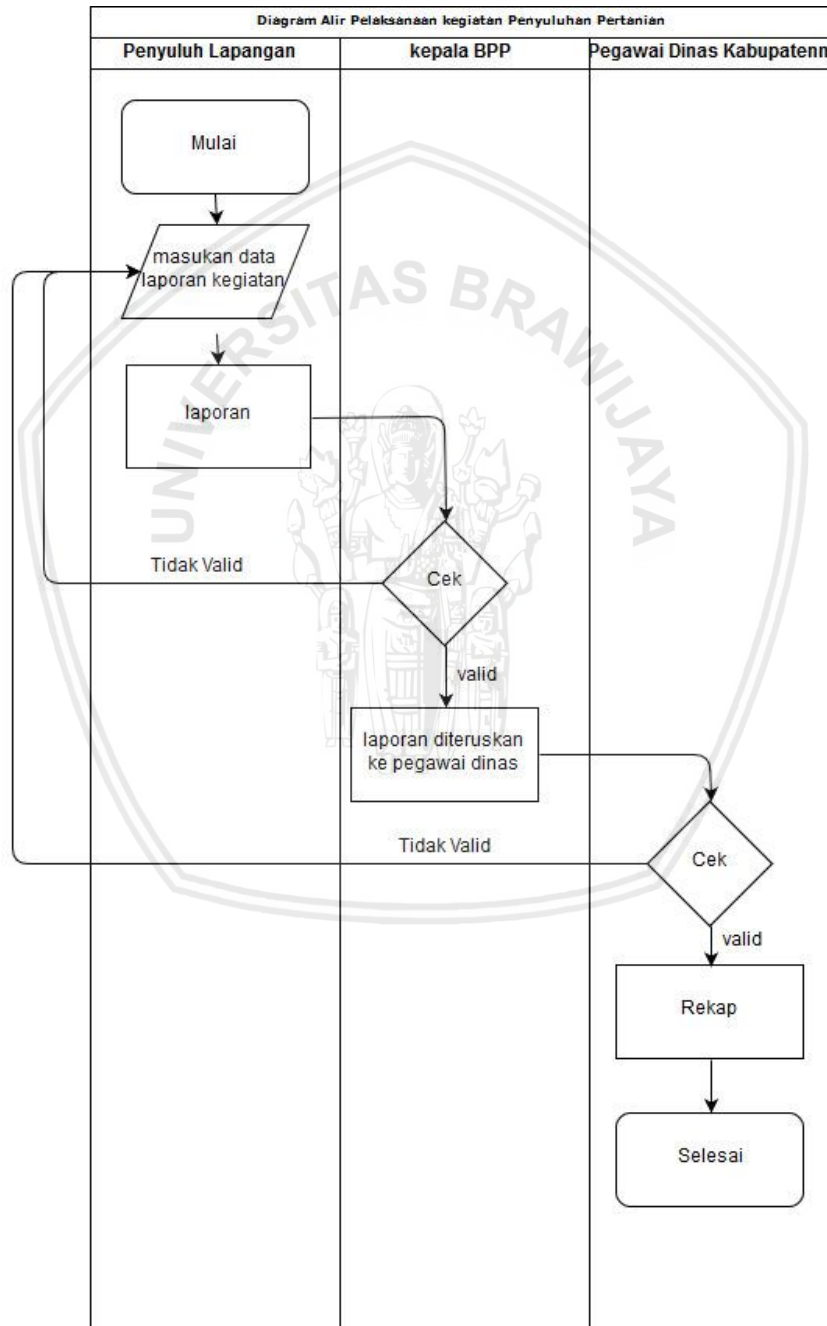
Proses alur bisnis dari pembuatan draft program yaitu langkah pertama penyuluh dapat membuat draft program dengan cara memasukan data pada *form* yang disediakan pada sistem yang nantinya draft program akan melewati dua tahap pemeriksaan, yang pertama melalui pemeriksaan oleh kepala BPP, dan selanjutnya pemeriksaan oleh pegawai dinas. Setelah melewati dua pemeriksaan maka pegawai dinas dapat mendownload surat tugas dari sistem dan meminta tanda tangan kepala dinas untuk disetujui, jika sudah disetujui oleh kepala dinas, maka surat tugas di terbitkan agar penyuluh dapat menjalankan kegiatan penyuluhan sesuai draft program. Untuk alur proses bisa di lihat pada Gambar 4.1.



Gambar 4.1 Proses Bisnis Pembuatan Draft Program dan Penerbitan Surat Tugas

4.2.2 Proses Bisnis Pembuatan laporan Penyuluhan

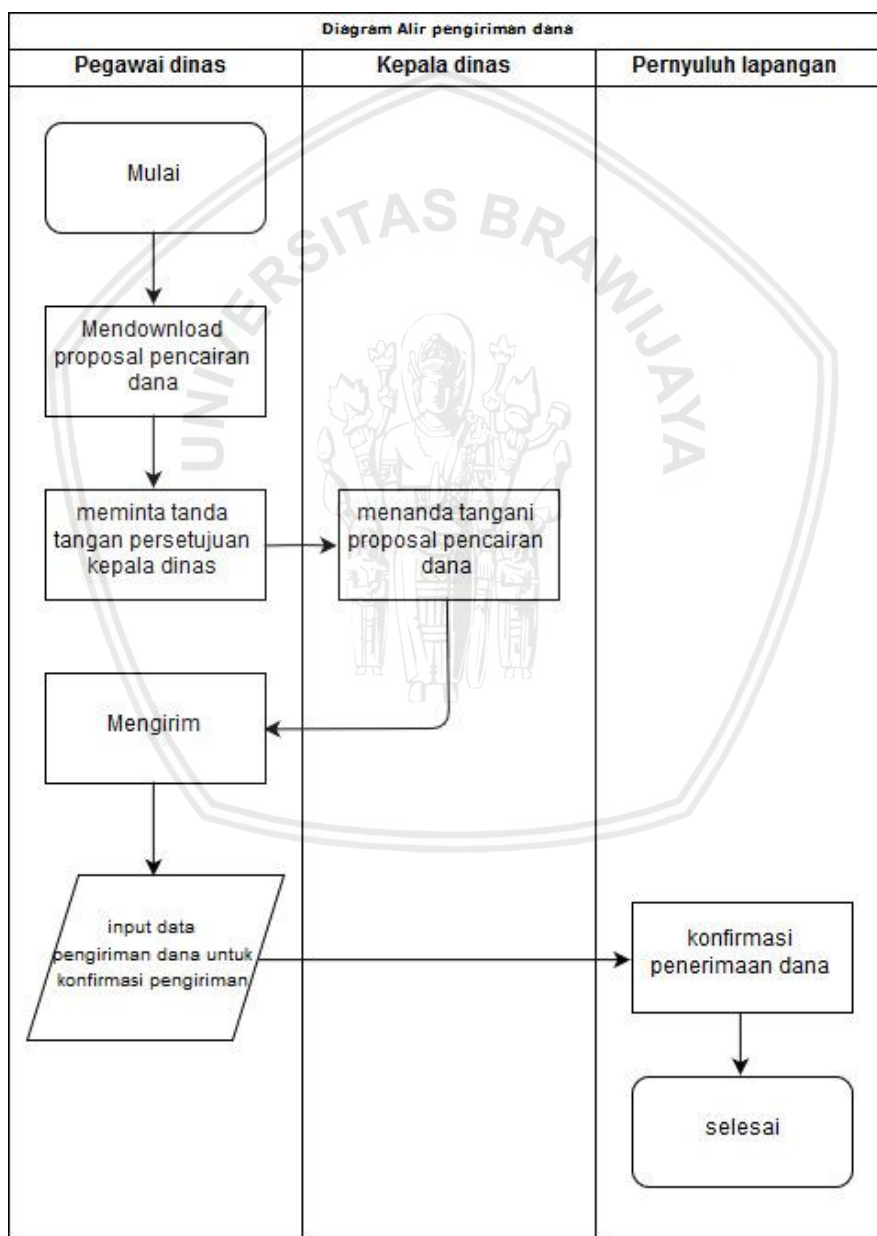
Langkah pertama dalam proses alir bisnis pembuatan laporan penyuluhan yaitu penyuluh dapat membuat laporan yang nantinya laporan kegiatan akan melewati pemeriksaan oleh kepala BPP dan pegawai dinas, jika sudah melewati dua kali pemeriksaan maka laporan kegiatan akan di rekap untuk mempertanggung jawabkan kegiatan yang sudah di lakukan. Untuk alur proses bisnis pengumpulan laporan sampai pemeriksaan penyuluhan dalam bentuk *flowchart* dapat di lihat pada Gambar 4.2.



Gambar 4.2 Proses Bisnis Pembuatan Laporan Kegiatan Penyuluhan

4.2.3 Proses Bisnis Pengiriman Dana

Laporan penyuluhan yang di kumpulkan oleh penyuluh juga melampirkan jumlah dana yang terpakai, dana yang terpakai akan di kirimkan oleh pegawai dinas, untuk pengiriman dana langkah awal yaitu pegawai dinas mendownload proposal dana dari sistem kemudian meminta tanda tangan persetujuan kepada kepala dinas, jika di setuju maka dana akan di kirim oleh pegawai dinas, sistem menyediakan sarana untuk verifikasi pengiriman dana oleh pegawai dinas dan sarana untuk penyuluh melakukan konfirmasi penerimaan dana, sehingga proses pengiriman dana dapat di lakukan lebih transparan. Untuk gambaran proses alur bisnis dalam bentuk *flowchart* dapat di lihat pada Gambar 4.3.



Gambar 4.3 Proses Bisnis Pengiriman Dana

4.3 Elisitasi Kebutuhan

Elisitasi kebutuhan sistem dilakukan dengan menentukan kebutuhan umum yang didapatkan ketika elisitasi kebutuhan. Penggalian atau elisitasi kebutuhan dengan cara wawancara di Dinas Pendidikan dan Ketahanan Pangan Kabupaten Padang Pariaman. Kebutuhan dari penelitian akan didapatkan dari hasil wawancara terhadap narasumber dari Kepala Seksi Penyuluhan ibu irawati febriani,S.P,M.Si, dan penyuluh desa Sungai Asam bapak Yusrizal,S.ST. Dari hasil wawancara yang dilakukan terdapat masalah yang muncul yaitu beberapa penyuluh merasa kesulitan dalam proses pengesahan draft programa, pengumpulan laporan penyuluhan, yang harus di lakukan bertahap-tahap dari proses cek oleh kepala bpp, kemudian cek oleh pegawai dinas, pembuatan surat tugas yang membutuhkan verifikasi dari kepala dinas, hingga menerbitkan surat tugas.

Bedasarkan masalah tersebut, maka didapatkan kebutuhan sebagai solusi dari masalah berupa sistem yang mampu mengurangi kerumitan proses pengesahan draft programa dan pengumpulan laporan kegiatan dengan menyediakan sebuah sarana yang bersifat *online*. Untuk dapat mempermudah dalam proses pengesahan kegiatan draft programa dan pengumpulan laporan kegiatan.

4.4 Analisis Kebutuhan

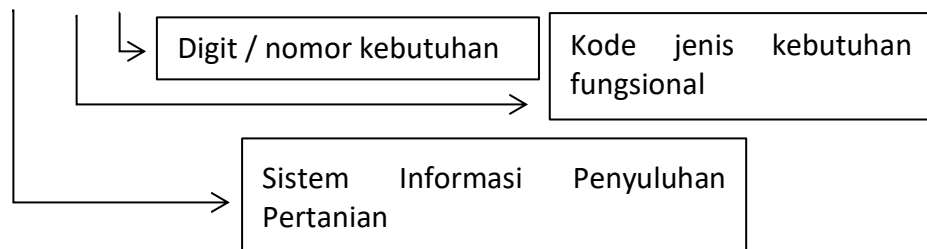
Tahap analisis kebutuhan ini akan melakukan identifikasi kebutuhan pengembangan sistem informasi penyuluhan pertanian. Terdapat beberapa tahapan dalam analisis kebutuhan, antara lain : identifikasi aktor, analisis kebutuhan fungsional, pembuatan *use case* diagram, pembuatan *use case* scenario, dan analisis kebutuhan non fungsional. Analisis kebutuhan disini dimaksudkan untuk memberikan gambaran secara jelas mengenai hal apa saja yang harus disediakan oleh sistem agar memenuhi kebutuhan dari pengguna.

4.4.1 Aturan Penomoran

Kebutuhan Fungsional:

Contoh: SIPP_F_01: Representasi kebutuhan fungsional Sistem Informasi Penyuluhan Pertanian nomor 01.

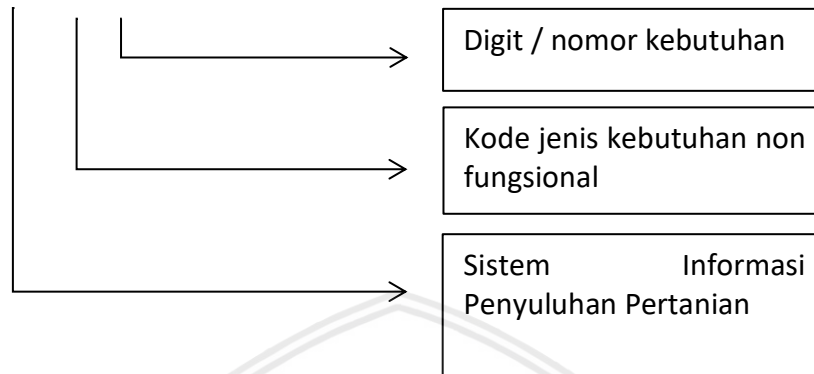
Kode: SIPP_F_01



Kebutuhan Non Fungsional :

Contoh: SIPP_NF_01: Representasi kebutuhan non fungsional Sistem Informasi Penyuluhan pertanian nomor 01.

Kode: SIPP_NF_01



4.4.2 Identifikasi Aktor

Identifikasi aktor yang terlibat pada sistem informasi Sistem informasi penyuluhan pertanian dapat membantu dalam menentukan hak akses akan setiap kebutuhan yang ada. Pada Tabel 4.1 dijabarkan mengenai identifikasi pengguna yang terlibat dalam sistem.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Penyuluh	Penyuluh merupakan orang yang memberikan pengetahuan tentang pertanian kepada kelompok petani. Dalam sistem ini penyuluh dapat membuat draft program, menambahkan laporan kegiatan, dan konfirmasi penerimaan dana.
2	Kepala BPP	Kepala BPP merupakan orang yang mengesalpi penyuluh-penyuluh di sebuah kecamatan. Dalam sistem kepala BPP dapat melakukan verifikasi laporan penyuluhan dan draft program.
3	Pegawai Dinas	Pegawai Dinas merupakan orang yang mengelola data-data terkait penyuluhan pertanian. Dalam sistem pegawai dinas dapat melakukan verifikasi laporan penyuluhan dan draft program, mengajukan surat tugas, mengajukan proposal dana, konfirmasi pengiriman dana, dan menerbitkan surat tugas.
4	Kepala Dinas	Kepala Dinas merupakan orang yang mengesalpi sebuah dinas pertanian dan ketahanan di kabupaten Padang

No	Aktor	Deskripsi
		Pariaman. Dalam sistem Kepala Dinas dapat melakukan verifikasi proposal dana dan surat tugas.
5	Admin Sistem	Admin Sistem merupakan orang yang mengelola data-data pengguna sistem. Admin dapat melakukan tambah, edit, hapus, lihat data <i>user</i> .

4.4.3 Analisis Kebutuhan Fungsional

Fase analisis kebutuhan fungsional dilakukan untuk mengidentifikasi setiap proses-proses yang harus dapat dikerjakan sistem. Penjabaran mengenai kebutuhan fungsional pada Sistem informasi penyuluhan pertanian dapat dilihat pada Tabel 4.2.

Tabel 4.2 Kebutuhan Fungsional

Kode	Nama Fungsi	Deskripsi	Pengguna
SIPP_F_001	<i>Login</i>	<p>Sistem harus mampu menyediakan fungsi untuk pengguna masuk ke sistem agar sistem mengenali pengguna.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan form yang terdiri dari <i>input username</i> dan <i>password</i>. 2. Sistem menyediakan form yang terdiri dari tombol <i>login</i> untuk memulai pemeriksaan data <i>input</i>. 3. Sistem menyediakan pesan peringatan jika <i>username</i> dan <i>password</i> yang di inputkan salah. 	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas, Admin Sistem.
SIPP_F_002	Lihat Profil	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan data diri pengguna.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan data diri pengguna yang berisi 	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.

Kode	Nama Fungsi	Deskripsi	Pengguna
		<p>foto, <i>username</i>, <i>email</i>, <i>password</i>, nip, jabatan.</p> <p>2. Sistem menyediakan tombol ubah data untuk menuju halaman ubah data diri pengguna.</p>	
SIPP_F_003	Ubah Data profil	<p>Sistem harus mampu menyediakan fungsi untuk memperbarui data diri pengguna.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol ganti foto untuk <i>input</i> foto baru. 2. Sistem menyediakan form untuk memasukan data yang terdiri dari nama, <i>username</i>, <i>email</i>, <i>password</i>, dan nip. 3. Sistem menyediakan tombol simpan untuk memperbarui data yang sudah di masukan. 4. Sistem menyediakan pesan peringatan jika data yang di inputkan kosong atau tidak sesuai. 	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.
SIPP_F_004	Lihat daftar notifikasi	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan semua daftar notifikasi.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan halaman daftar notifikasi yang berisi nama pengirim, status, tanggal notifikasi. 2. Sistem menampilkan semua daftar notifikasi yang di urutkan berdasarkan notifikasi terbaru. 	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.

Kode	Nama Fungsi	Deskripsi	Pengguna
		3. Sistem menyediakan tombol lihat untuk menuju halaman detail notifikasi.	
SIPP_F_005	Lihat Detail Notifikasi	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan halaman yang berisi detail notifikasi.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan halaman yang berisi detail notifikasi meliputi pengirim, isi notifikasi, jadwal notifikasi.</p>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.
SIPP_F_006	Buat Draft Programa	<p>Sistem harus mampu menyediakan fungsi untuk membuat draft programa.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan halaman untuk membuat draft programa yang di buat secara bertahap.</p> <p>2. Sistem meyediakan form latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan non perilaku, masalah bersifat perilaku dan non perilaku. Semua form yang di sediakan mempunyai syarat maksimal 1000 kata, jika melebihi syarat maka sistem menampilkan pesan peringatan.</p> <p>3. Sistem menyediakan form untuk <i>input</i> berkas draft programa dalam format doc atau pdf.</p>	Penyuluh.

Kode	Nama Fungsi	Deskripsi	Pengguna
		<p>4. Sistem menyediakan tombol sebelum dan selanjutnya untuk berpindah ke step sebelum atau selanjutnya.</p> <p>5. Sistem menyediakan tombol <i>submit</i> untuk <i>upload</i> data draft program yang sudah di <i>input</i> dan menuju halaman masukan jadwal kegiatan.</p>	
SIPP_F_007	Masukan Jadwal Kegiatan	<p>Sistem harus mampu menyediakan fungsi untuk memasukan jadwal kegiatan penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> untuk memasukan data jadwal kegiatan yaitu berupa <i>form</i> materi, jadwal, tempat, dan tombol simpan untuk menyimpan data jadwal kegiatan. Sistem juga menampilkan data-data kegiatan berupa tabel yang otomatis akan bertambah ketika jadwal kegiatan di masukan. 2. Sistem menyediakan <i>form</i> untuk menambahkan detail pengeluaran dana yaitu <i>form</i> barang, jumlah, harga satuan, dan harga total, kemudian tombol simpan untuk menyimpan data detail pengeluaran. 3. Sistem menyediakan tombol selesai untuk menyimpan data draft program dan jadwal kegiatan beserta detail pengeluaran dana 	Penyuluh.
SIPP_F_08	Lihat Daftar Draft Program	Sistem harus mampu menyediakan fungsi untuk	Penyuluh, Kepala

Kode	Nama Fungsi	Deskripsi	Pengguna
		<p>menampilkan daftar draft program.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan Tabel untuk menampilkan daftar draft program. 2. Sistem menyediakan data draft program berupa nama penyuluh, nama desa, nama kecamatan, tanggal pembuatan draft, status dan tombol lihat untuk menuju halaman detail draft program. 	<p>BPP, Pegawai Dinas.</p>
SIPP_F_09	Lihat Detail Draft Program	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan detail draft program.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan informasi data-data draft program meliputi tanggal pembuatan draft program, nama penyuluh, desa, kecamatan, dana yang di butuhkan, status, latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan non perilaku, masalah bersifat perilaku dan non perilaku. 2. Sistem menyediakan fungsi untuk mendownload berkas draft program. 3. Sistem menyediakan tombol lihat kelayakan untuk melihat kelayakan draft program 	<p>Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.</p>

Kode	Nama Fungsi	Deskripsi	Pengguna
		<p>jika sudah di verifikasi kepala bpp atau pegawai dinas.</p> <p>4. Sistem menyediakan data jadwal kegiatan dan detail pengeluaran dalam bentuk tabel.</p> <p>5. Sistem menyediakan tombol <i>process</i> untuk verifikasi program bagi kepala bpp dan pegawai dinas, dan untuk penyuluh, sistem menyediakan tombol kembali dan tombol lihat surat tugas jika surat tugas sudah di terbitkan.</p>	
SIPP_F_10	Verifikasi Draft Program	<p>Sistem harus mampu menyediakan fungsi untuk memverifikasi draft program.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan daftar kelayakan, ceklis kelayakan draft program, form penjelasan untuk kepala bpp atau pegawai dinas memasukan penjelasan kelayakan, form untuk memasukan dana yang disetujui.</p> <p>2. Sistem menyediakan tombol <i>submit</i> untuk memverifikasi draft program, dengan ketentuan jika salah satu ceklis kelayakan tidak terpenuhi maka status tidak valid, dan jika semua ceklis kelayakan sudah tercentang maka status draft program di setujui oleh kepala bpp atau pegawai dinas.</p>	Kepala BPP, Pegawai Dinas

Kode	Nama Fungsi	Deskripsi	Pengguna
SIPP_F_011	Lihat Jadwal dan Laporan Penyuluhan	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan jadwal dan laporan penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan Tabel untuk menampilkan daftar kegiatan penyuluhan. 2. Sistem menampilkan Tabel yang berisi data nama penyuluh, desa, kecamatan, tanggal kegiatan, status laporan, dan tombol lihat untuk menuju halaman detail laporan kegiatan. 	Penyuluh, Kepala BPP, pegawai dinas.
SIPP_F_012	Buat Laporan Penyuluhan	<p>Sistem harus mampu menyediakan fungsi untuk membuat laporan penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar jadwal kegiatan yang akan di pilih untuk di buat laporan, jika status laporan belum dikirim maka penyuluh dapat membuat laporan, jika tidak maka sistem akan menampilkan pesan peringatan. 2. Sistem menyediakan form untuk memasukan konten penyuluhan, dana yang terpakai, berkas laporan penyuluhan, presensi, foto kegiatan penyuluhan yang dapat di inputkan lebih dari satu, foto bukti pembayaran atau pengeluaran dana. 3. Sistem menyediakan tombol simpan untuk menyimpan laporan penyuluhan. 	Penyuluh

Kode	Nama Fungsi	Deskripsi	Pengguna
SIPP_F_013	Lihat Detail Laporan Penyuluhan	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan detail laporan penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan Tabel untuk menampilkan daftar laporan penyuluhan. 2. Sistem menyediakan data Tabel berupa tema penyuluhan, tanggal penyuluhan, status. 3. Sistem menyediakan tombol lihat untuk menuju halaman detail penyuluhan. 4. Pada halaman detail penyuluhan sistem menampilkan informasi berupa tema, status, dana terpakai, tanggal laporan di buat, <i>content</i> laporan penyuluhan, berkas laporan yang bisa di <i>download</i>, foto presensi, daftar foto kegiatan, tombol kembali untuk penyuluh kembali ke halaman sebelumnya, tombol proses untuk kepala bpp atau pegawai dinas untuk menuju halaman verifikasi laporan penyuluhan, Dan sistem menyediakan tombol lihat kelayakan untuk melihat kelayakan laporan kegiatan. 	Penyuluh, Kepala BPP, Pegawai Dinas
SIPP_F_014	Verifikasi Laporan Penyuluhan	<p>Sistem harus mampu menyediakan fungsi untuk memverifikasi laporan penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p>	Penyuluh, Kepala BPP, Pegawai Dinas

Kode	Nama Fungsi	Deskripsi	Pengguna
		<ol style="list-style-type: none"> 1. Sistem menyediakan daftar kelayakan, form ceklis kelayakan draft programa, dan form penjelasan untuk kepala bpp atau pegawai dinas memasukan penjelasan kelayakan. 2. Sistem menyediakan tombol <i>submit</i> untuk memverifikasi laporan penyuluhan, dengan ketentuan jika salah satu ceklis kelayakan tidak terpenuhi maka status tidak valid, dan jika semua ceklis kelayakan sudah tercentang maka status laporan penyuluhan di setujui oleh kepala bpp atau pegawai dinas. 	
SIPP_F_015	Ajukan Surat Tugas	<p>Sistem harus mampu menyediakan fungsi untuk mengajukan surat tugas penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar draft programa dengan status valid oleh pegawai dinas yang akan di pilih untuk di buat surat tugas. 2. Sistem menyediakan tombol ajukan yang ketika ditekan akan menampilkan popup apakah anda ingin mengajukan surat tugas. 	Pegawai Dinas
SIPP_F_016	Lihat Daftar Surat Tugas	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan daftar surat tugas kegiatan penyuluhan.</p> <p>Spesifikasi Kebutuhan :</p>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas



Kode	Nama Fungsi	Deskripsi	Pengguna
		<ol style="list-style-type: none"> 1. Sistem menyediakan Tabel yang menampilkan daftar surat tugas. 2. Sistem menampilkan data surat tugas berupa desa, tanggal surat tugas, status surat tugas. sistem menyediakan tombol lihat untuk menampilkan surat tugas yang akan dicetak, dan <i>download</i> untuk mengunduh surat tugas. 	
SIPP_F_017	Verifikasi Surat Tugas	<p>Sistem harus mampu menyediakan fungsi untuk memverifikasi surat tugas.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediaka tombol tolak dan setuju, tombol tolak akan menampilkan <i>popup</i> untuk mengkonfirmasi penolakan surat tugas. dan tombol setuju akan menampilkan <i>popup</i> untuk konfirmasi penerimaan surat tugas. 	Kepala Dinas
SIPP_F_018	Terbitkan Surat Tugas	<p>Sistem harus mampu menyediakan fungsi untuk menerbitkan surat tugas.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol terbitkan pada halaman detail surat tugas untuk menerbitkan surat tugas dengan ketentuan status surat tugas sudah di validasi oleh kepala dinas. 2. Sistem menampilkan pesan peringatan sebagai <i>feedback</i> dari sistem bahwa surat tugas sudah di terbitkan. 	Pegawai Dinas

Kode	Nama Fungsi	Deskripsi	Pengguna
SIPP_F_019	Ajukan Proposal Dana	<p>Sistem harus mampu menyediakan fungsi untuk mengajukan proposal dana untuk menurunkan dana kepada penyuluh yang harus di setujui oleh kepala dinas.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar draft programa dengan status surat tugas di terbitkan yang akan di pilih untuk di buat proposal dana. 2. Sistem menyediakan tombol ajukan untuk mengajukan proposal dana kepada kepala dinas. 	Pegawai Dinas
SIPP_F_020	Lihat Daftar Proposal Dana	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan daftar proposal dana.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan Tabel untuk menampilkan daftar proposal dana. 2. Sistem menyediakan data berupa nama penyuluh, jumlah dana yang dikirim, desa, kecamatan, status proposal dana. 3. Sistem menyediakan tombol lihat untuk menampilkan proposal dana dan tombol <i>download</i> untuk mengunduh proposal dana. 	Pegawai Dinas, Kepala Dinas
SIPP_F_021	Verifikasi Proposal Dana	<p>Sistem harus mampu menyediakan fungsi untuk memverifikasi proposal dana.</p> <p>Spesifikasi Kebutuhan :</p>	Kepala Dinas

Kode	Nama Fungsi	Deskripsi	Pengguna
		1. Sistem menyediakan tombol tolak dan setuju, tombol tolak akan menampilkan <i>popup</i> untuk mengkonfirmasi penolakan surat tugas. dan tombol setuju akan menampilkan <i>popup</i> untuk konfirmasi penerimaan surat tugas.	
SIPP_F_022	Lihat Daftar Pengiriman Dana	Sistem harus mampu menyediakan fungsi untuk menampilkan daftar pengiriman dana. Spesifikasi Kebutuhan : 1. Sstem menyediakan Tabel daftar pengiriman dana. 2. Sistem meyediakan data berupa nama penyuluh, desa, kecamatan, jumlah dana yang dikirim, tanggal dikirim, dan status. 3. Sistem menyediakan tombol lihat untuk menuju halaman detail pengiriman dana.	Penyuluh, Pegawai dinas
SIPP_F_023	Lihat Detail Pengiriman Dana	Sistem harus mampu menyediakan fungsi untuk menampilkan data pengiriman dana. Spesifikasi Kebutuhan : 1. Sistem menampilkan informasi berupa jumlah dana yang dikirim, nomor rekening pengirim, nama pengirim, bank pengirim, tanggal dikirim, status, dan foto bukti pengirima. 2. Sistem menyediakan tombol kembali untuk kembali ke halaman sebelumnya.	Penyuluh, Pegawai dinas

Kode	Nama Fungsi	Deskripsi	Pengguna
		3. Untuk penyuluh sistem menyediakan tombol konfirmasi penerimaan dana.	
SIPP_F_024	Konfirmasi penerimaan dana	<p>Sistem harus mampu menyediakan fungsi untuk mengkonfirmasi penerimaan dana.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan tombol konfirmasi penerimaan dana pada halaman detail pengiriman dana. 2. Sistem akan memunculkan <i>popup</i> ketika menekan tombol konfirmasi penerimaan dana untuk konfirmasi penerimaan dana. 	Penyuluh
SIPP_F_025	Lihat Daftar Pengguna	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan daftar pengguna.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan Tabel untuk menampilkan daftar penyuluh. 2. Sistem menyediakan data pengguna berupa nama, desa, kecamatan. 3. Sistem menyediakan tombol lihat untuk menuju halaman detail data penyuluh. 	Admin Sistem
SIPP_F_026	Tambah Pengguna	<p>Sistem harus mampu menyediakan fungsi untuk menambah data penyuluh.</p> <p>Spesifikasi Kebutuhan :</p> <ol style="list-style-type: none"> 1. Sistem menyediakan <i>form</i> nama, email, nomor telepon, nip, <i>password</i>, desa, kecamatan. 	Admin Sistem

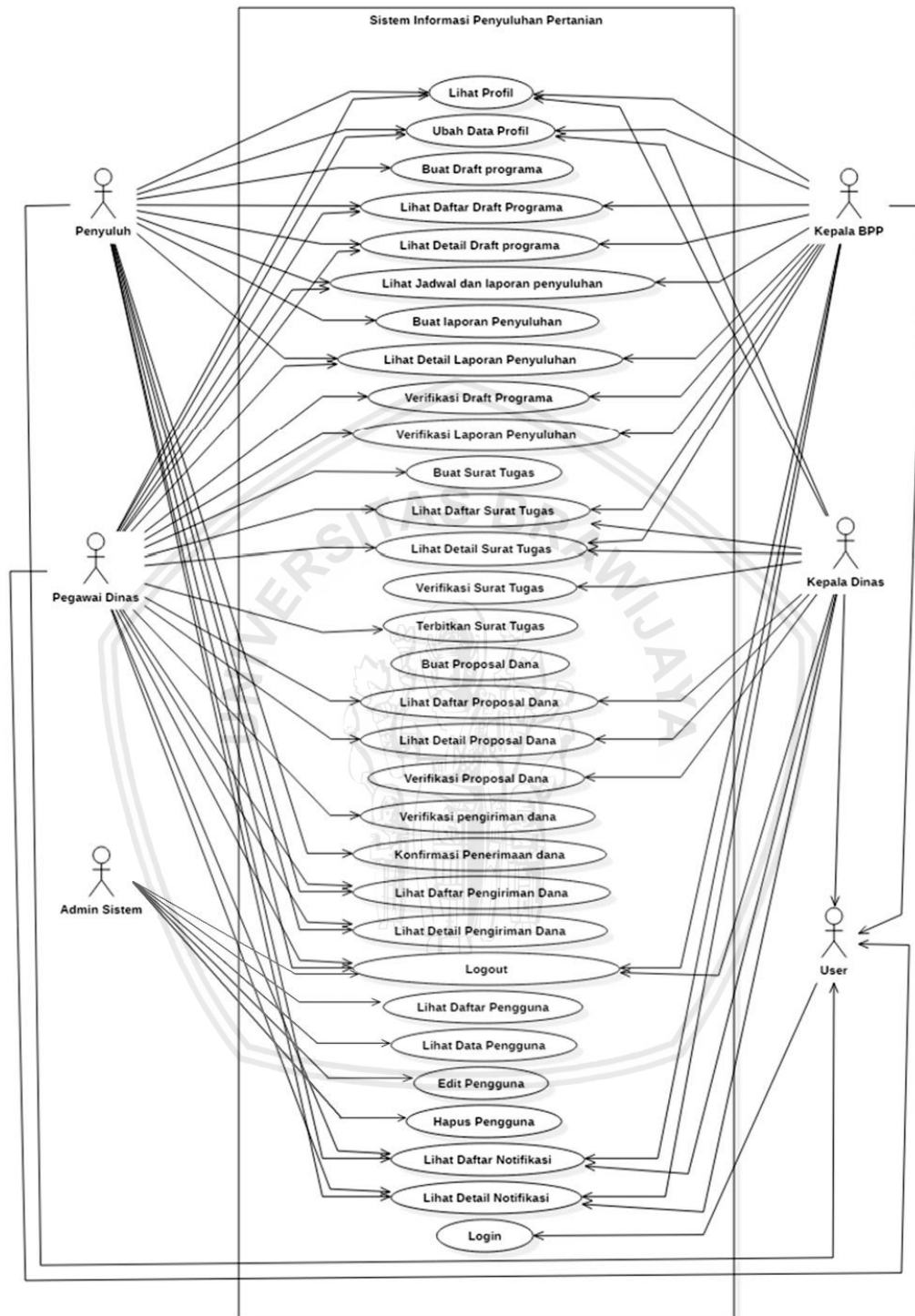
Kode	Nama Fungsi	Deskripsi	Pengguna
		<p>2. Sistem menyediakan form untuk menginputkan foto penyuluh.</p> <p>3. Sistem menyediakan tombol simpan untuk menyimpan data penyuluh.</p>	
SIPP_F_027	Lihat Data Pengguna	<p>Sistem harus mampu menyediakan fungsi untuk menampilkan detail data diri penyuluh.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan halaman yang menampilkan data diri pengguna berupa nama, email, nomor telepon, nip, desa, kecamatan, dan foto penyuluh.</p>	Admin Sistem
SIPP_F_028	Edit Pengguna	<p>Sistem harus mampu menyediakan fungsi untuk mengedit data penyuluh.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan tombol edit pada halaman detail data penyuluh untuk menampilkan pesan peringatan.</p> <p>2. Sistem menampilkan pesan peringatan yang berisi form nama, email, nomor telepon, nip, desa, kecamatan, <i>password</i>, dan tombol simpan untuk memperbarui data pengguna.</p>	Admin Sistem
SIPP_F_029	Hapus Pengguna	<p>Sistem harus mampu menyediakan fungsi untuk menghapus penyuluh.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan tombol hapus untuk menghapus</p>	Admin Sistem



Kode	Nama Fungsi	Deskripsi	Pengguna
		<p>menampilkan pesan peringatan.</p> <p>2. Sistem menampilkan pesan peringatan yang berisi pesan konfirmasi penghapusan data penyuluh dan tombol simpan untuk menghapus data penyuluh.</p>	
SIPP_F_030	<i>Logout</i>	<p>Sistem harus mampu menyediakan fungsi untuk <i>user</i> keluar dari halaman <i>user</i>.</p> <p>Spesifikasi Kebutuhan :</p> <p>1. Sistem menyediakan tombol keluar untuk menuju halaman <i>login</i>.</p>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas, Admin Sistem.

4.4.4 Pembuatan *use case diagram*

Use case diagram akan menjabarkan sebuah hubungan antar proses-proses pada sistem dengan pelaku atau aktor Pada *use case diagram* tergambar semua kebutuhan fungsional yang telah diidentifikasi sebelumnya dan dihubungkan dengan aktor-aktor yang berhak mengaksesnya seperti yang tertera pada Gambar 4.4.



Gambar 4.4 Use Case Diagram

4.4.5 Pembuatan use case scenario

Use case scenario menjabarkan secara detail dari langkah-langkah aktor dan sistem dalam mencapai tujuan dari salah satu kebutuhan fungsional yang telah



didefinisikan di tabel sebelum ini yaitu Tabel 4.2. Pada Tabel 4.3 sampai Tabel 4.33 merupakan *use case scenario* dari sistem yang dibangun.

1. *Use Case Login*

Tabel 4.3 Use Case Login

Skenario <i>Use Case</i> untuk Login (SIPP_F_001)	
<i>Objective</i>	<i>Use Case</i> ini berfungsi untuk mengenali <i>user</i> dan memberi hak akses kepada <i>user</i> mengakses halaman sesuai otoritas <i>user</i> .
<i>Actor</i>	Tamu.
<i>Pre-condition</i>	<i>User</i> berada pada halaman <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Pengguna memasukkan <i>email</i>. 2. Pengguna memasukkan <i>password</i>. 3. Pengguna menekan tombol masuk. 4. Sistem melakukan autentikasi terhadap masukan yang berupa <i>email</i> dan <i>password</i>. 5. Berhasil masuk ke halaman aktor.
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>email</i> dan <i>password</i> tidak di isi maka proses <i>login</i> tidak dapat di lanjutkan dan sistem meminta <i>user</i> untuk mengisi kolom yang kosong. 2. jika <i>email</i> atau <i>password</i> salah maka Sistem menampilkan informasi bahwa “<i>Email</i> atau <i>password</i> salah”.
<i>Post-condition</i>	Aktor berhasil masuk ke halaman masing-masing yang sudah di tentukan .

Tabel 4.3 merupakan *use case scenario* dari *login*. Aktor yang dapat melakukan login yaitu tamu. Langkah yang pertama yaitu dengan mengakses halaman *login*, aktor memasukan data berupa *email* dan *password*. Kemudian menekan tombol *login* untuk dapat melakukan pemeriksaan terhadap *email* dan *password*. Jika berhasil maka akan menuju halaman masing-masing aktor yang sudah di tentukan. *Alternative flow* yaitu ketika *password* atau *email* salah maka sistem akan menampilkan pesan *email* atau *password* salah. *Alternatif flow* yang kedua yaitu ketika *email* atau *password* dikosongkan maka sistem tidak dapat melanjutkan pemeriksaan dan akan fokus kepada form yang kosong agar diisi.

2. Use Case Lihat Profil

Tabel 4.4 Use Case Lihat Profil

Skenario Use Case untuk Lihat Profil (SIPP_F_002)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas dapat melihat data diri.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas sudah melakukan login.
<i>Main Success Scenario</i>	4. Aktor menekan tombol profil. 5. Sistem akan <i>direct</i> ke halaman profil dan menampilkan data diri.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat data diri.

Tabel 4.4 merupakan *use case* untuk lihat profil. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, pegawai dinas, dan kepala dinas. Fungsionalitas ini bertujuan untuk melihat data diri pengguna dengan menekan tombol profil pada *sidebar* maka sistem akan menuju halaman profil dan sistem akan menampilkan data diri pengguna.

3. Use Case Ubah Data profil

Tabel 4.5 Use Case Ubah Data profil

Skenario Use Case untuk Ubah Data Profil (SIPP_F_003)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas dapat merubah data diri.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas sudah melakukan login dan berada pada halaman profil.
<i>Main Success Scenario</i>	1. Aktor menekan <i>tab</i> ubah data. 2. Sistem akan berpindah ke <i>tab</i> lain dan menampilkan <i>form</i> untuk memasukan data diri yang baru.

Skenario <i>Use Case</i> untuk Ubah Data Profil (SIPP_F_003)	
	<ol style="list-style-type: none"> 3. Aktor memasukan data diri yang baru berupa nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, foto. 4. Aktor menekan tombol simpan. 5. Sistem menyimpan data baru ke dalam <i>database</i> dan kembali ke halaman profil dengan menampilkan pesan “berhasil menyimpan data”.
<i>Alternatif Flow</i>	Jika ada form nama, <i>username</i> , <i>email</i> , <i>nip</i> di kosongkan maka sistem akan menampilkan pesan “ <i>form</i> ini tidak boleh kosong”.
<i>Post-condition</i>	Aktor berhasil memperbarui data diri.

Tabel 4.5 merupakan *use case* untuk ubah data profil. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, pegawai dinas, dan kepala dinas. Fungsionalitas ini bertujuan untuk mengubah data diri pengguna. Dengan cara mengakses halaman profil terlebih dahulu kemudian menekan pilihan ubah data pada menu berbentuk *tab*. Sistem akan menyediakan form untuk menginputkan data baru. Aktor dapat menginputkan data dan menekan tombol simpan. Jika ada form nama, *username*, *email*, *nip* di kosongkan maka sistem akan menampilkan pesan “form ini tidak boleh kosong”. Jika berhasil maka data akan disimpan ke dalam database.

4. *Use Case* Lihat Daftar Notifikasi

Tabel 4.6 *Use Case* Lihat Daftar Notifikasi

Skenario <i>Use Case</i> untuk Lihat Daftar Notifikasi(SIPP_F_004)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas dapat melihat daftar notifikasi.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas sudah melakukan login.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol notifikasi pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman daftar notifikasi. 3. Sistem akan menampilkan daftar notifikasi dalam bentuk tabel. 4. Data notifikasi yang di tampilkan pada tabel berupa nama pengirim, jabatan, tanggal,



	deksripsi, dan tombol lihat untuk melihat detail notifikasi.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat daftar notifikasi.

Tabel 4.6 merupakan *use case* untuk lihat daftar notifikasi. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, pegawai dinas, dan kepala dinas. Fungsionalitas ini bertujuan untuk melihat daftar notifikasi. Notifikasi akan ditampilkan dalam bentuk table dan di sediakan tombol lihat untuk menuju halaman detail notifikasi

5. *Use Case* Lihat Detail Notifikasi

Tabel 4.7 Use Case Lihat Detail Notifikasi

Skenario <i>Use Case</i> untuk Lihat Detail Notifikasi (SIPP_F_005)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas dapat melihat data diri.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas, Kepala Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas, dan kepala dinas sudah melakukan login dan berada pada halaman daftar notifikasi.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat pada halaman daftar notifikasi. 2. Sistem akan <i>direct</i> ke halaman detail notifikasi dan menampilkan data berkas yang berhubungan dengan notifikasi. 3. Jika notifikasi merupakan pemberitahuan tentang laporan maka akan menampilkan detail laporan, jika draft maka akan menampilkan detail draft programa, jika Proposal dana maka akan menampilkan data proposal dana, jika surat tugas maka akan menampilkan data surat tugas, jika pengiriman dana, maka akan menampilkan pengiriman dana.
<i>Alternatif Flow</i>	-



Skenario <i>Use Case</i> untuk Lihat Detail Notifikasi (SIPP_F_005)	
<i>Post-condition</i>	Aktor berhasil melihat detail notifikasi.

Tabel 4.7 merupakan *use case* untuk lihat detail notifikasi. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, pegawai dinas, dan kepala dinas. Fungsionalitas ini berfungsi untuk menampilkan detail notifikasi, detail notifikasi ini nanti akan menampilkan detail dari data yang menyangkut dengan notifikasi. Jika notifikasi merupakan pemberitahuan tentang laporan maka akan menampilkan detail laporan, jika draft maka akan menampilkan detail draft program, jika proposal dana maka akan menampilkan data proposal dana, jika surat tugas maka akan menampilkan data surat tugas, jika pengiriman dana, maka akan menampilkan pengiriman dana.

6. *Use Case* Buat Draft Program

Tabel 4.8 *Use Case* Buat Draft Program

Skenario <i>Use Case</i> untuk Buat Draft Program (SIPP_F_006)	
<i>Objective</i>	Penyuluh dapat membuat draft program.
<i>Actor</i>	Penyuluh.
<i>Pre-condition</i>	Penyuluh sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol draft program kemudian memilih menu buat draft program pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman buat draft program. 3. Sistem menampilkan form yang dapat di <i>input</i> secara bertahap tahap, form terdiri dari form latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan non perilaku, masalah bersifat perilaku dan non perilaku, jumlah dana yang di butuhkan, dan <i>form</i> berkas <i>draft</i> program. 4. Sistem menyediakan tombol submit, ketika di tekan akan menyimpan data draft program dan menuju halaman <i>input</i> jadwal kegiatan.
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku



Skenario <i>Use Case</i> untuk Buat Draft Programa (SIPP_F_006)	
	<p>dan <i>non</i> perilaku, masalah bersifat perilaku dan <i>non</i> perilaku diisi lebih dari 1000 kata maka sistem akan menampilkan peringatan bahwa tidak boleh mengisi lebih dari 1000 kata.</p> <p>2. Jika salah satu dari <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat perilaku dan <i>non</i> perilaku, jumlah dana yang dibutuhkan, berkas <i>draft</i> programa maka sistem akan menampilkan pesan <i>error</i> "<i>form</i> tidak boleh kosong" pada masing-masing <i>form</i> yang kosong. Jika berkas <i>draft</i> programa tidak berformat <i>docx</i> atau <i>pdf</i>. Maka sistem akan menampilkan pesan <i>error</i> "berkas harus berformat <i>docx</i>, <i>pdf</i>" pada <i>form</i> berkas <i>draft</i> programa.</p>
<i>Post-condition</i>	Aktor berhasil membuat <i>draft</i> programa.

Tabel 4.8 merupakan *use case* untuk buat *draft* programa. Fungsionalitas ini untuk aktor penyuluh. Fungsionalitas ini untuk membuat *draft* programa baru oleh penyuluh. Dengan cara memasukan data pada *form* yang tersedia, jika *form* ada yang kosong dan berkas tidak berformat *pdf* dan *docx* maka sistem akan menampilkan pesan *error* pada masing-masing *form*. Jika berhasil sistem akan menampilkan pesan "berhasil menyimpan data".

7. *Use Case* Masukan Jadwal Kegiatan

Tabel 4.9 *Use Case* Masukan Jadwal Kegiatan

Skenario <i>Use Case</i> untuk Buat Draft Programa (SIPP_F_007)	
<i>Objective</i>	Penyuluh dapat memasukan jadwal kegiatan.
<i>Actor</i>	Penyuluh.
<i>Pre-condition</i>	Penyuluh sudah melakukan <i>login</i> dan berada pada halaman buat <i>draft</i> programa.
<i>Main Success Scenario</i>	1. Sistem menyediakan <i>form</i> untuk memasukan data kegiatan berupa <i>form</i> materi, tempat,



Skenario <i>Use Case</i> untuk Buat <i>Draft</i> Programa (SIPP_F_007)	
	<p>tanggal, dan tombol submit untuk menyimpan data laporan ke dalam <i>database</i>.</p> <p>2. Sistem menyediakan <i>form</i> untuk memasukan data detail penggunaan dana, yaitu berupa <i>form</i> nama barang, jumlah barang, harga satuan, total harga, dan tombol simpan untuk menyimpan detail penggunaan dana.</p> <p>3. Sistem menyediakan tombol selesai untuk menyimpan data kedalam <i>database</i> dan kembali ke halaman buat <i>draft</i> programa dengan menampilkan pesan “berhasil menyimpan data”.</p>
<i>Alternatif Flow</i>	1. Jika <i>form</i> materi, tanggal, dan jadwal di kosongkan maka sistem akan menampilkan pesan <i>error</i> “ <i>form</i> tidak boleh kosong”.
<i>Post-condition</i>	Aktor berhasil memasukan jadwal kegiatan.

Tabel 4.9 merupakan *use case* untuk memasukan jadwal penyuluhan. Fungsionalitas ini untuk memasukan jadwal penyuluhan oleh aktor penyuluh. Dengan cara memasukan data pada *form* yang tersedia, jika *form* ada yang kosong sistem akan menampilkan pesan *error* pada masing-masing *form*. Jika berhasil sistem akan menampilkan pesan “berhasil menyimpan data”.

8. *Use Case* Lihat Daftar *Draft* Programa

Tabel 4.10 *Use Case* Lihat Daftar *Draft* Programa

Skenario <i>Use Case</i> untuk Lihat Daftar <i>Draft</i> Programa (SIPP_F_008)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas dapat melihat daftar <i>draft</i> programa.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, dan pegawai dinas sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<p>1. Aktor menekan tombol <i>draft</i> programa dan memilih menu daftar <i>draft</i> pada <i>sidebar</i>.</p> <p>2. Sistem akan <i>direct</i> ke halaman daftar <i>draft</i> programa.</p> <p>3. Sistem menampilkan daftar <i>draft</i> dalam bentuk tabel.</p>

Skenario <i>Use Case</i> untuk Lihat Daftar <i>Draft</i> Programa (SIPP_F_008)	
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat daftar <i>draft</i> programa.

Tabel 4.10 merupakan *use case* untuk lihat daftar *draft* programa. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, dan pegawai dinas. Fungsionalitas ini untuk menampilkan data data *draft* programa. Dengan menekean tombol *draft* programa pada *sidebar* sistem akan menuju halaman daftar *draft* programa dan akan menampilkan daftar *draft* programa dalam bentuk tabel.

9. *Use Case* Lihat Detail *Draft* Programa

Tabel 4.11 *Use Case* Lihat Detail *Draft* Programa

Skenario <i>Use Case</i> untuk Lihat Detail <i>Draft</i> Programa (SIPP_F_009)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas dapat melihat detail <i>draft</i> programa.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas sudah melakukan <i>login</i> dan berada pada halaman daftar <i>draft</i> programa.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat pada halaman daftar <i>draft</i> programa. 2. Sistem akan <i>direct</i> ke halaman detail <i>draft</i> programa. 3. Sistem menampilkan detail <i>draft</i> programa berupa data latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan Sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat perilaku dan <i>non</i> perilaku, jumlah dana yang di butuhkan, berkas <i>draft</i> programa, tanggal pembuatan <i>draft</i> programa.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat detail <i>draft</i> programa.

Tabel 4.11 merupakan *use case* untuk lihat *draft* program. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, dan pegawai dinas. Fungsionalitas ini untuk menampilkan data *draft* program. Dengan menekan tombol lihat pada halaman daftar *draft* program maka sistem akan menuju halaman detail *draft* program dan menampilkan data-data *draft* program.

10. Use Case Verifikasi Draft Program

Tabel 4.12 Use Case Verifikasi Draft Program

Skenario Use Case untuk Verifikasi Draft Program (SIPP_F_010)	
<i>Objective</i>	Kepala BPP dan pegawai dinas dapat memverifikasi <i>draft</i> program.
<i>Actor</i>	Kepala BPP, dan pegawai dinas.
<i>Pre-condition</i>	Kepala BPP dan pegawai dinas sudah melakukan <i>login</i> dan berada pada halaman detail <i>draft</i> program.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>process</i> pada halaman detail <i>draft</i> program. 2. Sistem akan <i>direct</i> ke halaman verifikasi <i>draft</i> program. 3. Sistem menyediakan centang pada kelayakan <i>draft</i> program, <i>form</i> penjelasan untuk memberi penjelasan terkait kelayakan <i>draft</i> program, dan tombol simpan. 4. Aktor menekan tombol simpan, sistem menyimpan data verifikasi ke dalam <i>database</i> dan kembali ke halaman detail <i>draft</i> program dengan menampilkan pesan berhasil menyimpan data
<i>Alternatif Flow</i>	Jika salah satu form penjelasan kelayakan di kosongkan maka sistem akan menampilkan pesan " <i>form</i> tidak boleh kosong" pada setiap form yang kosong.
<i>Post-condition</i>	Aktor berhasil memverifikasi <i>draft</i> program.

Tabel 4.12 merupakan *use case* untuk memverifikasi *draft* program. Fungsionalitas ini untuk aktor kepala BPP dan pegawai dinas. Dengan menekan

tombol *process* pada halaman detail *draft* program, kemudian mencentang kelayakan yang sudah sesuai, memasukan penjelasan kelayakan, dan menekan tombol simpan untuk menyimpan data kedalam *database*.

11. Use Case Lihat Jadwal dan Laporan penyuluhan.

Tabel 4.13 Use Case Lihat Jadwal dan Laporan

Skenario <i>Use Case</i> untuk Lihat jadwal dan Laporan Penyuluhan (SIPP_F_011)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas dapat melihat jadwal kegiatan penyuluhan dan laporan kegiatan penyuluhan.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas udah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol jadwal dan laporan pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman jadwal dan laporan. 3. Sistem akan menampilkan jadwal dan laporan dalam bentuk tabel.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat jadwal dan laporan kegiatan penyuluhan.

Tabel 4.13 merupakan *use case* untuk lihat jadwal dan laporan kegiatan. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, dan pegawai dinas. Fungsionalitas ini untuk menampilkan jadwal kegiatan dan laporan. Dengan memilih menu jadwal dan laporan pada *sidebar* sistem akan menuju halaman jadwal dan laporan yang akan di tampilkan dalam bentuk tabel.

12. Use Case Buat Laporan Penyuluhan

Tabel 4.14 Use Case Membuat Laporan Penyuluhan

Skenario <i>Use Case</i> untuk Lihat Buat Laporan Penyuluhan (SIPP_F_012)	
<i>Objective</i>	Penyuluh dapat membuat laporan penyuluhan.
<i>Actor</i>	Penyuluh.
<i>Pre-condition</i>	Penyuluh sudah melakukan <i>login</i> .

Skenario <i>Use Case</i> untuk Lihat Buat Laporan Penyuluhan (SIPP_F_012)	
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol jadwal dan laporan penyuluhan kemudian memilih menu buat laporan penyuluhan pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman buat laporan penyuluhan. 3. Sistem menyediakan daftar kegiatan yang akan di buat laporan dan tombol pilih untuk memilih kegiatan yang akan di buat laporan. 4. Sistem menyediakan <i>form</i> konten penyuluhan, dana yang terpakai, berkas laporan, foto presensi, foto-foto kegiatan, dan foto bukti pengeluaran dana. 5. Aktor memasukan data pada <i>form</i> yang tersedia dan menekan tombol simpan. Sistem akan menyimpan data ke dalam <i>database</i> dan kembali ke halaman sebelumnya dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol pilih pada kegiatan yang mempunyai status selain belum dikirim, maka sistem akan menampilkan pesan peringatan bahwa hanya laporan yang berstatus belum dikirim yang hanya dapat dibuat laporan. 2. Jika salah satu <i>form</i> kosong maka sistem akan menampilkan peringatan bahwa <i>form</i> tidak boleh kosong.
<i>Post-condition</i>	Aktor berhasil membuat laporan penyuluhan.

Tabel 4.14 merupakan *use case* untuk membuat laporan penyuluhan. Fungsionalitas ini untuk aktor penyuluh. Fungsionalitas ini untuk membuat laporan penyuluhan. Langkah awal yaitu dengan memilih jadwal kegiatan mana yang akan di buat laporan, setelah di pilih sistem akan menuju halaman buat laporan penyuluhan, sistem dapat memasukan data pada *form* yang di sediakan dan menekan tombol simpan, maka sistem akan menyimpan data laporan penyuluhan. Jika salah satu form kosong maka sistem akan menampilkan pesan *error* pada masing-masing *form*.

13. Use Case Lihat Detail Laporan Penyuluhan

Tabel 4.15 Use Case Lihat Detail Laporan penyuluhan

Skenario Use Case untuk Lihat Detail Laporan Penyuluhan (SIPP_F_013)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas dapat melihat detail laporan kegiatan penyuluhan.
<i>Actor</i>	Penyuluh, Kepala BPP, Pegawai Dinas.
<i>Pre-condition</i>	Penyuluh, kepala BPP, pegawai dinas sudah melakukan <i>login</i> dan berada pada halaman jadwal dan laporan penyuluhan.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat pada halaman jadwal dan laporan penyuluhan. 2. Sistem akan <i>direct</i> ke halaman detail laporan penyuluhan. 3. Sistem menampilkan data laporan berupa tema, status, dana terpakai, tanggal laporan di buat, <i>content</i> laporan penyuluhan, berkas laporan yang bisa di download, foto presensi, daftar foto kegiatan.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat detail laporan penyuluhan.

Tabel 4.15 merupakan *use case* untuk lihat detail laporan penyuluhan. Fungsionalitas ini untuk aktor penyuluh, kepala BPP, dan pegawai dinas. Fungsionalitas ini untuk data-data laporan penyuluhan. Dengan cara menekan tombol lihat pada halaman daftar laporan penyuluhan sistem akan menuju halaman detail laporan penyuluhan dan sistem menampilkan data-data laporan penyuluhan.

14. Use Case Verifikasi Laporan Penyuluhan

Tabel 4.16 Use Case Verifikasi Laporan Penyuluhan

Skenario Use Case untuk Verifikasi Laporan Penyuluhan (SIPP_F_014)	
<i>Objective</i>	Kepala BPP dan pegawai dinas dapat memverifikasi laporan penyuluhan.
<i>Actor</i>	Kepala BPP, Pegawai Dinas.

Skenario <i>Use Case</i> untuk Verifikasi Laporan Penyuluhan (SIPP_F_014)	
<i>Pre-condition</i>	kepala BPP, pegawai dinas sudah melakukan <i>login</i> dan berada pada halaman detail laporan penyuluhan.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>process</i> pada halaman detail laporan penyuluhan. 2. Sistem akan <i>direct</i> ke halaman verifikasi laporan penyuluhan. 3. Sistem menyediakan centang pada kelayakan laporan penyuluhan, <i>form</i> penjelasan untuk memberi penjelasan terkait kelayakan laporan penyuluhan, dan tombol simpan. 4. Aktor menekan tombol simpan, sistem menyimpan data verifikasi ke dalam <i>database</i> dan kembali ke halaman detail laporan penyuluhan dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	Jika salah satu <i>form</i> penjelasan kelayakan di kosongkan maka sistem akan menampilkan pesan " <i>form</i> tidak boleh kosong" pada masing-masing <i>form</i> yang kosong.
<i>Post-condition</i>	Aktor berhasil memverifikasi laporan penyuluhan.

Tabel 4.16 merupakan *use case* untuk memverifikasi laporan penyuluhan. Fungsionalitas ini untuk aktor kepala BPP dan pegawai dinas. Langkah awal yaitu aktor memilih tombol proses yang berada pada halaman detail laporan, sistem akan menampilkan daftar kelayakan, sistem menyediakan centang dan *form* untuk mengisi penjelasan. Aktor dapat menekan tombol simpan untuk menyimpan data verifikasi kedalam *database*. Jika salah satu *form* penjelasan kosong maka sistem akan menampilkan pesan "*form* tidak boleh kosong" pada masing-masing *form* yang kosong.

15. *Use Case* Ajukan Surat Tugas

Tabel 4.17 *Use Case* Ajukan Surat Tugas

Skenario <i>Use Case</i> untuk Ajukan Surat Tugas (SIPP_F_015)	
<i>Objective</i>	Pegawai dinas dapat mengajukan surat tugas.
<i>Actor</i>	Pegawai dinas.

Skenario <i>Use Case</i> untuk Ajukan Surat Tugas (SIPP_F_015)	
<i>Pre-condition</i>	Pegawai dinas sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan surat tugas dan memilih ajukan surat tugas pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman ajukan surat tugas 3. Sistem akan menampilkan daftar <i>draft</i> programa yang berstatus sudah di setuju pegawai dinas dan tombol pilih untuk memilih draft programa yang akan di buat surat tugas. 4. Sistem menyediakan tombol ajukan untuk mengajukan surat tugas, ketika di tekan maka akan menampilkan <i>popup</i> konfirmasi pegajuan surat tugas.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil membuat surat tugas.

Tabel 4.16 merupakan *use case* untuk membuat surat tugas oleh aktor pegawai dinas. Aktor dapat melakukan dengan menekan tombol ajukan pada halaman ajukan surat tugas, ketika di tekan maka sistem akan memunculkan *popup* konfirmasi pengajuan surat tugas, terdapat pilihan ya untuk mengajukan surat tugas dan tidak untuk menutup *popup*.

16. *Use Case* Lihat Daftar Surat Tugas

Tabel 4.18 *Use Case* Lihat Daftar Surat Tugas

Skenario <i>Use Case</i> untuk Lihat Daftar Surat Tugas (SIPP_F_016)	
<i>Objective</i>	Pegawai dinas dan kepala dinas dapat melihat daftar surat tugas.
<i>Actor</i>	Pegawai dinas, kepala dinas.
<i>Pre-condition</i>	Pegawai dinas dan kepala dinas sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan menu surat tugas pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman daftar surat tugas.



Skenario <i>Use Case</i> untuk Lihat Daftar Surat Tugas (SIPP_F_016)	
	3. Sistem akan menampilkan daftar surat tugas dalam bentuk Tabel.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat daftar surat tugas.

Tabel 4.18 merupakan *use case* untuk melihat daftar surat tugas oleh aktor pegawai dinas dan kepala dinas. Langkah pertama yaitu dengan cara menekan daftar surat tugas pada *sidebar*, sistem akan menuju halaman daftar surat tugas dan akan menampilkan daftar surat tugas dalam bentuk tabel.

17. *Use Case* Verifikasi Surat Tugas

Tabel 4.19 *Use Case* Verifikasi Surat Tugas

Skenario <i>Use Case</i> untuk Verifikasi Surat Tugas (SIPP_F_017)	
<i>Objective</i>	Kepala dinas dapat memverifikasi surat tugas.
<i>Actor</i>	Kepala dinas.
<i>Pre-condition</i>	Kepala dinas sudah melakukan <i>login</i> dan berada pada halaman detail surat tugas.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>process</i> pada halaman detail surat tugas. 2. Sistem akan <i>direct</i> ke halaman verifikasi surat tugas. 3. Sistem menyediakan centang pada kelayakan surat tugas, <i>form</i> penjelasan untuk memberi penjelasan terkait kelayakan surat tugas, dan tombol simpan. 4. Aktor menekan tombol simpan, sistem menyimpan data verifikasi ke dalam <i>database</i> dan kembali ke halaman detail surat tugas dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	Jika salah satu <i>form</i> penjelasan kosong maka sistem akan menampilkan pesan " <i>form</i> tidak boleh kosong" pada masing-masing <i>form</i> yang kosong.
<i>Post-condition</i>	Aktor berhasil memverifikasi surat tugas.

Tabel 4.19 merupakan *use case* untuk melakukan verifikasi surat tugas oleh kepala dinas. Langkah awal yaitu dengan menekan tombol *process* pada halaman detail surat tugas, sistem akan menyediakan beberapa *form* untuk mengisi penjelasan kelayakan dan centang kelayakan, aktor dapat mencentang kelayakan dan mengisi *form* untuk penjelasan kelayakan kemudian menekan tombol simpan untuk menyimpan data verifikasi surat tugas. jika salah satu *form* ada yang kosong maka sistem akan menampilkan pesan "*form* tidak boleh kosong" pada masing-masing *form* yang kosong.

18. Use Case Terbitkan Surat Tugas

Tabel 4.20 Use Case Menerbitkan Surat Tugas

Skenario Use Case untuk Terbitkan Surat Tugas (SIPP_F_018)	
<i>Objective</i>	Pegawai dinas dapat menerbitkan surat tugas
<i>Actor</i>	Pegawai dinas.
<i>Pre-condition</i>	Pegawai dinas sudah melakukan <i>login</i> dan berada pada halaman detail surat tugas dengan syarat status surat tugas yaitu <i>valid</i> oleh kepala dinas.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol terbitkan pada halaman detail surat tugas. 2. Sistem akan menampilkan <i>popup</i> untuk konfirmasi penerbitan, dan sistem menyediakan tombol ya untuk menerbitkan dan tombol tidak untuk menutup <i>popup</i>. 3. Sistem memperbarui data pada <i>database</i> dan kembali ke halaman sebelumnya dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil menerbitkan surat tugas.

Tabel 4.20 merupakan *use case* untuk menerbitkan surat tugas oleh pegawai dinas, yaitu dengan cara menekan tombol terbitkan pada halaman detail surat tugas dengan syarat surat tugas sudah di validasi oleh kepala dinas.

19. Use Case Ajukan Proposal Dana

Tabel 4.21 Use Case Ajukan Proposal Dana

Skenario Use Case untuk Ajukan Proposal Dana (SIPP_F_019)	
<i>Objective</i>	Pegawai dinas dapat mengajukan proposal dana.
<i>Actor</i>	Pegawai dinas.
<i>Pre-condition</i>	Pegawai dinas sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan proposal dana dan memilih menu buat proposal dana pada <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman buat proposal dana. 3. Sistem menampilkan semua daftar <i>draft</i> programa yang berstatus surat tugas sudah di terbitkan, dan sistem menyediakan tombol ajukan untuk mengajukan proposal dana, ketikan tombol di tekan maka akan menampilkan <i>popup</i> konfirmasi pengajuan proposal dana, terdapat dua pilihan, ya untuk mengajukan proposal dana dan tidak untuk menutup <i>popup</i>.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil membuat proposal dana.

Tabel 4.21 merupakan *use case* untuk membuat proposal dana oleh pegawai dinas. Aktor dapat menekan tombol ajukan untuk mengajukan proposal dana kemudian sistem akan menampilkan *popup* untuk mengkonfirmasi pengajuan proposal dana, terdapat dua pilihan tombol, *yes* untuk melanjutkan pengajuan proposal dana dan tidak untuk menutup *popup*.

20. Use Case Lihat Daftar Proposal Dana

Tabel 4.22 Use Case Lihat Daftar Proposal Dana

Skenario Use Case untuk Lihat Daftar Proposal Dana (SIPP_F_020)	
<i>Objective</i>	Pegawai dinas dan kepala dinas dapat melihat daftar proposal dana.
<i>Actor</i>	Pegawai dians, kepala dinas.

Skenario <i>Use Case</i> untuk Lihat Daftar Proposal Dana (SIPP_F_020)	
<i>Pre-condition</i>	Pegawai dinas dan kepala dinas sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan proposal dana pada menu <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman daftar proposal dana. 3. Sistem menampilkan daftar proposal dana dalam bentuk Tabel.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat daftar proposal dana.

Tabel 4.22 merupakan *use case* untuk melihat daftar proposal dana oleh kepala dinas dan pegawai dinas. Langkah pertama yaitu memilih menu daftar proposal dana pada menu *sidebar*, sistem menuju halaman daftar proposal dana, sistem akan menampilkan daftar proposal dana dalam bentuk tabel.

21. *Use Case* Verifikasi Proposal Dana

Tabel 4.23 *Use Case* Verifikas Proposal Dana

Skenario <i>Use Case</i> untuk Verifikasi Proposal Dana (SIPP_F_021)	
<i>Objective</i>	Kepala dinas dapat memverifikasi proposal dana.
<i>Actor</i>	Kepala dinas.
<i>Pre-condition</i>	Kepala dinas sudah melakukan <i>login</i> dan berada pada halaman daftar proposal dana.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Sistem menyediakan tombol setuju dan tolak. 2. Tombol setuju untuk menyetujui proposal dana, ketika muncul maka akan <i>popup</i> konfirmasi setuju proposal dana, dan tombol tolak untuk menolak proposal dana.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil memverifikasi proposal dana.



Tabel 4.23 merupakan *use case* untuk memverifikasi proposal dana oleh kepala dinas. Langkah pertama yaitu menekan tombol *process* pada halaman detail proposal dana, sistem akan menuju halaman *process* proposal dana, sistem akan menampilkan centang kelayakan dan *form* penjelasan kelayakan, aktor dapat mengisi *form* dan menekan tombol simpan untuk menyimpan data verifikasi proposal dana.

22. Use Case Lihat Daftar Pengiriman Dana

Tabel 4.24 Use Case Lihat Daftar Pengiriman Dana

Skenario <i>Use Case</i> untuk Lihat Daftar Pengiriman Dana (SIPP_F_022)	
<i>Objective</i>	Penyuluh, pegawai dinas, dan kepala dinas dapat melihat daftar pengiriman dana.
<i>Actor</i>	Penyuluh, Pegawai Dinas, Kepala Dinas.
<i>Pre-condition</i>	Penyuluh, pegawai dinas, dan kepala dinas sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan menu pengiriman dana pada menu <i>sidebar</i>. 2. Sistem akan <i>direct</i> ke halaman daftar pengiriman dana. 3. Sistem menampilkan daftar pengiriman dana dalam bentuk Tabel.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat daftar pengiriman dana.

Tabel 4.24 merupakan *use case* untuk melihat daftar pengiriman dana oleh penyuluh dan kepala dinas. Langkah awal yaitu menekan tombol pengiriman dana pada *sidebar*, sistem akan menuju halaman daftar pengiriman dana, sistem akan menampilkan daftar pengiriman dana dalam bentuk tabel.

23. Use Case Lihat Detail Pengiriman Dana

Tabel 4.25 Use Case Lihat Detail Pengiriman Dana

Skenario <i>Use Case</i> untuk Lihat Detail Pengiriman Dana (SIPP_F_023)	
<i>Objective</i>	Penyuluh dan pegawai dinas dapat melihat detail pengiriman dana.
<i>Actor</i>	Penyuluh, pegawai dinas.

Skenario <i>Use Case</i> untuk Lihat Detail Pengiriman Dana (SIPP_F_023)	
<i>Pre-condition</i>	Penyuluh dan pegawai dinas sudah melakukan <i>login</i> dan berada pada halaman daftar pengiriman dana.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat pada halaman daftar pengiriman dana. 2. Sistem akan <i>direct</i> ke halaman detail pengiriman dana. 3. sistem menampilkan data pengiriman dana berupa tanggal pengiriman, jumlah pengiriman, nama pengirim, bank pengirim, nomor rekening pengirim, status penerimaan, dan foto bukti pengiriman.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat detail pengiriman dana.

Tabel 4.25 merupakan *use case* untuk melihat detail pengiriman dana oleh pegawai dinas dan penyuluh. Langkah pertama yaitu dengan menekan tombol lihat pada halaman daftar pengiriman dana, sistem akan menuju halaman detail pengiriman dana, dan sistem akan menampilkan data-data pengiriman dana.

24. *Use Case* Konfirmasi Penerimaan Dana

Tabel 4.26 *Use Case* Lihat Konfirmasi Penerimaan Dana

Skenario <i>Use Case</i> untuk konfirmasi penerimaan dana (SIPP_F_024)	
<i>Objective</i>	Penyuluh dapat mengkonfirmasi penerimaan dana.
<i>Actor</i>	Penyuluh.
<i>Pre-condition</i>	Penyuluh sudah melakukan <i>login</i> dan berada pada halaman detail pengiriman dana dengan status pengiriman dana belum di terima.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol konfirmasi penerimaan dana pada halaman detail pengiriman dana. 2. Sistem akan menampilkan <i>popup</i> konfirmasi penerimaan dana, tombol ya untuk

Skenario <i>Use Case</i> untuk konfirmasi penerimaan dana (SIPP_F_024)	
	mengkonfirmasi penerimaan dana, dan tombol tidak untuk menutup <i>popup</i> . 3. Sistem memperbarui data pada <i>database</i> dan kembali pada halaman sebelumnya dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil mengkonfirmasi penerimaan dana.

Tabel 4.26 merupakan *use case* untuk mengkonfirmasi penerimaan dana oleh penyuluh. Pada halaman detail pengiriman dana akan di sediakan tombol konfirmasi jika status pengiriman dana “dana dikirim”, ketika tombol konfirmasi di tekan maka akan menampilkan *popup* untuk konfirmasi penerimaan dana, tombol ya untuk konfirmasi, dan tombol tidak untuk menutup *popup*.

25. *Use Case* Lihat Daftar Pengguna

Tabel 4.27 *Use Case* Lihat Daftar Pengguna

Skenario <i>Use Case</i> untuk Lihat Daftar Pengguna (SIPP_F_025)	
<i>Objective</i>	Admin sistem dapat melihat daftar pengguna.
<i>Actor</i>	Admin Sistem.
<i>Pre-condition</i>	Admin sistem sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	1. Aktor melakukan <i>login</i> . 2. Sistem akan <i>direct</i> ke halaman daftar pengguna. 3. Sistem menampilkan daftar pengguna dalam bentuk Tabel.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat daftar pengguna.

Tabel 4.27 merupakan *use case* untuk menampilkan daftar pengguna oleh admin. Admin harus melakukan *login* terlebih dahulu, sistem akan menampilkan daftar pengguna dalam bentuk tabel.

26. Use Case Tambah Pengguna

Tabel 4.28 Use Case Tambah Pengguna

Skenario Use Case untuk Tambah Pengguna (SIPP_F_026)	
<i>Objective</i>	Admin sistem dapat menambahkan pengguna.
<i>Actor</i>	Admin sistem.
<i>Pre-condition</i>	Admin sistem sudah melakukan <i>login</i> dan berada pada halaman daftar pengguna.
<i>Main Success Scenario</i>	<ol style="list-style-type: none">1. Aktor menekan tambah pengguna pada halaman daftar pengguna.2. Sistem akan <i>direct</i> ke halaman tambah pengguna.3. Sistem menyediakan <i>form</i> nama, <i>username</i>, <i>password</i>, <i>nip</i>, <i>email</i>, desa, kecamatan, jabatan, foto, dan tombol simpan.4. Aktor memasukkan data pada form dan menekan tombol simpan, sistem akan menyimpan data kedalam <i>database</i> dan kembali ke halaman sebelumnya dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	Jika salah satu form kosong maka sistem akan menampilkan pesan " <i>form</i> tidak boleh kosong" pada masing-masing <i>form</i> yang kosong.
<i>Post-condition</i>	Aktor berhasil menambah pengguna.

Tabel 4.28 merupakan *use case* untuk menambah pengguna, sistem menyediakan tombol tambah pengguna pada halaman daftar pengguna, aktor menekan tombol tambah pengguna dan sistem menuju halaman tambah pengguna, sistem menampilkan *form-form* untuk aktor memasukkan data-data pengguna, admin dapat menekan tombol simpan untuk melakukan penyimpanan data kedalam *database* dan sistem akan menampilkan pesan "berhasil menyimpan data".

27. Use Case Lihat Data Pengguna

Tabel 4.29 Use Case Lihat Data pengguna

Skenario Use Case untuk Lihat Data Pengguna (SIPP_F_027)	
<i>Objective</i>	Admin sistem dapat melihat data pengguna.
<i>Actor</i>	Admin sistem.
<i>Pre-condition</i>	Admin sudah melakukan login dan berada pada halaman daftar pengguna.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol lihat pada halaman daftar pengguna. 2. Sistem akan <i>direct</i> ke halaman data pengguna 3. Sistem menampilkan data pengguna berupa nama, <i>username</i>, <i>nip</i>, <i>email</i>, foto, dan jabatan.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melihat data pengguna.

Tabel 4.29 merupakan *use case* untuk melihat data pengguna, sistem menyediakan tombol lihat pada halaman daftar pengguna, aktor dapat menekan tombol lihat dan sistem akan menuju halaman data pengguna dan sistem akan menampilkan data-data pengguna.

28. Use Case Edit Pengguna

Tabel 4.30 Use Case Edit Pengguna

Skenario Use Case untuk Edit Pengguna (SIPP_F_028)	
<i>Objective</i>	Admin sistem dapat mengedit data pengguna.
<i>Actor</i>	Admin sistem.
<i>Pre-condition</i>	Admin sistem sudah melakukan <i>login</i> dan berada pada halaman data pengguna.
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol edit pada halaman data pengguna. 2. Sistem akan <i>direct</i> ke halaman edit pengguna. 3. Sistem menyediakan form nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, foto, jabatan, dan tombol simpan. 4. Aktor memasukan data pada form dan menekan tombol simpan, Sistem menyimpan

Skenario <i>Use Case</i> untuk Edit Pengguna (SIPP_F_028)	
	data kedalam <i>database</i> dan kembali pada halaman data pengguna dengan menampilkan pesan berhasil menyimpan data.
<i>Alternatif Flow</i>	Jika salah satu <i>form</i> kosong maka sistem akan menampilkan pesan <i>form</i> tidak boleh kosong.
<i>Post-condition</i>	Aktor berhasil mengedit data pengguna.

Tabel 4.30 merupakan *use case* untuk admin dapat melakukan *edit* data pengguna. *Form* edit akan muncul ketika menekan tombol ubah data pada menu berbentuk *tab*. Sistem akan menampilkan *form* untuk aktor memasukkan data, dan tombol simpan untuk menyimpan data kedalam *database*.

29. *Use Case* Hapus Pengguna

Tabel 4.31 *Use Case* Hapus Data Pengguna

Skenario <i>Use Case</i> untuk Hapus Pengguna (SIPP_F_029)	
<i>Objective</i>	Admin sistem dapat menghapus data pengguna.
<i>Actor</i>	Admin sistem.
<i>Pre-condition</i>	Admin sistem sudah melakukan <i>login</i> dan berada pada halaman data pengguna
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol hapus pada halaman data pengguna. 2. Sistem akan menampilkan <i>popup</i> konfirmasi penghapusan data pengguna, tombol ya untuk menghapus dan tombol tidak untuk menutup <i>popup</i>. 3. Sistem akan menghapus data pengguna dalam <i>database</i> dan kembali ke halaman daftar pengguna dengan menampilkan pesan berhasil menghapus data.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil menghapus data pengguna.

Tabel 4.31 merupakan *use case* untuk menghapus pengguna. Sistem menyediakan tombol hapus pada halaman data pengguna. Aktor dapat menekan tombol hapus dan sistem akan menampilkan *popup* konfirmasi penghapusan data pengguna. Dan pilihan tombol ya untuk melanjutkan penghapusan dan tombol tidak untuk menutup *popup*.

30. Use Case Logout

Tabel 4.32 Use Case Logout

Skenario Use Case untuk Logout (SIPP_F_030)	
<i>Objective</i>	Penyuluh, kepala BPP, pegawai dinas, kepala dinas dan admin sistem dapat keluar dari halaman aktor.
<i>Actor</i>	Penyuluh, kepala BPP, pegawai dinas, kepala dinas dan admin sistem.
<i>Pre-condition</i>	Admin sistem sudah melakukan <i>login</i> .
<i>Main Success Scenario</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>logout</i> pada menu <i>navbar</i>. 2. Sistem akan keluar dari halaman aktor menuju halaman <i>login</i>.
<i>Alternatif Flow</i>	-
<i>Post-condition</i>	Aktor berhasil melakukan <i>logout</i> .

Tabel 4.32 merupakan *use case* untuk melakukan *logout*, pada halaman hak akses penyuluh, kepala BPP, pegawai dinas, kepala dinas, dan admin sistem menyediakan tombol *logout*. Aktor dapat menekan tombol *logout* yang berbentuk *dropdown*. Ketika tombol ditekan maka sistem akan mengeluarkan aktor dari halaman pengguna dan menuju halaman *login*.

4.4.6 Analisis kebutuhan Non Fungsional

Analisis kebutuhan *non* fungsional menjabarkan aturan yang berkaitan dengan kualitas sistem yang sedang dibangun dan dapat menentukan kepuasan bagi pengguna. Parameter yang dijadikan kebutuhan *non* fungsional adalah *compatibility testing* yang dijelaskan pada Tabel 4.33.

Tabel 4.33 Kebutuhan Non Fungsional

No	Kode Kebutuhan	Parameter	Deskripsi
1	SIPP_NF_001	Compatibility	Fitur dan komponen dalam sistem dapat diakses secara sempurna pada

			delapan browser yaitu browser internet explorer(IE), Edge, Firefox, Safari, Opera, Chrome, iOS, dan Android yang diujikan dengan menggunakan aplikasi sortsite.
--	--	--	---



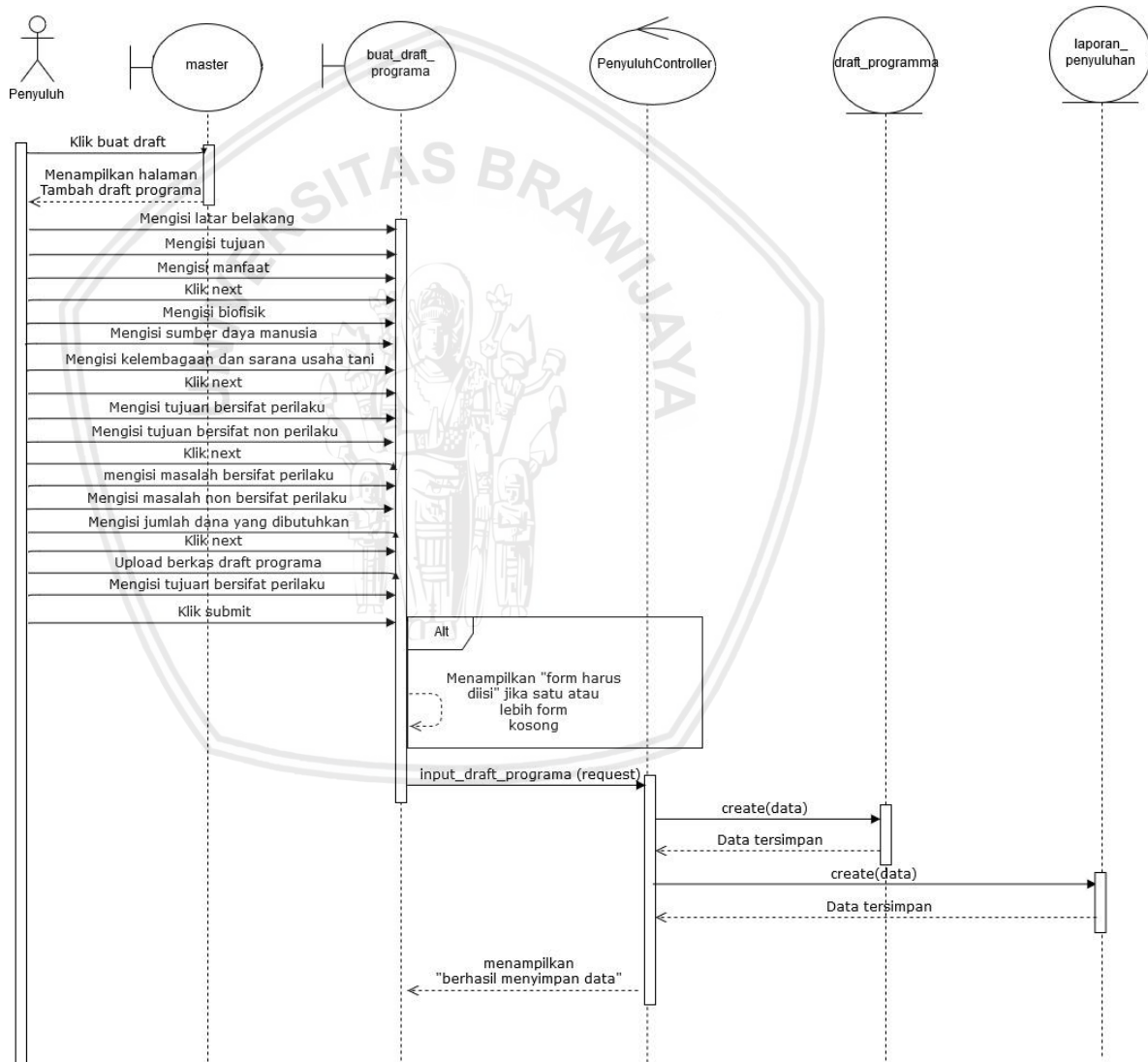
BAB 5 PERANCANGAN & IMPLEMENTASI

5.1 Perancangan Perangkat Lunak

Dalam tahapan perancangan, beberapa hal yang akan digunakan sebagai acuan dalam proses implementasi, antara lain pembuatan diagram perancangan perangkat lunak yang meliputi perancangan *sequence diagram*, perancangan *class diagram*, dan juga perancangan antarmuka.

5.1.1 Perancangan *Sequence Diagram*

1. *Sequence Diagram* Tambah Draft Program

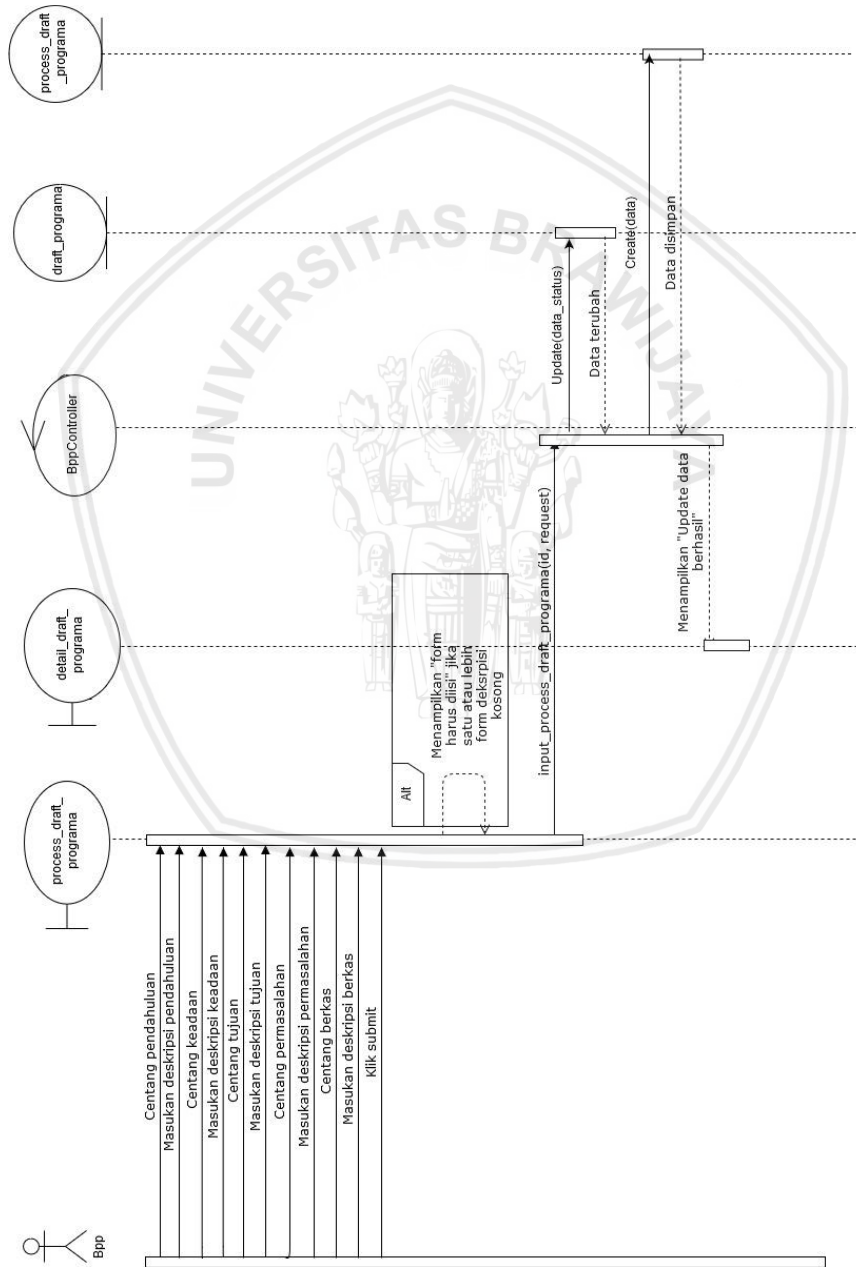


Gambar 5.1 *Sequence Diagram* Tambah Draft Program

Gambar 5.1 merupakan *sequence diagram* untuk melakukan tambah *draft* program oleh penyaluh. Langkah yang harus lakukan aktor yang pertama yaitu

dengan menekan tombol buat *draft* programa pada *sidebar*. Kemudian sistem akan menampilkan beberapa *form* untuk memasukan data secara bertahap. Selanjutnya menekan tombol *submit*, dan sistem akan mengirim data yang di masukan ke fungsi *input_draft_programa()* yang berada pada *Penyuluh Controller*. Kemudian *controller* menyimpan data ke dalam *database draft_programa*, ketika berhasil disimpan maka kembali lagi ke halaman *buat_draft_programa* menampilkan pesan sukses “berhasil menyimpan data”.

2. Sequence Diagram Verifikasi Draft Program

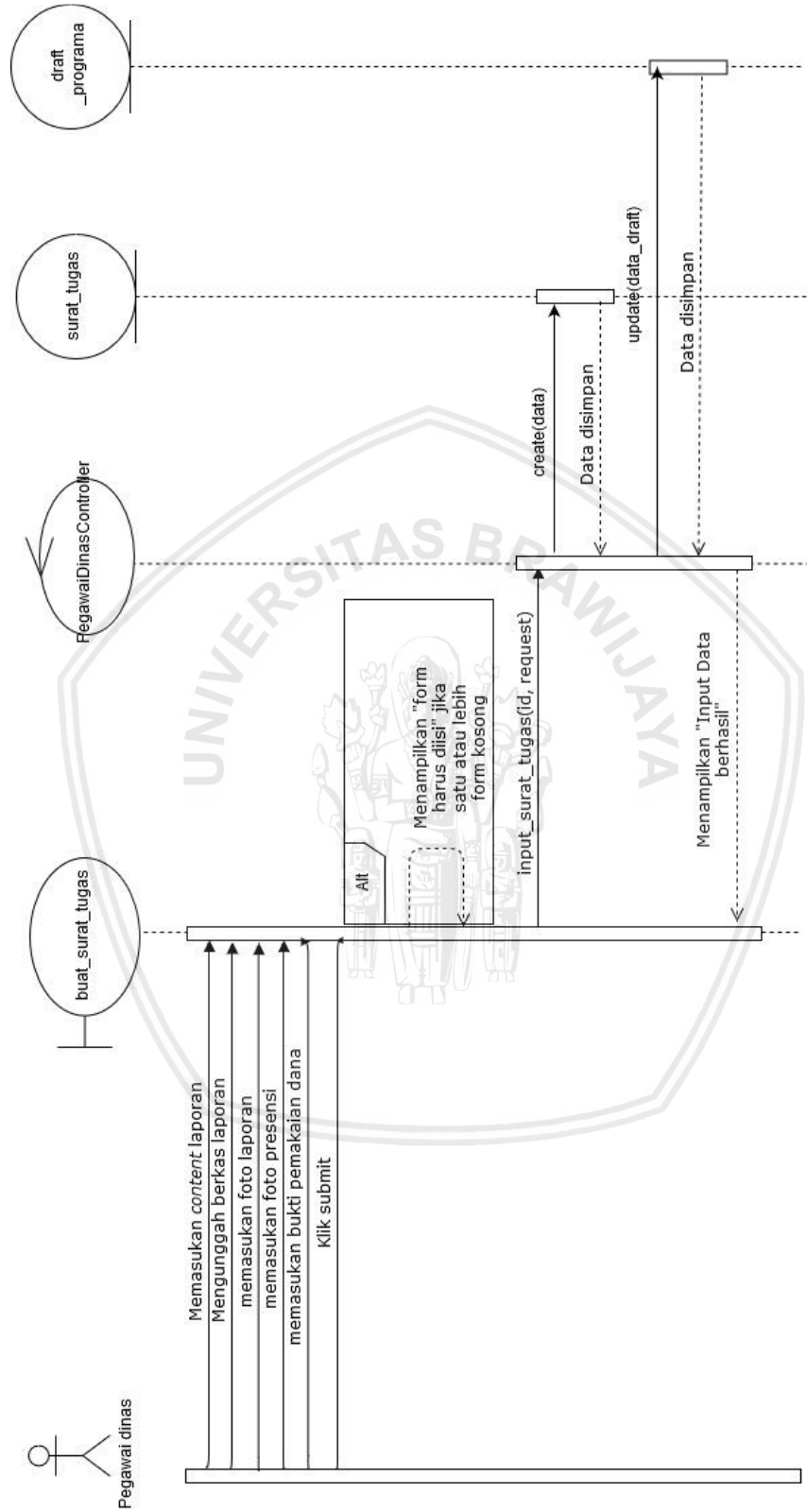


Gambar 5.2 Sequence Diagram Verifikasi Draft Program

Gambar 5.2 menjelaskan *sequence diagram* untuk verifikasi *draft* program. Hal yang pertama dilakukan adalah sistem menyediakan centang kelayakan dan *form* penjelasan kelayakan untuk aktor mencentang kelayakan yang sudah memenuhi syarat dan penjelasan untuk memberi penjelasan pada kelayakan *draft*, kemudian aktor dapat menekan tombol *submit* maka sistem akan mengirim data yang di masukan ke dalam fungsi *input_process_draft_programa* pada class *BppController*. Kemudian *controller* akan menyimpan data ke dalam *database*, terjadi dua kali penyimpanan yaitu yang pertama memperbarui data *draft* program pada *database draft* program, setelah berhasil disimpan akan di lanjutkan penyimpanan data *process draft* program kedalam *database process_draft_programa* untuk menyimpan data-data kelayakan *draft* program. Ketika berhasil menyimpan maka kembali ke halaman detail *draft* program dan menampilkan pesan “berhasil menyimpan data”.



3. Sequence Diagram Buat Laporan Penyuluhan

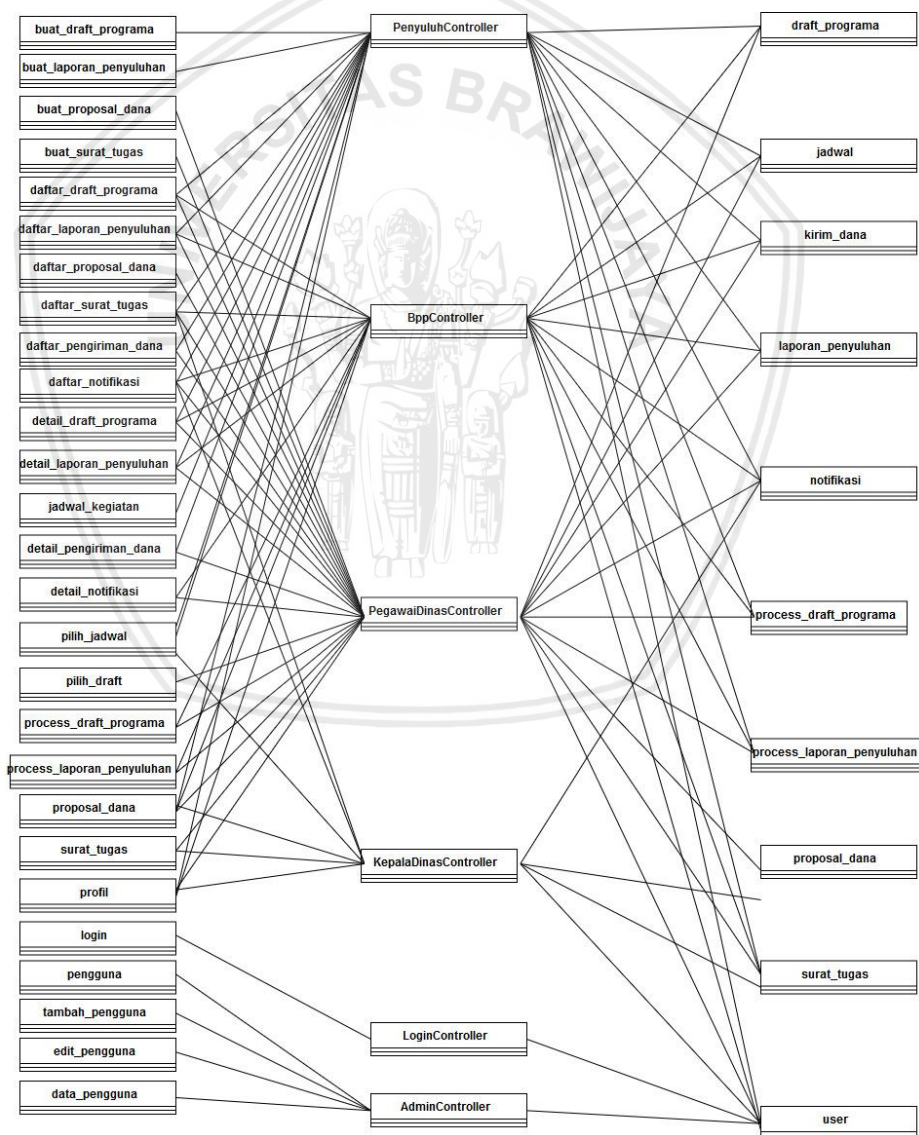


Gambar 5.3 Sequence Diagram Buat Laporan Penyuluhan

Gambar 5.3 adalah *sequence diagram* untuk membuat laporan penyuluhan. Sistem menyediakan *form* untuk aktor dapat memasukan data dan tombol simpan yang ketika di tekan akan menuju fungsi *input_laporan_penyuluhan()* pada *controller PegawaiController*. Kemudian sistem akan menyimpan data kedalam *database* laporan penyuluhan. Dan sistem akan kembali ke halaman buat laporan penyuluhan dengan menampilkan pesan “berhasil menyimpan data”.

5.1.2 Perancangan *Class Diagram*

Pada perancangan *class diagram* terdapat tiga kelas yaitu kelas *model*, *view*, dan *controller*. Setiap kelas mempunyai relasi dengan kelas lainnya, yaitu relasi asosiasi, relasi ini menjelaskan bahwa kelas saling menggunakan dengan kelas lainnya. Relasi dan kelas akan di jabarkan pada Gambar 5.4.



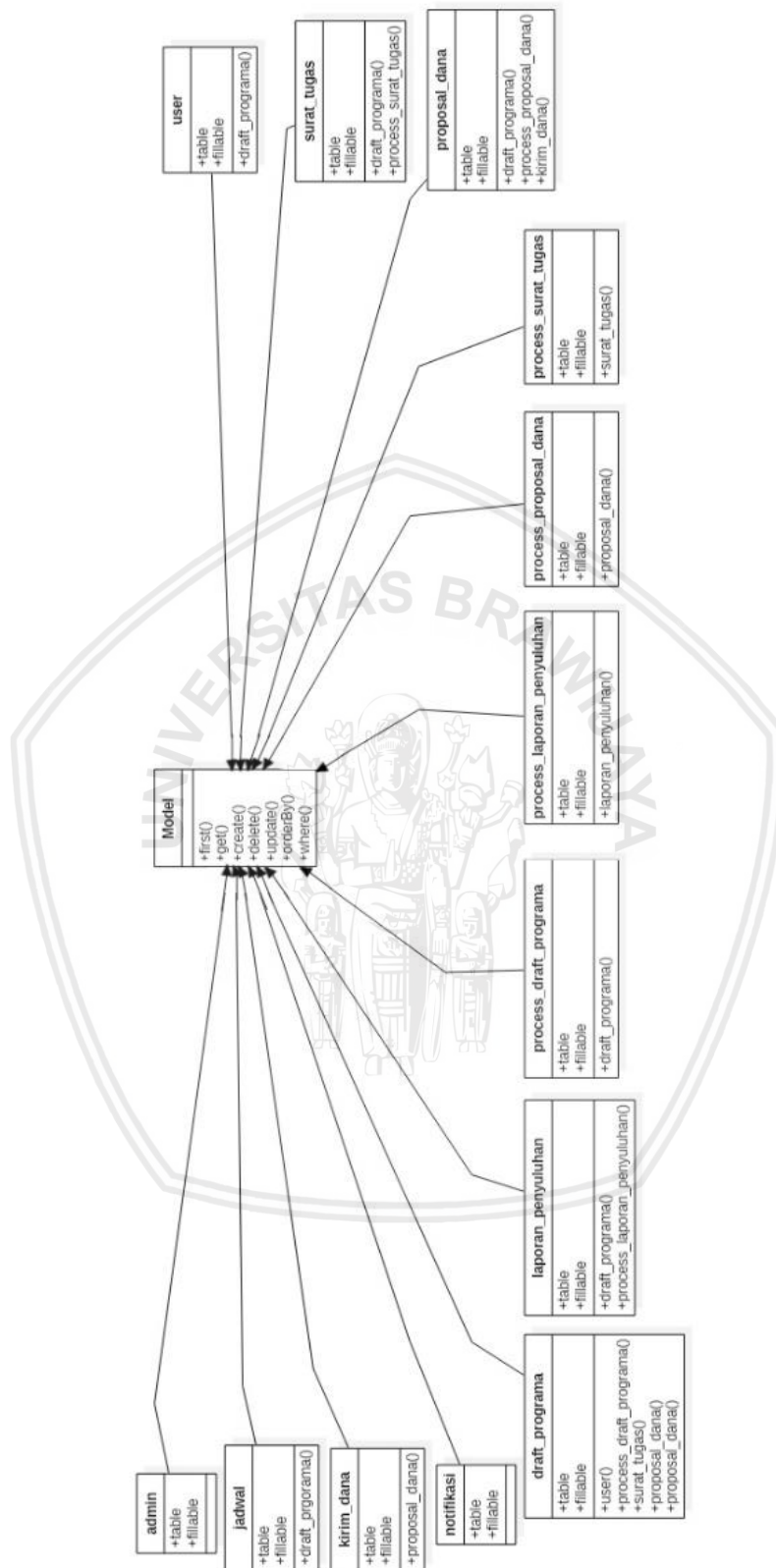
Gambar 5.4 *Class Diagram*



5.1.2.2 Detail Class Diagram Model

Pada sistem terdapat tiga belas *model*, yaitu *admin*, *jadwal*, *kirim_dana*, *notifikasi*, *draft_programa*, *laporan_penyaluhan*, *process_draft_programa*, *process_laporan_penyaluhan*, *process_proposal_dana*, *process_surat_tugas*, *user*, *surat_tugas*, *proposal_dana*. Setiap *model* memiliki relasi generalisasi dengan kelas *model* yang di sediakan oleh *Laravel*. Setiap *model* memiliki *attribute* dan *fungsi*. Pada Gambar 5.6 akan dijelaskan secara jelas.

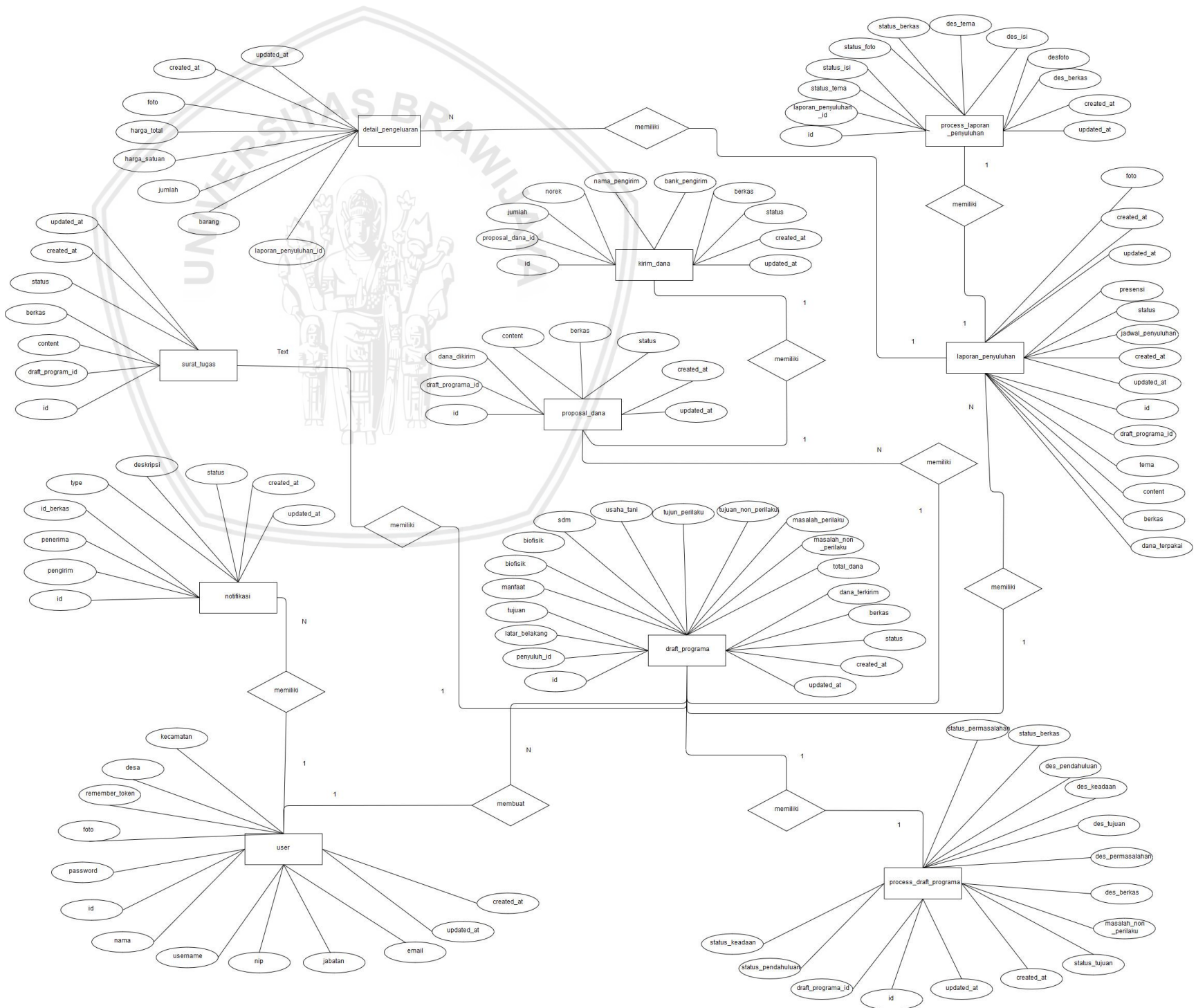




Gambar 5.6 Class Diagram Model

5.1.3 Perancangan Basis Data

Perancangan basis data sistem ini menggunakan *Entity Relationship Diagram* atau disingkat ERD. Sistem ini memiliki 9 entitas dan 9 relasi yang dapat dilihat pada Gambar 5.7. Entitas *user* sebagai penyimpan data pengguna sistem, Entitas *draft* program berisi data-data *draft* program yang nanti akan di masukan oleh pengguna di dalam sistem. Entitas *process* draft program berisi data-data verifikasi dari *draft* program oleh kepala BPP dan pegawai dinas. Entitas notifikasi berisi data-data notifikasi. Entitas surat tugas menyimpan data-data surat tugas yang memiliki hubungan dengan *draft* program karena untuk membuat surat tugas harus membuat *draft* program terlebih dahulu. Entitas proposal dana menyimpan data-data proposal dana yang berguna untuk mengajukan proposal pengiriman dana. Surat tugas menyimpan data-data surat tugas. Entitas kirim dana menyimpan data-data verifikasi pengiriman dana. Entitas laporan penyuluhan berisi data-data laporan kegiatan penyuluhan beserta jadwal penyuluhan di lakukan. Dan entitas *process* laporan penyuluhan berisi data verifikasi laporan penyuluhan oleh pegawai dinas dan kepala BPP.



Gambar 5.7 Perancangan Entity Relationship Diagram

5.1.4 Perancangan Komponen

Perancangan komponen merupakan perancangan kode program sebagai bentuk implementasi. Dalam tahap perancangan ini untuk membangun sebuah kode program maka dibutuhkan sebuah perancangan dengan menggunakan *pseudocode*. Perancangan komponen dijelaskan dari Tabel 5.1 hingga Tabel 5.3

1. Perancangan *pseudocode* untuk tambah *draft* program

1. Nama kelas : `PenyuluhController.php`
2. Nama operasi : `input_draft_programa()`

Tabel 5.1 Pseudocode tambah draft programa

No.	Pseudocode
1	Fungsi pemeriksaan kondisi apakah ada data yang kosong
2	Memeriksa inputan latar belakang kosong atau tidak
3	Memeriksa inputan tujuan kosong atau tidak
4	Memeriksa inputan manfaat kosong atau tidak
5	Memeriksa inputan biofisik kosong atau tidak
6	Memeriksa inputan sdm kosong atau tidak
7	Memeriksa inputan usaha tani kosong atau tidak
8	Memeriksa inputan tujuan perilaku kosong atau tidak
9	Memeriksa inputan tujuan non perilaku kosong atau tidak
10	Memeriksa inputan masalah perilaku kosong atau tidak
11	Memeriksa inputan masalah non perilaku kosong atau tidak
12	Memeriksa inputan total dana kosong atau tidak
13	Memeriksa inputan berkas kosong atau tidak
14	Deklarasi dan inisialisasi berkas
15	Menyimpan data berkas kedalam database
16	Deklarasi variabel data
17	Menyimpan data <i>request</i> id kedalam variabel
18	Menyimpan data <i>request</i> latar belakang kedalam variabel
19	Menyimpan data <i>request</i> tujuan kedalam variabel
20	Menyimpan data <i>request</i> manfaat kedalam variabel
21	Menyimpan data <i>request</i> biofisik kedalam variabel
22	Menyimpan data <i>request</i> sdm kedalam variabel
23	Menyimpan data <i>request</i> usaha tani kedalam variabel
24	Menyimpan data <i>request</i> tujuan perilaku kedalam variabel
25	Menyimpan data <i>request</i> tujuan non perilaku kedalam variabel
26	Menyimpan data <i>request</i> masalah perilaku kedalam variabel
27	Menyimpan data <i>request</i> masalah non perilaku kedalam variabel
	Menyimpan data <i>request</i> total dana kedalam variabel
28	Menyimpan data <i>request</i> berkas kedalam variabel
29	Menyimpan data <i>request</i> status kedalam variabel
30	Menyimpan data status kedalam variabel
31	Menyimpan data draft kedalam database
32	Deklarasi variabel notif
33	Menyimpan data kedalam variabel pesan
34	Menyimpan data kedalam variabel type
35	Kembali kehalaman sebelumnya
36	

2. Perancangan *pseudocode* untuk verifikasi *draft* program

1. Nama kelas : PegawaiDinasController.php
2. Nama operasi : process_draft_programa()

Tabel 5.2 Pseudocode Verifikasi Draft Program

No.	Pseudocode
1	Fungsi untuk memeriksa kondisi apakah ada data yang kosong
2	Memeriksa masukan deskripsi pendahuluan kosong atau tidak
3	Memeriksa masukan deskripsi keadaan kosong atau tidak
4	Memeriksa masukan deskripsi tujuan kosong atau tidak



No.	Pseudocode
5	Memeriksa masukan deskripsi permasalahan kosong atau tidak
6	Memeriksa masukan deskripsi berkas kosong atau tidak
7	Pesan jika deskripsi pendahuluan kosong
8	Pesan jika deskripsi keadaan kosong
9	Pesan jika deskripsi tujuan kosong
10	Pesan jika deskripsi permasalahan kosong
11	Pesan jika deskripsi berkas kosong
12	Pemilihan kondisi <i>if</i> centang kelayakan
13	jika ada yang tidak tercentang
14	Deklarasi variabel data
15	Penyimpanan masukan data <i>draft_programa_id</i> oleh aktor
16	Penyimpanan masukan data <i>status_pendahuluan</i> oleh aktor
17	Penyimpanan masukan data <i>des_pendahuluan</i> oleh aktor
18	Penyimpanan masukan data <i>status_keadaan</i> oleh aktor
19	Penyimpanan masukan data <i>des_keadaan</i> oleh aktor
20	Penyimpanan masukan data <i>status_tujuan</i> oleh aktor
21	Penyimpanan masukan data <i>des_tujuan</i> oleh aktor
22	Penyimpanan masukan data <i>status_permasalahan</i> oleh aktor
23	Penyimpanan masukan data <i>des_permasalahan</i> oleh aktor
24	Penyimpanan masukan data <i>status_berkas</i> oleh aktor
25	Penyimpanan masukan data <i>des_berkas</i> oleh aktor
26	Deklarasi variabel notifikasi
27	Menyimpan data kedalaman variabel pengirim
28	Menyimpan data kedalaman variabel penerima
29	Menyimpan data kedalaman variabel <i>id_berkas</i>
30	Menyimpan data kedalaman variabel <i>type</i>
31	Menyimpan data kedalaman variabel deskripsi
32	Menyimpan data kedalaman variabel status
33	Deklarasi variabel notifikasi2
34	Menyimpan data kedalaman variabel pengirim
35	Menyimpan data kedalaman variabel penerima
36	Menyimpan data kedalaman variabel <i>id_berkas</i>
37	Menyimpan data kedalaman variabel <i>type</i>
38	Menyimpan data kedalaman variabel deskripsi
39	Menyimpan data kedalaman variabel status
40	Simpan data notifikasi ke dalam database
41	Deklarasi variabel status
42	Inisialisasi variabel status
43	<i>Else</i> jika semua sudah tercentang
44	Deklarasi variabel data
45	Penyimpanan masukan data <i>draft_programa_id</i> oleh aktor
46	Penyimpanan masukan data <i>status_pendahuluan</i> oleh aktor
47	Penyimpanan masukan data <i>des_pendahuluan</i> oleh aktor
48	Penyimpanan masukan data <i>status_keadaan</i> oleh aktor
49	Penyimpanan masukan data <i>des_keadaan</i> oleh aktor
50	Penyimpanan masukan data <i>status_tujuan</i> oleh aktor
51	Penyimpanan masukan data <i>des_tujuan</i> oleh aktor
52	Penyimpanan masukan data <i>status_permasalahan</i> oleh aktor
53	Penyimpanan masukan data <i>des_permasalahan</i> oleh aktor
54	Penyimpanan masukan data <i>status_berkas</i> oleh aktor
55	Penyimpanan masukan data <i>des_berkas</i> oleh aktor
56	Deklarasi variabel notifikasi
57	Menyimpan data kedalaman variabel pengirim
58	Menyimpan data kedalaman variabel penerima
59	Menyimpan data kedalaman variabel <i>id_berkas</i>
60	Menyimpan data kedalaman variabel <i>type</i>
61	Menyimpan data kedalaman variabel deskripsi
62	Menyimpan data kedalaman variabel status

No.	Pseudocode
63	Deklarasi variabel notifikasi2
64	Menyimpan data kedalaman variabel pengirim
65	Menyimpan data kedalaman variabel penerima
66	Menyimpan data kedalaman variabel id_berkas
67	Menyimpan data kedalaman variabel type
68	Menyimpan data kedalaman variabel deskripsi
69	Menyimpan data kedalaman variabel status
70	Deklarasi variabel status
71	Inisialisasi variabel status
72	Penyimpanan data draft programa kedalama database
73	Penyimpanan data verifikasi kedalam database
74	Simpan data notifikasi ke dalam database
75	Simpan data notifikasi2 ke dalam database
76	Deklarasi variabel notif
77	Inisialisasi variabel notif
78	Kembali kehalaman sebelumnya

3. Perancangan pseudocode untuk tambah laporan penyuluhan

1. Nama kelas : `PenyuluhController.php`
2. Nama operasi : `input_laporan_penyuluhan()`

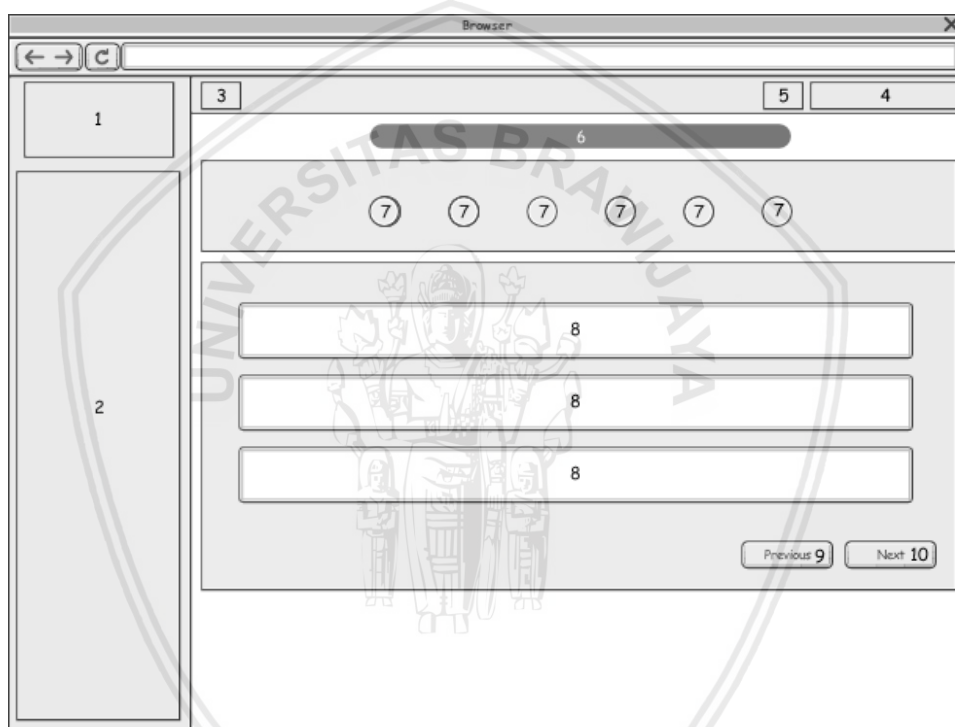
Tabel 5.3 Pseudocode Tambah Laporan Penyuluhan

No.	Pseudocode
1	Fungsi untuk memeriksa apakah ada data yang kosong
2	Memeriksa masukan deskripsi content kosong atau tidak
3	Memeriksa masukan deskripsi berkas kosong atau tidak
4	Memeriksa masukan deskripsi foto kosong atau tidak
5	Memeriksa masukan deskripsi dana terpakai kosong atau tidak
6	Memeriksa masukan deskripsi presensi kosong atau tidak
7	Pesan jika deskripsi konten kosong
8	Pesan jika deskripsi berkas kosong
9	Pesan jika deskripsi foto kosong
10	Pesan jika deskripsi dana terpakai kosong
11	Pesan jika deskripsi presensi kosong
12	Penyimpanan berkas laporan kedalam folder
13	Penyimpanan foto presensi kedalam folder
14	Penyimpanan foto kegiatan penyuluhan kedalam variabel
15	Perulangan <i>foreach</i> untuk melakukan penyimpanan karena jumlah foto kegiatan bisa lebih dari Satu
16	Penyimpanan foto kegiatan kedalam folder
17	Menyimpan nama-nama foto kedalam variabel dalam bentuk <i>json</i>
18	Menyimpan foto bukti pengiriman kedalam folder
19	Deklarasi variabel data
20	Inisialisasi variabel data content
21	Inisialisasi variabel data berkas
22	Inisialisasi variabel data foto
23	Inisialisasi variabel data dana_terpakai
24	Inisialisasi variabel data presensi
25	Inisialisasi variabel data status
26	Menyimpan foto bukti pengiriman ke database
27	Menyimpan laporan penyuluhan kedalam database
28	Deklarasi variabel notifikasi
29	Inisialisasi variabel notifikasi pengirim
30	Inisialisasi variabel notifikasi penerima

No.	Pseudocode
31	Inisialisasi variabel notifikasi id_berkas
32	Inisialisasi variabel notifikasi type
33	Inisialisasi variabel notifikasi deskripsi
34	Inisialisasi variabel notifikasi status
35	Menyimpan data notifikasi kedalam database
36	Deklrasi variabel notif
37	Inisialisasi variabel notif
38	Inisialisasi variabel notif
	Kembali kehalaman sebelumnya

5.1.5 Perancangan Antarmuka

5.1.5.1 Perancangan Antarmuka Tambah *Draft* Programa

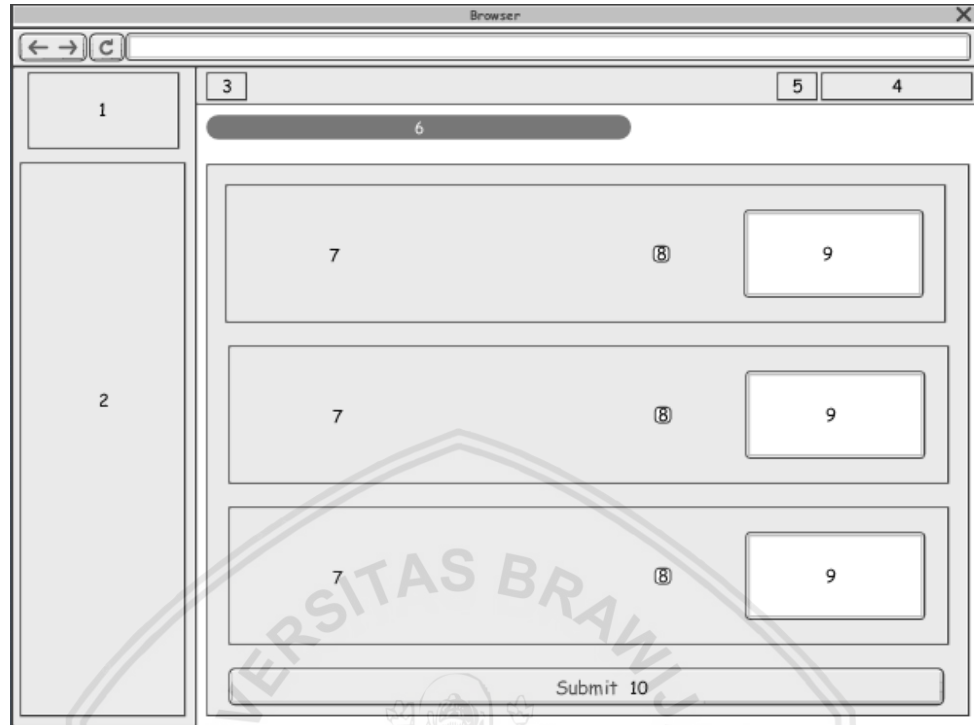


Gambar 5.8 Perancangan Antarmuka Tambah *Draft* programa

Keterangan:

- 1) Profil *user*
- 2) *Sidebar* menu
- 3) *Toogle Sidebar* menu
- 4) Menu pengguna
- 5) Notifikasi
- 6) Label halaman
- 7) Tab alur tambah *draft* programa
- 8) Input berupa text area
- 9) Tombol *Previous*
- 10) Tombol *Next*

5.1.5.2 Perancangan Antarmuka Verifikasi *Draft* Programa



Gambar 5.9 Perancangan Antarmuka Verifikasi *Draft* Programa

Keterangan:

- 1) Profil *user*
- 2) *Sidebar* menu
- 3) *Toogle Sidebar* menu
- 4) Menu pengguna
- 5) Notifikasi
- 6) Label judul halaman
- 7) Label konten
- 8) Checkbox untuk mencentang kelayakan
- 9) Input berupa text area untuk penjelasan kelayakan
- 10) Tombol Submit

5.1.5.3 Perancangan Antarmuka Tambah Laporan Penyuluhan



Gambar 5.10 Perancangan Antarmuka Tambah Laporan Penyuluhan

Keterangan:

- 1) Profil user
- 2) Sidebar menu
- 3) Toogle Sidebar menu
- 4) Menu pengguna
- 5) Notifikasi
- 6) Label judul halaman
- 7) Input text area untuk memasukan data laporan
- 8) Input file laporan penyuluhan
- 9) Input foto presensi
- 10) Input Foto-foto kegiatan penyuluhan
- 11) Tombol tambah input Gambar
- 12) Tombol Simpan
- 13) Input foto bukti pembayaran

5.2 Spesifikasi Sistem

Pada bagian ini akan menjelaskan spesifikasi dari sistem yang digunakan dalam pembangunan sebuah sistem informasi penyuluhan di dinas pertanian dan ketahanan pangan kabupaten padang pariaman. Terdapat 2 spesifikasi yang akan dijelaskan yaitu Spesifikasi *Hardware* dan Spesifikasi *Software*

5.2.1 Spesifikasi *Hardware*

Spesifikasi *Hardware* adalah spesifikasi yang digunakan dalam pengembangan sistem informasi penyuluhan di dinas pertanian dan ketahanan pangan kabupaten padang pariaman. Penjelasan tentang Spesifikasi *Hardware* yang digunakan dapat dilihat pada Tabel 5.4

Tabel 5.4 Spesifikasi *Hardware*

<i>Hardware</i>	Keterangan
<i>RAM</i>	4 GB
<i>Harddisk</i>	1000 GB
<i>Graphic Card</i>	NVIDIA GEFORCE 840
<i>Processor</i>	Intel(R) Core(TM) i5-5200 CPU @ 2.20GHz 2.20 GHz

5.2.2 Spesifikasi *Software*

Spesifikasi *Software* adalah spesifikasi yang digunakan dalam pengembangan sistem informasi penyuluhan di dinas pertanian dan ketahanan pangan kabupaten padang pariaman. Penjelasan tentang Spesifikasi *Software* yang digunakan dapat dilihat pada Tabel 5.5

Tabel 5.5 Spesifikasi *Software*

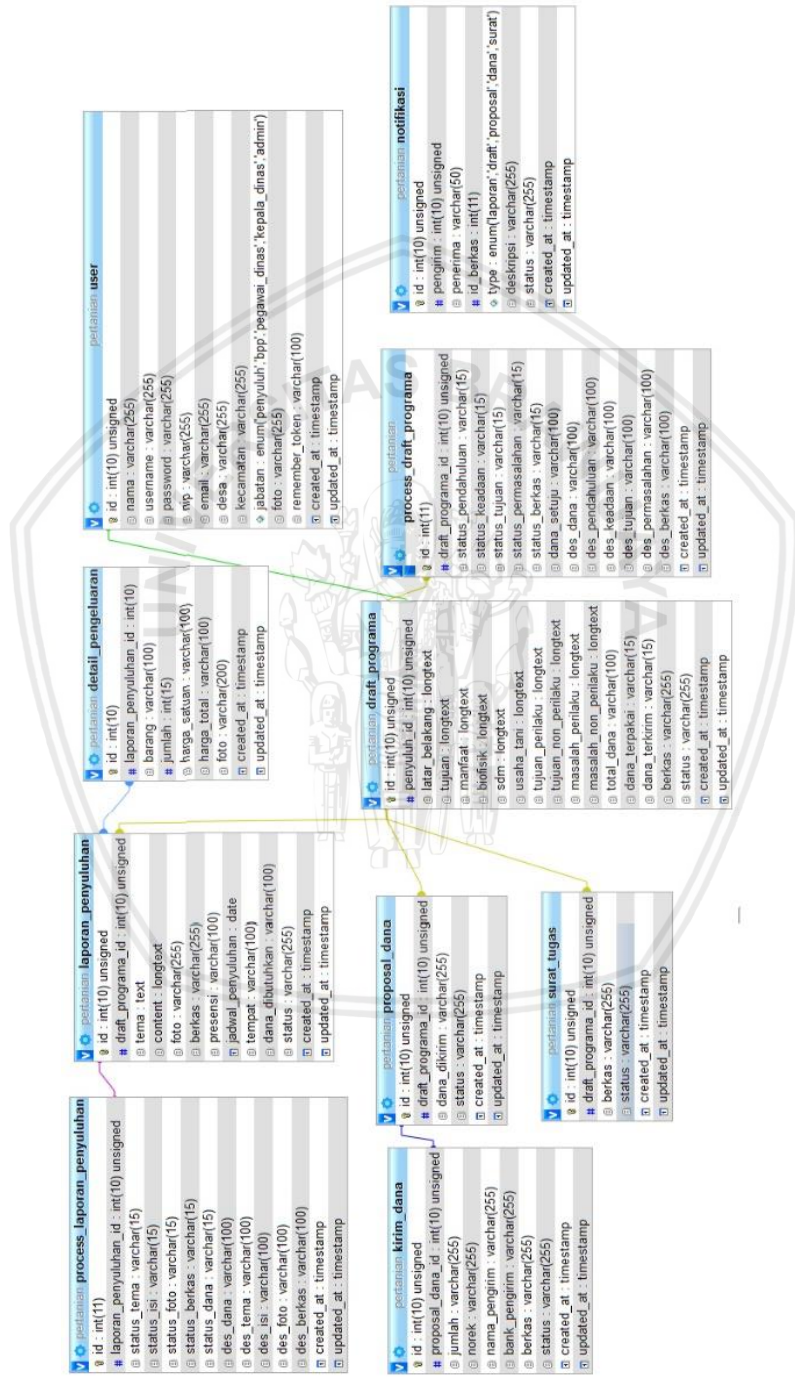
<i>Software</i>	Keterangan
Sistem Operasi	<i>Windows 10 64-bit</i>
<i>Framework</i>	<i>Laravel 5.5</i>
<i>Tools</i>	<i>Sublime Text 3, Mozilla Firefox</i>
DBMS	<i>MySQL</i>
<i>Server</i>	<i>Localhost</i>
Bahasa Pemrograman	<i>HTML, PHP, Javascript, CSS</i>

5.3 Implementasi Sistem Informasi Penyuluhan

Implementasi sistem informasi penyuluhan di dinas pertanian dan ketahanan pangan kabupaten padang pariaman dikembangkan dengan mengacu pada perancangan yang telah dirancang pada pembahasan bab sebelumnya. Adapun dua jenis implementasi sistem informasi yang digunakan adalah terdiri dari Implementasi Basis Data, kemudian Implementasi Algoritme dan terakhir Implementasi Antarmuka.

5.3.1 Implementasi Basis Data

Pada Gambar 5.11 merupakan hasil dari implementasi basis data yang digunakan pada sistem ini. Hasil pengimplementasiannya dari sistem ini yaitu terdapat 12 tabel *database*. Antara satu tabel dengan tabel yan lainnya memiliki kardinalitas.



Gambar 5.11 Implementasi Basis Data

Gambar 5.11 merupakan basis data sistem informasi penyuluhan. Terdapat beberapa tabel yaitu tabel draft_programa, jadwal, kirim_dana,

laporan_penyuluhan, notifikasi, process_draft_programa, process_laporan_penyuluhan, process_proposal_dana, process_surat_tugas, proposal_dana, surat-tugas, dan user. Setiap tabel memiliki atribut masing-masing dan *type* variabel masing yaitu ada berupa id, *varchar*, *timestamp* untuk menyimpan waktu, *longtext*. Setiap tabel memiliki hubungan atau relasi yaitu relasi *one to many*.

5.3.2 Implementasi Algoritme

Pada bab implementasi algoritme menjelaskan tentang algoritme yang diterapkan pada fungsi-fungsi sistem informasi penyuluhan yang telah terlebih dahulu dirancang pada pembahasan bab sebelumnya.

A. Fungsi Tambah *Draft* Program

Tabel 5.6 Implementasi algoritme tambah *draft* program

No.	Pseudocode
1	<code>\$this->validate(\$request, [</code>
2	<code>"f1-latar-belakang" => "required",</code>
3	<code>"f1-tujuan" => "required",</code>
4	<code>"f1-manfaat" => "required",</code>
5	<code>"f1-biofisik" => "required",</code>
6	<code>"f1-sdm" => "required",</code>
7	<code>"f1-usaha-tani" => "required",</code>
8	<code>"f1-tujuan-perilaku" => "required",</code>
9	<code>"f1-tujuan-non-perilaku" => "required",</code>
10	<code>"f1-masalah-perilaku" => "required",</code>
11	<code>"f1-masalah-non-perilaku" => "required",</code>
12	<code>"f1-total-dana" => "required",</code>
13	<code>"berkas" => "required"</code>
14	<code>],[</code>
15	<code>"f1-latar-belakang.required" => "form tidak</code> <code>boleh kosong",</code>
16	<code>"f1-tujuan.required" => "form tidak boleh</code> <code>kosong",</code>
17	<code>"f1-manfaat.required" => "form tidak boleh</code> <code>kosong",</code>
18	<code>"f1-biofisik.required" => "form tidak boleh</code> <code>kosong",</code>
19	<code>"f1-sdm.required" => "form tidak boleh</code> <code>kosong",</code>
21	<code>"f1-usaha-tani.required" => "form tidak boleh</code> <code>kosong",</code>
22	<code>"f1-tujuan-perilaku.required" => "form tidak</code> <code>boleh kosong",</code>
23	<code>"f1-tujuan-non-perilaku.required" => "form</code> <code>tidak boleh kosong",</code>
24	<code>"f1-masalah-perilaku.required" => "form</code> <code>tidak boleh kosong",</code>
25	<code>"f1-masalah-non-perilaku.required" => "form</code> <code>tidak boleh kosong",</code>
26	<code>"f1-total-dana.required" => "form tidak boleh</code> <code>kosong",</code>

No.	Pseudocode
27	<pre> "berkas.required" => "form tidak boleh kosong"]); \$berkas = \$request->berkas; \$berkasPath = \$berkas- >store('public/berkasDraft'); \$data = ["penyuluh_id" => auth()->guard('penyuluh')- >id(), "latar_belakang" => \$request['fl-latar- belakang'], "tujuan" => \$request['fl-tujuan'], 33 "manfaat" => \$request['fl-manfaat'], 34 "biofisik" => \$request['fl-biofisik'], 35 "sdm" => \$request['fl-sdm'], 36 "usaha_tani" => \$request['fl-usaha-tani'], 37 "tujuan_perilaku" => \$request['fl-tujuan- 38 perilaku'], "tujuan_non_perilaku" => \$request['fl- 39 tujuan-non-perilaku'], "masalah_perilaku" => \$request['fl-masalah- 41 perilaku'], "masalah_non_perilaku" => \$request['fl- 42 masalah-non-perilaku'], "berkas" => \$berkasPath, 43 "status" => "belum diperiksa bpp" 44]; \$draft = draft_programa::create(\$data)->id; \$notif = [47 "message" => "Input Data Berhasil", 48 "alert-type" => "success" 49]; return back()->with(\$notif); 50 51 </pre>

Pada Tabel 5.6 merupakan fungsi dari algoritme tambah *draft* programa. Fungsi ini dijalankan pada method *input_draft_programa()* yang berada pada class *PenyuluhController* dengan parameter *request* yang berfungsi menerima semua *inputan* yang dimasukan oleh actor, langkah awal dari fungsi ini yaitu melakukan pemeriksaan masukan data oleh actor dengan fungsi yang sudah di sediakan oleh *Laravel* yaitu *validate*, jika data yang di masukan tidak sesuai dengan yang di tentukan maka akan kembali ke halaman sebelum nya dengan menampilkan pesan *error*, jika sesuai maka langkah selanjutnya yaitu menghitung jumlah materi pada jadwal kegiatan yang nanti berguna untuk menjadi parameter pada perulangan for, selanjutnya di lakukan penyimpanan data materi dan jadwal ke dalam sebuah variable *data_jadwal* dimana di lakukan perulangan karena data materi dan jadwal lebih dari satu. Selanjutnya yaitu menyimpan data berkas yang dimasukan ke dalam sebuah folder yang sudah di tentukan, kemudian menyimpan masukan data-data oleh aktor kedalam sebuah variable yang bernama *data*. Selanjutnya data yang sudah di simpan ke dalam variabel sebelum nya akan di



simpan kedalam *database* draft_programa. Kemudian, akan di buat sebuah variabel bernama notif dan kembali ke halaman sebelumnya dengan membawa variabel notif sebagai pesan yang akan di tampilkan.

B. Fungsi Verifikasi *Draft* Program

Tabel 5.7 Implementasi algoritme varifikasi *draft* program

No.	Pseudocode
1	<code>\$this->validate(\$request, [</code>
2	<code>"des_pendahuluan" => "required",</code>
3	<code>"des_keadaan" => "required",</code>
4	<code>"des_tujuan" => "required",</code>
5	<code>"des_permasalahan" => "required",</code>
6	<code>"des_berkas" => "required",</code>
7	<code>],[</code>
8	<code>"des_pendahuluan.required" => "form tidak</code>
9	<code>boleh kosong",</code>
10	<code>"des_keadaan.required" => "form tidak boleh</code>
11	<code>kosong",</code>
12	<code>"des_tujuan.required" => "form tidak boleh</code>
13	<code>kosong",</code>
14	<code>"des_permasalahan.required" => "form tidak</code>
15	<code>boleh kosong",</code>
16	<code>"des_berkas.required" => "form tidak boleh</code>
17	<code>kosong",</code>
18	<code>]);</code>
19	<code>\$draft = draft_programa::where("id",\$id)-</code>
20	<code>>first();</code>
21	<code>if(\$request->pendahuluan && \$request->keadaan &&</code>
22	<code>\$request->tujuan && \$request->permasalahan && \$request-</code>
23	<code>>berkas){</code>
24	<code>\$data = [</code>
25	<code>"draft_programa_id" => \$id,</code>
26	<code>"status_pendahuluan" => \$request-</code>
27	<code>>pendahuluan,</code>
28	<code>"des_pendahuluan" => \$request-</code>
29	<code>>des_pendahuluan,</code>
30	<code>"status_keadaan" => \$request->keadaan,</code>
31	<code>"des_keadaan" => \$request->des_keadaan,</code>
32	<code>"status_tujuan" => \$request->tujuan,</code>
33	<code>"des_tujuan" => \$request->des_tujuan,</code>
34	<code>"status_permasalahan" => \$request-</code>
35	<code>>permasalahan,</code>
36	<code>"des_permasalahan" => \$request-</code>
37	<code>>des_permasalahan,</code>
38	<code>"status_berkas" => \$request->berkas,</code>
39	<code>"des_berkas" => \$request->des_berkas,</code>
40	<code>];</code>
41	<code>\$data_status = [</code>
42	<code>"status" => "valid oleh pegawai dinas"</code>
43	<code>];</code>



No.	Pseudocode
31	<code> \$data_notifikasi = [</code>
32	<code> "pengirim" => auth()-</code>
	<code>>guard('pegawai_dinas')->id(),</code>
33	<code> "penerima" => \$draft->user->kecamatan,</code>
34	<code> "id_berkas" => \$id,</code>
35	<code> "type" => "draft",</code>
36	<code> "deskripsi" => "draft ".\$draft->user-</code>
	<code>>nama." diterima pegawai dinas",</code>
37	<code> "status" => "belum di baca"</code>
38	<code>];</code>
39	<code> \$data_notifikasi2 = [</code>
40	<code> "pengirim" => auth()-</code>
	<code>>guard('pegawai_dinas')->id(),</code>
41	<code> "penerima" => \$draft->user->id,</code>
42	<code> "id_berkas" => \$id,</code>
43	<code> "type" => "draft",</code>
44	<code> "deskripsi" => "draft program diterima</code>
	<code>pegawai dinas",</code>
45	<code> "status" => "belum di baca"</code>
46	<code>];</code>
47	
48	<code> }else{</code>
49	<code> \$data = [</code>
50	<code> "draft_programa_id" => \$id,</code>
51	<code> "status_pendahuluan" => \$request-</code>
	<code>>pendahuluan,</code>
52	<code> "des_pendahuluan" => \$request-</code>
	<code>>des_pendahuluan,</code>
53	<code> "status_keadaan" => \$request->keadaan,</code>
54	<code> "des_keadaan" => \$request->des_keadaan,</code>
55	<code> "status_tujuan" => \$request->tujuan,</code>
56	<code> "des_tujuan" => \$request->des_tujuan,</code>
57	<code> "status_permasalahan" => \$request-</code>
	<code>>permasalahan,</code>
58	<code> "des_permasalahan" => \$request-</code>
	<code>>des_permasalahan,</code>
59	<code> "status_berkas" => \$request->berkas,</code>
60	<code> "des_berkas" => \$request->des_berkas,</code>
61	<code>];</code>
62	<code> \$data_notifikasi = [</code>
63	<code> "pengirim" => auth()-</code>
	<code>>guard('pegawai_dinas')->id(),</code>
64	<code> "penerima" => \$draft->user->kecamatan,</code>
65	<code> "id_berkas" => \$id,</code>
66	<code> "type" => "draft",</code>
67	<code> "deskripsi" => "draft ".\$draft->user-</code>
	<code>>nama." ditolak pegawai dinas",</code>
68	<code> "status" => "belum di baca"</code>
69	<code>];</code>
70	<code> \$data_notifikasi2 = [</code>



No.	Pseudocode
71	<code> "pengirim" => auth()-</code>
	<code>>guard('pegawai_dinas')->id(),</code>
72	<code> "penerima" => \$draft->user->id,</code>
73	<code> "id_berkas" => \$id,</code>
74	<code> "type" => "draft",</code>
75	<code> "deskripsi" => "draft program ditolak</code>
	<code>pegawai dinas",</code>
76	<code> "status" => "belum di baca"</code>
77	<code>];</code>
78	<code> \$data_status = [</code>
79	<code> "status" => "invalid oleh pegawai dinas"</code>
80	<code>];</code>
81	<code> }</code>
82	<code> draft_programa::where('id',\$id)-</code>
	<code>>update(\$data_status);</code>
83	
84	<code>process_draft_programa::where('draft_programa_id',\$id)-</code>
	<code>>update(\$data);</code>
85	<code> notifikasi::create(\$data_notifikasi);</code>
86	<code> notifikasi::create(\$data_notifikasi2);</code>
87	<code> \$notif = [</code>
88	<code> "message" => "Update Data Berhasil",</code>
89	<code> "alert-type" => "success"</code>
90	<code>];</code>
91	<code> return</code>
92	<code> redirect('pegawai_dinas/detail_draft_programa/'.\$id)-</code>
	<code>>with(\$notif);</code>

Pada Tabel 5.7 merupakan fungsi dari algoritme verifikasi *draft* program. Fungsi ini dijalankan pada method *input_process_draft_programa()* yang berada pada *class PegawaiDinasController* pada method ini membawa sebuah parameter *request* dimana parameter ini berfungsi untuk membawa semua masukan user pada sistem. Pada awal fungsi ini melakukan pemeriksaan terhadap masukan data oleh pengguna, jika data masi ada yang belum sesuai maka akan kembali ke halaman sebelumnya dengan menampilkan pesan *error*, jika sudah sesuai maka akan di lanjutkan dengan pemeriksaan terhadap masukan centang kelayakan oleh aktor, ini berfungsi untuk menentukan status *draft* program, jika semua kelayakan sudah tercentang maka status *draft* program adalah *valid* atau di setujui, jika ada yang belum tercentang maka status *draft* program adalah *invalid* atau tidak disetujui. Di dalam fungsi *if* ini melakukan penyimpanan data masukan kedalam variabel data yang berbentuk array. Selanjut nya melakukan penyimpanan data kedalam *database draft_programa* dan *process_draft_programa*. Selanjutnya, menyimpan data kedalam notif dan kembali ke halaman sebelumnya dengan membawa data *notif* yang nanti akan di tampilkan sebagai pesan sukses melakukan penyimpanan.

C. Fungsi Tambah Laporan Penyuluhan

Tabel 5.8 Implementasi algoritme tambah laporan penyuluhan

No.	Pseudocode
1	<code>\$this->validate(\$request, [</code>
2	<code>"content" => "required",</code>
3	<code>"berkas" => "required mimes:docx,pdf",</code>
4	<code>"foto" => "required",</code>
5	<code>"dana_terpakai" => "required",</code>
6	<code>"presensi" => "required",</code>
7	<code>],[</code>
8	<code>'content.required' => 'Content tidak boleh</code>
9	<code>kosong',</code>
10	<code>'berkas.required' => 'Berkas tidak boleh</code>
11	<code>kosong',</code>
12	<code>'foto.required' => 'Foto tidak boleh kosong',</code>
13	<code>'presensi.required' => 'Foto presensi tidak</code>
14	<code>boleh kosong',</code>
15	<code>'dana_terpakai.required' => 'dana_terpakai</code>
16	<code>tidak boleh kosong',</code>
17	<code>'berkas.mimes' => 'Foto harus bertype docx</code>
18	<code>atau pdf',</code>
19	<code>]);</code>
20	<code>\$berkas = \$request->berkas;</code>
21	<code>\$berkasPath = \$berkas-</code>
22	<code>>store('public/berkasLaporan');</code>
23	<code>\$fotoPresensi = \$request->presensi;</code>
24	<code>\$fotoPresensiPath = \$fotoPresensi-</code>
25	<code>>store('public/fotoPresensi');</code>
26	<code>\$foto = \$request->foto;</code>
27	<code>foreach (\$foto as \$photo) {</code>
28	<code>\$fotoPath[] = \$photo-</code>
29	<code>>store('public/fotoLaporan');</code>
30	<code>}</code>
31	<code>\$fotoPathJson = json_encode(\$fotoPath);</code>
32	<code>\$bukti = \$request->bukti;</code>
33	<code>foreach (\$bukti as \$bk) {</code>
34	<code>\$buktiPath[] = \$bk-</code>
35	<code>>store('public/fotoBukti');</code>
36	<code>}</code>
37	<code>\$data = [</code>
38	<code>"content" => \$request->content,</code>
39	<code>"berkas" => \$berkasPath,</code>
40	<code>"foto" => \$fotoPathJson,</code>
41	<code>"dana_terpakai" => \$request->dana_terpakai,</code>
42	<code>"presensi" => \$fotoPresensiPath,</code>
43	<code>"status" => "belum diperiksa bpp"</code>
44	<code>];</code>
45	<code>\$detail =</code>
46	<code>detail_pengeluaran::where("laporan_penyuluhan_id", \$reque</code>
47	<code>st->id)->get();</code>
48	<code>for (\$i=0;\$i<count(\$detail);\$i++) {</code>
49	<code>\$detail[\$i]->update(["foto" =></code>
50	<code>\$buktiPath[\$i]]);</code>
51	<code>}</code>



No.	Pseudocode
39	if (laporan_penyuluhan::where('id',\$request-
40	>id)->update(\$data)) {
	\$data_notifikasi = [
41	"pengirim" => auth()->guard('penyuluh')-
42	>id(),
	"penerima" => auth()->guard('penyuluh')-
43	>user()->kecamatan,
	"id_berkas" => \$request->id,
44	"type" => "laporan",
45	"deskripsi" => "laporan penyuluhan baru",
46	"status" => "belum di baca"
];
47	notifikasi::create(\$data_notifikasi);
48	\$notif = [
49	'message' => 'Input Data laporan
50	Berhasil',
51	'alert-type' => 'success'
];
52	}else{
53	\$notif = [
54	'message' => 'Input Data Laporan Gagal',
55	'alert-type' => 'error'
56];
57	}
58	return back()->with(\$notif);
59	
60	

Pada Tabel 5.8 merupakan fungsi dari algoritme tambah laporan penyuluhan. Fungsi ini dijalankan pada method *input_laporan_penyuluhan()* yang berada pada *class PenyuluhController*. Langkah awal fungsi ini yaitu melakukan pemeriksaan terhadap masukan data oleh aktor, jika data yang dimasukkan tidak sesuai maka kembali ke halaman sebelumnya dengan menampilkan pesan *error*, jika data yang di masukan sesuai maka selanjutnya menyimpan berkas laporan dan berkas presentasi ke dalam folder, selanjutnya menyimpan berkas foto kedalam folder dengan menggunakan perulangan karena jumlah foto bisa lebih dari satu. Selanjutnya, menyimpan data yang di masukan kedalam variabel dan menyimpan ke dalam *database*. Kemudian melakukan penyimpanan data notifikasi kedalam variabel dan ke dalam *database*. Pada tahap akhir menyimpan data kedalam variabel notif dalam bentuk array dan kembali ke halaman sebelumnya dengan membawa variabel notif yang nantinya akan di tampilkan dalam pesan sukses.

5.3.3 Implementasi Antar Muka

Pada bab Implementasi Antarmuka membahas tentang penjelasan hasil implementasi antarmuka yang telah dirancang pada bab sebelumnya. Antarmuka merupakan penghubung interaksi antara pengguna atau *user* dengan sistem.

A. Halaman Tambah *Draft* Programa

The screenshot displays a web application interface for creating a program draft. At the top, there is a navigation bar with a user profile 'Gal Gadot' and a notification bell. Below this is a sidebar menu with options: Profil, Laporan Penyuluhan, Draft Programma, Daftar Surat Tugas, and Daftar Pengiriman Dana. The main content area is titled 'FORM MEMBUAT DRAFT PROGRAMMA' and features a horizontal progress indicator with six stages: Pendahuluan, Keadaan, Tujuan (highlighted in red), Permasalahan, Berkas, and Jadwal. Underneath the progress bar, there are two text input fields. The first is labeled 'Tujuan Bersifat Perilaku (maksimal 1000 kata)' and contains the text 'perubahan yang akan dicapai dalam kurun waktu setahun bersifat perilaku'. The second is labeled 'Tujuan bersifat Non perilaku (maksimal 1000 kata)' and contains the text 'perubahan yang akan dicapai dalam kurun waktu setahun bersifat non perilaku'. Both fields show a character count of 'jumlah kata : 0'. At the bottom right of the form area, there are two buttons: 'Previous' (disabled) and 'Next' (active).

Gambar 5.12 Halaman Tambah Draft Programma

Pada Gambar 5.12 merupakan halaman tambah *draft* programma. Pada halaman ini aktor dapat memasukan data secara bertahap-tahap. Pada halaman ini sistem menyediakan beberapa *form text*, tanggal, berkas, dan tombol. *Form text* berupa masukan data latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan dan usaha tani, tujuan bersifat perilaku, tujuan bersifat *non* perilaku, masalah bersifat perilaku, masalah *non* bersifat perilaku, jumlah dana yang di butuhkan. Pada *form* berkas berupa masukan berkas *draft* programma. Pada form tanggal berguna untuk memasukan jadwal penyuluhan. Tombol berupa tombol *previous* untuk kembali ke tahap sebelumnya, tombol *next* pindah ke tahap selanjutnya, tombol *submit* untuk melakukan penyimpanan ke dalam *database*.

B. Halaman Verifikasi *Draft* Programa

Ceklis Kelayakan	Verifikasi(Ceklis jika sudah layak)	Penjelasan
Pendahuluan Pendahuluan di lengkapi dengan latar belakang, tujuan dan manfaat. latar belakang Latar belakang, menjelaskan munculnya masalah atau pertanyaan penelitian yang merupakan inferensi atau pengambilan kesimpulan dari fakta-fakta pendukung yang terdapat di draft programa sebelumnya atau di lapangan (misalnya hasil pengamatan atau wawancara). Latar belakang harus bisa menunjukkan mengapa permasalahan yang diangkat dianggap penting. tujuan Tujuan, dituliskan dalam kalimat pernyataan yang sederhana dan jelas sesuai dengan masalah penelitian dan hasil yang ingin dicapai. manfaat Manfaat, menuliskan kontribusi skripsi terhadap ruang lingkup yang lebih dan/atau terhadap para pemangku kepentingan (stakeholders).	<input type="checkbox"/>	
Keadaan Dilengkapi dengan data biofisik, sumber daya manusia, dan kelembagaan usaha tani Keadaan, menuliskan keadaan atau kondisi yang ada pada target penelitian baik secara umum atau khusus.	<input type="checkbox"/>	
Tujuan		

Gambar 5.13 Halaman Verifikasi *Draft* Programa

Pada Gambar 5.13 merupakan halaman verifikasi *draft* programa. Halaman ini menyediakan proses verifikasi yang dapat di lakukan dengan menceklis kelayakan dan memberi penjelasan terhadap kelayakan *draft*. Sistem menyediakan tombol untuk mencentang kelayakan, *form* text untuk memberikan penjelasan terhadap kelayakan *draft* programa, dan tombol simpan untuk menyimpan data verifikasi kedalam *database*.

C. Halaman Tambah Laporan Penyuluhan

Form Laporan Kegiatan Penyuluhan Pertanian

Konten penyuluhan (maksimal 1000 kata)

Jumlah kata : 0

Berkas Laporan penyuluhan (docx/pdf)

Browse... surat tugas.docx

Foto Presensi (jpg/jpeg/png)

Browse... 2017-07-24_211717.jpg

Foto Kegiatan penyuluhan (jpg/jpeg/png)

Browse... No file selected.

Masukkan foto bukti pembayaran

barang yang dibeli	jumlah barang	harga satuan	total harga	Bukti Pembayaran
bensin	3	10.000	30.000	Browse... 01ad809b.jpg
airqa gelas dus	3	25.000	75.000	Browse... 15d04d8951_25-10-2018_20.jpg

Simpan

Gambar 5.14 Halaman Tambah Laporan penyuluhan

Pada Gambar 5.14 merupakan halaman untuk menambahkan laporan penyuluhan. Sistem menyediakan beberapa form *text*, berkas, dan tombol. *Form*

text berguna untuk memasukan data berupa konten penyuluhan dan jumlah dana yang terpakai, *form* berkas berupa berkas laporan, berkas foto presentasi, berkas bukti pembayaran berupa foto dan berkas foto-foto kegiatan, tombol berupa tombol tambah yang berguna untuk menambah *form* foto kegiatan dan tombol submit yang berguna untuk menyimpan data kedalam *database*.



BAB 6 PENGUJIAN

Pengujian dilakukan di akhir pengerjaan *software development life cycle*. Pengujian bertujuan untuk memastikan apakah sistem yang dibuat sudah sesuai dengan tahap sebelumnya yaitu analisis kebutuhan dan perancangan sistem. Metode pengujian yang dilakukan adalah metode *whitebox testing* untuk melakukan pengujian kode program dan metode *blackbox testing* untuk melakukan pengujian pada fungsionalitas dari sistem ini.

6.1 Whitebox Testing

Whitebox testing bertujuan untuk menguji sistem berdasarkan implementasi kode. Teknik yang digunakan pada *whitebox testing* ini adalah *Cyclomatic Complexity* dan *basis path testing*. *Test case* dalam pengujian ini berdasarkan pada jalur independen yang didapatkan dari hasil pembuatan *basis path testing* dan *Cyclomatic Complexity*.

6.1.1 Whitebox Testing Tambah Draft Programa

1. Algoritme

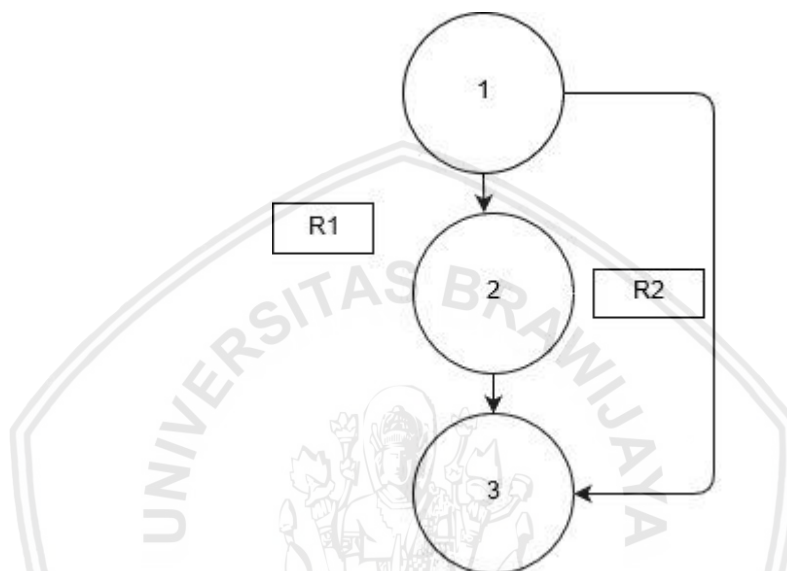
Tabel 6.1 Algoritme Tambah Draft Programa

No.	pseudocode
1	Fungsi pemeriksaan kondisi apakah ada data yang kosong
1	Memeriksa inputan latar belakang kosong atau tidak
1	Memeriksa inputan tujuan kosong atau tidak
1	Memeriksa inputan manfaat kosong atau tidak
1	Memeriksa inputan biofisik kosong atau tidak
1	Memeriksa inputan sdm kosong atau tidak
1	Memeriksa inputan usaha tani kosong atau tidak
1	Memeriksa inputan tujuan perilaku kosong atau tidak
1	Memeriksa inputan tujuan non perilaku kosong atau tidak
1	Memeriksa inputan masalah perilaku kosong atau tidak
1	Memeriksa inputan masalah non perilaku kosong atau tidak
1	Memeriksa inputan total dana kosong atau tidak
1	Memeriksa inputan berkas kosong atau tidak
2	Deklarasi dan inisialisasi berkas
2	Menyimpan data berkas kedalam database
2	Deklarasi variabel data
2	Menyimpan data <i>request</i> id kedalam variabel
2	Menyimpan data <i>request</i> latar belakang kedalam variabel
2	Menyimpan data <i>request</i> tujuan kedalam variabel
2	Menyimpan data <i>request</i> manfaat kedalam variabel
2	Menyimpan data <i>request</i> biofisik kedalam variabel
2	Menyimpan data <i>request</i> sdm kedalam variabel
2	Menyimpan data <i>request</i> usaha tani kedalam variabel
2	Menyimpan data <i>request</i> tujuan perilaku kedalam variabel
2	Menyimpan data <i>request</i> tujuan non perilaku kedalam variabel
2	Menyimpan data <i>request</i> masalah perilaku kedalam variabel
2	Menyimpan data <i>request</i> masalah non perilaku kedalam variabel
2	Menyimpan data <i>request</i> total dana kedalam variabel
2	Menyimpan data <i>request</i> berkas kedalam variabel
2	Menyimpan data <i>request</i> status kedalam variabel
2	Menyimpan data status kedalam variabel

No.	pseudocode
2	Menyimpan data draft kedalam database
2	Dekarasi variabel notif
2	Menyimpan data kedalam variabel pesan
2	Menyimpan data kedalam variabel type
3	Kembali kehalaman sebelumnya

2. Basis Path Testing

1.1 Flow Graph



Gambar 6.1 Flow Graph method tambah data program

1.2 Cyclomatic Complexity

- $V(G) = 2, (R1, R2)$
- $V(G) = \text{jumlah edges} - \text{jumlah nodes} + 2 = 3 - 2 + 2 = 2$
- $V(G) = \text{jumlah predicate nodes} + 1 = 1 + 1 = 2$

1.3 Independent Path

- Jalur 1: 1-3
- Jalur 2: 1-2-3

1.4 Test Case

Tabel 6.2 Test Case Tambah Draft Program

No. Jalur	Kondisi	Expected Result	Actual Result	Status
1	fungsi tambah draft program dijalankan dengan salah satu	Gagal menambah	Gagal menambah	Valid

No. Jalur	Kondisi	Expected Result	Actual Result	Status
	<i>form</i> kosong yaitu data latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan non perilaku, masalah bersifat perilaku dan <i>non</i> perilaku.	<i>draft</i> programa.	<i>Draft</i> prgorama.	
2	Fungsi <i>draft</i> programa dijalankan dengan mengisi semua data yaitu data latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan non perilaku, masalah bersifat perilaku dan <i>non</i> perilaku.	Berhasil menambah <i>draft</i> programa.	Berhasil menambah <i>draft</i> programa.	Valid

6.1.2 Whitebox Testing Verifikasi Draft Prgorama

1. Algoritme

Tabel 6.3 Algoritma Verifikasi Draft Programa

No.	Pseudocode
1	Fungsi untuk memeriksa apakah ada data yang kosong
1	Memeriksa masukan deskripsi pendahuluan kosong atau tidak
1	Memeriksa masukan deskripsi keadaan kosong atau tidak
1	Memeriksa masukan deskripsi tujuan kosong atau tidak
1	Memeriksa masukan deskripsi permasalahan kosong atau tidak
1	Memeriksa masukan deskripsi berkas kosong atau tidak
1	Pesan jika deskripsi pendahuluan kosong
1	Pesan jika deskripsi keadaan kosong
1	Pesan jika deskripsi tujuan kosong
1	Pesan jika deskripsi permasalahan kosong



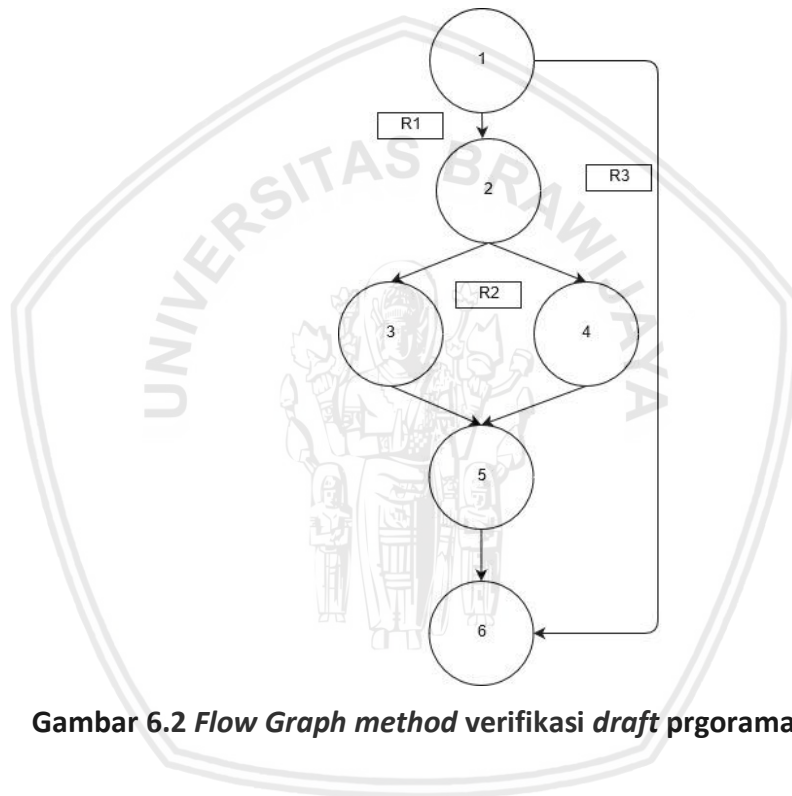
No.	Pseudocode
1	Pesan jika deskripsi berkas kosong
2	Pemilihan kondisi <i>if</i> centang kelayakan tercentang semua
3	Deklarasi variabel data
3	Penyimpanan masukan data <i>draft_programa_id</i> oleh aktor
3	Penyimpanan masukan data <i>status_pendahuluan</i> oleh aktor
3	Penyimpanan masukan data <i>des_pendahuluan</i> oleh aktor
3	Penyimpanan masukan data <i>status_keadaan</i> oleh aktor
3	Penyimpanan masukan data <i>des_keadaan</i> oleh aktor
3	Penyimpanan masukan data <i>status_tujuan</i> oleh aktor
3	Penyimpanan masukan data <i>des_tujuan</i> oleh aktor
3	Penyimpanan masukan data <i>status_permasalahan</i> oleh aktor
3	Penyimpanan masukan data <i>des_permasalahan</i> oleh aktor
3	Penyimpanan masukan data <i>status_berkas</i> oleh aktor
3	Penyimpanan masukan data <i>des_berkas</i> oleh aktor
3	Deklrasi variabel notifikasi
3	Menyimpan data kedalaman variabel pengirim
3	Menyimpan data kedalaman variabel penerima
3	Menyimpan data kedalaman variabel <i>id_berkas</i>
3	Menyimpan data kedalaman variabel <i>type</i>
3	Menyimpan data kedalaman variabel deskripsi
3	Menyimpan data kedalaman variabel status
3	Deklrasi variabel notifikasi2
3	Menyimpan data kedalaman variabel pengirim
3	Menyimpan data kedalaman variabel penerima
3	Menyimpan data kedalaman variabel <i>id_berkas</i>
3	Menyimpan data kedalaman variabel <i>type</i>
3	Menyimpan data kedalaman variabel deskripsi
3	Menyimpan data kedalaman variabel status
3	Simpan data notifikasi ke dalam database
3	Deklarasi variabel status
3	Inisialisasi variabel status
4	<i>Else</i> jika semua sudah tercentang
5	Deklarasi variabel data
5	Penyimpanan masukan data <i>draft_programa_id</i> oleh aktor
5	Penyimpanan masukan data <i>status_pendahuluan</i> oleh aktor
5	Penyimpanan masukan data <i>des_pendahuluan</i> oleh aktor
5	Penyimpanan masukan data <i>status_keadaan</i> oleh aktor
5	Penyimpanan masukan data <i>des_keadaan</i> oleh aktor
5	Penyimpanan masukan data <i>status_tujuan</i> oleh aktor
5	Penyimpanan masukan data <i>des_tujuan</i> oleh aktor
5	Penyimpanan masukan data <i>status_permasalahan</i> oleh aktor
5	Penyimpanan masukan data <i>des_permasalahan</i> oleh aktor
5	Penyimpanan masukan data <i>status_berkas</i> oleh aktor
5	Penyimpanan masukan data <i>des_berkas</i> oleh aktor
5	Deklrasi variabel notifikasi
5	Menyimpan data kedalaman variabel pengirim
5	Menyimpan data kedalaman variabel penerima
5	Menyimpan data kedalaman variabel <i>id_berkas</i>
5	Menyimpan data kedalaman variabel <i>type</i>
5	Menyimpan data kedalaman variabel deskripsi
5	Menyimpan data kedalaman variabel status
5	Deklrasi variabel notifikasi2
5	Menyimpan data kedalaman variabel pengirim
5	Menyimpan data kedalaman variabel penerima
5	Menyimpan data kedalaman variabel <i>id_berkas</i>
5	Menyimpan data kedalaman variabel <i>type</i>
5	Menyimpan data kedalaman variabel deskripsi
5	Menyimpan data kedalaman variabel status



No.	Pseudocode
5	Deklarasi variabel status
5	Inisialisasi variabel status
6	Penyimpanan data draft programa kedalama database
6	Penyimpanan data verifikasi kedalam database
6	Simpan data notifikasi ke dalam database
6	Simpan data notifikasi2 ke dalam database
6	Deklrasi variabel notif
6	Inisialisasi variabel notif
7	Kembali kehalaman sebelumnya

2. Basis Path Testing

2.1 Flow Graph



Gambar 6.2 Flow Graph method verifikasi draft prgorama

2.2 Cyclomatic Complexity

- $V(G) = 3$, (R1,R2, dan R3)
- $V(G) = \text{jumlah edges} - \text{jumlah nodes} + 2 = 7 - 6 + 2 = 3$
- $V(G) = \text{jumlah predicate nodes} + 1 = 2 + 1 = 3$

2.3 Independent Path

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

2.4 Test Case

Tabel 6.4 Test Case Verifikasi Draft Program

No. Jalur	Kondisi	Expected Result	Actual Result	Status
1	fungsi verifikasi <i>draft</i> program dijalankan dengan salah satu <i>form</i> kosong yaitu penjelasan kelayakan pendahuluan, penjelasan kelayakan keadaan, penjelasan kelayakan tujuan, penjelasan kelayakan permasalahan, penjelasan kelayakan berkas.	Gagal memverifikasi <i>draft</i> program.	Gagal memverifikasi <i>Draft</i> program.	Valid
2	fungsi verifikasi <i>draft</i> program dijalankan dengan mencentang semua kelayakan dan mengisi semua data yaitu penjelasan kelayakan pendahuluan, penjelasan kelayakan keadaan, penjelasan kelayakan tujuan, penjelasan kelayakan permasalahan, penjelasan kelayakan berkas.	Berhasil memverifikasi <i>draft</i> program dan status <i>draft</i> program valid.	Berhasil memverifikasi <i>draft</i> program dan status <i>draft</i> program valid.	Valid
3	fungsi verifikasi <i>draft</i> program dijalankan dengan tidak mencentang satu atau lebih kelayakan dan mengisi semua data yaitu penjelasan kelayakan pendahuluan, penjelasan kelayakan keadaan, penjelasan kelayakan tujuan, penjelasan kelayakan permasalahan,	Berhasil memverifikasi <i>draft</i> program dan status <i>draft</i> program unvalid.	Berhasil memverifikasi <i>draft</i> program dan status <i>draft</i> program unvalid.	valid

No. Jalur	Kondisi	Expected Result	Actual Result	Status
	penjelasan kelayakan berkas.			

6.1.3 Whitebox Testing Tambah Laporan Penyuluhan

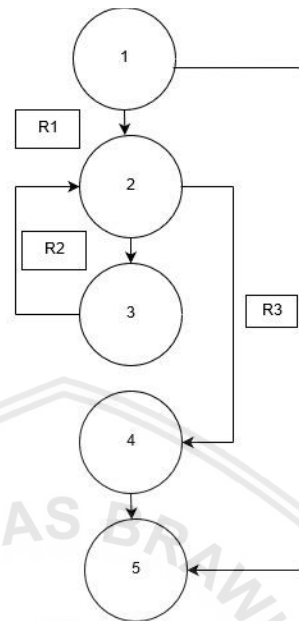
1. Algoritme

Tabel 6.5 Algoritma Tambah Laporan Penyuluhan

No.	Pseudocode
1	Fungsi untuk memeriksa apakah ada data yang kosong
1	Memeriksa masukan deskripsi content kosong atau tidak
1	Memeriksa masukan deskripsi berkas kosong atau tidak
1	Memeriksa masukan deskripsi foto kosong atau tidak
1	Memeriksa masukan deskripsi dana terpakai kosong atau tidak
1	Memeriksa masukan deskripsi presensi kosong atau tidak
1	Pesan jika deskripsi konten kosong
1	Pesan jika deskripsi berkas kosong
1	Pesan jika deskripsi foto kosong
1	Pesan jika deskripsi dana terpakai kosong
1	Pesan jika deskripsi presensi kosong
1	Penyimpanan berkas laporan kedalam folder
1	Penyimpanan foto presensi kedalam folder
1	Penyimpanan foto kegiatan penyuluhan kedalam variabel
2	Perulangan <i>foreach</i> untuk melakukan penyimpanan karena jumlah foto kegiatan bisa lebih dari Satu
3	Penyimpanan foto kegiatan kedalam folder
3	Menyimpan nama-nama foto kedalam variabel dalam bentuk <i>json</i>
4	Menyimpan foto bukti pengiriman kedalam folder
4	Deklrasi variabel data
4	Inisialisasi variabel data
4	Inisialisasi variabel data
4	Inisialisasi variabel data
4	Inisialisasi variabel data
4	Inisialisasi variabel data
4	Inisialisasi variabel data
4	Inisialisasi variabel data
4	Menyimpan foto bukti pengiriman kedalam database
4	Menyimpan laporan penyuluhan kedalam database
4	Deklrasi variabel notiifikasi
4	Inisialisasi variabel notifikasi
4	Inisialisasi variabel notifikasi
4	Inisialisasi variabel notifikasi
4	Inisialisasi variabel notifikasi
4	Inisialisasi variabel notifikasi
4	Inisialisasi variabel notifikasi
4	Menyimpan data notifikasi kedalam database
4	Deklrasi variabel notif
4	Inisialisasi variabel notif
4	Inisialisasi variabel notif
5	Kembali kehalaman sebelumnya



2. Basis Path Testing
1.1 Flow Graph



Gambar 6.3 Flow Graph Tambah laporan Penyuluhan

1.2 Cyclomatic Complexity

- $V(G) = 3$, (R1, R2, dan R3)
- $V(G) = \text{jumlah edges} - \text{jumlah nodes} + 2 = 6 - 5 + 2 = 3$
- $V(G) = \text{jumlah predicate nodes} + 1 = 2 + 1 = 3$

1.3 Independent Path

- Jalur 1: 1-5
- Jalur 2: 1-2-3-2-4-5
- Jalur 3: 1-2-3-2-3-2-4-5

3. Test Case

Tabel 6.6 Test Case Tambah Laporan Penyuluhan

No. Jalur	Kondisi	Expected Result	Actual Result	Status
1	fungsi tambah laporan penyuluhan dijalankan dengan salah satu form kosong yaitu form	Gagal menambah laporan penyuluhan.	Gagal menambah laporan penyuluhan .	Valid

No. Jalur	Kondisi	<i>Expected Result</i>	<i>Actual Result</i>	Status
	konten penyuluhan, dana yang terpakai, berkas laporan penyuluhan, presensi, foto kegiatan penyuluhan yang dapat di inputkan lebih dari satu, foto bukti pembayaran atau pengeluaran dana.			
2	fungsi tambah laporan dijalankan penyuluhan dengan mengisi satu foto kegiatan dan mengisi semua data yaitu data konten penyuluhan, dana yang terpakai, berkas laporan penyuluhan, presensi, foto kegiatan penyuluhan yang dapat di inputkan lebih dari satu, foto bukti pembayaran atau pengeluaran dana.	Berhasil menambah laporan penyuluhan dan foto kegiatan satu buah.	Berhasil menambah laporan penyuluhan dan foto kegiatan satu buah.	Valid
3	fungsi tambah laporan penyuluhan dijalankan dengan mengisi foto kegiatan lebih dari satu dan mengisi semua data yaitu data konten penyuluhan, dana yang terpakai,	Berhasil menambah laporan penyuluhan dan foto kegiatan lebih dari satu.	Berhasil menambah laporan penyuluhan dan foto kegiatan lebih dari satu.	Valid

No. Jalur	Kondisi	Expected Result	Actual Result	Status
	berkas laporan penyuluhan, presensi, foto kegiatan penyuluhan yang dapat di inputkan lebih dari satu, foto bukti pembayaran atau pengeluaran dana..			

6.2 Blackbox Testing

Blackbox testing bertujuan untuk menguji sistem berdasarkan pada fungsionalitas sistem tanpa melihat kode dari sistem. Pengujian akan dilakukan dengan menjalankan kasus uji berdasarkan pada *use case* yang telah ditentukan pada tahap analisis kebutuhan. Pada penelitian ini teknik yang digunakan untuk menguji fungsionalitas yaitu *Boundary Value Analysis*.

1. Kasus uji *login*

Tabel 6.7 Tabel Kasus Uji Login

Nama Kasus Uji	<i>Login</i>
Kode Kebutuhan	SIPP_F_001
Tujuan Pengujian	Untuk memastikan <i>user</i> dapat mengakses halaman sesuai dengan otoritas <i>user</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Aktor memasukkan data <i>email</i> kedalam <i>form email</i>. 2. Aktor memasukkan data <i>password</i> kedalam <i>form password</i>. 3. Aktor menekan tombol masuk.
Hasil yang diharapkan	Sistem menampilkan halaman <i>user</i> berdasarkan otoritas <i>user</i> .
Hasil Pengujian	Sistem menampilkan halaman <i>user</i> berdasarkan otoritas <i>user</i> .
Status Validasi	Valid.

Tabel 6.8 Tabel Kasus Uji *login* alternatif : Jika *email* dan *password* tidak di isi.

Nama Kasus Uji	<i>Login</i>
Kode Kebutuhan	SIPP_F_001
Tujuan Pengujian	untuk memastikan jika <i>form</i> dikosongkan maka sistem menampilkan pesan <i>error</i> .
Prosedur Uji	1. Aktor mengosongkan <i>form email</i> atau <i>password</i> 2. Aktor menekan tombol masuk.
Hasil yang diharapkan	sistem menampilkan pesan <i>error</i> pada <i>form</i> yang kosong.
Hasil Pengujian	sistem menampilkan pesan <i>error</i> pada <i>form</i> yang kosong.
Status Validasi	Valid.

Tabel 6.9 Tabel Kasus Uji *email* atau *password* salah

Nama Kasus Uji	<i>Login</i>
Kode Kebutuhan	SIPP_F_001
Tujuan Pengujian	Untuk memastikan jika <i>email</i> atau <i>password</i> yang dimasukan oleh pengguna salah atau tidak sama dengan yang tersimpan pada <i>database</i> maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	1. Aktor memasukkan data <i>email</i> pada <i>form</i> email tidak sesuai. 2. Aktor memasukkan data <i>password</i> pada <i>form password</i> tidak sesuai. 3. Aktor menekan tombol masuk.
Hasil yang diharapkan	Sistem menampilkan pesan <i>error</i> bahwa " <i>Email</i> atau <i>password</i> salah".
Hasil Pengujian	Sistem menampilkan pesan <i>error</i> bahwa " <i>Email</i> atau <i>password</i> salah".
Status Validasi	Valid.

2. Kasus uji lihat profil

Tabel 6.10 Tabel Kasus Uji Lihat Profil

Nama Kasus Uji	Lihat Profil
Kode Kebutuhan	SIPP_F_002

Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan data diri pengguna
Prosedur Uji	Aktor menekan tombol profil pada <i>menu sidebar</i>
Hasil yang diharapkan	Sistem berhasil menampilkan data diri pengguna
Hasil Pengujian	Sistem berhasil menampilkan data diri pengguna
Status Validasi	Valid

3. Kasus uji ubah data profil

Tabel 6.11 Tabel Kasus Uji Ubah Data profil

Nama Kasus Uji	Ubah Data Profil
Kode Kebutuhan	SIPP_F_003
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat melakukan perubahan data diri pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan <i>tab</i> ubah data. 2. Pengguna memasukan data diri yang baru berupa nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, foto. 3. Pengguna menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat memperbarui data diri.
Hasil Pengujian	Sistem dapat memperbarui data diri.
Status Validasi	Valid

Tabel 6.12 Tabel Kasus Uji *form* nama, *username*, *email*, *nip* kosong

Nama Kasus Uji	Ubah Data Profil
Kode Kebutuhan	SIPP_F_003
Tujuan Pengujian	Untuk memastikan jika <i>form</i> nama, <i>username</i> , <i>email</i> , <i>nip</i> dikosongkan maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan <i>tab</i> ubah data. 2. Pengguna tidak memasukan data diri yang baru berupa nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, foto. 3. Pengguna menekan tombol simpan.



Hasil yang diharapkan	Sistem menampilkan pesan <i>error</i> untuk mengisi form yang kosong
Hasil Pengujian	Sistem menampilkan pesan <i>error</i> untuk mengisi <i>form</i> yang kosong.
Status Validasi	Valid

4. Kasus uji lihat daftar notifikasi

Tabel 6.13 Tabel Kasus Uji Lihat Profil

Nama Kasus Uji	Lihat Profil
Kode Kebutuhan	SIPP_F_004
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan daftar notifikasi.
Prosedur Uji	Pengguna menekan tombol notifikasi pada menu <i>sidebar</i>
Hasil yang diharapkan	Sistem menampilkan daftar notifikasi
Hasil Pengujian	Sistem menampilkan daftar notifikasi
Status Validasi	Valid

5. Kasus uji lihat detail notifikasi

Tabel 6.14 Tabel Kasus Uji Lihat Detail Notifikasi

Nama Kasus Uji	Lihat Detail Notifikasi
Kode Kebutuhan	SIPP_F_005
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan detail notifikasi
Prosedur Uji	Pengguna menekan tombol lihat pada halaman daftar notifikasi
Hasil yang diharapkan	Sistem menampilkan detail notifikasi
Hasil Pengujian	Sistem menampilkan detail notifikasi
Status Validasi	Valid

6. Kasus uji buat draft programa

Tabel 6.15 Tabel Kasus Uji Buat Draft Programa

Nama Kasus Uji	Buat <i>Draft</i> Programa
Kode Kebutuhan	SIPP_F_006
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat membuat <i>draft</i> programa.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>draft</i> programa kemudian memilih menu buat draft programa pada <i>sidebar</i>. 2. Pengguna mengisi <i>form</i> terdiri dari <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat perilaku dan <i>non</i> perilaku, jumlah dana yang di butuhkan, <i>form</i> berkas <i>draft</i> programa. 3. Pengguna menekan tombol submit.
Hasil yang diharapkan	Sistem dapat membuat <i>draft</i> programa.
Hasil Pengujian	Sistem dapat membuat <i>draft</i> programa.
Status Validasi	Valid

Tabel 6.16 Tabel Kasus Uji Buat *Draft* Programa : mengisi lebih dari 1000 kata

Nama Kasus Uji	Buat <i>Draft</i> Programa
Kode Kebutuhan	SIPP_F_006
Tujuan Pengujian	Untuk memastikan jika <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat perilaku dan <i>non</i> perilaku diisi dengan kata lebih dari 1000 maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol draft programa kemudian memilih menu buat <i>draft</i> programa pada menu <i>sidebar</i>. 2. pengguna mengisi <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat

	perilaku dan <i>non</i> perilaku dengan mengisi lebih dari 1000 kata pada <i>form</i> .
Hasil yang diharapkan	sistem akan menampilkan pesan <i>error</i> bahwa tidak boleh mengisi lebih dari 1000 kata.
Hasil Pengujian	sistem akan menampilkan pesan <i>error</i> bahwa tidak boleh mengisi lebih dari 1000 kata.
Status Validasi	Valid

Tabel 6.17 Tabel Kasus Uji Buat *Draft* Programa : Mengosongkan salah satu *form*

Nama Kasus Uji	Buat <i>Draft</i> Programa
Kode Kebutuhan	SIPP_F_006
Tujuan Pengujian	Untuk memastikan jika <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat perilaku dan <i>non</i> perilaku dikosongkan maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>draft</i> programa kemudian memilih menu buat draft programa pada menu <i>sidebar</i>. 2. Pengguna mengosongkan <i>form</i> latar belakang, tujuan, manfaat, biofisik, sumber daya manusia, kelembagaan sarana usaha tani, tujuan bersifat perilaku dan <i>non</i> perilaku, masalah bersifat perilaku dan non perilaku. 3. Pengguna menekan tombol <i>submit</i>.
Hasil yang diharapkan	sistem akan menampilkan pesan <i>form</i> tidak boleh kosong pada <i>form</i> yang kosong.
Hasil Pengujian	sistem akan menampilkan pesan <i>form</i> tidak boleh kosong pada <i>form</i> yang kosong.
Status Validasi	Valid

7. Kasus uji masukan jadwal kegiatan

Tabel 6.18 Tabel Kasus Uji Masukan Jadwal Kegiatan

Nama Kasus Uji	Masukan Jadwal Kegiatan
Kode Kebutuhan	SIPP_F_007



Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menyimpan data jadwal kegiatan penyuluhan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna memasukkan data materi penyuluhan, jadwal penyuluhan, daftar pengeluaran dan tempat penyuluhan. 2. pengguna menekan tombol simpan.
Hasil yang diharapkan	Sistem menampilkan daftar <i>draft</i> program.
Hasil Pengujian	Sistem menampilkan daftar <i>draft</i> program.
Status Validasi	Valid

Tabel 6.19 Tabel Kasus Uji Masukan Jadwal Kegiatan dengan mengosongkan salah satu form

Nama Kasus Uji	Masukan Jadwal Kegiatan
Kode Kebutuhan	SIPP_F_007
Tujuan Pengujian	Untuk memastikan sistem dapat menampilkan pesan <i>error</i> apabila pengguna mengosongkan salah satu <i>form</i> materi, tanggal dan tempat.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna mengosongkan data materi penyuluhan, jadwal penyuluhan, dan tempat penyuluhan. 2. Pengguna menekan tombol simpan.
Hasil yang diharapkan	sistem menampilkan pesan <i>error form</i> tidak boleh kosong pada <i>form</i> yang kosong.
Hasil Pengujian	sistem menampilkan pesan <i>error form</i> tidak boleh kosong pada <i>form</i> yang kosong.
Status Validasi	Valid.

8. Kasus uji lihat *draft* program

Tabel 6.20 Tabel Kasus Uji Lihat *Draft* Program

Nama Kasus Uji	Lihat <i>Draft</i> Program
Kode Kebutuhan	SIPP_F_008
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan daftar <i>draft</i> program.
Prosedur Uji	1. pengguna menekan tombol <i>draft</i> program dan memilih menu daftar <i>draft</i> pada <i>sidebar</i>



Hasil yang diharapkan	Sistem dapat menampilkan daftar <i>draft</i> programa.
Hasil Pengujian	Sistem dapat menampilkan daftar <i>draft</i> programa.
Status Validasi	Valid

9. Kasus uji lihat detail *draft* programa

Tabel 6.21 Tabel Kasus Uji Lihat Detail *Draft* Programa

Nama Kasus Uji	Lihat Detail <i>Draft</i> Programa
Kode Kebutuhan	SIPP_F_009
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan detail <i>draft</i> programa.
Prosedur Uji	Pengguna menekan tombol lihat pada halaman daftar <i>draft</i> programa.
Hasil yang diharapkan	Sistem dapat menampilkan detail <i>draft</i> programa.
Hasil Pengujian	Sistem dapat menampilkan detail <i>draft</i> programa.
Status Validasi	Valid

10. Kasus uji verifikasi *draft* programa

Tabel 6.22 Tabel Kasus Uji Verifikasi *Draft* Programa

Nama Kasus Uji	Uji Verifikasi <i>Draft</i> Programa
Kode Kebutuhan	SIPP_F_010
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat memverifikasi <i>draft</i> programa.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>process</i> pada halaman detail <i>draft</i> programa. 2. Aktor menceklis kelayakan yang sudah sesuai, dan mengisi <i>form</i> penjelasan kelayakan. 3. Pengguna menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat memverifikasi <i>draft</i> programa.
Hasil Pengujian	Sistem dapat memverifikasi <i>draft</i> programa.
Status Validasi	Valid

Tabel 6.23 Tabel Kasus Uji satu *form* penjelasan kelayakan di kosongkan

Nama Kasus Uji	satu <i>form</i> penjelasan kelayakan di kosongkan
Kode Kebutuhan	SIPP_F_010
Tujuan Pengujian	Untuk memastikan jika penjelasan kelayakan dikosongkan maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	1. Pengguna tidak mengisi <i>form</i> penjelasan kelayakan 2. Pengguna menekan tombol simpan
Hasil yang diharapkan	Sistem dapat menampilkan pesan <i>error</i> agar <i>form</i> tidak boleh kosong.
Hasil Pengujian	Sistem dapat menampilkan pesan <i>error</i> agar <i>form</i> tidak boleh kosong.
Status Validasi	Valid

11. Kasus uji lihat jadwal dan laporan penyuluhan

Tabel 6.24 Tabel Kasus Uji Lihat Jadwal dan Laporan Penyuluhan

Nama Kasus Uji	Lihat jadwal dan Laporan Penyuluhan
Kode Kebutuhan	SIPP_F_011
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan jadwal kegiatan penyuluhan dan laporan kegiatan penyuluhan.
Prosedur Uji	Aktor memilih menu jadwal dan laporan pada menu <i>sidebar</i> .
Hasil yang diharapkan	Sistem dapat menampilkan jadwal dan laporan kegiatan penyuluhan.
Hasil Pengujian	Sistem dapat menampilkan jadwal dan laporan kegiatan penyuluhan.
Status Validasi	Valid

12. Kasus uji buat laporan penyuluhan

Tabel 6.25 Tabel Kasus Uji Membuat Laporan Penyuluhan

Nama Kasus Uji	Membuat Laporan Penyuluhan
----------------	----------------------------

Kode Kebutuhan	SIPP_F_012
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat membuat laporan penyuluhan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol jadwal dan laporan penyuluhan kemudian memilih menu buat laporan penyuluhan pada <i>sidebar</i>. 2. Pengguna memasukan data pada <i>form</i> konten, berkas laporan, foto kegiatan, foto presensi, dan foto bukti pembayaran yang tersedia . 3. Menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat membuat laporan penyuluhan.
Hasil Pengujian	Sistem dapat membuat laporan penyuluhan.
Status Validasi	Valid

Tabel 6.26 Tabel Kasus Uji Status Belum Dikirim

Nama Kasus Uji	Status Belum Dikirim
Kode Kebutuhan	SIPP_F_012
Tujuan Pengujian	Untuk memastikan jika memilih laporan yang berstatus selain belum dikirm pada saat akan memilih laporan yang akan diinput maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol jadwal dan laporan penyuluhan kemudian memilih menu buat laporan penyuluhan pada menu <i>sidebar</i>. 2. Pengguna menekan tombol pilih pada laporan yang berstatus bukan “belum dikirim”
Hasil yang diharapkan	Sistem dapat menampilkan pesan <i>error</i> bahwa hanya laporan yang berstatus belum dikirim yang hanya dapat dibuat laporan.
Hasil Pengujian	Sistem dapat menampilkan pesan <i>error</i> bahwa hanya laporan yang berstatus belum dikirim yang hanya dapat dibuat laporan.
Status Validasi	Valid

Tabel 6.27 Tabel Kasus Uji *Form Kosong*

Nama Kasus Uji	<i>Form Kosong</i>
Kode Kebutuhan	SIPP_F_012
Tujuan Pengujian	Untuk memastikan jika <i>form</i> materi, jadwal, dan tempat dikosongkan maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol jadwal dan laporan penyuluhan kemudian memilih menu buat laporan penyuluhan pada menu <i>sidebar</i>. 2. Pengguna tidak memasukkan data pada <i>form</i> materi, jadwal, dan tempat. 3. Pengguna menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat menampilkan <i>error</i> bahwa <i>form</i> tidak boleh kosong.
Hasil Pengujian	sistem dapat menampilkan <i>error</i> bahwa <i>form</i> tidak boleh kosong.
Status Validasi	Valid

13. Kasus uji lihat detail laporan penyuluhan

Tabel 6.28 Tabel Kasus Uji Lihat Detail Laporan penyuluhan

Nama Kasus Uji	Lihat Detail Laporan penyuluhan
Kode Kebutuhan	SIPP_F_013
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan detail Laporan Penyuluhan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol lihat pada halaman jadwal dan laporan penyuluhan kemudian memilih menu daftar laporan penyuluhan pada menu <i>sidebar</i>. 2. Pengguna menekan tombol pilih pada jadwal yang akan dibuatkan laporan.
Hasil yang diharapkan	sistem menampilkan detail Laporan Penyuluhan
Hasil Pengujian	sistem menampilkan detail Laporan Penyuluhan
Status Validasi	Valid

14. Kasus uji verifikasi laporan penyuluhan

Tabel 6.29 Kasus Uji Verifikasi Laporan Penyuluhan

Nama Kasus Uji	Verifikasi Laporan Penyuluhan
Kode Kebutuhan	SIPP_F_014
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat memverifikasi laporan penyuluhan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>process</i> pada halaman detail laporan penyuluhan. 2. Pengguna mencentang kelayakan dan mengisi <i>form</i> penjelasan kelayakan 3. Pengguna menekan tombol simpan
Hasil yang diharapkan	Sistem dapat memverifikasi laporan penyuluhan.
Hasil Pengujian	Sistem dapat memverifikasi laporan penyuluhan.
Status Validasi	Valid

Tabel 6.30 Tabel Kasus Uji Form Kosong

Nama Kasus Uji	<i>Form</i> Kosong
Kode Kebutuhan	SIPP_F_014
Tujuan Pengujian	Untuk memastikan jika <i>form</i> penjelasan kelayakan dikosongkan amak sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>process</i> pada halaman detail laporan penyuluhan. 2. Pengguna tidak mengisi <i>form</i> penjelasan kelayakan. 3. Pengguna menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat menampilkan pesan peringatan agar <i>form</i> tidak boleh kosong.
Hasil Pengujian	Sistem dapat menampilkan pesan peringatan agar <i>form</i> tidak boleh kosong.
Status Validasi	Valid

15. Kasus uji ajukan surat tugas

Tabel 6.31 Tabel Kasus Uji Ajukan Surat Tugas

Nama Kasus Uji	Ajukan Surat Tugas
Kode Kebutuhan	SIPP_F_015
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat membuat surat tugas.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan surat tugas dan memilih buat ajukan tugas pada menu <i>sidebar</i>. 2. Pengguna menekan tombol ajukan dan akan tampil <i>popup</i> 3. Pengguna menekan tombol ya pada <i>popup</i> untuk konfirmasi pengajuan surat tugas.
Hasil yang diharapkan	sistem dapat mengajukan surat tugas
Hasil Pengujian	sistem dapat mengajukan surat tugas
Status Validasi	Valid

16. Kasus uji lihat daftar surat tugas

Tabel 6.32 Tabel Kasus Uji Lihat Daftar Surat Tugas

Nama Kasus Uji	Lihat Daftar Surat Tugas
Kode Kebutuhan	SIPP_F_016
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan daftar surat tugas.
Prosedur Uji	Pengguna menekan menu surat tugas pada menu <i>sidebar</i> .
Hasil yang diharapkan	Sistem dapat menampilkan daftar surat tugas.
Hasil Pengujian	sistem dapat menampilkan daftar surat tugas.
Status Validasi	Valid

17. Kasus uji verifikasi surat tugas

Tabel 6.33 Tabel Kasus Uji Verifikasi Surat Tugas

Nama Kasus Uji	Verifikasi Surat Tugas
Kode Kebutuhan	SIPP_F_017
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat memverifikasi surat tugas.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol setuju pada halaman daftar surat tugas dan akan muncul <i>popup</i>. 2. Pengguna memilih ya pada <i>popup</i> untuk konfirmasi menyetujui surat tugas.
Hasil yang diharapkan	Sistem dapat memverifikasi surat tugas.
Hasil Pengujian	Sistem dapat memverifikasi surat tugas.
Status Validasi	Valid

18. Kasus uji terbitkan surat tugas

Tabel 6.34 Tabel Kasus Uji Menerbitkan Surat Tugas

Nama Kasus Uji	Menerbitkan Surat Tugas
Kode Kebutuhan	SIPP_F_018
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menerbitkan surat tugas.
Prosedur Uji	Pengguna menekan tombol terbitkan pada halaman detail surat tugas.
Hasil yang diharapkan	Sistem dapat menerbitkan surat tugas.
Hasil Pengujian	Sistem dapat menerbitkan surat tugas.
Status Validasi	Valid

19. Kasus uji ajukan proposal dana

Tabel 6.35 Tabel Kasus Uji Ajukan Proposal Dana

Nama Kasus Uji	Ajukan Proposal Dana
Kode Kebutuhan	SIPP_F_019
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat mengajukan proposal dana
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan proposal dana dan memilih menu ajukan proposal dana pada menu <i>sidebar</i>. 2. Pengguna menekan tombol ajukan dan sistem akan menampilkan <i>popup</i>. 3. Pengguna menekan tombol ya pada <i>popup</i> untuk konfirmasi pengajuan proposal dana.
Hasil yang diharapkan	Sistem dapat mengajukan proposal dana
Hasil Pengujian	Sistem dapat mengajukan proposal dana
Status Validasi	Valid

20. Kasus uji lihat daftar proposal dana

Tabel 6.36 Tabel Kasus Uji Lihat Daftar Proposal Dana

Nama Kasus Uji	Lihat Daftar Proposal Dana
Kode Kebutuhan	SIPP_F_020
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan daftar proposal dana.
Prosedur Uji	Pengguna memilih proposal dana pada menu <i>sidebar</i> .
Hasil yang diharapkan	sistem menampilkan daftar proposal dana.
Hasil Pengujian	sistem menampilkan daftar proposal dana.
Status Validasi	Valid

21. Kasus uji verifikasi proposal dana

Tabel 6.37 Tabel Kasus Uji Verifikas Proposal Dana

Nama Kasus Uji	Verifikas Proposal Dana
Kode Kebutuhan	SIPP_F_021
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat memverifikasi proposal dana.
Prosedur Uji	1. Pengguna menekan tombol setuju pada halaman daftar proposal dana dan sistem akan menampilkan <i>popup</i> 2. Pengguna menekan tombol ya pada <i>popup</i> konfirmasi setuju proposal dana
Hasil yang diharapkan	sistem dapat memverifikasi proposal dana.
Hasil Pengujian	sistem dapat memverifikasi proposal dana dana.
Status Validasi	Valid

22. Kasus uji lihat daftar pengiriman dana

Tabel 6.38 Tabel Kasus Uji Lihat Daftar Pengiriman Dana

Nama Kasus Uji	Lihat Daftar Pengiriman Dana
Kode Kebutuhan	SIPP_F_022
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan daftar pengiriman dana.
Prosedur Uji	Pengguna menekan menu pengiriman dana pada menu <i>sidebar</i> .
Hasil yang diharapkan	sistem dapat menampilkan daftar pengiriman dana
Hasil Pengujian	sistem dapat menampilkan daftar pengiriman dana
Status Validasi	Valid

23. Kasus uji lihat detail pengiriman dana

Tabel 6.39 Tabel Kasus Uji Lihat Detail Pengiriman Dana

Nama Kasus Uji	Lihat Detail Pengiriman Dana
Kode Kebutuhan	SIPP_F_023

Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan detail pengiriman dana.
Prosedur Uji	Pengguna menekan tombol lihat pada halaman daftar pengiriman dana.
Hasil yang diharapkan	Sistem dapat menampilkan detail pengiriman dana
Hasil Pengujian	sistem dapat menampilkan detail pengiriman dana
Status Validasi	Valid

24. Kasus uji konfirmasi penerimaan dana

Tabel 6.40 Tabel Kasus Uji Konfirmasi Penerimaan Dana

Nama Kasus Uji	konfirmasi penerimaan dana
Kode Kebutuhan	SIPP_F_024
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat konfirmasi penerimaan dana.
Prosedur Uji	<ol style="list-style-type: none"> 1. Pengguna menekan tombol konfirmasi penerimaan dana pada halaman detail pengiriman dana dan sistem akan menampilkan <i>popup</i>. 2. Pengguna menekan tombol Ya pada <i>popup</i> untuk konfirmasi bahwa sudah menerima dana.
Hasil yang diharapkan	Sistem dapat mengkonfirmasi penerimaan dana
Hasil Pengujian	sistem dapat mengkonfirmasi penerimaan dana
Status Validasi	Valid

25. Kasus uji lihat daftar pengguna

Tabel 6.41 Tabel Kasus Uji Lihat Daftar pengguna

Nama Kasus Uji	Lihat Daftar Pengguna
Kode Kebutuhan	SIPP_F_025
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan daftar pengurus.
Prosedur Uji	Pengguna melakukan <i>login</i> .

Hasil yang diharapkan	sistem dapat menampilkan daftar pengguna.
Hasil Pengujian	sistem dapat menampilkan daftar pengguna.
Status Validasi	Valid

26. Kasus uji tambah pengguna

Tabel 6.42 Tabel Kasus Uji Tambah Pengguna

Nama Kasus Uji	Tambah Pengguna
Kode Kebutuhan	SIPP_F_026
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menambah pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Admin menekan tambah pengguna pada halaman daftar pengguna. 2. Admin memasukan data pada <i>form</i> nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, dan foto. 3. Admin menekan tombol simpan.
Hasil yang diharapkan	sistem dapat menambah pengguna.
Hasil Pengujian	sistem dapat menambah pengguna.
Status Validasi	Valid

Tabel 6.43 Kasus Uji Tambah Pengguna *alternative* : *Form* tidak lengkap

Nama Kasus Uji	<i>Form</i> Tidak Lengkap
Kode Kebutuhan	SIPP_F_026
Tujuan Pengujian	Untuk memastikan jika <i>form</i> nama, <i>username</i> , <i>email</i> , <i>nip</i> , <i>password</i> , dan foto dikosongkan maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Admin menekan tambah pengguna pada halaman daftar pengguna. 2. Admin tidak memasukan data pada <i>form</i> nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, dan foto. 3. Admin menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat menampilkan pesan peringatan <i>form</i> tidak boleh kosong.

Hasil Pengujian	Sistem dapat menampilkan pesan peringatan <i>form</i> tidak boleh kosong.
Status Validasi	Valid

27. Kasus uji lihat data pengguna

Tabel 6.44 Tabel Kasus Uji Lihat Data Pengguna

Nama Kasus Uji	Lihat Data Pengguna
Kode Kebutuhan	SIPP_F_07
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menampilkan data pengguna.
Prosedur Uji	Admin menekan tombol lihat pada halaman daftar pengguna.
Hasil yang diharapkan	Sistem dapat menampilkan data pengguna.
Hasil Pengujian	Sistem dapat menampilkan data pengguna.
Status Validasi	Valid

28. Kasus uji *edit* data pengguna

Tabel 6.45 Tabel Kasus Uji *Edit* pengguna

Nama Kasus Uji	<i>Edit</i> Pengguna
Kode Kebutuhan	SIPP_F_028
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat mengedit data pengguna.
Prosedur Uji	<ol style="list-style-type: none"> 1. Admin menekan tombol <i>edit</i> pada halaman data pengguna. 2. Admin memasukan data pada <i>form</i> nama, <i>username</i>, <i>email</i>, <i>nip</i>, <i>password</i>, dan foto. 3. Admin menekan tombol simpan.
Hasil yang diharapkan	Data pengguna berhasil diubah.
Hasil Pengujian	Data pengguna berhasil diubah.
Status Validasi	Valid

Tabel 6.46 Tabel Kasus Uji Edit Pengguna *Alternative* : Form Tidak Lengkap

Nama Kasus Uji	Form Tidak Lengkap
Kode Kebutuhan	SIPP_F_028
Tujuan Pengujian	Untuk memastikan jika <i>form</i> nama, <i>username</i> , <i>email</i> , <i>nip</i> , <i>password</i> , dan foto dikosongkan maka sistem akan menampilkan pesan <i>error</i> .
Prosedur Uji	1. Admin menekan tombol <i>edit</i> pada halaman data pengguna. 2. Admin tidak memasukan data pada <i>form</i> nama, <i>username</i> , <i>email</i> , <i>nip</i> , <i>password</i> , dan foto. 3. Admin menekan tombol simpan.
Hasil yang diharapkan	Sistem dapat menampilkan pesan peringatan <i>form</i> tidak boleh kosong.
Hasil Pengujian	Sistem dapat menampilkan pesan peringatan <i>form</i> tidak boleh kosong.
Status Validasi	Valid

29. Kasus uji hapus pengguna

Tabel 6.47 Tabel Kasus Uji Hapus Pengguna

Nama Kasus Uji	Hapus Pengguna
Kode Kebutuhan	SIPP_F_029
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menghapus pengguna.
Prosedur Uji	1. Admin menekan tombol hapus pada halaman data pengguna dan sistem akan menampilkan <i>popup</i> . 2. Admin memilih pilihan ya pada <i>popup</i> untuk konfirmasi menghapus pengguna.
Hasil yang diharapkan	Data pengguna berhasil dihapus.
Hasil Pengujian	Data pengguna berhasil dihapus.
Status Validasi	Valid

30. Kasus uji *logout*

Tabel 6.48 Tabel Kasus Uji Logout

Nama Kasus Uji	<i>Logout</i>
Kode Kebutuhan	SIPP_F_030
Tujuan Pengujian	Untuk memastikan bahwa pengguna dapat melakukan <i>logout</i> dan keluar dari halaman otoritas pengguna.
Prosedur Uji	Pengguna menekan tombol <i>logout</i> pada <i>navbar</i> .
Hasil yang diharapkan	Pengguna berhasil <i>logout</i> dan dari halaman otoritas pengguna.
Hasil Pengujian	Pengguna berhasil <i>logout</i> dan dari halaman otoritas pengguna.
Status Validasi	Valid

6.3 Compatibility Testing

Compatibility testing berguna untuk menentukan kompatibilitas sistem yang dibangun untuk menampilkan halaman pada beberapa *browser* yang berbeda. Uji kompatibilitas ini menggunakan sebuah *software* yaitu *software sortsite*. Cara kerja *sortsite* yaitu dengan mengakses *website* yang dibangun dan *software* akan melakukan pengujian dengan menjalankan pada delapan browser. Untuk prosedur dan hasil yang lebih jelas dapat dilihat pada Tabel 6.49 dan Gambar 6.4.

Tabel 6.49 Compatibility Testing

Kode Kebutuhan	SIPP_NF_001
Langkah pengujian	<i>Software sistem</i> akan mencoba menjalankan sistem yang dibangun pada <i>browser internet explorer(IE)</i> , <i>browser Edge</i> , <i>browser Firefox</i> , <i>browser Safari</i> , <i>browser Opera</i> , <i>browser Chrome</i> , <i>browser iOS</i> , dan <i>browser Android</i> .
Hasil yang diharapkan	Sistem dapat berjalan pada delapan browser dengan baik.
Hasil	Setelah dilakukan maka didapatkan hasil bahwa sistem dapat berjalan pada delapan browser dengan baik.
Status	Valid

Errors	Accessibility	Compatibility	Search	Standards	Usability							
This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.												
Browser	IE	Edge	Firefox	Safari	Opera	Chrome	iOS		Android		Key	
Version	11	18	66	12	60	74	≤ 10	11	12	≤ 3		4*
Critical Issues	✓	✓	●	✓	✓	●	✓	✓	✓	✓	✓	● Missing content or functionality
Major Issues	●	✓		●	✓		✓	✓	✓	✓	✓	● Major layout or performance problems
Minor Issues	●	✓	●	●	✓		✓	✓	✓	✓	✓	● Minor layout or performance problems

* Most Android devices from 4.4 onwards use Chrome as the default browser, older versions use the original Android stock browser

Gambar 6.4 Hasil *Compatibility Testing*

Hasil Pengujian kompatibilitas dapat dilihat pada Gambar 6.4. Sistem ini dapat berjalan pada 8 browser, 6 browser diantaranya browser *IE*, *edge*, *safari*, *opera*, *ios*, dan *android*. terdapat dua *issue* yaitu ketika dijalankan pada browser *firefox* dan *chrome*, hal ini dikarenakan sistem tidak menggunakan *https* sehingga saat melakukan *login* data *email* dan *password* tidak akan di enkripsi. *Major issue* dan *minor issue* terjadi karena browser tidak support dengan *CSS* yang digunakan. Tetapi *issue* yang muncul tidak berpengaruh pada fungsi sistem karena tetap dapat berjalan dengan baik.



BAB 7 PENUTUP

Pada bagian ini membahas hasil penelitian yang telah dilakukan dan saran untuk kemajuan sistem informasi penyuluhan di dinas pertanian dan ketahanan pangan kabupaten Padang pariaman.

7.1 Kesimpulan

Penelitian yang telah dilakukan dengan berbagai tahapan seperti yang telah dijabarkan pada bab-bab sebelumnya dapat disimpulkan bahwa:

1. Berdasarkan analisis kebutuhan yang sudah dilakukan, sistem informasi penyuluhan pertanian mempunyai 1 kebutuhan *non* fungsional dan 31 fungsionalitas yang dapat mempermudah kegiatan penyuluhan pertanian, dan terdapat fitur utama yaitu mengelola *draft* programa dari pengumpulan hingga pengesahan *draft* programa secara *online*, mengelola laporan penyuluhan dari pengumpulan hingga pemeriksaan laporan penyuluhan secara *online*, dan mengelola proposal dana dari pengumpulan, pemeriksaan, verifikasi pengiriman sampai konfirmasi penerimaan dana. Semua fitur dapat diakses oleh pengguna sesuai haknya oleh 5 aktor, yakni penyuluh, kepala BPP, pegawai dinas, dan kepala dinas. Kebutuhan fungsional yang sudah didapatkan dapat juga dibuatkan model dengan *use case scenario* dan *use case diagram*.
2. Perancangan sistem yang dilakukan berdasarkan hasil analisis kebutuhan menghasilkan perancangan *sequence diagram*, *class diagram*, basis data, algoritme, dan antarmuka. *Sequence diagram* dibuat dari *use case scenario* dengan memilih 3 fungsi utama. Pada perancangan *class diagram* komponen atribut dan fungsi merujuk pada *sequence diagram*. Perancangan basis data menggunakan konsep ERD yang menggambarkan hubungan antar entitas dan atribut masing-masing entitas. Dan perancangan antarmuka beris *mockup* mengenai tampilan yang dari sistem, dan untuk *mockup* menggunakan 3 fungsional utama.
3. Implementasi yang dilakukan berdasarkan hasil perancangan sistem menghasilkan implementasi basis data, algoritme, dan antarmuka. Implementasi basis data menggunakan *MySQL server*, implementasi algoritme menggunakan Bahasa *PHP* dan menggunakan *framework Laravel*. Dan antarmuka menggunakan *html*, *css* dan *javascript*.
4. Pengujian untuk sistem informasi yang telah selesai diimplementasikan menggunakan *whitebox testing*, *blackbox testing* dan *compatibility testing*. Kebutuhan fungsional pengguna telah sesuai dengan hasil implementasi yang dapat diketahui dari hasil *whitebox testing* dan *blackbox testing* yang menghasilkan hasil uji sempurna pada seluruh kasus uji. Kebutuhan *non*-fungsional dapat terpenuhi pada aspek *compatibility* di 8 browser yang dapat diketahui dari hasil *compatibility testing*.

7.2 Saran

Saran yang sekiranya dapat membantu bagi siapa saja yang ingin melakukan penelitian berkaitan dengan sistem informasi penyuluhan pertanian dan ketahanan pangan kabupaten padang pariaman selanjutnya antara lain:

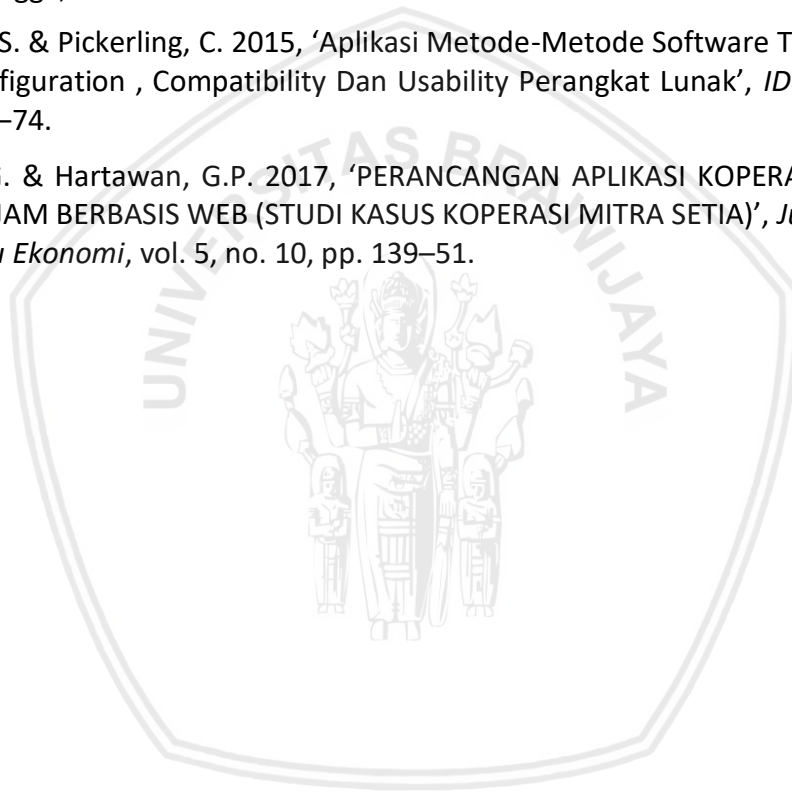
1. Sistem mampu diimplementasikan kedalam perangkat *mobile* melalui metode pengembangan *native* maupun *hybrid*.
2. Penambahan fungsionalitas yang dapat membuat sistem memberi penilaian terhadap penyuluh dari laporan yang sudah di tentukan dengan bantuan dari konsep kecerdasan buatan.
3. Penambahan fungsionalitas yang dapat membuat sistem menentukan potensi wilayah dari laporan yang sudah di kumpulkan dengan bantuan dari konsep kecerdasan buatan



DAFTAR REFERENSI

- Adjid, D. 2001, *Penyuluhan Pertanian*, Yayasan Sinar Tani, Jakarta.
- Agus, M. 2016, 'Pengujian Sistem Informasi Akademik menggunakan McCall Software Quality Framework', *Jurnal Informatika Sunan Kalijaga*, vol. 1, no. 1, pp. 47–57.
- Aisyah, D.A. & Falgenti, K. 2017, 'Sistem Informasi Penjualan Berbasis Kinerja pada Proyek Apartemen Mega City Bekasi', *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 3, no. 3, pp. 343–52.
- Astuti, P. 2018, 'Penggunaan Metode Black Box Testing (Boundary Value Analysis) Pada Sistem Akademik (Sma/Smk)', *Faktor Exacta*, vol. 11, no. 2, p. 186.
- Doro, E. & Betshani, S. 2009, 'Analisis Data Dengan Menggunakan ERD dan Model Konseptual Data Warehouse', *Jurnal informatika*, vol. 5, no. 71, pp. 71–85.
- Hajar, S. 2017, 'Rancang bangun sistem monitoring kegiatan kelompok tani berbasis web di kecamatan pattallassang kabupaten gowa sulawesi selatan', *Sistem Informasi Fakultas Sains dan Teknologi UIN Alauddin Makassar*.
- Hasibuan, A.M. 2016, *Peran Penyuluh Pertanian terhadap Kelompok Tani dalam Pengembangan Usahatani Padi Sawah*, pp. 1–63.
- Hendini, A. 2016, 'Pemodelan UML sistem informasi Monitoring Penjualan dan stok barang', *JURNAL KHATULISTIWA INFORMATIKA*, vol. IV, no. 2, pp. 107–16.
- Hermawan, R., Hidayat, A. & Utomo, V.G. 2016, 'Sistem Informasi Penjadwalan Kegiatan Belajar Mengajar Berbasis Web', *Indonesian Journal on Software Engineering Sistem*, vol. 2, no. 1, pp. 31–8.
- Isnardi 2016, 'Monitoring Bus Trans Padang Berbasis Web', *Jurnal J-Click*, vol. 3, no. 2, pp. 32–7.
- Iswandy, E. 2015, *SISTEM PENUNJANG KEPUTUSAN UNTUK MENENTUKAN PENERIMAAN MAHASISWA DAN PELAJAR KURANG MAMPU*, vol. 3, no. 2.
- Mualim, W. & Putra, G.U. 2017, *IMPLEMENTASI FRAMEWORK MVC PADA SISTEM INFORMASI*, vol. 9, no. 2.
- Mustaqbal, M.S., Firdaus., R.F. & Rahmadi, H. 2015, 'Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis', *Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis*, vol. I, no. 3, p. 34.
- Pare, S. 2013, 'Desain Dan Implementasi E-Commerce Pada Toko As 88 Celluler Merauke', *Mustek*, vol. 2, no. ISSN 2089-6697, pp. 222–9.
- Pressman, R.S. 2010, *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*, Andi, Yogyakarta.
- Rosa & M. Salahuddin 2014, *Rekayasa perangkat lunak terstruktur dan berorientasi objek*, Informatika, Bandung.

- Saputra, H., Pradana, F. & Priyambadha, B. 2017, 'Pengembangan Elektronik-Sistem Akuntabilitas Kinerja Instansi Pemerintah untuk Dinas Perindustrian Kota Malang', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, vol. 1, no. 10.
- Shopyan, N. 2017, 'Rancang Bangun Sistem Informasi Manajemen Pemberdayaan Kelompok Tani Pada Dinas Pertanian Kabupaten Barru', universitas islam alauddin makassar.
- Siyoksih, Y.H., Nur, H. & Tou, H.J. 2018, *STUDI KOMODITI PERTANIAN UNGGULAN DI KABUPATEN PADANG PARIAMAN*.
- Sommerville, L. 2011, *Software Engineering (Rekayasa Perangkat Lunak)*, Erlangga, Jakarta.
- Tjandra, S. & Pickerling, C. 2015, 'Aplikasi Metode-Metode Software Testing Pada Configuration , Compatibility Dan Usability Perangkat Lunak', *IDeaTech*, pp. 367–74.
- Tofik, I.G. & Hartawan, G.P. 2017, 'PERANCANGAN APLIKASI KOPERASI SIMPAN PINJAM BERBASIS WEB (STUDI KASUS KOPERASI MITRA SETIA)', *Jurnal Ilmiah Ilmu Ekonomi*, vol. 5, no. 10, pp. 139–51.



LAMPIRAN A HASIL WAWANCARA

a. Wawancara dengan kepala seksi penyuluhan

Transkrip wawancara Penelitian dengan narasumber yaitu Kasi (Kepala Seksi) Penyuluhan Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman

Nama : Irawati febriani,SP,M.Sc

Jabatan : Kepala Seksi Penyuluhan

Hari/Tanggal : Rabu/13 Februari 2019

Pukul : 13.00 WIB

Tempat : Kantor Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman

Peneliti	Siapa saja yang terlibat dalam kegiatan penyuluhan ini ?
Narasumber	Penyuluh lapangan, kepala BPP(Badan Penyuluhan Pertanian), pegawai dinas (kepala seksi penyuluhan), dan kepala dinas.
Peneliti	Apa peran masing-masing pelaku dalam kegiatan penyuluhan ini ?
Narasumber	Penyuluh lapangan yaitu orang yang bertugas memberikan pengetahuan tentang penyuluhan terhadap masyarakat, kepala BPP sebagai koordinator penyuluh di tiap kecamatan, pegawai dinas sebagai pengelola penyuluhan di tingkat kabupaten, dan kepala dinas adalah orang yang mengepalai dinas pertanian ini.
Peneliti	Berapa jumlah penyuluh dan kepala BPP di Kabupaten Padang pariaman?
Narasumber	Satu penyuluh bekerja di satu desa, satu kepala BPP bekerja di satu kecamatan. kabupaten Padang Pariaman memiliki 73 desa dan 17 kecamatan, sehingga terdapat 73 penyuluh dan 17 kepala BPP.
Peneliti	Bagaimana alur kegiatan penyuluhan?
Narasumber	Langkah pertama yaitu penyusunan rancangan kerja atau <i>draft</i> program yaitu dengan di adakan rapat di masing-masing desa Bersama perangkat desa, kepala BPP, penyuluh lapangan dan kelompok tani. Selanjutnya penyuluh lapangan melakukan pengumpulan data wilayah kerja, dan evaluasi kerja tahun sebelumnya, selanjutnya diadakan kembali rapat Bersama dengan perangkat desa, penyuluh, dan kepala BPP terkait data yang diperoleh. Selanjutnya mengidentifikasi masalah yang ada.

	Penyusunan dan pengumpulan <i>draft</i> program. <i>Draft</i> program yang dikumpulkan oleh penyuluh akan diperiksa oleh kepala BPP. Jika sudah benar maka semua <i>draft</i> program dalam satu kecamatan akan dikirimkan kepada pegawai dinas, pegawai ginas melakukan pemeriksaan dan pengumpulan kembali. Jika <i>draft</i> yang diumpulkan salah maka akan dikembalikan kepada masing-masing bpp dan direvisi kembali. Jika sudah benar maka akan di buat surat tugas yang harus di tanda tangani oleh kepala dinas dan surat tugas di terbitkan. Kemudian penyuluh dapat melakukan kegiatan sesuai <i>draft</i> program yang sudah dirancang. Setelah kegiatan penyuluh akan mengumpulkan laporan sebagai hasil kegiatan dan laporan pertanggung jawaban pengeluaran dana yang sudah dikirimkan oleh dinas kepada penyuluh untuk kegiatan penyuluhan sesuai rancangan <i>draft</i> prgorama.
Peneliti	Untuk <i>draft</i> program, apa saja unsur yang harus di buat pada <i>draft</i> program.
Narasumber	Unsur yang harus ada dalam <i>draft</i> prgorama yaitu yang pertama harus ada nya pendahuluan seperti latar belakang, tujuan, dan manfaat. Selanjutnya harus ada nya data tentang keadaan daerah meliputi biofisik, sumber daya manusia, kelembagaan usaha tani, selanjutnya harus ada tujuan, dan permasalahan, serta adanya rancangan kegiatan yang berhubungan dengan kegiatan yang akan dilaksanakan satu tahun kedepan.
Peneliti	Untuk <i>draft</i> program, apa saja poin dalam pemeriksaan program?
Narasumber	Akan di periksa berdasarkan poin-poin penting seperti pendahuluan, keadaan daerah yang akan di adakan kegiatan, tujuan, masalah, berkas yang di <i>upload</i> , dan juga dana yang akan dipakai beserta detail pengeluarannya.
Peneliti	Kegiatan biasanya dilakukan berapa kali?
Narasumber	Biasa nya 2-3 kali dalam seminggu tetapi itu kembali lagi ke masing-masing penyuluh dan kelompok tani, karena jadwal penyuluhan sudah di musyawarahkan oleh kelompok tani dengan penyuluh pada saat penyusunan <i>draft</i> program. Jadi jadwal sudah masuk kedalam rancangan <i>draft</i> program.
Peneliti	Apa poin yang dilihat pada pengumpulan laporan ?
Narasumber	Poin yang terpenting yaitu rincian pengeluaran dana sebagai pertanggung jawaban pemakaian dana yang bisa dilihat pada foto bukti pengeluaran dana yang dapat dilihat pada laporan

	penyuluhan. Selanjutnya akan di <i>check</i> foto kegiatan, foto presensi untuk memastikan bahwa penyuluhan benar ada nya.
Peneliti	Laporan yang sudah melewati dua pemeriksaan dan sudah diterima akan diapakan?
Narasumber	Laporan yang dikumpulkan akan direkap dan akan dibahas ketika ada acara evaluasi kegiatan.
Peneliti	Bagaimana alur pengiriman dana kegiatan?
Narasumber	Setelah <i>draft</i> programa disetujui, maka pegawai dinas akan membuat sebuah proposal dana untuk pencairan dana yang harus disetujui oleh kepala dinas, setelah disetujui maka dana akan dikirimkan. Pengiriman dana akan dilakukan 4 tahap. yaitu per 25% dana yang diajukan dalam <i>draft</i> programa.
Peneliti	Setelah pengiriman dana tahap 1, maka kapan pengiriman dana tahap 2 dilakukan?
Narasumber	Setelah dana tahap 1 selesai digunakan yang tertera pada laporan kegiatan.
Peneliti	Bagaimana alur pengumpulan laporan kegiatan ?
Narasumber	Pengumpulan laporan masih dilakukan secara manual, laporan penyuluhan kegiatan dibuat saat kegiatan selesai. Setelah selesai maka laporan akan dikumpulkan kepada kepala BPP di kecamatan wilayah kerja. Kepala BPP akan melakukan pemeriksaan. Jika sudah benar maka laporan akan dikumpulkan ke kantor dinas pertanian dan diperiksa kembali oleh pegawai dinas. Jika salah maka akan dikembalikan.

b. Wawancara dengan penyuluh lapangan

Transkrip wawancara Penelitian dengan narasumber yaitu Kasi (Kepala Seksi) Penyuluhan Dinas Pertanian dan Ketahanan Pangan Kabupaten Padang Pariaman

Nama : Yusrizal,S.ST
 Jabatan : Penyuluh Lapangan
 Hari/Tanggal : Sabtu/16 Februari 2019
 Pukul : 17.00 WIB
 Tempat : Rumah bapak yusrizal

Peneliti	Bagaimana alur kegiatan penyuluhan ?
----------	--------------------------------------



Narasumber	Penyuluhan biasanya dilakukan jam 10 pagi, di tempat yang sudah ditentukan, kegiatan bisa berupa materi atau langsung praktek, biasanya kalau materi kegiatan berlangsung sekitar 2jam, kalau praktek kegiatan berlangsung 3jam atau lebih dan kegiatan dimulai lebih pagi. Setelah kegiatan selesai maka penyuluh akan membuat laporan hasil kegiatan.
Peneliti	Biasanya berapa kelompok tani yang menghadiri?
Narasumber	Ada 3-5 kelompok tani, 1 kelompok tani biasanya berjumlah 5 orang, tetapi biasanya tidak semua anggota yang hadir.
Peneliti	Apa saja kendala yang ditemui dalam kegiatan penyuluhan?
Narasumber	<ol style="list-style-type: none"> 1. Dalam pengumpulan <i>draft</i> program, harus melewati proses yang rumit, yaitu harus melalui pemeriksaan oleh kepala BPP, jika <i>draft</i> program tidak <i>valid</i>, maka pembuatan dan pencetakan <i>draft</i> program akan diulang, ketika <i>valid</i> juga akan diperiksa oleh pegawai dinas, jika hasil pemeriksaan dari pegawai dinas juga tidak <i>valid</i> maka <i>draft</i> program akan dibuat ulang dan dicetak ulang. 2. Kendala dalam pengumpulan laporan penyuluhan dan pengeluaran dana, juga sama halnya dengan <i>draft</i> program yang harus melewati dua tahap pemeriksaan yang ketika tidak <i>valid</i> maka akan dilakukan pembuatan dan pencetakan ulang.
Peneliti	Apakah sebelumnya sudah ada sistem tentang kegiatan penyuluhan ini?
Narasumber	Belum ada.
Peneliti	Jika dibuatkan sebuah sistem yang nantinya akan mempermudah bapak dalam pengumpulan <i>draft</i> dan laporan, apakah bapak setuju?
Narasumber	Setuju saja, jika penggunaan sistem tidak mempersulit kami.