



PEMBANGUNAN SISTEM INFORMASI MANAJEMEN ASET PADA DINAS PERIKANAN KABUPATEN TABANAN

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

I Made Jeri Permana Adi

NIM: 155150201111190



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2019

PENGESAHAN

PEMBANGUNAN SISTEM INFORMASI MANAJEMEN ASET PADA DINAS PERIKANAN KABUPATEN TABANAN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
I Made Jeri Permana Adi
NIM: 155150201111190

Skrripsi ini telah diuji dan dinyatakan lulus pada
18 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Denny Sagita Rusdianto, S.Kom, M.Kom
NIP: 19851124 201504 1 001

Dosen Pembimbing 2



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

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kumilawan, 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, 31 Juli 2019



I Made Jeri Permana Adi

NIM: 155150201111190



PRAKATA

Segala puji saya panjatkan Kepada Tuhan Yang Maha Esa karena berkat anugrah dan petunjuk-Nya saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Komputer (S.Kom) pada Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang. Judul yang penulis ajukan adalah “Pembangunan Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan”.

Dalam kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak-pihak yang telah membantuk penulis selama penyusunan skripsi ini, diantaranya:

1. Bapak Denny Sagita Rusdianto, S.Kom, M.Kom selaku dosen pembimbing satu yang telah membimbing serta memberikan motivasi kepada penulis.
2. Ibu Faizatul Amalia, S.Pd., M.Pd selaku dosen pembimbing dua yang telah membimbing serta memberikan motivasi kepada penulis.
3. Bapak Wayan Firdaus Mahmudy, S.S.i., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
5. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi teknik Informatika.
6. Kepada staf Dinas Perikanan kabupaten Tabanan yang telah membantu penulis dalam penelitian ini.
7. Bapak I Nyoman Subagia, Ni Wayan Sumiasih, I Putu Andrea Renaldi dan Ni Made Putri Dwijayanti, S.Pd selaku orang tua dan kakak kandung penulis yang selalu memberi semangat dan doa hingga terselesaikannya skripsi ini.
8. Bapak I Nyoman Punia Yasa selaku paman yang memberikan motivasi kepada penulis.
9. Kepada kontrakan neptunus 12, sahabat serta orang terdekat penulis yang selalu membantu, menghibur serta memberikan motivasi kepada penulis.
10. Semua pihak yang telah membantu terselesainya skripsi ini yang tidak dapat penulis sebut satu persatu.

Dalam pembuatan skripsi ini, penulis menyadari bahwa masih terdapat kekurangan dikarenakan keterbatasan dan kemampuan yang penulis miliki. Demi Kesempurnaan 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, 23 Juli 2019

Penulis

jeripermana.made@gmail.com



ABSTRAK

I Made Jeri Permana Adi, Pembangunan Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan

Pembimbing: Denny Sagita Rusdianto, S.Kom, M.Kom dan Faizatul Amalia, S.Pd., M.Pd.

Aset berperan dalam kegiatan operasional pada Dinas Perikanan Kabupaten Tabanan, sehingga diperlukannya pengelolaan aset yang baik agar tidak menghambat kegiatan operasional. Kegiatan pelaporan aset dilakukan setiap satu semester sekali, dalam pelaporan aset sering terjadi ketidaksesuaian data, hal ini terjadi karena tidak adanya tempat yang terpusat untuk menyimpan semua data aset. Maka dari itu diperlukannya sebuah sistem informasi yang terpusat untuk menyimpan semua aset agar lebih mudah dalam melakukan pengelolaan. Pembangunan sistem informasi menggunakan model *waterfall* dan diimplementasikan menggunakan web. Tahapan analisis menghasilkan 56 persyaratan fungsional dan 1 persyaratan non-fungsional. Tahapan perancangan sistem terdiri 41 *class diagram*, 9 kelas merupakan turunan dari *Controller* dan 19 turunan dari *Model*. Selain itu hasil perancangan juga menghasilkan pemodelan interaksi objek yang dimodelkan dalam *sequence diagram*, *Physical Data Model* yang berisikan tabel-tabel sebagai tempat untuk penyimpanan data, beberapa perancangan komponen dan beberapa perancangan antarmuka pengguna. Tahap implementasi menggunakan *framework* Laravel 5.7 dengan menggunakan bahasa pemrograman PHP, HTML, CSS, dan juga *Javascript* serta memakai pola perancangan *Model-View-Controller*. Implementasi *database* menggunakan *MySQL* dan *web server* menggunakan *Apache*. Pengujian unit dilakukan menggunakan 3 sampel uji dengan metode *basis path testing* yang menghasilkan hasil bernilai 100% valid. Pengujian validasi mendapatkan nilai 100% valid dari semua kasus uji berdasarkan kebutuhan fungsional. Pengujian *compatibility* mendapatkan hasil 75% dengan sistem berjalan pada *Critical issue* pada firefox versi 66 dan Chrome versi 74. Berjalan dengan *major issues* pada semua *platform* serta berjalan dengan *minor issues* pada *IE* versi 11, *firefox* versi 66 dan *safari* pada versi 12. Pengujian integrasi menggunakan *integration incremental top-down* yang mendapatkan nilai valid dari 1 sampel uji.

Kata kunci: Sistem Informasi, Sistem Informasi Manajemen Aset, Laravel, web



ABSTRACT

I Made Jeri Permana Adi, Pembangunan Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan

Supervisors: Denny Sagita Rusdianto, S.Kom, M.Kom dan Faizatul Amalia, S.Pd., M.Pd.

Assets have an important role in operational activities at the Dinas Perikanan Kabupaten Tabanan. It means that a good assets management is needed so it will not inhibit the operational activities. Assets reporting activities are conducted once every semester. In the reporting activities, data discrepancy is often occurred because there is no centralized place to store all the assets information. Therefore, a centralized information system to record all assets is needed to make them easier to manage. The information system is developed using waterfall model and implemented as a webservice. At the requirements analysis phase, 56 functional requirements and 1 non-functional requirements are obtained. The system design phase consists of 41 classes diagrams, with 9 classes are derived from the controller and 16 classes are derived from the model. In addition, the design phase also produce the interaction of the objects modeled in sequence diagrams, physical data model that contains tables to store the data, some component design and some user interface design. The implementation phase uses Laravel 5.7 and Bootstrap framework using PHP, HTML, CSS, and Javascript with the model-view-controller pattern. Databases are implemented using MySQL and Apache web server. Unit testing is carried out using 3 test samples with a base path testing method with 100% valid result. Validation testing is carried out in all the functional requirements along with the alternatives with 100% valid result. Compatibility testing is also carried with 75% result that showed the system is running with critical issue on Firefox version 66 and Chrome version 74 and with major Issues on all platforms. While on IE version 11, Firefox version 66 and Safari version 12 the system runs with minor issues. Integration testing also conducted using integration incremental top-down with 100% result form 1 test sample.

Keywords: Information System, Assets Management Information System, Laravel, web



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xix
DAFTAR LAMPIRAN	xxi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dinas Perikanan Kabupaten Tabanan	5
2.2.1 Visi dan Misi	5
2.2.2 Struktur Organisasi.....	6
2.3 Sistem Informasi Manajemen Aset.....	8
2.3.1 Sistem Informasi.....	8
2.3.2 Manajemen Aset.....	8
2.3.3 Sistem Informasi Manajemen Aset.....	8
2.4 Rekayasa Perangkat Lunak	9
2.5 <i>System Development Life Cycle (SDLC)</i>	9
2.5.1 <i>Waterfall Model</i>	9
2.6 Kerangka Kerja (<i>Framework</i>)	10
2.7 <i>MVC (Model-View-Controller)</i>	12



2.8 MySQL	12
2.9 Proses Bisnis.....	12
2.10 Business Process Modeling and Notation (BPMN)	13
2.11 Unified Modeling Language (UML).....	13
2.11.1 Use Case Diagram	13
2.11.2 Sequence Diagram	14
2.11.3 Class Diagram.....	15
2.12 Pengujian Perangkat Lunak.....	15
2.12.1 White Box Testing	16
2.12.2 Black Box Testing.....	16
BAB 3 METODOLOGI PENELITIAN	17
3.1 Studi Literatur	17
3.2 Rekayasa Kebutuhan.....	18
3.3 Perancangan Sistem.....	19
3.4 Implementasi	19
3.5 Pengujian dan Analisis	19
3.6 Kesimpulan dan Saran	19
BAB 4 REKAYASA KEBUTUHAN.....	20
4.1 Elisitasi dan Analisis Kebutuhan.....	20
4.1.1 Elisitasi Kebutuhan	20
4.1.2 Analisis Kebutuhan.....	28
4.2 Gambaran Umum Sistem.....	36
4.3 Identifikasi Aktor.....	36
4.4 Kebutuhan Fungsional Sistem	37
4.5 Pemodelan Kebutuhan	43
4.5.1 Use Case Diagram	43
4.5.2 Use Case Scenario	43
BAB 5 PERANCANGAN DAN IMPLEMENTASI	89
5.1 Perancangan Sistem.....	89
5.1.1 Perancangan Sequence Diagram	89
5.1.2 Perancangan Class Diagram.....	90
5.1.3 Perancangan Database	92



5.1.4 Perancangan Antarmuka.....	92
5.2 Spesifikasi Sistem.....	98
5.2.1 Perancangan Komponen.....	98
5.2.2 Spesifikasi <i>Hardware</i>	105
5.2.3 Spesifikasi <i>Software</i>	105
5.3 Implementasi Sistem	106
5.3.1 Implementasi Komponen.....	106
5.3.2 Implementasi <i>Database</i>	123
5.3.3 Implementasi Antarmuka	123
BAB 6 PENGUJIAN DAN ANALISIS.....	127
6.1 Pengujian Unit.....	127
6.1.1 Pengujian Fungsi <i>create</i> pada <i>Class</i> AsetController.....	127
6.1.2 Pengujian Fungsi <i>requestReportKadis</i> pada <i>Class</i> AsetController	137
6.1.3 Pengujian Fungsi <i>create</i> Pada <i>Class</i> PemegangKendaraanController.....	140
6.2 Pengujian Integrasi	142
6.3 Pengujian <i>Compatibility</i>	147
6.4 Pengujian Validasi	147
6.4.1 Pengujian Validasi Mendaftarkan Pengguna	147
6.4.2 Pengujian Validasi <i>Login</i>	148
6.4.3 Pengujian Validasi Daftar Pengguna	149
6.4.4 Pengujian Validasi Sunting Pengguna	149
6.4.5 Pengujian Validasi Hapus Pengguna	150
6.4.6 Pengujian Validasi Daftar Aset	151
6.4.7 Pengujian Validasi Tambah Aset	151
6.4.8 Pengujian Validasi Sunting Aset.....	152
6.4.9 Pengujian Validasi Hapus Aset	153
6.4.10 Pengujian Validasi Permintaan Laporan Aset	153
6.4.11 Pengujian Validasi Daftar Laporan Aset.....	154
6.4.12 Pengujian Validasi Sunting Laporan Aset.....	154
6.4.13 Pengujian Validasi Laporan Aset	155



6.4.14 Pengujian Validasi Daftar Validasi Aset.....	156
6.4.15 Pengujian Validasi Validasi Laporan Aset.....	156
6.4.16 Pengujian Validasi <i>Export</i> Laporan Aset.....	158
6.4.17 Pengujian Validasi Daftar Persediaan	159
6.4.18 Pengujian Validasi Tambah Persediaan	159
6.4.19 Pengujian Validasi Sunting Persediaan	160
6.4.20 Pengujian Validasi <i>Order</i> Persediaan	161
6.4.21 Pengujian Validasi Daftar Pengeluaran.....	162
6.4.22 Pengujian Validasi Penerbitan Surat	162
6.4.23 Pengujian Validasi Hapus <i>Order</i> Persediaan	163
6.4.24 Pengujian Validasi Permintaan Laporan Persediaan	163
6.4.25 Pengujian Validasi Daftar laporan Persediaan	164
6.4.26 Pengujian Validasi Laporan Persediaan	164
6.4.27 Pengujian Validasi Daftar Validasi Persediaan.....	165
6.4.28 Pengujian Validasi Validasi Laporan Persediaan.....	165
6.4.29 Pengujian Validasi <i>Export</i> Laporan Persediaan.....	166
6.4.30 Pengujian Validasi Tambah Pemegang Kendaraan.....	168
6.4.31 Pengujian Validasi Daftar Pemegang Kendaraan.....	169
6.4.32 Pengujian Validasi Hapus Kendaraan.....	169
6.4.33 Pengujian Validasi Permintaan Laporan Pemeliharaan Kendaraan	170
6.4.34 Pengujian Validasi Daftar pemeliharaan Kendaraan	171
6.4.35 Pengujian Validasi Perbarui Pemeliharaan Kendaraan.....	171
6.4.36 Pengujian Validasi Hapus Hasil Pemeliharaan Kendaraan	172
6.4.37 Pengujian Validasi Laporan Pemeliharaan Kendaraan	172
6.4.38 Pengujian Validasi Daftar Validasi Pemeliharaan Kendaraan	173
6.4.39 Pengujian Validasi Validasi Laporan Pemeliharaan Kendaraan	173
6.4.40 Pengujian Validasi Daftar Hasil Pemeliharaan Kendaraan	175



6.4.41 Pengujian Validasi <i>Export</i> Laporan Pemeliharaan Kendaraan	176
6.4.42 Pengujian Validasi Daftar Pengecekan Inventaris Ruangan.....	176
6.4.43 Pengujian Validasi Tambah Aset Pada Inventaris Ruangan.....	177
6.4.44 Pengujian Validasi Hapus Aset Pada Inventaris Ruangan	177
6.4.45 Pengujian Validasi Permintaan Laporan Pengecekan Inventaris Ruangan	178
6.4.46 Pengujian Validasi Perbarui Data Inventaris Ruangan	179
6.4.47 Pengujian Validasi Daftar Hasil Pengecekan Inventaris Ruangan.....	180
6.4.48 Pengujian Validasi Laporan Pengecekan Inventaris Ruangan.....	180
6.4.49 Pengujian Validasi Sunting Hasil Pengecekan Inventaris Ruangan.....	181
6.4.50 Pengujian Validasi Daftar Validasi Pengecekan Inventaris Ruangan	182
6.4.51 Pengujian Validasi Validasi Laporan Pengecekan Inventaris Ruangan	182
6.4.52 Pengujian Validasi <i>Export</i> Hasil Pengecekan Inventaris Ruangan.....	184
6.4.53 Pengujian Validasi Lihat Daftar Aset	186
6.4.54 Pengujian Validasi Lihat Daftar Persediaan	186
6.4.55 Pengujian Validasi Lihat Daftar Pemegang Kendaraan	187
6.4.56 Pengujian Validasi <i>Logout</i>	187
BAB 7 PENUTUP	188
7.1 Kesimpulan.....	188
7.2 Saran	189
DAFTAR REFERENSI	190
LAMPIRAN A BUKTI FISIK.....	192
LAMPIRAN B WAWANCARA.....	199



DAFTAR TABEL

Tabel 4.1 Identifikasi Aktor	36
Tabel 4.2 Kebutuhan Fungsional.....	37
Tabel 4.3 Kebutuhan Non Fungsional.....	43
Tabel 4.4 <i>Use Case Scenario</i> Mendaftarkan Pengguna.....	45
Tabel 4.5 <i>Use Case Scenario Login</i>	45
Tabel 4.6 <i>Use Case Scenario</i> Daftar Pengguna.....	46
Tabel 4.7 <i>Use Case Scenario</i> Sunting Pengguna.....	47
Tabel 4.8 <i>Use Case Scenario</i> Hapus Pengguna.....	47
Tabel 4.9 <i>Use Case Scenario</i> Daftar Aset	48
Tabel 4.10 <i>Use Case Scenario</i> Tambah Aset	49
Tabel 4.11 <i>Use Case Scenario</i> Sunting Aset	50
Tabel 4.12 <i>Use Case Scenario</i> Hapus Aset	50
Tabel 4.13 <i>Use Case Scenario</i> Permintaan Laporan Aset.....	51
Tabel 4.14 <i>Use Case Scenario</i> Daftar Laporan Aset	52
Tabel 4.15 <i>Use Case Scenario</i> Sunting Laporan Aset	53
Tabel 4.16 <i>Use Case Scenario</i> Laporan Aset	53
Tabel 4.17 <i>Use Case Scenario</i> Daftar Validasi Aset.....	54
Tabel 4.18 <i>Use Case Scenario</i> Validasi Laporan Aset.....	55
Tabel 4.19 <i>Use Case Scenario Export</i> Laporan Aset.....	57
Tabel 4.20 <i>Use Case Scenario</i> Daftar Persediaan.....	57
Tabel 4.21 <i>Use Case Scenario</i> Tambah Persediaan.....	58
Tabel 4.22 <i>Use Case Scenario</i> Sunting Persediaan.....	59
Tabel 4.23 <i>Use Case Scenario Order</i> Persediaan.....	60
Tabel 4.24 <i>Use Case Scenario</i> Daftar Pengeluaran	60
Tabel 4.25 <i>Use Case Scenario</i> Penerbitan Surat	61
Tabel 4.26 <i>Use Case Scenario</i> Hapus <i>Order</i> Persediaan	62
Tabel 4.27 <i>Use Case Scenario</i> Permintaan Laporan Persediaan.....	63
Tabel 4.28 <i>Use Case Scenario</i> Daftar Laporan Persediaan.....	63
Tabel 4.29 <i>Use Case Scenario</i> Laporan Persediaan.....	64
Tabel 4.30 <i>Use Case Scenario</i> Daftar Validasi Persediaan	65



Tabel 4.31 <i>Use Case Scenario</i> Validasi Laporan Persediaan	66
Tabel 4.32 <i>Use Case Scenario</i> Export Laporan Persediaan	67
Tabel 4.33 <i>Use Case Scenario</i> Tambah Pemegang Kendaraan	67
Tabel 4.34 <i>Use Case Scenario</i> Daftar Pemegang Kendaraan	68
Tabel 4.35 <i>Use Case Scenario</i> Hapus Kendaraan	69
Tabel 4.36 <i>Use Case Scenario</i> Permintaan Laporan Pemeliharaan Kendaraan	70
Tabel 4.37 <i>Use Case Scenario</i> Daftar Pemeliharaan Kendaraan.....	70
Tabel 4.38 <i>Use Case Scenario</i> Perbarui Pemeliharaan Kendaraan	71
Tabel 4.39 <i>Use Case Scenario</i> Hapus Hasil Pemeliharaan Kendaraan	72
Tabel 4.40 <i>Use Case Scenario</i> Laporan Pemeliharaan Kendaraan.....	73
Tabel 4.41 <i>Use Case Scenario</i> Daftar Validasi Pemeliharaan Kendaraan	73
Tabel 4.42 <i>Use Case Scenario</i> Validasi Laporan Pemeliharaan Kendaraan	74
Tabel 4.43 <i>Use Case Scenario</i> Daftar Hasil Pemeliharaan Kendaraan	76
Tabel 4.44 <i>Use Case Scenario</i> Export Laporan Pemeliharaan Kendaraan	76
Tabel 4.45 <i>Use Case Scenario</i> Daftar Pengecekan Inventaris Ruangan	77
Tabel 4.46 <i>Use Case Scenario</i> Tambah Aset Pada Inventaris Ruangan	78
Tabel 4.47 <i>Use Case Scenario</i> Hapus Aset Pada Inventaris Ruangan	78
Tabel 4.48 <i>Use Case Scenario</i> Permintaan Laporan Inventaris Ruangan.....	79
Tabel 4.49 <i>Use Case Scenario</i> Perbarui Data Inventaris Ruangan	80
Tabel 4.50 <i>Use Case Scenario</i> Daftar Hasil Pengecekan Inventaris Ruangan	81
Tabel 4.51 <i>Use Case Scenario</i> Laporan Pengecekan Inventaris Ruangan	82
Tabel 4.52 <i>Use Case Scenario</i> Sunting Hasil Pengecekan Inventaris Ruangan	83
Tabel 4.53 <i>Use Case Scenario</i> Daftar Validasi Pengecekan Inventaris Ruangan ..	83
Tabel 4.54 <i>Use Case Scenario</i> Validasi Laporan Pengecekan Inventaris Ruangan	84
Tabel 4.55 <i>Use Case Scenario</i> Export Hasil Laporan Pengecekan Inventaris Ruangan	86
Tabel 4.56 <i>Use Case Scenario</i> Lihat Daftar Aset.....	86
Tabel 4.57 <i>Use Case Scenario</i> Lihat Daftar Persediaan.....	87
Tabel 4.58 <i>Use Case Scenario</i> Lihat Daftar Pemegang Kendaraan	87
Tabel 4.59 <i>Use Case Scenario</i> Logout.....	88
Tabel 5.4 Keterangan Perancangan Antarmuka Tambah Aset	94



Tabel 5.5 Keterangan Perancangan Antarmuka Permintaan Pembuatan Laporan Aset.....	95
Tabel 5.6 Keterangan Perancangan Antarmuka Tambah Pemegang Kendaraan .	97
Tabel 5.1 Algoritme Fungsi <i>create()</i>	98
Tabel 5.2 Algoritme Fungsi <i>requestReportkadis()</i>	102
Tabel 5.3 Algoritme Fungsi <i>create()</i>	105
Tabel 5.7 Spesifikasi <i>Hardware</i>	105
Tabel 5.8 Spesifikasi <i>Software</i>	105
Tabel 5.9 Kode Program Fungsi <i>create()</i>	106
Tabel 5.10 Penjelasan Kode Program Fungsi <i>create()</i>	111
Tabel 5.11 Kode Program Fungsi <i>requestReportKadis()</i>	118
Tabel 5.12 Penjelasan Kode Program Fungsi <i>requestReportKadis()</i>	120
Tabel 5.13 Kode Program Fungsi <i>create</i>	122
Tabel 5.14 Penjelasan Kode Program Fungsi <i>create()</i>	123
Tabel 6.1 <i>Pseudocode</i> Fungsi <i>create()</i>	127
Tabel 6.2 Hasil Pengujian Unit Fungsi <i>create()</i>	133
Tabel 6.3 <i>Pseudocode</i> Fungsi <i>requestReportKadis()</i>	137
Tabel 6.4 Hasil Pengujian Unit Fungsi <i>requestReportKadis()</i>	140
Tabel 6.5 <i>Pseudocode</i> Fungsi <i>create()</i>	141
Tabel 6.6 Hasil Pengujian Unit Fungsi <i>requestReportKadis()</i>	142
Tabel 6.7 <i>Scenario Uji Integration Testing</i>	143
Tabel 6.8 Algoritme <i>stubUpdateRequestChecking()</i>	143
Tabel 6.9 Algoritme <i>UpdateRequestChecking()</i>	143
Tabel 6.10 <i>UpdateRequestChecking()</i>	144
Tabel 6.11 <i>UpdateRequestChecking()</i>	145
Tabel 6.12 Hasil <i>Integration testing</i>	146
Tabel 6.13 Skenario Uji Mendaftarkan Pengguna	147
Tabel 6.14 Skenario Uji Mendaftarkan Pengguna Alternatif 1	148
Tabel 6.15 Skenario Uji <i>Login</i>	148
Tabel 6.16 Skenario Uji <i>Login</i> Alternatif 1	148
Tabel 6.17 Skenario Uji <i>Login</i> Alternatif 2	149
Tabel 6.18 Skenario Uji Daftar Pengguna	149



Tabel 6.19 Skenario Uji Sunting Pengguna	149
Tabel 6.20 Skenario Uji Sunting Pengguna Alternatif 1	150
Tabel 6.21 Skenario Uji Hapus Pengguna	150
Tabel 6.22 Skenario Uji Daftar Aset	151
Tabel 6.23 Skenario Uji Tambah Aset	151
Tabel 6.24 Skenario Uji Tambah Aset Alternatif 1	151
Tabel 6.25 Skenario Uji Sunting Aset	152
Tabel 6.26 Skenario Uji Sunting Aset Alternatif 1	152
Tabel 6.27 Skenario Uji Hapus Aset	153
Tabel 6.28 Skenario Uji Permintaan Laporan Aset	153
Tabel 6.29 Skenario Uji Permintaan Laporan Aset Alternatif 1	154
Tabel 6.30 Skenario Uji Daftar Laporan Aset	154
Tabel 6.31 Skenario Uji Sunting Laporan Aset	154
Tabel 6.32 Skenario Uji Sunting Laporan Aset Alternatif 1	155
Tabel 6.33 Skenario Uji Laporan Aset	155
Tabel 6.34 Skenario Uji Validasi Laporan Aset	156
Tabel 6.35 Skenario Uji Validasi Laporan Aset	156
Tabel 6.36 Skenario Uji Validasi Laporan Aset Alternatif 1	157
Tabel 6.37 Skenario Uji Validasi Laporan Aset Alternatif 2	157
Tabel 6.38 Skenario Uji Validasi Laporan Aset Alternatif 3	157
Tabel 6.39 Skenario Uji <i>Export</i> Laporan Aset	158
Tabel 6.40 Skenario Uji <i>Export</i> Laporan Aset Alternatif 1	158
Tabel 6.41 Skenario Uji Daftar Persediaan	159
Tabel 6.42 Skenario Uji Tambah Persediaan	159
Tabel 6.43 Skenario Uji Tambah Persediaan Alternatif 1	159
Tabel 6.44 Skenario Uji Sunting Persediaan	160
Tabel 6.45 Skenario Uji Sunting Persediaan Alternatif 1	160
Tabel 6.46 Skenario Uji <i>Order</i> Persediaan	161
Tabel 6.47 Skenario Uji <i>Order</i> Persediaan Alternatif 1	161
Tabel 6.48 Skenario Uji <i>Order</i> Persediaan Alternatif 2	161
Tabel 6.49 Skenario Uji Daftar Pengeluaran	162
Tabel 6.50 Skenario Uji Penerbitan Surat	162



Tabel 6.51 Skenario Uji Hapus <i>Order</i> Persediaan	163
Tabel 6.52 Skenario Uji Permintaan Laporan Persediaan.....	163
Tabel 6.53 Skenario Uji Permintaan Laporan Persediaan Alternatif 1	164
Tabel 6.54 Skenario Uji Daftar Laporan Persediaan	164
Tabel 6.55 Skenario Uji Laporan Persediaan.....	164
Tabel 6.56 Skenario Uji Daftar Validasi Persediaan	165
Tabel 6.57 Skenario Uji Validasi Laporan Persediaan	165
Tabel 6.58 Skenario Uji Validasi Laporan Persediaan Alternatif 1.....	166
Tabel 6.59 Skenario Uji <i>Export</i> Laporan Persediaan	166
Tabel 6.60 Skenario Uji <i>Export</i> Laporan Persediaan Alternatif 1.....	167
Tabel 6.61 Skenario Uji <i>Export</i> Laporan Persediaan Alternatif 2.....	167
Tabel 6.62 Skenario Uji <i>Export</i> Laporan Persediaan Alternatif 3.....	168
Tabel 6.63 Skenario Uji Tambah Pemegang Kendaraan	168
Tabel 6.64 Skenario Uji Tambah Pemegang Kendaraan Alternatif 1.....	169
Tabel 6.65 Skenario Uji Daftar Pemegang Kendaraan	169
Tabel 6.66 Skenario Uji Hapus Kendaraan	169
Tabel 6.67 Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan.....	170
Tabel 6.68 Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan Alternatif 1	170
Tabel 6.69 Skenario Daftar Pemeliharaan Kendaraan	171
Tabel 6.70 Skenario Uji Perbarui Pemeliharaan Kendaraan	171
Tabel 6.71 Skenario Uji Memperbarui Pemeliharaan Kendaraan Alternatif 1 ...	172
Tabel 6.72 Skenario Hapus Hasil Pemeliharaan Kendaraan	172
Tabel 6.73 Skenario Uji Laporan Pemeliharaan Kendaraan.....	173
Tabel 6.74 Skenario Uji Daftar Validasi Pemeliharaan Kendaraan	173
Tabel 6.75 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan	173
Tabel 6.76 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 1	174
Tabel 6.77 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 2	174
Tabel 6.78 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 3	175
Tabel 6.79 Skenario Uji Daftar Hasil Pemeliharaan Kendaraan.....	176
Tabel 6.80 Skenario Uji <i>Export</i> Laporan Pemeliharaan Kendaraan	176
Tabel 6.81 Skenario Uji Daftar Pengecekan Inventaris Ruangan.....	176



Tabel 6.82 Skenario Uji Tambah Aset Pada Inventaris Ruangan	177
Tabel 6.83 Skenario Uji Hapus Aset Pada Inventaris Ruangan	177
Tabel 6.84 Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan .	178
Tabel 6.85 Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan Alternatif 1	178
Tabel 6.86 Skenario Uji Perbarui Data Inventaris Ruangan	179
Tabel 6.87 Skenario Uji Memperbarui Pengecekan Inventaris Ruangan Alternatif 1	179
Tabel 6.88 Skenario Uji Daftar Hasil Pengecekan Inventaris Ruangan	180
Tabel 6.89 Skenario Uji Laporan Pengecekan Inventaris Ruangan.....	180
Tabel 6.90 Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan	181
Tabel 6.91 Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan Alternatif 1	181
Tabel 6.92 Skenario Uji Daftar Validasi Pengecekan Inventaris Ruangan	182
Tabel 6.93 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan	182
Tabel 6.94 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 1.....	183
Tabel 6.95 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 2.....	183
Tabel 6.96 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 3.....	184
Tabel 6.97 Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan.....	184
Tabel 6.98 Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan Alternatif 1	185
Tabel 6.99 Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan Alternatif 2	185
Tabel 6.100 Skenario Uji Lihat Daftar Aset	186
Tabel 6.101 Skenario Uji Lihat Daftar Persediaan.....	186
Tabel 6.102 Skenario Uji Lihat Daftar Pemegang Kendaraan	187
Tabel 6.103 Skenario Uji <i>Logout</i>	187



DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Perangkat Daerah Perikanan dan Kelautan	7
Gambar 2.2 <i>Waterfall Model</i>	10
Gambar 2.3 Analisis Goolge <i>Trends</i> Mengenai <i>framework</i> Yang Diminati	11
Gambar 2.4 Alur Kerja MVC Pada <i>Framework</i> Laravel	11
Gambar 2.5 <i>The MVC Collaboration Diagram</i>	12
Gambar 2.6 Contoh <i>Use Case Diagram</i>	14
Gambar 2.7 Contoh <i>Sequence Diagram</i>	14
Gambar 2.8 Contoh <i>Class Diagram</i> dan Asosiasi Antar <i>Class</i>	15
Gambar 2.9 Contoh Komponen <i>Class Diagram</i>	15
Gambar 3.1 Diagram Alir Penelitian	17
Gambar 4.1 Bisnis Proses Saat Ini (<i>as-is</i>) Pelaporan Aset	22
Gambar 4.2 Bisnis Proses Saat Ini (<i>as-is</i>) Pelaporan Persediaan	23
Gambar 4.3 Bisnis Proses Saat Ini (<i>as-is</i>) <i>Order</i> Persediaan	24
Gambar 4.4 Bisnis Proses Saat Ini (<i>as-is</i>) Pemeliharaan Kendaraan	25
Gambar 4.5 Bisnis Proses Saat Ini (<i>as-is</i>) Pengecekan Inventaris Ruangan	26
Gambar 4.6 Bisnis Proses Saat Ini (<i>as-is</i>) Pencatatan Aset	27
Gambar 4.7 Bisnis Proses <i>to-be</i> Pelaporan Aset	30
Gambar 4.8 Bisnis Proses <i>to-be</i> Pelaporan Persediaan	31
Gambar 4.9 Bisnis Proses <i>to-be Order</i> Persediaan	32
Gambar 4.10 Bisnis Proses <i>to-be</i> Pemeliharaan Kendaraan Operasional	33
Gambar 4.11 Bisnis Proses <i>to-be</i> Pengecekan Invenatris Ruangan	34
Gambar 4.12 Bisnis Proses <i>to-be</i> Pencatatan Aset	35
Gambar 4.13 <i>Use Diagram</i> Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan	44
Gambar 5.1 <i>Sequence Diagram</i> Tambah Aset	89
Gambar 5.2 <i>Sequence Diagram</i> Permintaan Laporan Aset	90
Gambar 5.3 <i>Sequence Diagram</i> Tambah Pemegang Kendaraan	90
Gambar 5.4 Perancangan <i>Class Diagram</i>	91
Gambar 5.6 Perancangan Antarmuka Tambah Aset	92
Gambar 5.5 <i>Conceptual Data Model</i>	93

Gambar 5.7 Perancangan Antarmuka Permintaan Pembuatan Laporan Aset	95
Gambar 5.8 Perancangan Antarmuka Tambah Pemegang Kendaraan	96
Gambar 5.10 Implementasi Antarmuka Tambah Aset	124
Gambar 5.9 <i>Physical Data Model</i>	125
Gambar 5.11 Implementasi Antarmuka Permintaan Pembuatan Laporan Aset	126
Gambar 5.12 Implementasi Antarmuka Tambah Pemegang Kendaraan	126
Gambar 6.1 <i>Flow Graph</i> Fungsi <i>create()</i>	132
Gambar 6.2 <i>Flow Graph</i> Fungsi <i>requestReportKadis()</i>	139
Gambar 6.3 <i>Flow Graph</i> Fungsi <i>create()</i>	141
Gambar 6.4 Diagram Hirarki <i>Integration Testing</i>	142
Gambar 6.5 Hasil Pengujian Fungsi <i>stubUpdateRequestChecking()</i>	144
Gambar 6.6 Hasil Pengujian Fungsi <i>stubUpdateRequestChecking()</i>	145
Gambar 6.7 <i>Flow Graph Integration</i>	146
Gambar 6.8 Hasil Pengujian <i>Compatibility</i>	147



DAFTAR LAMPIRAN

LAMPIRAN A BUKTI FISIK.....	192
A.1 Kartu Inventaris Ruangan	192
A.2 Buku Penerimaan Barang	192
A.3 Buku Pengeluaran ATK	193
A.4 Buku Pengeluaran Barang	194
A.5 Surat <i>Order</i> Barang	195
A.6 Surat Perintah Pengeluaran Barang	196
A.7 Buku Pemeliharaan Kendaraan	197
A.8 Laporan Kartu Inventaris Barang (KIB) A	198
LAMPIRAN B WAWANCARA.....	199
B.1 Kepala Subbagian Keuangan dan Perencanaan	199
B.2 Staf Pengurus Barang.....	201
B.3 Staf Penyimpanan Barang.....	201



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Penggunaan teknologi informasi memberikan manfaat dalam melakukan suatu pekerjaan kepada penggunanya. Salah satu manfaatnya adalah dalam manajemen aset. Manajemen aset merupakan kegiatan pengorganisasian, perencanaan, memantau akuisisi, perawatan, perbaikan, dan penjualan aset infrastruktur dan rekayasa untuk mendukung pemberian layanan (Brown et.al, 2012).

Seiring berjalannya waktu, setiap instansi baik Swasta maupun BUMN dipastikan memiliki aset yang selalu bertambah setiap tahunnya. Aset memerlukan manajemen yang tepat supaya memudahkan melakukan pengelolaan aset. Dengan adanya manajemen aset yang baik akan mempermudah dalam melakukan pelaporan aset secara rutin. Dalam pelaporan terdapat empat kriteria, yaitu: mudah untuk dipahami, relevan dan *up-to-date*, keandalan, dan dapat diperbandingkan (Saputri, 2014). Kebutuhan akan informasi aset sangat diperlukan dalam melakukan peningkatan efisiensi pada suatu instansi.

Dinas Perikanan Kabupaten Tabanan merupakan lembaga Organisasi Perangkat Daerah (OPD) di lingkungan Pemerintah Kabupaten Tabanan yang telah berdiri sejak tahun 1950-an. Dinas Perikanan Kabupaten Tabanan dalam menjalankan tugasnya dibantu oleh sarana operasioanl. Baik itu yang berupa aset tetap ataupun berupa aset habis pakai. Aset harus selalu diperhatikan kondisi dan jumlahnya apabila terjadi ketidaksesuaian data akan menjadi temuan, yang akan menjadi masalah bagi Dinas Perikanan Kabupaten Tabanan.

Dari hasil wawancara dan observasi yang telah dilaksanakan di Dinas Perikanan Kabupaten Tabanan terdapat permasalahan dalam pengelolaan aset. Selama ini Dinas Perikanan Kabupaten Tabanan telah menyediakan beberapa aset yang digunakan dalam proses pembudidayaan ikan. Dalam melakukan pengelolaan aset tersebut, Dinas Perikanan Kabupaten Tabanan menggunakan *Ms. Excel*. Adapun di dalam pengelolaan aset tersebut salah satunya dilakukan pencatatan terhadap aset yang masuk dan yang sudah ada pada Dinas Perikanan Kabupaten Tabanan.

Dalam melakukan pencatatan aset baru, Dinas Perikanan Kabupaten Tabanan sebelumnya harus melewati proses pemeriksaan yang dilaksanakan oleh tim pemeriksa dari internal Dinas Perikanan Kabupaten Tabanan, apabila sudah selesai dan layak untuk diterima maka akan diserahkan kepada Kepala Subbagian Umum dan Kepegawaian untuk distribusikan dan diberikan kepada pengurus barang untuk dikelola. Kemudian akan dilaporkan pada pelaporan aset di periode berikutnya. Semua aset yang dimiliki dinas dikelola oleh staf bagian pengurus barang, baik itu dalam melakukan pencatatan aset, pelaporan dan juga pengecekan inventaris ruangan. Dalam melakukan pelaporan aset, staf bagian pengurus barang meminta rincian aset yang terdapat di setiap bidang kepada penanggung jawab aset di setiap bidang. Selanjutnya staf bagian pengurus barang membuat laporan berdasarkan konsep yang telah dibuat ke dalam *Ms.Excel*. Pada saat pembuatan laporan aset sering terjadi kesalahan dalam memasukkan data

aset hal ini terjadi karena aset yang baru masuk akan dilaporkan pada pelaporan periode berikutnya serta tidak sesuai informasi mengenai aset yang diberikan dari masing-masing bidang. Selain aset tetap terdapat juga pelaporan aset habis pakai yang dibuat mengacu pada data persediaan yang sudah dicatat dalam buku persediaan yang didalamnya berisi data penerimaan persediaan dan juga data persediaan yang keluar. Setiap aset yang ada pada Dinas Perikanan Kabupaten Tabanan mendapatkan pemeliharaan secara rutin, dalam hal ini aset yang dipelihara hanya aset operasional, seperti kendaraan bermotor. Pemeliharaan yang didapatkan seperti *service* dan pembayaran pajak untuk kendaraan bermotor. Untuk laporan pemeliharaan kendaraan dibebankan kepada pihak yang bertanggung jawab atas aset yang dibawanya dengan menyerahkan nota transaksi dari *service* atau pembayaran pajak kendaraan bermotor kepada pihak yang menangani aset. Selanjutnya dari pihak pengurus barang mencatat pemeliharaan pada buku berdasarkan nota yang telah diberikan. Pemegang kendaraan terkadang tidak memberikan pelaporan rutin mengenai kendaraan yang dipegangnya mengenai kondisi laik kendaraan sehingga menyebabkan staf pengurus barang tidak mengetahui kondisi terkini dari kendaraan operasional. Dinas Perikanan Kabupaten Tabanan melakukan pengecekan inventaris ruangan pada setiap ruangan atau gedung secara berkala setiap semesternya. Semua aset yang ada di setiap ruangan akan dilakukan pengecekan yang didasarkan pada laporan inventaris ruangan pada tiap ruangan. Dalam kegiatan pengecekan inventaris ruangan sering terjadi ketidaksesuaian data antara data yang ada pada kartu inventaris barang dari pelaporan sebelumnya dengan data yang baru diambil dari pengecekan inventaris ruangan. Hal ini disebabkan peminjaman barang dari bidang lain tidak melakukan pemberitahuan kepada staf pengurus barang mengenai informasi barang yang dipinjam.

Dari penjabaran di atas maka penulis memberikan sebuah inovasi dalam melakukan manajemen aset dengan memanfaatkan perkembangan teknologi informasi yaitu membangun sistem informasi manajemen aset yang nantinya dapat memenuhi kebutuhan Dinas Perikanan Kabupaten Tabanan. Sistem informasi manajemen aset ini nantinya akan mengimplementasikan *web*. Pada umumnya sistem informasi lebih banyak mengimplementasikan *web*, dikarenakan *web* memiliki beberapa keuntungan sebagai contohnya adalah akses informasi lebih mudah dan *setup server* yang lebih mudah dan juga aplikasi web dapat berjalan pada *platform* dan *OS* yang berbeda. Maka dari itu penulis mengambil judul "*Pembangunan Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan*". Adapun yang diharapkan dari pembangunan sistem ini yaitu dapat mempercepat kinerja dan mengatasi permasalahan yang dihadapi dalam kegiatan pengelolaan aset dan memberikan informasi tepat kepada pihak yang membutuhkan pada Dinas Perikanan Kabupaten Tabanan.

1.2 Rumusan Masalah

Dari penjabaran di atas, didapatkan rumusan masalah:

1. Bagaimanakah hasil analisis kebutuhan Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan?
2. Bagaimanakah rancangan Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan berdasarkan analisis kebutuhan?
3. Bagaimanakah hasil implementasi Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan dengan hasil rancangan?
4. Bagaimanakah hasil pengujian fungsional, *compatibility*, pengujian unit dan pengujian integrasi Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan dari hasil implementasi?

1.3 Tujuan

Adapun tujuan yang ingin didapatkan pada penelitian ini adalah:

1. Merekayasa kebutuhan pengguna Sistem Informasi Manajemen Aset pada Dinas Perikanan Kabupaten Tabanan.
2. Merancang Sistem Informasi Manajemen Aset pada Dinas Perikanan Kabupaten Tabanan dari hasil rekayasa kebutuhan.
3. Mengimplementasikan Sistem Informasi Manajemen Aset pada Dinas Perikanan Kabupaten Tabanan dari hasil perancangan sistem.
4. Melakukan pengujian fungsional, *compatibility*, pengujian unit, dan pengujian integrasi Sistem Informasi Manajemen Aset pada Dinas Perikanan Kabupaten Tabanan dari hasil implementasi sistem.

1.4 Manfaat

Pembuatan penelitian ini nantinya diharapkan sistem yang dibuat penulis dapat memberikan manfaat yang signifikan dalam memanajemen aset dan data-data aset Dinas Perikanan Kabupaten Tabanan yang saling terintegrasikan.

1.5 Batasan Masalah

Agar permasalahan tidak melebar dari fokus penelitian yang dilakukan maka penulis membatasi penelitian ini, yaitu dalam pembangunan sistem informasi manajemen aset yang menggunakan data dari Dinas Perikanan Kabupaten Tabanan dan pengembangan sistem berbasis web.

1.6 Sistematika Pembahasan

BAB I Pendahuluan

Bab ini memuat latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.



BAB II Landasan Kepustakaan

Bab ini memuat dasar teori dan referensi yang mendasari perancangan, implementasi dan pengujian sistem informasi manajemen aset.

BAB III Metodologi Penelitian

Bab ini memuat metode dan langkah-langkah penelitian yang digunakan dalam penelitian.

BAB IV Rekayasa Kebutuhan

Bab ini memuat elisitasi dan analisis kebutuhan Sistem Informasi Manajemen Aset pada Dinas Perikanan Kabupaten Tabanan.

BAB V Perancangan dan Implementasi

Bab ini menjelaskan mengenai perancangan yang didasarkan dari analisis kebutuhan yang nantinya akan diimplementasikan pada kode program.

BAB VI Pengujian dan Analisis

Bab ini memuat tentang alur dari pengujian serta hasil dari analisis pengujian terhadap sistem yang sudah dibangun.

BAB VII Penutup

Bab ini memuat kesimpulan dan saran yang didapatkan dalam pelaksanaan penelitian.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Aset berperan penting dalam kegiatan operasional suatu instansi, maka dari itu dibutuhkan informasi yang *up-to-date* untuk mengelola aset agar tidak mengganggu dalam kegiatan operasional. Kebutuhan informasi akan aset sangat diperlukan untuk meningkatkan kinerja di dalam sebuah instansi (Pambudi et.al, 2016).

Penelitian yang serupa mengenai sistem informasi manajemen aset telah dilakukan oleh peneliti sebelumnya salah satunya dibuat oleh Angelina Permatasari pada tahun 2015 tentang “Aset Manajemen Rumah Sakit Berbasis *Web System*” dari penelitian tersebut memberikan kemudahan dalam melakukan manajemen aset pada rumah sakit menggunakan *Web Base System* sebagai *tools* yang meningkatkan kegiatan manajemen aset serta dapat berjalan dengan ekonomis, efektif dan efisien (Permatasari, 2015).

Penelitian yang serupa mengenai sistem informasi aset yang dibuat oleh Rusli, Chendry dan Inayatullah yang dibuat pada tahun 2017 berjudul “Sistem Informasi Manajemen Aset Berbasis *Web* Pada *The Pitstop*” dalam penelitian yang dilakukan mengenai sistem informasi manajemen aset memberikan dampak kemudahan dalam mendapatkan laporan mengenai kondisi aset dan juga meminimalisir ketidaksesuaian pencatatan dalam penambahan dan pengurangan aset dan juga memberikan kemudahan pemantauan aset secara periodik dan *real-time* (Rusli, Chendry & Inayatullah, 2017).

Dari beberapa rangkuman diatas, didapatkan bahwa dengan memakai sistem informasi manajemen aset dapat memberikan kemudahan dalam mengelola aset serta mengurangi terjadinya ketidaksesuaian data aset. Penelitian ini bertujuan untuk melakukan analisis dan perancangan aplikasi berbasis *web*, menerapkan *SDLC Waterfall model*, dan pada implementasi menerapkan bahasa pemrograman HTML, PHP yang menggunakan *Framework* Laravel, pola perancangan *Model-View-Controller* (MVC), dan menggunakan *database* MySQL.

2.2 Dinas Perikanan Kabupaten Tabanan

Dinas Perikanan Kabupaten Tabanan merupakan lembaga Organisasi Perangkat Daerah (OPD) di lingkungan Pemerintah Kabupaten Tabanan telah berdiri sejak tahun 1950-an. Bergerak di bidang maritime dan juga pembudidayaan ikan (Dinas Perikanan Kabupaten Tabanan, 2016).

2.2.1 Visi dan Misi

Dalam menjalankan kegiatannya Dinas Perikanan Kabupaten Tabanan telah menganalisis dan menjabarkannya dalam rencana strategis yaitu:

a. Telaah terhadap visi

Kepala dan Wakil Kepala Daerah Kabupaten Tabanan telah menetapkan visi pembangunan yaitu Tabanan Sejahtera Aman dan Berprestasi (Tabanan Serasi) sehati membangun negeri, dengan maksud:

1. Membangun dilandasi rasa empati, rasa memiliki (*sense of belonging*) sehingga pembangunan dan hasil-hasilnya adalah milik kita bersama.
2. Membangun dilakukan dengan semangat gotong royong karena diyakini tidak ada masalah yang berat jika kita pikul bersama-sama.
3. Membangun dengan semangat persaudaraan untuk menumbuhkan rasa bangga sebagai masyarakat Tabanan.
4. Tidak ada sekat perbedaan yang mengurangi nilai persatuan dan kesatuan kita sebagai satu keluarga besar, masyarakat Tabanan.

b. Telaah terhadap misi

Kepala dan Wakil Kepala Daerah Kabupaten Tabanan telah menetapkan lima Misi sebagai penjabaran atas visi pembangunan, yaitu:

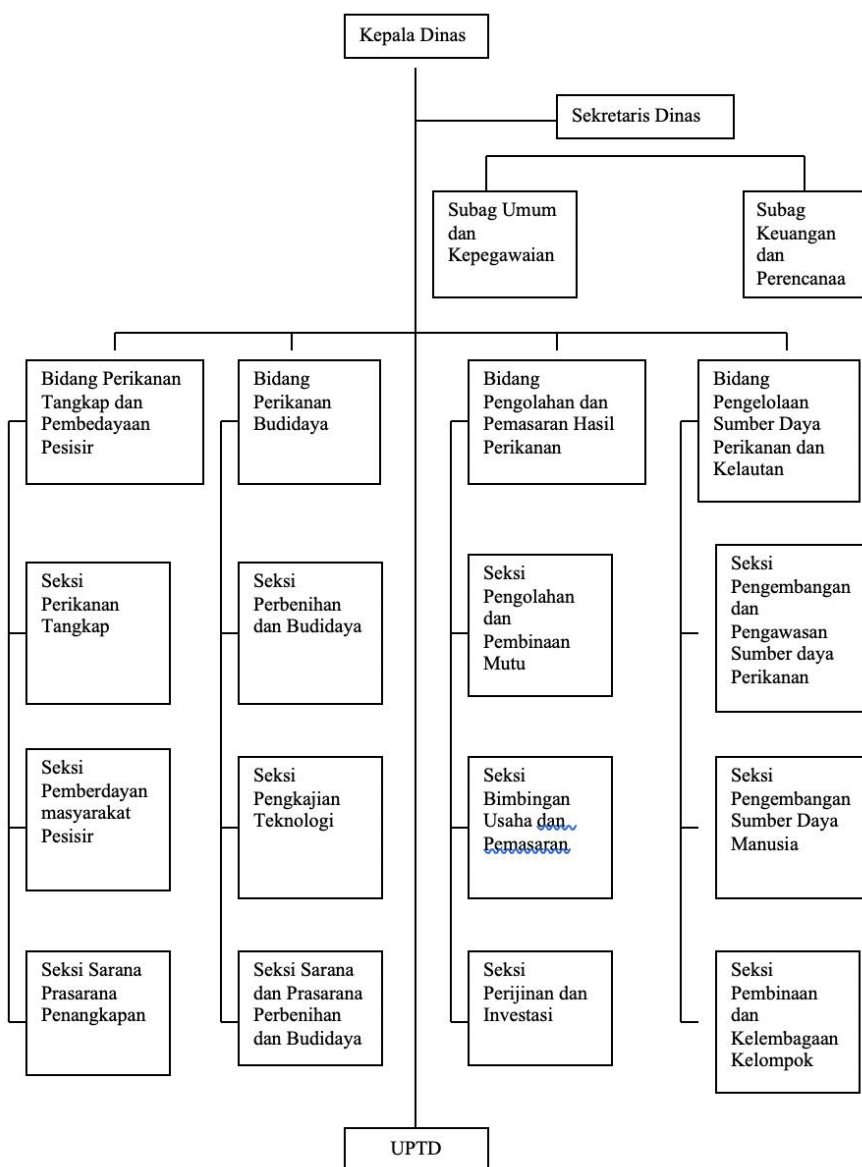
1. Membentuk sumber daya manusia yang berkualitas, berbudaya, dan berkeadilan sosial.
2. Meningkatkan derajat kesehatan masyarakat melalui peningkatan akses dan pelayanan prima.
3. Menggerakkan ekonomi kerakyatan melalui pemberdayaan masyarakat berbasis pertanian dan pariwisata.
4. Mempercepat pengembangan pusat-pusat pertumbuhan wilayah melalui peningkatan kualitas infrastruktur.
5. Memperkuat birokrasi yang berorientasi kinerja, transparan, dan berdaya saing berbasis teknologi informasi.

Dinas Perikanan Kabupaten Tabanan sesuai kewenangan dan urusan maka difokuskan pada tiga misi yang menjadi arah kebijakan pemerintah daerah adalah:

1. Memperkuat ekonomi Tabanan dengan pelibatan masyarakat lokal seluas-luasnya untuk menggarap potensi unggulan di setiap wilayah.
2. Mensinergikan pembangunan pertanian dengan pariwisata agar pariwisata berkontribusi positif terhadap kesejahteraan petani dan nelayan sehingga tidak menjadi ancaman bagi terwujudnya pertanian berkelanjutan.
3. Menciptakan nilai tambah terhadap produksi pertanian sehingga pertanian semakin diminati dan menciptakan peluang kerja bagi masyarakat pedesaan.

2.2.2 Struktur Organisasi

Berikut adalah gambar struktur organisasi Dinas Perikanan Kabupaten Tabanan sesuai dengan Peraturan Daerah Kabupaten Tabanan Nomor 13 Tahun 2016 pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi Perangkat Daerah Perikanan dan Kelautan

Sumber: Dinas Perikanan Kabupaten Tabanan(2016)

Dinas Perikanan Kabupaten Tabanan dipimpin oleh Kepala Dinas yang dibantu oleh Sekretaris Dinas. Di bawah Sekretaris Dinas terdapat dua subbagian, yaitu Subbagian Umum dan Kepegawaian serta Subbagian Keuangan dan Perencanaan. Kedua subbagian ini yang menangani mengenai aset, baik itu dalam kegiatan pelaporan aset, pelaporan persediaan, mengenai pemeliharaan kendaraan, pengecekan inventaris ruangan dan juga pencatatan mengenai aset yang baru masuk ke dalam Dinas Perikanan Kabupaten Tabanan.



2.3 Sistem Informasi Manajemen Aset

2.3.1 Sistem Informasi

Sistem dapat diartikan sebagai kumpulan dari komponen yang saling berhubungan satu sama lain (Indrajit, 2007). Suatu sistem berisikan kumpulan dari bagian yang bekerja bersama menuju tujuan. Tujuan dari sistem adalah untuk menerima masukan dan mengubahnya menjadi keluaran. Sistem informasi diartikan sebagai suatu kumpulan dari komponen-komponen dalam suatu organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi (Indrajit, 2007).

2.3.2 Manajemen Aset

Tahapan dalam melakukan manajemen aset terdiri dari lima tahapan, yaitu (Akbar & Wijaya, 2008):

1. Inventaris Aset
Pada kegiatan inventarisasi aset terdiri dari, inventarisasi fisik dari bentuk dan yuridis/legal.
2. Legal Audit
Merupakan suatu lingkup kerja manajemen aset yang berupa inventarisasi status penguasaan aset, sistem dan prosedur penguasaan.
3. Penilaian Aset
Untuk mengetahui nilai atas aset yang dikuasai.
4. Optimalisasi Aset
Optimalisasi aset merupakan proses kerja dalam manajemen aset yang bertujuan untuk mengoptimalkan potensi fisik, lokasi, nilai, jumlah/volume, legal dan ekonomi yang dimiliki aset tersebut.
5. Pengawasan dan Pengendalian
Merupakan kegiatan yang sering mendapatkan perhatian lebih. Kegiatan yang paling sesuai dalam meningkatkan kinerja pengawasan dan pengendalian adalah dengan pembangunan sistem informasi yang memungkinkan transparansi dalam pengolahan aset dapat terjamin dengan baik tanpa perlu adanya kekhawatiran akan pengawasan dan pengendalian yang lemah.

2.3.3 Sistem Informasi Manajemen Aset

Menurut Hartono dalam Wardhani dan Zulfiandri (2015) menyatakan bahwa pengimplementasian Sistem Informasi Manajemen Aset pada dasarnya bertujuan dalam menyediakan serta melakukan pendataan data-data / dokumen yang menyertai keberadaan aset dan memudahkan dalam melakukan pengelolaan aset seperti pengadaan, penerimaan, perubahan data, hingga penghapusan aset.

Sistem informasi manajemen aset akan mengoptimalkan penggunaan aset yang akan mengurangi ketidakefisienan dan melakukan penghitungan aset secara tepat dan merinci. Dengan pengolahan data secara cepat dan akurat akan dapat memberikan pelayanan yang memuaskan mengenai informasi aset kepada pihak yang membutuhkan. Laporan yang dibuat akan menjadi lebih terstruktur karena

data aset sudah ada pada *database*, sehingga memudahkan dalam melakukan pelaporan pengelolaan aset (Priyanto, 2011).

2.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan salah satu disiplin ilmu yang bergerak dalam pembuatan *software* yang dimulai dari spesifikasi sistem sampai dengan *maintenance* ketika sistem sudah diimplementasikan (Sommerville, 2011). Mulai dari mengumpulkan data *requirement*, *requirements specification*, *design*, *implementation*, *testing* dan juga *maintenance*. Dalam pembuatan sebuah perangkat lunak terdapat *user requirement* dan *system requirement*. Keduanya mencerminkan bagaimana suatu perangkat lunak akan dibuat. Perangkat lunak dapat dikatakan baik apabila memenuhi beberapa karakteristik (Sommerville, 2010), yaitu:

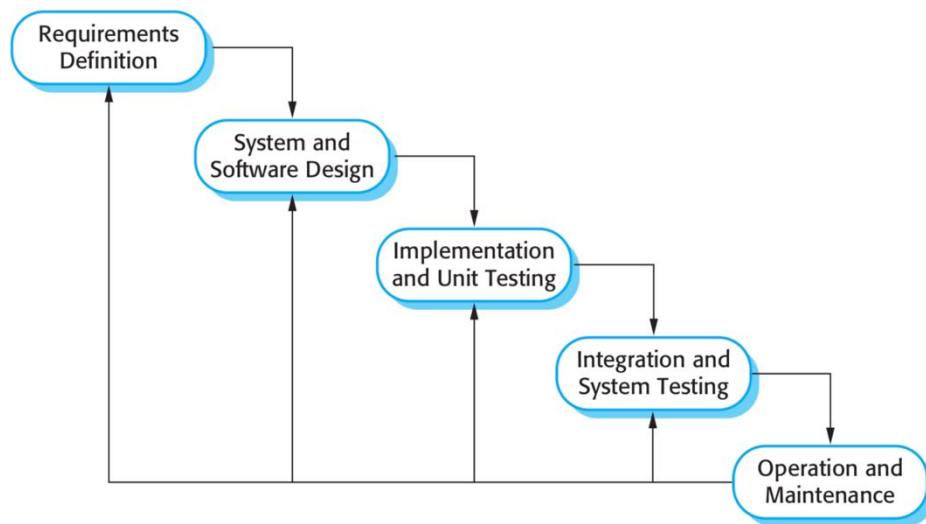
1. *Maintainability*, *software* harus dapat memenuhi perubahan kebutuhan.
2. *Dependability and Security*, *software* harus dapat diandalkan.
3. *Efficiency*, *software* harus efisien dalam penggunaan *resource*.
4. *Usability*, *software* dapat dijalankan dengan mudah.

2.5 System Development Life Cycle (SDLC)

SDLC (System Development Life Cycle) adalah proses pengembangan sistem yang dipakai dalam membuat atau mengembangkan sebuah sistem informasi. *SDLC* banyak diterapkan ketika akan membuat atau mengembangkan sistem karena produk yang dihasilkan akan sesuai dengan kebutuhan pengguna sistem, akan lebih efisien dan efektif serta data menghemat biaya dan waktu dalam melakukan *maintenance* setelah sistem diterapkan (Indriyani, 2012).

2.5.1 Waterfall Model

Waterfall Model adalah salah satu metode pengembangan perangkat lunak yang paling umum dari ke semua model pengembangan *software*. Model *Waterfall* dalam proses pengembangan *software* dilaksanakan secara linier atau prosesnya saling berurutan atau saling memengaruhi dari proses sebelumnya. Pada Gambar 2.2 merupakan proses pengembangan model *Waterfall*.



Gambar 2.2 Waterfall Model

Sumber: Sommerville(2011)

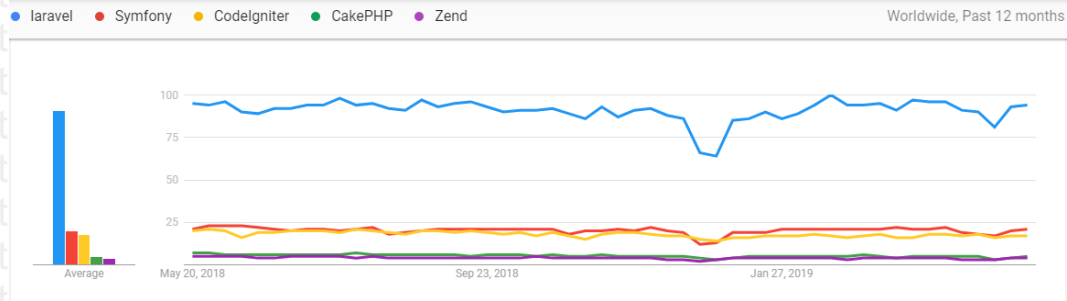
Pada Gambar 2.2 menjelaskan mengenai prinsip dasar dari metode *waterfall* dalam pembangunan perangkat lunak (Sommerville, 2011):

1. *Requirement analysis and definition* adalah tahap pengumpulan kebutuhan yang dipakai dalam pembangunan *software* secara lengkap dan melakukan analisis guna untuk mendefinisikan kebutuhan dalam pembangunan sistem. Dalam tahap ini semua kebutuhan harus didefinisikan secara rinci karena akan memengaruhi kualitas dari sistem yang akan dibangun.
2. *System and software design* ketika semua kebutuhan telah didefinisikan secara terperinci maka selanjutnya adalah membuat *design*.
3. *Implementation and unit testing* dalam tahapan ini melakukan penerapan *design* yang sudah dibuat selanjutnya melakukan pengujian terhadap *design* yang telah diimplementasikan. Dalam pengujian ini melibatkan pengujian verifikasi dari setiap unit yang ada sesuai dengan spesifikasi kebutuhannya.
4. *Integrated and system testing*, melakukan integrasi dari masing-masing unit program dan selanjutnya diuji secara keseluruhan.
5. *Operation and maintenance* merupakan kegiatan menjalankan atau memasang program ke dalam lingkungannya dan melakukan pemeliharaan terhadap sistem.

2.6 Kerangka Kerja (Framework)

Laravel merupakan *framework* pengembangan *web MVC* yang ditulis pada bahasa pemrograman PHP. Laravel adalah salah satu dari beberapa *framework* PHP yang menawarkan *code modularity* (*bundles system*, *migration* dan *artisan CLI*) yang benar. Memudahkan mengubah dan memperluas fungsi *caching*, *session*, *database* dan *authentication functionality* (Parkar et.al, 2016).

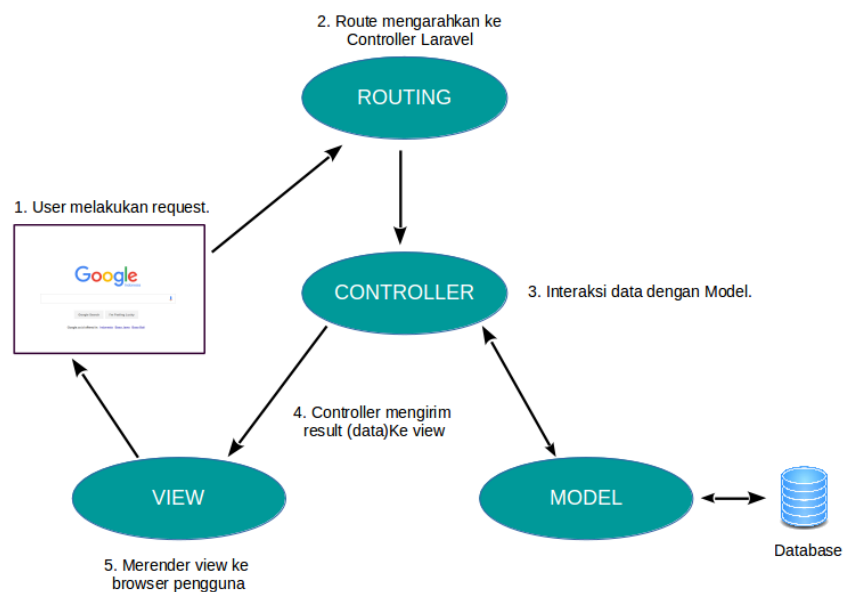
Framework Laravel menjadi *framework* yang paling banyak diminati oleh *programmer* PHP dalam pembangunan sebuah aplikasi. Karena *framework* ini mudah dalam pemakaiannya dan tidak mengherankan peminat dari *framework* Laravel dari tahun ke tahun selalu meningkat. Berikut hasil analisis *Google Trends* mengenai banyaknya peminat terhadap *framework* Laravel dari tahun tanggal 20 Mei 2018 sampai 19 Mei 2019 pada Gambar 2.3.



Gambar 2.3 Analisis Goolge Trends Mengenai *framework* Yang Diminati

Sumber: <https://trends.google.com/> (2019)

Alur kerja dari *framework* laravel ini adalah *user* mengirimkan *request* kepada *web server* kemudian diteruskan ke sistem *routing* Laravel. Di dalam sistem *routing*, laravel akan memproses *request* yang masuk dan mengarahkannya sesuai dengan *request url* yang telah didefinisikan sebelumnya. *Controller* melakukan komunikasi dengan *model* yang berhubungan dengan *database*, selanjutnya *controller* akan melakukan *render view* dalam bentuk HTML dan dikirim kembali ke *browser user*. Pada Gambar 2.4 merupakan alur kerja MVC pada *framework* Laravel.



Gambar 2.4 Alur Kerja MVC Pada *Framework* Laravel

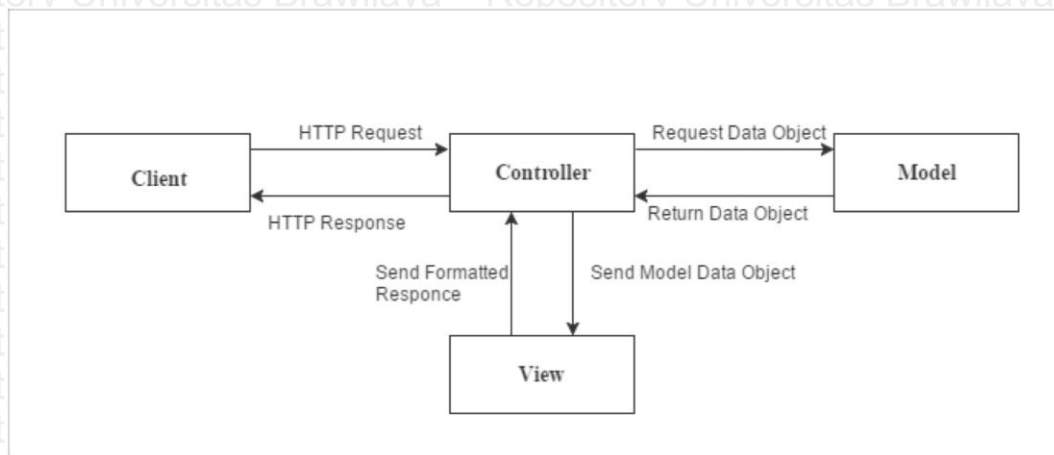
Sumber: www.mahirkoding.com/arsitektur-laravel-5-2/ (2019)



Keunggulan dari *framework* laravel yaitu, *performance* lebih cepat, *reload* data lebih stabil, terdapat fitur seperti *blade* yang menggunakan konsep *HMVC* (*Hierarchical Model View Controller*), dan juga terdapat fitur pengelolaan *migrations* untuk pembuatan skema *table* pada *database*.

2.7 MVC (Model-View-Controller)

MVC merupakan pengembangan sebuah aplikasi yang memisahkan komponen utamanya dalam membangun sebuah sistem. Pada MVC *pattern* terdapat tiga modul dalam membangun aplikasi, yaitu pertama *Model* berhubungan dalam melakukan pengolahan data (*insert*, *update*, *delete* dan *search*) pada *database*. Kedua *View* bertanggung jawab untuk menampilkan data yang disediakan oleh *model* kepada pengguna. Ketiga adalah *Controller* berfungsi untuk mengatur hubungan antara *model* dan *view*. *Controller* menerima *request* data dari *user*, selanjutnya memanggil *model* untuk melakukan operasi yang diminta dan mengirim data ke *view* dalam bentuk aplikasi (Parker at.al, 2016). Berikut alur dalam pengembangan sistem informasi menggunakan MVC dapat dilihat pada Gambar 2.5.



Gambar 2.5 The MVC Collaboration Diagram

Sumber: Parker at.al (2016)

2.8 MySQL

MySQL merupakan sebuah *Database Management System* (DBMS). Basis data relasional dengan data yang disimpan pada tabel-tabel terpisah sesuai dengan keperluan, dan tidak menyimpan semua data dalam satu tabel yang besar. Tabel-tabel dihubungkan menggunakan relasi-relasi yang menghubungkan data dari beberapa tabel (Iswahyudi & Susanti, 2015).

2.9 Proses Bisnis

Proses bisnis merupakan serangkaian aktivitas yang memerlukan masukan serta menghasilkan keluaran sesuai kebutuhan. Kegiatan tersebut dilakukan secara terkoordinasi untuk mencapai tujuan organisasi. Setiap proses bisnis tidak

hanya berinteraksi pada satu organisasi melainkan dapat berinteraksi dengan proses bisnis pada organisasi lainnya (Weske, 2007).

Proses bisnis digambarkan dalam sebuah diagram, tiap diagram menggambarkan aktivitas dan tugas dari proses bisnis dan hubungan di antara keduanya. Dalam proses bisnis terdapat dua, yaitu proses bisnis saat ini atau *as-is* dan proses bisnis *to-be* yang akan diimplementasikan. Proses bisnis *as-is* didapatkan dari kegiatan wawancara langsung dengan *stakeholder* yang terlibat. Guna untuk memberikan informasi mengenai keadaan saat ini pada suatu perusahaan atau organisasi serta untuk mengetahui kekurangan yang terdapat pada proses bisnis saat ini. Sedangkan proses bisnis *to-be* merupakan kegiatan untuk mengidentifikasi dan memodelkan proses bisnis yang sudah mengalami peningkatan (*improvement*) berdasarkan analisis terhadap proses bisnis *as-is*.

2.10 Business Process Modeling and Notation (BPMN)

Business Process Modeling Notation merupakan notasi grafis yang menggambarkan langkah-langkah logika pada suatu proses bisnis. Notasi BPM ini dirancang untuk mengkoordinasikan urutan proses dan urutan pesan yang mengalir antara peserta dalam suatu kegiatan yang berbeda.

BPMN dirancang agar mudah dipahami oleh semua pemangku kepentingan bisnis. Maka dari itu BPMN berfungsi untuk menjembatani kesenjangan komunikasi yang sering terjadi antara desain dan implementasi proses bisnis. Terdapat empat kategori elemen dasar, yaitu (Weske, 2007):

1. *Flow Object*: elemen utama yang menggambarkan perilaku proses bisnis, terdiri dari *events*, *activity*, *gateways*.
2. *Artifacts*: menggambarkan informasi tambahan mengenai proses bisnis. Terdiri dari *data object*, *groups*, dan *text annotations*. Setiap *artifacts* dapat dikaitkan dengan *flow elements*.
3. *Connection Object*: menghubungkan *flow object*, *swimlanes*, atau *artifacts*. Terdiri dari *sequence flows*, *message flows*, *association*, dan *data associations*.
4. *Swimlanes*: terdapat *pools* menggambarkan suatu organisasi. *Lanes* merepresentasikan entitas dalam organisasi.

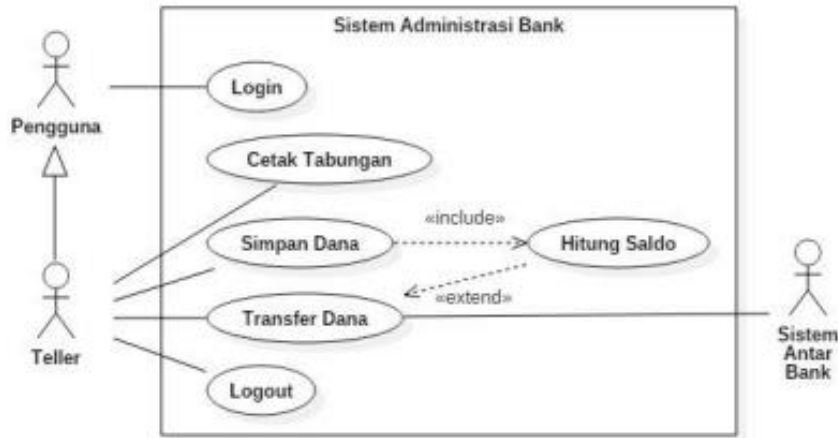
2.11 Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah bahasa yang telah menjadi standar dalam pembangunan perangkat lunak yang berguna untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML merupakan konsep *class* dan *operation* dalam konsep dasarnya yang berguna dalam melakukan analisis dan merancang sebuah aplikasi (Pressman, 2010).

2.11.1 Use Case Diagram

Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem (Sommerville, 2011). Menurut Larman dalam Kurniawan (2018) *Use case*

diagram dapat menjadi gambaran yang baik dalam menjelaskan konteks dari sebuah sistem. Diagram ini dapat memberikan manfaat dalam menyusun kebutuhan sistem dengan melakukan komunikasi pada klien serta untuk merancang *test case* bagi semua *feature* yang ada pada sistem. Pada Gambar 2.6 merupakan sebuah contoh dari *use case diagram*.

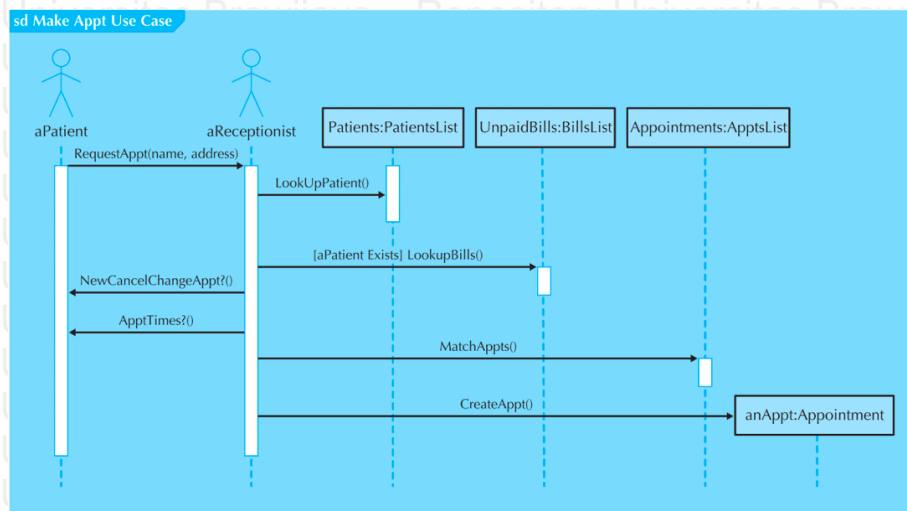


Gambar 2.6 Contoh Use Case Diagram

Sumber: Kurniawan (2018)

2.11.2 Sequence Diagram

Sequence diagram merupakan diagram yang menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu (Sommerville, 2011). *Sequence Diagram* menggambarkan rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu *even* untuk menghasilkan keluar tertentu. Contoh dari *sequence diagram* pada Gambar 2.7 berikut.



Gambar 2.7 Contoh Sequence Diagram

Sumber: Dennis, Wixom, & Tegarden (2009)

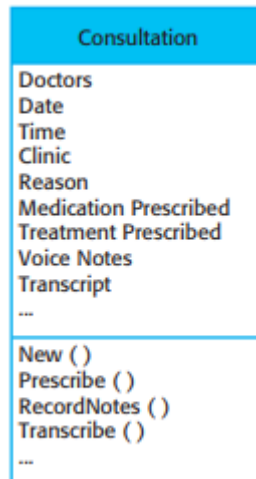
2.11.3 Class Diagram

Class diagram merupakan himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute/properties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak/beraksi dan memberikan reaksi (Dennis, Wixom and Tegarden, 2009). *Class diagram* menunjukkan kelas objek di dalam sistem dan juga asosiasi antara *class*. Contoh dari *class diagram* dan asosiasinya pada Gambar 2.8. Pada setiap *class diagram* terdapat tiga bagian, yaitu: nama dari *class*, *attribute*, dan juga *operation* (berisikan *method-method*). Pada Gambar 2.9 merupakan contoh gambar dari komponen *class diagram*.



Gambar 2.8 Contoh Class Diagram dan Asosiasi Antar Class

Sumber: Sommerville (2011)



Gambar 2.9 Contoh Komponen Class Diagram

Sumber: Sommerville (2011)

2.12 Pengujian Perangkat Lunak

Pengujian merupakan proses mengevaluasi suatu sistem atau komponen dengan tujuan untuk menemukan bahwa sistem yang telah dibuat sudah memenuhi standar yang telah ada atau tidak. Dengan kata lain pengujian adalah untuk menemukan kesenjangan, kesalahan yang bertentangan dengan fungsionalitas yang ada sebelumnya.



2.12.1 White Box Testing

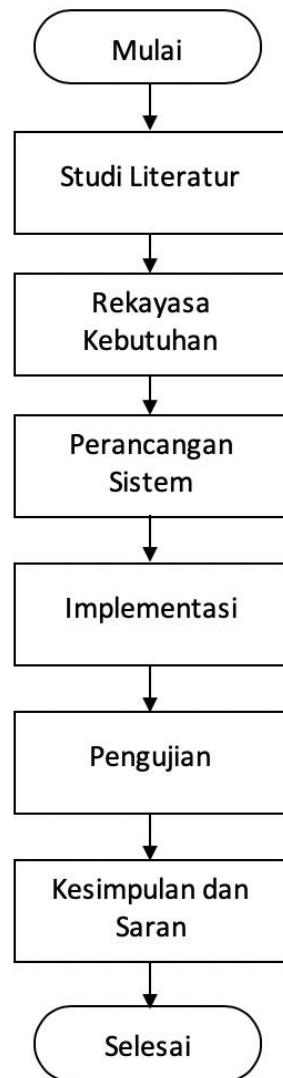
White box testing kadang juga disebut dengan *glassbox testing* merupakan filosofi *design test-case* yang menggunakan struktur *control* yang dijelaskan sebagai bagian *design* komponen *level* untuk mendapatkan kasus uji. Metode *white box testing* nanti akan melakukan pengujian pada setiap unit seperti *class*, komponen atau objek dari *software*. Pengujian *white box* melakukan pengujian yang memastikan semua jalur *independent path* sudah dilakukan pengujian minimal sekali, menguji semua keputusan logika baik itu benar ataupun salah, menguji perulangan pada batasan struktur. Pada *white box testing* akan memakai teknik *basis path testing*, yaitu pengujian struktur program yang dilakukan sesuai dengan jalur logika. Dengan mendefinisikan suatu jalur unik pada struktur program dengan melakukan pengujian minimal sekali (Pressman, 2010). Untuk dapat melakukan *basis path testing* sebelumnya harus menentukan *flow graph* terlebih dahulu. Selanjutnya baru bisa menentukan *cyclomatic complexity* yang dipakai mengukur tingkat kompleksitas pada struktur program yang akan diuji serta mendefinisikan kasus uji yang akan dilakukan (Kurniawan, 2007). Ketika sudah menentukan *flow graph* selanjutnya menghitung tingkat kompleksitas menggunakan persamaan $V(G) = \text{jumlah area } (R)$, $V(G) = \text{Edge} - \text{Node} + 2$ dan $V(G) = \text{Predicate} + 1$. Selain menggunakan *basis path testing* juga pada *white box testing* menggunakan *integration testing* yang dipakai untuk membangun arsitektur *software* yang bersifat sementara yang dipakai untuk menemukan kesalahan yang mungkin terjadi pada antarmuka (Pressman, 2010). Terdapat dua strategi pada *integration*, yaitu: (1) *top-down integration* merupakan pendekatan tambahan dalam pembangunan *software* yang dimulai dari modul utama ke bawah dengan menggantikan komponen menggunakan *stub*, dan (2) *bottom-up integration*, memulai pengujian dari komponen yang paling bawah. Fungsionalitas yang disediakan dari komponen bawah yang selalu ada dan kebutuhan *stub*-nya dihilangkan (Pressman, 2010).

2.12.2 Black Box Testing

Black-box testing berfokus pada persyaratan fungsional dari *software*. Pengujian ini mencoba untuk mencari kesalahan pada beberapa kategori, seperti terdapat fungsi yang tidak sesuai atau hilang, terjadi *error* pada antarmuka, kesalahan pada struktur data atau dalam akses *database* eksternal, kesalahan pada kinerja, dan inisialisasi dan *termination errors* (Pressman, 2010).

BAB 3 METODOLOGI PENELITIAN

Bab ini memuat mengenai tahapan dalam melakukan penelitian. Dalam hal ini peneliti melakukan kajian terhadap studi literatur, rekayasa kebutuhan, perancangan dari hasil analisis yang didapatkan, melakukan implementasi dari hasil rancangan yang diperoleh, melakukan pengujian terhadap sistem yang dibangun dan melakukan penarikan kesimpulan berdasarkan pada hasil keseluruhan dari pembangunan sistem yang dilakukan, digambarkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Pada tahap studi *literature* memuat teori dasar yang akan digunakan menunjang penulisan dan sebagai pendukung penulisan penelitian. Sumber atau referensi pendukung bisa diambil dari buku, jurnal, *e-book*, artikel, laporan

penelitian dan bantuan mesin pencari (*search engine*) yang berkaitan dengan penelitian ini. Pada penulisan skripsi ini penulis mencari studi literatur mengenai:

1. Sistem Informasi Manajemen Aset
2. *System Development Life Cycle*
3. *Waterfall Model*
4. PHP
5. Kerangka kerja (*Framework*)
6. Laravel
7. *Unified Modelling Language (UML)*
8. Pengujian perangkat lunak

Model pengembangan sistem memakai pengembangan *waterfall model*. Pengembangan *waterfall model* merupakan sebuah pendekatan yang bersifat linier. Tahap awal sebelumnya belum terselesaikan maka tahap selanjutnya tidak akan bisa dilanjutkan dan pendekatan *waterfall* ini tidak dapat diulang lagi ke tahap sebelumnya. Karena spesifikasi kebutuhan sistem bersifat *generic*, artinya semua kebutuhan sudah ditentukan dan ditetapkan di awal perancangan sehingga cocok untuk menerapkan metode pendekatan *waterfall*. Dalam pengembangan sistem ini tahap yang dilalui hanya sampai tahapan *integration and system testing*.

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan merupakan kegiatan yang dilakukan guna mendapatkan data selanjutnya melakukan analisis permasalahan yang terjadi serta memberikan sebuah solusi mengenai kebutuhan pengguna untuk pembangunan perangkat lunak. Tahapan yang dilakukan dalam melakukan kegiatan rekayasa kebutuhan dengan cara wawancara dan melaksanakan observasi secara langsung pada Dinas Perikanan Kabupaten Tabanan.

Tahapan ini dilakukan untuk mengetahui alur dari proses bisnis serta pihak-pihak yang terlibat di dalamnya. Tahapan elisitasi kebutuhan dilakukan dengan wawancara kepada Kepala Subbagian Keuangan dan Perencanaan yaitu bapak Agung dan juga stafnya yaitu ibu Luh De dan melakukan observasi lapangan guna mendapatkan data proses terjadinya kegiatan pelaporan aset, pelaporan persediaan, pemeliharaan kendaraan operasional, pengecekan inventaris ruangan, dan pencatatan penerimaan aset baru pada Dinas Perikanan.

Hasil elisitasi kemudian dianalisis guna mengetahui masalah yang terjadi kemudian dicarikan solusi terkait permasalahan yang terjadi. Solusi dari permasalahan tersebut menjadi kebutuhan pengguna yang harus ada pada sistem yang akan dibangun. Setelah mendapatkan hasil analisis selanjutnya hasil kebutuhan kemudian mendefinisikannya menjadi kebutuhan fungsional. Pemodelan yang dilakukan menggunakan metode *Unified Modeling Language*

(UML) yang menggunakan pendekatan berorientasi objek. Nantinya dapat mengidentifikasi aktor-aktor yang terlibat, *use case diagram* dan *use case scenario*.

3.3 Perancangan Sistem

Tahap perancangan sistem adalah menerjemahkan kebutuhan sistem dan pengguna menjadi sebuah visualisasi model sistem informasi. Dari hasil tahapan ini nantinya akan didapatkan *sequence diagram*, *class diagram* berdasarkan pada hasil rekayasa kebutuhan yang telah dilakukan dan juga membuat tampilan antarmuka pengguna.

3.4 Implementasi

Pada implementasi menerjemahkan perancangan sistem ke dalam bahasa pemrograman. Implementasi sistem dibangun memakai bahasa pemrograman PHP menggunakan kerangka kerja Laravel dalam pengembangan GUI dan menggunakan *database* MySQL. Pola arsitektur pembangunan sistem memakai pola perancangan *Model-View-Controller* (MVC). *Model* adalah tempat mengelola data (*database*). *View* merupakan tampilan atau *presentation* dari sistem. *Controller* merupakan tempat menghubungkan *view* dengan *model*.

3.5 Pengujian dan Analisis

Pengujian sistem berguna untuk memastikan bahwa sistem dapat berjalan sesuai dengan kebutuhan sistem. Dalam pengujian sistem informasi manajemen aset ini menggunakan *black box testing* untuk pengujian validasi fungsionalitas dan *compatibility*. Dalam pengujian validasi fungsionalitas dilakukan pengujian dengan menjalankan sistem sesuai dengan fungsi-fungsi yang ada dengan melakukan percobaan terhadap semua fitur. Sedangkan pengujian *compatibility* adalah pengujian sistem agar dapat berjalan pada semua sistem operasi, *platform*, aplikasi ataupun jaringan yang berbeda. Dalam pengujian unit yang menggunakan metode *white box* yang digunakan untuk mengecek apakah komponen-komponen terkecil pada aplikasi yaitu kelas dan *method-method* didalamnya sudah berjalan sebagaimana mestinya. Serta melakukan pengujian integrasi untuk mengetahui apakah sistem yang dibangun dapat diintegrasikan dengan kelas lainnya.

3.6 Kesimpulan dan Saran

Tahap pengambilan kesimpulan dan saran dilaksanakan ketika proses analisis kebutuhan sampai tahap pengujian telah selesai dilakukan. Kesimpulan diambil dengan menjawab rumusan masalah yang telah dibuat. Sedangkan untuk saran dimaksudkan untuk memperbaiki kesalahan yang ada serta menyempurnakan penulisan dan memberikan pertimbangan.



BAB 4 REKAYASA KEBUTUHAN

Rekayasa kebutuhan merupakan proses penggalian kebutuhan diteruskan dengan menganalisis masalah yang terjadi serta memberikan solusi untuk kebutuhan pengguna yang harus ada dalam pembangunan sistem informasi. Proses rekayasa kebutuhan dimulai dengan melakukan tahapan elisitasi kebutuhan, analisis kebutuhan, spesifikasi kebutuhan, dan identifikasi aktor.

4.1 Elisitasi dan Analisis Kebutuhan

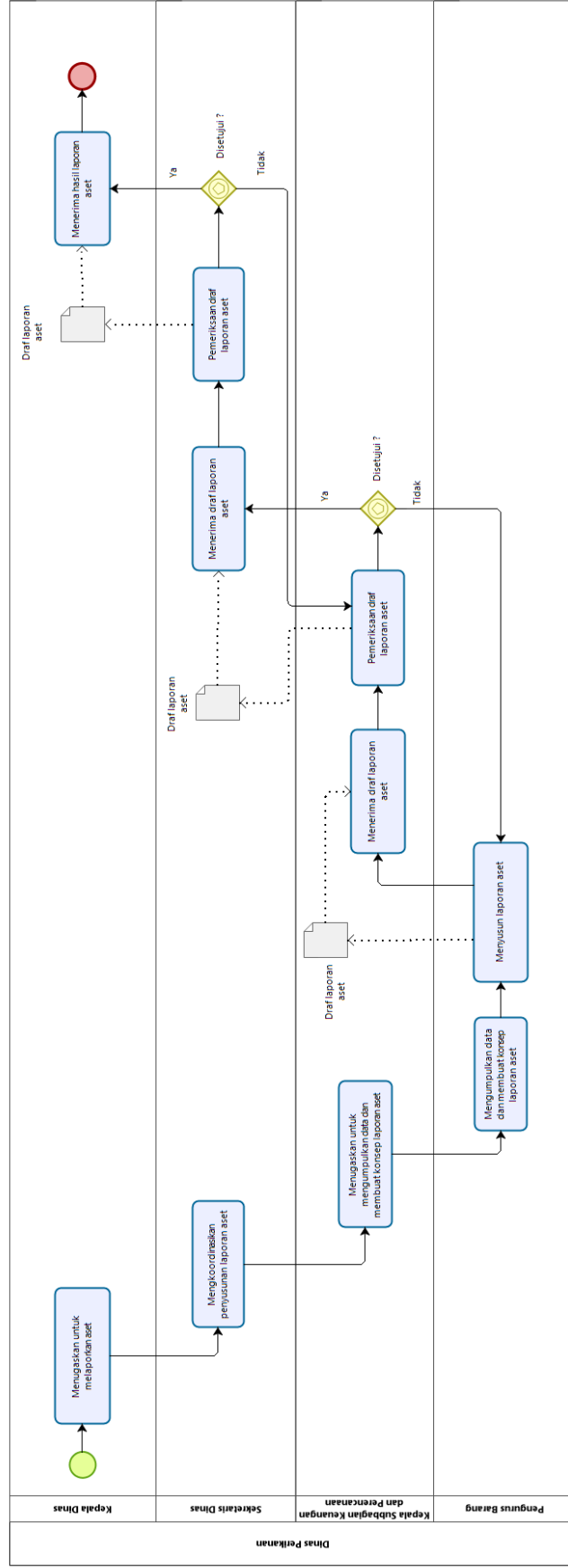
4.1.1 Elisitasi Kebutuhan

Pada tahap elisitasi kebutuhan, penulis mendapatkan hasil wawancara secara langsung dari Kepala Subbagian dan juga staf Dinas Perikanan Kabupaten Tabanan. Hasil wawancara digunakan untuk mengetahui alur proses bisnis saat ini dalam kegiatan pelaporan aset, pelaporan persediaan, pemeliharaan kendaraan operasional, pengecekan inventaris ruangan serta pencatatan penerimaan aset, yang akan dijelaskan sebagai berikut:

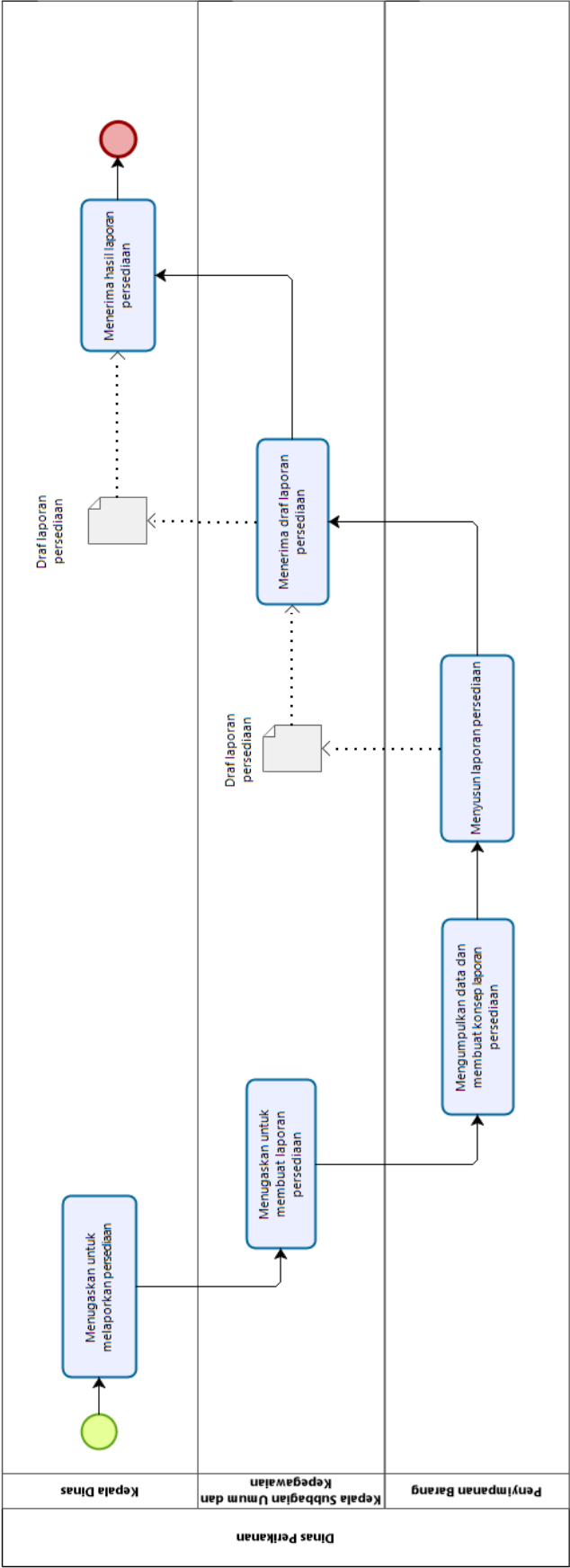
1. Dinas Perikanan merupakan Organisasi Perangkat Daerah (OPD) di lingkungan Pemerintah Kabupaten Tabanan terkait dalam memperdayakan usaha pembudidayaan ikan, memperdayakan usaha perikanan tangkap di laut dan perairan umum. Dinas perikanan memiliki divisi-divisi sesuai dengan bidang yang sudah diatur pada Peraturan Daerah Kabupaten Tabanan Nomor 13 Tahun 2016.
2. Dalam pelaporan aset, dibebankan kepada staf pengurus barang. Laporan aset ini dibagi menjadi empat kategori Kartu Inventaris Barang (KIB), yaitu KIB A yang berisikan aset tanah, KIB B berisikan aset peralatan dan mesin, KIB C berisikan aset gedung dan bangunan, dan KIB D berisikan jalan, jaringan dan juga irigasi. Staf pengurus barang mengumpulkan data dari semua bidang dan membuat konsep laporan aset selanjutnya laporan aset diserahkan kepada Kepala Dinas tetapi sebelumnya harus melewati tahap pemeriksaan laporan aset yang dilakukan Sekretaris Dinas beserta dengan Kepala Subbagian Keuangan dan Perencanaan. Proses bisnis pelaporan aset saat ini pada Dinas Perikanan Kabupaten Tabanan yang ada pada Gambar 4.1.
3. Pelaporan persediaan merupakan pelaporan mengenai aset habis pakai pada Dinas Perikanan Kabupaten Tabanan. Kegiatan pelaporan persediaan dibebankan kepada staf penyimpanan barang setelah data terkumpul dan konsep laporan sudah jadi selanjutnya dibuat pada *microsoft excel* berdasarkan data dan konsep yang telah dibuat. Pada Gambar 4.2 adalah proses bisnis untuk pelaporan persediaan. Kegiatan *order* persediaan yang dilakukan oleh bidang lain dengan mendatangi staf penyimpanan barang dan memberitahu persediaan yang akan diminta. Selanjutnya staf penyimpanan barang mencatat pada buku pengeluaran. Staf penyimpanan

barang menerbitkan Surat *Order* Barang (SOB) dan Surat Perintah Pengeluaran Barang (SPPB). Pada Gambar 4.3 adalah proses bisnis untuk *order* persediaan.

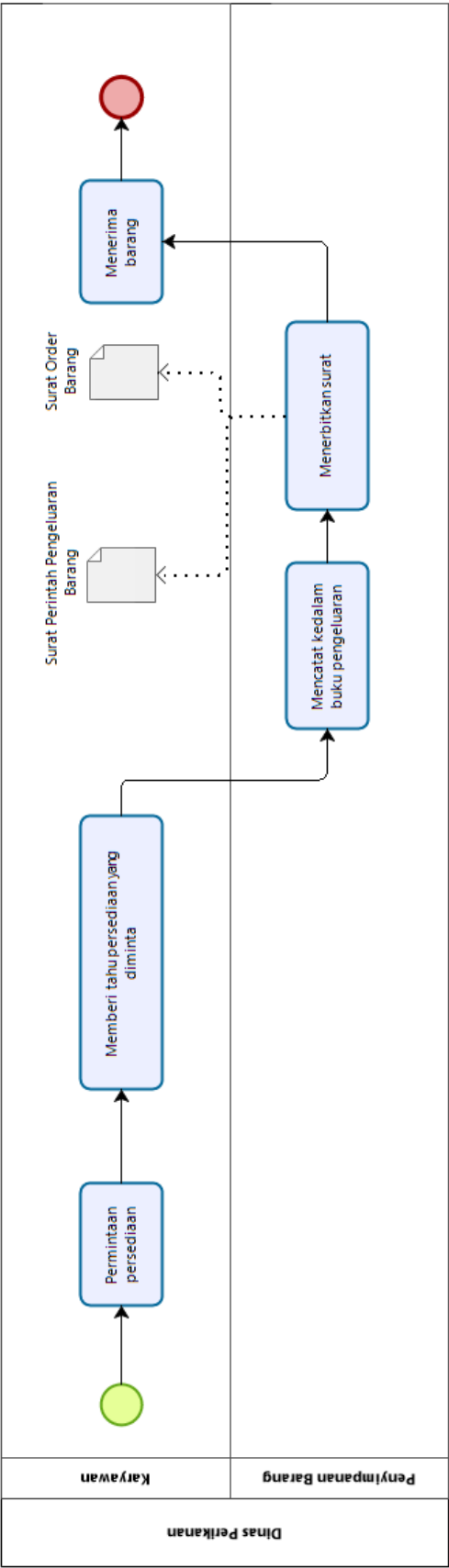
4. Untuk pemeliharaan kendaraan operasional dibebankan kepada pihak yang bertanggung jawab atas kendaraan yang dibawanya. Dengan menyerahkan bukti dari pemeliharaan serta bukti pembayaran pajak dalam bentuk laporan kepada staf pengurus barang dan mencatatnya dalam sebuah buku beserta buktinya. Dan untuk kendaraan yang tidak ada pemegangnya akan dibebankan kepada Dinas Perikanan Kabupaten Tabanan yang dilakukan oleh staf pengurus barang. Pemeliharaan kendaraan operasional digambarkan pada bisnis proses pada Gambar 4.4.
5. Pengecekan inventaris ruangan dilaksanakan setiap semester, yang melakukan pengecekan inventaris ruangan dilakukan oleh staf pengurus barang secara langsung turun ke lapangan kemudian membuat laporan pengecekan inventaris ruangan pada *microsoft excel* dan menyerahkan hasilnya kepada Kepala Subbagian Umum dan Kepegawaian untuk ditanda tangani. Pengecekan inventaris ruangan digambarkan pada Gambar 4.5.
6. Untuk penerimaan aset baru sebelumnya harus melewati pemeriksaan langsung dari tim pemeriksaan dari Dinas Perikanan Kabupaten Tabanan. Setelah tim pemeriksa selesai melakukan pemeriksaan dan disetujui maka akan diserahkan kepada Kepala Subbagian Umum dan Kepegawaian untuk koordinasi pendistribusian aset. Selanjutnya diserahkan kepada staf pengurus barang untuk dikelola dan dilaporkan pada periode berikutnya. Penerimaan aset baru digambarkan pada bisnis proses pada Gambar 4.6.



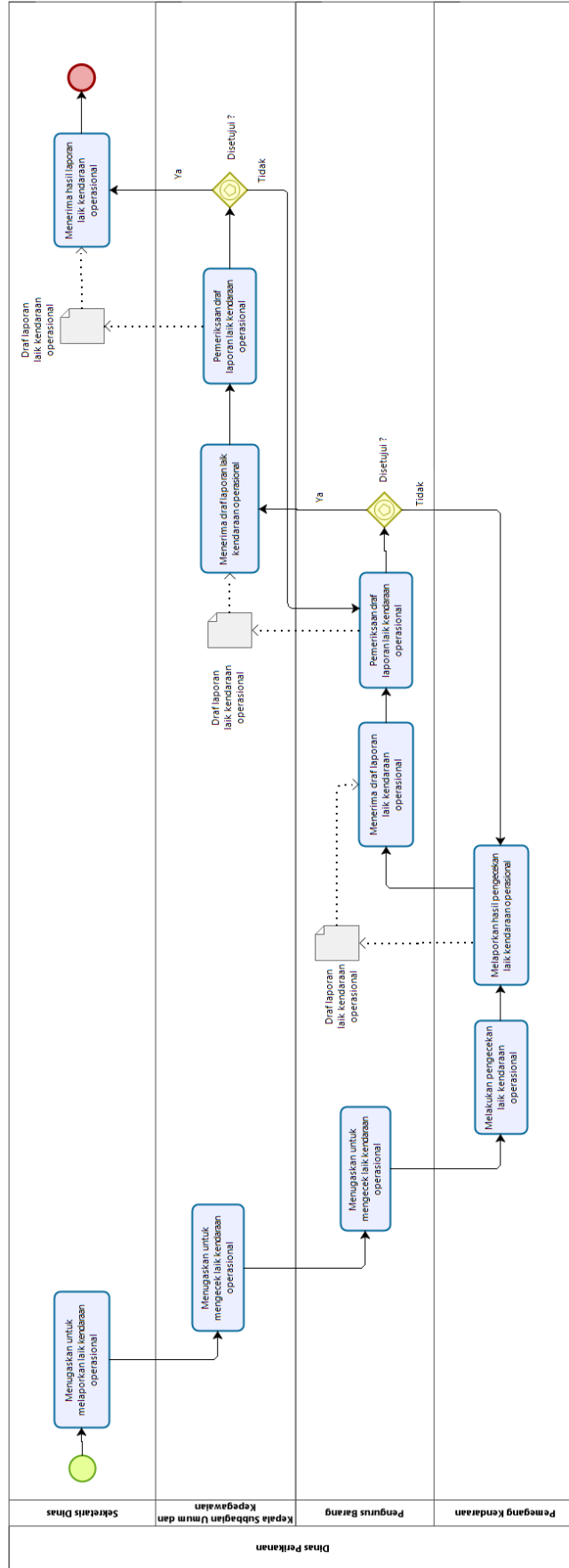
Gambar 4.1 Bisnis Proses Saat Ini (as-is) Pelaporan Aset



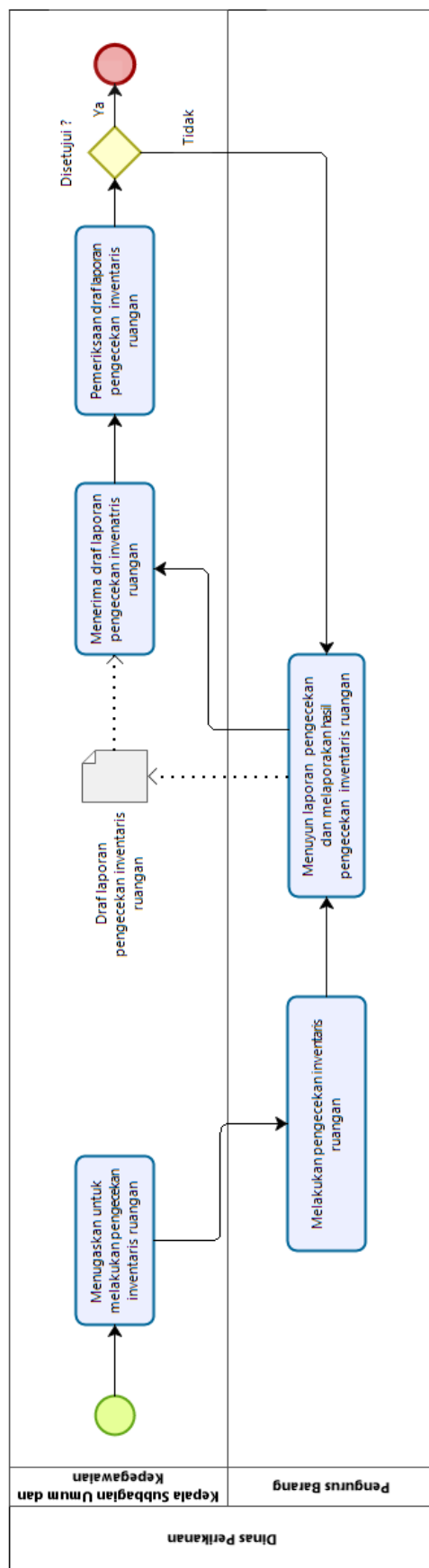
Gambar 4.2 Bisnis Proses Saat Ini (as-is) Pelaporan Persediaan



Gambar 4.3 Bisnis Proses Saat Ini (as-is) Order Persediaan

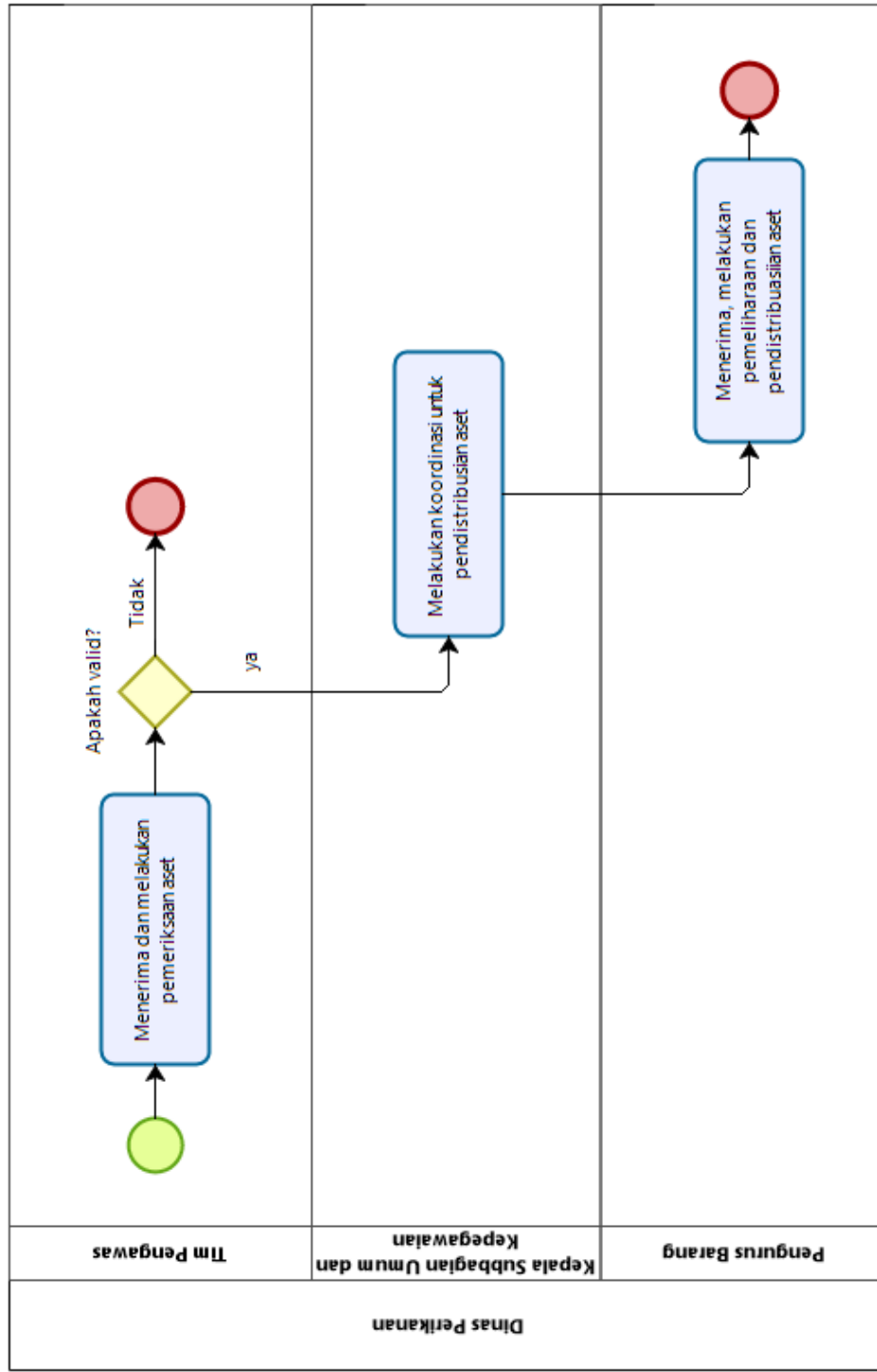


Gambar 4.4 Bisnis Proses Saat Ini (as-is) Pemeliharaan Kendaraan



Powered by
bizagi
Modeler

Gambar 4.5 Bisnis Proses Saat Ini (*as-is*) Pengecekan Inventaris Ruangan



Gambar 4.6 Bisnis Proses Saat Ini (as-is) Pencatatan Aset

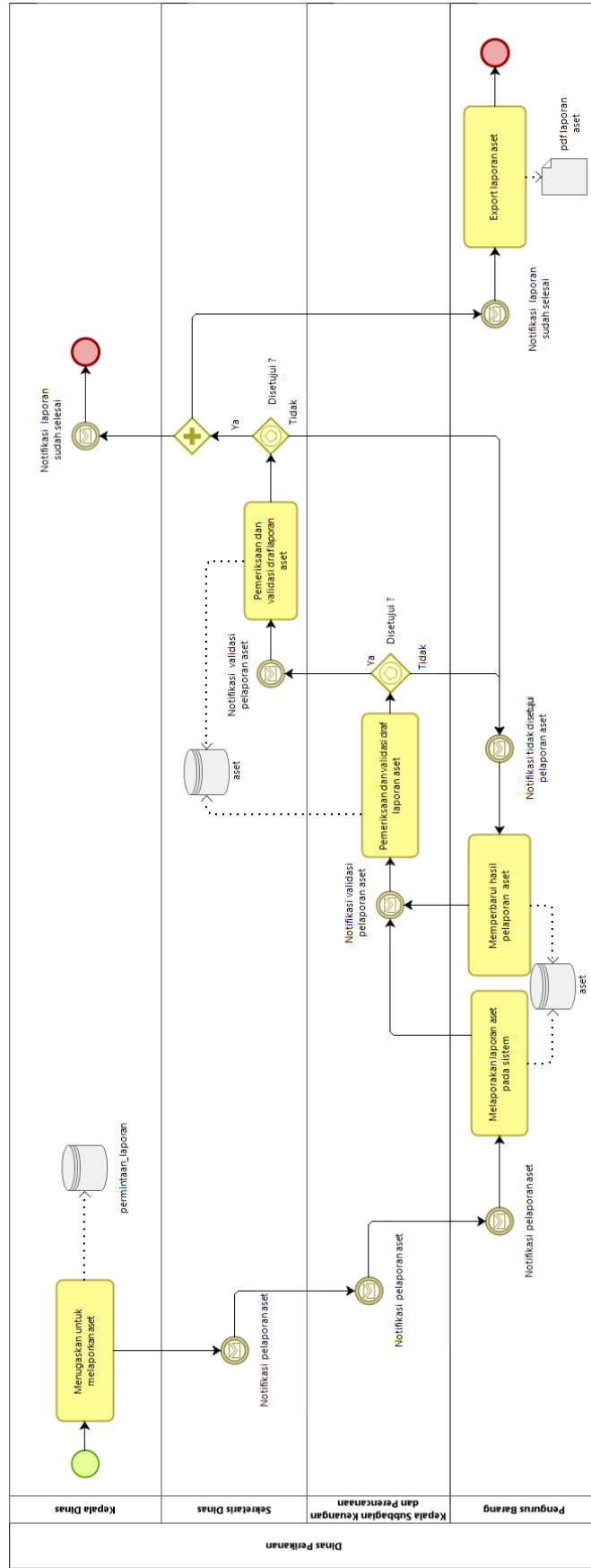
4.1.2 Analisis Kebutuhan

Penulis melakukan analisis kebutuhan berdasarkan *Business Process Modelling* (BPM) yang diperoleh dari hasil observasi dan wawancara. Berikut adalah hasilnya.

1. Pada Gambar 4.7 merupakan pemodelan sistem pelaporan aset. *Stakeholder* yang terlibat dalam pelaporan aset, yaitu Kepala Dinas, Sekretaris Dinas, Kepala Subbagian Keuangan dan Perencanaan serta staf pengurus barang. Kepala Dinas masuk ke dalam sistem dan memilih pelaporan aset selanjutnya sistem akan mengirimkan notifikasi pelaporan aset kepada Sekretaris Dinas, Kepala Subbagian Keuangan dan Perencanaan serta staf pengurus barang. Kemudian staf pengurus barang masuk ke laman pelaporan aset dan melaporkan aset. Sistem akan mengirimkan notifikasi *email* kepada Kepala Subbagian Keuangan dan Perencanaan untuk melakukan validasi laporan aset. Ketika sudah divalidasi akan diteruskan ke Sekretaris Dinas untuk divalidasi yang terakhir, yang sebelumnya sistem akan mengirimkan notifikasi validasi laporan aset kepada Sekretaris Dinas dan terakhir Kepala Dinas dan staf pengurus barang menerima notifikasi laporan aset telah selesai. Selanjutnya staf pengurus barang dapat meng-*export* laporan aset untuk disahkan.
2. Pada Gambar 4.8 merupakan pemodelan sistem pelaporan persediaan. Pada pelaporan persediaan, pertama Kepala Dinas meminta pembuatan laporan persediaan, selanjutnya sistem mengirimkan notifikasi pembuatan laporan persediaan kepada Kepala Subbagian Umum dan Kepegawaian dan staf penyimpanan barang. Staf penyimpanan barang masuk ke laman pelaporan persediaan dan melaporkan persediaan, kemudian meminta Kepala Subbagian Umum dan Kepegawaian untuk melakukan validasi laporan persediaan. Sistem secara otomatis akan memberikan notifikasi validasi laporan persediaan kepada Kepala Subbagian Umum dan Kepegawaian untuk melakukan validasi. Ketika pelaporan aset sudah selesai divalidasi maka sistem akan mengirimkan notifikasi kepada Kepala Dinas dan staf penyimpanan barang bahwa laporan persediaan sudah selesai. Kemudian staf penyimpanan barang dapat meng-*export* laporan persediaan untuk disahkan. Sedangkan untuk *order* persediaan dilakukan dengan mendatangi langsung staf penyimpanan barang dan memberitahu persediaan yang akan diminta selanjutnya staf penyimpanan barang memasukkan pada sistem dan menerbitkan surat *order* barang dan surat perintah pengeluaran barang. Persediaan diberikan ke bidang yang melakukan *order*. Untuk pemodelan sistem *order* persediaan pada Gambar 4.9.
3. Pada Gambar 4.10 merupakan pemodelan sistem pemeliharaan Kendaraan Operasional. Dilakukan oleh Sekretaris Dinas, Kepala Subbagian Umum dan Kepegawaian, staf pengurus barang, serta pemegang kendaraan. Sekretaris Dinas meminta pembuatan laporan pemeliharaan kendaraan operasional selanjutnya sistem mengirimkan notifikasi kepada

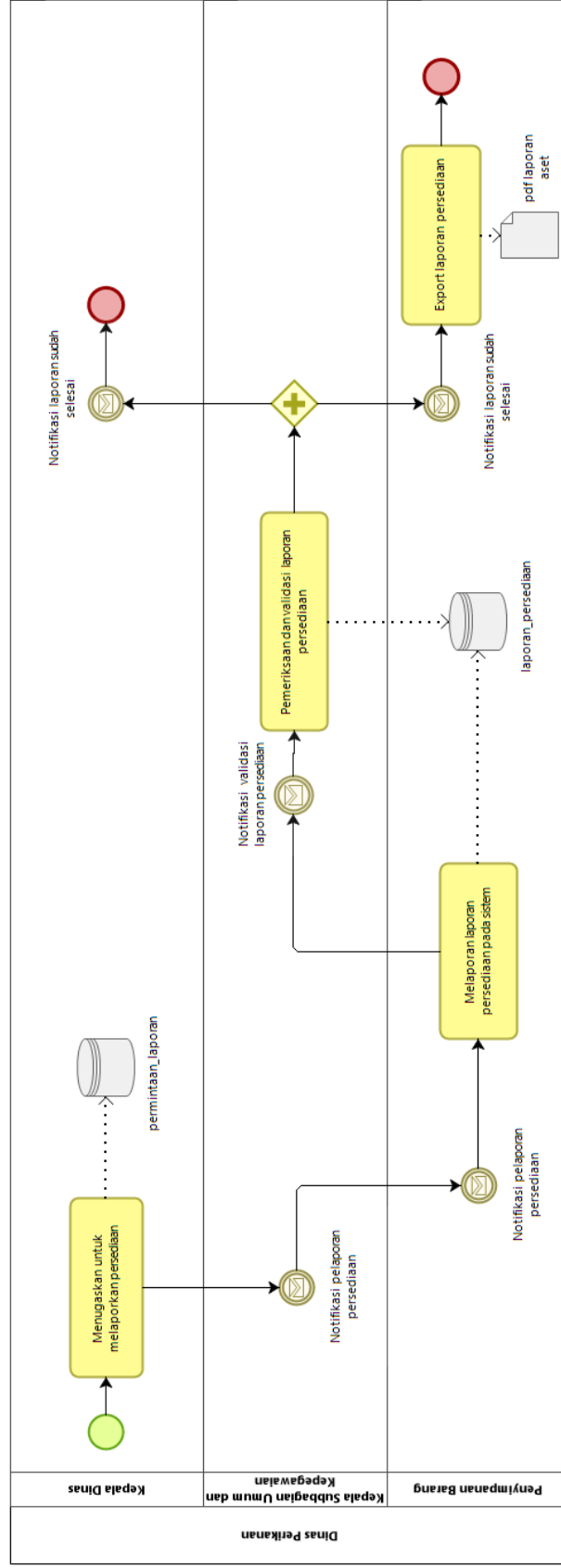
Kepala Subbagian Umum dan Kepegawaian, staf pengurus barang dan pemegang kendaraan. Pemegang kendaraan melakukan pengecekan laik kendaran dan membuat laporan pemeliharaan dengan memasukkan data-data pemeliharaan. Staf pengurus barang menerima notifikasi validasi laporan pemeliharaan kendaraan operasional dan melakukan validasi ketika tidak divalidasi akan dikembalikan ke pemegang kendaraan untuk diperbaiki. Apabila sudah disetujui dan divalidasi akan diberikan ke Kepala Subbagian Umum dan Kepegawaian untuk divalidasi yang sebelumnya sudah mendapatkan notifikasi validasi laporan pemeliharaan kendaraan operasional. Ketika sudah validasi sistem akan mengirimkan notifikasi ke Sekretaris Dinas dan pemegang kendaraan tentang laporan pemeliharaan kendaran operasional telah selesai. Kemudian pemegang kendaraan dapat meng-*export* laporan pemeliharaan kendaraan untuk disahkan.

4. Gambar 4.11 merupakan pemodelan sistem pengecekan inventaris ruangan. Pada pengecekan inventaris ruangan, pertama Kepala Subbagian Umum dan Kepegawaian meminta pembuatan laporan pengecekan inventaris ruangan, selanjutnya sistem mengirimkan notifikasi pembuatan laporan pengecekan inventaris ruangan kepada staf pengurus barang. Staf pengurus barang menerima notifikasi dan melakukan pengecekan inventaris ruangan ke lapangan. Staf pengurus barang masuk ke laman daftar pengecekan inventaris ruangan dan melihat daftar inventaris ruangan berdasarkan nama ruangan. Selanjutnya staf pengurus barang memasukkan hasil pengecekan inventaris ruangan dan melaporkannya ke Kepala Subbagian Umum dan Kepegawaian untuk divalidasi yang sebelumnya sudah menerima notifikasi validasi pelaporan pengecekan inventaris ruangan oleh sistem. Ketika laporan sudah selesai divalidasi oleh Kepala Subbagian Umum dan Kepegawaian akan mengirimkan notifikasi bahwa laporan sudah selesai divalidasi dan dapat melakukan *export* laporan pengecekan inventaris ruangan.
5. Pemodelan sistem pencatatan aset pada Gambar 4.12 dengan tim pemeriksa melakukan pemeriksaan terhadap aset dan sudah disetujui akan diberikan kepada Kepala Subbagian Umum dan Kepegawaian. Kemudian melakukan koordinasi dengan staf pengurus barang. Staf pengurus barang memasukkan data aset yang sudah selesai diperiksa ke dalam sistem dan staf pengurus barang melakukan pengelolaan terhadap aset tersebut.

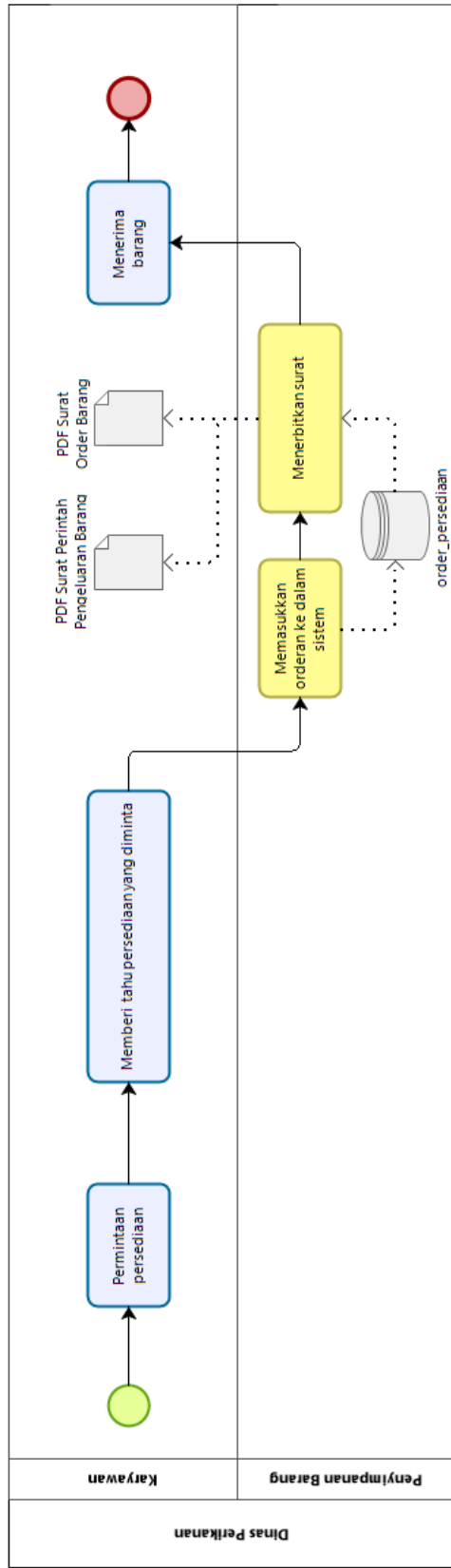


Powered by
bizagi
Modeler

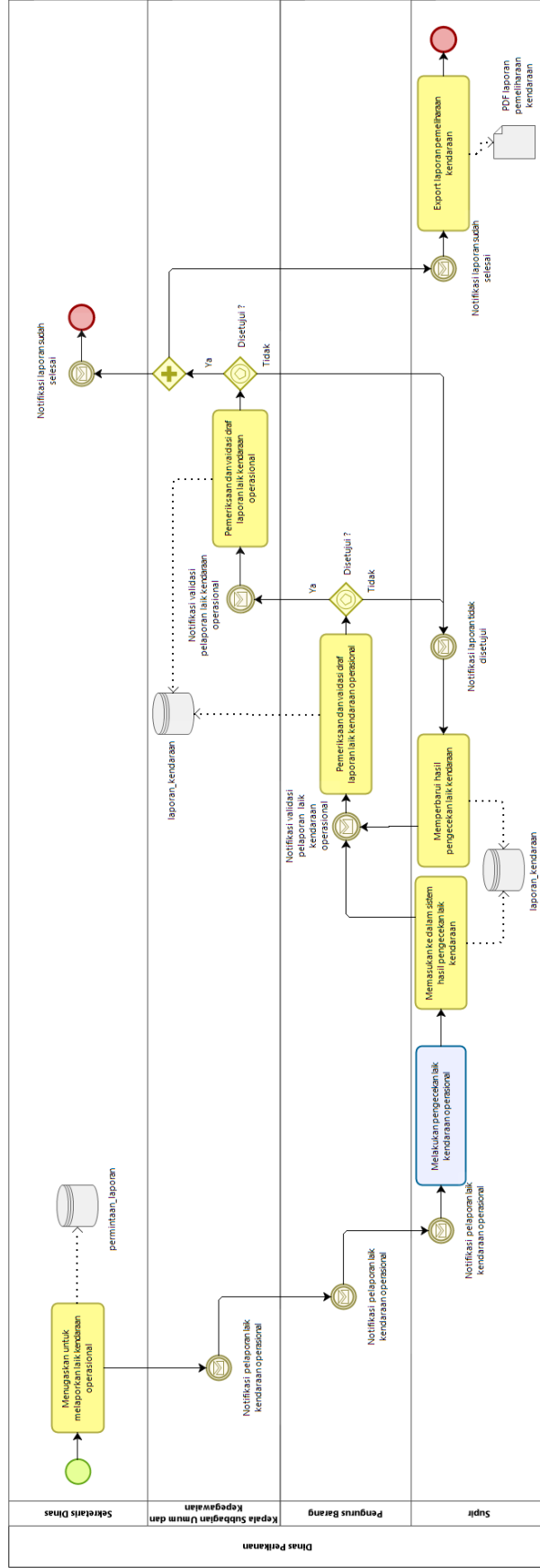
Gambar 4.7 Bisnis Proses *to-be* Pelaporan Aset



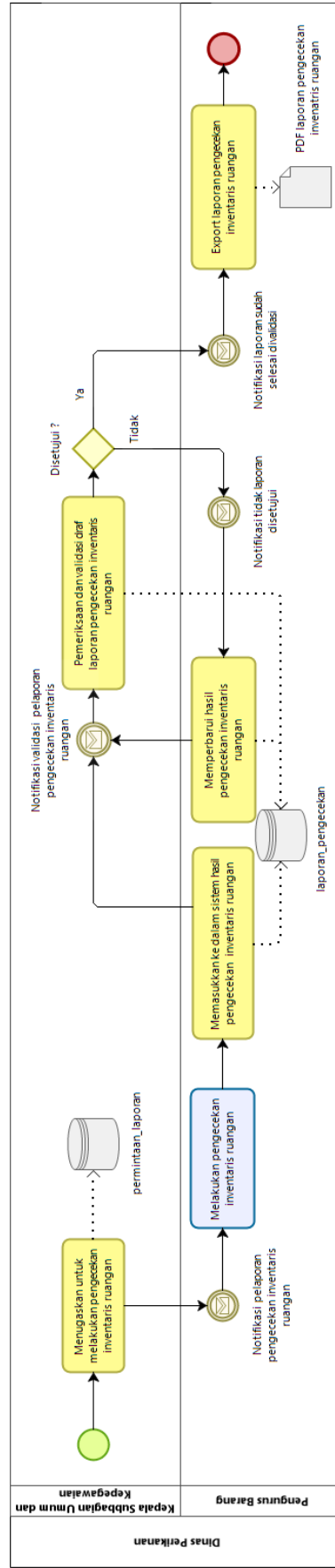
Gambar 4.8 Bisnis Proses *to-be* Pelaporan Persediaan



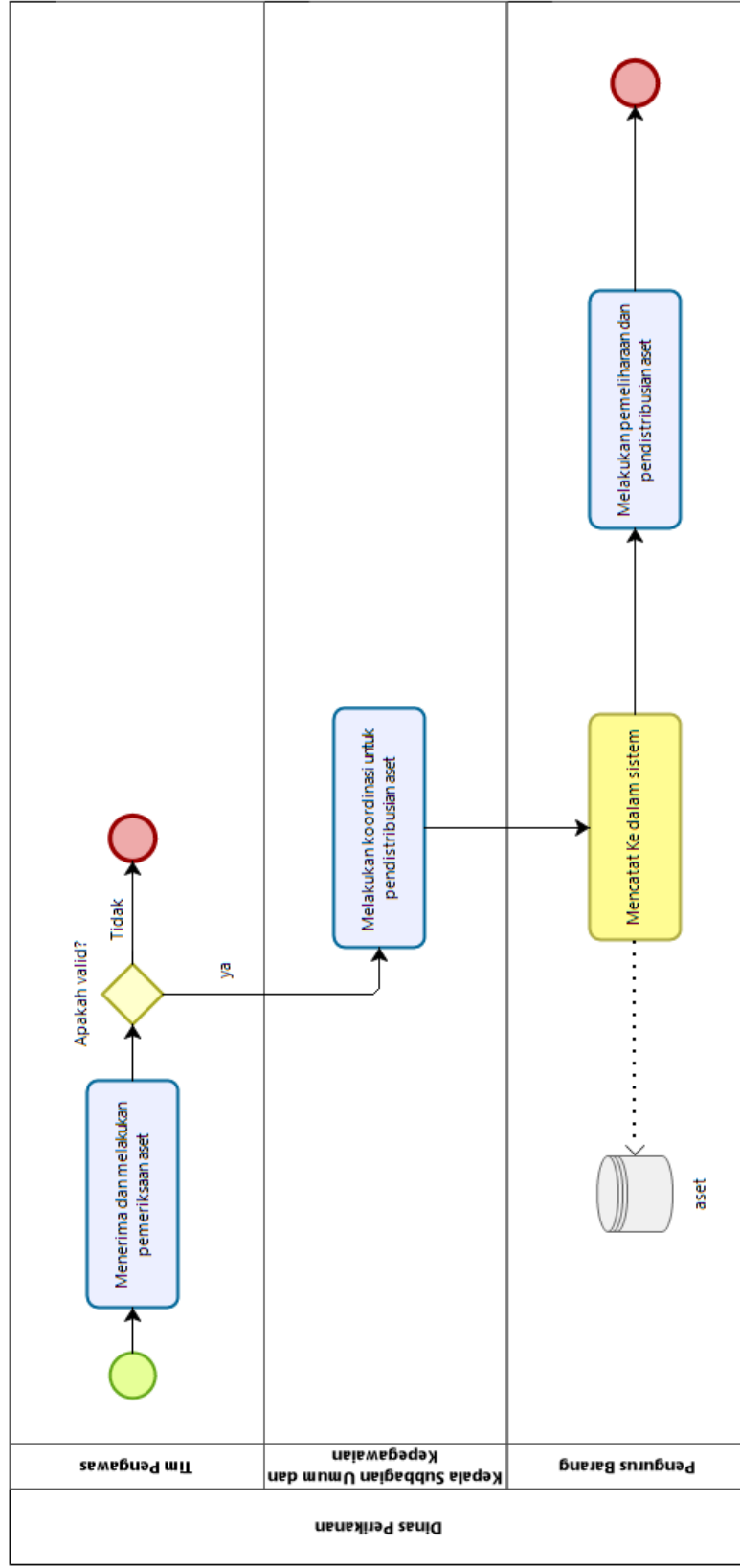
Gambar 4.9 Bisnis Proses to-be Order Persediaan



Gambar 4.10 Bisnis Proses to-be Pemeliharaan Kendaraan Operasional



Gambar 4.11 Bisnis Proses to-be Pengecekan Invenstris Ruangan



Gambar 4.12 Bisnis Proses *to-be* Pencatatan Aset

4.2 Gambaran Umum Sistem

Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan ini terdapat enam modul utama, yaitu mengenai pelaporan aset, pelaporan persediaan, pemeliharaan kendaraan operasional, pengecekan inventaris ruangan, *order* persediaan dan juga pencatatan penerimaan aset baru. Dengan adanya modul pelaporan aset dan persediaan yang nantinya dapat mempermudah dan mempercepat dalam kegiatan pelaporan dan pegawai lainnya mengetahui semua aset yang ada. Dengan adanya modul pemeliharaan kendaraan operasional dapat mempercepat pelaporan operasional yang dilakukan oleh pemegang kendaraan. Dengan adanya modul pengecekan inventaris ruangan maka dapat membantu kegiatan pengecekan inventaris ruangan dan dapat mengetahui keadaan aset. Dengan adanya modul pencatatan aset dapat mengetahui aset yang masuk ke dinas perikanan dan mempermudah dalam pembuatan laporan.

4.3 Identifikasi Aktor

Dari hasil elisitasi kebutuhan, mendapatkan aktor yang terlibat pada sistem. Aktor merupakan orang atau sistem lain yang berinteraksi dengan sistem. Pada sistem informasi manajemen aset ini terdapat beberapa aktor yang terlibat, yaitu: Pengguna, Kepala Dinas, Sekretaris Dinas, Kepala Subbagian Umum dan Kepegawaian, Kepala Subbagian Keuangan dan Perencanaan, pengurus barang, penyimpanan barang, pemegang kendaraan, dan admin. Untuk penjelasan masing-masing aktor pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Pengguna	Merupakan aktor yang belum terdaftar pada sistem dan dapat melihat semua daftar aset, persediaan dan daftar pemegang kendaraan dinas.
2	Kepala Dinas	Merupakan aktor yang dapat melakukan permintaan pembuatan laporan aset dan laporan persediaan.
3	Sekretaris Dinas	Merupakan aktor yang dapat melakukan permintaan pembuatan laporan kendaraan operasional dan juga melakukan validasi laporan aset.
4	Kepala Subbagian Umum dan Kepegawaian	Merupakan aktor yang dapat melakukan permintaan pembuatan laporan pengecekan inventaris ruangan dan melakukan validasi laporan persediaan, laporan pemeliharaan kendaraan operasional,

Tabel 4.1 Identifikasi Aktor (Lanjutan)

No	Aktor	Deskripsi
		dan laporan pengecekan inventaris ruangan.
5	Kepala Subbagian Keuangan dan Perencanaan	Merupakan aktor yang melakukan validasi terhadap pembuatan laporan aset.
6	Pengurus Barang	Merupakan aktor yang membuat laporan aset dan laporan pengecekan inventaris ruangan. Serta mengelola aset baik itu memasukkan, memperbarui dan juga menghapus aset pada sistem, serta memasukkan daftar pemegang kendaraan.
7	Penyimpanan Barang	Merupakan aktor yang bertanggung jawab terhadap persediaan baik itu persediaan masuk maupun keluar.
8	Pemegang Kendaraan	Merupakan aktor yang bertanggung jawab atas pelaporan kendaraan operasional yang dipegang.
9	Admin	Merupakan aktor yang bertanggung jawab untuk membuka akses untuk pengguna yang belum terdaftar.

4.4 Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem adalah kebutuhan dari sistem yang harus ada, bagaimana sistem merespon terhadap masukan tertentu dan bagaimana sistem harus bertindak apa pada situasi tertentu (Sommerville, 2011). Untuk mempermudah kodifikasi dan *traceability* digunakan penomoran untuk setiap kebutuhan fungsional, yaitu menggunakan kode SIMA-PFS.X.Y dan untuk kebutuhan non fungsional diberikan kode SIMA-PNFS.X. SIMA adalah singkatan dari Sistem Informasi Manajemen Aset. PFS untuk kebutuhan persyaratan fungsional sistem, PNFS untuk persyaratan non fungsional sistem, X mendefinisikan kebutuhan utama dan Y mendefinisikan spesifikasi kebutuhan.

Tabel 4.2 Kebutuhan Fungsional

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
1	SIMA-PFS.01	Sistem dapat melakukan fungsi untuk mendaftarkan pengguna ke sistem.	

Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
1.1	SIMA-PFS.01.1	Sistem dapat mengirimkan notifikasi berupa <i>e-mail</i> kepada pengguna mengenai <i>account</i> pengguna.	Mendaftarkan pengguna
2	SIMA-PFS.02	Sistem menyediakan akses kepada pengguna yang terdaftar.	Login
2.1	SIMA-PFS.02-1	Sistem harus dapat menyediakan <i>form login</i> .	
3	SIMA-PFS.03	Sistem harus dapat menampilkan daftar pengguna.	Daftar pengguna
4	SIMA-PFS.04	Sistem dapat melakukan fungsi memperbarui Pengguna.	Sunting pengguna
5	SIMA-PFS.05	Sistem dapat melakukan fungsi menghapus pengguna.	Hapus pengguna
6	SIMA-PFS.06	Sistem dapat melakukan fungsi lihat daftar aset.	Daftar aset
7	SIMA-PFS.07	Sistem dapat melakukan fungsi menambahkan aset.	Tambah aset
8	SIMA-PFS.08	Sistem dapat melakukan fungsi memperbarui data aset.	Sunting aset
9	SIMA-PFS.09	Sistem dapat melakukan fungsi menghapus aset.	Hapus aset
10	SIMA-PFS.10	Sistem dapat melakukan fungsi permintaan pembuatan laporan aset.	Permintaan laporan aset
10.1	SIMA-PFS.10.1	Sistem dapat mengirimkan notifikasi berupa <i>e-mail</i> kepada sekretaris dinas.	
		Sistem dapat mengirimkan notifikasi <i>e-mail</i> ke kepala subagian keuangan dan perencanaan.	
		Sistem dapat mengirimkan notifikasi <i>e-mail</i> ke pengurus barang untuk pembuatan laporan aset.	
11	SIMA-PFS.11	Sistem dapat melakukan fungsi lihat daftar laporan aset.	Daftar laporan aset
12	SIMA-PFS.12	Sistem dapat melakukan fungsi memperbarui data laporan aset.	Sunting laporan aset



Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
13	SIMA-PFS.13	Sistem dapat melakukan fungsi membuat laporan aset.	Laporan aset
13.1	SIMA-PFS.13.1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> mengenai validasi laporan aset ke Kasubag Keuangan dan Sekretaris dinas.	Laporan aset
14	SIMA-PFS.14	Sistem dapat melakukan fungsi lihat daftar validasi aset.	Daftar validasi aset
15	SIMA-PFS.15	Sistem dapat melakukan fungsi melakukan validasi laporan aset.	Validasi laporan aset
15.1	SIMA-PFS.15-1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> mengenai validasi laporan aset ke sekretaris dinas.	
16	SIMA-PFS.16	Sistem dapat melakukan fungsi meng- <i>export</i> laporan aset.	<i>Export</i> laporan aset
17	SIMA-PFS.17	Sistem harus dapat menampilkan semua daftar persediaan.	Daftar persediaan
18	SIMA-PFS.18	Sistem dapat melakukan fungsi menambahkan persediaan.	Tambah persediaan
19	SIMA-PFS.19	Sistem dapat melakukan fungsi memperbarui data persediaan.	Sunting persediaan
20	SIMA-PFS.20	Sistem dapat melakukan fungsi <i>order</i> persediaan.	<i>Order</i> persediaan
21	SIMA-PFS.21	Sistem dapat melakukan fungsi lihat daftar pengeluaran	Daftar pengeluaran
22	SIMA-PFS.22	Sistem dapat melakukan fungsi penerbitan surat.	Penerbitan surat
23	SIMA-PFS.23	Sistem dapat melakukan fungsi hapus <i>order</i> persediaan.	Hapus <i>order</i> persediaan
24	SIMA-PFS.24	Sistem dapat melakukan fungsi pembuatan laporan persediaan.	Permintaan laporan persediaan
24.1	SIMA-PFS.24-1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> ke kasubag umum dan perencanaan.	
25	SIMA-PFS.25	Sistem dapat melakukan fungsi lihat daftar laporan persediaan.	Daftar laporan persediaan

Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
26	SIMA-PFS.26	Sistem dapat melakukan fungsi membuat laporan persediaan.	Laporan persediaan
26.1	SIMA-PFS.26-1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> mengenai validasi laporan persediaan ke penyimpanan barang dan kasubag umum dan kepegawaian.	Laporan persediaan
27	SIMA-PFS.27	Sistem dapat melakukan fungsi lihat daftar validasi persediaan.	Daftar validasi persediaan
28	SIMA-PFS.28	Sistem dapat melakukan fungsi melakukan validasi laporan persediaan.	Validasi laporan persediaan
28.1	SIMA-PFS.28-1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> mengenai validasi laporan aset ke kasubag umum dan kepegawaian.	
		Sistem dapat mengirimkan notifikasi laporan aset sudah selesai divalidasi berupa <i>e-mail</i> kepada kepala dinas.	
29	SIMA-PFS.29	Sistem dapat melakukan fungsi meng- <i>export</i> laporan persediaan.	<i>Export</i> laporan persediaan
30	SIMA-PFS.30	Sistem dapat melakukan fungsi menambahkan pemegang kendaraan.	Tambah pemegang kendaraan
31	SIMA-PFS.31	Sistem dapat melakukan fungsi lihat daftar pemegang kendaraan.	Daftar pemegang kendaraan
32	SIMA-PFS.32	Sistem dapat melakukan fungsi menghapus pemegang kendaraan.	Hapus kendaraan
33	SIMA-PFS.33	Sistem dapat melakukan fungsi pembuatan laporan pemeliharaan kendaraan.	Permintaan laporan pemeliharaan kendaraan
33.1	SIMA-PFS.33.1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> ke kasubag umum &	
33.1	SIMA-PFS.33.1	perencanaan, pengurus barang, dan pemegang kendaraan.	
34	SIMA-PFS.34	Sistem dapat melakukan fungsi lihat daftar pemeliharaan kendaraan.	Daftar pemeliharaan kendaraan

Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
35	SIMA-PFS.35	Sistem dapat melakukan fungsi memperbarui data pemeliharaan kendaraan kendaraan.	Perbarui pemeliharaan kendaraan
36	SIMA-PFS.36	Sistem dapat melakukan fungsi menghapus data hasil pemeliharaan kendaraan.	Hapus hasil pemeliharaan kendaraan
37	SIMA-PFS.37	Sistem dapat melakukan fungsi membuat laporan pemeliharaan kendaraan.	Laporan pemeliharaan kendaraan
37.1	SIMA-PFS.37-1	Sistem dapat mengirimkan notifikasi validasi laporan pemeliharaan kendaraan operasional berupa <i>e-mail</i> kepada pengurus barang.	Laporan pemeliharaan kendaraan
38	SIMA-PFS.38	Sistem dapat melakukan fungsi lihat daftar validasi pemeliharaan kendaraan.	Daftar validasi pemeliharaan kendaraan
39	SIMA-PFS.39	Sistem dapat melakukan fungsi melakukan validasi laporan pemeliharaan kendaraan.	Validasi laporan pemeliharaan kendaraan
39.1	SIMA-PFS.39-1	Sistem dapat mengirimkan notifikasi validasi laporan pemeliharaan kendaraan operasional berupa <i>e-mail</i> kepada kasubag umum dan perencanaan.	
		Sistem dapat mengirimkan notifikasi laporan pemeliharaan kendaraan operasional sudah selesai divalidasi berupa <i>e-mail</i> kepada sekretaris dinas.	
40	SIMA-PFS.40	Sistem dapat melakukan fungsi hasil pemeliharaan kendaraan.	Daftar hasil pemeliharaan kendaraan
41	SIMA-PFS.41	Sistem dapat melakukan fungsi meng- <i>export</i> laporan pemeliharaan kendaraan.	Export laporan pemeliharaan kendaraan
42	SIMA-PFS.42	Sistem dapat melakukan fungsi daftar pengecekan inventaris ruangan.	Daftar pengecekan inventaris ruangan

Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
43	SIMA-PFS.43	Sistem dapat melakukan fungsi tambah aset pada inventaris ruangan.	Tambah aset pada inventaris ruangan
44	SIMA-PFS.44	Sistem dapat melakukan fungsi hapus aset pada inventaris ruangan.	Hapus aset pada inventaris ruangan
45	SIMA-PFS.45	Sistem dapat melakukan fungsi pembuatan pengecekan inventaris ruangan.	Permintaan laporan pengecekan inventaris ruangan
45.1	SIMA-PFS.45.1	Sistem dapat mengirimkan notifikasi <i>e-mail</i> ke Kepala Subbagian Umum & Kepegawaian.	
46	SIMA-PFS.46	Sistem dapat melakukan fungsi memperbarui pengecekan inventaris ruangan.	Perbarui data inventaris ruangan
47	SIMA-PFS.47	Sistem dapat melakukan fungsi lihat daftar hasil pengecekan inventaris ruangan.	Daftar hasil pengecekan inventaris ruangan
48	SIMA-PFS.48	Sistem dapat melakukan fungsi membuat laporan pengecekan inventaris ruangan.	Laporan pengecekan inventaris ruangan
48.1	SIMA-PFS.48.1	Sistem dapat mengirimkan notifikasi validasi laporan pengecekan inventaris ruangan berupa <i>e-mail</i> kepada Kasubag umum dan kepegawaian.	
49	SIMA-PFS.49	Sistem dapat melakukan fungsi memperbarui data hasil pengecekan inventaris ruangan.	Sunting hasil pengecekan inventaris ruangan
50	SIMA-PFS.50	Sistem dapat melakukan fungsi lihat daftar validasi pengecekan inventaris ruangan.	Daftar validasi pengecekan inventaris ruangan
51	SIMA-PFS.51	Sistem dapat melakukan fungsi melakukan validasi laporan pengecekan inventaris ruangan.	Validasi laporan pengecekan inventaris ruangan

Tabel 4.2 Kebutuhan Fungsional (Lanjutan)

No	Kode Kebutuhan	Deskripsi Kebutuhan	Deskripsi
51.1	SIMA-PFS.51-1	Sistem dapat mengirimkan notifikasi laporan pengecekan inventaris ruangan sudah selesai divalidasi berupa <i>e-mail</i> kepada pengurus barang.	Validasi laporan pengecekan inventaris ruangan
52	SIMA-PFS.52	Sistem dapat melakukan fungsi meng- <i>export</i> laporan pengecekan inventaris ruangan.	<i>Export</i> hasil laporan pengecekan invenatris ruangan
53	SIMA-PFS.53	Sistem dapat melakukan fungsi lihat daftar aset bagi pengguna.	Lihat daftar aset
54	SIMA-PFS.54	Sistem dapat melakukan fungsi lihat daftar persediaan bagi pengguna.	Lihat daftar persediaan
55	SIMA-PFS.55	Sistem dapat melakukan fungsi lihat daftar pemegang kendaraan bagi pengguna.	Lihat daftar pemegang kendaraan
56	SIMA-PFS.56	Sistem dapat melakukan fungsi keluar dari sistem.	<i>Logout</i>

Tabel 4.3 Kebutuhan Non Fungsional

Notasi	Nama	Deskripsi Kebutuhan	Deskripsi
1	SIMA-PNFS.1	<i>Compatibility</i>	Sistem bisa dijalankan dengan baik sesuai dengan fungsionalitas pada semua <i>platform</i> .

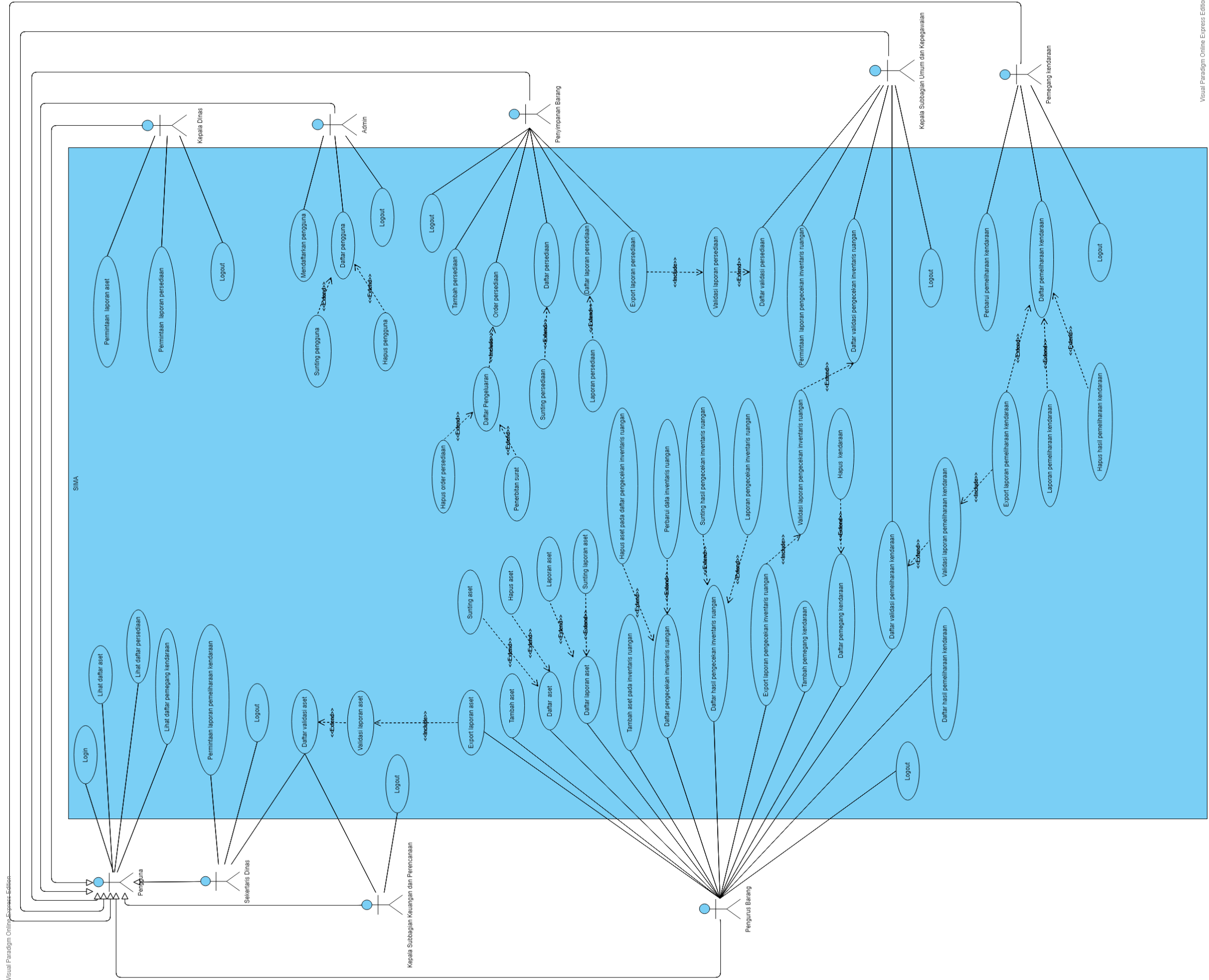
4.5 Pemodelan Kebutuhan

4.5.1 Use Case Diagram

Use Case Diagram merupakan pemodelan perilaku aktor dengan sistem berdasarkan kebutuhan fungsional sebelumnya. Berikut *use case diagram* sistem pada Gambar 4.13.

4.5.2 Use Case Scenario

Use Case Scenario menjelaskan lebih detail mengenai *use case diagram*. Ini memberikan penjabaran mengenai *use case*, tujuan dari *use case*, aktor yang terlibat, *pre-condition*, *main-flow* dari proses, *alternative flow* dan *post-condition* adalah hasil akhir yang ingin dicapai.



Gambar 4.13 Use Diagram Sistem Informasi Manajemen Aset Pada Dinas Perikanan Kabupaten Tabanan

4.5.2.1 Use Case Scenario Mendaftarkan Pengguna

Pada Tabel 4.4 terdapat *use case scenario* mendaftar pengguna.

Tabel 4.4 Use Case Scenario Mendaftarkan Pengguna

Flow of Events for Mendaftarkan Pengguna	
Kode Kebutuhan	SIMA-PFS.01
Objective	Pengguna didaftarkan ke sistem.
Aktor	Admin.
Pre Condition	1. Pengguna belum terdaftar di SIMA. 2. Admin sudah berada pada laman admin.
Main flow	1. Admin memilih menu "Pengguna" di <i>sidebar</i> menu. 2. Admin menekan "Daftar Pengguna" pada <i>sidebar</i> menu. 3. Sistem melihat daftar pengguna. 4. Admin menekan tanda <i>plus</i> pada bagian kanan tulisan "Tambah Karyawan". 5. Sistem menampilkan laman yang berisikan <i>field</i> yang terdiri dari nama, nomor induk pegawai (NIP/NIK), asal unit/bagian, <i>e-mail</i>, nomor telpon, <i>username</i> dan <i>password</i>. 6. Admin mengisi <i>field</i> yang tersedia secara lengkap. 7. Admin menekan <i>button</i> "Simpan". 8. Sistem melihatkan pesan sukses.
Alternative Flow	6.1 Jika <i>field</i> tambah pengguna ada yang tidak terisi akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> yang belum terisi.
Post Condition	Admin berhasil mendaftar pengguna.

4.5.2.2 Use Case Scenario Login

Pada Tabel 4.5 menggambarkan perilaku aktor mengenai tahapan pada *use case login*.

Tabel 4.5 Use Case Scenario Login

Flow of Events for Login	
Kode Kebutuhan	SIMA-PFS.02
Objective	Pengguna bisa masuk ke SIMA.
Aktor	Pengguna.

Tabel 4.5 Use Case Scenario Login (Lanjutan)

Flow of Events for Login	
Kode Kebutuhan	SIMA-PFS.02
Pre Condition	1. Pengguna sudah terdaftar. 2. Pengguna berada pada laman awal.
Main flow	1. Pengguna menekan tombol “Sign In”. 2. Sistem melihatkan laman <i>login</i> yang berisi <i>field</i> yang terdiri dari <i>username</i> dan <i>password</i>. 3. Pengguna memasukkan <i>username</i> dan <i>password</i>. 4. Pengguna menekan “Sign In”. 5. Sistem melihatkan laman <i>homepage</i> sesuai dengan pengguna yang <i>Sign In</i>.
Alternative Flow	5.1 Bila <i>username</i> tidak terdaftar akan menampilkan pesan “Username salah!”. 5.2 Bila <i>password</i> tidak sesuai akan menampilkan pesan “Password salah!”.
Post Condition	Pengguna berhasil masuk ke SIMA.

4.5.2.3 Use Case Scenario Daftar Pengguna

Pada Tabel 4.6 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar pengguna.

Tabel 4.6 Use Case Scenario Daftar Pengguna

Flow of Events for Daftar Pengguna	
Kode Kebutuhan	SIMA-PFS.03
Objective	Admin bisa lihat semua daftar pengguna.
Aktor	Admin.
Pre Condition	Admin berada pada SIMA.
Main flow	1. Admin memilih “Pengguna” pada <i>sidebar menu</i>. 2. Admin menekan submenu “Daftar Pengguna” pada <i>sidebar menu</i>. 3. Sistem melihatkan daftar pengguna.
Alternative Flow	-
Post Condition	Admin bisa lihat daftar pengguna.
Objective	Admin bisa lihat daftar pengguna.

4.5.2.4 Use Case Scenario Sunting Pengguna

Pada Tabel 4.7 menggambarkan perilaku aktor mengenai tahapan pada *use case* sunting pengguna.

Tabel 4.7 Use Case Scenario Sunting Pengguna

Flow of Events for Sunting Pengguna	
Kode Kebutuhan	SIMA-PFS.04
Objective	Admin memperbarui pengguna yang sudah terdaftar.
Aktor	Admin
Pre Condition	Admin sudah dalam laman admin.
Main flow	1. Admin menekan “Pengguna” pada <i>sidebar</i> menu. 2. Admin menekan submenu “Daftar Pengguna” pada <i>sidebar</i> menu. 3. Sistem melihatkan daftar pengguna. 4. Admin memilih pengguna yang akan diperbarui dengan menekan <i>button</i> “Sunting”. 5. Sistem melihatkan <i>pop-up detail</i> dari pengguna. 6. Admin mengisi <i>field</i> data pengguna yang akan diperbarui sesuai dengan <i>field</i> yang tersedia dan simpan. 7. Sistem melihatkan pesan sukses.
Alternative Flow	6.1 Jika admin tidak mengisikan <i>field</i> yang tersedia maka akan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Post Condition	Admin bisa memperbarui pengguna.

4.5.2.5 Use Case Scenario Hapus Pengguna

Pada Tabel 4.8 menggambarkan perilaku aktor mengenai tahapan pada *use case* hapus pengguna.

Tabel 4.8 Use Case Scenario Hapus Pengguna

Flow of Events for Hapus Pengguna	
Kode Kebutuhan	SIMA-PFS.05
Objective	Admin bisa menghapus pengguna yang ada pada sistem.
Aktor	Admin.
Pre Condition	Admin berada pada laman admin.
Main flow	1. Admin menekan “Pengguna” pada <i>sidebar menu</i>.

Tabel 4.8 Use Case Scenario Hapus Pengguna (Lanjutan)

Flow of Events for Hapus Pengguna	
Kode Kebutuhan	SIMA-PFS.05
	- Admin menekan submenu “Daftar Pengguna” pada <i>sidebar menu</i>. - Sistem menampilkan daftar pengguna. - Admin memilih pengguna yang akan dihapus. - Admin menekan tombol “Hapus” pada daftar pengguna. - Sistem akan melihatkan pesan “Yakin akan menghapus karyawan ini”. - Admin menekan “Ok”. - Sistem menampilkan pesan sukses.
Alternative Flow	-
Post Condition	Admin berhasil menghapus pengguna.

4.5.2.6 Use Case Scenario Daftar Aset

Pada Tabel 4.9 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar aset.

Tabel 4.9 Use Case Scenario Daftar Aset

Flow of Events for Daftar Aset	
Kode Kebutuhan	SIMA-PFS.06
Objective	Pengurus barang bisa lihat daftar aset pada sistem.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	- Pengurus barang menekan “Aset Dinas” pada <i>sidebar menu</i>. - Pengurus barang menekan submenu “Daftar Aset Dinas” pada <i>sidebar menu</i>. - Sistem melihatkan laman daftar aset.
Alternative Flow	-
Post Condition	Pengurus bisa lihat daftar aset.



4.5.2.7 Use Case Scenario Tambah Aset

Pada Tabel 4.10 menggambarkan perilaku aktor mengenai tahapan pada *use case* tambah aset.

Tabel 4.10 Use Case Scenario Tambah Aset

Flow of Events for Tambah Aset	
Kode Kebutuhan	SIMA-PFS.07
Objective	Pengurus barang bisa menambahkan aset.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Aset Dinas” pada <i>sidebar</i> menu. 2. Pengurus menekan submenu “Daftar Aset” pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar aset. 4. Pengurus barang menekan <i>dropdown</i> Masukan Kategori Aset. 5. Sistem menampilkan kategori aset. 6. Pengurus barang memilih kategori sesuai dengan kategori aset yang akan dimasukkan. 7. Sistem menampilkan <i>field form</i> yang berisikan tempat untuk memasukkan tipe aset, nama aset, nomor registrasi dan yang lainnya sesuai dengan pilihan kategori yang dipilih pada menu <i>dropown</i>. 8. Pengurus barang mengisi <i>form</i> tambah aset sesuai dengan <i>field</i> yang tersedia seperti tipe aset, nama aset, nomor registrasi sesuai dengan pilihan kategori yang dipilih pada menu <i>dropown</i>. 9. Pengurus barang menekan “Simpan”. 10. Sistem melihatkan pesan sukses.
Alternative Flow	8.1 Jika <i>field form</i> tambah aset ada yang tidak terisi maka menampilkan pesan <i>error</i> sesuai dengan <i>field form</i> yang belum terisi.
Post Condition	Pengurus barang berhasil menambahkan aset.

4.5.2.8 Use Case Scenario Sunting Aset

Pada Tabel 4.11 menggambarkan perilaku aktor mengenai tahapan pada *use case* sunting aset.

**Tabel 4.11 Use Case Scenario Sunting Aset**

Flow of Events for Sunting Aset	
Kode Kebutuhan	SIMA-PFS.08
Objective	Pengurus barang bisa memperbarui aset.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Aset Dinas” pada <i>sidebar</i> menu. 2. Pengurus menekan submenu “Daftar Aset” pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar aset. 4. Pengurus barang memilih aset yang akan diperbarui kemudian menekan tombol “Sunting” pada aset yang akan diperbarui. 5. Sistem menampilkan detail dari aset dalam bentuk <i>pop-up</i> dengan berisikan detail dari aset. 6. Pengurus barang mengisi <i>field form</i> sunting data aset sesuai dengan bagian <i>field</i> yang akan diperbarui. 7. Pengurus barang menekan “Simpan”. 8. Sistem mellihatkan pesan sukses.
Alternative Flow	6.1 Jika <i>field form</i> sunting aset dikosongkan maka akan menampilkan pesan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Post Condition	Pengurus barang bisa memperbarui data aset.

4.5.2.9 Use Case Scenario Hapus Aset

Pada Tabel 4.12 menggambarkan perilaku aktor mengenai tahapan pada *use case* hapus aset.

Tabel 4.12 Use Case Scenario Hapus Aset

Flow of Events for Hapus Aset	
Kode Kebutuhan	SIMA-PFS.09
Objective	Pengurus barang bisa menghapus aset.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.

Tabel 4.12 *Use Case Scenario* Hapus Aset (Lanjutan)

Flow of Events for Hapus Aset	
Kode Kebutuhan	SIMA-PFS.09
Main flow	1. Pengurus barang menekan “Aset Dinas” pada <i>sidebar</i> menu. 2. Pengurus menekan submenu “Daftar Aset” pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar aset. 4. Pengurus barang memilih aset yang akan dihapus. 5. Pengurus barang menekan tombol “Hapus” pada aset yang akan dihapus. 6. Sistem memberikan pesan “Anda yakin menghapus data aset ini?”. 7. Pengurus barang menekan “Ok”. 8. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Pengurus barang bisa menghapus aset.

4.5.2.10 *Use Case Scenario* Permintaan Laporan Aset

Pada Tabel 4.13 menggambarkan perilaku aktor mengenai tahapan pada *use case* permintaan pembuatan laporan aset.

Tabel 4.13 *Use Case Scenario* Permintaan Laporan Aset

Flow of Events for Permintaan Laporan Aset	
Kode Kebutuhan	SIMA-PFS.10
Objective	Kepala Dinas bisa melakukan permintaan pembuatan laporan aset.
Aktor	Kepala Dinas.
Pre Condition	Kepala Dinas berada pada laman kepala dinas.
Main flow	1. Kepala Dinas memilih menu “Permintaan Laporan” di <i>sidebar</i> menu. 2. Sistem melihatkan laman permintaan pembuatan laporan. 3. Kepala Dinas memilih “Aset” pada bagian <i>dropdown</i> dan juga menambahkan catatan apabila ingin menambahkan catatan dalam pembuatan laporan.

Tabel 4.13 Use Case Scenario Permintaan Laporan Aset (Lanjutan)

Flow of Events for Permintaan Laporan Aset	
Kode Kebutuhan	SIMA-PFS.10
	4. Kepala Dinas menekan tombol “Request Laporan”. 5. Sistem melihatkan pesan sukses.
Alternative Flow	4.1 Bila pilihan laporan belum dipilih pada <i>menu dropdown</i> dan menekan tombol <i>request</i> laporan maka akan muncul pesan “The nama permintaan field is required”.
Post Condition	Kepala Dinas berhasil melakukan permintaan pembuatan laporan aset.

4.5.2.11 Use Case Scenario Daftar Laporan Aset

Pada Tabel 4.14 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar laporan aset.

Tabel 4.14 Use Case Scenario Daftar Laporan Aset

Flow of Events for Daftar Laporan Aset	
Kode Kebutuhan	SIMA-PFS.11
Objective	Pengurus barang bisa lihat daftar laporan aset.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Aset Dinas” pada <i>sidebar</i> menu. 2. Pengurus menekan submenu “Pelaporan Aset” pada <i>sidebar</i> menu. 3. Sistem menampilkan laman daftar laporan aset.
Alternative Flow	-
Post Condition	Pengurus barang berhasil lihat daftar laporan aset.

4.5.2.12 Use Case Scenario Sunting Laporan Aset

Pada Tabel 4.15 menggambarkan perilaku aktor mengenai tahapan pada *use case* sunting laporan aset.

Tabel 4.15 Use Case Scenario Sunting Laporan Aset

Flow of Events for Sunting Laporan Aset	
Kode Kebutuhan	SIMA-PFS.12
Objective	Pengurus barang bisa perbarui laporan aset.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Aset Dinas” pada <i>sidebar</i> menu. 4. Pengurus menekan submenu “Pelaporan Aset” pada <i>sidebar</i> menu. 5. Sistem menampilkan laman pelaporan aset. 6. Pengurus barang melihat detail dari aset yang akan dilaporkan dengan menekan tombol “Lihat Detail”. 7. Sistem menampilkan laman <i>detail</i> aset. 8. Pengurus barang memilih data aset yang akan diperbarui dengan menekan “Sunting”. 9. Sistem menampilkan <i>pop-up modal edit</i> laporan aset. 10. Pengurus mengisi <i>fields</i> yang akan di perbarui dan menekan “Simpan”. 11. Sistem melihatkan pesan sukses.
Alternative Flow	10.1 Jika staf pengurus barang tidak mengisi <i>field</i> yang tersedia maka akan menampilkan pesan <i>error</i> sesuai <i>field</i> kosong.
Post Condition	Staf pengurus barang berhasil memperbarui laporan aset.

4.5.2.13 Use Case Scenario Laporan Aset

Pada Tabel 4.16 menggambarkan perilaku aktor mengenai tahapan pada *use case* laporan aset.

Tabel 4.16 Use Case Scenario Laporan Aset

Flow of Events for Laporan Aset	
Kode Kebutuhan	SIMA-PFS.13
Objective	Pengurus barang bisa membuat laporan aset pada sistem.

Tabel 4.16 Use Case Scenario Laporan Aset (Lanjutan)

Flow of Events for Laporan Aset	
Kode Kebutuhan	SIMA-PFS.13
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Aktor	Pengurus barang.
Main flow	1. Pengurus barang menekan “Aset Dinas” pada <i>sidebar</i> menu. 2. Pengurus menekan submenu “Pelaporan Aset” pada <i>sidebar</i> menu. 3. Sistem menampilkan laman pelaporan aset. 4. Pengurus barang melihat detail dari aset yang akan dilaporkan dengan menekan tombol “Lihat Detail”. 5. Sistem menampilkan laman <i>detail</i> aset. 6. Pengurus barang melakukan pengecekan dan menekan tombol “Laporkan” untuk melaporkan aset. 7. Sistem menampilkan pesan “Anda Yakin Meminta Validasi Laporan?”. 8. Pengurus menekan tombol “Laporkan”. 9. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Pengurus barang berhasil membuat laporan aset.

4.5.2.14 Use Case Scenario Daftar Validasi Aset

Pada Tabel 4.17 menggambarkan perilaku aktor mengenai tahapan pada *use case export* laporan aset.

Tabel 4.17 Use Case Scenario Daftar Validasi Aset

Flow of Events for Daftar Validasi Aset	
Kode Kebutuhan	SIMA-PFS.14
Objective	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan dapat lihat daftar validasi aset.
Aktor	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan.
Pre Condition	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan pada laman masing-masing.

Tabel 4.17 Use Case Scenario Daftar Validasi Aset (Lanjutan)

Flow of Events for Daftar Validasi Aset	
Kode Kebutuhan	SIMA-PFS.14
Main flow	1. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan “Daftar Validasi” pada <i>sidebar menu</i>. 2. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan submenu “Aset” pada <i>sidebar menu</i>. 3. Sistem menampilkan daftar aset yang diminta untuk divalidasi.
Alternative Flow	-
Pre Condition	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan berhasil lihat daftar validasi aset.

4.5.2.15 Use Case Scenario Validasi Laporan Aset

Pada Tabel 4.18 menggambarkan perilaku aktor mengenai tahapan pada *use case* validasi laporan.

Tabel 4.18 Use Case Scenario Validasi Laporan Aset

Flow of Events for Validasi Laporan Aset	
Kode Kebutuhan	SIMA-PFS.15
Objective	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan dapat melakukan validasi laporan.
Aktor	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan.
Pre Condition	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan berada pada laman masing-masing.
Main flow	1. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan “Daftar Validasi” pada <i>sidebar menu</i>. 2. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan submenu “Aset” pada <i>sidebar menu</i>. 3. Sistem menampilkan daftar aset yang diminta untuk divalidasi. 4. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan melihat <i>detail</i> dengan menekan “Lihat Detail”.

Tabel 4.18 Use Case Scenario Validasi Laporan Aset (Lanjutan)

Flow of Events for Validasi Laporan Aset	
Kode Kebutuhan	SIMA-PFS.15
Main flow	5. Sistem menampilkan detail laporan aset pada tabel. 6. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menandai <i>checkbox</i> dan menekan tombol “Validasi”. 7. Sistem menampilkan pesan “Anda Yakin Meminta Validasi Laporan ?”. 8. Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan tombol “Validasi!”. 9. Sistem akan menampilkan pesan sukses.
Alternative Flow	6.1 Jika Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan tombol “Tolak Validasi” menandai pada <i>checkbox</i> dan menekan tombol simpan pada <i>pop-up message</i> akan muncul pesan sukses “Laporan aset berhasil ditolak”. 6.2 Jika Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan tombol “Tolak Validasi” tanpa menandai pada <i>checkbox</i> dan menekan simpan pada <i>pop-up message</i> akan muncul pesan <i>error</i> “Anda belum menandai aset pada <i>checkbox</i>!”. 6.3 Jika Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan menekan tombol “Validasi” tanpa menandai pada <i>checkbox</i> dan menekan simpan pada <i>pop-up message</i> akan muncul pesan <i>error</i> “Anda belum menandai aset pada <i>checkbox</i>!”.
Post Condition	Sekretaris Dinas dan Kepala Subbagian Keuangan dan Perencanaan berhasil melakukan validasi laporan.

4.5.2.16 Use Case Scenario Export Laporan Aset

Pada Tabel 4.19 menggambarkan perilaku aktor mengenai tahapan pada *use case export* laporan aset.

Tabel 4.19 Use Case Scenario Export Laporan Aset

Flow of Events for Export Laporan Aset	
Kode Kebutuhan	SIMA-PFS.16
Objective	Pengurus barang bisa meng-export laporan aset yang sudah divalidasi.
Aktor	Pengurus barang.
Pre Condition	1. Sebelumnya laporan aset sudah divalidasi oleh Kepala Subbagian Keuangan dan Perencanaan dan Sekretaris Dinas. 2. Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Aset Dinas” pada sidebar menu. 2. Pengurus barang menekan submenu “Export Pelaporan Aset” pada sidebar menu. 3. Sistem menampilkan laman export laporan aset. 4. Pengurus barang memilih kategori aset dan kode lokasi aset yang akan di export pada dropdown menu dan menekan Export. 5. Sistem melihatkan hasil export laporan aset dalam bentuk pdf sesuai dengan masukan.
Alternative Flow	4.1 Jika pengurus barang tidak mengisi field yang tersedia akan melihatkan pesan error.
Post Condition	Pengurus barang berhasil export laporan aset.

4.5.2.17 Use Case Scenario Daftar Persediaan

Pada Tabel 4.20 menggambarkan perilaku aktor mengenai tahapan pada use case daftar persediaan.

Tabel 4.20 Use Case Scenario Daftar Persediaan

Flow of Events for Daftar Persediaan	
Kode Kebutuhan	SIMA-PFS.17
Objective	Penyimpanan barang bisa lihat daftar persediaan.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman pengurus barang.
Main flow	1. Penyimpanan barang memilih “Persediaan” pada sidebar menu.

Tabel 4.20 Use Case Scenario Daftar Persediaan (Lanjutan)

Flow of Events for Daftar Persediaan	
Kode Kebutuhan	SIMA-PFS.17
	2. Penyimpanan barang menekan submenu “Daftar Persediaan” pada <i>sidebar</i> menu. 3. Sistem melihatkan daftar persediaan.
Alternative Flow	-
Post Condition	Penyimpanan barang bisa lihat daftar persediaan.

4.5.2.18 Use Case Scenario Tambah Persediaan

Pada Tabel 4.21 menggambarkan perilaku aktor mengenai tahapan pada *use case* tambah persediaan.

Tabel 4.21 Use Case Scenario Tambah Persediaan

Flow of Events for Tambah Persediaan	
Kode Kebutuhan	SIMA-PFS.18
Objective	Penyimpanan barang bisa menambahkan barang persediaan.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman penyimpanan barang.
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i> . 2. Penyimpanan barang menekan submenu “Daftar Persediaan” pada <i>sidebar menu</i> . 3. Sistem menampilkan daftar persediaan. 4. Penyimpanan barang menekan tombol <i>plus</i> di sebelah kanan tulisan tambah persediaan. 5. Sistem menampilkan <i>form</i> tambah persediaan. 6. Penyimpanan barang mengisi <i>form</i> tambah persediaan seperti kategori, asal, nomor dokumen dan yang lainnya sesuai dengan <i>field</i> yang tersedia dan menekan simpan. 7. Sistem melihatkan pesan sukses.
Alternative Flow	6.1 Jika <i>field form</i> tambah persediaan ada yang tidak terisi akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> yang belum terisi.
Post Condition	Penyimpanan barang berhasil menambah data persediaan.

4.5.2.19 Use Case Scenario Sunting Persediaan

Pada Tabel 4.22 menggambarkan perilaku aktor mengenai tahapan pada *use case* sunting persediaan.

Tabel 4.22 Use Case Scenario Sunting Persediaan

Flow of Events for Sunting Persediaan	
Kode Kebutuhan	SIMA-PFS.19
Objective	Penyimpanan barang bisa memperbarui persediaan.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman penyimpanan barang.
Main flow	1. Penyimpanan barang menekan “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Daftar Persediaan” pada <i>sidebar menu</i>. 3. Sistem menampilkan daftar persediaan. 4. Penyimpanan barang memilih data persediaan yang akan diperbarui. 5. Penyimpanan barang menekan “Sunting” pada persediaan yang akan diperbarui. 6. Sistem melihatkan <i>detail</i> dari persediaan dalam bentuk <i>pop-up</i> berisikan <i>detail</i> dari persediaan. 7. Penyimpanan barang mengisi <i>field form</i> data persediaan yang akan diperbarui sesuai dengan <i>field</i> yang tersedia kemudian menekan “Simpan”. 8. Sistem melihatkan pesan berhasil.
Alternative Flow	7.1 Jika <i>field form</i> sunting persediaan ada yang tidak terisi akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.
Post Condition	Penyimpanan barang berhasil melakukan memperbarui data persediaan.

4.5.2.20 Use Case Scenario Order Persediaan

Pada Tabel 4.23 menggambarkan perilaku aktor mengenai tahapan pada *use case* order persediaan.

Tabel 4.23 Use Case Scenario Order Persediaan

Flow of Events for Order Persediaan	
Kode Kebutuhan	SIMA-PFS.20
Objective	Penyimpanan barang bisa memasukkan <i>order</i> persediaan pada sistem.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman penyimpanan barang.
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Order Persediaan”. 3. Sistem menampilkan laman <i>order</i> persediaan. 4. Penyimpanan barang memasukkan data sesuai dengan <i>field order</i> persediaan yang tersedia dan menekan tombol “Simpan” ketika semua <i>field form</i> sudah terisi. 5. Sistem melihatkan pesan sukses.
Alternative Flow	4.1 Jika <i>field order</i> persediaan ada yang tidak terisi akan menampilkan pesan <i>error</i> sesuai dengan aset <i>field</i> yang kosong. 4.2 Jika terdapat aset yang sudah habis akan menampilkan pesan <i>error</i> “Ada persediaan yang kosong”.
Post Condition	Penyimpanan barang berhasil memasukkan <i>order</i> persediaan.

4.5.2.21 Use Case Scenario Daftar Pengeluaran

Pada Tabel 4.24 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar pengeluaran.

Tabel 4.24 Use Case Scenario Daftar Pengeluaran

Flow of Events for Daftar Pengeluaran Surat	
Kode Kebutuhan	SIMA-PFS.21
Objective	Penyimpanan barang bisa lihat daftar pengeluaran surat.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang pada laman penyimpanan barang.

Tabel 4.24 *Use Case Scenario* Daftar Pengeluaran (Lanjutan)

Flow of Events for Penerbitan Surat	
Kode Kebutuhan	SIMA-PFS.21
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Penerbitan Surat” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman daftar pengeluaran sesuai dengan nama pemohon persediaan.
Alternative Flow	-
Post Condition	Penyimpanan barang berhasil lihat daftar pengeluaran.

4.5.2.22 *Use Case Scenario* Penerbitan Surat

Pada Tabel 4.25 menggambarkan perilaku aktor mengenai tahapan pada *use case* penerbitan surat.

Tabel 4.25 *Use Case Scenario* Penerbitan Surat

Flow of Events for Penerbitan Surat	
Kode Kebutuhan	SIMA-PFS.22
Objective	Penyimpanan barang bisa menerbitkan SOB dan SPPB.
Aktor	Penyimpanan barang.
Pre Condition	1. Penerbitan surat dapat dilakukan ketika sudah ada <i>order-an</i> persediaan dari bidang lain. 2. Penyimpanan barang berada pada laman penyimpanan barang.
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Penerbitan Surat” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman penerbitan surat ditampilkan sesuai dengan nama pemohon persediaan. 4. Penyimpanan barang memilih nama pemohon sesuai yang akan diterbitkan suratnya. 5. Sistem menampilkan laman rincian dari <i>order</i> pemohon. 6. Penyimpanan barang menekan tombol penerbitan surat.

**Tabel 4.25 Use Case Scenario Penerbitan Surat (Lanjutan)**

Flow of Events for Penerbitan Surat	
Kode Kebutuhan	SIMA-PFS.22
	7. Sistem menampilkan laman pdf yang berisikan surat SOB dan SPPB sesuai dengan nama pemohon.
Alternative Flow	-
Post Condition	Penyimpanan barang berhasil menerbitkan surat.

4.5.2.23 Use Case Scenario Hapus Order Persediaan

Pada Tabel 4.26 menggambarkan perilaku aktor mengenai tahapan pada *use case* hapus *order* persediaan.

Tabel 4.26 Use Case Scenario Hapus Order Persediaan

Flow of Events for Hapus Order Persediaan	
Kode Kebutuhan	SIMA-PFS.23
Objective	Penyimpanan barang bisa menghapus <i>order</i> persediaan.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman penyimpanan barang.
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Penerbitan Surat” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman penerbitan surat ditampilkan sesuai dengan nama pemohon persediaan. 4. Penyimpanan barang memilih nama pemohon sesuai yang akan diterbitkan suratnya. 5. Sistem menampilkan laman rincian dari <i>order</i> pemohon. 6. Penyimpanan barang menekan hapus. 7. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Penyimpanan barang berhasil menghapus <i>order</i> persediaan.

4.5.2.24 Use Case Scenario Permintaan Laporan Persediaan

Pada Tabel 4.27 menggambarkan perilaku aktor mengenai tahapan pada *use case* permintaan laporan persediaan.

Tabel 4.27 Use Case Scenario Permintaan Laporan Persediaan

Flow of Events for Permintaan Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.24
Objective	Kepala Dinas bisa melakukan permintaan pembuatan laporan persediaan.
Aktor	Kepala Dinas.
Pre Condition	Kepala Dinas berada pada laman kepala dinas.
Main flow	1. Kepala Dinas menekan “Permintaan Laporan” pada <i>sidebar</i> menu. 2. Sistem menampilkan laman permintaan pembuatan laporan. 3. Kepala Dinas memilih “Persediaan” pada bagian <i>dropdown</i> dan juga menambahkan catatan apabila ingin menambahkan catatan dalam pembuatan laporan. 4. Kepala Dinas menekan tombol “Request Laporan”. 5. Sistem melihatkan pesan berhasil.
Alternative Flow	4.1 Bila pilihan laporan belum dipilih pada menu <i>dropdown</i> maka akan muncul pesan “The nama permintaan <i>field is required</i> ”.
Post Condition	Kepala Dinas berhasil melakukan permintaan pembuatan laporan persediaan.

4.5.2.25 Use Case Scenario Daftar Laporan Persediaan

Pada Tabel 4.28 menggambarkan perilaku aktor mengenai tahapan pada *use case scenario* daftar laporan persediaan.

Tabel 4.28 Use Case Scenario Daftar Laporan Persediaan

Flow of Events for Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.25
Objective	Penyimpanan barang dapat lihat daftar laporan persediaan.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman Penyimpanan barang.

Tabel 4.28 Use Case Scenario Daftar Laporan Persediaan (Lanjutan)

Flow of Events for Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.25
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Pelaporan Persediaan” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman daftar laporan persediaan.
Alternative Flow	-
Post Condition	Penyimpanan barang bisa lihat daftar laporan persediaan.

4.5.2.26 Use Case Scenario Laporan Persediaan

Pada Tabel 4.29 menggambarkan perilaku aktor mengenai tahapan pada *use case scenario* laporan persediaan.

Tabel 4.29 Use Case Scenario Laporan Persediaan

Flow of Events for Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.26
Objective	Penyimpanan barang membuat laporan persediaan.
Aktor	Penyimpanan barang.
Pre Condition	Penyimpanan barang berada pada laman Penyimpanan barang.
Main flow	1. Penyimpanan barang memilih “Persediaan” pada <i>sidebar menu</i>. 2. Penyimpanan barang menekan submenu “Pelaporan Persediaan” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman daftar laporan persediaan. 4. Penyimpanan barang menekan tombol “Lihat Detail”. 5. Sistem akan menampilkan rincian dari laporan persediaan. 6. Penyimpanan barang menekan tombol “Laporkan” untuk melaporkan persediaan. 7. Sistem menampilkan pesan “Anda Yakin Meminta Validasi Laporan ?”.

Tabel 4.29 Use Case Scenario Laporan Persediaan (Lanjutan)

Flow of Events for Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.26
	8. Penyimpanan barang menekan tombol "Laporkan". 9. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Penyimpanan barang bisa membuat laporan persediaan.

4.5.2.27 Use Case Scenario Daftar Validasi Persediaan

Pada Tabel 4.30 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar validasi persediaan.

Tabel 4.30 Use Case Scenario Daftar Validasi Persediaan

Flow of Events for Daftar Validasi Persediaan	
Kode Kebutuhan	SIMA-PFS.27
Objective	Penyimpanan barang dan kepala subbagian umum dan kepegawaian dapat lihat daftar validasi laporan persediaan.
Aktor	Penyimpanan barang dan Kepala subbagian umum dan kepegawaian.
Pre Condition	Penyimpanan barang dan kepala subbagian umum dan kepegawaian berada pada laman masing-masing.
Main flow	1. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian menekan "Validasi Laporan" pada <i>sidebar</i> menu. 2. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian memilih submenu "Persediaan" pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar validasi laporan persediaan.
Alternative Flow	-
Post Condition	Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian berhasil lihat daftar validasi persediaan.

4.5.2.28 Use Case Scenario Validasi Laporan Persediaan

Pada Tabel 4.31 menggambarkan perilaku aktor mengenai tahapan pada *use case* validasi laporan persediaan.

Tabel 4.31 Use Case Scenario Validasi Laporan Persediaan

Flow of Events for Validasi Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.28
Objective	Penyimpanan barang dan kepala subbagian umum dan kepegawaian melakukan validasi laporan persediaan.
Aktor	Penyimpanan barang dan Kepala subbagian umum dan kepegawaian.
Pre Condition	Penyimpanan barang dan kepala subbagian umum dan kepegawaian berada pada laman masing-masing.
Main flow	1. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian menekan "Validasi Laporan" pada <i>sidebar</i> menu. 2. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian memilih submenu "Persediaan" pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar validasi laporan persediaan. 4. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian menekan tombol "Lihat Detail" untuk melihat detail validasi. 5. Sistem menampilkan detail laporan persediaan. 6. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian menandai <i>checkbox</i> dan menekan tombol "Validasi". 7. Sistem menampilkan pesan "Anda Yakin Meminta Validasi Laporan?". 8. Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian menekan tombol "Validasi". 9. Sistem mengirimkan hasil validasi dan mengosongkan laporan persediaan yang ada pada tabel.
Alternative Flow	6.1 Jika <i>checkbox</i> belum ditandai akan melihatkan pesan <i>error</i> "Anda belum menandai <i>checkbox</i> ".
Post Condition	Penyimpanan barang dan Kepala Subbagian Umum dan Kepegawaian berhasil melakukan validasi laporan persediaan.



4.5.2.29 Use Case Scenario Export Laporan Persediaan

Pada Tabel 4.32 menggambarkan perilaku aktor mengenai tahapan pada *use case export* laporan persediaan.

Tabel 4.32 Use Case Scenario Export Laporan Persediaan

Flow of Events for Export Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.29
Objective	Penyimpanan Barang bisa meng-export laporan persediaan yang sudah divalidasi.
Aktor	Penyimpanan Barang.
Pre Condition	1. Sebelumnya laporan persediaan sudah divalidasi oleh Kepala Subbagian Umum dan Kepegawaian. 2. Penyimpanan Barang berada pada laman penyimpanan barang.
Main flow	1. Penyimpanan Barang memilih "Persediaan" pada sidebar menu. 2. Penyimpanan Barang menekan submenu "Mutasi Persediaan". 3. Sistem menampilkan laman mutasi persediaan. 4. Penyimpanan Barang memilih semester dan tahun mutasi selanjutnya menekan tombol <i>export</i>. 5. Sistem akan menampilkan hasil <i>export</i> dalam bentuk pdf sesuai dengan masukan.
Alternative Flow	-
Post Condition	Penyimpanan Barang berhasil meng-export laporan persediaan.

4.5.2.30 Use Case Scenario Tambah Pemegang Kendaraan

Pada Tabel 4.33 menggambarkan perilaku aktor mengenai tahapan pada *use case* tambah pemegang kendaraan.

Tabel 4.33 Use Case Scenario Tambah Pemegang Kendaraan

Flow of Events for Tambah Pemegang Kendaraan	
Kode Kebutuhan	SIMA-PFS.30
Objective	Pengurus barang bisa menambahkan data pemegang kendaraan ke dalam sistem.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang pada laman pengurus barang.

Tabel 4.33 Use Case Scenario Tambah Pemegang Kendaraan (Lanjutan)

Flow of Events for Tambah Pemegang Kendaraan	
Kode Kebutuhan	SIMA-PFS.30
Main flow	1. Pengurus barang menekan “Pemeliharaan Kendaraan” pada <i>sidebar menu</i>. 2. Pengurus barang menekan submenu “Daftar Pemegang Kendaraan” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman daftar pemegang kendaraan. 4. Pengurus barang memilih nama pemegang dan juga nomor polisi kendaraan di bagian <i>dropdown menu</i> kemudian menekan “Simpan”. 5. Sistem menampilkan pesan sukses.
Alternative Flow	4.1 Jika bagian <i>dropdown</i> ada yang tidak dipilih serta menekan “Simpan” maka melihatkan pesan <i>error</i> sesuai dengan <i>field dropdown</i> kosong.
Post Condition	Pengurus barang berhasil menambahkan pemegang kendaraan baru ke sistem.

4.5.2.31 Use Case Scenario Daftar Pemegang Kendaraan

Pada Tabel 4.34 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar pemegang kendaraan.

Tabel 4.34 Use Case Scenario Daftar Pemegang Kendaraan

Flow of Events for Daftar Pemegang Kendaraan	
Kode Kebutuhan	SIMA-PFS.31
Objective	Pengurus barang bisa lihat daftar pemegang kendaraan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pemeliharaan Kendaraan” pada <i>sidebar menu</i>. 2. Pengurus barang menekan submenu “Daftar Pemegang Kendaraan” pada <i>sidebar menu</i>. 3. Sistem menampilkan daftar pemegang kendaraan.
Alternative Flow	-
Post Condition	Pengurus barang berhasil lihat daftar pemegang kendaraan.

4.5.2.32 Use Case Scenario Hapus Kendaraan

Pada Tabel 4.35 menggambarkan perilaku aktor mengenai tahapan pada *use case* hapus kendaraan.

Tabel 4.35 Use Case Scenario Hapus Kendaraan

Flow of Events for Hapus Kendaraan	
Kode Kebutuhan	SIMA-PFS.32
Objective	Pengurus barang bisa menghapus pemegang kendaraan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pemeliharaan Kendaraan” pada <i>sidebar menu</i>. 2. Pengurus barang menekan submenu “Daftar Pemegang Kendaraan” pada <i>sidebar menu</i>. 3. Sistem menampilkan daftar pemegang kendaraan. 4. Pengurus barang menekan “Lihat Detail” sesuai dengan nama pemegang kendaraan. 5. Sistem melihatkan <i>detail</i> dari pemegang kendaraan. 6. Pengurus barang menekan tombol “Hapus” pada detail pemegang kendaraan yang akan dihapus. 7. Sistem memberikan pesan “Yakin menghapus kendaraan ini?”. 8. Pengurus barang menekan “Ok”. 9. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Pengurus barang berhasil menghapus pemegang kendaraan.

4.5.2.33 Use Case Scenario Permintaan Laporan Pemeliharaan Kendaraan

Pada Tabel 4.36 menggambarkan perilaku aktor mengenai tahapan pada *use case* permintaan pembuatan laporan pemeliharaan kendaraan.

Tabel 4.36 Use Case Scenario Permintaan Laporan Pemeliharaan Kendaraan

Flow of Events for Permintaan Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.33
Objective	Sekretaris Dinas bisa melakukan permintaan pembuatan laporan pemeliharaan kendaraan.
Aktor	Sekretaris Dinas.
Pre Condition	Sekretaris dinas berada pada laman sekretaris dinas.
Main flow	1. Sekretaris Dinas menekan "Permintaan Laporan" pada <i>sidebar</i>. 2. Sistem menampilkan laman permintaan pembuatan laporan. 3. Sekretaris Dinas memilih kendaraan operasional pada bagian <i>dropdown</i> dan juga menambahkan catatan apabila ingin menambahkan catatan dalam pembuatan laporan. 4. Sekretaris Dinas menekan tombol "Request Laporan". Sistem melihatkan pesan sukses.
Alternative Flow	4.1 Jika pilihan laporan belum dipilih pada <i>menu dropdown</i> maka akan muncul pesan "The nama permintaan <i>field is required</i> ".
Post Condition	Sekretaris Dinas berhasil melakukan permintaan pembuatan laporan pemeliharaan kendaraan.

4.5.2.34 Use Case Scenario Daftar Pemeliharaan Kendaraan

Pada Tabel 4.37 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar pemeliharaan kendaraan.

Tabel 4.37 Use Case Scenario Daftar Pemeliharaan Kendaraan

Flow of Events for Daftar Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.34
Objective	Pemegang kendaraan bisa memperbarui pemeliharaan kendaraan.
Aktor	Pemegang kendaraan.
Pre Condition	Pemegang kendaraan berada pada laman pemegang kendaraan.
Main flow	1. Pemegang kendaraan menekan "Pemeliharaan Kendaraan" pada <i>sidebar menu</i>.

**Tabel 4.37 Use Case Scenario Daftar Pemeliharaan Kendaraan (Lanjutan)**

Flow of Events for Daftar Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.34
	2. Pemegang kendaraan menekan submenu "Update Pemeliharaan". 3. Sistem akan melihatakan laman daftar pemeliharaan kendaraan.
Alternative Flow	-
Post Condition	Pemegang kendaraan berhasil lihat daftar pemeliharaan kendaraan.

4.5.2.35 Use Case Scenario Perbarui Pemeliharaan Kendaraan

Pada Tabel 4.38 menggambarkan perilaku aktor mengenai tahapan pada *use case* perbarui pemeliharaan kendaraan.

Tabel 4.38 Use Case Scenario Perbarui Pemeliharaan Kendaraan

Flow of Events for Perbarui Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.35
Objective	Pemegang kendaraan bisa memperbaiki pemeliharaan kendaraan.
Aktor	Pemegang kendaraan.
Pre Condition	Pemegang kendaraan berada pada laman pemegang kendaraan.
Main flow	1. Pemegang kendaraan menekan "Pemeliharaan Kendaraan" pada <i>sidebar menu</i> . 2. Pemegang kendaraan menekan submenu "Update Pemeliharaan". 3. Sistem menampilkan laman <i>update</i> pemeliharaan. 4. Pemegang kendaraan memasukkan hasil pemeliharaan laik kendaraan sesuai dengan <i>field</i> yang tersedia dan menekan "Simpan". 5. Sistem melihatkan pesan sukses.
Alternative Flow	4.1 Jika kondisi kendaraan tidak dimasukkan maka akan menampilkan pesan <i>error</i> "The kondisi kendaraan <i>field is required</i> "
Post Condition	Pemegang kendaraan berhasil memperbaiki data pemeliharaan kendaraan.

4.5.2.36 Use Case Scenario Hapus Hasil Pemeliharaan Kendaraan

Pada Tabel 4.39 menggambarkan perilaku aktor mengenai tahapan pada *use case* hapus hasil pemeliharaan kendaraan.

Tabel 4.39 Use Case Scenario Hapus Hasil Pemeliharaan Kendaraan

Flow of Events for Hapus Hasil Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.36
Objective	Pemegang kendaraan menghapus hasil pemeliharaan kendaraan yang sudah ada di dalam sistem.
Aktor	Pemegang kendaraan.
Pre Condition	Pemegang kendaraan berada pada laman pemegang kendaraan.
Main flow	1. Pemegang kendaraan menekan "Pemeliharaan" pada <i>sidebar</i> menu. 2. Pemegang kendaraan menekan submenu "Pelaporan Pemeliharaan" pada <i>sidebar</i> menu. 3. Sistem menampilkan laman daftar pelaporan pemeliharaan. 4. Pemegang kendaraan memilih hasil pemeliharaan kendaraan yang akan dihapus. 5. Pemegang kendaraan menekan "Hapus" pada pelaporan yang akan dihapus. 6. Sistem memberikan pesan "Yakin menghapus hasil pengecekan laik kendaraan?" ketika menekan tombol hapus. 7. Pengurus barang menekan "Ok". 8. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Pemegang kendaraan berhasil menghapus data hasil pemeliharaan kendaraan.

4.5.2.37 Use Case Scenario Laporan Pemeliharaan Kendaraan

Pada Tabel 4.40 menggambarkan perilaku aktor mengenai tahapan pada *use case* laporan pemeliharaan kendaraan.

**Tabel 4.40 Use Case Scenario Laporan Pemeliharaan Kendaraan**

Flow of Events for Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.37
Objective	Pemegang kendaraan bisa membuat laporan pemeliharaan kendaraan.
Aktor	Pemegang kendaraan.
Pre Condition	Pemegang kendaraan berada pada laman pemegang kendaran.
Main flow	1. Pemegang kendaraan memilih “Pemeliharaan” pada <i>sidebar</i> menu. 2. Pemegang kendaraan menekan submenu “Pelaporan Pemeliharaan” pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar laporan pemeliharaan kendaraan. 4. Pemegang kendaraan menekan tombol “Laporkan”. 5. Sistem menampilkan pesan “Anda yakin Meminta Validasi Laporan ?”. 6. Pemegang kendaraan menekan “Laporkan!”. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Pemegang kendaraan berhasil membuat laporan pemeliharaan kendaraan.

4.5.2.38 Use Case Scenario Daftar Validasi Pemeliharaan Kendaraan

Pada Tabel 4.41 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar validasi pemeliharaan kendaraan.

Tabel 4.41 Use Case Scenario Daftar Validasi Pemeliharaan Kendaraan

Flow of Events for Daftar Validasi Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.38
Objective	Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian dapat lihat daftar validasi pemeliharaan kendaraan.
Aktor	Pengurus Barang dan Kepala Subbagian Umum dan Kepegawaian.
Pre Condition	Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian berada pada laman masing-masing.

Tabel 4.41 Use Case Scenario Daftar Validasi Pemeliharaan Kendaraan (Lanjutan)

Flow of Events for Daftar Validasi Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.38
Main flow	1. Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan “Daftar Validasi” pada <i>sidebar menu</i>. 2. Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan submenu “Pemeliharaan Kendaraan” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman daftar validasi pemeliharaan kendaraan.
Alternative Flow	-
Post Condition	Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian berhasil lihat daftar validasi pemeliharaan kendaraan.

4.5.2.39 Use Case Scenario Validasi Laporan Pemeliharaan Kendaraan

Pada Tabel 4.42 menggambarkan perilaku aktor mengenai tahapan pada *use case* validasi laporan pemeliharaan kendaraan.

Tabel 4.42 Use Case Scenario Validasi Laporan Pemeliharaan Kendaraan

Flow of Events for Validasi Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.39
Objective	Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian melakukan validasi laporan pemeliharaan kendaraan.
Aktor	Pengurus Barang dan Kepala Subbagian Umum dan Kepegawaian.
Pre Condition	Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian berada pada laman masing-masing.
Main flow	1. Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan “Daftar Validasi” pada <i>sidebar menu</i>. 2. Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan submenu “Pemeliharaan Kendaraan” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman validasi pemeliharaan kendaraan.

Tabel 4.42 Use Case Scenario Validasi Laporan Pemeliharaan Kendaraan (Lanjutan)

Flow of Events for Validasi Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.39
	4. Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian memilih berdasarkan pengguna dan melihat detail validasi dengan menekan tombol “Lihat Detail”. 5. Sistem menampilkan daftar pemeliharaan kendaraan yang akan divalidasi. 6. Pengurus barang dan Kepala Subbagian Umum dan—Kepegawaian menandai <i>checkbox</i> dan menekan tombol “Validasi”. 7. Sistem akan menampilkan pesan “Anda Yakin Meminta Validasi Laporan ?”. 8. Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian selanjutnya menekan tombol “Validasi”. 9. Sistem menampilkan pesan sukses.
Alternative Flow	7.1 Jika pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan “Tolak Validasi” dan menandai <i>checkbox</i> kemudian menekan tombol simpan pada <i>pop-up message</i> maka akan menampilkan pesan sukses “Laporan berhasil ditolak!”. 7.1 Jika pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan tombol “Tolak Validasi” tanpa menandai <i>checkbox</i> kemudian menekan tombol simpan pada <i>pop-up message</i> maka akan menampilkan pesan error “Anda belum menandai <i>checkbox</i>!”. 7.3 Jika pengurus barang dan Kepala Subbagian Umum dan Kepegawaian menekan tombol “Validasi” tanpa menandai <i>checkbox</i> maka akan menampilkan pesan error “Anda belum menandai <i>checkbox</i>!”.
Post Condition	Pengurus barang dan Kepala Subbagian Umum dan Kepegawaian berhasil melakukan validasi laporan pemeliharaan kendaraan.



4.5.2.40 Use Case Scenario Daftar Hasil Pemeliharaan Kendaraan

Pada Tabel 4.43 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar hasil pemeliharaan kendaraan.

Tabel 4.43 Use Case Scenario Daftar Hasil Pemeliharaan Kendaraan

Flow of Events for Daftar Hasil Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.40
Objective	Pengurus barang bisa lihat hasil pemeliharaan kendaraan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang memilih “Pemeliharaan Kendaraan” pada <i>sidebar</i> menu. 2. Pengurus barang menekan submenu “Daftar Hasil Pemeliharaan” pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar hasil pemeliharaan kendaraan sesuai dengan nama pemegang kendaraan.
Alternative Flow	-
Post Condition	Pengurus barang bisa lihat daftar hasil pemeliharaan kendaraan.

4.5.2.41 Use Case Scenario Export Laporan Pemeliharaan Kendaraan

Pada Tabel 4.44 menggambarkan perilaku aktor mengenai tahapan pada *use case* export laporan pemeliharaan kendaraan.

Tabel 4.44 Use Case Scenario Export Laporan Pemeliharaan Kendaraan

Flow of Events for Export Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.41
Objective	Pemegang kendaraan meng-export laporan pemeliharaan kendaraan yang sudah selesai divalidasi.
Aktor	Pemegang kendaraan.
Pre Condition	1. Sebelumnya laporan pemeliharaan kendaraan sudah divalidasi oleh Kepala Subbagian Umum dan Kepegawaian dan pengurus barang. 2. Pemegang kendaraan berada pada laman pemegang kendaraan.

Tabel 4.44 Use Case Scenario Export Laporan Pemeliharaan Kendaraan (Lanjutan)

Flow of Events for Export Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.41
Main flow	1. Pemegang kendaraan memilih “Pemeliharaan” pada <i>sidebar</i> menu. 2. Pemegang kendaraan menekan submenu “Pelaporan Pemeliharaan” pada <i>sidebar</i> menu. 3. Sistem menampilkan daftar laporan pemeliharaan kendaraan. 4. Pemegang kendaraan menekan tombol “Export”. 5. Sistem akan melihatkan hasil <i>export</i> dalam bentuk pdf.
Alternative Flow	-
Post Condition	Pemegang kendaraan berhasil meng- <i>export</i> laporan pemeliharaan kendaraan.

4.5.2.42 Use Case Scenario Daftar Pengecekan Inventaris Ruangan

Pada Tabel 4.45 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar pengecekan inventaris ruangan.

Tabel 4.45 Use Case Scenario Daftar Pengecekan Inventaris Ruangan

Flow of Events for Daftar Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.42
Objective	Pengurus barang bisa lihat daftar pengecekan inventaris ruangan pada sistem.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pengecekan” pada <i>sidebar</i> menu. 2. Pengurus barang menekan submenu “Daftar Pengecekan”. 3. Sistem melihatkan laman daftar pengecekan inventaris ruangan.
Alternative Flow	-
Post Condition	Pengurus barang berhasil lihat daftar pengecekan inventaris ruangan.



4.5.2.43 Use Case Scenario Tambah Aset Pada Inventaris Ruangan

Pada Tabel 4.46 menggambarkan perilaku aktor mengenai tahapan pada *use case* tambah aset pada inventaris ruangan.

Tabel 4.46 Use Case Scenario Tambah Aset Pada Inventaris Ruangan

Flow of Events for Tambah Aset Pada Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.43
Objective	Pengurus barang bisa menambahkan aset pada inventaris ruangan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pengecekan” pada sidebar menu. 2. Pengurus barang menekan submenu “Daftar Pengecekan”. 3. Sistem menampilkan laman daftar pengecekan. 4. Pengurus barang memasukkan nama ruangan, nama barang dan jumlah barang kemudian menekan simpan. 5. Sistem melihatkan pesan sukses.
Alternative Flow	4.1 jika pengurus barnag tidak mengisikan <i>fields</i> yang tersedia akan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Post Condition	Pengurus barang bisa menambahkan aset pada daftar pengecekan inventaris ruangan.

4.5.2.44 Use Case Scenario Hapus Aset Pada Inventaris Ruangan

Pada Tabel 4.47 menggambarkan perilaku aktor mengenai tahapan pada *use case* tambah aset pada inventaris ruangan.

Tabel 4.47 Use Case Scenario Hapus Aset Pada Inventaris Ruangan

Flow of Events for Hapus Aset Pada Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.44
Objective	Pengurus barang bisa menghapus aset pada inventaris ruangan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.

Tabel 4.47 Use Case Scenario Hapus Aset Pada Inventaris Ruangan (Lanjutan)

<i>Flow of Events for Hapus Aset Pada Inventaris Ruangan</i>	
Kode Kebutuhan	SIMA-PFS.44
<i>Main flow</i>	1. Pengurus barang menekan “Pengecekan” pada <i>sidebar</i> menu. 2. Pengurus barang menekan submenu “Daftar Pengecekan”. 3. Sistem menampilkan laman daftar pengecekan 4. Pengurus barang menekan tombol “Lihat <i>Detail</i>” sesuai dengan nama ruangan yang akan dihapus asetnya. 5. Sistem menampilkan rincian dari inventaris ruangan yang dipilih. 6. Pengurus barang menekan hapus. 7. Sistem melihatkan pesan “Anda yakin menghapus aset ini dari daftar inventaris?” 8. Pengurus barang menekan tombol “Ok”. 9. Sistem melihatkan pesan sukses.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Pengurus barang bisa menghapus aset di daftar pengecekan inventaris ruangan.

4.5.2.45 Use Case Scenario Permintaan Laporan Inventaris Ruangan

Pada Tabel 4.48 menggambarkan perilaku aktor mengenai tahapan pada use case permintaan laporan inventaris ruangan.

Tabel 4.48 Use Case Scenario Permintaan Laporan Inventaris Ruangan

<i>Flow of Events for Permintaan Laporan Inventaris Ruangan</i>	
Kode Kebutuhan	SIMA-PFS.45
<i>Objective</i>	Kepala Subbagian Umum dan Kepegawaian meminta untuk pembuatan laporan inventaris ruangan.
Aktor	Kepala Subbagian Umum dan Kepegawaian.
<i>Pre Condition</i>	Kepala Subbagian Umum dan Kepegawaian berada pada laman masing-masing.
<i>Main flow</i>	1. Kepala Subbagian Umum dan Kepegawaian menekan “Permintaan Laporan” pada <i>sidebar</i> menu.

Tabel 4.48 Use Case Scenario Permintaan Laporan Inventaris Ruangan (Lanjutan)

Flow of Events for Permintaan Laporan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.45
	2. Sistem menampilkan laman permintaan pembuatan laporan. 3. Kepala Subbagian Umum dan Kepegawaian memilih pengecekan inventaris ruangan pada bagian <i>dropdown</i> dan juga menambahkan catatan apabila ingin menambahkan catatan dalam pembuatan laporan. 4. Kepala Subbagian Umum dan Kepegawaian menekan tombol "<i>Request Laporan</i>". 5. Sistem melihatkan pesan sukses.
Alternative Flow	4.1 Jika pilihan laporan belum dipilih pada <i>menu dropdown</i> maka akan muncul pesan " <i>The nama permintaan field is required</i> ".
Post Condition	Kepala Subbagian Umum dan Kepegawaian berhasil melakukan permintaan laporan inventaris ruangan.

4.5.2.46 Use Case Scenario Perbarui Data Inventaris Ruangan

Pada Tabel 4.49 menggambarkan perilaku aktor mengenai tahapan pada *use case* perbarui data inventaris ruangan.

Tabel 4.49 Use Case Scenario Perbarui Data Inventaris Ruangan

Flow of Events for Perbarui Data Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.46
Objective	Pengurus barang bisa memperbarui data inventaris ruangan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan "<i>Pengecekan</i>" pada <i>sidebar</i> menu. 2. Pengurus barang menekan submenu "<i>Daftar Pengecekan</i>" pada <i>sidebar</i> menu. 3. Sistem menampilkan laman daftar pengecekan.

Tabel 4.49 Use Case Scenario Perbarui Data Inventaris Ruangan (Lanjutan)

Flow of Events for Perbarui Data Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.46
	4. Pengurus barang memilih ruangan yang akan diperbarui dengan menekan tombol "Lihat Detail". 5. Sistem menampilkan daftar pengecekan berdasarkan ruangan. 6. Pengurus barang memasukkan kode lokasi, tanggal pengecekan, penanggung jawab ruangan dan memasukkan jumlah kondisi aset yang ada di ruangan kemudian menekan "Simpan". 7. Sistem melihatkan pesan sukses.
Alternative Flow	6.1 Jika <i>field form</i> pengecekan aset ada yang tidak terisi akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Post Condition	Pengurus barang berhasil memperbarui pengecekan inventaris ruangan.

4.5.2.47 Use Case Scenario Daftar Hasil Pengecekan Inventaris Ruangan

Pada Tabel 4.50 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar hasil pengecekan inventaris ruangan.

Tabel 4.50 Use Case Scenario Daftar Hasil Pengecekan Inventaris Ruangan

Flow of Events for Daftar Hasil Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.47
Objective	Pengurus barang dapat lihat daftar hasil pengecekan inventaris ruangan.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan "Pengecekan" pada <i>sidebar menu</i> . 2. Pengurus barang menekan submenu "Pelaporan Pengecekan" pada <i>sidebar menu</i> . 3. Sistem menampilkan laman daftar hasil pengecekan inventaris ruangan.
Alternative Flow	-
Post Condition	Pengurus barang berhasil lihat daftar hasil pengecekan inventaris ruangan.



4.5.2.48 Use Case Scenario Laporan Pengecekan Inventaris Ruangan

Pada Tabel 4.51 menggambarkan perilaku aktor mengenai tahapan pada *use case* laporan pengecekan inventaris ruangan.

Tabel 4.51 Use Case Scenario Laporan Pengecekan Inventaris Ruangan

Flow of Events for Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.48
Objective	Pengurus barang membuat laporan pengecekan inventaris ruangan pada sistem.
Aktor	Pengurus barang.
Pre Condition	1. Pengurus barang pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pengecekan” pada sidebar menu. 2. Pengurus barang menekan submenu “Pelaporan Pengecekan” pada sidebar menu. 3. Sistem menampilkan laman pelaporan pengecekan. 4. Pengurus barang memilih ruangan yang akan dilaporkan hasil pengecekannya dengan menekan tombol “Lihat Detail”. 5. Sistem menampilkan daftar hasil pengecekan berdasarkan ruangan. 6. Pengurus barang menekan tombol “Laporkan”. 7. Sistem akan menampilkan pesan “Anda Yakin Meminta Validasi laporan ?”. 8. Pengurus barang menekan tombol “Laporkan”. 9. Sistem melihatkan pesan sukses.
Alternative Flow	-
Post Condition	Pengurus barang berhasil membuat laporan pengecekan inventaris ruangan.

4.5.2.49 Use Case Scenario Sunting Hasil Pengecekan Inventaris Ruangan

Pada Tabel 4.52 menggambarkan perilaku aktor mengenai tahapan pada *use case* sunting hasil pengecekan inventari ruangan.

Tabel 4.52 Use Case Scenario Sunting Hasil Pengecekan Inventaris Ruangan

Flow of Events for Sunting Hasil Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.49
Objective	Pengurus barang bisa memperbarui laporan hasil pengecekan inventaris ruangan yang sudah ada didalam sistem.
Aktor	Pengurus barang.
Pre Condition	Pengurus barang berada pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pengecekan” pada <i>sidebar menu</i>. 2. Pengurus barang menekan submenu “Pelaporan Pengecekan” pada <i>sidebar menu</i>. 3. Sistem menampilkan laman pelaporan pengecekan. 4. Pengurus barang memilih ruangan yang akan diperbarui dengan menekan tombol “Lihat Detail”. 5. Sistem menampilkan daftar pengecekan berdasarkan ruangan. 6. Pengurus barang memilih hasil pengecekan aset yang akan diperbarui dan mengisi <i>field</i> yang tersedia kemudian menekan “Simpan”. 7. Sistem melihatkan pesan sukses.
Alternative Flow	6.1Jika pengurus barang tidak mengisi <i>fields</i> sesuai dengan yang tersedia maka akan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Post Condition	Pengurus barang berhasil memperbarui hasil pengecekan inventaris ruangan.

4.5.2.50 Use Case Scenario Daftar Validasi Pengecekan Inventaris Ruangan

Pada Tabel 4.53 menggambarkan perilaku aktor mengenai tahapan pada *use case* daftar validasi pengecekan inventaris ruangan.

Tabel 4.53 Use Case Scenario Daftar Validasi Pengecekan Inventaris Ruangan

Flow of Events for Daftar Validasi Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.50
Objective	Kepala Subbagian Umum dan Kepegawaian dapat lihat daftar validasi pengecekan inventaris ruangan.
Aktor	Kepala subbagian umum dan kepegawaian.

Tabel 4.53 Use Case Scenario Daftar Validasi Pengecekan Inventaris Ruangan (Lanjutan)

Flow of Events for Daftar Validasi Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.50
<i>Pre Condition</i>	Kepala Subbagian Umum dan Kepegawaian berada pada laman Kepala Subbagian Umum dan Kepegawaian.
<i>Main flow</i>	1. Kepala Subbagian Umum dan Kepegawaian menekan “Daftar Validasi” pada <i>sidebar menu</i>. 2. Kepala Subbagian Umum dan Kepegawaian menekan submenu “Hasil Pengecekan Aset”. 3. Sistem menampilkan daftar validasi pengecekan inventaris ruangan.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Kepala subbagian umum dan kepegawaian berhasil lihat daftar validasi pengecekan inventaris ruangan.

4.5.2.51 Use Case Scenario Validasi Laporan Pengecekan Inventaris Ruangan

Pada Tabel 4.54 menggambarkan perilaku aktor mengenai tahapan pada *use case* validasi laporan pengecekan inventaris ruangan.

Tabel 4.54 Use Case Scenario Validasi Laporan Pengecekan Inventaris Ruangan

Flow of Events for Validasi Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.51
<i>Objective</i>	Kepala Subbagian Umum dan Kepegawaian melakukan validasi laporan pengecekan inventaris ruangan.
Aktor	Kepala subbagian umum dan kepegawaian.
<i>Pre Condition</i>	Kepala Subbagian Umum dan Kepegawaian berada pada laman Kepala Subbagian Umum dan Kepegawaian.
<i>Main flow</i>	1. Kepala Subbagian Umum dan Kepegawaian menekan “Daftar Validasi” pada <i>sidebar menu</i>. 2. Kepala Subbagian Umum dan Kepegawaian menekan submenu “Hasil Pengecekan Aset”. 3. Sistem menampilkan daftar hasil pengecekan aset. 4. Kepala Subbagian Umum dan Kepegawaian memilih laporan hasil pengecekan aset yang akan divalidasi dengan menekan tombol “Lihat Detail”.

Tabel 4.54 Use Case Scenario Validasi Laporan Pengecekan Inventaris Ruangan (Lanjutan)

Flow of Events for Validasi Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.51
	5. Sistem menampilkan hasil pengecekan aset. 6. Kepala Subbagian Umum dan Kepegawaian menandai <i>checkbox</i> menekan tombol “Validasi”. 7. Sistem akan menampilkan pesan “Anda Yakin Meminta Validasi laporan?”. 8. Kepala Subbagian Umum dan Kepegawaian. menekan tombol “Validasi”. 9. Sistem menampilkan pesan sukses.
Alternative Flow	7.1 Jika Kepala Subbagian Umum dan Kepegawaian menekan tombol “Tolak Validasi” dan menandai <i>checkbox</i> kemudian menekan simpan pada <i>pop-up message</i> akan menampilkan pesan sukses “Laporan berhasil ditolak!”. 7.2 Jika Kepala Subbagian Umum dan Kepegawaian menekan tombol “Tolak Validasi” dan tanpa menandai <i>checkbox</i> kemudian menekan tombol simpan pada <i>pop-up message</i> maka akan menampilkan pesan <i>error</i> “Anda belum menandai <i>checkbox</i> !”. 7.3 Jika Kepala Subbagian Umum dan Kepegawaian menekan tombol “Validasi” dan tanpa menandai <i>checkbox</i> maka akan menampilkan pesan <i>error</i> “Anda belum menandai <i>checkbox</i> !”.
Post Condition	Kepala Subbagian Umum dan Kepegawaian berhasil melakukan validasi laporan hasil pengecekan inventaris ruangan.

4.5.2.52 Use Case Scenario Export Hasil Laporan Pengecekan Inventaris Ruangan

Pada Tabel 4.55 Kepala Subbagian Umum dan Kepegawaian *use case export* hasil laporan inventaris ruangan.

Tabel 4.55 Use Case Scenario Export Hasil Laporan Pengecekan Inventaris Ruangan

Flow of Events for Export Hasil Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.52
Objective	Pengurus barang bisa meng-export hasil laporan pengecekan inventaris ruangan yang sudah divalidasi.
Aktor	Pengurus barang.
Pre Condition	1. Sebelumnya laporan pengecekan aset sudah divalidasi oleh Kepala Subbagian Umum dan Kepegawaian. 2. Pengurus barang pada laman pengurus barang.
Main flow	1. Pengurus barang menekan “Pengecekan” pada sidebar menu. 2. Pengurus barang menekan submenu “Hasil Laporan Pengecekan”. 3. Sistem menampilkan laman hasil laporan pengecekan. 4. Pengurus barang memasukkan semester, nama penanggung jawab dan juga semester dan menekan tombol <i>export</i>. 5. Sistem akan melihatkan hasil <i>export</i> dalam bentuk pdf sesuai dengan masukan.
Alternative Flow	4.1 Bila ada <i>field</i> yang kosong akan menampilkan <i>error</i> sesuai dengan <i>field</i> kosong.
Post Condition	Penyimpanan barang bisa meng-export hasil laporan pengecekan inventaris ruangan.

4.5.2.53 Use Case Scenario Lihat Daftar Aset

Pada Tabel 4.56 Kepala Subbagian Umum & Kepegawaian *use case* lihat daftar aset.

Tabel 4.56 Use Case Scenario Lihat Daftar Aset

Flow of Events for Lihat Daftar Aset	
Kode Kebutuhan	SIMA-PFS.53
Objective	Pengguna bisa lihat daftar aset.
Aktor	Pengguna
Pre Condition	Semua Pengguna pada laman awal.

Tabel 4.56 Use Case Scenario Lihat Daftar Aset (Lanjutan)

Flow of Events for Lihat Daftar Aset	
Kode Kebutuhan	SIMA-PFS.53
Main flow	1. Pengguna menekan <i>button</i> yang akan dilihat daftar asetnya sesuai pada kategori aset. 2. Sistem menampilkan rincian laman aset yang sudah dipilih.
Alternative Flow	-
Post Condition	Pengguna bisa lihat daftar aset.

4.5.2.54 Use Case Scenario Lihat Daftar Persediaan

Pada Tabel 4.57 Kepala Subbagian Umum dan Kepegawaian *use case* lihat daftar persediaan.

Tabel 4.57 Use Case Scenario Lihat Daftar Persediaan

Flow of Events for Lihat Daftar Persediaan	
Kode Kebutuhan	SIMA-PFS.54
Objective	Pengguna bisa lihat daftar Persediaan.
Aktor	Pengguna
Pre Condition	Semua Pengguna pada laman awal.
Main flow	1. Pengguna menekan tulisan persediaan pada <i>navbar menu</i>. 2. Sistem melihatkan rincian laman persediaan.
Alternative Flow	-
Post Condition	Pengguna bisa lihat daftar persediaan.

4.5.2.55 Use Case Scenario Lihat Daftar Pemegang Kendaraan

Pada Tabel 4.58 Kepala Subbagian Umum dan Kepegawaian *use case* lihat daftar pemegang kendaraan.

Tabel 4.58 Use Case Scenario Lihat Daftar Pemegang Kendaraan

Flow of Events for Lihat Daftar Pemegang Kendaraan	
Kode Kebutuhan	SIMA-PFS.55
Objective	Pengguna bisa lihat daftar persediaan.
Aktor	Pengguna
Pre Condition	Semua Pengguna berada pada laman awal.

Tabel 4.58 Use Case Scenario Lihat Daftar Pemegang Kendaraan (Lanjutan)

<i>Flow of Events for Lihat Daftar Pemegang Kendaraan</i>	
Kode Kebutuhan	SIMA-PFS.55
<i>Main flow</i>	1. Pengguna menekan tulisan kendaraan pada navbar menu. 2. Sistem mellihatkan rincian laman kendaraan.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Pengguna bisa lihat daftar pemegang kendaraan.

4.5.2.56 Use Case Scenario Logout

Pada Tabel 4.59 Kepala Subbagian Umum dan Kepegawaian *use case logout*.

Tabel 4.59 Use Case Scenario Logout

<i>Flow of Events for Logout</i>	
Kode Kebutuhan	SIMA-PFS.56
<i>Objective</i>	Pengguna telah pada laman sesuai dengan aktor.
Aktor	Pengguna
<i>Pre Condition</i>	Pengguna berada pada laman sesuai aktor.
<i>Main flow</i>	1. Pengguna menekan tulisan <i>logout</i>. 2. Sistem mellihatkan laman awal.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Aktor bisa <i>logout</i> .

BAB 5 PERANCANGAN DAN IMPLEMENTASI

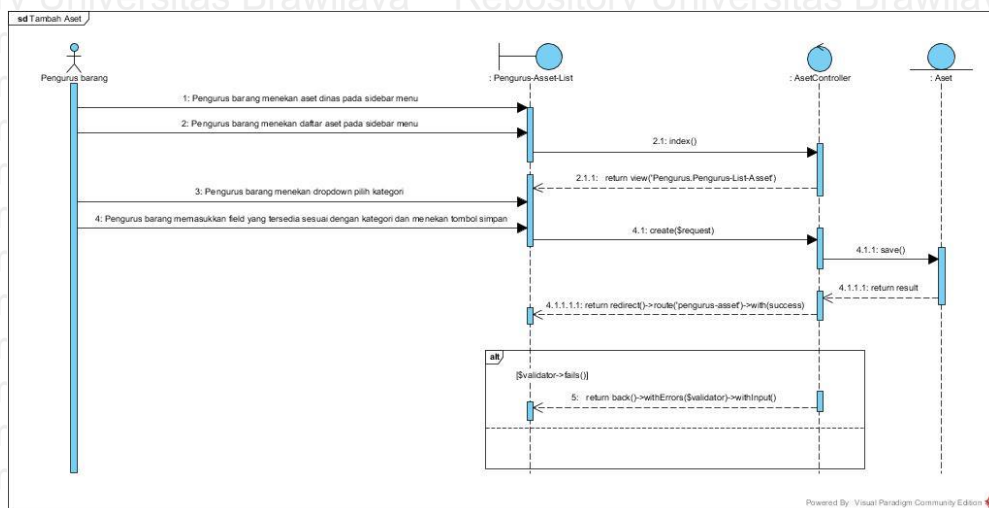
5.1 Perancangan Sistem

5.1.1 Perancangan *Sequence Diagram*

Dalam perancangan *sequence diagram* berisikan aliran dari *use case scenario* yang memberikan uraian mengenai aktor, *method* dan urutan dari *use case scenario*.

5.1.1.1 *Sequence Diagram* Tambah Aset

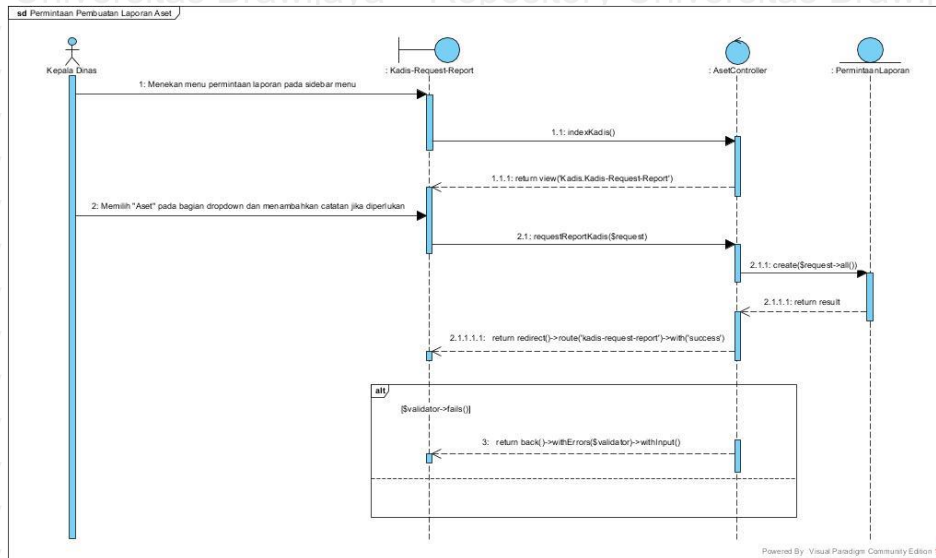
Sequence diagram tambah aset yang dilakukan oleh aktor pengurus barang. Pengurus barang dapat menambahkan aset ke dalam sistem setelah sebelumnya sudah selesai melalui tahapan pemeriksaan aset yang dilakukan oleh bagian pemeriksa aset dari Dinas Perikanan Kabupaten Tabanan. Pertama Pengurus barang menekan tombol aset dinas. Maka selanjutnya akan menampilkan laman daftar aset dan juga tambah aset. Pengurus barang memilih kategori aset yang akan dimasukkan ke dalam pada bagian *dropdown*. Ketika pilihan kategori sudah dipilih maka secara otomatis akan menampilkan *form* yang berisikan *field* sesuai dengan kategori yang telah dipilih sebelumnya. Kemudian pengurus barang mengisi *field* yang tersedia pada *form* tambah aset dan menekan tombol simpan. Untuk alurnya dapat dilihat pada Gambar 5.1.



Gambar 5.1 *Sequence Diagram* Tambah Aset

5.1.1.2 *Sequence Diagram* Permintaan Laporan Aset

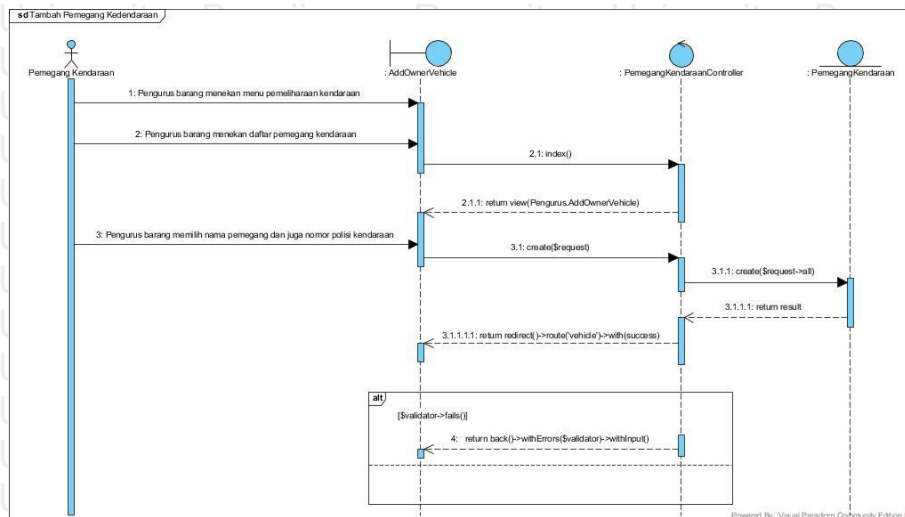
Sequence diagram permintaan pembuatan laporan yang dilakukan oleh aktor kepala dinas. Kepala dinas menekan tombol permintaan laporan pada *sidebar* menu dan menampilkan laman permintaan pembuatan laporan. Kepala dinas menekan *dropdown* yang berisikan pelaporan yang akan diminta untuk dibuat dalam hal ini pelaporan aset. Dan mengisi catatan apabila ada catatan yang ingin disampaikan. Selanjutnya pengurus barang menekan tombol *request* laporan. Dapat dilihat pada Gambar 5.2.



Gambar 5.2 Sequence Diagram Permintaan Laporan Aset

5.1.1.3 Sequence Diagram Tambah Pemegang Kendaraan

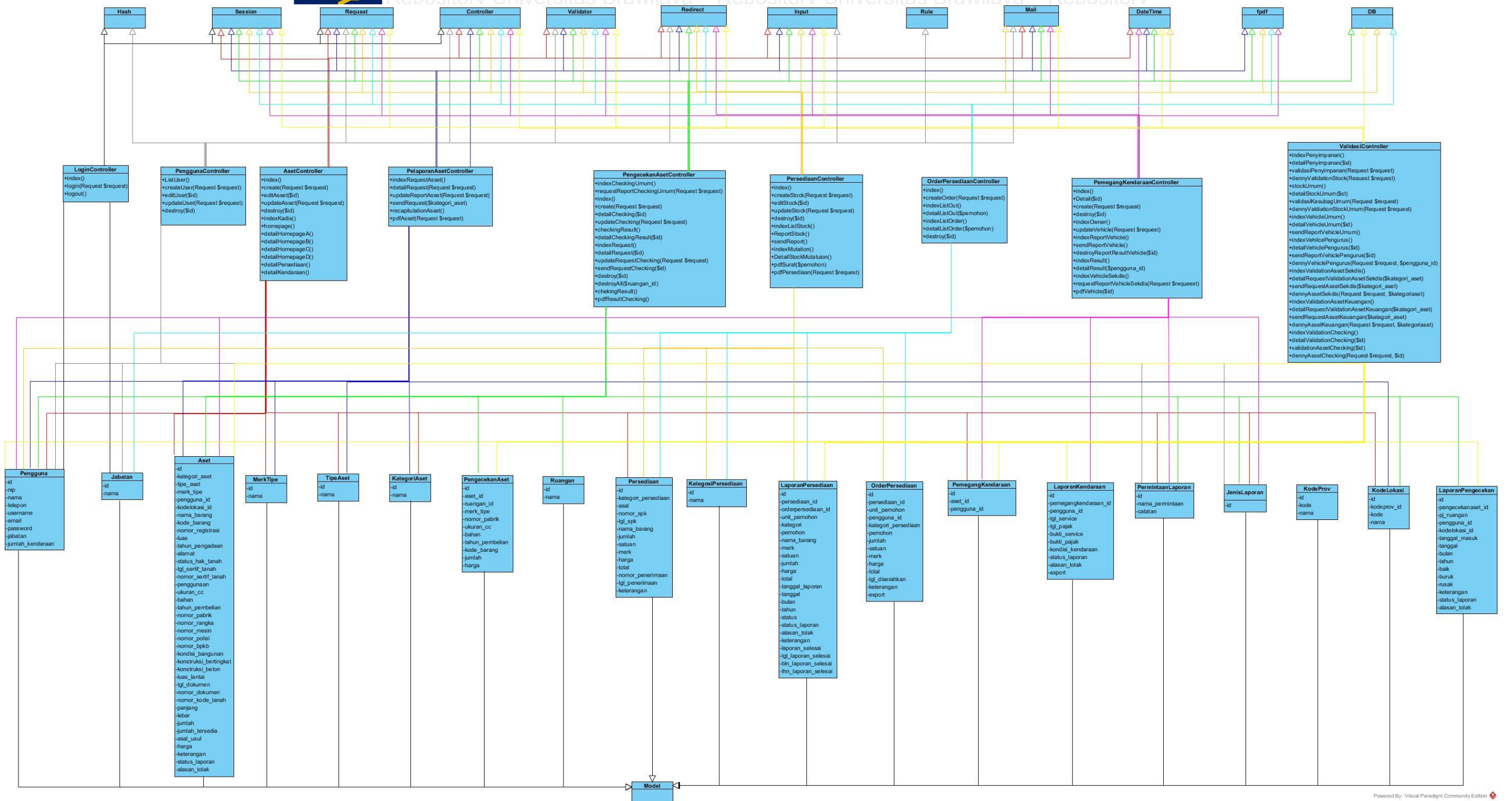
Sequence diagram tambah pemegang kendaraan yang dilakukan oleh aktor pengurus barang. Pengurus barang menekan menu pemeliharaan kendaraan dan menekan submenu daftar pemegang kendaraan pada *sidebar* menu. Nanti akan menampilkan laman daftar pemegang kendaraan dan *form* untuk menambah pemegang kendaraan. Pengurus barang memilih nama dan juga nomor polisi pada *dropdown* pada *form* tambah pemegang kendaraan dan menekan tombol simpan. Untuk *sequence diagram* tambah pemegang kendaraan dapat dilihat pada Gambar 5.3.



Gambar 5.3 Sequence Diagram Tambah Pemegang Kendaraan

5.1.2 Perancangan Class Diagram

Perancangan *class diagram* berisikan nama *attribute* seperti yang terdapat pada Gambar 5.4.



Gambar 5.4 Perancangan Class Diagram



5.1.3 Perancangan *Database*

Pada Gambar 5.5 merupakan *Conceptual Data Model (CDM)* SIMA. CDM merupakan konsep awal dalam pembangunan sistem terutama dalam pembuatan *database*. Perancangan *database* ini terdapat 19 entitas, yaitu : jabatan, pengguna, merk_tipe, tipe_aset, kategori_aset, aset, ruangan, pengecekan_aset, laporan_pengecekan, kategori_persediaan, persediaan, order_persediaan, laporan_persediaan, pemegang_kendaraan, laporan_kendaraan, permintaan_laporan, jenis_laporan, kode_prov, dan kode_lokasi.

5.1.4 Perancangan Antarmuka

Antarmuka pengguna atau *user interface* adalah tampilan yang digunakan untuk berinteraksi antara pengguna dengan komputer. Maka dari desain antarmuka harus bisa *user friendly*, atau bisa dikatakan suatu desain *user interface* yang bagus apabila dapat mempermudah penggunaanya dalam menggunakan sistem tersebut. Berikut adalah beberapa perancangan antarmuka yang akan dibuat.

5.1.4.1 Perancangan Antarmuka Tambah Aset

Gambar 5.6 merupakan rancangan antarmuka Tambah Aset. Untuk penjelasan *wireframe* secara lebih spesifik dijelaskan pada Tabel 5.4.

Gambar 5.5 Perancangan Antarmuka Tambah Aset



Tabel 5.1 Keterangan Perancangan Antarmuka Tambah Aset

No	Nama Komponen	Type	Penjelasan
1	Nama sistem	<i>Unordered List</i>	Melihatkan SIMA.
2	Tombol menu	<i>Unordered List</i>	Berfungsi melihatkan dan menyembunyikan <i>sidebar</i> menu.
3	Nama pengguna	<i>Unordered List</i>	Berfungsi melihatkan nama pengguna yang berhasil <i>login</i> .
4	Tombol <i>logout</i>	<i>Unordered List</i>	Berfungsi untuk keluar dari sistem.
5	<i>Sidebar</i> menu	<i>Unordered List</i>	Berfungsi untuk menampilkan <i>sidebar</i> menu sesuai dengan pengguna yang <i>login</i> .
6	Tanda tambah	<i>Button</i>	Berfungsi untuk menampilkan dan menyembunyikan <i>form</i> tambah aset.
7-9	Nama <i>field</i>	<i>Label</i>	Berfungsi untuk menampilkan nama <i>field</i> yang akan ditampilkan.
10-12	Masukkan <i>field</i>	<i>Text field</i>	Berisikan <i>field</i> data aset yang akan dimasukkan ke dalam sistem.
13	Simpan	<i>Button</i>	Berfungsi dalam menyimpan data yang telah dimasukkan pada <i>text field</i> ke dalam <i>database</i> .
14-18	<i>Header table</i>	<i>Button</i>	Berfungsi membedakan aset berdasarkan Kartu Inventaris Barang (KIB).
19	<i>Show</i>	<i>Combo box</i>	Berguna dalam melihat jumlah baris pada tabel.
20	<i>Search</i>	<i>Text field</i>	Untuk pencarian pada tabel.
21	Tabel	<i>Table</i>	Tabel yang berfungsi untuk menampilkan data aset.
22	Sunting	<i>Button</i>	Berfungsi untuk menampilkan data aset yang akan diperbarui pada <i>modal-edit</i> .

Tabel 5.1 Keterangan Perancangan Antarmuka Tambah Aset (Lanjutan)

No	Nama Komponen	Tipe	Penjelasan
23	Hapus	<i>Button</i>	Berfungsi untuk menghapus data pengguna pada tabel.
24-25	<i>Next and previous</i>	<i>Button</i>	Berfungsi untuk melihat laman selanjutnya maupun sebelumnya.
26	<i>Footer</i>	<i>Label</i>	Berisikan Dinas Perikanan, tahun dan juga versi dari aplikasi.

5.1.4.2 Perancangan Antarmuka Permintaan Laporan Aset

Gambar 5.7 merupakan rancangan antarmuka Permintaan Laporan Aset. Untuk penjelasan *wireframe* secara pada Tabel 5.5.

Gambar 5.7 Perancangan Antarmuka Permintaan Pembuatan Laporan Aset

Tabel 5.2 Keterangan Perancangan Antarmuka Permintaan Pembuatan Laporan Aset

No	Nama Komponen	Tipe	Penjelasan
1	Nama sistem	<i>Unordered List</i>	Melihatkan SIMA.
2	Tombol menu	<i>Unordered List</i>	Berfungsi untuk melihat dan menyembunyikan <i>sidebar</i> menu.
3	Nama pengguna	<i>Unordered List</i>	Berfungsi untuk melihat nama pengguna yang berhasil <i>login</i> .



Tabel 5.2 Keterangan Perancangan Antarmuka Permintaan Pembuatan Laporan Aset (Lanjutan)

No	Nama Komponen	Tipe	Penjelasan
4	Tombol <i>logout</i>	<i>Unordered List</i>	Berfungsi untuk keluar dari sistem.
5	<i>Sidebar</i> menu	<i>Unordered List</i>	Berfungsi untuk menampilkan <i>sidebar</i> menu sesuai dengan pengguna yang <i>login</i> .
6-7	Nama <i>field</i>	<i>Label</i>	Berfungsi melihatkan nama <i>filed</i> yang akan ditampilkan.
8	Masukkan <i>field</i>	<i>Combo box</i>	Berisikan laporan yang akan diminta.
9	Masukkan <i>field</i>	<i>Textare</i>	Berfungsi untuk tempat memasukkan catatan.
10	<i>Request</i> Laporan	<i>Button</i>	Berfungsi menyimpan data ke <i>database</i> .
11	<i>Footer</i>	<i>Label</i>	Berisikan Dinas Perikanan, tahun dan juga versi dari aplikasi.

5.1.4.3 Perancangan Antarmuka Tambah Pemegang Kendaraan

Gambar 5.8 merupakan rancangan antarmuka Tambah Pemegang Kendaraan. Untuk penjelasan *wireframe* pada Tabel 5.6.

Gambar 5.8 Perancangan Antarmuka Tambah Pemegang Kendaraan



Tabel 5.3 Keterangan Perancangan Antarmuka Tambah Pemegang Kendaraan

No	Nama Komponen	Type	Penjelasan
1	Nama sistem	<i>Unordered List</i>	Melihatkan SIMA.
2	Tombol menu	<i>Unordered List</i>	Berfungsi melihatkan dan menyembunyikan <i>sidebar</i> menu.
3	Nama pengguna	<i>Unordered List</i>	Berfungsi melihatkan nama pengguna yang berhasil <i>login</i> .
4	Tombol <i>logout</i>	<i>Unordered List</i>	Berfungsi untuk keluar dari sistem.
5	<i>Sidebar</i> menu	<i>Unordered List</i>	Berfungsi untuk menampilkan <i>sidebar</i> menu sesuai dengan pengguna yang <i>login</i> .
6-7	Nama <i>field</i>	<i>Label</i>	Berfungsi untuk menampilkan nama <i>field</i> yang akan ditampilkan.
8-9	Masukkan <i>field</i>	<i>Combo box</i>	Berisikan nama pengguna dan nomor polisi kendaraan.
10	Simpan	<i>Button</i>	Berfungsi menyimpan data ke <i>database</i> .
11	<i>Show</i>	<i>Combo box</i>	Berguna dalam melihat jumlah baris pada tabel.
12	<i>Search</i>	<i>Text field</i>	Untuk pencarian pada tabel.
13	Tabel	<i>Table</i>	Tabel yang berfungsi untuk menampilkan data pemegang kendaraan.
14-16	Lihat <i>detail</i>	<i>Button</i>	Berfungsi untuk melihat rincian kendaraan yang dipegang oleh pemegang kendaraan.
17-18	<i>Next</i> and <i>previous</i>	<i>Button</i>	Berguna dalam melihat laman selanjutnya maupun sebelumnya.
19	<i>Footer</i>	<i>Label</i>	Berisikan Dinas Perikanan, tahun dan juga versi dari aplikasi.



5.2 Spesifikasi Sistem

5.2.1 Perancangan Komponen

Pada tahapan perancangan komponen dengan melakukan rancangan algoritme dalam bentuk *pseudocode* guna untuk memperjelas alur logika yang digunakan dalam pengembangan aplikasi. *Pseudocode* dipakai untuk acuan pada tahap implementasi kode program.

5.2.1.1 Perancangan Komponen Tambah Aset Dengan Fungsi *Create*

Pada Tabel 5.1 merupakan alur logika dari fungsi *create*. Pertama melakukan validasi terhadap masukan oleh pengurus barang ketika tidak sesuai akan melihatkan pesan *error* bahwa terdapat *field form* kosong. Jika validator sudah sesuai maka akan menyimpan data aset ke *database*.

Nama *class* : AsetController

Nama fungsi : *create()*

Algoritme :

Tabel 5.4 Algoritme Fungsi *create()*

Pseudocode Fungsi <i>create()</i>	
1	Mulai
2	aseta <- nilai aset a asetb <- nilai aset b asetc <- nilai aset c asetd <- nilai aset d asete <- nilai aset e
3	aset <- objek Aset baru
4	IF (kategori_aset==1)
5	validator <- memvalidasi input dari variabel aSETa
6	IF (validator->fails() ==TRUE)
7	Return pesan error ENDIF
8	aset->kodelokasi_id = Auth()->id() <- mencari id kodelokasi_id kodelokasi <- nilai input kodelokasi_id dalam objek aset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomorregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset

Tabel 5.4 Algoritme Fungsi *create()* (Lanjutan)

Pseudocode Fungsi <i>create()</i>	
	kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset <- nilai dari input tipe_aset luas <- nilai dari input luas tahun_pengadaan <- nilai dari input tahun_pengadaan alamat <- nilai dari input alamat status_hak_tanah <- nilai dari input status_hak_tanah tgl_sertif_tanah <- nilai dari input tgl_sertif_tanah nomor_sertif_tanah <- nilai dari input nomor_sertif_tanah penggunaan <- nilai dari input penggunaan harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
9	ELSEIF (kategori_aset==2)
10	validator <- memvalidasi input dari variabel asetb
11	IF (validator->fails() ==TRUE)
12	Return pesan error ENDIF
13	aset->kodelokasi_id = Auth()->id() kodelokasi <- nilai input kodelokasi_id dalam objek aset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomerregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset <- nilai dari input tipe_aset merk_tipe <- nilai dari input merk_tipe ukuran_cc <- nilai dari input ukuran_cc bahan <- nilai dari input bahan tahun_pembelian <- nilai dari input tahun_pembelian nomor_pabrik <- nilai dari input nomor_pabrik nomor_mesin = nilai dari input nomor_mesin nomor_rangka = nilai dari input nomor_rangka

Tabel 5.4 Algoritme Fungsi *create()* (Lanjutan)

Pseudocode Fungsi <i>create()</i>	
	nomor_polisi = nilai dari input nomor_polisi nomor_bpkb <- nilai dari input nomor_bpkb jumlah <- nilai dari input jumlah jumlah_tersedia <- nilai dari input jumlah harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
14	Elseif (kategori_aset==3)
15	validator <- memvalidasi input dari variabel asetc
16	IF (validator->fails() ==TRUE)
17	Return pesan error ENDIF
18	aset->kodelokasi_id = Auth()->id() kodelokasi <- nilai input kodelokasi_id dalam objek aset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomerregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset = nilai dari input variabel tipe_aset kondisi_bangunan <- nilai dari input kondisi_bangunan konstruksi_bertikat <- nilai dari input konstruksi_bertikat konstruksi_beton <- nilai dari input konstruksi_beton; luas_lantai <- nilai dari input luas_lantai alamat <- nilai dari input alamat tgl_dokumen <- nilai dari input tgl_dokumen nomor_dokumen <- nilai dari input nomor_dokumen luas <- nilai dari input luas; status_hak_tanah <- nilai dari input status_hak_tanah nomor_kode_tanah <- nilai dari input nomor_kode_tanah harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul

Tabel 5.4 Algoritme Fungsi *create()* (Lanjutan)

Pseudocode Fungsi <i>create()</i>	
	keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
19	IF (kategori_aset==4)
20	validator <- memvalidasi input dari variabel asetc
21	IF (validator->fails() ==TRUE)
22	Return pesan error ENDIF
23	aset->kodelokasi_id = Auth()->id() kodelokasi <- nilai input kodelokasi_id dalam objek aset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomorregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset = nilai dari input variabel tipe_aset konstruksi_beton <- nilai dari input konstruksi_beton panjang <- nilai dari input panjang lebar <- nilai dari input lebar luas <- nilai dari input luas alamat <- nilai dari input alamat tgl_dokumen <- nilai dari input tgl_dokumen nomor_dokumen <- nilai dari input nomor_dokumen status_hak_tanah <- nilai dari input status_hak_tanah nomor_kode_tanah <- nilai dari input nomor_kode_tanah harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
24	ELSE
25	validator <- memvalidasi input dari variabel asetc
26	IF (validator->fails() ==TRUE)
27	Return pesan error ENDIF
28	kodelokasi_id <- inputan kodelokasi_id = NULL kategori <- nilai input dari kategori_aset

Tabel 5.4 Algoritme Fungsi *create()* (Lanjutan)

Pseudocode Fungsi <i>create()</i>	
	<pre> kodekategori <- menghitung kategori_aset pada objek Aset nores = variabel kodekategori + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset < nilai dari input kategori_aset tipe_aset <- nilai dari input tipe_aset merk_tipe <- nilai dari input merk_tipe ukuran_cc <- nilai dari input ukuran_cc bahan <- nilai dari input bahan tahun_pembelian <- nilai dari input variabel tahun_pembelian jumlah <- nilai dari input jumlah; jumlah_tersedia <- nilai dari input jumlah harga <- nilai dari input variabel harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- id dari sesi pengguna </pre>
29	ENDIF
30	save() <- menyimpan data dari objek aset
31	Redirect ke halaman aset dinas dan menampilkan pesan sukses
32	Selesai

5.2.1.2 Perancangan Komponen Permintaan Laporan Aset dengan Fungsi *requestReportKadis*

Pada tabel 5.2 merupakan alur logika dari fungsi *requestReportKadis*. Pertama melakukan validasi terhadap masukan yang dimasukkan oleh kepala dinas ketika tidak sesuai nanti melihatkan pesan *error* bahwa ada *field form* kosong. Jika validator sudah sesuai maka akan menyimpan permintaan laporan ke *database* dan mengirimkan notifikasi *email* ke pengurus barang.

Nama class : AsetController
 Nama fungsi : *requestReportKadis()*
 Algoritme :

Tabel 5.5 Algoritme Fungsi *requestReportKadis()*

Pseudocode Fungsi <i>requestReportKadis()</i>	
1	Mulai

Tabel 5.5 Algoritme Fungsi *requestReportkadis()* (Lanjutan)

Pseudocode Fungsi <i>requestReportKadis()</i>	
2	validator <- memvalidasi input yang dikirimkan
3	IF (validator->fails() ==TRUE)
4	Return pesan error ENDIF
5	Menyimpan data objek PermintaanLaporan ke database
6	if (nama_permintaan = 1)
7	month <- mengambil bulan saat ini
8	If (month <=6)
9	semester = "Semester 1 dari bulan Januari s/d Juni"
10	Else semester = "Semester 2 dari bulan Juli s/d Desember"
11	END IF
12	now <- dari inputan waktu saat ini dari objek datetime, date = now+format(d-m-Y), catatan <- mengambil data dari inputan, semester = mengambil data dari inputan, permintaanlaporan <- mengambil nama_permanintaan=1 pada objek PermintaanLaporan permintaanlaporan1 <- mengambil nama_permanintaan=1 pada objek PermintaanLaporan pengguna <- mengambil jabatan = 7 pada objek Pengguna pengguna1 <- mengambil jabatan = 2 pada objek Pengguna pengguna2 <- mengambil jabatan = 5 pada objek Pengguna pengguna3 <- mengambil jabatan = 3 pada objek Pengguna to<- nilai dari email penggunan cc <- nilai array dari email pengguna2 dan email pengguna3 form <- nilai email dari pengguna1 subject <- "Pembuatan Laporan Aset" Mengirimkan email
13	ELSEIF (nama_permintaan = 2)
14	month <- mengambil bulan saat ini
15	If (month <=6)
16	semester = "Semester 1 dari bulan Januari s/d Juni"
17	Else semester = "Semester 2 dari bulan Juli s/d Desember"
18	END IF

Tabel 5.5 Algoritme Fungsi *requestReportkadis()* (Lanjutan)

Pseudocode Fungsi <i>requestReportKadis()</i>	
19	<pre> now <- dari inputan waktu saat ini dari objek datetime, date = now+format(d-m-Y), catatan <- mengambil data dari inputan, semester = mengambil data dari inputan, permintaanlaporan <- mengambil nama_permanintaan=2 pada objek PermintaanLaporan permintaanlaporan1 <- mengambil nama_permanintaan=2 pada objek PermintaanLaporan pengguna <- mengambil jabatan = 6 pada objek Pengguna pengguna1 <- mengambil jabatan = 2 pada objek Pengguna pengguna2 <- mengambil jabatan = 4 pada objek Pengguna to <- nilai dari email penggunan cc <- nilai array dari email pengguna2 dan email pengguna3 form <- nilai email dari pengguna1 subject <- "Pembuatan Laporan Persediaan" Mengirimkan email </pre>
20	Endif
21	Redirect ke halaman permintaan laporan dan menampilkan pesan berhasil
22	Selesai

5.2.1.3 Perancangan Komponen Tambah Pemegang Kendaraan Dengan Fungsi *Create()*

Pada Tabel 5.3 merupakan alur logika dari fungsi *create*. Pertama melakukan validasi terhadap masukan yang dimasukkan oleh pengurus barang ketika tidak sesuai akan menampilkan pesan *error* bahwa ada *field form* kosong. Jika validator sudah sesuai maka akan menyimpan data pemegang kendaraan berdasarkan *request id* ke *database* dengan mengurangi pada *field* jumlah_tersedia sesuai dengan aset yang diambil pada *table* aset dan menambahkan pada *field* jumlah_kendaraan pada *table* pengguna sesuai dengan pengguna yang dipilih untuk memegang kendaraan.

Nama *class* : PemegangKendaraanController

Nama fungsi : *create()*

Algoritme :

Tabel 5.6 Algoritme Fungsi *create()*

Pseudocode Fungsi <i>create()</i>	
1	Mulai
2	validator <- memvalidasi input yang dikirimkan
3	IF (validator->fails() ==TRUE)
4	Return pesan error ENDIF
5	Mencari aset_id pada objek Aset yang diinputkan dan decrement jumlah_tersedia = 1 Mencari pengguna_id pada objek Pengguna yang diinputkan dan increment jumlah_kendaraan = 1 Menyimpan data pada objek PemegangKendaraan ke database
6	Redirect dan menampilkan pesan berhasil
7	Selesai

5.2.2 Spesifikasi *Hardware*

Spesifikasi *hardware* yang digunakan dalam membangun sistem. Dapat dilihat pada Tabel 5.7.

Tabel 5.7 Spesifikasi *Hardware*

Elemen	Spesifikasi
<i>System Model</i>	Aspire 45-471G
<i>Processor</i>	Intel(R) Core(TM)i3-2367M CPU @ 1.40GHz(4 CPUs), ~1.4GHz
<i>Memory</i>	8 GB RAM
<i>Display</i>	Intel(R) HD Graphics 3000
<i>Hard Disk</i>	500 GB

5.2.3 Spesifikasi *Software*

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem dapat dilihat pada Tabel 5.8.

Tabel 5.8 Spesifikasi *Software*

Elemen	Spesifikasi
Bahasa Pemrograman	PHP 7.1, HTML
Kerangka kerja	Laravel 5.7, Bootstrap
<i>Database</i>	MySQL
<i>Web Server</i>	Apache



5.3 Implementasi Sistem

5.3.1 Implementasi Komponen

5.3.1.1 Implementasi Perancangan Komponen Tambah Aset Dengan Fungsi *create*

Pada tabel 5.9 merupakan alur logika dari fungsi *create*. Pertama melakukan validasi terhadap masukan yang dimasukkan oleh pengurus barang ketika tidak sesuai akan melihatkan pesan *error* bahwa ada *field form* kosong. Jika validator sudah sesuai maka akan menyimpan data aset ke *database*.

Nama *class* : AsetController

Nama fungsi : *create()*

Tabel 5.9 Kode Program Fungsi *create()*

Kode Program Fungsi <i>create()</i>	
1	public function create(Request \$request)
2	{
3	\$aseta = ([
4	'kategori_aset' => 'required',
5	'kodelokasi_id' => 'required',
6	'nama_barang' => 'required max:100',
7	'kode_barang' => 'required max:100',
8	'tipe_aset' => 'required',
9	'luas' => 'required max:100',
10	'tahun_pengadaan' => 'required max:100',
11	'alamat' => 'required max:100',
12	'status_hak_tanah' => 'required max:100',
13	'tgl_sertif_tanah' => 'required max:100',
14	'nomor_sertif_tanah' => 'required max:100',
15	'penggunaan' => 'required max:100',
16	'asal_usul' => 'required max:100',
17	'harga' => 'required',
18	'keterangan'
19	]);
20	\$asetb = ([
21	'kategori_aset' => 'required',
22	'kodelokasi_id' => 'required',
23	'nama_barang' => 'required max:100',
24	'kode_barang' => 'required max:100',
25	'tipe_aset' => 'required',
26	'merk_tipe' => 'required max:100',
27	'ukuran_cc' => 'max:100',
28	'bahan' => 'required max:100',
29	'tahun_pembelian' => 'required',
30	'nomor_pabrik' => 'max:100',
31	'nomor_rangka' => 'max:100',
32	'nomor_mesin' => 'max:100',
33	'nomor_polisi' => 'max:100',
34	'nomor_bpkb' => 'max:100',
35	'jumlah' => 'required max:100',
36	'asal_usul' => 'required max:100',
37	'harga' => 'required',
38	'keterangan'
39	]);

Tabel 5.9 Kode Program Fungsi *create()* (Lanjutan)

Kode Program Fungsi <i>create</i>	
40	\$asetc = ([
41	'kategori_aset' => 'required',
42	'kodelokasi_id' => 'required',
43	'nama_barang' => 'required max:100',
44	'kode_barang' => 'required max:100',
45	'tipe_aset' => 'required',
46	'kondisi_bangunan' => 'required max:100',
47	'konstruksi_bertikat' => 'required max:100',
48	'konstruksi_beton' => 'required max:100',
49	'luas_lantai' => 'required max:100',
50	'alamat' => 'required max:100 100',
51	'tgl_dokumen' => 'required max:100',
52	'nomor_dokumen' => 'required max:100',
53	'luas' => 'max:100',
54	'status_hak_tanah' => 'required max:100',
55	'nomor_kode_tanah' => 'max:100',
56	'asal_usul' => 'required max:100',
57	'harga' => 'required',
58	'keterangan'
60	]);
61	\$asetd = ([
62	'kategori_aset' => 'required',
63	'kodelokasi_id' => 'required',
64	'nama_barang' => 'required max:100',
65	'kode_barang' => 'required max:100 100',
66	'tipe_aset' => 'required',
67	'konstruksi_beton' => 'required max:100',
68	'panjang' => 'required',
69	'lebar' => 'required',
70	'luas' => 'required',
71	'alamat' => 'required max:100',
72	'tgl_dokumen' => 'required max:100',
73	'nomor_dokumen' => 'required max:100',
74	'status_hak_tanah' => 'required max:100',
75	'nomor_kode_tanah' => 'max:100',
76	'asal_usul' => 'required max:100',
77	'harga' => 'required',
78	'keterangan',
79	]);
80	
81	\$asete = ([
82	'kategori_aset' => 'required',
83	'nama_barang' => 'required max:100',
84	'kode_barang' => 'required max:100',
85	'tipe_aset' => 'max:100',
86	'merk_tipe' => 'max:100',
87	'ukuran_cc' => 'max:100',
88	'bahan' => 'required max:100',
89	'tahun_pembelian' => 'required',
90	'jumlah' => 'required max:100',
91	'asal_usul' => 'required max:100',
92	'harga' => 'required',
93	'keterangan',
94	]);



Tabel 5.9 Kode Program Fungsi *create()* (Lanjutan)

Kode	Program Fungsi <i>create</i>
95	
96	
97	<code>\$aset = new Aset ;</code>
98	<code>if (\$request->kategori_aset==1) {</code>
99	
100	<code>\$validator = Validator::make(\$request->all(),</code>
101	<code>\$aset);</code>
102	<code>if (\$validator->fails()) {</code>
103	<code>return back()->withErrors(\$validator)-</code>
104	<code>>withInput();</code>
105	<code>}</code>
106	<code>\$aset->kodelokasi_id = Auth()->id();</code>
107	<code>\$kodelokasi = \$request->kodelokasi_id;</code>
108	<code>\$nomorregistrasi =</code>
109	<code>Aset::where('kodelokasi_id',\$kodelokasi)->count();</code>
110	<code>\$nores = \$nomorregistrasi + 1;</code>
111	<code>\$aset->nomor_registrasi = '000'.\$nores;</code>
112	
113	<code>\$aset->kode_barang = \$request->kode_barang;</code>
114	<code>\$aset->nama_barang = \$request->nama_barang;</code>
115	<code>\$aset->kategori_aset = \$request-</code>
116	<code>>kategori_aset;</code>
117	<code>\$aset->kodelokasi_id = \$request-</code>
118	<code>>kodelokasi_id;</code>
119	<code>\$aset->type_aset = \$request->type_aset;</code>
120	<code>\$aset->luas = \$request->luas;</code>
121	<code>\$aset->tahun_pengadaan = \$request->tahun_pengadaan;</code>
122	<code>\$aset->alamat = \$request->alamat;</code>
123	<code>\$aset->status_hak_tanah = \$request-</code>
124	<code>>status_hak_tanah;</code>
125	<code>\$aset->tgl_sertif_tanah= \$request->tgl_sertif_tanah;</code>
126	<code>\$aset->nomor_sertif_tanah = \$request-</code>
127	<code>>nomor_sertif_tanah;</code>
128	<code>\$aset->penggunaan = \$request->penggunaan;</code>
129	<code>\$aset->harga = \$request->harga;</code>
130	<code>\$aset->asal_usul = \$request->asal_usul;</code>
131	<code>\$aset->keterangan = \$request->keterangan;</code>
132	<code>\$aset->pengguna_id = Session::get('id');</code>
133	
134	<code>}elseif (\$request->kategori_aset==2) {</code>
135	<code>\$validator = Validator::make(\$request->all(),</code>
136	<code>\$asetb);</code>
137	
138	<code>if (\$validator->fails()) {</code>
139	<code>return back()->withErrors(\$validator)-</code>
140	<code>>withInput();</code>
141	<code>}</code>
142	<code>\$aset->kodelokasi_id = Auth()->id();</code>
143	<code>\$kodelokasi = \$request->kodelokasi_id;</code>
144	<code>\$nomorregistrasi =</code>
145	<code>Aset::where('kodelokasi_id',\$kodelokasi)->count();</code>
146	<code>\$nores = \$nomorregistrasi + 1;</code>
147	<code>\$aset->nomor_registrasi = '000'.\$nores;</code>
148	
149	



Tabel 5.9 Kode Program Fungsi *create()* (Lanjutan)

Kode Program Fungsi <i>create</i>		
150	\$aset->kode_barang	= \$request->kode_barang;
151	\$aset->nama_barang	= \$request->nama_barang;
152	\$aset->kategori_aset	= \$request->kategori_aset;
153	\$aset->kodelokasi_id	= \$request->kodelokasi_id;
154	\$aset->tipe_aset	= \$request->tipe_aset;
155	\$aset->merk_tipe	= \$request->merk_tipe;
156	\$aset->ukuran_cc	= \$request->ukuran_cc;
157	\$aset->bahan	= \$request->bahan;
158	\$aset->tahun_pembelian	= \$request->
159	tahun_pembelian;	
160	\$aset->nomor_pabrik	= \$request->nomor_pabrik;
161	\$aset->nomor_mesin	= \$request->nomor_mesin;
162	\$aset->nomor_rangka	= \$request->nomor_rangka;
163	\$aset->nomor_polisi	= \$request->nomor_polisi;
164	\$aset->nomor_bpkb	= \$request->nomor_bpkb;
165	\$aset->jumlah	= \$request->jumlah;
166	\$aset->jumlah_tersedia	= \$request->jumlah;
167	\$aset->asal_usul	= \$request->asal_usul;
168	\$aset->harga	= \$request->harga;
169	\$aset->keterangan	= \$request->keterangan;
170	\$aset->pengguna_id	= Session::get('id');
171		
172	}elseif (\$request->kategori_aset==3) {	
173	\$validator	= Validator::make(\$request->all(),
174	\$asetc);	
175	if (\$validator->fails()) {	
176	return back()->withErrors(\$validator)->withInput();	
177	}	
178	\$aset->kodelokasi_id	= Auth()->id();
179	\$kodelokasi	= \$request->kodelokasi_id;
180	\$nomorregistrasi	=
181	Aset::where('kodelokasi_id',\$kodelokasi)->count();	
182	\$nores = '000'.\$nomorregistrasi + 1;	
183	\$aset->nomor_registrasi	= '000'.\$nores;
184		
185	\$aset->kode_barang	= \$request->kode_barang;
186	\$aset->nama_barang	= \$request->nama_barang;
187	\$aset->kategori_aset	= \$request->kategori_aset;
188	\$aset->kodelokasi_id	= \$request->kodelokasi_id;
189	\$aset->tipe_aset	= \$request->tipe_aset;
190	\$aset->kondisi_bangunan	= \$request->
191	>kondisi_bangunan;	
192	\$aset->konstruksi_bertikat=	\$request->
193	>konstruksi_bertikat;	
194	\$aset->konstruksi_beton	= \$request->
195	>konstruksi_beton;	
196	\$aset->luas_lantai	= \$request->luas_lantai;
197	\$aset->alamat	= \$request->alamat;
198	\$aset->tgl_dokumen	= \$request->tgl_dokumen;
199	\$aset->nomor_dokumen	= \$request->nomor_dokumen;
200	\$aset->luas	= \$request->luas;
201	\$aset->status_hak_tanah	= \$request->
202	>status_hak_tanah;	
203	\$aset->nomor_kode_tanah	= \$request->
204	>nomor kode tanah;	

**Tabel 5.9 Kode Program Fungsi *create()* (Lanjutan)**

Kode Program Fungsi <i>create</i>	
205	\$aset->asal_usul = \$request->asal_usul;
206	\$aset->harga = \$request->harga;
207	\$aset->keterangan = \$request->keterangan;
208	\$aset->pengguna_id = Session::get('id');
209	
210	}elseif (\$request->kategori_aset==4) {
211	\$validator = Validator::make(\$request->all(),
212	\$asetd);
213	
214	if (\$validator->fails()) {
215	return back()->withErrors(\$validator)->withInput();
216	}
217	\$aset->kodelokasi_id = Auth()->id();
218	\$kodelokasi = \$request->kodelokasi_id;
219	\$nomorregistrasi =
220	Aset::where('kodelokasi_id',\$kodelokasi)->count();
221	\$nores = \$nomorregistrasi + 1;
222	\$aset->nomor_registrasi = '000'.\$nores;
223	\$aset->kode_barang = \$request->kode_barang;
224	\$aset->nama_barang = \$request->nama_barang;
225	\$aset->kategori_aset = \$request->kategori_aset;
226	\$aset->kodelokasi_id = \$request->kodelokasi_id;
227	\$aset->tipe_aset = \$request->tipe_aset;
228	\$aset->konstruksi_beton = \$request->
229	>konstruksi_beton;
230	\$aset->panjang = \$request->panjang;
231	\$aset->lebar = \$request->lebar;
232	\$aset->luas = \$request->luas;
233	\$aset->alamat = \$request->alamat;
234	\$aset->tgl_dokumen = \$request->tgl_dokumen;
235	\$aset->nomor_dokumen = \$request->nomor_dokumen;
236	\$aset->status_hak_tanah = \$request->
237	>status_hak_tanah;
238	\$aset->nomor_kode_tanah = \$request->
239	>nomor_kode_tanah;
240	\$aset->asal_usul = \$request->asal_usul;
241	\$aset->harga = \$request->harga;
242	\$aset->kondisi_bangunan = \$request->
243	>kondisi_bangunan;
244	\$aset->keterangan = \$request->keterangan;
245	\$aset->pengguna_id = Session::get('id');
246	
247	}else {
248	\$validator = Validator::make(\$request->all(),
249	\$asete);
250	
251	if (\$validator->fails()) {
252	return back()->withErrors(\$validator)-
253	>withInput();
254	}

Tabel 5.9 Kode Program Fungsi *create()* (Lanjutan)

Kode Program Fungsi <i>create</i>	
255	\$request->kodelokasi_id ==NULL;
256	\$kategori = \$request->kategori_aset;
257	\$kodekategori
258	Aset::where('kategori_aset',\$kategori)->count();
259	\$nores = \$kodekategori + 1;
260	\$aset->nomor_registrasi = '000'.\$nores;
261	
262	\$aset->kode_barang = \$request->kode_barang;
263	\$aset->nama_barang = \$request->nama_barang;
264	\$aset->kategori_aset = \$request->kategori_aset;
265	\$aset->tiipe_aset = \$request->tiipe_aset;
266	\$aset->merk_tipe = \$request->merk_tipe;
267	\$aset->ukuran_cc = \$request->ukuran_cc;
268	\$aset->bahan = \$request->bahan;
269	\$aset->tahun_pembelian = \$request->
270	>tahun_pembelian;
271	\$aset->jumlah_tersedia = \$request->jumlah;
272	\$aset->jumlah = \$request->jumlah;
273	\$aset->asal_usul = \$request->asal_usul;
274	\$aset->harga = \$request->harga;
275	\$aset->keterangan = \$request->keterangan;
276	\$aset->pengguna_id = Session::get('id');
277	}
278	\$aset->save();
279	return redirect()->route('pengurus-asset')-
280	>with(['success' => 'Aset berhasil ditambah!']);
281	}

Tabel 5.10 Penjelasan Kode Program Fungsi *create()*

Penjelasan Fungsi <i>create()</i>	
1	Membuat fungsi <i>create</i> dengan masukan dengan parameter <i>request</i> .
3-19	Inisialisasi variabel aseta dengan array berdasarkan <i>fields</i> yang ada pada aseta.
20-39	Inisialisasi variabel asetb dengan array berdasarkan <i>fields</i> yang ada pada asetb.
40-60	Inisialisasi variabel asetc dengan array berdasarkan <i>fields</i> yang ada pada asetc.
61-79	Inisialisasi variabel asetd dengan array berdasarkan <i>fields</i> yang ada pada asetd.
81-94	Inisialisasi variabel asete dengan array berdasarkan <i>fields</i> yang ada pada asete.
97	Inisialisasi variabel aset = objek Aset baru.
98	Seleksi kondisi jika masukan kategori_aset = 1 maka <i>statement</i> ini yang akan dijalankan.
100	Inisialisasi variabel validator berfungsi untuk memvalidasi masukan yang dikirimkan.
102	Seleksi kondisi jika validator tidak sesuai atau salah.
103	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
105	Batas seleksi kondisi IF.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
106	Inisialisasi kodelokasi_id yang ada pada variabel aset dengan fungsi auth berdasarkan id.
107	Inisialisasi variabel kodelokasi berdasarkan nilai masukan dari kodelokasi_id.
108	Inisialisasi variabel nomorregistrasi = objek Aset dengan parameter kodelokasi_id di count.
110	Inisialisasi variabel nores = variabel nomorregistrasi + 1.
111	Inisialisasi nomor_registrasi pada variabel aset dengan nilai '000' diikuti variabel dari nores.
113	Inisialisasi kode_barang pada variabel aset dengan masukan sesuai nilai dari kode_barang.
114	Inisialisasi nama_barang pada variabel aset dengan masukan sesuai nilai dari nama_barang.
115	Inisialisasi kategori_aset pada variabel aset dengan masukan sesuai nilai dari kategori_aset.
117	Inisialisasi kodelokasi_id pada variabel aset dengan masukan sesuai nilai dari kodelokasi_id.
119	Inisialisasi tipe_aset pada variabel aset dengan masukan sesuai nilai dari tipe_aset.
120	Inisialisasi luas pada variabel aset dengan masukan sesuai nilai dari luas.
121	Inisialisasi luas pada variabel aset dengan masukan sesuai nilai dari tahun_pengadaan.
122	Inisialisasi alamat pada variabel aset dengan masukan sesuai nilai dari alamat.
123	Inisialisasi status_hak_tanah pada variabel aset dengan masukan sesuai nilai dari status_hak_tanah.
125	Inisialisasi tgl_sertif_tanah pada variabel aset dengan masukan sesuai nilai dari tgl_sertif_tanah.
126	Inisialisasi nomor_sertif_tanah pada variabel aset dengan masukan sesuai nilai dari nomor_sertif_tanah.
128	Inisialisasi penggunaan pada variabel aset dengan masukan sesuai nilai dari penggunaan.
129	Inisialisasi harga pada variabel aset dengan masukan sesuai nilai dari harga.
130	Inisialisasi asal_usul pada variabel aset dengan masukan sesuai nilai dari asal_usul.
131	Inisialisasi keterangan pada variabel aset dengan masukan sesuai nilai dari keterangan.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
132	Inisialisasi pengguna_id pada variabel aset dengan masukan sesuai nilai dari sesi yang <i>login</i> .
134	Seleksi kondisi jika masukan kategori_aset = 2 maka <i>statement</i> ini yang akan dijalankan.
135	Inisialisasi variabel validator berfungsi untuk memvalidasi masukan yang dikirimkan.
138	Seleksi kondisi jika validator tidak sesuai atau salah.
139	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
141	Batas seleksi kondisi IF.
142	Inisialisasi kodelokasi_id yang ada pada variabel aset dengan fungsi auth berdasarkan id.
143	Inisialisasi variabel kodelokasi berdasarkan nilai masukan dari kodelokasi_id.
144	Inisialisasi variabel nomorregistrasi = objek Aset dengan parameter kodelokasi_id di count.
146	Inisialisasi variabel nores = variabel nomorregistrasi + 1.
147	Inisialisasi nomor_registrasi pada variabel aset dengan nilai '000' diikuti variabel dari nores.
150	Inisialisasi kode_barang pada variabel aset dengan masukan sesuai nilai dari kode_barang.
151	Inisialisasi nama_barang pada variabel aset dengan masukan sesuai nilai dari nama_barang.
152	Inisialisasi kategori_aset pada variabel aset dengan masukan sesuai nilai dari kategori_aset.
153	Inisialisasi kodelokasi_id pada variabel aset dengan masukan sesuai nilai dari kodelokasi_id.
154	Inisialisasi tipe_aset pada variabel aset dengan masukan sesuai nilai dari tipe_aset.
155	Inisialisasi merk_tipe pada variabel aset dengan masukan sesuai nilai dari merk_tipe.
156	Inisialisasi ukuran_cc pada variabel aset dengan masukan sesuai nilai dari ukuran_cc.
157	Inisialisasi bahan pada variabel aset dengan masukan sesuai nilai dari bahan.
158	Inisialisasi tahun_pembelian pada variabel aset dengan masukan sesuai nilai dari tahun_pembelian.
160	Inisialisasi nomor_pabrik pada variabel aset dengan masukan sesuai nilai dari nomor_pabrik.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
161	Inisialisasi nomor_mesin pada variabel aset dengan masukan sesuai nilai dari nomor_mesin.
162	Inisialisasi nomor_rangka pada variabel aset dengan masukan sesuai nilai dari nomor_rangka.
163	Inisialisasi nomor_polisi pada variabel aset dengan masukan sesuai nilai dari nomor_polisi.
164	Inisialisasi nomor_bpkb pada variabel aset dengan masukan sesuai nilai dari nomor_bpkb.
165	Inisialisasi jumlah pada variabel aset dengan masukan sesuai nilai dari jumlah.
166	Inisialisasi jumlah_tersedia pada variabel aset dengan masukan sesuai nilai dari jumlah_tersedia.
167	Inisialisasi asal_usul pada variabel aset dengan masukan sesuai nilai dari asal_usul.
168	Inisialisasi harga pada variabel aset dengan masukan sesuai nilai dari harga.
169	Inisialisasi keterangan pada variabel aset dengan masukan sesuai nilai dari keterangan.
170	Inisialisasi pengguna_id pada variabel aset dengan masukan sesuai nilai dari sesi pengguna yang login.
172	Seleksi kondisi jika masukan kategori_aset = 3 maka <i>statement</i> ini yang akan dijalankan.
173	Inisialisasi variabel validator berfungsi untuk memvalidasi masukan yang dikirimkan.
175	Seleksi kondisi jika validator tidak sesuai atau salah.
176	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
177	Batas seleksi kondisi IF.
178	Inisialisasi kodelokasi_id yang ada pada variabel aset dengan fungsi auth berdasarkan id.
179	Inisialisasi variabel kodelokasi berdasarkan nilai masukan dari kodelokasi_id.
180	Inisialisasi variabel nomorregistrasi = objek Aset dengan parameter kodelokasi_id di count.
182	Inisialisasi variabel nores = variabel nomorregistrasi + 1.
183	Inisialisasi nomor_registrasi pada variabel aset dengan nilai '000' diikuti variabel dari nores.
185	Inisialisasi kode_barang pada variabel aset dengan masukan sesuai nilai dari kode_barang.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
186	Inisialisasi nama_barang pada variabel aset dengan masukan sesuai nilai dari nama_barang.
187	Inisialisasi kategori_aset pada variabel aset dengan masukan sesuai nilai dari kategori_aset.
188	Inisialisasi kodelokasi_id pada variabel aset dengan masukan sesuai nilai dari kodelokasi_id.
189	Inisialisasi tipe_aset pada variabel aset dengan masukan sesuai nilai dari tipe_aset.
190	Inisialisasi kondisi_bangunan pada variabel aset dengan masukan sesuai nilai dari kondisi_bangunan.
192	Inisialisasi konstruksi_bertingkat pada variabel aset dengan masukan sesuai nilai dari konstruksi_bertingkat.
194	Inisialisasi konstruksi_beton pada variabel aset dengan masukan sesuai nilai dari konstruksi_beton.
196	Inisialisasi luas_lantai pada variabel aset dengan masukan sesuai nilai dari luas_lantai.
197	Inisialisasi alamat pada variabel aset dengan masukan sesuai nilai dari alamat.
198	Inisialisasi tgl_dokumen pada variabel aset dengan masukan sesuai nilai dari tgl_dokumen.
199	Inisialisasi nomor_dokumen pada variabel aset dengan masukan sesuai nilai dari nomor_dokumen.
200	Inisialisasi luas pada variabel aset dengan masukan sesuai nilai dari luas.
201	Inisialisasi status_hak_tanah pada variabel aset dengan masukan sesuai nilai dari status_hak_tanah.
203	Inisialisasi nomor_kode_tanah pada variabel aset dengan masukan sesuai nilai dari nomor_kode_tanah.
205	Inisialisasi asal_usul pada variabel aset dengan masukan sesuai nilai dari asal_usul.
206	Inisialisasi harga pada variabel aset dengan masukan sesuai nilai dari harga.
207	Inisialisasi keterangan pada variabel aset dengan masukan sesuai nilai dari keterangan.
208	Inisialisasi pengguna_id pada variabel aset dengan masukan sesuai nilai dari sesi yang login.
210	Seleksi kondisi jika masukan kategori_aset = 4 maka <i>statement</i> ini yang akan dijalankan.
211	Inisialisasi variabel validator berfungsi untuk memvalidasi masukan yang dikirimkan.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
214	Seleksi kondisi jika validator tidak sesuai atau salah.
215	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
216	Batas seleksi kondisi IF.
217	Inisialisasi kodelokasi_id yang ada pada variabel aset dengan fungsi auth berdasarkan id.
218	Inisialisasi variabel kodelokasi berdasarkan nilai masukan dari kodelokasi_id.
219	Inisialisasi variabel nomorregistrasi = objek Aset dengan parameter kodelokasi_id di count.
221	Inisialisasi variabel nores = variabel nomorregistrasi + 1.
222	Inisialisasi nomor_registrasi pada variabel aset dengan nilai '000' diikuti variabel dari nores.
223	Inisialisasi kode_barang pada variabel aset dengan masukan sesuai nilai dari kode_barang.
224	Inisialisasi nama_barang pada variabel aset dengan masukan sesuai nilai dari nama_barang.
225	Inisialisasi kategori_aset pada variabel aset dengan masukan sesuai nilai dari kategori_aset.
226	Inisialisasi kodelokasi_id pada variabel aset dengan masukan sesuai nilai dari kodelokasi_id.
227	Inisialisasi tipe_aset pada variabel aset dengan masukan sesuai nilai dari tipe_aset.
228	Inisialisasi konstruksi_beton pada variabel aset dengan masukan sesuai nilai dari konstruksi_beton.
230	Inisialisasi panjang pada variabel aset dengan masukan sesuai nilai dari panjang.
231	Inisialisasi lebar pada variabel aset dengan masukan sesuai nilai dari lebar.
232	Inisialisasi luas pada variabel aset dengan masukan sesuai nilai dari luas.
233	Inisialisasi alamat pada variabel aset dengan masukan sesuai nilai dari alamat.
234	Inisialisasi tgl_dokumen pada variabel aset dengan masukan sesuai nilai dari tgl_dokumen.
235	Inisialisasi nomor_dokumen pada variabel aset dengan masukan sesuai nilai dari nomor_dokumen.
236	Inisialisasi status_hak_tanah pada variabel aset dengan masukan sesuai nilai dari status_hak_tanah.
238	Inisialisasi nomor_kode_tanah pada variabel aset dengan masukan sesuai nilai dari nomor_kode_tanah.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
240	Inisialisasi asal_usul pada variabel aset dengan masukan sesuai nilai dari asal_usul.
241	Inisialisasi harga pada variabel aset dengan masukan sesuai nilai dari harga.
242	Inisialisasi kondisi_bangunan pada variabel aset dengan masukan sesuai nilai dari kondisi_bangunan.
244	Inisialisasi keterangan pada variabel aset dengan masukan sesuai nilai dari keterangan.
245	Inisialisasi pengguna_id pada variabel aset dengan masukan sesuai nilai dari sesi pengguna yang login.
247	Seleksi kondisi else
248	Inisialisasi variabel validator berfungsi untuk memvalidasi masukan yang dikirimkan.
251	Seleksi kondisi jika validator tidak sesuai atau salah.
252	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
254	Batas seleksi kondisi IF.
255	Inisialisasi kodelokasi_id pada variable request == NULL
256	Inisialisasi kategori berdasarkan nilai masukan dari kategori_aset
257	Inisialisasi variabel kodekategori = objek Aset dengan parameter kategori_aset di count.
259	Inisialisasi variabel nores = variabel nomorregistrasi + 1.
260	Inisialisasi nomor_registrasi pada variabel aset dengan nilai '000' diikuti variabel dari nores.
262	Inisialisasi kode_barang pada variabel aset dengan masukan sesuai nilai dari kode_barang.
263	Inisialisasi nama_barang pada variabel aset dengan masukan sesuai nilai dari nama_barang.
264	Inisialisasi kategori_aset pada variabel aset dengan masukan sesuai nilai dari kategori_aset.
265	Inisialisasi tipe_aset pada variabel aset dengan masukan sesuai nilai dari tipe_aset.
266	Inisialisasi merk_tipe pada variabel aset dengan masukan sesuai nilai dari merk_tipe.
267	Inisialisasi ukuran_cc pada variabel aset dengan masukan sesuai nilai dari ukuran_cc.
268	Inisialisasi bahan pada variabel aset dengan masukan sesuai nilai dari bahan.

Tabel 5.10 Penjelasan Kode Program Fungsi *create()* (Lanjutan)

Penjelasan Fungsi <i>create()</i>	
269	Inisialisasi tahun_pembelian pada variabel aset dengan masukan sesuai nilai dari tahun_pembelian.
271	Inisialisasi jumlah_tersedia pada variabel aset dengan masukan sesuai nilai dari jumlah_tersedia.
272	Inisialisasi jumlah pada variabel aset dengan masukan sesuai nilai dari jumlah.
273	Inisialisasi asal_usul pada variabel aset dengan masukan sesuai nilai dari asal_usul.
274	Inisialisasi harga pada variabel aset dengan masukan sesuai nilai dari harga.
275	Inisialisasi keterangan pada variabel aset dengan masukan sesuai nilai dari keterangan.
276	Inisialisasi pengguna_id pada variabel aset dengan masukan sesuai nilai dari sesi pengguna yang sedang <i>login</i> .
277	Batas akhir seleksi IF
278	Inisialisasi fungsi save pada variabel aset.
279	Mengembalikan ke halaman aset dinas.
281	Batas akhir fungsi <i>create</i> .

5.3.1.2 Implementasi Perancangan Komponen Permintaan Laporan Aset Dengan Fungsi *requestReportKadis*

Pada tabel 5.11 merupakan alur logika dari fungsi *requestReportKadis*. Pertama melakukan validasi terhadap masukan yang dimasukkan oleh kepala dinas ketika tidak sesuai melihatkan pesan *error* bahwa ada *field form* kosong. Jika validator sudah sesuai maka akan menyimpan permintaan laporan ke *database* dan mengirimkan notifikasi *email* ke pengurus barang.

Nama *class* : AsetController

Nama fungsi : *requestReportKadis()*

Tabel 5.11 Kode Program Fungsi *requestReportKadis()*

Kode Program Fungsi <i>requestReportKadis()</i>	
1	public function requestReportKadis(Request \$request)
2	{
3	\$validator = Validator::make(\$request->all(), [
4	'nama_permintaan' => "required max:32",
5	'catatan' => 'max:255',
6	]);
7	
8	if (\$validator->fails()) {
9	return back()->withErrors(\$validator)->withInput();
10	}
11	}

Tabel 5.11 Kode Program Fungsi *requestReportKadis()* (Lanjutan)

Kode Program Fungsi <i>requestReportKadis()</i>	
12	<code>PermintaanLaporan::create(\$request->all());</code>
13	<code>if (\$request->nama_permintaan==1) {</code>
14	<code> \$month = date('m');</code>
15	<code> if(\$month <=6){</code>
16	<code> \$semester = "Semester 1 dari bulan Januari s/d Juni";</code>
17	<code> }else{</code>
18	<code> \$semester = "Semester 2 dari bulan Juli s/d</code>
19	<code>Desember";</code>
20	<code> }</code>
21	<code> \$now = new DateTime();</code>
22	<code> \$date = \$now->format('d-m-Y');</code>
23	<code> \$catatan = Input::get('catatan');</code>
24	<code> \$permintaanlaporan</code>
25	<code>PermintaanLaporan::where('nama_permintaan',1)->get();</code>
26	<code> \$permintaanlaporan1</code>
27	<code>PermintaanLaporan::where('nama_permintaan',1)->first();</code>
28	<code> \$pengguna = Pengguna::where('jabatan',7)->first();</code>
29	<code> \$pengguna1 = Pengguna::where('jabatan',2)->first();</code>
30	<code> \$pengguna2 = Pengguna::where('jabatan',5)->first();</code>
31	<code> \$pengguna3 = Pengguna::where('jabatan',3)->first();</code>
32	<code> \$to = \$pengguna->email;</code>
33	<code> \$cc = ([\$pengguna2->email, \$pengguna3->email]);</code>
34	<code> \$from = \$pengguna1->email;</code>
35	<code> \$subject = 'Pembuatan Laporan Aset';</code>
36	<code> Mail::send('Email.PermintaanLaporan',</code>
37	<code> compact('pengguna', 'pengguna1', 'pengguna2', 'pengguna3', 'per</code>
38	<code> mintaanlaporan', 'permintaanlaporan1', 'semester', 'date', 'cat</code>
39	<code> atan'), function (\$message) use (\$to,\$from,\$subject) {</code>
40	<code> \$message->from(\$from)</code>
41	<code> ->to(\$to)</code>
42	<code> ->subject(\$subject);</code>
43	<code> });</code>
44	<code> }elseif (\$request->nama_permintaan==2) {</code>
45	<code> \$month = date('m');</code>
46	<code> if(\$month <=6){</code>
47	<code> \$semester = "Semester 1 dari bulan Januari s/d Juni";</code>
48	<code> }else {</code>
49	<code> \$semester = "Semester 2 dari bulan Juli s/d</code>
50	<code>Desember";</code>
51	<code> }</code>
52	<code> \$now = new DateTime();</code>
53	<code> \$date = \$now->format('d-m-Y');</code>
54	<code> \$catatan = Input::get('catatan');</code>
55	<code> \$permintaanlaporan</code>
56	<code>PermintaanLaporan::where('nama_permintaan',2)->get();</code>
57	<code> \$permintaanlaporan1</code>
58	<code>PermintaanLaporan::where('nama_permintaan',2)->first();</code>
59	<code> \$pengguna = Pengguna::where('jabatan',8)->first();</code>
60	<code> \$pengguna1 = Pengguna::where('jabatan',2)->first();</code>
61	<code> \$pengguna2 = Pengguna::where('jabatan',4)->first();</code>
62	<code> \$pengguna3 = Pengguna::where('jabatan',6)->first();</code>
63	<code> \$to = \$pengguna->email;</code>
64	<code> \$cc = ([\$pengguna2->email, \$pengguna3->email]);</code>
65	<code> \$from = \$pengguna1->email;</code>
66	<code> \$subject = 'Pembuatan Laporan Persediaan';</code>

Tabel 5.11 Kode Program Fungsi *requestReportKadis()* (Lanjutan)

Kode Program Fungsi <i>requestReportKadis()</i>	
67	Mail::send('Email.PermintaanLaporan',
68	compact('pengguna','pengguna','permintaanlaporan','permint
69	aanlaporan1','semester','date','catatan'), function
70	(\$message) use (\$to,\$cc,\$from,\$subject) {
71	\$message->from(\$from)
72	->to(\$to)
73	->cc(\$cc)
74	->subject(\$subject);
75	});
76	}
77	return redirect()->route('kadis-request-report')-
78	>with(['success' => 'Permintaan laporan berhasil dikirim!']);
79	}

Tabel 5.12 Penjelasan Kode Program Fungsi *requestReportKadis()*

Penjelasan Fungsi <i>requestReportKadis()</i>	
1	Membuat fungsi <i>requestReportKadis</i> dengan masukan dengan parameter <i>request</i> .
3	Inisialisasi variabel validator berfungsi untuk memvalidasi masukan yang dikirimkan.
8	Seleksi kondisi jika validator tidak sesuai atau salah.
9	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
10	Batas seleksi kondisi IF.
12	Menyimpan ke objek PermintaanLaporan dari semua masukan.
13	Seleksi kondisi jika nama_permintaan = 1 maka <i>statement</i> ini yang akan dijalankan.
14	Inisialisasi variabel month sama dengan bulan saat ini.
15	Seleksi kondisi if dengan parameter dari variabel month kurang dari 6.
16	Inisialisasi variabel semester = "Semester 1 dari bulan Januari s/d Juni".
17	Jika tidak memenuhi masuk ke seleksi else.
18	Inisialisasi variabel semester = "Semster 2 dari bulan Juli s/d Desember".
20	Batas akhir seleksi kondisi if.
21	Inisialisasi variabel now dari fungsi DateTime sekarang.
22	Inisialisasi variabel date = variabel now dengan format hari, bulan, tahun.
23	Inisialisasi variabel catatan berdasarkan masukan yang dikirimkan.
24	Inisialisasi variabel permintaanlaporan dari objek PermintaanLaporan berdasarkan nama_permintaan = 1.
26	Inisialisasi variabel permintaanlaporan1 dari objek PermintaanLaporan dengan parameter nama_permintaan = 1.

Tabel 5.12 Penjelasan Kode Program Fungsi *requestReportKadis()* (Lanjutan)

Penjelasan Fungsi <i>requestReportKadis()</i>	
28	Inisialisasi variabel pengguna dari objek Pengguna dengan parameter jabatan = 7.
29	Inisialisasi variabel pengguna1 dari objek Pengguna dengan parameter jabatan = 2.
30	Inisialisasi variabel pengguna2 dari objek Pengguna dengan parameter jabatan = 5.
31	Inisialisasi variabel pengguna3 dari objek Pengguna dengan parameter jabatan = 3.
32	Inisialisasi variabel to berisikan email dari variabel pengguna.
33	Inisialisasi variabel cc dengan array berisikan email dari variabel pengguna2 dan pengguna3.
34	Inisialisasi variabel form berisikan email dari variabel pengguna1.
35	Inisialisasi variabel subject = "Pembuatan Laporan Aset".
36-43	Inisialisasi objek mail yang berfungsi untuk mengirimkan email.
44	Seleksi kondisi else if dengan parameter nama_permintaan = 2 maka statement ini yang akan dijalankan.
45	Inisialisasi variabel month sama dengan bulan saat ini.
46	Seleksi kondisi if dengan parameter dari variabel month kurang dari 6.
47	Inisialisasi variabel semester = "Semester 1 dari bulan Januari s/d Juni".
48	Jika tidak memenuhi masuk ke seleksi else.
50	Inisialisasi variabel semester = "Semster 2 dari bulan Juli s/d Desember".
51	Batas akhir seleksi kondisi if.
52	Inisialisasi variabel now dari fungsi DateTime sekarang.
53	Inisialisasi variabel date = variabel now dengan format hari, bulan, tahun.
54	Inisialisasi variabel catatan berdasarkan masukan yang dikirimkan.
55	Inisialisasi variabel permintaanlaporan dari objek PermintaanLaporan berdasarkan nama_permintaan = 2.
57	Inisialisasi variabel permintaanlaporan1 dari objek PermintaanLaporan dengan parameter nama_permintaan = 2.
59	Inisialisasi variabel pengguna dari objek Pengguna dengan parameter jabatan = 8.
60	Inisialisasi variabel pengguna1 dari objek Pengguna dengan parameter jabatan = 2.
61	Inisialisasi variabel pengguna2 dari objek Pengguna dengan parameter jabatan = 4.

Tabel 5.12 Penjelasan Kode Program Fungsi *requestReportKadis()* (Lanjutan)

Penjelasan Fungsi <i>requestReportKadis()</i>	
62	Inisialisasi variabel <i>pengguna3</i> dari objek <i>Pengguna</i> dengan parameter jabatan = 6.
63	Inisialisasi variabel <i>to</i> berisikan email dari variabel <i>pengguna</i> .
64	Inisialisasi variabel <i>cc</i> dengan array berisikan email dari variabel <i>pengguna2</i> dan <i>pengguna3</i> .
65	Inisialisasi variabel <i>form</i> berisikan email dari variabel <i>pengguna1</i> .
66	Inisialisasi variabel <i>subject</i> = "Pembuatan Laporan Aset".
67-75	Inisialisasi objek mail yang berfungsi untuk mengirimkan email.
76	Batas akhir dari seleksi kondisi <i>if</i> .
77	Mengembalikan ke halaman permintaan laporan.
79	Batas akhir dari fungsi.

5.3.1.3 Implementasi Perancangan Komponen Tambah Pemegang Kendaraan Dengan Fungsi *create*

Pada Tabel 5.13 merupakan alur logika dari fungsi *create*. Pertama melakukan validasi terhadap masukan yang dimasukkan oleh pengurus barang ketika tidak sesuai melihatkan pesan *error* bahwa ada *field form* kosong. Jika validator sudah sesuai maka akan menyimpan data pemegang kendaraan berdasarkan *request* id ke *database* dengan mengurangi pada *field* jumlah_tersedia sesuai dengan aset yang diambil pada *table* aset dan menambahkan pada *field* jumlah_kendaraan pada *table* pengguna sesuai dengan pengguna yang dipilih untuk memegang kendaraan.

Nama *class* : PemegangKendaraanController

Nama fungsi : *create()*

Tabel 5.13 Kode Program Fungsi *create*

Kode Program Fungsi <i>create</i>	
1	<code>public function create(Request \$request)</code>
2	<code>{</code>
3	<code> \$validator = Validator::make(\$request->all(), [</code>
4	<code> 'pengguna_id' => 'required max:100',</code>
5	<code> 'aset_id' => 'required max:100',</code>
6	<code>]);</code>
7	<code></code>
8	<code> if (\$validator->fails()) {</code>
9	<code> return back()->withErrors(\$validator)->withInput();</code>
10	<code> }</code>
11	<code> Aset::find(\$request->aset_id)-</code>
12	<code> ->decrement('jumlah_tersedia',1);</code>



Tabel 5.13 Kode Program Fungsi *create* (Lanjutan)

Kode Program Fungsi <i>create</i>	
13	Pengguna::find(\$request->pengguna_id)-
14	>increment('jumlah_kendaraan', 1);
15	
16	// Store the blog post...
17	PemegangKendaraan::create(\$request->all());
18	
19	return redirect()->route('vehicle')->with(['success'
20	=>'Aset berhasil ditambah!']);
21	}

Tabel 5.14 Penjelasan Kode Program Fungsi *create()*

Penjelasan Fungsi <i>create()</i>	
1	Membuat fungsi <i>create</i> dengan masukan dengan parameter <i>request</i> .
3	Inisialisasi variabel validator yang berfungsi untuk memvalidasi masukan yang dikirimkan.
8	Seleksi kondisi jika validator tidak sesuai atau salah.
9	Mengembalikan pesan <i>error</i> berdasarkan sesuai dengan validator yang kosong.
10	Batas seleksi kondisi IF.
11	Inisialisasi objek Aset dengan parameter aset_id dari masukan untuk melakukan increment pada jumlah_tersedia = 1.
13	Inisialisasi objek Pengguna dengan parameter pengguna_id dari masukan untuk melakukan decrement pada jumlah_kendaraan = 1.
17	Menyimpan semua data yang dikirimkan dengan menggunakan fungsi <i>create</i> ke objek PemegangKendaraan.
19	Mengembalikan ke halaman tambah pemegang kendaraan.
20	Batas akhir dari fungsi.

5.3.2 Implementasi *Database*

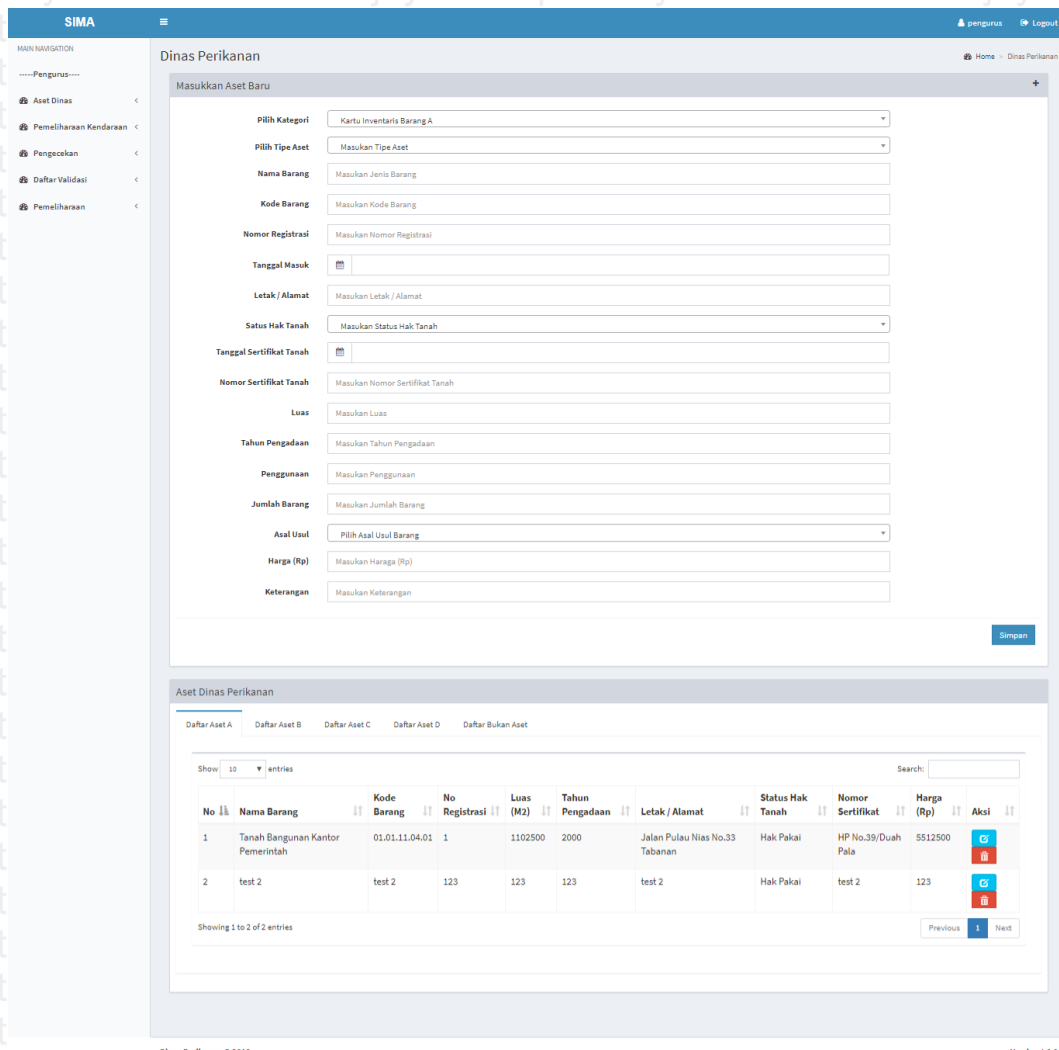
Sistem informasi manajemen aset dibangun meningkatkan efisiensi dan data aset menjadi terpusat. Pada gambar 5.9 *physical data model* dari SIMA yang dibangun.

5.3.3 Implementasi Antarmuka

5.3.3.1 Implementasi Antarmuka Tambah Aset

Pada Gambar 5.10 merupakan implementasi antarmuka tambah aset. Berisikan *combo box* untuk memilih kategori, status hak tanah dan asal usul. Terdapat *text field* untuk memasukkan data aset serta tabel yang berisikan daftar aset yang

dibedakan berdasarkan KIB dan terdapat tombol sunting dan hapus aset serta tombol untuk menyimpan data aset.



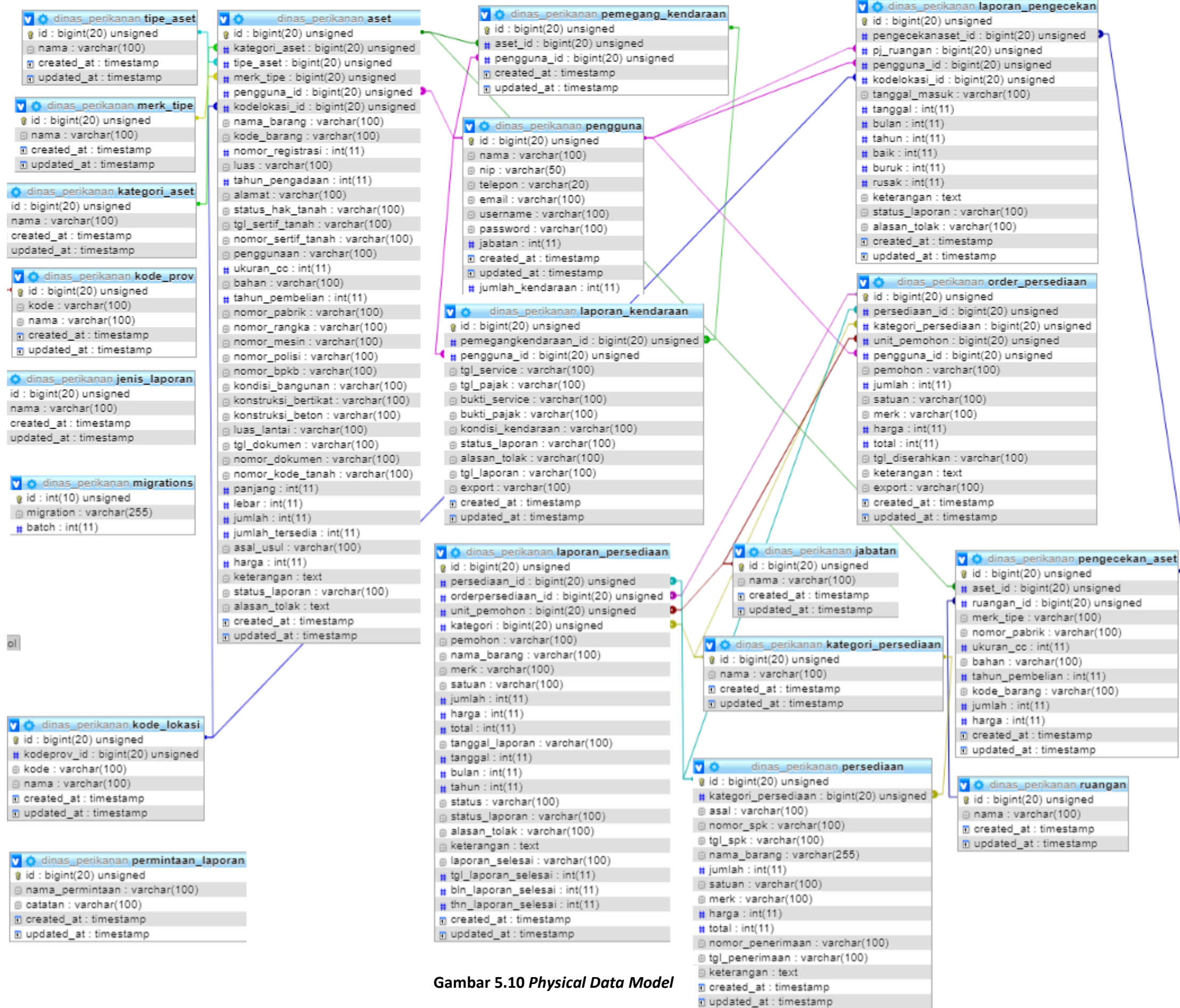
Gambar 5.9 Implementasi Antarmuka Tambah Aset

5.3.3.2 Implementasi Antarmuka Permintaan Laporan Aset

Pada Gambar 5.11 merupakan implementasi antarmuka permintaan pembuatan laporan aset. Berisikan *combo box* untuk memilih laporan yang akan dibuat dan *textarea* untuk tempat memasukkan catatan serta tombol untuk mengirimkan permintaan pembuatan laporan aset.

5.3.3.3 Implementasi Antarmuka Tambah Pemegang Kendaraan

Pada Gambar 5.12 merupakan implementasi antarmuka tambah pemegang kendaraan. Berisikan *combo box* untuk memilih nama penanggung jawab pemegang kendaraan dan nomor polisi dari kendaraan yang akan diberikan serta tombol untuk menyimpan data pemegang kendaraan. Terdapat tabel yang berisikan daftar dari nama pengguna yang memegang kendaraan.



Gambar 5.10 Physical Data Model

SIMA Home Kadis Logout

MAIN NAVIGATION

- Permintaan Laporan
- Pemeliharaan

Dinas Perikanan

Permintaan Pembuatan Laporan

Masukan Pilihan:

Tambahkan Catatan:

Request Laporan

Dinas Perikanan © 2018 Version 1.0.0

Gambar 5.11 Implementasi Antarmuka Permintaan Pembuatan Laporan Aset

SIMA Home Pengurus Logout

MAIN NAVIGATION

- Aset Dinas
- Pemeliharaan Kendaraan
- Pengecekan
- Daftar Validasi
- Pemeliharaan

Dinas Perikanan

Masukan Data Pemegang

Pilih Nama Pemegang:

Pilih Nomor Polisi:

Simpan

Daftar Kendaraan

Show: 10 entries Search:

No	Nama Pemegang	Jumlah Kendaraan	Aksi
1	Sekretaris	1	Edit
2	Kadis	1	Edit
3	Jerl permana	1	Edit

Showing 1 to 3 of 3 entries Previous 1 Next

Dinas Perikanan © 2018 Version 1.0.0

Gambar 5.12 Implementasi Antarmuka Tambah Pemegang Kendaraan

BAB 6 PENGUJIAN DAN ANALISIS

Pengujian dilaksanakan ketika semua tahapan implementasi selesai yang bertujuan untuk menemukan kesalahan pada sistem. Pengujian dilakukan dengan dua metode yaitu dengan metode *black box testing* dan *white box testing*. *Black box testing* merupakan pengujian yang difokuskan pada persyaratan fungsional atau kesesuaian hasil dari masukan atau keluaran. Untuk pengujian *white box* menggunakan *unit testing*. *Unit testing* dilaksanakan dengan metode *basis path testing*, yaitu membuat *independent path* yang mengacu pada rancangan algoritme berupa *pseudocode* yang diubah menjadi *flowgraph* terlebih dahulu. Pada pengujian non fungsional menggunakan pengujian *compatibility* yang digunakan untuk menguji kemampuan aplikasi yang sudah dibuat dapat berjalan pada *platform* yang berbeda.

6.1 Pengujian Unit

Dalam melakukan *unit testing* menggunakan metode *basis path testing*. Metode ini meliputi pemodelan algoritme dalam suatu *flow graph*, menghitung *cyclomatic complexity*, menentukan *independent path*, dan memberi kasus uji setiap basis set yang ditentukan berdasarkan *independent path*.

6.1.1 Pengujian Fungsi *create* pada *Class AsetController*

Pengujian unit fungsi *create* digunakan untuk memasukkan aset ke dalam sistem berdasarkan kategori aset.

Nama Class : AsetController
Nama Fungsi : *create*
Algoritme :

Tabel 6.1 Pseudocode Fungsi *create()*

Pseudocode Fungsi <i>create()</i>	
1	Mulai
2	aseta <- nilai aset a asetb <- nilai aset b asetc <- nilai aset c asetd <- nilai aset d asete <- nilai aset e
3	aset <- objek Aset baru
4	IF (kategori_aset==1)
5	validator <- memvalidasi input dari variabel aseta
6	IF (validator->fails() ==TRUE)
7	Return pesan error ENDIF
8	aset->kodelokasi_id = Auth()->id()

**Tabel 6.1 Pseudocode Fungsi create() (Lanjutan)**

Pseudocode Fungsi create()	
	<pre> kodelokasi <- nilai input kodelokasi_id dalam objek asset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomerregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset <- nilai dari input tipe_aset luas <- nilai dari input luas tahun_pengadaan <- nilai dari input tahun_pengadaan alamat <- nilai dari input alamat status_hak_tanah <- nilai dari input status_hak_tanah tgl_sertif_tanah <- nilai dari input tgl_sertif_tanah nomor_sertif_tanah <- nilai dari input nomor_sertif_tanah penggunaan <- nilai dari input penggunaan harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna </pre>
9	ELSEIF (kategori_aset==2)
10	validator <- memvalidasi input dari variabel asetb
11	IF (validator->fails() ==TRUE)
12	Return pesan error ENDIF
13	<pre> aset->kodelokasi_id = Auth()->id() kodelokasi <- nilai input kodelokasi_id dalam objek asset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomerregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset </pre>

Tabel 6.1 Pseudocode Fungsi create() (Lanjutan)

Pseudocode Fungsi create()	
	kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset <- nilai dari input tipe_aset merk_tipe <- nilai dari input merk_tipe ukuran_cc <- nilai dari input ukuran_cc bahan <- nilai dari input bahan tahun_pembelian <- nilai dari input tahun_pembelian nomor_pabrik <- nilai dari input nomor_pabrik nomor_mesin = nilai dari input nomor_mesin nomor_rangka = nilai dari input nomor_rangka nomor_polisi = nilai dari input nomor_polisi nomor_bpkb <- nilai dari input nomor_bpkb jumlah <- nilai dari input jumlah jumlah_tersedia <- nilai dari input jumlah harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
14	Elseif (kategori_aset==3)
15	validator <- memvalidasi input dari variabel asetc
16	IF (validator->fails() ==TRUE)
17	Return pesan error ENDIF
18	aset->kodelokasi_id = Auth()->id() kodelokasi <- nilai input kodelokasi_id dalam objek aset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomerregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset = nilai dari input variabel tipe_aset kondisi_bangunan <- nilai dari input kondisi_bangunan konstruksi_bertikat <- nilai dari input konstruksi_bertikat konstruksi_beton <- nilai dari input konstruksi_beton;

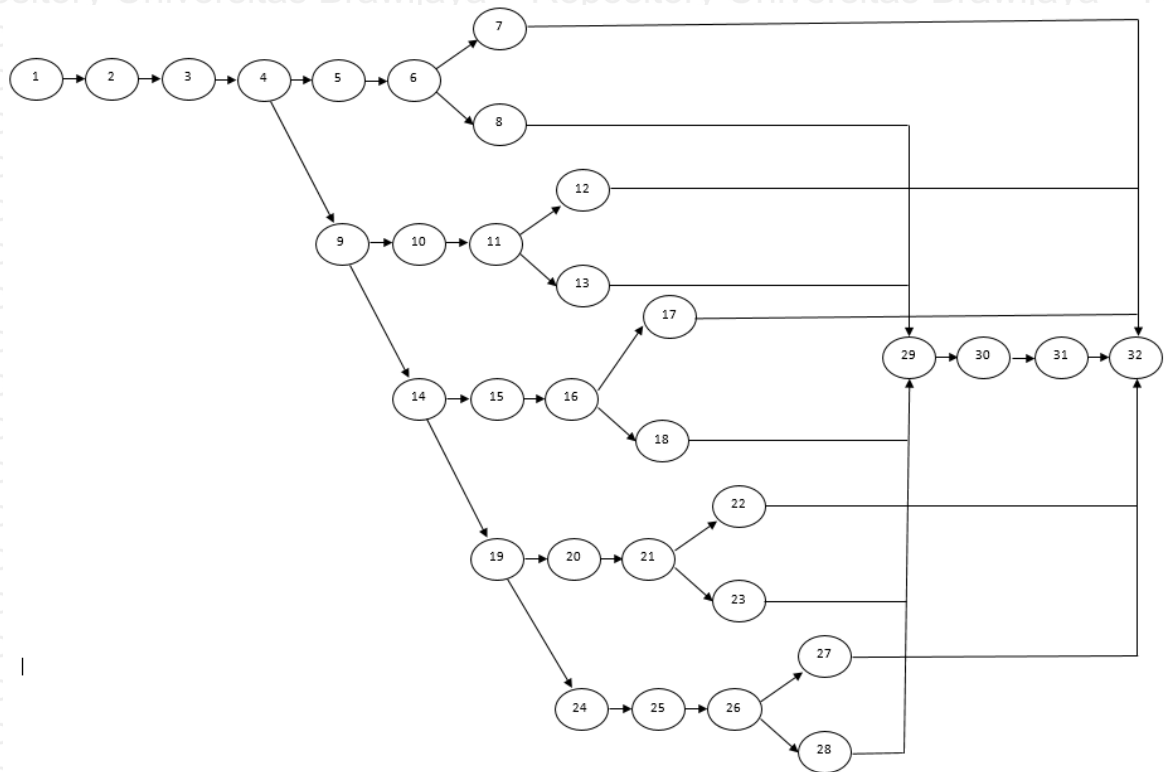
Tabel 6.1 Pseudocode Fungsi create() (Lanjutan)

Pseudocode Fungsi create()	
	luas_lantai <- nilai dari input luas_lantai alamat <- nilai dari input alamat tgl_dokumen <- nilai dari input tgl_dokumen nomor_dokumen <- nilai dari input nomor_dokumen luas <- nilai dari input luas; status_hak_tanah <- nilai dari input status_hak_tanah nomor_kode_tanah <- nilai dari input nomor_kode_tanah harga <- nilai dari input harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
19	IF (kategori_aset==4)
20	validator <- memvalidasi input dari variabel asetd
21	IF (validator->fails() ==TRUE)
22	Return pesan error ENDIF
23	aset->kodelokasi_id = Auth()->id() kodelokasi <- nilai input kodelokasi_id dalam objek aset nomorregistrasi <- menghitung jumlah kodelokasi_id pada objek Aset nores = nomerregistrasi + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset <- nilai dari input kategori_aset kodelokasi_id <- nilai dari input kodelokasi_id tipe_aset = nilai dari input variabel tipe_aset konstruksi_beton <- nilai dari input konstruksi_beton panjang <- nilai dari input panjang lebar <- nilai dari input lebar luas <- nilai dari input luas alamat <- nilai dari input alamat tgl_dokumen <- nilai dari input tgl_dokumen nomor_dokumen <- nilai dari input nomor_dokumen status_hak_tanah <- nilai dari input status_hak_tanah nomor_kode_tanah <- nilai dari input nomor_kode_tanah harga <- nilai dari input harga

Tabel 6.1 Pseudocode Fungsi create() (Lanjutan)

Pseudocode Fungsi create()	
	asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- mengambil id dari sesi pengguna
24	ELSE
25	validator <- memvalidasi input dari variabel asete
26	IF (validator->fails() ==TRUE)
27	Return pesan error ENDIF
28	kodelokasi_id <- inputan kodelokasi_id = NULL kategori <- nilai input dari kategori_aset kodekategori <- menghitung kategori_aset pada objek Aset nores = variabel kodekategori + 1 nomor_registrasi = '000'.nores kode_barang <- nilai dari input kode_barang nama_barang <- nilai dari input nama_barang kategori_aset < nilai dari input kategori_aset tipe_aset <- nilai dari input tipe_aset merk_tipe <- nilai dari input merk_tipe ukuran_cc <- nilai dari input ukuran_cc bahan <- nilai dari input bahan tahun_pembelian <- nilai dari input variabel tahun_pembelian jumlah <- nilai dari input jumlah; jumlah_tersedia <- nilai dari input jumlah harga <- nilai dari input variabel harga asal_usul <- nilai dari input asal_usul keterangan <- nilai dari input keterangan pengguna_id <- id dari sesi pengguna
29	ENDIF
30	save() <- menyimpan data dari objek asset
31	Redirect ke halaman aset dinas dan menampilkan pesan sukses
32	Selesai

Berdasarkan *pseudocode* pada Tabel 6.1 didapatkan *basis path testing*. Pada Gambar 6.1 merupakan *flow graph* dari terjemahan *pseudocode*.



Gambar 6.1 Flow Graph Fungsi create()

Untuk mendapatkan perhitungan *cyclomatic complexity* dengan menggunakan rumus:

$$V(G) = \text{total Region} = 10$$

$$V(G) = \text{total Edge} - \text{total Node} + 2 = 40 - 32 + 2 = 10$$

$$V(G) = \text{Predicate Node} + 1 = 9 + 1 = 10$$

Dari hasil perhitungan *cyclomatic complexity*, didapatkan lima jalur basis path, yaitu:

1. 1-2-3-4-5-6-7-32
2. 1-2-3-4-5-6-8-29-30-31-32
3. 1-2-3-4-9-10-11-12-32
4. 1-2-3-4-9-10-11-13-29-30-31-32
5. 1-2-3-4-9-14-15-16-17-32
6. 1-2-3-4-9-14-15-16-18-29-30-31-32
7. 1-2-3-4-9-14-19-20-21-22-32
8. 1-2-3-4-9-14-19-20-21-23-29-30-31-32
9. 1-2-3-4-9-14-19-24-25-26-27-32
10. 1-2-3-4-9-14-19-24-25-26-28-29-30-31-32

Tabel 6.2 Hasil Pengujian Unit Fungsi *create()*

No	Data Uji	Hasil Yang diharapkan	Hasil	Valid / Tidak Valid
1	Jika kategori_aset=1, Kodelokasi_id = NULL, nama_barang=NULL, kode_barang=NULL, nomor_registrasi, =NULL tipe_aset =NULL, luas =NULL, tahun_pengadaan =NULL, alamat=NULL, status_hak_tanah =NULL, tgl_sertif_tanah =NULL, nomor_sertif_tanah =NULL, penggunaan =NULL, jumlah =NULL, asal_usul =NULL, harga=NULL	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid
2	Jika kategori_aset=1, Kodelokasi_id = 1, nama_barang=tanah, kode_barang=01.01, nomor_registrasi =001, tipe_aset =7, luas =1100, tahun_pengadaan =2000, alamat=tabanan, status_hak_tanah =hak pakai, tgl_sertif_tanah =03/01/2000, nomor_sertif_tanah =mhkn14, penggunaan =bangunan kantor, jumlah =1, asal_usul =pembelian, harga=11000	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Valid

Tabel 6.2 Hasil Pengujian Unit Fungsi *create()* (Lanjutan)

No	Data Uji	Hasil Yang diharapkan	Hasil	Valid / Tidak Valid
3	Jika kategori_aset=2, Kodelokasi_id = NULL, nama_barang = NULL kode_barang = NULL, nomor_registrasi = NULL, tanggal_masuk = NULL, tipe_aset = NULL, merk_tipe = NULL, bahan = NULL, tahun_pembelian = NULL, jumlah = NULL, asal_usul = NULL, harga = NULL	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid
4	Jika kategori_aset=2, Kodelokasi_id = 1, nama_barang = Vega Zr kode_barang = 01.02.01, nomor_registrasi = 0001, tipe_aset = 1, merk_tipe = 1, bahan = campuran, tahun_pembelian = 2000, jumlah = 1, asal_usul = pembelian, harga = 110000	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Valid
5	Jika kategori_aset = 3, kodelokasi_id = NULL, nama_barang = NULL, kode_barang = NULL, nomor_registrasi = NULL, tipe_aset = NULL, konstruksi_beton = NULL, panjang = NULL, lebar = NULL, luas = NULL, alamat = NULL, tgl_dokumen = NULL,	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid

Tabel 6.2 Hasil Pengujian Unit Fungsi *create()* (Lanjutan)

No	Data Uji	Hasil Yang diharapkan	Hasil	Valid / Tidak Valid
5	nomor_dokumen = NULL, status_hak_tanah = NULL, jumlah = NULL, asal_usul = NULL, harga = NULL	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid
6	Jika kategori_aset = 3, kodelokasi_id = 1, nama_barang = kantor, kode_barang = 0.04.01, nomor_registrasi = 0001, tipe_aset = 1, konstruksi_beton = betom, panjang = 10, lebar = 10, luas = 10, alamat = tabanan, tgl_dokumen = 01/03/2000, nomor_dokumen = mkhn123, status_hak_tanah = hak pakai, jumlah = 1, asal_usul = pembelian, harga = 100000	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Valid
7	Jika kategori_aset = 4, kodelokasi_id = NULL, nama_barang = NULL, kode_barang = NULL, nomor_registrasi = NULL, tipe_aset = NULL, konstruksi_beton = NULL, panjang = NULL, lebar = NULL, luas = NULL, alamat = NULL, tgl_dokumen = NULL,	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid

Tabel 6.2 Hasil Pengujian Unit Fungsi *create()* (Lanjutan)

No	Data Uji	Hasil Yang diharapkan	Hasil	Valid / Tidak Valid
7	nomor_dokumen= NULL, status_hak_tanah= NULL, jumlah= NULL, asal_usul= NULL, harga= NULL	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid
8	Jika kategori_aset = 4, kodelokasi_id = 1, nama_barang = jalan beton, kode_barang = 0.01.04, nomor_registrasi = 0001, tipe_aset = 1, konstruksi_beton = beton, panjang = 10, lebar = 10, luas= 10, alamat= tabanan, tgl_dokumen= 03/01/2000, nomor_dokumen= mhkas, status_hak_tanah= hak pakai, jumlah= 1, asal_usul= pembelian, harga= 110000	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Valid
9	Jika kategori_aset 5, nama_barang = NULL, kode_barang = NULL, nomor_registrasi = NULL, bahan= NULL, tahun_pembelian= NULL, jumlah= NULL, asal_usul= NULL, harga= NULL	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Sistem akan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> pada <i>form</i> yang kosong.	Valid

**Tabel 6.3 Pseudocode Fungsi requestReportKadis() (Lanjutan)**

No	Data Uji	Hasil Yang diharapkan	Hasil	Valid / Tidak Valid
10	Jika kategori_aset 5, nama_barang = Pigura burung garuda, kode_barang = 07.01.01, nomor_registrasi = 001, bahan= campuran, tahun_pembelian= 02/03/2000, jumlah= 1, asal_usul= 1, harga= 50000	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Sistem akan menyimpan data aset yang dimasukkan dan menampilkan pesan sukses.	Valid

6.1.2 Pengujian Fungsi requestReportKadis pada Class AsetController

Pengujian unit fungsi *requestReportKadis* digunakan untuk melakukan permintaan pembuatan laporan aset dan persediaan.

Nama Class : AsetController
 Nama Fungsi : requestReportKadis()
 Algoritme :

Tabel 6.3 Pseudocode Fungsi requestReportKadis()

Pseudocode Fungsi requestReportKadis ()	
1	Mulai
2	validator <- memvalidasi input yang dikirimkan
3	IF (validator->fails() ==TRUE)
4	Return pesan error ENDIF
5	Menyimpan data objek PermintaanLaporan ke database
6	if (nama_permintaan = 1)
7	month <- mengambil bulan saat ini
8	If (month <=6)
9	semester = "Semester 1 dari bulan Januari s/d Juni"
10	Else semester = "Semester 2 dari bulan Juli s/d Desember"
11	END IF

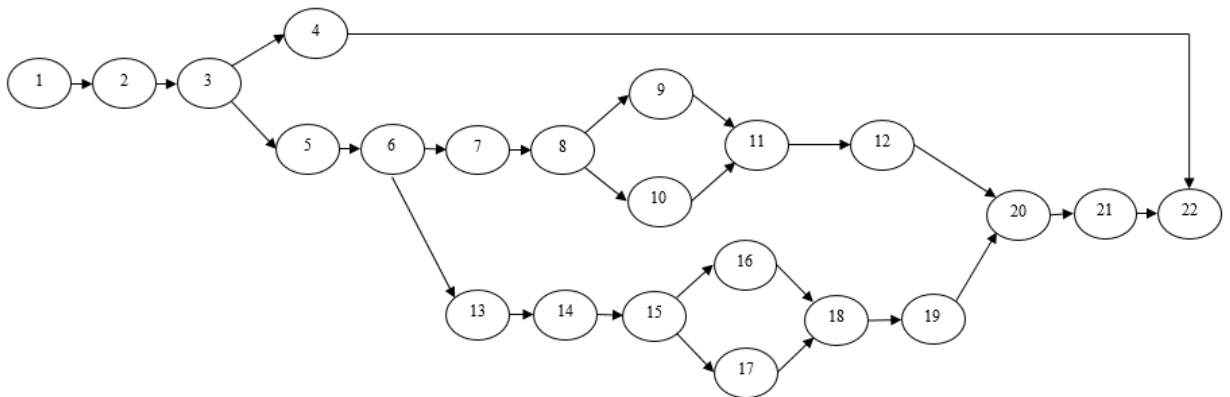
Tabel 6.3 Pseudocode Fungsi requestReportKadis() (Lanjutan)

Pseudocode Fungsi requestReportKadis()	
12	now <- dari inputan waktu saat ini dari objek datetime, date = now+format(d-m-Y), catatan <- mengambil data dari inputan, semester = mengambil data dari inputan, permintaanlaporan <- mengambil nama_permanintaan=1 pada objek PermintaanLaporan permintaanlaporan1 <- mengambil nama_permanintaan=1 pada objek PermintaanLaporan pengguna <- mengambil jabatan = 7 pada objek Pengguna pengguna1 <- mengambil jabatan = 2 pada objek Pengguna pengguna2 <- mengambil jabatan = 5 pada objek Pengguna pengguna3 <- mengambil jabatan = 3 pada objek Pengguna to <- nilai dari email penggunan cc <- nilai array dari email pengguna2 dan email pengguna3 form <- nilai email dari pengguna1 subject <- "Pembuatan Laporan Aset" Mengirimkan email
13	ELSEIF (nama_permintaan = 2)
14	month <- mengambil bulan saat ini
15	If (month <=6)
16	semester = "Semester 1 dari bulan Januari s/d Juni"
17	Else semester = "Semester 2 dari bulan Juli s/d Desember"
18	END IF
19	now <- dari inputan waktu saat ini dari objek datetime, date = now+format(d-m-Y), catatan <- mengambil data dari inputan, semester = mengambil data dari inputan, permintaanlaporan <- mengambil nama_permanintaan=2 pada objek PermintaanLaporan permintaanlaporan1 <- mengambil nama_permanintaan=2 pada objek PermintaanLaporan pengguna <- mengambil jabatan = 8 pada objek Pengguna pengguna1 <- mengambil jabatan = 2 pada objek Pengguna pengguna2 <- mengambil jabatan = 4 pada objek Pengguna pengguna3 <- mengambil jabatan = 6 pada objek Pengguna to<- nilai dari email penggunan

Tabel 6.3 Pseudocode Fungsi requestReportKadis() (Lanjutan)

Pseudocode Fungsi requestReportKadis()	
	cc <- nilai array dari email pengguna2 dan email pengguna3 form <- nilai email dari pengguna1 subject <- "Pembuatan Laporan Persediaan" Mengirimkan email
20	Endif
21	Redirect ke halaman permintaan laporan dan menampilkan pesan berhasil
22	Selesai

Berdasarkan *pseudocode* pada Tabel 6.1 didapatkan *basis path testing*. Pada Gambar 6.2 merupakan *flow graph* dari terjemahan *pseudocode*.



Gambar 6.2 Flow Graph Fungsi requestReportKadis()

Untuk mendapatkan perhitungan *cyclomatic complexity* dengan menggunakan rumus:

$$V(G) = \text{total Region} = 5$$

$$V(G) = \text{total Edge} - \text{total Node} + 2 = 25 - 22 + 2 = 5$$

$$V(G) = \text{Predicate Node} + 1 = 4 + 1 = 5$$

Dari hasil perhitungan *cyclomatic complexity*, didapatkan lima jalur *basis path*, yaitu:

1. 1-2-3-4-23
2. 1-2-3-5-6-7-8-9-12-21-22-23
3. 1-2-3-5-6-7-8-10-11-12-21-22-23
4. 1-2-3-5-6-13-14-15-16-19-21-22-23
5. 1-2-3-5-6-13-14-15-17-18-19-21-22-23

Tabel 6.4 Hasil Pengujian Unit Fungsi *requestReportKadis()*

No	Data Uji	Hasil Yang diharapkan	Hasil	Valid / Tidak Valid
1	Jika Nama_permintaan = NULL	Sistem akan menampilkan pesan “The nama permintaan <i>field is required</i> ”.	Sistem akan menampilkan pesan “The nama permintaan <i>field is required</i> ”.	Valid
2	Jika Nama_permintaan = 1 dan pada bulan januari sampai juni	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Valid
3	Jika Nama_permintaan = 1 dan bulan juli sampai desember	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Valid
4	Jika Nama_permintaan = 2 dan sampai dengan bulan juni	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Valid
5	Jika Nama_permintaan = 2 dan bulan juli sampai desember	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Sistem akan menyimpan permintaan pembuatan laporan dan mengirimkan notifikasi <i>email</i> .	Valid

6.1.3 Pengujian Fungsi *create* Pada Class PemegangKendaraanController

Pengujian unit fungsi *create* digunakan untuk menambahkan pemegang atas kendaraan dinas yang dilakukan oleh pengurus barang.

Nama Class : PemegangKendaraanController

Nama Fungsi : *create()*

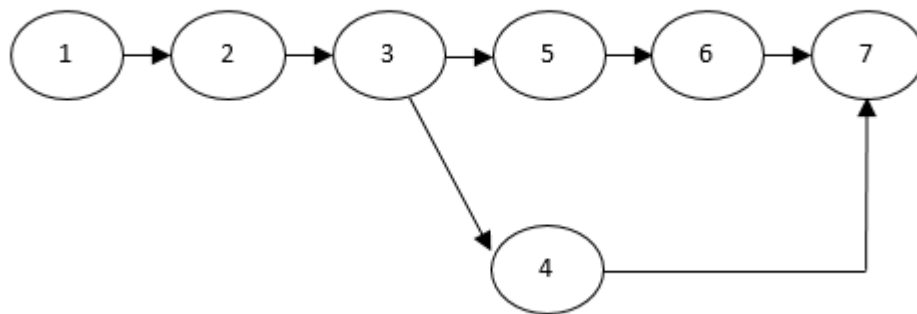
Algoritme :



Tabel 6.5 Pseudocode Fungsi create()

Pseudocode Fungsi create()	
1	Mulai
2	validator <- memvalidasi input yang dikirimkan
3	IF (validator->fails() ==TRUE)
4	Return pesan error ENDIF
5	Mencari aset_id pada objek Aset yang diinputkan dan decrement jumlah_tersedia = 1 Mencari pengguna_id pada objek Pengguna yang diinputkan dan increment jumlah_kendaraan = 1 Menyimpan data pada objek PemegangKendaraan ke database
6	Redirect dan menampilkan pesan berhasil
7	Selesai

Berdasarkan *pseudocode* pada Tabel 6.5 didapatkan *basis path testing*. Pada Gambar 6.3 merupakan *flow graph* dari terjemahan *pseudocode*.



Gambar 6.3 Flow Graph Fungsi create()

Untuk mendapatkan perhitungan *cyclomatic complexity* dengan menggunakan rumus:

$$V(G) = \text{total Region} = 2$$

$$V(G) = \text{total Edge} - \text{total Node} + 2 = 7 - 7 + 2 = 2$$

$$V(G) = \text{Predicate Node} + 1 = 1 + 1 = 2$$

Dari hasil perhitungan *cyclomatic complexity*, didapatkan dua jalur *basis path*, yaitu:

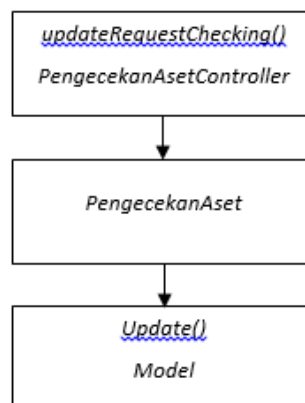
1. 1-2-3-4-7
2. 1-2-3-5-6-7

Tabel 6.6 Hasil Pengujian Unit Fungsi *requestReportKadis()*

No	Data Uji	Hasil Yang Diharapkan	Hasil	Valid /Tidak Valid
1	Jika Pegguna_id = NULL, Aset_id = NULL	Sistem akan menampilkan pesan error “The pengguna id field is required” dan “The aset_id field is required”.	Sistem akan menampilkan pesan error “The pengguna id field is required” dan “The aset_id field is required”.	Valid
2	Jika Pegguna_id = 16, Aset_id = 1	Sistem akan menyimpan data pemegang kendaraan dan menampilkan pesan sukses.	Sistem akan menyimpan data pemegang kendaraan dan menampilkan pesan sukses.	Valid

6.2 Pengujian Integrasi

Pengujian ini menggunakan pendekatan *Integration Incremental Top-Down*. Memakai sampel uji dari fungsi *updateRequestChecking()* pada kelas PengecekanAsetController dengan memanggil fungsi *update()* pada kelas Model. Disini kelas PengecekanAset merupakan *extend* dari kelas Model. Pada saat melakukan pengujian memakai *stub* untuk menggantikan modul yang akan diuji. Penggambaran pengujian ini terdapat pada Gambar 6.4. Untuk *scenario* pengujiannya terdapat pada Tabel 6.7.



Gambar 6.4 Diagram Hirarki *Integration Testing*



Tabel 6.7 Scenario Uji Integration Testing

Skenario Uji <i>Integration Testing</i>		
Menjalankan <i>updateRequestChecking()</i> di <i>PengecekanAsetController</i> serta menjalankan <i>stubUpdateRequestChecking()</i> di <i>Model</i>	fungsi di <i>class</i> serta fungsi di <i>class</i>	Fungsi <i>updateRequestChecking()</i> di <i>class</i> <i>PengecekanAsetController</i> di <i>running</i> guna menguji fungsi <i>update()</i> yang memakai <i>stubUpdateRequestChecking()</i> dengan memberikan nilai <i>input</i> baik = 1, buruk = 1, rusak = 1,

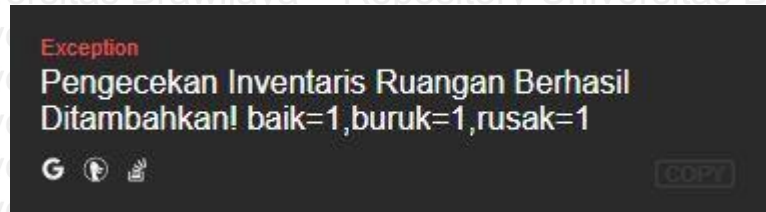
Tabel 6.8 Algoritme *stubUpdateRequestChecking()*

Algoritme <i>stubUpdateRequestChecking()</i>	
1	Mulai
2	Menerima parameter data
3	Mengembalikan isi dari variable
4	Selesai

Tabel 6.9 Algoritme *UpdateRequestChecking()*

Pseudocode <i>UpdateRequestChecking()</i>	
1	Mulai
2	<i>laporanpengecekan</i> <- berisi objek <i>LaporanPengecekan</i> id
3	<i>baik</i> = 1 <- mengisikan nilai 1 pada variabel <i>baik</i> <i>buruk</i> = 1 <- mengisikan nilai 1 pada variabel <i>buruk</i> <i>rusak</i> = 1 <- mengisikan nilai 1 pada variabel <i>rusak</i>
4	IF (<i>baik</i> != "" && <i>buruk</i> != "" && <i>rusak</i> != "")
5	<i>data</i> = array(<i>"baik"</i> <- dari inputan variabel <i>baik</i> <i>"buruk"</i> <- dari inputan variabel <i>buruk</i> <i>"rusak"</i> <- dari inputan variabel <i>rusak</i>) <i>stubUpdateRequestChecking(data)</i> <- memanggil fungsi <i>stub</i> dengan isian variabel <i>data</i> Mengembalikan isi dari variable <i>data</i>
6	ELSE Mengembalikan pesan error
7	ENDIF
8	Selsai

Hasil dari pengujian dengan menggunakan *variable* baik = 1, buruk = 1 dan rusak = 1 merupakan fungsi dari *updateRequestChecking()* yang berhasil memanggil fungsi *stubUpdateRequestChecking()* dengan mengembalikan nilai dari parameter dari *variable* tadi dengan hasil seperti Gambar 6.5.



Gambar 6.5 Hasil Pengujian Fungsi *stubUpdateRequestChecking()*

Kemudian *variable* baik, buruk dan rusak dikosongkan untuk menguji fungsi *updateRequestChecking()* apakah berhasil atau tidak memanggil fungsi *stubUpdateRequestChecking()* yang semestinya masuk ke *statement else*. Untuk *pseudocode* pada Tabel 6.10 dan hasilnya pada Gambar 6.6.

Tabel 6.10 *UpdateRequestChecking()*

Pseudocode <i>UpdateRequestChecking()</i>	
1	Mulai
2	laporanpengecekan <- berisi objek LaporanPengecekan id
3	baik = "" <- ketika variabel baik kosong buruk = "" <- ketika variabel buruk kosong rusak = "" <- ketika variabel kosong kosong
4	IF (baik != "" && buruk != "" && rusak != "")
5	Inisialisasi variable data = array("baik" <- dari inputan variabel baik "buruk" <- dari inputan variabel buruk "rusak" <- dari inputan variabel rusak) stubUpdateRequestChecking(data) <- memanggil fungsi stub dengan isian variabel data Mengembalikan isi dari variable data
6	ELSE Mengembalikan pesan error
7	ENDIF
8	Selsai



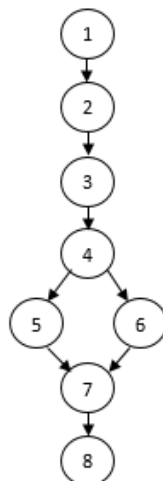
Gambar 6.6 Hasil Pengujian Fungsi *stubUpdateRequestChecking()*

Langkah berikutnya adalah melakukan pengujian jalur untuk memastikan bahwa integrasi fungsi *UpdateRequestChecking()* sudah berjalan dengan benar.

Tabel 6.11 *UpdateRequestChecking()*

<i>Pseudocode UpdateRequestChecking()</i>	
1	Mulai
2	laporanpengecekan <- berisi objek LaporanPengecekan id
3	baik <- nilai dari inputan baik buruk <- nilai dari inputan buruk rusak <- nilai dari inputan rusak
4	IF (baik != "" && buruk != "" && rusak != "")
5	Inisialisasi variable data = array("baik" <- dari inputan variabel baik "buruk" <- dari inputan variabel buruk "rusak" <- dari inputan variabel rusak) update(data) <- memanggil fungsi update pada model Mengembalikan pesan sukses dan kebalik ke halaman tadi
6	ELSE Mengembalikan pesan error dan kebalik ke halaman tadi
7	ENDIF
8	Selsai

Integration testing menggunakan teknik *white-box testing* yang memakai metode *basis path testing*. Untuk *flow graph* nya dapat dilihat pada Gambar 6.7.



Gambar 6.7 Flow Graph Integration

Untuk mendapatkan perhitungan *cyclomatic complexity* dengan menggunakan rumus:

$$V(G) = \text{total Region} = 2$$

$$V(G) = \text{total Edge} - \text{total Node} + 2 = 8 - 8 + 2 = 2$$

$$V(G) = \text{Predicate Node} + 1 = 1 + 1 = 2$$

Dari hasil perhitungan *cyclomatic complexity*, didapatkan lima jalur *basis path*, yaitu:

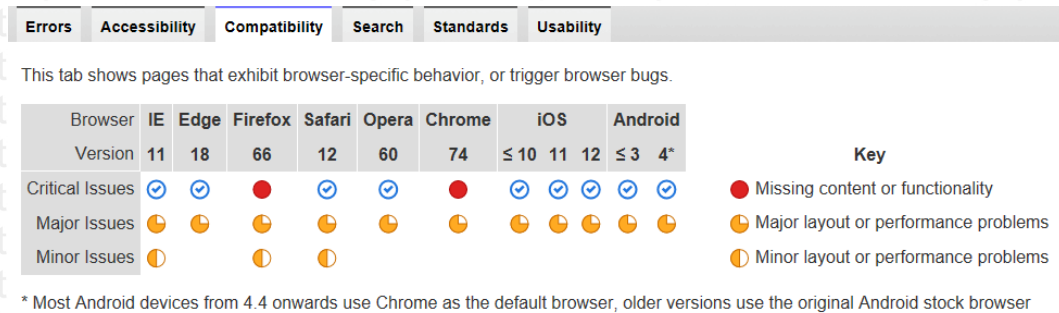
1. 1-2-3-4-5-7-8
2. 1-2-3-5-6-7-8

Tabel 6.12 Hasil Integration testing

No	Data Uji	Hasil Yang Diharapkan	Hasil	Valid / Tidak Valid
1	Jika baik =1, buruk = 1, rusak = 1	Sistem akan melihat pesan sukses "Daftar pengecekan aset berhasil dimasukkan!"	Sistem akan melihat pesan sukses "Daftar pengecekan aset berhasil dimasukkan!"	Valid
2	Jika baik = "", buruk = "", rusak = ""	Sistem akan melihat pesan error "Daftar pengecekan aset gagal dimasukkan!"	Sistem akan melihat pesan error "Daftar pengecekan aset gagal dimasukkan!"	Valid

6.3 Pengujian *Compatibility*

Pengujian *compatibility* dilaksanakan apakah SIMA yang sudah dibangun dapat berjalan sesuai dengan semestinya di semua *platform*. Pengujian *compatibility* menggunakan *tool Sortsite*. Dengan bantuan *tool Sortsite* nantinya akan dapat mengetahui apakah terdapat masalah ketika dijalankan di semua *platform* pada Gambar 6.8.



Gambar 6.8 Hasil Pengujian *Compatibility*

Dari hasil pengujian didapatkan hasil 75% bahwa sistem dapat berjalan pada *firefox* versi 66 dan *Chrome* versi 74 dengan *Critical issue* dikarenakan tidak aktifnya fitur *https* saat melakukan pengujian. Berjalan dengan *major issues* pada semua *platform* serta berjalan dengan *minor issues* pada *IE* versi 11, *firefox* versi 66 dan *safari* pada versi 12.

6.4 Pengujian Validasi

6.4.1 Pengujian Validasi Mendaftarkan Pengguna

Pada Tabel 6.13 adalah hasil pengujian persyaratan fungsional pada *use case* mendaftarkan pengguna.

Tabel 6.13 Skenario Uji Mendaftarkan Pengguna

Skenario Uji Mendaftarkan Pengguna	
Kode Kebutuhan	SIMA-PFS.01
Skenario	1. Klik menu daftar pengguna 2. Klik tanda tambah 3. Mengisikan <i>form</i> 4. Klik tombol simpan
<i>Expected Result</i>	Pengguna berhasil didaftarkan.
<i>Result</i>	Pengguna berhasil didaftarkan.
Valid /Tidak Valid	Valid

Pada Tabel 6.14 merupakan alternatif 1 dari *use case* mendaftarkan pengguna.

Tabel 6.14 Skenario Uji Mendaftarkan Pengguna Alternatif 1

Skenario Uji Mendaftarkan Pengguna Alternatif 1	
Kode Kebutuhan	SIMA-PFS.01
Skenario	1. Klik menu daftar pengguna. 2. Klik tanda tambah. 3. Tidak mengisikan <i>form</i>. 4. Klik simpan.
<i>Expected Result</i>	Pengguna tidak bisa didaftarkan serta melihat pesan <i>error</i> sesuai dengan <i>field</i> dari <i>form</i> yang kosong.
<i>Result</i>	Pengguna tidak berhasil didaftarkan dan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Valid /Tidak Valid	Valid

6.4.2 Pengujian Validasi Login

Pada Tabel 6.15 adalah hasil pengujian persyaratan fungsional pada *use case use case login*.

Tabel 6.15 Skenario Uji Login

Skenario Uji Login	
Kode Kebutuhan	SIMA-PFS.02
Skenario	1. Klik tombol <i>login</i>. 2. Mengisikan <i>form</i> untuk <i>username</i> dan <i>password</i>. 3. Klik tombol <i>Sign In</i>.
<i>Expected Result</i>	Pengguna berhasil masuk pada laman masing-masing.
<i>Result</i>	Pengguna berhasil masuk pada laman masing-masing.
Valid /Tidak Valid	Valid

Pada Tabel 6.16 alternatif 1 dari *use case login*.

Tabel 6.16 Skenario Uji Login Alternatif 1

Skenario Uji Login Alternatif 1	
Kode Kebutuhan	SIMA-PFS.02
Skenario	1. Klik tombol <i>Sign In</i>. 2. Mengisikan <i>username</i> yang salah dan <i>password</i>. 3. Klik tombol <i>Sign In</i>.
<i>Expected Result</i>	Pengguna gagal <i>Sign In</i> dan melihat pesan " <i>Username tidak terdaftar</i> ".
<i>Result</i>	Pengguna gagal <i>Sign In</i> dan melihat pesan " <i>Username tidak terdaftar</i> ".
Valid /Tidak Valid	Valid

Pada Tabel 6.17 merupakan alternatif 2 dari *use case login*.

Tabel 6.17 Skenario Uji Login Alternatif 2

Skenario Uji Login Alternatif 2	
Kode Kebutuhan	SIMA-PFS.02
Skenario	1. Klik tombol <i>login</i> . 2. Mengisikan <i>username</i> dan dengan <i>password</i> yang salah. Klik <i>Sign In</i> .
<i>Expected Result</i>	Pengguna gagal <i>Sign In</i> dan melihatkan pesan " <i>Password yang anda masukan salah!</i> ".
<i>Result</i>	Pengguna gagal <i>Sign In</i> dan melihatkan pesan " <i>Password yang anda masukan salah!</i> ".
Valid /Tidak Valid	Valid

6.4.3 Pengujian Validasi Daftar Pengguna

Pada Tabel 6.18 adalah hasil pengujian peryaratan fungsional *use case* daftar pengguna.

Tabel 6.18 Skenario Uji Daftar Pengguna

Skenario Uji Daftar Pengguna	
Kode Kebutuhan	SIMA-PFS.03
Skenario	1. Klik menu pengguna. 2. Klik submenu daftar pengguna.
<i>Expected Result</i>	Berhasil melihatkan daftar pengguna.
<i>Result</i>	Berhasil melihatkan daftar pengguna.
Valid /Tidak Valid	Valid

6.4.4 Pengujian Validasi Sunting Pengguna

Pada Tabel 6.19 yang merupakan rincian hasil pengujian fungsional dari *use case* sunting pengguna.

Tabel 6.19 Skenario Uji Sunting Pengguna

Skenario Uji Sunting Pengguna	
Kode Kebutuhan	SIMA-PFS.04
Skenario	1. Klik menu pengguna. 2. Klik submenu daftar pengguna. 3. Klik tombol sunting pada data pengguna yang akan diperbarui. 4. Mengisikan <i>field</i> yang tersedia.


Tabel 6.19 Skenario Uji Sunting Pengguna (Lanjutan)

Skenario Uji Sunting Pengguna	
Kode Kebutuhan	SIMA-PFS.04
	5. Klik simpan.
<i>Expected Result</i>	Berhasil memperbarui pengguna.
<i>Result</i>	Berhasil memperbarui pengguna.
Valid /Tidak Valid	Valid

Pada Tabel 6.20 merupakan alternatif 1 dari *use case* sunting pengguna.

Tabel 6.20 Skenario Uji Sunting Pengguna Alternatif 1

Skenario Uji Sunting Pengguna Alternatif 1	
Kode Kebutuhan	SIMA-PFS.04
Skenario	1. Klik menu pengguna. 2. Klik submenu daftar pengguna. 3. Klik tombol sunting pada data pengguna yang akan diperbarui. 4. Tidak mengisikan <i>field</i> yang tersedia. 5. Klik simpan.
<i>Expected Result</i>	Gagal memperbarui pengguna.
<i>Result</i>	Gagal memperbarui pengguna.
Valid /Tidak Valid	Valid

6.4.5 Pengujian Validasi Hapus Pengguna

Pada Tabel 6.21 yang adalah rincian hasil pengujian peryaratan fungsional *use case* hapus pengguna.

Tabel 6.21 Skenario Uji Hapus Pengguna

Skenario Uji Hapus Pengguna	
Kode Kebutuhan	SIMA-PFS.05
Skenario	1. Klik menu pengguna. 2. Klik submenu daftar pengguna. 3. Klik hapus pada pengguna yang akan dihapus. 3. Muncul <i>alert</i> kemudian klik ok.
<i>Expected Result</i>	Berhasil menghapus pengguna.
<i>Result</i>	Berhasil menghapus pengguna.
Valid /Tidak Valid	Valid

6.4.6 Pengujian Validasi Daftar Aset

Pada Tabel 6.22 adalah hasil pengujian persyaratan fungsional *use case* daftar aset.

Tabel 6.22 Skenario Uji Daftar Aset

Skenario Uji Daftar Aset	
Kode Kebutuhan	SIMA-PFS.06
Skenario	1. Klik menu aset dinas. 2. Klik submenu daftar aset.
<i>Expected Result</i>	Berhasil melihatkan daftar aset.
<i>Result</i>	Berhasil melihatkan daftar aset.
Valid /Tidak Valid	Valid

6.4.7 Pengujian Validasi Tambah Aset

Pada Tabel 6.23 adalah hasil pengujian persyaratan fungsional *use case* tambah aset.

Tabel 6.23 Skenario Uji Tambah Aset

Skenario Uji Tambah Aset	
Kode Kebutuhan	SIMA-PFS.07
Skenario	1. Klik menu aset dinas. 2. Klik sub menu daftar aset. 3. Memilih kategori aset pada <i>combo box</i> . 4. Mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 5. Klik tombol simpan.
<i>Expected Result</i>	Berhasil menambah aset ke dalam sistem.
<i>Result</i>	Berhasil menambah aset ke dalam sistem.
Valid /Tidak Valid	Valid

Pada Tabel 6.24 merupakan alternatif 1 dari *use case* tambah aset.

Tabel 6.24 Skenario Uji Tambah Aset Alternatif 1

Skenario Uji Tambah Aset Alternatif 1	
Kode Kebutuhan	SIMA-PFS.07
Skenario	1. Klik menu aset dinas. 2. Klik sub menu daftar aset. 3. Memilih kategori aset pada <i>combo box</i> . 4. Tidak mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 5. Klik simpan.

**Tabel 6.24 Skenario Uji Tambah Aset Alternatif 1 (Lanjutan)**

Skenario Uji Tambah Aset Alternatif 1	
<i>Expected Result</i>	Tidak bisa menyimpan aset dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.
<i>Result</i>	Tidak bisa menyimpan aset dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.
Valid /Tidak Valid	Valid

6.4.8 Pengujian Validasi Sunting Aset

Pada Tabel 6.25 adalah hasil pengujian persyaratan fungsional *use case* sunting aset.

Tabel 6.25 Skenario Uji Sunting Aset

Skenario Uji Sunting Aset	
Kode Kebutuhan	SIMA-PFS.08
Skenario	1. Klik menu aset dinas. 2. Klik sub menu daftar aset. 3. Klik tombol sunting sesuai dengan aset yang akan diperbarui. 4. Mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 5. Klik simpan.
<i>Expected Result</i>	Berhasil perbarui aset dan menyimpan ke dalam sistem.
<i>Result</i>	Berhasil perbarui aset dan menyimpan ke dalam sistem.
Valid /Tidak Valid	Valid

Pada Tabel 6.26 merupakan alternatif 1 dari *use case* sunting aset.

Tabel 6.26 Skenario Uji Sunting Aset Alternatif 1

Skenario Uji Sunting Aset Alternatif 1	
Kode Kebutuhan	SIMA-PFS.08
Skenario	1. Klik menu aset dinas. 2. Klik sub menu daftar aset. 3. Klik tombol sunting sesuai dengan aset yang akan diperbarui. 4. Mengosongkan <i>field</i> data aset yang tersedia. 5. Klik simpan.
<i>Expected Result</i>	Tidak bisa memperbarui aset dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.



Tabel 6.26 Skenario Uji Sunting Aset Alternatif 1 (Lanjutan)

Skenario Uji Sunting Aset Alternatif 1	
<i>Result</i>	Tidak bisa memperbarui aset dan mellihatkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.
Valid /Tidak Valid	Valid

6.4.9 Pengujian Validasi Hapus Aset

Pada Tabel 6.27 adalah hasil pengujian persyaratan fungsional *use case* hapus aset.

Tabel 6.27 Skenario Uji Hapus Aset

Skenario Uji Hapus Aset	
Kode Kebutuhan	SIMA-PFS.09
Skenario	1. Klik menu aset dinas. 2. Klik sub menu daftar aset. 3. Klik tombol hapus sesuai dengan aset yang akan dihapus. 4. Muncul <i>alert</i> dan klik tombol oke.
<i>Expected Result</i>	Berhasil menghapus aset.
<i>Result</i>	Berhasil menghapus aset.
Valid /Tidak Valid	Valid

6.4.10 Pengujian Validasi Permintaan Laporan Aset

Pada Tabel 6.28 adalah hasil pengujian persyaratan fungsional *use case* permintaan laporan aset.

Tabel 6.28 Skenario Uji Permintaan Laporan Aset

Skenario Uji Permintaan Laporan Aset	
Kode Kebutuhan	SIMA-PFS.10
Skenario	1. Klik menu permintaan laporan 2. Memilih laporan aset pada <i>combo box</i> 3. Mengisikan <i>form</i> catatan 4. Klik tombol <i>request</i> laporan
<i>Expected Result</i>	Berhasil mengirimkan permintaan pembuatan laporan aset
<i>Result</i>	Berhasil mengirimkan permintaan pembuatan laporan aset
Valid /Tidak Valid	Valid

Pada Tabel 6.29 merupakan alternatif 1 dari *use case* permintaan laporan aset.

Tabel 6.29 Skenario Uji Permintaan Laporan Aset Alternatif 1

Skenario Uji Permintaan Laporan Aset Alternatif 1	
Kode Kebutuhan	SIMA-PFS.10
Skenario	1. Klik menu permintaan laporan 2. Tidak memilih laporan pada <i>combo box</i> 3. Mengisikan <i>form</i> catatan 4. Klik tombol <i>request</i> laporan
<i>Expected Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan aset dan menampilkan pesan <i>error</i> “The nama permintaan <i>field is required</i> ”
<i>Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan aset dan menampilkan pesan <i>error</i> “The nama permintaan <i>field is required</i> ”
Valid /Tidak Valid	Valid

6.4.11 Pengujian Validasi Daftar Laporan Aset

Pada Tabel 6.30 adalah hasil pengujian fungsional *use case* daftar laporan aset.

Tabel 6.30 Skenario Uji Daftar Laporan Aset

Skenario Uji Daftar Laporan Aset	
Kode Kebutuhan	SIMA-PFS.11
Skenario	1. Klik menu validasi laporan. 2. Klik submenu aset.
<i>Expected Result</i>	Berhasil lihat daftar laporan aset.
<i>Result</i>	Berhasil lihat daftar laporan aset.
Valid /Tidak Valid	Valid

6.4.12 Pengujian Validasi Sunting Laporan Aset

Pada Tabel 6.31 adalah hasil pengujian persyaratan fungsional *use case* sunting laporan aset.

Tabel 6.31 Skenario Uji Sunting Laporan Aset

Skenario Uji Sunting Laporan Aset	
Kode Kebutuhan	SIMA-PFS.12
Skenario	1. Klik menu validasi laporan. 2. Klik sub menu pelaporan aset.

Tabel 6.31 Skenario Uji Sunting Laporan Aset (Lanjutan)

Skenario Uji Sunting Laporan Aset	
	3. Klik tombol lihat detail sesuai dengan aset yang akan diperbarui. 4. Klik tombol sunting pada data aset. 5. Mengisikan <i>field</i> yang tersedia. 6. Klik tombol “simpan”.
Kode Kebutuhan	SIMA-PFS.12
<i>Expected Result</i>	Berhasil memperbarui laporan aset.
<i>Result</i>	Berhasil memperbarui laporan aset.
Valid /Tidak Valid	Valid

Pada Tabel 6.32 merupakan alternatif 1 dari *use case* sunting laporan aset.

Tabel 6.32 Skenario Uji Sunting Laporan Aset Alternatif 1

Skenario Uji Sunting Laporan Aset Alternatif 1	
Kode Kebutuhan	SIMA-PFS.12
Skenario	1. Klik menu validasi laporan. 2. Klik sub menu pelaporan aset. 3. Klik tombol lihat detail sesuai dengan aset yang akan diperbarui. 4. Klik tombol sunting pada data aset. 5. Tidak mengisikan <i>field</i> yang tersedia. 6. Klik tombol “simpan”.
<i>Expected Result</i>	Gagal memperbarui laporan aset.
<i>Result</i>	Gagal memperbarui laporan aset.
Valid /Tidak Valid	Valid

6.4.13 Pengujian Validasi Laporan Aset

Pada Tabel 6.30 adalah hasil pengujian persyaratan fungsional *use case* laporan aset.

Tabel 6.33 Skenario Uji Laporan Aset

Skenario Uji Laporan Aset	
Kode Kebutuhan	SIMA-PFS.13
Skenario	1. Klik menu validasi laporan. 2. Klik sub menu pelaporan aset. 3. Klik tombol lihat detail sesuai dengan aset yang akan dilaporkan. 4. Klik tombol Laporkan. 5. Klik tombol laporkan! pada <i>pop-up alert</i> .

**Tabel 6.33 Skenario Uji Laporan Aset (Lanjutan)**

Skenario Uji Laporan Aset	
Kode Kebutuhan	SIMA-PFS.13
<i>Expected Result</i>	Berhasil mengirmkan laporan aset.
<i>Result</i>	Berhasil mengirmkan laporan aset.
Valid /Tidak Valid	Valid

6.4.14 Pengujian Validasi Daftar Validasi Aset

Pada Tabel 6.34 adalah hasil pengujian fungsional *use case* validasi laporan aset.

Tabel 6.34 Skenario Uji Validasi Laporan Aset

Skenario Uji Validasi Laporan Aset	
Kode Kebutuhan	SIMA-PFS.14
Skenario	1. Klik menu validasi laporan. 2. Klik submenu aset.
<i>Expected Result</i>	Berhasil lihat daftar validasi laporan aset.
<i>Result</i>	Berhasil lihat daftar validasi laporan aset.
Valid /Tidak Valid	Valid

6.4.15 Pengujian Validasi Validasi Laporan Aset

Pada Tabel 6.35 adalah hasil pengujian fungsional *use case* validasi laporan aset.

Tabel 6.35 Skenario Uji Validasi Laporan Aset

Skenario Uji Validasi Laporan Aset	
Kode Kebutuhan	SIMA-PFS.15
Skenario	1. Klik menu validasi laporan. 2. Klik submenu aset. 3. Klik tombol lihat detail sesuai dengan aset yang akan divalidasi. 4. Memberikan tanda pada <i>checkbox</i> . 5. Klik tombol validasi. 6. Klik tombol validasi! pada <i>pop-up alert</i> .
<i>Expected Result</i>	Berhasil melakukan validasi laporan aset.
<i>Result</i>	Berhasil melakukan validasi laporan aset.
Valid /Tidak Valid	Valid

Pada Tabel 6.36 merupakan alternatif 1 dari *use case* validasi laporan aset.

Tabel 6.36 Skenario Uji Validasi Laporan Aset Alternatif 1

Skenario Uji Validasi Laporan Aset Alternatif 1	
Kode Kebutuhan	SIMA-PFS.15
Skenario	1. Klik menu validasi laporan. 2. Klik submenu aset. 3. Klik tombol lihat detail sesuai dengan aset yang akan divalidasi. 4. Memberikan tanda pada <i>checkbox</i>. 5. Klik tombol tolak validasi. 6. Mengisikan catatan dan menekan tombol simpan pada <i>pop-up alert</i>.
<i>Expected Result</i>	Berhasil melakukan tolak validasi laporan aset.
<i>Result</i>	Berhasil melakukan tolak validasi laporan aset.
Valid /Tidak Valid	Valid

Pada Tabel 6.37 merupakan alternatif 2 dari *use case* validasi laporan aset.

Tabel 6.37 Skenario Uji Validasi Laporan Aset Alternatif 2

Skenario Uji Validasi Laporan Aset Alternatif 2	
Kode Kebutuhan	SIMA-PFS.15
Skenario	1. Klik menu validasi laporan. 2. Klik submenu aset. 3. Klik tombol lihat detail sesuai dengan aset yang akan divalidasi. 4. Tidak memberikan tanda pada <i>checkbox</i>. 5. Klik tombol tolak validasi. 6. Mengisikan catatan dan menekan tombol simpan pada <i>pop-up alert</i>.
<i>Expected Result</i>	Tidak bisa melakukan tolak validasi laporan aset.
<i>Result</i>	Tidak bisa melakukan tolak validasi laporan aset.
Valid /Tidak Valid	Valid

Pada Tabel 6.38 merupakan alternatif 3 dari *use case* validasi laporan aset.

Tabel 6.38 Skenario Uji Validasi Laporan Aset Alternatif 3

Skenario Uji Validasi Laporan Aset Alternatif 3	
Kode Kebutuhan	SIMA-PFS.15
Skenario	1. Klik menu validasi laporan. 2. Klik submenu aset.

Tabel 6.38 Skenario Uji Validasi Laporan Aset Alternatif 3 (Lanjutan)

Skenario Uji Validasi Laporan Aset Alternatif 3	
Kode Kebutuhan	SIMA-PFS.15
	3. Klik tombol lihat detail sesuai dengan aset yang akan divalidasi. 4. Tidak memberikan tanda pada <i>checkbox</i> . 5. Klik tombol validasi.
<i>Expected Result</i>	Tidak bisa melakukan validasi laporan aset.
<i>Result</i>	Tidak bisa melakukan validasi laporan aset.
Valid /Tidak Valid	Valid

6.4.16 Pengujian Validasi *Export* Laporan Aset

Pada Tabel 6.39 adalah hasil pengujian persyaratan fungsional *use case export* laporan persediaan.

Tabel 6.39 Skenario Uji *Export* Laporan Aset

Skenario Uji <i>Export</i> Laporan Aset	
Kode Kebutuhan	SIMA-PFS.16
Skenario	1. Pastikan laporan aset sudah selesai divalidasi. 2. Klik menu aset dinas. 3. Klik submenu <i>export</i> pelaporan aset. 4. Pilih kategori aset dan kode lokasi. 5. Klik tombol <i>export pdf</i> .
<i>Expected Result</i>	Berhasil meng- <i>export</i> laporan aset.
<i>Result</i>	Berhasil meng- <i>export</i> laporan aset.
Valid /Tidak Valid	Valid

Pada Tabel 6.40 merupakan alternatif 1 dari *use case export* laporan aset.

Tabel 6.40 Skenario Uji *Export* Laporan Aset Alternatif 1

Skenario Uji <i>Export</i> Laporan Aset	
Kode Kebutuhan	SIMA-PFS.16
Skenario	1. Pastikan laporan aset sudah selesai divalidasi. 2. Klik menu aset dinas. 3. Klik submenu <i>export</i> pelaporan aset. 4. Tidak mengisi <i>field</i> yang tersedia. 5. Klik tombol <i>export pdf</i> .
<i>Expected Result</i>	Gagal meng- <i>export</i> laporan aset.
<i>Result</i>	Gagal meng- <i>export</i> laporan aset.
Valid /Tidak Valid	Valid

6.4.17 Pengujian Validasi Daftar Persediaan

Pada Tabel 6.41 adalah hasil pengujian persyaratan fungsional *use case* daftar persediaan.

Tabel 6.41 Skenario Uji Daftar Persediaan

Skenario Uji Daftar Persediaan	
Kode Kebutuhan	SIMA-PFS.17
Skenario	1. Klik menu persediaan. 2. Klik submenu daftar persediaan.
<i>Expected Result</i>	Berhasil menampilkan daftar persediaan.
<i>Result</i>	Berhasil menampilkan daftar persediaan.
Valid /Tidak Valid	Valid

6.4.18 Pengujian Validasi Tambah Persediaan

Pada Tabel 6.42 yang adalah hasil pengujian persyaratan fungsional dari *use case* tambah persediaan.

Tabel 6.42 Skenario Uji Tambah Persediaan

Skenario Uji Tambah Persediaan	
Kode Kebutuhan	SIMA-PFS.18
Skenario	1. Klik menu persediaan. 2. Klik sub menu daftar persediaan. 3. Klik tanda tambah. 4. Mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 5. Klik tombol simpan.
<i>Expected Result</i>	Berhasil menambah persediaan ke dalam sistem.
<i>Result</i>	Berhasil menambah persediaan ke dalam sistem.
Valid /Tidak Valid	Valid

Pada Tabel 6.43 merupakan alternatif 1 dari *use case* tambah persediaan.

Tabel 6.43 Skenario Uji Tambah Persediaan Alternatif 1

Skenario Uji Tambah Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.18
Skenario	1. Klik menu persediaan. 2. Klik sub menu daftar persediaan. 3. Klik tanda tambah. 4. Tidak mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 5. Klik simpan.

**Tabel 6.43 Skenario Uji Tambah Persediaan Alternatif 1 (Lanjutan)**

Skenario Uji Tambah Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.18
<i>Expected Result</i>	Tidak bisa menyimpan persediaan dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
<i>Result</i>	Tidak bisa menyimpan persediaan dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Valid /Tidak Valid	Valid

6.4.19 Pengujian Validasi Sunting Persediaan

Pada Tabel 6.44 yang adalah hasil pengujian persyaratan fungsional dari *use case* sunting persediaan.

Tabel 6.44 Skenario Uji Sunting Persediaan

Skenario Uji Sunting Persediaan	
Kode Kebutuhan	SIMA-PFS.19
Skenario	1. Klik menu persediaan. 2. Klik submenu daftar persediaan. 3. Klik tombol sunting pada data persediaan yang akan diperbarui. 4. Mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 5. Klik simpan.
<i>Expected Result</i>	Berhasil perbarui persediaan.
<i>Result</i>	Berhasil perbarui persediaan.
Valid /Tidak Valid	Valid

Pada Tabel 6.45 merupakan alternatif 1 dari *use case* sunting persediaan persediaan.

Tabel 6.45 Skenario Uji Sunting Persediaan Alternatif 1

Skenario Uji Sunting Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.19
Skenario	1. Klik menu persediaan. 2. Klik sub menu daftar persediaan. 3. Klik tombol sunting pada data persediaan yang akan diperbarui. 4. Mengosongkan <i>field</i> data persediaan. 5. Klik simpan.
<i>Expected Result</i>	Tidak bisa memperbarui persediaan dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.
<i>Result</i>	Tidak bisa memperbarui persediaan dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> yang kosong.
Valid /Tidak Valid	Valid

6.4.20 Pengujian Validasi *Order* Persediaan

Pada Tabel 6.46 yang adalah hasil pengujian persyaratan fungsional *use case order* persediaan.

Tabel 6.46 Skenario Uji *Order* Persediaan

Skenario Uji <i>Order</i> Persediaan	
Kode Kebutuhan	SIMA-PFS.20
Skenario	1. Klik menu persediaan. 2. Klik sub menu <i>order</i> persediaan. 3. Mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 4. Klik tombol simpan.
<i>Expected Result</i>	Berhasil memasukkan <i>order</i> persediaan ke dalam sistem.
<i>Result</i>	Berhasil memasukkan <i>order</i> persediaan ke dalam sistem.
Valid /Tidak Valid	Valid

Pada Tabel 6.47 merupakan alternatif 1 dari *use case order* persediaan.

Tabel 6.47 Skenario Uji *Order* Persediaan Alternatif 1

Skenario Uji <i>Order</i> Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.20
Skenario	1. Klik menu persediaan. 2. Klik sub menu <i>order</i> persediaan. 3. Tidak mengisikan <i>form</i> sesuai dengan <i>field</i> yang tersedia. 4. Klik simpan.
Kode Kebutuhan	SIMA-PFS.17
<i>Expected Result</i>	Tidak bisa menyimpan <i>order</i> persediaan dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
<i>Result</i>	Tidak bisa menyimpan <i>order</i> persediaan dan melihatkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Valid /Tidak Valid	Valid

Pada Tabel 6.48 merupakan alternatif 2 dari *use case order* persediaan.

Tabel 6.48 Skenario Uji *Order* Persediaan Alternatif 2

Skenario Uji <i>Order</i> Persediaan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.20
Skenario	1. Klik menu persediaan. 2. Klik sub menu <i>order</i> persediaan.

**Tabel 6.48 Skenario Uji Order Persediaan Alternatif 2 (Lanjutan)**

Skenario Uji Order Persediaan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.20
	3. Mengisikan <i>form</i> jumlah <i>order</i> melebihi jumlah ketersediaan persediaan. 4. Klik tombol simpan.
<i>Expected Result</i>	Tidak bisa menyimpan <i>order</i> persediaan dan melihatkan pesan “Ada barang yang stoknya tidak mencukupi!”.
<i>Result</i>	Tidak bisa menyimpan <i>order</i> persediaan dan melihatkan pesan “Ada barang yang stoknya tidak mencukupi!”.
Valid /Tidak Valid	Valid

6.4.21 Pengujian Validasi Daftar Pengeluaran

Pada Tabel 6.49 yang adalah hasil pengujian persyaratan fungsional *use case* daftar pengeluaran.

Tabel 6.49 Skenario Uji Daftar Pengeluaran

Skenario Uji Daftar Pengeluaran	
Kode Kebutuhan	SIMA-PFS.21
Skenario	1. Klik menu persediaan. 2. Klik sub menu penerbitan surat.
<i>Expected Result</i>	Berhasil lihat daftar pengeluaran berdasarkan pemohon.
<i>Result</i>	Berhasil lihat daftar pengeluaran berdasarkan pemohon.
Valid /Tidak Valid	Valid

6.4.22 Pengujian Validasi Penerbitan Surat

Pada Tabel 6.50 yang adalah hasil pengujian persyaratan fungsional dari *use case* penerbita surat.

Tabel 6.50 Skenario Uji Penerbitan Surat

Skenario Uji Penerbitan Surat	
Kode Kebutuhan	SIMA-PFS.22
Skenario	1. Klik menu persediaan. 2. Klik sub menu penerbitan surat. 3. Klik lihat <i>detail</i> berdasarkan nama pemohon 4. Klik tombol penerbitan surat.

**Tabel 6.50 Skenario Uji Penerbitan Surat (Lanjutan)**

Skenario Uji Penerbitan Surat	
Kode Kebutuhan	SIMA-PFS.22
<i>Expected Result</i>	Berhasil menerbitkan surat.
<i>Result</i>	Berhasil menerbitkan surat.
Valid /Tidak Valid	Valid

6.4.23 Pengujian Validasi Hapus *Order Persediaan*

Pada Tabel 6.51 yang adalah hasil pengujian persyaratan fungsional *use case* hapus *order persediaan*.

Tabel 6.51 Skenario Uji Hapus *Order Persediaan*

Skenario Uji Hapus <i>Order Persediaan</i>	
Kode Kebutuhan	SIMA-PFS.23
Skenario	1. Klik menu persediaan. 2. Klik sub menu penerbitan surat. 3. Klik lihat <i>detail</i> berdasarkan nama pemohon. 4. Klik tombol hapus.
<i>Expected Result</i>	Berhasil menghapus <i>order persediaan</i> .
<i>Result</i>	Berhasil menghapus <i>order persediaan</i> .
Valid /Tidak Valid	Valid

6.4.24 Pengujian Validasi Permintaan Laporan Persediaan

Pada Tabel 6.52 yang adalah hasil pengujian persyaratan fungsional *use case* permintaan laporan persediaan.

Tabel 6.52 Skenario Uji Permintaan Laporan Persediaan

Skenario Uji Permintaan Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.24
Skenario	1. Klik menu permintaan laporan. 2. Memilih laporan persediaan pada <i>combo box</i>. 3. Mengisikan <i>form</i> catatan. 4. Klik tombol <i>request</i> laporan.
<i>Expected Result</i>	Berhasil mengirimkan permintaan pembuatan laporan persediaan.
<i>Result</i>	Berhasil mengirimkan permintaan pembuatan laporan persediaan.
Valid /Tidak Valid	Valid



Pada Tabel 6.53 merupakan alternatif 1 dari permintaan laporan persediaan.

Tabel 6.53 Skenario Uji Permintaan Laporan Persediaan Alternatif 1

Skenario Uji Permintaan Laporan Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.24
Skenario	1. Klik menu permintaan laporan. 2. Tidak memilih laporan persediaan pada <i>combo box</i>. 3. Mengisikan <i>form</i> catatan. 4. Klik tombol <i>request</i> laporan.
<i>Expected Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan persediaan dan menampilkan pesan <i>error</i> "The nama permintaan <i>field is required</i> ".
<i>Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan persediaan dan menampilkan pesan <i>error</i> "The nama permintaan <i>field is required</i> ".
Valid /Tidak Valid	Valid

6.4.25 Pengujian Validasi Daftar laporan Persediaan

Pada Tabel 6.54 yang adalah hasil pengujian persyaratan fungsional *use case* daftar laporan persediaan.

Tabel 6.54 Skenario Uji Daftar Laporan Persediaan

Skenario Uji Daftar Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.25
Skenario	1. Klik menu persediaan. 2. Klik sub menu pelaporan persediaan.
<i>Expected Result</i>	Berhasil lihat daftar laporan persediaan.
<i>Result</i>	Berhasil lihat daftar laporan persediaan.
Valid /Tidak Valid	Valid

6.4.26 Pengujian Validasi Laporan Persediaan

Pada Tabel 6.55 yang adalah hasil pengujian persyaratan fungsional *use case* laporan persediaan.

Tabel 6.55 Skenario Uji Laporan Persediaan

Skenario Uji Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.26
Skenario	1. Klik menu persediaan. 2. Klik sub menu pelaporan persediaan.

**Tabel 6.55 Skenario Uji Laporan Persediaan (Lanjutan)**

Skenario Uji Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.26
	3. Klik tombol Laporkan. 4. Klik tombol laporkan! pada <i>pop-up alert</i> .
<i>Expected Result</i>	Berhasil mengirimkan laporan persediaan.
<i>Result</i>	Berhasil mengirimkan laporan persediaan.
Valid /Tidak Valid	Valid

6.4.27 Pengujian Validasi Daftar Validasi Persediaan

Pada Tabel 6.56 yang adalah hasil pengujian persyaratan fungsional *use case* daftar validasi persediaan.

Tabel 6.56 Skenario Uji Daftar Validasi Persediaan

Skenario Uji Validasi Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.27
Skenario	1. Klik menu daftar validasi. 2. Klik submenu persediaan.
<i>Expected Result</i>	Berhasil lihat daftar validasi laporan persediaan.
<i>Result</i>	Berhasil lihat daftar validasi laporan persediaan.
Valid /Tidak Valid	Valid

6.4.28 Pengujian Validasi Validasi Laporan Persediaan

Pada Tabel 6.57 yang adalah hasil pengujian persyaratan fungsional *use case* validasi laporan persediaan.

Tabel 6.57 Skenario Uji Validasi Laporan Persediaan

Skenario Uji Validasi Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.28
Skenario	1. Klik menu daftar validasi. 2. Klik submenu persediaan. 3. Klik tombol lihat <i>detail</i> sesuai dengan persediaan yang akan divalidasi. 4. Menandai checkbox dan tekan tombol validasi. 5. Klik tombol validasi! pada <i>pop-up alert</i> .
<i>Expected Result</i>	Berhasil validasi laporan persediaan.
<i>Result</i>	Berhasil validasi laporan persediaan.
Valid /Tidak Valid	Valid

Pada Tabel 6.58 merupakan alternatif 1 dari *use case* validasi laporan persediaan.

Tabel 6.58 Skenario Uji Validasi Laporan Persediaan Alternatif 1

Skenario Uji Validasi Laporan Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.28
Skenario	1. Klik menu daftar validasi. 2. Klik submenu persediaan. 3. Klik tombol lihat <i>detail</i> sesuai dengan persediaan yang akan divalidasi. 4. Tidak menandai <i>checkbox</i> dan tekan tombol validasi. 6. Klik tombol validasi pada <i>pop-up alert</i>.
<i>Expected Result</i>	Gagal memvalidasi laporan persediaan.
<i>Result</i>	Gagal memvalidasi laporan persediaan.
Valid /Tidak Valid	Valid

6.4.29 Pengujian Validasi *Export* Laporan Persediaan

Pada Tabel 6.59 yang adalah hasil pengujian persyaratan fungsional dari *use case export* laporan persediaan.

Tabel 6.59 Skenario Uji *Export* Laporan Persediaan

Skenario Uji <i>Export</i> Laporan Persediaan	
Kode Kebutuhan	SIMA-PFS.29
Skenario	1. Pastikan laporan persediaan sudah selesai divalidasi. 2. Klik menu persediaan. 3. Klik sub menu mutasi persediaan. 4. Memilih semester dan tahun. 5. Klik tombol cari. 6. Klik tombol <i>export pdf</i>.
<i>Expected Result</i>	Bisa meng- <i>export</i> mutasi laproan persediaan sesuai dengan masukan.
<i>Result</i>	Bisa meng- <i>export</i> mutasi laporan persediaan sesuai dengan masukan.
Valid /Tidak Valid	Valid

Pada Tabel 6.60 merupakan alternatif 1 dari *use case export* laporan persediaan.

Tabel 6.60 Skenario Uji Export Laporan Persediaan Alternatif 1

Skenario Uji Export Laporan Persediaan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.29
	1. Pastikan laporan persediaan sudah selesai divalidasi. 2. Klik menu persediaan. 3. Klik sub menu mutasi persediaan. 4. Tidak memilih semester dan tahun. 5. Klik tombol cari. 7. Klik tombol <i>export pdf</i>.
<i>Expected Result</i>	Gagal meng- <i>export</i> mutasi laproan persediaan sesuai dengan masukan dan mellihatkan pesan <i>error</i> .
<i>Result</i>	Gagal meng- <i>export</i> mutasi laporan persediaan sesuai dengan masukan dan mellihatkan pesan <i>error</i> .
Valid /Tidak Valid	Valid

Pada Tabel 6.60 merupakan alternatif 2 dari *use case export* laporan persediaan.

Tabel 6.61 Skenario Uji Export Laporan Persediaan Alternatif 2

Skenario Uji Export Laporan Persediaan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.29
	1. Pastikan laporan persediaan sudah selesai divalidasi. 2. Klik menu persediaan. 3. Klik sub menu mutasi persediaan. 4. Tidak memilih salah satu dari semester atau tahun. 5. Klik tombol cari. 8. Klik tombol <i>export pdf</i>.
<i>Expected Result</i>	Gagal meng- <i>export</i> mutasi laproan persediaan sesuai dengan masukan dan mellihatkan pesan <i>error</i> .
<i>Result</i>	Gagal meng- <i>export</i> mutasi laporan persediaan sesuai dengan masukan dan mellihatkan pesan <i>error</i> .
Valid /Tidak Valid	Valid

Pada Tabel 6.62 merupakan alternatif 3 dari *use case export* laporan persediaan.

Tabel 6.62 Skenario Uji Export Laporan Persediaan Alternatif 3

Skenario Uji Export Laporan Persediaan Alternatif 3	
Kode Kebutuhan	SIMA-PFS.29
	1. Pastikan laporan persediaan sudah selesai divalidasi. 2. Klik menu persediaan. 3. Klik sub menu mutasi persediaan. 4. Memilih semester atau tahun tidak sesuai dengan kondisi waktu saat itu. 5. Klik tombol cari. 6. Klik tombol <i>export pdf</i>.
<i>Expected Result</i>	Gagal meng- <i>export</i> mutasi laporan persediaan sesuai dengan masukan dan melihatkan pesan <i>error</i> .
<i>Result</i>	Gagal meng- <i>export</i> mutasi laporan persediaan sesuai dengan masukan dan melihatkan pesan <i>error</i> .
Valid /Tidak Valid	Valid

6.4.30 Pengujian Validasi Tambah Pemegang Kendaraan

Pada Tabel 6.63 yang adalah hasil pengujian persyaratan fungsional *use case* tambah pemegang kendaraan.

Tabel 6.63 Skenario Uji Tambah Pemegang Kendaraan

Skenario Uji Tambah Pemegang Kendaraan	
Kode Kebutuhan	SIMA-PFS.30
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu daftar pemegang kendaraan. 3. Memilih nama pemegang dan nomor polisi kendaraan pada <i>combo box</i>. 4. Klik tombol simpan.
<i>Expected Result</i>	Berhasil menambah pemegang kendaraan pada sistem.
<i>Result</i>	Berhasil menambah pemegang kendaraan pada sistem.
Valid /Tidak Valid	Valid

Pada Tabel 6.64 merupakan alternatif 1 dari *use case* tambah pemegang kendaraan.

Tabel 6.64 Skenario Uji Tambah Pemegang Kendaraan Alternatif 1

Skenario Uji Tambah Pemegang Kendaraan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.30
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu daftar pemegang kendaraan. 3. Tidak memilih nama pemegang dan nomor polisi kendaraan pada <i>combo box</i>. 4. Klik tombol simpan.
<i>Expected Result</i>	Tidak bisa mengirimkan memperbarui pemeliharaan kendaraan serta melihatkan pesan <i>error</i> .
<i>Result</i>	Tidak bisa mengirimkan memperbarui pemeliharaan kendaraan serta melihatkan pesan <i>error</i> .
Valid /Tidak Valid	Valid

6.4.31 Pengujian Validasi Daftar Pemegang Kendaraan

Pada Tabel 6.65 yang adalah hasil pengujian persyaratan fungsional dari *use case* hapus kendaraan.

Tabel 6.65 Skenario Uji Daftar Pemegang Kendaraan

Skenario Uji Daftar Pemegang Kendaraan	
Kode Kebutuhan	SIMA-PFS.31
Skenario	1. Klik menu daftar pemegang kendaraan.
<i>Expected Result</i>	Berhasil lihat daftar pemegang kendaran.
<i>Result</i>	Berhasil lihat daftar pemegang kendaran.
Valid /Tidak Valid	Valid

6.4.32 Pengujian Validasi Hapus Kendaraan

Pada Tabel 6.66 yang adalah hasil pengujian persyaratan fungsional dari *use case* hapus kendaraan.

Tabel 6.66 Skenario Uji Hapus Kendaraan

Skenario Uji Hapus Kendaraan	
Kode Kebutuhan	SIMA-PFS.32
Skenario	1. Klik menu daftar pemegang kendaraan. 2. Klik lihat diatail sesuan dengan nama pemegang kendaraan. 3. Klik tombol hapus. 4. Muncul <i>pop-up alert</i> kemudian tekan oke.

**Tabel 6.66 Skenario Uji Hapus Kendaraan (Lanjutan)**

Skenario Uji Hapus Kendaraan	
Kode Kebutuhan	SIMA-PFS.32
<i>Expected Result</i>	Berhasil menghapus kendaraan dari daftar pemegang kendaraan.
<i>Result</i>	Berhasil menghapus kendaraan dari daftar pemegang kendaraan.
Valid /Tidak Valid	Valid

6.4.33 Pengujian Validasi Permintaan Laporan Pemeliharaan Kendaraan

Pada Tabel 6.67 adalah hasil pengujian persyaratan fungsional *use case* permintaan pembuatan laporan pemeliharaan kendaraan.

Tabel 6.67 Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan

Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.33
Skenario	1. Klik menu permintaan laporan. 2. Memilih laporan pemeliharaan kendaraan pada <i>combo box</i>. 3. Mengisikan <i>form</i> catatan. 4. Klik tombol <i>request</i> laporan.
<i>Expected Result</i>	Berhasil mengirimkan permintaan pembuatan laporan pemeliharaan kendaraan.
<i>Result</i>	Berhasil mengirimkan permintaan pembuatan laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

Pada Tabel 6.68 merupakan alternatif 1 dari *use case* permintaan pembuatan laporan pemeliharaan kendaraan.

**Tabel 6.68 Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan
Alternatif 1**

Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.33
Skenario	1. Klik menu permintaan laporan. 2. Tidak memilih laporan pemeliharaan kendaraan pada <i>combo box</i>. 3. Mengisikan <i>form</i> catatan. 4. Klik tombol <i>request</i> laporan.

Tabel 6.68 Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan Alternatif 1 (Lanjutan)

Skenario Uji Permintaan Laporan Pemeliharaan Kendaraan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.33
<i>Expected Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan pemeliharaan kendaraan dan menampilkan pesan <i>error</i> “The nama permintaan <i>field</i> is required”.
<i>Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan pemeliharaan kendaraan dan menampilkan pesan <i>error</i> “The nama permintaan <i>field</i> is required”.
Valid /Tidak Valid	Valid

6.4.34 Pengujian Validasi Daftar pemeliharaan Kendaraan

Pada Tabel 6.69 adalah hasil pengujian persyaratan fungsional *use case* daftar pemeliharaan kendaraan.

Tabel 6.69 Skenario Daftar Pemeliharaan Kendaraan

Skenario Uji Daftar Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.34
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu pelaporan pemeliharaan.
<i>Expected Result</i>	Berhasil lihat daftar pemeliharaan kendaraan.
<i>Result</i>	Berhasil lihat daftar pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.35 Pengujian Validasi Perbarui Pemeliharaan Kendaraan

Pada Tabel 6.70 adalah hasil pengujian persyaratan fungsional *use case* perbarui pemeliharaan kendaraan.

Tabel 6.70 Skenario Uji Perbarui Pemeliharaan Kendaraan

Skenario Uji Perbarui Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.35
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu <i>update</i> pemeliharaan. 3. Mengisikan <i>field</i> yang tersedia pada <i>form</i> pemeliharaan kendaraan. 4. Klik simpan.
<i>Expected Result</i>	Berhasil perbarui pemeliharaan kendaraan.
<i>Result</i>	Berhasil perbarui pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

Pada Tabel 6.71 merupakan alternatif 1 dari *use case* memperbarui pemeliharaan kendaraan.

Tabel 6.71 Skenario Uji Memperbarui Pemeliharaan Kendaraan Alternatif 1

Skenario Uji Memperbarui Pemeliharaan Kendaraan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.35
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu <i>update</i> pemeliharaan. 3. Mengisikan memilih <i>combo box</i> kondisi kendaraan. 4. Klik tombol simpan.
<i>Expected Result</i>	Tidak bisa mengirimkan memperbarui pemeliharaan kendaraan dan menampilkan pesan <i>error</i> "The kondisi kendaraan <i>field is required</i> ".
<i>Result</i>	Tidak bisa mengirimkan memperbarui pemeliharaan kendaraan dan menampilkan pesan <i>error</i> "The kondisi kendaraan <i>field is required</i> ".
Valid /Tidak Valid	Valid

6.4.36 Pengujian Validasi Hapus Hasil Pemeliharaan Kendaraan

Pada Tabel 6.72 yang adalah hasil pengujian persyaratan fungsional *use case* hapus hasil pemeliharaan kendaraan.

Tabel 6.72 Skenario Hapus Hasil Pemeliharaan Kendaraan

Skenario Uji Hapus Hasil Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.36
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu pelaporan pemeliharaan. 3. Klik hapus pada hasil pemeliharaan yang akan dihapus 4. Klik oke.
<i>Expected Result</i>	Berhasil menghapus hasil pemeliharaan kendaraan.
<i>Result</i>	Berhasil menghapus hasil pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.37 Pengujian Validasi Laporan Pemeliharaan Kendaraan

Pada Tabel 6.73 adalah hasil pengujian persyaratan fungsional *use case* laporan pemeliharaan persediaan.

**Tabel 6.73 Skenario Uji Laporan Pemeliharaan Kendaraan**

Skenario Uji Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.37
Skenario	1. Klik menu Pemeliharaan Kendaraan. 2. Klik sub menu pelaporan pemeliharaan. 3. Klik tombol Laporkan. 4. Klik tombol laporkan! pada <i>pop-up alert</i>.
<i>Expected Result</i>	Berhasil mengirimkan laporan pemeliharaan kendaraan.
<i>Result</i>	Berhasil mengirimkan laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.38 Pengujian Validasi Daftar Validasi Pemeliharaan Kendaraan

Pada Tabel 6.74 adalah hasil pengujian persyaratan fungsional *use case* daftar validasi pemeliharaan kendaraan.

Tabel 6.74 Skenario Uji Daftar Validasi Pemeliharaan Kendaraan

Skenario Uji Daftar Validasi Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.38
Skenario	1. Klik menu daftar validasi. 2. Klik submenu pemeliharaan kendaraan.
<i>Expected Result</i>	Berhasil lihat daftar validasi laporan pemeliharaan kendaraan.
<i>Result</i>	Berhasil lihat daftar validasi laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.39 Pengujian Validasi Validasi Laporan Pemeliharaan Kendaraan

Pada Tabel 6.75 adalah hasil pengujian persyaratan fungsional *use case* validasi laporan pemeliharaan kendaraan.

Tabel 6.75 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan

Skenario Uji Validasi Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.39
Skenario	1. Klik menu daftar validasi. 2. Klik submenu pemeliharaan kendaraan. 3. Klik tombol lihat detail sesuai dengan nama pemegang kendaraan yang akan divalidasi. 4. Klik tombol validasi. 5. Klik tombol validasi! pada <i>pop-up alert</i>.



Tabel 6.75 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan (Lanjutan)

Skenario Uji Validasi Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.39
<i>Expected Result</i>	Berhasil melakukan validasi laporan pemeliharaan kendaraan.
<i>Result</i>	Berhasil melakukan validasi laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

Pada Tabel 6.76 merupakan alternatif 1 dari *use case* validasi laporan pemeliharaan kendaraan.

Tabel 6.76 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 1

Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.39
Skenario	1. Klik menu daftar validasi. 2. Klik submenu pemeliharaan kendaraan. 3. Klik tombol lihat detail sesuai dengan nama pengguna pemegang kendaraan yang akan divalidasi. 4. Menandai <i>checkbox</i> dan klik tombol tolak validasi. 5. Mengisikan catatan dan menekan tombol simpan pada <i>pop-up alert</i>.
<i>Expected Result</i>	Berhasil melakukan tolak validasi laporan pemeliharaan kendaraan.
<i>Result</i>	Berhasil melakukan tolak validasi laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

Pada Tabel 6.77 merupakan alternatif 2 dari *use case* validasi laporan pemeliharaan kendaraan.

Tabel 6.77 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 2

Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.39
Skenario	1. Klik menu daftar validasi. 2. Klik submenu pemeliharaan kendaraan. 3. Klik tombol lihat detail sesuai dengan nama pengguna pemegang kendaraan yang akan divalidasi.

Tabel 6.77 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 2 (Lanjutan)

Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.39
	4. Tidak menandai <i>checkbox</i> dan klik tombol tolak validasi. 5. Mengisikan catatan dan menekan tombol simpan pada <i>pop-up alert</i> .
<i>Expected Result</i>	Tidak dapat melakukan tolak validasi laporan pemeliharaan kendaraan.
<i>Result</i>	Tidak dapat melakukan tolak validasi laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

Pada Tabel 6.78 merupakan alternatif 3 dari *use case* validasi laporan pemeliharaan kendaraan.

Tabel 6.78 Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 3

Skenario Uji Validasi Laporan Pemeliharaan Kendaraan Alternatif 3	
Kode Kebutuhan	SIMA-PFS.39
Skenario	1. Klik menu daftar validasi. 2. Klik submenu pemeliharaan kendaraan. 3. Klik tombol lihat detail sesuai dengan nama pengguna pemegang kendaraan yang akan divalidasi. 4. Tidak menandai <i>checkbox</i> dan klik tombol validasi.
<i>Expected Result</i>	Tidak dapat melakukan validasi laporan pemeliharaan kendaraan.
<i>Result</i>	Tidak dapat melakukan validasi laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.40 Pengujian Validasi Daftar Hasil Pemeliharaan Kendaraan

Pada Tabel 6.79 yang adalah hasil pengujian persyaratan fungsional dari *use case* daftar hasil pemeliharaan kendaraan.

Tabel 6.79 Skenario Uji Daftar Hasil Pemeliharaan Kendaraan

Skenario Uji Daftar Hasil Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.40
Skenario	1. Klik menu pemeliharaan kendaraan. 2. Klik submenu daftar hasil pemeliharaan kendaraan.
<i>Expected Result</i>	Berhasil melihat daftar hasil pemeliharaan kendaraan.
<i>Result</i>	Berhasil melihat daftar hasil pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.41 Pengujian Validasi Export Laporan Pemeliharaan Kendaraan

Pada Tabel 6.80 yang adalah hasil pengujian persyaratan fungsional *use case* export laporan persediaan.

Tabel 6.80 Skenario Uji Export Laporan Pemeliharaan Kendaraan

Skenario Uji Export Laporan Pemeliharaan Kendaraan	
Kode Kebutuhan	SIMA-PFS.41
Skenario	1. Pastikan laporan pemeliharaan kendaraan sudah selesai divalidasi. 2. Klik menu pemeliharaan. 3. Klik submenu pelaporan pemeliharaan. 4. Klik tombol <i>export pdf</i> ketika laporan pemeliharaan sudah selesai divalidasi.
<i>Expected Result</i>	Berhasil meng-export laporan pemeliharaan kendaraan.
<i>Result</i>	Berhasil meng-export laporan pemeliharaan kendaraan.
Valid /Tidak Valid	Valid

6.4.42 Pengujian Validasi Daftar Pengecekan Inventaris Ruangan

Pada Tabel 6.81 yang adalah hasil pengujian persyaratan fungsional *use case* daftar pengecekan inventaris ruangan.

Tabel 6.81 Skenario Uji Daftar Pengecekan Inventaris Ruangan

Skenario Uji Daftar Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.42
Skenario	1. Klik menu pengecekan. 2. Klik submenu daftar Pengecekan.
<i>Expected Result</i>	Berhasil lihat daftar pengecekan inventaris ruangan.

**Tabel 6.81 Skenario Uji Daftar Pengecekan Inventaris Ruangan (Lanjutan)**

Skenario Uji Daftar Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.42
Result	Berhasil lihat daftar pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

6.4.43 Pengujian Validasi Tambah Aset Pada Inventaris Ruangan

Pada Tabel 6.82 yang adalah hasil pengujian persyaratan fungsional dari *use case* tambah aset pada inventaris ruangan.

Tabel 6.82 Skenario Uji Tambah Aset Pada Inventaris Ruangan

Skenario Uji Tambah Aset Pada Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.43
Skenario	1. Klik menu pengecekan. 2. Klik submenu daftar pengecekan. 3. Mengisikan isian sesuai dengan <i>field</i> yang tersedia. 4. Klik simpan.
Expected Result	Berhasil menambahkan aset pada inventaris ruangan.
Result	Berhasil menambahkan aset pada inventaris ruangan.
Valid /Tidak Valid	Valid

6.4.44 Pengujian Validasi Hapus Aset Pada Inventaris Ruangan

Pada Tabel 6.83 yang adalah hasil pengujian persyaratan fungsional dari *use case* hapus aset pada inventaris ruangan.

Tabel 6.83 Skenario Uji Hapus Aset Pada Inventaris Ruangan

Skenario Uji Hapus Aset Pada Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.44
Skenario	1. Klik menu pengecekan. 2. Klik submenu daftar pengecekan. 3. Klik tombol lihat detail sesuai dengan ruangan pengecekan aset untuk menghapus aset dari daftar pengecekan. 4. Klik hapus pada data yang akan dihapus. 5. Muncul <i>alert</i> dan klik tombol oke.
Expected Result	Berhasil menghapus aset dari daftar pengecekan inventaris ruangan.

**Tabel 6.83 Skenario Uji Hapus Aset Pada Inventaris Ruangan (Lanjutan)**

Skenario Uji Hapus Aset Pada Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.44
Result	Berhasil menghapus aset dari daftar pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

6.4.45 Pengujian Validasi Permintaan Laporan Pengecekan Inventaris Ruangan

Pada Tabel 6.84 adalah hasil pengujian persyaratan fungsional *use case* permintaan pembuatan laporan pengecekan inventaris ruangan.

Tabel 6.84 Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan

Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.45
Skenario	1. Klik menu permintaan laporan. 2. Memilih laporan pengecekan aset pada <i>combo box</i>. 3. Mengisikan <i>form</i> catatan. 4. Klik tombol <i>request</i> laporan.
Expected Result	Berhasil mengirimkan permintaan pembuatan laporan pengecekan inventaris ruangan.
Result	Berhasil mengirimkan permintaan pembuatan laporan pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

Pada Tabel 6.85 merupakan alternatif 1 dari *use case* permintaan pembuatan laporan pengecekan aset.

Tabel 6.85 Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan Alternatif 1

Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.45
Skenario	1. Klik menu permintaan laporan. 2. Tidak memilih laporan pengecekan aset pada <i>combo box</i>. 3. Mengisikan <i>form</i> catatan. 4. Klik tombol <i>request</i> laporan.

Tabel 6.85 Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan Alternatif 1 (Lanjutan)

Skenario Uji Permintaan Laporan Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.45
<i>Expected Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan pengecekan aset dan menampilkan pesan error " <i>The nama permintaan field is required</i> ".
<i>Result</i>	Tidak bisa mengirimkan permintaan pembuatan laporan pengecekan aset dan menampilkan pesan error " <i>The nama permintaan field is required</i> ".
Valid /Tidak Valid	Valid.

6.4.46 Pengujian Validasi Perbarui Data Inventaris Ruangan

Pada Tabel 6.86 adalah hasil pengujian persyaratan fungsional dari *use case* perbarui data inventaris ruangan.

Tabel 6.86 Skenario Uji Perbarui Data Inventaris Ruangan

Skenario Uji Perbarui Data Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.46
Skenario	1. Klik menu pengecekan. 2. Klik submenu daftar pengecekan. 3. Klik lihat detail pada ruangan yang akan diperbarui pengecekannya. 4. Mengisikan semua <i>field</i> yang tersedia pada laman pengecekan aset. 5. Klik simpan.
<i>Expected Result</i>	Berhasil perbarui pengecekan aset.
<i>Result</i>	Berhasil perbarui pengecekan aset.
Valid /Tidak Valid	Valid

Pada Tabel 6.87 merupakan alternatif 1 dari *use case* memperbarui pengecekan aset.

Tabel 6.87 Skenario Uji Memperbarui Pengecekan Inventaris Ruangan Alternatif 1

Skenario Uji Memperbarui Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.46
Skenario	1. Klik menu pengecekan. 2. Klik submenu daftar pengecekan.

Tabel 6.87 Skenario Uji Memperbarui Pengecekan Inventaris Ruangan Alternatif 1 (Lanjutan)

Skenario Uji Memperbarui Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.46
	3. Klik lihat detail pada ruangan yang akan diperbarui pengecekannya. 4. Tidak mengisikan <i>field</i> yang tersedia pada laman pengecekan aset. 5. Klik tombol simpan.
<i>Expected Result</i>	Tidak menyimpan hasil dari pengecekan aset dan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
<i>Result</i>	Tidak menyimpan hasil dari pengecekan aset dan menampilkan pesan <i>error</i> sesuai dengan <i>field</i> kosong.
Valid /Tidak Valid	Valid

6.4.47 Pengujian Validasi Daftar Hasil Pengecekan Inventaris Ruangan

Pada Tabel 6.88 yang adalah hasil pengujian persyaratan fungsional *use case* daftar hasil pengecekan inventaris ruangan.

Tabel 6.88 Skenario Uji Daftar Hasil Pengecekan Inventaris Ruangan

Skenario Uji Daftar hasil Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.47
Skenario	1. Klik menu pengecekan. 2. Klik sub menu pelaporan pengecekan.
<i>Expected Result</i>	Berhasil lihat daftar hasil laporan pengecekan aset.
<i>Result</i>	Berhasil lihat daftar hasil laporan pengecekan aset.
Valid /Tidak Valid	Valid

6.4.48 Pengujian Validasi Laporan Pengecekan Inventaris Ruangan

Pada Tabel 6.89 adalah hasil pengujian persyaratan fungsional *use case* laporan pengecekan inventaris ruangan.

Tabel 6.89 Skenario Uji Laporan Pengecekan Inventaris Ruangan

Skenario Uji Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.48
Skenario	1. Klik menu pengecekan. 2. Klik sub menu pelaporan pengecekan. 3. Klik tombol lihat detail pada ruangan yang akan dilaporkan hasil pengecekannya. 4. Klik tombol Laporkan. 6. Klik tombol laporkan! pada <i>pop-up alert</i> .

**Tabel 6.89 Skenario Uji Laporan Pengecekan Inventaris Ruangan (Lanjutan)**

Skenario Uji Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.48
<i>Expected Result</i>	Berhasil mengirimkan laporan pengecekan aset.
<i>Result</i>	Berhasil mengirimkan laporan pengecekan aset.
Valid /Tidak Valid	Valid

6.4.49 Pengujian Validasi Sunting Hasil Pengecekan Inventaris Ruangan

Pada Tabel 6.90 adalah hasil pengujian persyaratan fungsional *use case* sunting hasil pengecekan inventaris ruangan.

Tabel 6.90 Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan

Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.49
Skenario	1. Klik menu pengecekan. 2. Klik sub menu pelaporan pengecekan. 3. Klik tombol lihat detail pada ruangan yang akan dilaporkan hasil pengecekannya. 4. Klik tombol sunting pada data pengecekan yang akan diperbarui. 5. Mengisikan sesuai dengan <i>field</i> yang tersedia. 6. Klik tombol “simpan”.
<i>Expected Result</i>	Berhasil memperbarui hasil pengecekan inventaris ruangan.
<i>Result</i>	Berhasil memperbarui hasil pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

Pada Tabel 6.91 merupakan alternatif 1 dari *use case* sunting hasil pengecekan inventaris ruangan.

Tabel 6.91 Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan Alternatif 1

Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.49
Skenario	1. Klik menu pengecekan. 2. Klik sub menu pelaporan pengecekan. 3. Klik tombol lihat detail pada ruangan yang akan dilaporkan hasil pengecekannya. 4. Klik tombol sunting pada data pengecekan yang akan diperbarui. 5. Tidak mengisikan <i>field</i> sesuai dengan <i>field</i> yang tersedia. 6. Klik tombol “simpan”.



Tabel 6.91 Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan Alternatif 1 (Lanjutan)

Skenario Uji Sunting Hasil Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.49
<i>Expected Result</i>	Gagal memperbarui hasil pengecekan inventaris ruangan.
<i>Result</i>	Gagal memperbarui hasil pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

6.4.50 Pengujian Validasi Daftar Validasi Pengecekan Inventaris Ruangan

Pada Tabel 6.92 adalah hasil pengujian persyaratan fungsional dari *use case* validasi laporan pengecekan inventaris ruangan.

Tabel 6.92 Skenario Uji Daftar Validasi Pengecekan Inventaris Ruangan

Skenario Uji Daftar Validasi Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.50
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan aset.
<i>Expected Result</i>	Berhasil lihat daftar validasi pengecekan inventaris ruangan.
<i>Result</i>	Berhasil lihat daftar validasi pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

6.4.51 Pengujian Validasi Validasi Laporan Pengecekan Inventaris Ruangan

Pada Tabel 6.93 yang adalah hasil pengujian persyaratan fungsional dari *use case* validasi laporan pengecekan inventaris ruangan.

Tabel 6.93 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan

Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.51
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan aset. 3. Klik tombol lihat detail sesuai dengan nama ruangan yang akan divalidasi. 4. Menandai <i>checkbox</i> dan klik tombol validasi. 5. Klik tombol validasi! pada <i>pop-up alert</i> .
<i>Expected Result</i>	Berhasil melakukan validasi laporan pengecekan inventaris ruangan.



**Tabel 6.93 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan
(Lanjutan)**

Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.51
Result	Berhasil melakukan validasi laporan pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

Pada Tabel 6.94 merupakan alternatif 1 dari *use case* validasi laporan pengecekan inventaris ruangan.

**Tabel 6.94 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan
Alternatif 1**

Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.51
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan aset. 3. Klik tombol lihat detail sesuai dengan nama ruangan yang akan divalidasi. 4. Menandai <i>checkbox</i> dan klik tombol tolak validasi. 5. Mengisikan alasan tolak validasi. 6. Klik tombol simpan pada <i>pop-up alert</i>.
Expected Result	Berhasil melakukan tolak validasi laporan pengecekan inventaris ruangan.
Result	Berhasil melakukan tolak validasi laporan pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

Pada Tabel 6.95 merupakan alternatif 2 dari *use case* validasi laporan pengecekan inventaris ruangan.

**Tabel 6.95 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan
Alternatif 2**

Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.51
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan aset. 3. Klik tombol lihat detail sesuai dengan nama ruangan yang akan divalidasi. 4. Tidak menandai <i>checkbox</i> dan klik tombol tolak validasi. 5. Mengisikan alasan tolak validasi. 7. Klik tombol simpan pada <i>pop-up alert</i>.



Tabel 6.95 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 2 (Lanjutan)

Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.51
<i>Expected Result</i>	Tidak dapat melakukan tolak validasi laporan pengecekan inventaris ruangan.
<i>Result</i>	Tidak dapat melakukan tolak validasi laporan pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

Pada Tabel 6.78 merupakan alternatif 3 dari *use case* validasi laporan pengecekan inventaris ruangan.

Tabel 6.96 Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif 3

Skenario Uji Validasi Laporan Pengecekan Inventaris Ruangan Alternatif	
Kode Kebutuhan	SIMA-PFS.51
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan aset. 3. Klik tombol lihat detail sesuai dengan nama ruangan yang akan divalidasi. 4. Tidak menandai <i>checkbox</i> dan klik tombol validasi. 5. Mengisikan alasan tolak validasi. 6. Klik tombol simpan pada <i>pop-up alert</i>.
<i>Expected Result</i>	Tidak dapat melakukan validasi laporan pengecekan inventaris ruangan.
<i>Result</i>	Tidak dapat melakukan validasi laporan pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

6.4.52 Pengujian Validasi *Export* Hasil Pengecekan Inventaris Ruangan

Pada Tabel 6.97 yang adalah hasil pengujian persyaratan fungsional *use case export* laporan pengecekan inventaris ruangan.

Tabel 6.97 Skenario Uji *Export* Hasil Pengecekan Inventaris Ruangan

Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.52
Skenario	1. Pastikan hasil pengecekan inventaris sudah selesai divalidasi. 2. Klik menu daftar validasi. 3. Klik submenu hasil pengecekan. 4. Mengisikan semester, penanggung jawab ruangan dan tahun kemudian klik <i>export</i>.

**Tabel 6.97 Skenario Uji *Export* Hasil Pengecekan Inventaris Ruangan (Lanjutan)**

Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan	
Kode Kebutuhan	SIMA-PFS.52
<i>Expected Result</i>	Berhasil meng- <i>export</i> hasil laporan pengecekan inventaris ruangan.
<i>Result</i>	Berhasil meng- <i>export</i> hasil laporan pengecekan inventaris ruangan.
Valid /Tidak Valid	Valid

Pada Tabel 6.98 merupakan alternatif 1 dari *use case export* laporan pengecekan inventaris ruangan.

Tabel 6.98 Skenario Uji *Export* Hasil Pengecekan Inventaris Ruangan Alternatif 1

Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan Alternatif 1	
Kode Kebutuhan	SIMA-PFS.52
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan. 3. Tidak mengisikan semester, penanggung jawab ruangan dan tahun kemudian klik <i>export</i>.
<i>Expected Result</i>	Tidak dapat meng- <i>export</i> hasil laporan pengecekan inventaris ruangan dan melihat pesan <i>error</i> sesuai <i>field</i> yang tidak terisi.
<i>Result</i>	Tidak dapat meng- <i>export</i> hasil laporan pengecekan inventaris ruangan dan melihat pesan <i>error</i> sesuai <i>field</i> yang tidak terisi.
Valid /Tidak Valid	Valid

Pada Tabel 6.99 merupakan alternatif 2 dari *use case export* laporan pengecekan inventaris ruangan.

Tabel 6.99 Skenario Uji *Export* Hasil Pengecekan Inventaris Ruangan Alternatif 2

Skenario Uji <i>Export</i> Hasil Pengecekan Inventaris Ruangan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.52
Skenario	1. Klik menu daftar validasi. 2. Klik submenu hasil pengecekan. 3. Tidak mengisikan salah satu dari semester, penanggung jawab ruangan atau tahun kemudian klik <i>export</i>.

Tabel 6.99 Skenario Uji Export Hasil Pengecekan Inventaris Ruangan Alternatif 2 (Lanjutan)

Skenario Uji Export Hasil Pengecekan Inventaris Ruangan Alternatif 2	
Kode Kebutuhan	SIMA-PFS.52
<i>Expected Result</i>	Tidak dapat meng-export hasil laporan pengecekan inventaris ruangan dan melihat pesan <i>error</i> sesuai <i>field</i> yang tidak terisi.
<i>Result</i>	Tidak dapat meng-export hasil laporan pengecekan inventaris ruangan dan melihat pesan <i>error</i> sesuai <i>field</i> yang tidak terisi.
Valid /Tidak Valid	Valid

6.4.53 Pengujian Validasi Lihat Daftar Aset

Pada Tabel 6.100 yang adalah hasil pengujian persyaratan fungsional dari *use case* lihat daftar aset.

Tabel 6.100 Skenario Uji Lihat Daftar Aset

Skenario Uji Lihat Daftar Aset	
Kode Kebutuhan	SIMA-PFS.53
Skenario	1. Mengakses SIMA. 2. Klik aset sesuai dengan kategori.
<i>Expected Result</i>	Berhasil melihat daftar aset.
<i>Result</i>	Berhasil melihat daftar aset.
Valid /Tidak Valid	Valid

6.4.54 Pengujian Validasi Lihat Daftar Persediaan

Pada Tabel 6.101 yang adalah hasil pengujian persyaratan fungsional *use case* lihat daftar persediaan.

Tabel 6.101 Skenario Uji Lihat Daftar Persediaan

Skenario Uji Lihat Daftar Persediaan	
Kode Kebutuhan	SIMA-PFS.54
Skenario	1. Mengakses SIMA. 2. Klik persediaan pada <i>navbar menu</i> .
<i>Expected Result</i>	Berhasil melihat daftar persediaan.
<i>Result</i>	Berhasil melihat daftar persediaan.
Valid /Tidak Valid	Valid

6.4.55 Pengujian Validasi Lihat Daftar Pemegang Kendaraan

Pada Tabel 6.102 yang hasil pengujian persyaratan fungsional *use case* lihat daftar pemegang kendaraan.

Tabel 6.102 Skenario Uji Lihat Daftar Pemegang Kendaraan

Skenario Uji Lihat Daftar Pemegang kendaraan	
Kode Kebutuhan	SIMA-PFS.55
Skenario	1. Mengakses SIMA. 2. Klik kendaraan pada <i>navbar menu</i> .
<i>Expected Result</i>	Berhasil melihat daftar pemegang kendaraan.
<i>Result</i>	Berhasil melihat daftar pemegang kendaraan.
Valid /Tidak Valid	Valid

6.4.56 Pengujian Validasi Logout

Pada Tabel 6.103 yang hasil pengujian persyaratan fungsional *use case* *logout*.

Tabel 6.103 Skenario Uji Logout

Skenario Uji Logout	
Kode Kebutuhan	SIMA-PFS.56
Skenario	1. Klik tulisan <i>Logout</i> .
<i>Expected Result</i>	Pengguna berhasil <i>logout</i> .
<i>Result</i>	Pengguna berhasil <i>logout</i> .
Valid /Tidak Valid	Valid



BAB 7 PENUTUP

7.1 Kesimpulan

1. Berdasarkan hasil analisis yang dilakukan dalam pembangunan sistem manajemen aset menghasilkan 56 kebutuhan fungsional dan 1 kebutuhan non fungsional. Dari hasil analisis juga didapatkan sembilan aktor yang dapat mengakses sistem manajemen aset, yaitu pengguna, kepala dinas, sekretaris dinas, kepala subbagian umum dan kepegawaian, kepala subbagian keuangan dan perencanaan, staf pengurus barang, staf penyimpanan barang, pemegang kendaraan dan admin. Disini pengguna merupakan aktor yang belum valid, yang artinya tidak bisa mengakses fungsionalitas utama dari sistem. Sedangkan kepala dinas, sekretaris dinas, kepala subbagian umum dan kepegawaian, kepala subbagian keuangan dan perencanaan, staf pengurus barang, staf penyimpanan barang dan admin adalah aktor yang valid yang dapat mengakses fungsionalitas utama dari sistem. Hasil analisis juga akan dimodelkan ke dalam *use case diagram* dan *use case scenario*. Hasil analisis ini didapatkan dari wawancara dan observasi dengan staf dan kasubag dari Dinas Perikanan Kabupaten Tabanan.
2. Hasil perancangan sistem terdiri 41 *class diagram*, 9 kelas merupakan turunan dari *Controller* dan 19 turunan dari *Model*. Selain itu hasil perancangan juga menghasilkan pemodelan interaksi objek yang dimodelkan dalam *sequence diagram*, *Physical Data Model* yang berisikan tabel-tabel sebagai tempat untuk penyimpanan data, beberapa perancangan komponen dan beberapa perancangan antarmuka pengguna.
3. Hasil implementasi berdasarkan hasil perancangan sistem berupa Sistem Informasi Manajemen Aset (SIMA) berbasis web, menggunakan kerangka kerja Laravel 5.7 dan Bootstrap dengan menggunakan bahasa pemrograman PHP, HTML, CSS, dan juga *Javascript* memakai pola perancangan *Model-View-Controller*. Implementasi *database* menggunakan *MySQL* dan *web server* menggunakan *Apache*. Hasil implementasi menghasilkan artefak sistem yang didokumentasikan, basis data, komponen serta antarmuka pengguna.
4. Hasil pengujian validasi pada keseluruhan kasus uji yang didasarkan pada persyaratan kebutuhan mendapatkan nilai 100% valid. Pengujian unit dilakukan dengan menggunakan 3 sampel uji dengan metode *basis path testing*, menghasilkan nilai 100% valid untuk ketiga sampel uji. Sedangkan pengujian *compatibility* dilakukan dengan menggunakan *tool Sortsite* yang menunjukkan hasil 75% tingkat keberhasilan dengan sistem berjalan pada *firefox* versi 66 dan *Chrome* versi 74 dengan *critical issue* dikarenakan tidak aktifnya fitur https saat melakukan pengujian. Berjalan dengan *major issues* pada semua *platform* serta berjalan dengan *minor issues* pada *IE* versi 11, *firefox* versi 66 dan *safari* pada versi 12. Pengujian integrasi memakai *integration incremental top-down* dengan bantuan *stub*. Memakai 1 sampel



uji dari *class* PengecekanAsetController pada fungsi *updateRequestChecking()* dengan *class Model* pada fungsi *Update()* dan mendapatkan nilai valid.

7.2 Saran

Saran yang dapat diberikan penulis, SIMA yang dibangun hanya menggunakan data dari Dinas Perikanan dan mengimplementasikan dengan menggunakan web yang akan memberikan kemudahan dalam melakukan manajemen aset dan juga mempermudah dalam melakukan pemantauan aset pada Dinas Perikanan Kabupaten Tabanan. Kedepannya SIMA dapat diintegrasikan dengan sistem informasi lainnya yang saling berkaitan yang berada pada lingkup Pemerintah Kabupaten Tabanan. Serta ditambahkannya *feature* peminjaman aset guna untuk meminimalisir ketidaksesuaian data aset.



DAFTAR REFERENSI

- Akbar, R., Wijaya, I.K. 2008. Manajemen Aset Sebagai Upaya Pelestarian Bangunan Bersejarah Di Kota Bandung. *Jurnal Perencanaan Wilayah Dan Kota*. Vol 19 (1): 13-33.
- Brown, Laue, Keast, Montgomery. 2012. *AAMCoG: Guide To Intefrated Strategic Asset Management Version 2.0*. Brisbane: An Australian Government Initiative.
- Dennis, Wixom, Tegarden. 2015. *Systems Analysis & Design: An Object-Orented Approach with UML 5th Edition*. Danvers: John Wiley & Sons, Inc.
- Dinas Perikanan Dan Kelautan. 2016. Rencana Strategis Dinas Perikanan Dan Kelautan Tahun 2016-2021. Tabanan: Dinas Perikanan Dan Kelautan.
- Indrajit, Richardus Eko. 2007. *Manajemen Sistem Informasi Dan Teknologi Informasi*. Jakarta: PT Elex Media Komputindo.
- Indriyani, Ni Made. 2012. Analisis Sistem Informasi Online Booking 'Trobex Android'. *Jurnal Elektronik Ilmu Komputer Universitas Udayana*. Vol 1(2): 59-65.
- Iswahyudi, C., Susanti, E. 2015. *Membangun Aplikasi Web Statistik Untuk Manganalisa Trafik Web Menggunakan PHP Dan MYSQL*. Yogyakarta: Institut Sains dan Teknologi AKPRIND.
- Kurniawan, Tri Astoto. 2007. Pengujian Struktur Program Dengan Pengujian Jalur Dasar (*Basis Path Testing*): Teori Dan Aplikasi. *Jurnal EEECCIS*. Vol 1(1): 29-32.
- Kurniwan, Tri Astoto. 2018. Pemodelan Use Case (UML): Evaluasi Terhadap Beberapa Kesalahan Dalam Praktik. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 5, 77-86.
- Maharikoding, 2019, Arsitekru Laravel 5.2 di www.mahirkoding.com (akses 12 Juni 2019).
- Pambudi G.S, Sriyanto, Arvianto A. 2016. Rancang Bangun Sistem Informasi Manajemen Aset Berbasis Web Untuk Optimalisasi Penelusuran Aset Di Teknik Industri UNDIP. *Jurnal Teknik Industri*. Vol XI (3): 187-196.
- Parkar, Shinde, Gadade, Prasad M. 2016. Utilization of Laravel Framework for Development of Web Based Recruitment Tool. *Journal of Computer Engineering*. PP: 36-41.
- Permatasari, Angelina. 2015. Aset Manajemen Rumah Sakit Berbasis Web System. *Jurnal UNISSULA*. Vol 2 (1): 70-75.
- Pressman. Roger S. 2010. *Software Engineering: A Practitioners's Approach 7th Edition*. New York: McGraw-Hill Companies, Inc.
- Priyanto, Bayu Eko. 2011. Rancang Bangun Sistem Informasi Manajemen Untuk Pengelolaan Asset Retirement Di PT. Toyota Motor Manufacturing Indonesia



(TMMIN). *Skripsi*. Publikasi. Bogor: Institut Pertanian Bogor.

Rusli, Chendry, S.S., Inayatullah. 2017. Sistem Informasi Manajemen Aset Berbasis Web Pada The Pitsop. *Jurnal Sistem Informasi STMIK MDP Palembang*. Vol X: 1-10.

Saputri, Erni Indah. 2014. Sistem Informasi Pengelolaan Inventaris Berbasis Client-Server Pada PT. Cahaya Agung Cemerlang Semarang. *Jurnal Mahasiswa Sekolah Tinggi Elektronika Dan Komputer Semarang*. Vol. 1 No. 1.

Sommerville, Ian. 2011. *Software Engineering 9th Edition*. Boston: Pearson Education, Inc.

Weske, Mathias. 2007. *Business Process Management: Concepts, Languages, Architectures*. Germany: Springer.

Zulfiandri, Wardhani, D.C. 2015. Rancang Bangun Sistem Informasi Manajemen Aset Berbasis *Private Cloud* Studi Kasus: TVRI Nasional. *Jurnal Ilmiah SimanteC*. Vol 5 (1): 49-58.

LAMPIRAN A BUKTI FISIK

A.1 Kartu Inventaris Ruangan



KABUPATEN TABANAN
KARTU INVENTARIS RUANGAN
PER 31-Des-2016

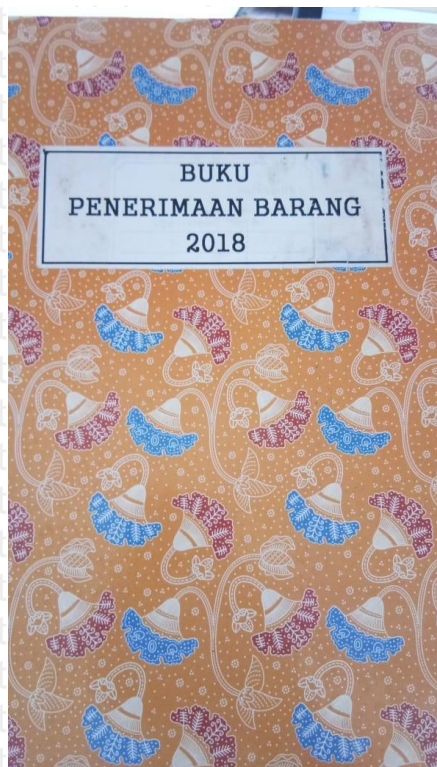
Provinsi : PROVINSI BALI
Kab./Kota : KABUPATEN TABANAN
Bidang : Bidang Perikanan
Unit Organisasi : DINAS PERIKANAN DAN KELAUTAN
Sub Unit Organisasi : DINAS PERIKANAN DAN KELAUTAN
U P B : DINAS PERIKANAN DAN KELAUTAN
Ruangan : KEPALA DINAS

No.	Jenis Barang / Nama Barang	Merk / Model	No. Seri Pabrik	Ukuran	Bahan	Tahun Pembelian/ Pembelian	No. Kode Barang	Jumlah Barang/ Register	Harga Beli/ Perolehan	Ket Baik (B)
1	Piring Besi/Metal	Brother			Besi	2006	02.06.01.04.01	1	1,400.00	1
2	Mesa Kayu/Rotan	-			Kayu	2003	02.06.02.01.01	1	6,000.00	1
3	Mesa Rapat	-			Kayu	2002	02.06.02.01.30	1	3,500.00	1
4	Mesa Tambahan	-			campuran	2002	02.06.02.01.38	1	1,350.00	1
5	Kursi Rapat	-			campuran	2002	02.06.02.01.27	6	7,560.00	6
6	Kursi Tamu	-			Kayu	2007	02.06.01.01.28	1	4,950.00	1
7	Kursi Ruler	-			campuran	2003	02.06.02.01.30	4	4,700.00	4
8	Mesa Komputer	-			campuran	2005	02.06.02.01.37	1	715.00	1
9	AC Unit	Panasonic			campuran	2003	02.06.02.04.03	1	3,157.00	1
10	Televisi	LCD LG 32"			campuran	2002	02.06.02.06.03	1	3,000.00	1
11	Lemari Arsip untuk arsip Dinamis	-			Kayu	2004	02.06.04.07.06	1	2,334.00	1
12	Projektor + Attachment	benq			campuran	2004	02.07.01.01.03	1	5,500.00	1
13	Projektor + Attachment	sony			campuran	2004	02.07.01.01.03	1	7,500.00	1
14	Microphone/Wireless Mic	tsa			campuran	2004	02.07.01.01.40	2	21,700.00	2
15	Handicam	Sony			campuran	2000	02.07.01.01.03	1	7,000.00	1
16	Microphone	tsa			campuran	2004	02.07.02.01.07	2	1,100.00	2
17	Recomile	HP Office jet 3608 all in			campuran	2003	02.07.02.01.30	1	1,287.00	1
Jumlah Harga									85,753.00	

MENGETAHUI
KEPALA DINAS PERIKANAN DAN

PENGURUS BARANG

A.2 Buku Penerimaan Barang





NO	Tanggal	Dept	Dokumen / SPK	Tanggal	Nama Barang	Jumlah	Harga Satuan (Rp)	Total Harga (Rp)	B. H. Perencanaan	Ket.
1.	2-1-2018	Ende								
2.	10-1-2018	Uth. Sijanti	523/43/Dit/1/2018	18-1-2018	1. Berharif 100 50.000 2. Map. 500 50.000 3. Vopar. 100 10.000	5 bh 25 bh 4 bh	1.100 27.200 4.200	349.000 27.500 42.000	523/43/Dit/1/2018	PAP Ganda 100 TH. 2018
3.	1-2-2018	Uth. Sijanti	523/24/Dit/1/2018	10-1-2018	1. Berharif 100 50.000 2. Berharif 20 2.000 3. Map. 100 10.000 4. Map. 100 10.000 5. Map. 100 10.000 6. Map. 100 10.000 7. Map. 100 10.000 8. Map. 100 10.000 9. Map. 100 10.000 10. Map. 100 10.000 11. Map. 100 10.000 12. Map. 100 10.000 13. Map. 100 10.000 14. Map. 100 10.000 15. Map. 100 10.000 16. Map. 100 10.000 17. Map. 100 10.000 18. Map. 100 10.000	62 bh 20 bh 27 bh 27 bh 1 bh 4 bh 11 bh 10 bh 3 bh 3 bh 3 bh 3 bh 3 bh 3 bh 3 bh 3 bh 3 bh 3 bh	79.700 1.100 27.200 27.200 1.100 4.900 11.900 10.800 3.400 3.400 3.400 3.400 3.400 3.400 3.400 3.400 3.400 3.400	4.981.900 3.200.000 4.800.000 132.500 612.500 38.200 38.200 34.300 112.500 105.000 7.500 67.800 67.800 218.800 10.317.100	523/43/Dit/1/2018	18-1-2018 PAP
4.	31-1-2018	Uth. Sijanti	523/087/Pagar/1/2018	30-1-2018	1. Berharif 100 50.000 2. Map. 100 10.000 3. Map. 100 10.000 4. Map. 100 10.000	2 bh 10 bh 10 bh 2 bh	79.700 1.100 4.900 27.200	159.400 16.500 48.000 378.300	523/087/Pagar/1/2018	PUGAR
5.	31-1-2018	Uth. Sijanti	523/59/Dit/1/2018	30-1-2018	8.000	79.700	637.600	523/59/Dit/1/2018	30-1-2018	Pagar

A.3 Buku Pengeluaran ATK





No	Tanggal	No. Urut	Nama Barang	Jumlah (Unit)	Harga Satuan (Rp)	Total Harga (Rp)	Unit Kerja Pemohon	Tanggal Diserahkan	Ket.
7	12-3-2018	1	Kertas HVS 60 gram	8 rim	79.700	637.600	Sekretariat	12-3-2018	PDD-Set
		2	Kertas HVS 30 gram	4 rim	58.100	232.400			
		3	Map Ordener Tekn	3 buah	22.100	54.300			
		4	Map folder biasa	16 buah	1.100	17.600			
		5	Map Stapleholder	16 buah	4.900	78.400			
		6	Spindel Stapler	3 Boks	4.500	13.500			
		7	Paper Clip	3 Boks	4.200	12.600			
						1.166.900			
8	4-4-2018	1	Map Ordener	15 buah	170.000	1.650.000	Sekretariat	10-4-2018	MAP
9	4-4-2018	1	Kertas HVS 60 gram	8 rim	79.700	637.600	Sekretariat	9-4-2018	BAP-Sekret
		2	Kertas HVS 30 gram	5 rim	88.100	440.500			
		3	Map Ordener Tekn	4 buah	27.200	108.800			
		4	Map folder biasa	14 buah	1.100	15.400			
		5	Map Stapleholder	14 buah	4.900	68.600			
		6	Spindel Stapler	1 buah	11.900	11.900			
		7							

A.4 Buku Pengeluaran Barang



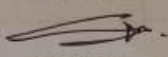
A.5 Surat Order Barang

E-6 Lampiran 1.

SURAT ORDER BARANG (SOB)

Unit Kerja Pemohon : Dinas Perikanan
Tanggal Permohonan : 2 Januari 2018

No.	Tanggal	Jenis Barang	Merak	Jumlah	Harga	Ket.
1	2	3	4	5	6	7
1	02/01/2018	Kertas HVS 50 gram		1 rim	79.700	Saldo ATK Th. 2017
2		Map folio biasa		5 buah	1.100	
3		Paper clip		1 kotak	4.200	

Pemohon,

A.A Ral Wirawan

A.7 Buku Pemeliharaan Kendaraan

No.	Date	Description	Amount
27.4	SP2D : 12737 / SP2D - 64 / 2018	(17.7.2018)	
	197 / SPJ - 64 / Diskan / 2018	(12.7.2018)	
	Belanja bahan bakar minyak / gas dan pelumas		
	biaya pembelian bahan bakar bulan juni		
	Rp 1.388.400 /		
28.4	SP2D : 12737 / SP2D - 64 / 2018	(17.7.2018)	
	187 / SPJ - 64 / Diskan / 2018	(6.7.2018)	
	Belanja pajak kendaraan bermotor berupa samsat		
	kendaraan roda 1 & roda 2 milik Dinas.		
	Rp 3.308.900 /		
29.4	SP2D : 13264 / SP2D - 64 / 2018	(26.7.2018)	
	205 / SRJ - 64 / Diskan / 2018	(18.7.2018)	
	Belanja bahan bakar minyak / gas dan pelumas berupa		
	pembelian bahan bakar bulan juni.		
	Rp 1.798.310 /		
30.4	SP2D : 12737 / SP2D - 64 / 2018	(17.7.2018)	
	183 / SPJ - 64 / Diskan / 2018	(6.7.2018)	
	Belanja pemeliharaan peralatan dan mesin.		
	- Attamedia > 1 Ketut Arta Wirawan		
	SPO : 523 / 378 / PAP / V / 2018		
	PA : 523 / 548 / PAP / V / 2018	(9.5.2018)	
		(2.7.2018)	

Never put off till tomorrow what you can do today

SINAR DUNIA



A.8 Laporan Kartu Inventaris Barang (KIB) A

KABUPATEN TABANAN
REKAPITULASI KARTU INVENTARIS BARANG (KIB) A
TANAH

Provinsi : PROVINSI GALE
Kab./Kota : KABUPATEN TABANAN
Bidang : Bidang Perikanan
Unit Organisasi : DINAS PERIKANAN
Sub Unit Organisasi : UPT Dinas Perikanan
NO. KODE LOKASI : 12.14.03.11.03.02.00

No.	Jenis Barang / Nama Barang	Nomor		Luas (M2)	Tahun Pengadaan	Letak/ Alamat	Status Tanah		Penggunaan	Asal usul	Harga (ribuan Rp)	Keterangan	
		Kode Barang	Regis- ter				Hak	Sertifikat					
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Tanah Bangunan Kantor Pengusaha	01.01.11.04.01	0001	11.025.00	2000	Siten Publik Nask. No. 33 Tabanan	Hak Pakai	1/3/2000	HP No 79/Csuh Pata	Bangunan Kantor dan Kibm	Pembelian	5.512.500,00	Di atas tanah beres. Bangunan kantor dan kibm
Jumlah Harga											5.512.500,00		

MENGEZAHUT
KEPALA DINAS PERIKANAN
I MADE SUKAWA S.P., MM
NIP. 19560304198031002

Tabanan, 30 Juni 2018
PENGURUS BARANG PENGGUNA
I MADE AGUS WIDJANA, S.P.
NIP. 199101072015031003



LAMPIRAN B WAWANCARA

Pertanyaan berikut digunakan untuk membantu penulis dalam mendapatkan data dan informasi mengenai kebutuhan dalam pembangunan sistem.

B.1 Kepala Subbagian Keuangan dan Perencanaan

Narasumber: Agung Raka Bhakta W, S.Pi., M.MA

- a. Bagaimana prosedur yang dilakukan oleh Dinas Perikanan Kabupaten Tabanan untuk pencatatan aset yang baru masuk?

Jawaban: ketika ada aset baru yang masuk dari kegiatan pengadaan, sebelum dilakukan pemeriksaan terhadap aset dengan melakukan validasi terhadap dokumen aset dengan kondisi aset yang diterima. Ketika sudah selesai dilakukan pemeriksaan akan diserahkan ke kasubag umum dan kepegawaian untuk melakukan koordinasi dengan pengurus barang untuk pengelolaan aset dan dan dimasukkan ke pelaporan periode selanjutnya.

- b. Apa saja pelaporan aset yang dibuat Dinas Perikanan Kabupaten Tabanan?

Jawaban: pada Dinas Perikanan Kabupaten Tabanan pelaporan yang dibuat antara lain pelaporan aset itu sendiri, pelaporan pengecekan inventaris ruangan, pelaporan persediaan dan juga ada pelaporan terhadap pemeliharaan kendaraan operasional ini dibeban kepada yang mendapatkan kendaraan.

- c. Bagaimana prosedur dalam pelaporan aset, persediaan, pengecekan inventaris ruangan, dan pemeliharaan kendaraan pada Dinas Perikanan Kabupaten Tabanan?

Jawaban: (1) Prosedur pada pelaporan aset itu pertama kepala dinas meminta sekretaris dinas untuk pembuatan laporan aset, kedua sekretaris dinas datang ke kasubag keuangan dan perencanaan melakukan koordinasi untuk penyusunan pelaporan aset, ketiga kasubag keuangan dan perencanaan datang ke staf pengurus barang menugaskan untuk mengumpulkan data dan membuat konsep laporan, keempat pengurus barang menyusun laporan berdasarkan konsep laporan yang sudah dibuat. Ketika laporan aset selesai disusun akan diberikan ke kepada kasubag keuangan dan perencanaan dan sekretaris dinas untuk dilakukan pemeriksaan laporan. Selanjutnya apabila laporan ada kesalahan yang diperiksa oleh sekretaris dinas akan dikembalikan ke kasubag keuangan dan perencanaan. Ketika laporan tidak sesuai pada pemeriksaan yang dilakukan kasubag keuangan dan perencanaan akan dikembalikan ke pengurus barang untuk diperbaiki. Apabila sesudah sesuai akan dicetak dan diberikan ke kepala dinas untuk disahkan. (2) pelaporan persediaan, pertama kepala dinas meminta pembuatan laporan persediaan ke kasubag umum dan kepegawaian, kedua kasubag umum dan kepegawaian menugaskan staf penyimpanan barang untuk membuat laporan persediaan, ketiga staf penyimpanan barang mengumpulkan data dan



membuatkan konsep laporan persediaan. Data semua sudah terkumpul dan konsep laporan sudah selesai maka selanjutnya membuat laporan persediaan berdasarkan konsep dan data yang sudah didapatkan. Laporan persediaan diserahkan ke kasubag umum dan kepegawaian untuk diperiksa dan benar maka penyimpanan barang mencetak dan memberikan kepada kepala dinas untuk disahkan. (3) pengecekan inventaris ruangan, pertama disini kasubag umum dan kepegawaian menugaskan pengurus barang melakukan pemeriksaan inventaris ruangan. Kedua pengurus barang melakukan pengecekan berdasarkan pada kartu inventaris ruangan yang ada sebelumnya dan menyesuaikan pada kondisi diruangan kemudian mencatat di *Microsoft excel* untuk dibuatkan kartu inventaris barang yang baru sesuai hasil dari pengecekan. Ketiga, setelah selesai melakukan pemeriksaan hasil pemeriksaan akan diberikan ke kasubag umum dan kepegawaian untuk diperiksa dan apabila tidak sesuai akan dikembalikan untuk di perbaiki dan apabila sudah sesuai akan akan dicetak diberikan ke kasubag umum dan kepegawaian untuk di sahkan. (4) pelaporan pemeliharaan kendaraan, pertama sekretaris dinas meminta pembuatan laporan pemeliharaan kendaraan kepada kasubag umum dan kepegawaian. Kedua kasubag umum dan kepegawaian menugaskan mengecek laik kendaraan operasional kepada staf pengurus barang. Ketiga pengurus barang meminta kepada yang memegang kendaraan dinas untuk melakukan pelaporan terhadap laik kendaraan mereka. Keempat pemegang kendaraan melakukan pengecekan laik kendaraan dan melaporkan kepada staf pengurus barang. Staf pengurus barang melakukan pemeriksaan apabila tidak sesuai akan dikembalikan dan apabila sudah sesuai akan diteruskan kepada kasubag umum dan kepegawaian untuk diperiksa dan ketika tidak sesuai akan dikembalikan kepada staf pengurus barang dan apabila sudah selesai akan dicetak dan diserahkan ke sekretaris dinas.

- d. Siapakah yang berwenang dalam pembuatan laporan baik itu laporan aset, persediaan, inventaris ruangan, dan pemeliharaan kendaraan?

Jawaban: Yang tadi saya jelaskan yang berwenang pada pembuatan laporan aset dan pengecekan inventaris ruangan pada staf pengurus barang, pada pembuatan laporan persediaan pada staf penyimpanan barang, dan pembuatan laporan pemeliharaan kendaraan pada pemegang kendaraan dinas.

- e. Dalam setahun berapa kali dilakukan kegiatan pelaporan?

Jawaban: Pembuatann laporan dilakukan setiap semester dengan kata lain tiap emam bulan sekali jadi dalam satu tahun ada dua laporan yang dibuat.

- f. Siapakah yang berwenang memberikan kendaraan dinas kepada pemegang kendaraan?

Jawaban: Yang melakukan staf pengurus barang karena staf pengurus barang yang melakukan pengelolaan terhadap aset pada Dinas Perikanan kabupaten Tabanan.



B.2 Staf Pengurus Barang

Narasumber: Ni Luh Gede Niriasih, S.E

- a. Anda sebagai staf pengurus barang, pelaporan apa saja yang anda kerjakan?

Jawaban: Saya sebagai staf pengurus barang disini mengerjakan pelaporan aset dan juga pengecekan inventaris ruangan.

- b. Dalam pelaporan aset, bagaimana cara mendapatkan data untuk pembuatan laporan aset?

Jawaban: Pengumpulan data aset dilakukan dengan meminta daftar aset pada tiap bidang pada Dinas Perikanan Kabupaten Tabanan. Setelah semua terkumpul kemudian membuat laporan pada *Microsoft excel* berdasarkan Kartu Inventaris Barang (KIB), yaitu KIB A yang berisikan aset tanah, KIB B berisikan aset peralatan dan mesin, KIB C berisikan aset gedung dan bangunan, dan KIB D berisikan jalan, jaringan dan juga irigasi.

- c. Pada saat melakukan pengecekan inventaris ruangan, kapan dilakukan pemeriksaan inventaris ruangan? Dan apa acuan dalam melakukan pemeriksaan inventaris ruangan?

Jawaban: Pengecekan inventaris ruangan dilakukan setiap semester dengan menggunakan acuan pada kartu inventaris ruangan yang sebelumnya dan membuat kartu inventaris barang yang baru pada *Microsoft excel* berdasarkan pengecekan inventaris ruangan yang telah dilakukan.

B.3 Staf Penyimpanan Barang

Narasumber: Ni Kade Suka Ariyanti, S.E

- a. Anda sebagai staf penyimpanan barang, pelaporan apa saja yang anda kerjakan?

Jawaban: Saya mengerjakan pelaporan persediaan.

- a. Dalam pelaporan persediaan, bagaimana cara mendapatkan data untuk pembuatan laporan persediaan?

Jawaban: Pembuatan laporan persediaan dibuat berdasarkan persediaan yang masuk dan persediaan yang keluar. Dilaporkan setiap semester.

- b. Bagaimana prosedur *order* persediaan sampai barang diambil oleh bidang terkait?

Jawaban: Pertama karyawan mendatangi staf penyimpanan barang, kedua penyimpanan barang mencatat persediaan yang diminta pada buku pengeluaran. Ketiga penyimpanan barang menerbitkan surat *order* barang dan juga surat perintah pengeluaran barang berdasarkan persediaan yang diminta. Keempat penyimpanan barang menyerahkan persediaan yang diminta kepada karyawan.