

48.519 / H / 13



RSSI

005-3

Wah

p-1

2012

TUGAS AKHIR - KS09 1336

**PENGEMBANGAN SISTEM INFORMASI  
PENGELOLAAN KEUANGAN NEGARA MODUL  
ADMINISTRASI MENGGUNAKAN TEKNOLOGI  
JAVA SERVER PAGES DAN POSTGRESQL**

WAHYUDIONO  
NRP 5209 108 718

Dosen Pembimbing  
Dr. Eng. Febriliyan Samopa, S.Kom., M.Kom  
Alwi, S.E

JURUSAN SISTEM INFORMASI  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2012

PERPUSTAKAAN ITS	
Tgl. Terima	6-8-2012
Terima Dari	H
No Agenda Psp.	-



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

FINAL PROJECT - KS09 1336

**DEVELOPMENT OF STATE FINANCIAL  
MANAGEMENT INFORMATION SYSTEM  
ADMINISTRATION MODULE USING JAVASERVER  
PAGES AND POSTGRESQLTECHNOLOGY**

WAHYUDIONO  
NRP 5209 108 718

Supervisor  
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom  
Alwi, S.E

INFORMATION SYSTEM DEPARTMENT  
Information Technology Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2012

**PENGEMBANGAN SISTEM INFORMASI  
PENGELOLAAN KEUANGAN NEGARA MODUL  
ADMINISTRASI MENGGUNAKAN TEKNOLOGI  
JAVASERVER PAGES DAN POSTGRESQL**

**TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**WAHYUDIONO**  
5209 108 718

Surabaya, 2012

**KETUA  
JURUSAN SISTEM INFORMASI**



**Dr.Eng. Febrilijan Samopa, S.Kom., M.Kom**  
NIP. 197302191998021001



**PENGEMBANGAN SISTEM INFORMASI DAN  
PENGELOLAAN KEUANGAN NEGARA MODUL  
ADMINISTRASI BERTEKNOLOGI JAVA DAN  
POSTGRESQL**

**TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**WAHYUDIONO**  
**5209 108 718**

Disetujui Tim Penguji :

Tanggal Ujian : 16 Juli 2012

Periode Wisuda: September 2012

**Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom** (Pembimbing I)

**Alwi, SE**

(Pembimbing II)

**Faizal Johan Atletiko, S.Kom, MT**

(Penguji I)

**Radityo Prasetyanto W, S.Kom, M.Kom**

(Penguji II)

**PENGEMBANGAN SISTEM INFORMASI  
PENGELOLAAN KEUANGAN NEGARA MODUL  
ADMINISTRASI MENGGUNAKAN TEKNOLOGI  
JAVA SERVER PAGES DAN POSTGRESQL**

**Nama Mahasiswa : Wahyudiono**  
**NRP : 5209 108 718**  
**Jurusan : Sistem Informasi FTIF-ITS**  
**Dosen Pembimbing : Dr. Eng. Febriliyan Samopa, S.Kom,  
M.Kom**

**Abstrak**

*Salah satu kewajiban Satuan Kerja (Satker) di ruang lingkup Direktorat Jenderal Perbendaharaan (DJPBN) adalah menatausahakan seluruh transaksi yang berkaitan dengan penggunaan dana APBN dalam rangka pelaksanaan anggaran satuan kerja. Salah satu tujuan dari penatausahaan transaksi tersebut adalah menghasilkan laporan keadaan kas dan realisasi anggaran yang akurat dan dapat dipertanggungjawabkan.*

*Direktorat Jenderal Perbendaharaan Negara telah meluncurkan berbagai jenis aplikasi dalam rangka penatausahaan transaksi satuan kerjanya (satker). Tujuannya adalah menghasilkan laporan keadaan kas dan realisasi anggaran yang akurat dan dapat dipertanggungjawabkan. Beberapa aplikasi yang telah tersedia saat inibelum terintegrasi antara satu dengan lainnya. Hal tersebut tentu saja menjadi sebuah penghambat bagi pencapaian efisiensi dan efektifitas pengelolaan data dan transaksi.*

*Pada pengerjaan tugas akhir ini dikembangkan sebuah Aplikasi Sistem Informasi Pengelolaan Keuangan Negara (SIAPKAN) yang terfokus pada Modul Administrasi. Dirancang dengan tujuan agar dapat mengintegrasikan semua aplikasi yang digunakan oleh satker. SIAPKAN merupakan aplikasi berbasis*

*Web berteknologi Java (Java Servlet dan JSP) dan sistem manajemen basis data PostgreSQL.*

*Dengan menggunakan bahasa pemrograman Java dan Database Management System (DBMS) PostgreSQL, modul Administrasi yang menjadi fokus utama tugas akhir sekaligus gerbang awal operasional aplikasi SIAPKAN. Melalui modul Administrasi inilah kebutuhan referensi tiap modul disediakan. Berbagai menu di dalam modul Administrasi antara lain Pengelolaan Referensi, Pengaturan pengguna, role pengguna, waktu kerja, Bantuan dan Backup data tersedia untuk kebutuhan Administrator selaku pengguna. Demikian kebutuhan pengelolaan data setiap modul dapat berjalan dengan baik.*

**Kata Kunci:** *Administrasi, DJPBN, Java, PostgreSQL, Satker, SIAPKAN*

**DEVELOPMENT OF STATE FINANCIAL  
MANAGEMENT INFORMATION SYSTEM  
ADMINISTRATION MODULE USING JAVA SERVER  
PAGES AND POSTGRESQL TECHNOLOGY**

**Name** : Wahyudiono  
**NRP** : 5209 108 718  
**Department** : Information System FTIF-ITS  
**Supervisor** : Dr. Eng. Febriliyan Samopa, S.Kom,  
M.Kom

**Abstract**

*One of the obligations of Public Instance Unit a.k.a **Satker** in the scope of the Directorate General of Treasury (DJPBN) is to administer all transactions relating to the use of state funds for implementation of the budget unit. One of the goals of the administration of the transaction is a cash generating reports and a state budget that accurate and reliable.*

*The State Treasury General Directorat (DJPBN) had already released various applications in addition to help their Satker in transaction administrating. The objectives are creating the accurate and realible cash and budget realization reports. Some of applications which are available today has not been integrated between one and another. Those become obstacles to reach efficient and effective data transaction and management.*

*On this Final Project, the Administration Module of the State Financial Management Information System (SIAPKAN) is developed. Designed to integrate all applications which existing now used by the Satker. SIAPKAN is a Web Based application using Java (Java Servlet and JSP) and PostgreSQL Database Management System technology.*

*By using Java programming language and Database Management System (DBMS) PostgreSQL, the Administration*

*Module which is the main focus of this Final Project has become the first gate on SIAPKAN application. Through the Administration Module the needs of others Module on SIAPKAN can be fulfilled. Various menu on the Administration Module there are user configurations, user roles, Working Time, Help, and Backup data menu are available for Administrator as a user. Thus the needs of data management for each module can be operated well.*

**Keywords:**Administration, DJPBN, Java, PostgreSQL, Satker, SIAPKAN



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah Subhana Wata'alla, Sang Pencipta Ilmu yang telah memberikan segala kesempatan dan kekuatan, sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul "Pengembangan Sistem Informasi Pengelolaan Keuangan Negara Modul Administrasi Menggunakan Teknologi Java Server Pages Dan PostgreSQL".

Tentunya tugas akhir ini sulit untuk diwujudkan tanpa dukungan, saran, dan bantuan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu penulis baik dalam pelaksanaan tugas akhir hingga selesainya penyusunan laporan tugas akhir ini. Ucapan terima kasih penulis sampaikan kepada:

- Bapak Febriliyan Samopa, selaku dosen pembimbing I , dan Bapak Alwi selaku dosen pembimbing II atas segala ketulusan membagi ilmu dalam pengerjaan tugas akhir dan kesediaan memberikan waktu di tengah kesibukan beliau.
- Istriku, Nidaan Khafian, yang selalu mendukung secara suka dan duka, baik material maupun moral.
- Anakku tercinta, Rasyad Hanifan Aozora Diono, yang selalu memberi inspirasi dan harapan.
- Ayah, ibu, dan kedua adik-adik saya yang selalu mendukung dalam suka dan duka.
- Abi dan Umi ku yang selalu siap mendukung dalam suka dan duka.
- Seluruh Tim pengembang aplikasi SIAPKAN, Suichal, Mas Ali, Alfie, Hari, Novan, dan Yoga yang selalu bersedia membantu proses penyelesaian.
- Direktorat Jenderal Perbendaharaan, khususnya pihak DSP dan DTP yang telah bersedia menjadi studi kasus Tugas Akhir dan berpartisipasi dengan memberikan segala informasi yang dibutuhkan,

- Bendits 2009 yang telah berbagi suka dan duka selama masa kuliah.
- Seluruh civitas akademik Jurusan Sistem Informasi ITS Surabaya baik para dosen, karyawan, dan mahasiswa yang telah bekerja sama baik dengan penulis.

Serta semua pihak yang tidak dapat disebutkan satu persatu yang telah ikut membantu baik secara langsung maupun tidak langsung selama penulisan tugas akhir ini. Semoga kebaikan yang telah diberikan dibalas oleh Allah SWT baik langsung maupun di akhirat kelak.

Tidak ada suatu karya manusia yang sempurna, demikian pula dengan tugas akhir ini. Untuk itu dalam kesempatan ini pula penulis mohon maaf atas segala ketidaksempurnaan yang ada. Kritik dan saran yang membangun dapat disampaikan melalui email [kuromoncha@gmail.com](mailto:kuromoncha@gmail.com). Harapan penulis, semoga tugas akhir ini dapat bermanfaat bagi pembaca.

Surabaya, Juli 2012

Penulis

## DAFTAR ISI

Abstrak .....	iii
Abstract .....	v
Kata Pengantar .....	vii
Daftar Isi .....	ix
Daftar Gambar .....	xxi
Daftar Tabel .....	xxvii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	5
1.3. Batasan Masalah .....	5
1.4. Tujuan .....	6
1.5. Relevansi atau Manfaat .....	6
1.6. Sistematika Penulisan .....	7
BAB II TINJAUAN PUSTAKA .....	9
2.1. Administrator .....	9
2.2. Satuan Kerja .....	10
2.2.1. Definisi Satker .....	10
2.2.2. Pengelompokan Satker .....	10
2.3. Java .....	11
2.4. Servlets .....	12
2.5. Java Server Pages .....	14
2.6. Postgre SQL .....	15
2.7. JDBC .....	16
2.8. Arsitektur Model Controller View (MVC) .....	16
BAB III METODOLOGI .....	19
3.1. Studi Literatur .....	19
3.2. Analisis Kebutuhan .....	19
3.3. Perancangan Aplikasi .....	19
3.4. Pembuatan Aplikasi/Coding .....	20
3.5. Integrasi Aplikasi .....	20
3.6. Evaluasi/Uji Coba .....	20
3.7. Migrasi Data .....	21

3.8. Dokumentasi Tugas Akhir .....	21
<b>BAB IV PERANCANGAN PERANGKAT LUNAK .....</b>	<b>23</b>
4.1. Analisis Kebutuhan .....	24
4.1.1. Pengguna .....	24
4.1.2. Kebutuhan fungsional/Proses .....	25
4.2. Arsitektur Teknologi .....	28
4.2.1. Lapisan presentasi .....	28
4.2.2. Lapisan web server / server aplikasi .....	29
4.2.3. Lapisan database .....	30
4.3. Hubungan Modul Administrasi dengan Modul Lain .....	30
4.4. Hubungan Antar Modul SIAPKAN .....	31
4.5. <i>Workflow</i> Aplikasi .....	35
4.6. Domain Model .....	37
4.7. <i>Manajemen Session</i> .....	40
4.7.1. Manajemen Session Pengguna .....	40
4.7.2. Manajemen Invoke Database .....	41
4.7.3. Manajemen Logout .....	41
4.7.4. Manajemen Session Atribut .....	41
4.8. GUI Story Board .....	42
4.9. Class Diagram .....	52
4.10. Use Case Diagram .....	58
4.10.1. Use Case level 1 modul Administrasi .....	59
4.10.2. Use Case level 2 modul Administrasi .....	60
4.11. Robustness Diagram .....	66
4.12. Sequence Diagram .....	66
4.13. Test Case .....	66
<b>BAB V IMPLEMENTASI DAN UJI COBA SISTEM .....</b>	<b>67</b>
5.1. Lingkungan Implementasi .....	67
5.2. Struktur Project Directory .....	68
5.3. File Konfigurasi .....	72
5.4. Program <i>Scripting</i> .....	73
5.5. Implementasi Tampilan .....	77
5.6. Pembuatan Aplikasi .....	82
5.6.1. Halaman Index/form login .....	82
5.6.2. Manajemen Login .....	83

5.6.3.	Implementasi Login.....	84
5.6.4.	Fungsi Validasi dan DatePicker dengan JavaScript	87
5.6.5.	Proses Tayang Data .....	89
5.6.6.	Proses Rekam Data.....	92
5.6.7.	Proses Edit Data .....	94
5.6.8.	Proses Pencarian Data .....	95
5.6.9.	Proses Backup Data.....	96
5.6.10.	Proses Logout .....	98
5.7.	Integrasi Modul .....	98
5.8.	Migrasi data .....	101
5.9.	Uji Coba dan Evaluasi.....	101
BAB VI KESIMPULAN DAN SARAN.....		109
6.1.	Kesimpulan .....	109
6.2.	Saran .....	110
DAFTAR PUSTAKA.....		111
Lampiran A DESKRIPSI USE CASE.....		A-1
A.1.	Deskripsi Use CaseProses Login.....	A-2
A.2.	Deskripsi Use Case Tayang Bendahara.....	A-3
A.3.	Deskripsi Use Case Rekam Bendahara.....	A-3
A.4.	Deskripsi Use Case UbahBendahara .....	A-4
A.5.	Deskripsi Use Case Hapus Bendahara.....	A-5
A.6.	Deskripsi Use Case Tayang Beban.....	A-5
A.7.	Deskripsi Use Case Rekam Beban .....	A-6
A.8.	Deskripsi Use Case Ubah Beban .....	A-7
A.9.	Deskripsi Use Case Hapus Beban.....	A-7
A.10.	Deskripsi Use Case Tayang Fungsi .....	A-8
A.11.	eskripsi Use Case Rekam Fungsi.....	A-9
A.12.	Deskripsi Use Case Ubah Fungsi .....	A-9
A.13.	Deskripsi Use Case Hapus Fungsi.....	A-10
A.14.	Deskripsi Use Case Tayang Jenis Dokumen .....	A-11
A.15.	Deskripsi Use Case Rekam Jenis Dokumen .....	A-11
A.16.	Deskripsi Use Case Ubah Jenis Dokumen.....	A-12
A.17.	Deskripsi Use Case Hapus Jenis Dokumen .....	A-13
A.18.	Deskripsi Use Case Tayang MAK.....	A-13

A.19. Deskripsi Use Case Rekam MAK .....	A-14
A.20. Deskripsi Use Case Ubah MAK .....	A-15
A.21. Deskripsi Use Case Hapus MAK .....	A-15
A.22. Deskripsi Use Case Tayang Jenis SPM.....	A-16
A.23. Deskripsi Use Case Rekam Jenis SPM.....	A-17
A.24. Deskripsi Use Case Ubah Jenis SPM .....	A-17
A.25. Deskripsi Use Case Hapus Jenis SPM.....	A-18
A.26. Deskripsi Use Case Tayang Cara Bayar.....	A-19
A.27. Deskripsi Use Case Rekam Cara Bayar.....	A-19
A.28. Deskripsi Use Case Ubah Cara Bayar .....	A-20
A.29. Deskripsi Use Case Hapus Cara Bayar.....	A-21
A.30. Deskripsi Use Case Tayang Jenis Dokumen .....	A-21
A.31. Deskripsi Use Case Rekam Jenis Dokumen .....	A-22
A.32. Deskripsi Use Case Ubah Jenis Dokumen.....	A-23
A.33. Deskripsi Use Case Hapus Jenis Dokumen .....	A-23
A.34. Deskripsi Use Case Tayang Sumber Dana .....	A-24
A.35. Deskripsi Use Case Rekam Sumber Dana.....	A-25
A.36. Deskripsi Use Case Ubah Sumber Dana .....	A-25
A.37. Deskripsi Use Case Hapus Sumber Dana.....	A-26
A.38. Deskripsi Use Case Tayang Jenis SPM.....	A-27
A.39. Deskripsi Use Case Rekam Jenis SPM.....	A-27
A.40. Deskripsi Use Case Ubah Jenis SPM .....	A-28
A.41. Deskripsi Use Case Hapus Jenis SPM.....	A-29
A.42. Deskripsi Use Case Tayang Aset Tetap.....	A-30
A.43. Deskripsi Use Case Rekam Aset Tetap .....	A-30
A.44. Deskripsi Use Case Ubah Aset Tetap .....	A-31
A.45. Deskripsi Use Case Hapus Aset Tetap .....	A-32
A.46. Deskripsi Use Case Tayang Cara Perolehan .....	A-32
A.47. Deskripsi Use Case Rekam Cara Perolehan .....	A-33
A.48. Deskripsi Use Case Ubah Cara Perolehan.....	A-34
A.49. Deskripsi Use Case Hapus Cara Perolehan .....	A-34
A.50. Deskripsi Use Case Tayang SSKel Barang .....	A-35
A.51. Deskripsi Use Case Rekam SSKel Barang.....	A-36
A.52. Deskripsi Use Case Ubah SSKel Barang.....	A-36
A.53. Deskripsi Use Case Hapus SSKel BMN .....	A-37

A.54. Deskripsi Use Case Tayang Map Barang .....	A-38
A.55. Deskripsi Use Case RekamMap Barang.....	A-38
A.56. Deskripsi Use Case UbahMap Barang .....	A-39
A.57. Deskripsi Use Case HapusMap Barang.....	A-40
A.58. Deskripsi Use Case Tayang Kanwil .....	A-41
A.59. Deskripsi Use Case RekamKanwil.....	A-41
A.60. Deskripsi Use Case UbahKanwil.....	A-42
A.61. Deskripsi Use Case HapusKanwil.....	A-43
A.62. Deskripsi Use Case Tayang Akun .....	A-43
A.63. Deskripsi Use Case RekamAkun.....	A-44
A.64. Deskripsi Use Case UbahAkun .....	A-45
A.65. Deskripsi Use Case HapusAkun.....	A-45
A.66. Deskripsi Use Case Tayang Blokir.....	A-46
A.67. Deskripsi Use Case RekamBlokir .....	A-46
A.68. Deskripsi Use Case UbahBlokir .....	A-47
A.69. Deskripsi Use Case HapusBlokir .....	A-48
A.70. Deskripsi Use Case Tayang Departemen .....	A-49
A.71. Deskripsi Use Case RekamDepartemen .....	A-49
A.72. Deskripsi Use Case UbahDepartemen.....	A-50
A.73. Deskripsi Use Case HapusDepartemen .....	A-51
A.74. Deskripsi Use Case Tayang GBKPK .....	A-51
A.75. Deskripsi Use Case RekamGBKPK .....	A-52
A.76. Deskripsi Use Case UbahGBKPK.....	A-53
A.77. Deskripsi Use Case HapusGBKPK .....	A-53
A.78. Deskripsi Use Case Tayang Kegiatan.....	A-54
A.79. Deskripsi Use Case RekamKegiatan .....	A-55
A.80. Deskripsi Use Case UbahKegiatan.....	A-55
A.81. Deskripsi Use Case HapusKegiatan .....	A-56
A.82. Deskripsi Use Case Tayang Unit.....	A-57
A.83. Deskripsi Use Case RekamUnit .....	A-57
A.84. Deskripsi Use Case UbahUnit .....	A-58
A.85. Deskripsi Use Case HapusUnit.....	A-59
A.86. Deskripsi Use Case Tayang IB .....	A-59
A.87. Deskripsi Use Case RekamIB.....	A-60
A.88. Deskripsi Use Case UbahIB .....	A-61

A.89. Deskripsi Use Case HapusIB.....	A-61
A.90. Deskripsi Use Case Tayang Output.....	A-62
A.91. Deskripsi Use Case RekamOutput.....	A-63
A.92. Deskripsi Use Case UbahOutput.....	A-63
A.93. Deskripsi Use Case HapusOutput.....	A-64
A.94. Deskripsi Use Case Tayang Profil Pengguna.....	A-65
A.95. Deskripsi Use Case RekamProfil Pengguna.....	A-65
A.96. Deskripsi Use Case UbahProfil Pengguna.....	A-66
A.97. Deskripsi Use Case HapusProfil Pengguna.....	A-67
A.98. Deskripsi Use Case Tayang Waktu Kerja.....	A-68
A.99. Deskripsi Use Case Rekam Waktu Kerja.....	A-68
A.100. Deskripsi Use Case Ubah Waktu Kerja.....	A-69
A.101. Deskripsi Use Case Hapus Waktu Kerja.....	A-70
A.102. Deskripsi Use Case Backup Data.....	A-70
A.103. Deskripsi Use Case Tayang Konten.....	A-71
A.104. Deskripsi Use Case Tayang Peraturan.....	A-71
A.105. Deskripsi Use Case RekamPeraturan.....	A-72
A.106. Deskripsi Use Case UbahPeraturan.....	A-73
A.107. Deskripsi Use Case HapusPeraturan.....	A-73
A.108. Deskripsi Use Case Tayang Tentang.....	A-74
A.109. Deskripsi Use Case Proses Logout.....	A-75
Lampiran B DIAGRAM ROBUSTNESS ANALYSIS.....	B-1
B.1. Diagram Robustness Rekam User.....	B-2
B.2. Diagram Robustness Rekam Bank.....	B-2
B.3. Diagram Robustness Hapus Bank.....	B-3
B.4. Diagram Robustness Rekam Dokumen.....	B-3
B.5. Diagram Robustness Hapus Dokumen.....	B-4
B.6. Diagram Robustness Rekam Kegiatan.....	B-5
B.7. Diagram Robustness Hapus Kegiatan.....	B-5
B.8. Diagram Robustness Rekam Lokasi.....	B-6
B.9. Diagram Robustness Hapus Lokasi.....	B-6
B.10. Diagram Robustness Rekam MAK.....	B-7
B.11. Diagram Robustness Hapus MAK.....	B-7
B.12. Diagram Robustness Rekam Output.....	B-8
B.13. Diagram Robustness Hapus Output.....	B-8



B.14. Diagram Robustness Rekam Pejabat .....	B-9
B.15. Diagram Robustness Hapus Pejabat .....	B-9
B.16. Diagram Robustness Rekam Program .....	B-10
B.17. Diagram Robustness HapusProgram .....	B-10
B.18. Diagram Robustness RekamJenis Kegiatan.....	B-11
B.19. Diagram Robustness HapusJenis Kegiatan.....	B-11
B.20. Diagram Robustness RekamSatuan Kerja .....	B-12
B.21. Diagram Robustness HapusSatuan Kerja .....	B-12
Lampiran C Sequence Diagram.....	C-1
C.1. Diagram Sequence Login.....	C-2
C.2. Diagram Sequence Pengelolaan Beban .....	C-3
C.3. Diagram Sequence Rekam Bank .....	C-3
C.4. Diagram Sequence Rekam Dokumen .....	C-4
C.5. <i>Diagram Sequence Rekam Kegiatan</i> .....	C-4
C.6. Diagram Sequence Rekam Lokasi .....	C-5
C.7. Diagram Sequence Rekam MAK.....	C-6
C.8. Diagram Sequence Rekam Output.....	C-6
C.9. Diagram Sequence Rekam Pejabat .....	C-7
C.10. Diagram Sequence Rekam Program .....	C-7
C.11. Diagram Sequence Rekam Satker.....	C-8
C.12. Diagram Sequence Ubah Bank .....	C-8
C.13. Diagram Sequence Ubah Dokumen.....	C-9
C.14. Diagram Sequence Ubah Kegiatan .....	C-9
C.15. Diagram Sequence Ubah Lokasi.....	C-10
C.16. Diagram Sequence Ubah MAK .....	C-11
C.17. Diagram Sequence Ubah Output .....	C-12
C.18. Diagram Sequence Ubah Pejabat.....	C-13
C.19. Diagram Sequence Ubah Program.....	C-14
C.20. Diagram Sequence Ubah Satker .....	C-15
C.21. Diagram Sequence Ubah Pengguna.....	C-16
C.22. Diagram Sequence Logout.....	C-16
Lampiran D.....	D-1
Test Case .....	D-1
D.1. Test CaseLogin .....	D-2
D.2. Test Case Rekam Satker .....	D-3

D.3. Test Case Ubah Satker.....	D-4
D.4. Test Case Hapus Satker .....	D-6
D.5. Test Case Rekam KPP .....	D-6
D.6. Test Case Ubah KPP.....	D-7
D.7. Test Case Hapus KPP .....	D-8
D.8. Test Case RekamBank.....	D-9
D.9. Test Case Ubah Bank .....	D-10
D.10. Test Case Hapus Bank.....	D-10
D.11. Test Case Rekam KPPN.....	D-11
D.12. Test Case Ubah KPPN.....	D-11
D.13. Test Case Hapus KPPN .....	D-12
D.14. Test Case Rekam Cara Bayar .....	D-13
D.15. Test Case Ubah Cara Bayar.....	D-13
D.16. Test Case Hapus Cara Bayar .....	D-14
D.17. Test Case Rekam Lokasi .....	D-14
D.18. Test Case Ubah Lokasi.....	D-15
D.19. Test Case Hapus Lokasi .....	D-16
D.20. Test Case Rekam Cara Tarik .....	D-16
D.21. Test Case Ubah Cara Tarik.....	D-17
D.22. Test Case Hapus Cara Tarik .....	D-17
D.23. Test Case Rekam MAK.....	D-18
D.24. Test Case Ubah MAK.....	D-19
D.25. Test Case Hapus MAK .....	D-19
D.26. Test Case Rekam Gol&Pangkat .....	D-20
D.27. Test Case Ubah Gol&Pangkat .....	D-20
D.28. Test Case Hapus Gol&Pangkat .....	D-21
D.29. Test Case Rekam Output .....	D-21
D.30. Test Case Ubah Output.....	D-22
D.31. Test Case Hapus Output .....	D-23
D.32. Test Case Rekam Kegiatan.....	D-23
D.33. Test Case Ubah Kegiatan .....	D-24
D.34. Test Case Hapus Kegiatan .....	D-25
D.35. Test Case Rekam Pejabat .....	D-25
D.36. Test Case Ubah Pejabat .....	D-26
D.37. Test Case Hapus Pejabat.....	D-27

D.38. Test Case Rekam Kewenangan .....	D-27
D.39. Test Case Ubah Kewenangan .....	D-28
D.40. Test Case Hapus Kewenangan .....	D-28
D.41. Test Case Rekam Program.....	D-29
D.42. Test Case Ubah Program .....	D-29
D.43. Test Case Hapus Program.....	D-30
D.44. Test Case Rekam KPKNL .....	D-31
D.45. Test Case Ubah KPKNL.....	D-31
D.46. Test Case Hapus KPKNL .....	D-32
D.47. Test Case Rekam Sumber Dana .....	D-33
D.48. Test Case Ubah Sumber Dana .....	D-33
D.49. Test Case Hapus Sumber Dana.....	D-34
D.50. Test Case Rekam Cara Perolehan.....	D-34
D.51. Test Case Ubah Cara Perolehan .....	D-35
D.52. Test Case Hapus Cara Perolehan.....	D-35
D.53. Test Case Rekam Aset Tetap.....	D-36
D.54. Test Case Ubah Aset Tetap .....	D-37
D.55. Test Case Hapus Aset Tetap .....	D-37
D.56. Test Case Rekam SSKel Barang.....	D-38
D.57. Test Case Ubah SSKel Barang .....	D-39
D.58. Test Case Hapus SSKel Barang.....	D-39
D.59. Test Case Rekam Bendahara .....	D-40
D.60. Test Case Ubah Bendahara.....	D-41
D.61. Test Case Hapus Bendahara .....	D-41
D.62. Test Case Rekam Fungsi .....	D-42
D.63. Test Case Ubah Fungsi .....	D-43
D.64. Test Case Hapus Fungsi .....	D-43
D.65. Test Case Rekam Beban .....	D-44
D.66. Test Case Ubah Beban.....	D-44
D.67. Test Case Hapus Beban .....	D-45
D.68. Test Case Rekam Jenis Dokumen.....	D-45
D.69. Test Case Ubah Jenis Dokumen .....	D-46
D.70. Test Case Hapus Jenis Dokumen.....	D-47
D.71. Test Case Rekam MAP Barang .....	D-47
D.72. Test Case Ubah MAP Barang.....	D-48

D.73. Test Case Hapus MAP Barang .....	D-49
D.74. Test Case Rekam Kanwil .....	D-49
D.75. Test Case Ubah Kanwil .....	D-50
D.76. Test Case Hapus Kanwil.....	D-51
D.77. Test Case Rekam Jenis SPM .....	D-51
D.78. Test Case Ubah Jenis SPM .....	D-52
D.79. Test Case Hapus Jenis SPM .....	D-53
D.80. Test Case Rekam Jenis SPP.....	D-53
D.81. Test Case Ubah Jenis SPP .....	D-54
D.82. Test Case Hapus Jenis SPP.....	D-55
D.83. Test Case Rekam Jenis Beban .....	D-55
D.84. Test Case Ubah Jenis Beban.....	D-56
D.85. Test Case Hapus Jenis Beban .....	D-56
D.86. Test Case Rekam Akun .....	D-57
D.87. Test Case Ubah Akun .....	D-57
D.88. Test Case Hapus Akun .....	D-58
D.89. Test Case Rekam Blokir .....	D-59
D.90. Test Case Ubah Blokir.....	D-59
D.91. Test Case Hapus Blokir .....	D-60
D.92. Test Case Rekam Departemen.....	D-60
D.93. Test Case Ubah Departemen .....	D-61
D.94. Test Case Hapus Departemen.....	D-62
D.95. Test Case Rekam GBKPK.....	D-62
D.96. Test Case Ubah GBKPK .....	D-63
D.97. Test Case Hapus GBKPK.....	D-64
D.98. Test Case Rekam Unit .....	D-64
D.99. Test Case Ubah Unit.....	D-65
D.100. Test Case Hapus Unit .....	D-65
D.101. Test Case Rekam IB .....	D-66
D.102. Test Case Ubah IB .....	D-67
D.103. Test Case Hapus IB .....	D-67
D.104. Test Case Rekam Profil Pengguna .....	D-68
D.105. Test Case Ubah Profil Pengguna .....	D-68
D.106. Test Case Hapus Profil Pengguna.....	D-69
D.107. Test Case Rekam Waktu Kerja.....	D-69

D.108. Test Case Ubah Waktu Kerja .....	D-70
D.109. Test Case Hapus Waktu Kerja.....	D-71
D.110. Test Case Backup Data.....	D-71
D.111. Test Case Rekam Peraturan .....	D-72
D.112. Test Case Ubah Peraturan.....	D-72
D.113. Test Case Hapus Peraturan .....	D-73
D.114. Test Case Logout .....	D-73
Biodata Penulis.....	E-1

## DAFTAR GAMBAR

Gambar 2. 1-Arsitektur MVC.....	17
Gambar 3. 1-Diagram alur metodologi penelitian tugas akhir ....	22
Gambar 4. 1-Gambar Iconix Process.....	23
Gambar 4. 2-Arsitektur Teknologi SIAPKAN modul Administrasi .....	28
Gambar 4. 3-Keterkaitan modul Administrasi dengan modul- modul lainnya.....	30
Gambar 4. 4-Desain Hubungan Antar Modul SIAPKAN .....	32
Gambar 4. 5-Diagram keterkaitan antar modul .....	33
Gambar 4. 6-Workflow Modul Administrasi .....	35
Gambar 4. 7-Workflow User pada modul Administrator pada Modul Administrasi.....	36
Gambar 4. 8-Domain Model Modul Administrasi (bagian atas).38	
Gambar 4. 9-Domain Model Modul Administrasi (bagian bawah) .....	38
Gambar 4. 10-Tampilan form Login .....	42
Gambar 4. 11-Peringatan salah mengisikan username/password	43
Gambar 4. 12-Salah menginputkan karakter pada username dan password.....	43
Gambar 4. 13-Pengguna berhasil login .....	44
Gambar 4. 14-Masuk pada halaman Utama Administrator .....	44
Gambar 4. 15-Desain Dropdown Menu Referensi .....	45
Gambar 4. 16-Desain Dropdown Menu Pengaturan .....	45
Gambar 4. 17-Desain Dropdown Menu Bantuan .....	46
Gambar 4. 18-Desain Tampilan Menu Tentang .....	46
Gambar 4. 19-Desain Tayang Referensi Umum – Bank .....	47
Gambar 4. 20-Desain form Rekam/Ubah Bank.....	47
Gambar 4. 21-Desain Tayang GBKPK .....	48
Gambar 4. 22-Desain Rekam/Ubah GBKPK .....	48
Gambar 4. 23-Desain Tayang dan Hapus Pengguna .....	49
Gambar 4. 24-Desain Rekam/Ubah Pengguna .....	49
Gambar 4. 25-Desain Pengaturan Waktu Kerja/Yulian Date.....	50

Gambar 4. 26-Desain Daftar Waktu Kerja .....	50
Gambar 4. 27-Desain Menu Bantuan .....	51
Gambar 4. 28-Desain pop-up notifikasi logout. ....	51
Gambar 4. 29-GUI StoryBoard .....	52
Gambar 4. 30-Potongan class diagram Modul Administrasi (bagian atas) .....	53
Gambar 4. 31-Potongan class diagram Modul Administrasi (bagian bawah) .....	53
Gambar 4. 32 Use Case Diagram modul Administrasi.....	59
Gambar 4. 33-Use Case level 1 modul Administrasi .....	60
Gambar 4. 34-Use Case Diagram level 2: Pengelolaan Referensi .....	60
Gambar 4. 35-Use Case Diagram level 2: Pengelolaan Pengaturan .....	61
Gambar 4. 36-Use Case level 2: Pengelolaan Bantuan .....	61
Gambar 4. 37-Use Case level 2: Tayang Tentang SIAPKAN.....	62
Gambar 5. 1-Project directory SIAPKAN.....	69
Gambar 5. 2-Pembagian direktori berdasarkan metode MVC ....	70
Gambar 5. 3-Sub Direktori dari Web Directory .....	71
Gambar 5. 4-File Konfigurasi.....	72
Gambar 5. 5-File Persistence.xml.....	72
Gambar 5. 6-File Web.xml.....	72
Gambar 5. 7-Membuat Entity Class dari database yang ada. ....	73
Gambar 5. 8-Potongan entity class “TBank” diambil dari tabel t_bank .....	74
Gambar 5. 9-Potongan file DaftarBank.java .....	75
Gambar 5. 10-Potongan file BankServlet.....	76
Gambar 5. 11-Potongan file DaftarBank untuk rekam data .....	76
Gambar 5. 12-Potongan file BankController untuk proses rekam data bank.....	77
Gambar 5. 13-Tampilan form login/ index.jsp .....	77
Gambar 5. 14 Halaman Utama Modul Administrasi .....	78
Gambar 5. 15-Halaman Tayang Referensi Bank.....	78
Gambar 5. 16-Halaman utama menu Pengaturan .....	79
Gambar 5. 17-Halaman Bantuan tayang Peraturan .....	79

Gambar 5. 18-Halaman Tentang .....	80
Gambar 5. 19-Halaman Tayang Pengguna.....	80
Gambar 5. 20-Halaman Rekam Profil Pengguna .....	81
Gambar 5. 21-Halaman Edit Profil Pengguna .....	81
Gambar 5. 22-bagian atas (head) script index.jsp .....	82
Gambar 5. 23-bagian bawah (body) index.jsp.....	83
Gambar 5. 24-Index.jsp setelah running project SIAPKAN .....	83
Gambar 5. 25-Potongan LoginServlet, request parameter username dan password .....	84
Gambar 5. 26-Potongan DaftarPengguna.Java, method boolean check.....	85
Gambar 5. 27-class entity TBank .....	89
Gambar 5. 28-Potongan method getBank pada model DaftarBank .....	90
Gambar 5. 29-Potongan iterasi session pageref untuk url listbank pada BankServlet.....	90
Gambar 5. 30-Potongan method listBank pada potongan kelas BankController untuk menayangkan data Bank .....	91
Gambar 5. 31-Implementasi tayang data (referensi bank).....	92
Gambar 5. 32-Potongan Method rekamBank pada class DaftarBank .....	92
Gambar 5. 33-Potongan method prosesrekamBank pada kelas BankController .....	93
Gambar 5. 34-Iterator else if membawa session pageref url listbank pada kelas BankServlet .....	93
Gambar 5. 35-Implementasi Rekam data Bank.....	94
Gambar 5. 36-Potongan metbhod editBank .....	94
Gambar 5. 37-Potongan method ubahBank & prosesUbahBank pada kelas BankController .....	95
Gambar 5. 38-Potongan iterasi else if mengambil sesion pageref url ubahbank pada kelas BankServlet.....	95
Gambar 5. 39-Potongan method findBankbyNama pada kelas DaftarBank .....	96
Gambar 5. 40-method cariBank pada kelas BankController .....	96



Gambar 5. 41-Potongan iterasi else if dengan session pageref url caribank pada kelas BankServlet.....	96
Gambar 5. 42-Potongan method tesBackup pada kelas DaftarBackup.....	97
Gambar 5. 43-Potongan method listBackup pada kelas BackupController .....	97
Gambar 5. 44-Potongan iterasi else if session pageref url listbackup pada kelas BackupServlet .....	98
Gambar 5. 45-Potongan proses remove session pada LogoutServlet .....	98
Gambar 5. 46-Subversion google code untuk integrasi modul....	99
Gambar 5. 47-Integrasi Modul pada Project SIAPKAN .....	99
Gambar 5. 48-Hasil Migrasi data .....	102
Gambar B. 1 Diagram Robustness Rekam User.....	B-2
Gambar B. 2 Diagram Robustness Rekam Bank.....	B-2
Gambar B. 3 Diagram Robustness Hapus Bank.....	B-3
Gambar B. 4 Diagram Robustness Rekam Dokumen .....	B-3
Gambar B. 5 Diagram Robustness Hapus Dokumen.....	B-4
Gambar B. 6 Diagram Robustness Rekam Kegiatan.....	B-5
Gambar B. 7 Diagram Robustness Hapus Kegiatan.....	B-5
Gambar B. 8 Diagram Robustness Rekam Lokasi .....	B-6
Gambar B. 9 Diagram Robustness Hapus Lokasi .....	B-6
Gambar B. 10 Diagram Robustness Rekam MAK.....	B-7
Gambar B. 11 Diagram Robustness Hapus MAK.....	B-7
Gambar B. 12 Diagram Robustness Rekam Output .....	B-8
Gambar B. 13 Diagram Robustness Hapus Output .....	B-8
Gambar B. 14 Diagram Robustness Rekam Pejabat .....	B-9
Gambar B. 15 Diagram Robustness Hapus Pejabat .....	B-9
Gambar B. 16 Diagram Robustness Rekam Program .....	B-10
Gambar B. 17 Diagram Robustness Hapus Program.....	B-10
Gambar B. 18 Diagram Robustness Rekam Jenis Kegiatan....	B-11
Gambar B. 19 Diagram Robustness Hapus Jenis Kegiatan.....	B-11
Gambar B. 20 Diagram Robustness Rekam Satuan Kerja .....	B-12
Gambar B. 21 Diagram Robustness Hapus Satuan Kerja.....	B-12
Gambar C. 1 Diagram Sequence Login.....	C-2

Gambar C. 2 Diagram Sequence Pengelolaan Beban.....	C-3
Gambar C. 3 Diagram Sequence Rekam Bank.....	C-3
Gambar C. 4 Diagram Sequence Rekam Dokumen .....	C-4
Gambar C. 5 Diagram Sequence Rekam Kegiatan.....	C-4
Gambar C. 6 Diagram Sequence Rekam Lokasi .....	C-5
Gambar C. 7 Diagram Sequence Rekam MAK.....	C-6
Gambar C. 8 Diagram Sequence Rekam Output.....	C-6
Gambar C. 9 Diagram Sequence Rekam Pejabat .....	C-7
Gambar C. 10 Diagram Sequence Rekam Program .....	C-7
Gambar C. 11 Diagram Sequence Rekam Satker .....	C-8
Gambar C. 12 Diagram Sequence Ubah Bank .....	C-8
Gambar C. 13 Diagram Sequence Ubah Dokumen .....	C-9
Gambar C. 14 Diagram Sequence Ubah Kegiatan .....	C-9
<i>Gambar C. 15 Diagram Sequence Ubah Lokasi.....</i>	<i>C-10</i>
Gambar C. 16 Diagram Sequence Ubah MAK .....	C-11
Gambar C. 17 Diagram Sequence Ubah Output.....	C-12
Gambar C. 18 Diagram Sequence Ubah Pejabat .....	C-13
Gambar C. 19 Diagram Sequence Ubah Program .....	C-14
Gambar C. 20 Diagram Sequence Ubah Satker .....	C-15
Gambar C. 21 Diagram Sequence Ubah Pengguna.....	C-16
Gambar C. 22 Diagram Sequence Logout.....	C-16

## DAFTAR TABEL

Tabel 4. 1 Analisis Hubungan Kebutuhan dan Pengguna .....	25
Tabel 4. 3 Desain Use Case Level 3 Modul Administrasi .....	62
Tabel 5. 1 Spesifikasi client .....	67
Tabel 5. 2 Spesifikasi Perangkat Lunak .....	67
Tabel 5. 3 Uji Coba Interaksi .....	102
Tabel 5. 4 Hasil Unit Test .....	104
Tabel A. 1 Deskripsi Use Case Proses Login .....	A-2
Tabel A. 2 Deskripsi Use Case Tayang Bendahara .....	A-3
Tabel A. 3 deskripsi Use Case Rekam Bendahara .....	A-3
Tabel A. 4 Deskripsi Use Case Ubah Bendahara .....	A-4
Tabel A. 5 Deskripsi Use Case Hapus Bendahara .....	A-5
Tabel A. 6 Deskripsi Use Case Tayang Beban .....	A-5
Tabel A. 7 Deskripsi Use Case Rekam Beban .....	A-6
Tabel A. 8 Deskripsi Use Case Ubah Beban .....	A-7
Tabel A. 9 Deskripsi Use Case Hapus Beban .....	A-7
Tabel A. 10 Deskripsi Use Case Tayang Fungsi .....	A-8
Tabel A. 11 Deskripsi Use Case Rekam Fungsi .....	A-9
Tabel A. 12 Deskripsi Use Case Ubah Fungsi .....	A-9
Tabel A. 13 Deskripsi Use Case Hapus Fungsi .....	A-10
Tabel A. 14 Deskripsi Use Case Tayang Jenis Dokumen .....	A-11
Tabel A. 15 Deskripsi Use Case Rekam Jenis Dokumen .....	A-11
Tabel A. 16 Deskripsi Use Case Ubah Jenis Dokumen .....	A-12
Tabel A. 17 Deskripsi Use Case Hapus Jenis Dokumen .....	A-13
Tabel A. 18 Deskripsi Use Case Tayang MAK .....	A-13
Tabel A. 19 Deskripsi Use Case Rekam MAK .....	A-14
Tabel A. 20 Deskripsi Use Case Ubah MAK .....	A-15
Tabel A. 21 Deskripsi Use Case Hapus MAK .....	A-15
Tabel A. 22 Deskripsi Use Case Tayang Jenis SPM .....	A-16
Tabel A. 23 Deskripsi Use Case Rekam Jenis SPM .....	A-17
Tabel A. 24 Deskripsi Use Case Ubah Jenis SPM .....	A-17
Tabel A. 25 Deskripsi Use Case Hapus Jenis SPM .....	A-18
Tabel A. 26 Deskripsi Use Case Tayang Cara Bayar .....	A-19

Tabel A. 27 Deskripsi Use Case Rekam Cara Bayar .....	A-19
Tabel A. 28 Deskripsi Use Case Ubah Cara Bayar .....	A-20
Tabel A. 29 Deskripsi Use Case Hapus Cara Bayar.....	A-21
Tabel A. 30 Deskripsi Use Case Tayang Jenis Dokumen .....	A-21
Tabel A. 31 Deskripsi Use Case Rekam Jenis Dokumen.....	A-22
Tabel A. 32 Deskripsi Use Case Ubah Jenis Dokumen.....	A-23
Tabel A. 33 Deskripsi Use Case Hapus Jenis Dokumen .....	A-23
Tabel A. 34 Deskripsi Use Case Tayang Sumber Dana .....	A-24
Tabel A. 35 Deskripsi Use Case Rekam Sumber Dana.....	A-25
Tabel A. 36 Deskripsi Use Case Ubah Sumber Dana .....	A-25
Tabel A. 37 Deskripsi Use Case Hapus Sumber Dana.....	A-26
Tabel A. 38 Deskripsi Use Case Tayang Jenis SPM.....	A-27
Tabel A. 39 Deskripsi Use Case Rekam Jenis SPM.....	A-27
Tabel A. 40 Deskripsi Use Case Ubah Jenis SPM .....	A-28
Tabel A. 41 Deskripsi Use Case Hapus Jenis SPM.....	A-29
Tabel A. 42 Deskripsi Use Case Tayang Aset Tetap .....	A-30
Tabel A. 43 Deskripsi Use Case Rekam Aset Tetap .....	A-30
Tabel A. 44 Deskripsi Use Case Ubah Aset Tetap.....	A-31
Tabel A. 45 Deskripsi Use Case Hapus Aset Tetap .....	A-32
Tabel A. 46 Deskripsi Use Case Tayang Cara Perolehan .....	A-32
Tabel A. 47 Deskripsi Use Case Rekam Cara Perolehan .....	A-33
Tabel A. 48 Deskripsi Use Case Ubah Cara Perolehan.....	A-34
Tabel A. 49 Deskripsi Use Case Hapus Cara Perolehan .....	A-34
Tabel A. 50 Deskripsi Use Case Tayang SSKel Barang .....	A-35
Tabel A. 51 Deskripsi Use Case Rekam SSKel Barang.....	A-36
Tabel A. 52 Deskripsi Use Case Ubah SSKel Barang .....	A-36
Tabel A. 53 Deskripsi Use Case Hapus SSKel BMN .....	A-37
Tabel A. 54 Deskripsi Use Case Tayang Map Barang .....	A-38
Tabel A. 55 Deskripsi Use Case Rekam Map Barang.....	A-38
Tabel A. 56 Deskripsi Use Case Ubah Map Barang .....	A-39
Tabel A. 57 Deskripsi Use Case Hapus Map Barang.....	A-40
Tabel A. 58 Deskripsi Use Case Tayang Kanwil .....	A-41
Tabel A. 59 Deskripsi Use Case Rekam Kanwil.....	A-41
Tabel A. 60 Deskripsi Use Case Ubah Kanwil .....	A-42
Tabel A. 61 Deskripsi Use Case Hapus Kanwil.....	A-43

Tabel A. 62 Deskripsi Use Case Tayang Akun .....	A-43
Tabel A. 63 Deskripsi Use Case Rekam Akun.....	A-44
Tabel A. 64 Deskripsi Use Case Ubah Akun .....	A-45
Tabel A. 65 Deskripsi Use Case Hapus Akun.....	A-45
Tabel A. 66 Deskripsi Use Case Tayang Blokir.....	A-46
Tabel A. 67 Deskripsi Use Case Rekam Blokir .....	A-46
Tabel A. 68 Deskripsi Use Case Ubah Blokir .....	A-47
Tabel A. 69 Deskripsi Use Case Hapus Blokir .....	A-48
Tabel A. 70 Deskripsi Use Case Tayang Departemen .....	A-49
Tabel A. 71 Deskripsi Use Case Rekam Departemen .....	A-49
Tabel A. 72 Deskripsi Use Case Ubah Departemen.....	A-50
Tabel A. 73 Deskripsi Use Case Hapus Departemen .....	A-51
Tabel A. 74 Deskripsi Use Case Tayang GBKPK .....	A-51
Tabel A. 75 Deskripsi Use Case Rekam GBKPK .....	A-52
Tabel A. 76 Deskripsi Use Case Ubah GBKPK.....	A-53
Tabel A. 77 Deskripsi Use Case Hapus GBKPK .....	A-53
Tabel A. 78 Deskripsi Use Case Tayang Kegiatan .....	A-54
Tabel A. 79 Deskripsi Use Case Rekam Kegiatan .....	A-55
Tabel A. 80 Deskripsi Use Case Ubah Kegiatan.....	A-55
Tabel A. 81 Deskripsi Use Case Hapus Kegiatan .....	A-56
Tabel A. 82 Deskripsi Use Case Tayang Unit.....	A-57
Tabel A. 83 Deskripsi Use Case Rekam Unit .....	A-57
Tabel A. 84 Deskripsi Use Case Ubah Unit .....	A-58
Tabel A. 85 Deskripsi Use Case Hapus Unit.....	A-59
Tabel A. 86 Deskripsi Use Case Tayang IB .....	A-59
Tabel A. 87 Deskripsi Use Case Rekam IB.....	A-60
Tabel A. 88 Deskripsi Use Case Ubah IB .....	A-61
Tabel A. 89 Deskripsi Use Case Hapus IB.....	A-61
Tabel A. 90 Deskripsi Use Case Tayang Output.....	A-62
Tabel A. 91 Deskripsi Use Case Rekam Output .....	A-63
Tabel A. 92 Deskripsi Use Case Ubah Output .....	A-63
Tabel A. 93 Deskripsi Use Case Hapus Output.....	A-64
Tabel A. 94 Deskripsi Use Case Tayang Profil Pengguna .....	A-65
Tabel A. 95 Deskripsi Use Case Rekam Profil Pengguna.....	A-65
Tabel A. 96 Deskripsi Use Case Ubah Profil Pengguna .....	A-66

Tabel A. 97 Deskripsi Use Case Hapus Profil Pengguna.....	A-67
Tabel A. 98 Deskripsi Use Case Tayang Waktu Kerja .....	A-68
Tabel A. 99 Deskripsi Use Case Rekam Waktu Kerja.....	A-68
Tabel A. 100 Deskripsi Use Case Ubah Waktu Kerja.....	A-69
Tabel A. 101 Deskripsi Use Case Hapus Waktu Kerja .....	A-70
Tabel A. 102 Deskripsi Use Case Backup Data .....	A-70
Tabel A. 103 Deskripsi Use Case Tayang Konten .....	A-71
Tabel A. 104 Deskripsi Use Case Tayang Peraturan.....	A-71
Tabel A. 105 Deskripsi Use Case Rekam Peraturan .....	A-72
Tabel A. 106 Deskripsi Use Case Ubah Peraturan .....	A-73
Tabel A. 107 Deskripsi Use Case Hapus Peraturan .....	A-73
Tabel A. 108 Deskripsi Use Case Tayang Tentang.....	A-74
Tabel A. 109 Deskripsi Use Case Proses Logout .....	A-75
Tabel D. 1 Test Case Login.....	D-2
Tabel D. 2 Test Case Rekam Satker .....	D-3
Tabel D. 3 Test Case Ubah Satker.....	D-4
Tabel D. 4 Test Case Hapus Satker .....	D-6
Tabel D. 5 Test Case Rekam KPP .....	D-6
Tabel D. 6 Test Case Ubah KPP .....	D-7
Tabel D. 7 Test Case Hapus KPP.....	D-8
Tabel D. 8 Test Case Rekam Bank.....	D-9
Tabel D. 9 Test Case Ubah Bank .....	D-10
Tabel D. 10 Test Case Hapus Bank.....	D-10
Tabel D. 11 Test Case Rekam KPPN .....	D-11
Tabel D. 12 Test Case Ubah KPPN.....	D-11
Tabel D. 13 Test Case Hapus KPPN .....	D-12
Tabel D. 14 Test Case Rekam Cara Bayar .....	D-13
Tabel D. 15 Test Case Ubah Cara Bayar.....	D-13
Tabel D. 16 Test Case Hapus Cara Bayar .....	D-14
Tabel D. 17 Test Case Rekam Lokasi .....	D-14
Tabel D. 18 Test Case Ubah Lokasi .....	D-15
Tabel D. 19 Test Case Hapus Lokasi .....	D-16
Tabel D. 20 Test Case Rekam Cara Tarik.....	D-16
Tabel D. 21 Test Case Ubah Cara Tarik.....	D-17
Tabel D. 22 Test Case Hapus Cara Tarik .....	D-17

Tabel D. 23 Test Case Rekam MAK .....	D-18
Tabel D. 24 Test Case Ubah MAK .....	D-19
Tabel D. 25 Test Case Hapus MAK .....	D-19
Tabel D. 26 Test Case Rekam Gol&Pangkat .....	D-20
Tabel D. 27 Test Case Ubah Gol&Pangkat .....	D-20
Tabel D. 28 Test Case Hapus Gol&Pangkat .....	D-21
Tabel D. 29 Test Case Rekam Output .....	D-21
Tabel D. 30 Test Case Ubah Output .....	D-22
Tabel D. 31 Test Case Hapus Output .....	D-23
Tabel D. 32 Test Case Rekam Kegiatan .....	D-23
Tabel D. 33 Test Case Ubah Kegiatan .....	D-24
Tabel D. 34 Test Case Hapus Kegiatan .....	D-25
Tabel D. 35 Test Case Rekam Pejabat .....	D-25
Tabel D. 36 Test Case Ubah Pejabat .....	D-26
Tabel D. 37 Test Case Hapus Pejabat .....	D-27
Tabel D. 38 Test Case Rekam Kewenangan .....	D-27
Tabel D. 39 Test Case Ubah Kewenangan .....	D-28
Tabel D. 40 Test Case Hapus Kewenangan .....	D-28
Tabel D. 41 Test Case Rekam Program .....	D-29
Tabel D. 42 Test Case Ubah Program .....	D-29
Tabel D. 43 Test Case Hapus Program .....	D-30
Tabel D. 44 Test Case Rekam KPKNL .....	D-31
Tabel D. 45 Test Case Ubah KPKNL .....	D-31
Tabel D. 46 Test Case Hapus KPKNL .....	D-32
Tabel D. 47 Test Case Rekam Sumber Dana .....	D-33
Tabel D. 48 Test Case Ubah Sumber Dana .....	D-33
Tabel D. 49 Test Case Hapus Sumber Dana .....	D-34
Tabel D. 50 Test Case Rekam Cara Perolehan .....	D-34
Tabel D. 51 Test Case Ubah Cara Perolehan .....	D-35
Tabel D. 52 Test Case Hapus Cara Perolehan .....	D-35
Tabel D. 53 Test Case Rekam Aset Tetap .....	D-36
Tabel D. 54 Test Case Ubah Aset Tetap .....	D-37
Tabel D. 55 Test Case Hapus Aset Tetap .....	D-37
Tabel D. 56 Test Case Rekam SSKel Barang .....	D-38
Tabel D. 57 Test Case Ubah SSKel Barang .....	D-39

Tabel D. 58 Test Case Hapus SSKel Barang.....	D-39
Tabel D. 59 Test Case Rekam Bendahara.....	D-40
Tabel D. 60 Test Case Ubah Bendahara.....	D-41
Tabel D. 61 Test Case Hapus Bendahara.....	D-41
Tabel D. 62 Test Case Rekam Fungsi.....	D-42
Tabel D. 63 Test Case Ubah Fungsi.....	D-43
Tabel D. 64 Test Case Hapus Fungsi.....	D-43
Tabel D. 65 Test Case Rekam Beban.....	D-44
Tabel D. 66 Test Case Ubah Beban.....	D-44
Tabel D. 67 Test Case Hapus Beban.....	D-45
Tabel D. 68 Test Case Rekam Jenis Dokumen.....	D-45
Tabel D. 69 Test Case Ubah Jenis Dokumen.....	D-46
Tabel D. 70 Test Case Hapus Jenis Dokumen.....	D-47
Tabel D. 71 Test Case Rekam MAP Barang.....	D-47
Tabel D. 72 Test Case Ubah MAP Barang.....	D-48
Tabel D. 73 Test Case Hapus MAP Barang.....	D-49
Tabel D. 74 Test Case Rekam Kanwil.....	D-49
Tabel D. 75 Test Case Ubah Kanwil.....	D-50
Tabel D. 76 Test Case Hapus Kanwil.....	D-51
Tabel D. 77 Test Case Rekam Jenis SPM.....	D-51
Tabel D. 78 Test Case Ubah Jenis SPM.....	D-52
Tabel D. 79 Test Case Hapus Jenis SPM.....	D-53
Tabel D. 80 Test Case Rekam Jenis SPP.....	D-53
Tabel D. 81 Test Case Ubah Jenis SPP.....	D-54
Tabel D. 82 Test Case Hapus Jenis SPP.....	D-55
Tabel D. 83 Test Case Rekam Jenis Beban.....	D-55
Tabel D. 84 Test Case Ubah Jenis Beban.....	D-56
Tabel D. 85 Test Case Hapus Jenis Beban.....	D-56
Tabel D. 86 Test Case Rekam Akun.....	D-57
Tabel D. 87 Test Case Ubah Akun.....	D-57
Tabel D. 88 Test Case Hapus Akun.....	D-58
Tabel D. 89 Test Case Rekam Blokir.....	D-59
Tabel D. 90 Test Case Ubah Blokir.....	D-59
Tabel D. 91 Test Case Hapus Blokir.....	D-60
Tabel D. 92 Test Case Rekam Departemen.....	D-60



Tabel D. 93 Test Case Ubah Departemen .....	D-61
Tabel D. 94 Test Case Hapus Departemen.....	D-62
Tabel D. 95 Test Case Rekam GBKPK.....	D-62
Tabel D. 96 Test Case Ubah GBKPK .....	D-63
Tabel D. 97 Test Case Hapus GBKPK.....	D-64
Tabel D. 98 Test Case Rekam Unit .....	D-64
Tabel D. 99 Test Case Ubah Unit.....	D-65
Tabel D. 100 Test Case Hapus Unit .....	D-65
Tabel D. 101 Test Case Rekam IB .....	D-66
Tabel D. 102 Test Case Ubah IB.....	D-67
Tabel D. 103 Test Case Hapus IB .....	D-67
Tabel D. 104 Test Case Rekam Profil Pengguna .....	D-68
Tabel D. 105 Test Case Ubah Profil Pengguna.....	D-68
Tabel D. 106 Test Case Ubah Profil Pengguna.....	D-69
Tabel D. 107 Test Case Rekam Waktu Kerja.....	D-69
Tabel D. 108 Test Case Ubah Waktu Kerja .....	D-70
Tabel D. 109 Test Case Hapus Waktu Kerja.....	D-71
Tabel D. 110 Test Case Backup Data.....	D-71
Tabel D. 111 Test Case Rekam Peraturan.....	D-72
Tabel D. 112 Test Case Ubah Peraturan.....	D-72
Tabel D. 113 Test Case Hapus Peraturan.....	D-73
Tabel D. 114 Test Case Logout.....	D-73

# BAB I

## PENDAHULUAN

Pada bab ini, akan dijelaskan tentang Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan Tugas Akhir, dan Relevansi atau Manfaat Kegiatan Tugas Akhir.

### 1.1. Latar Belakang

Penerapan teknologi informasi dan sistem informasi saat ini sudah menggapai seluruh instansi pemerintahan di Indonesia dalam pencapaian peningkatan kinerja sumber daya manusia maupun otomatisasi pengelolaan aset. Tidak terlepas pula pada Direktorat Jenderal Perbendaharaan (DJPBN) yang merupakan sebuah instansi vertikal dibawah naungan Kementerian Keuangan, turut menerapkan sistem informasi sebagai pendukung pencapaian misinya, antara lain:

1. Mewujudkan pelaksanaan anggaran berbasis kinerja
2. Mewujudkan pengelolaan kas negara yang transparan dan akuntabel,
3. Menghasilkan pelayanan di bidang perbendaharaan dan informasi keuangan yang cepat, tepat dan akurat,
4. Mewujudkan dukungan teknis perbendaharaan yang handal, *terintegrasi, terotomatisasi, dan mudah diterapkan,*
5. Menyempurnakan proses bisnis sistem perbendaharaan sesuai *bestpractice,*
6. Melaksanakan pemberdayaan dan integrasi seluruh sumber daya organisasi secara optimal, dan
7. Mewujudkan pengelolaan kas negara yang efisien dan optimal

Sistem informasi yang terintegrasi dan terkelola dengan baik tentunya sangat dibutuhkan oleh DJPBN dalam melaksanakan tugasnya. Dilihat dari sisi lingkungannya, DJPBN memiliki 181 unit instansi vertikal bernama Kantor Pelayanan Perbendaharaan Negara (KPPN) dan 30 Kantor Wilayah Direktorat Jenderal

Perbendaharaan yang tersebar di seluruh Propinsi di Indonesia guna melayani lebih dari tiga puluh ribu Satuan Kerja (Satker).

Satker-Satker tersebut adalah instansi pemerintah terkecil yang merupakan bagian dari struktur kementerian di pemerintahan. Satker mendapatkan dana dari Anggaran Pendapatan dan Belanja Negara (APBN) untuk melaksanakan seluruh kegiatannya. Satker memiliki susunan hierarki yaitu kepala satker (kepala dinas, kepala kantor, atau direktur), bendahara satker, pejabat pembuat komitmen, pejabat penanda tangan Surat Perintah Membayar (SPM), Pembuat Daftar Gaji (PDG) dan para pegawai yang bekerja pada instansi tersebut. Sebagai contoh Satker adalah KPPN, Kantor Dinas Pendidikan Surabaya, Kantor Poltabes Surabaya dan lain-lain.

Sesuai dengan Undang Undang Nomor 17 tahun 2003 tentang Keuangan Negara, dalam mendukung kinerja Satkernya, DJPBN mengembangkan berbagai aplikasi yang berguna sebagai alat bantu mereka dalam menatausahakan pencairan dana APBN. pengembangan aplikasi tersebut diharapkan mampu memenuhi prinsip tepat waktu, akuntabel serta telah sesuai dengan standar akuntansi pemerintah. Beberapa aplikasi yang telah dikembangkan DJPBN antara lain:

Aplikasi Rencana Kerja Anggaran Kementerian/Lembaga (RKAKL), digunakan sebagai pendukung Satuan Kerja dalam penyusunan rencana kerja dan anggaran tingkat kementerian/lembaga.

1. Aplikasi Daftar Isian Pelaksanaan Anggaran (DIPA), digunakan sebagai pendukung Satuan Kerja dalam penyusunan Daftar Isian Pelaksanaan Anggaran untuk pelaksanaan pencairan dana.
2. Aplikasi Surat Perintah Membayar (SPM) digunakan sebagai pendukung Satuan Kerja dalam pembuatan Surat Perintah Membayar sebagai alat pencairan dan pembayaran kegiatan yang dilakukan selama tahun anggaran berjalan.

3. Aplikasi Sistem Informasi Keuangan (SISKA) untuk pembukuan bendahara Satker digunakan sebagai pendukung Satuan Kerja dalam proses pelaksanaan teknis pencairan dana serta pembukuan dan penatausahaan anggaran.
4. Aplikasi Persediaan digunakan sebagai pendukung Satuan Kerja dalam proses pengaturan dan pengelolaan Persediaan pada masing-masing instansi.
5. Aplikasi Sistem Informasi Manajemen Keuangan Barang Milik Negara (SIMAK BMN) digunakan sebagai pendukung Satuan Kerja dalam proses pengaturan dan pengelolaan Aset Tetap dan Barang Milik Negara, dan
6. Aplikasi Sistem Akuntansi Kuasa Pengguna Anggaran (SAKPA) digunakan sebagai pendukung Satuan Kerja dalam proses pelaporan keuangan dalam tahun anggaran berjalan.

Aplikasi-aplikasi tersebut pada awalnya memang cukup bermanfaat dan efisien dari segi waktu dan biaya. Namun sejalan dengan semakin besarnya kebutuhan pemrosesan data transaksi dari Satker ke DJPBN ataupun sebaliknya, tentunya tidak bisa dilakukan dengan banyak aplikasi dengan sistem database yang saling terpisah. Tentu saja proses sinkronisasi atau rekonsiliasi data rutin dijalankan antar aplikasi satu dengan lainnya agar data yang ada selalu akuntabel. Namun permasalahan seperti duplikasi data seperti pada fungsi Rekam Ubah Hapus (RUH) untuk data yang sama pada aplikasi yang berbeda, pindah rekam data, dan hilangnya data membuat proses tersebut berjalan lebih lama dan tidak efisien.

Karena itulah diperlukan sebuah sistem informasi yang terintegrasi sebagai upaya menyatukan seluruh fungsi pada aplikasi-aplikasi Satker tersebut. Sistem Informasi berupa aplikasi yang akan dibangun tersebut bernama Sistem Informasi Akuntansi dan Pengelolaan Keuangan (SIAPKAN). Aplikasi SIAPKAN ini berbasis Web dengan menggunakan bahasa pemrograman Java (JavaServer dan JSP) serta sistem manajemen

basis data PostgreSQL. Di dalamnya terdiri beberapa modul antara lain:

1. Modul Administrasi, merupakan modul yang digunakan untuk manajemen database dan arus data, manajemen pengguna dan manajemen referensi-referensi yang berkaitan dengan modul lain,
2. Modul Anggaran, merupakan modul yang diadopsi dari Aplikasi RKAKL dan aplikasi DIPA,
3. Modul Pembayaran, merupakan modul yang diadopsi dari Aplikasi SPM,
4. Modul Aset Tetap, merupakan modul yang diadopsi dari Aplikasi SIMAK-BMN,
5. Modul Persediaan, merupakan modul yang diadopsi dari Aplikasi Persediaan.
6. Modul Bendahara, merupakan modul yang diadopsi dari Aplikasi SSKA, dan
7. Modul Pelaporan, merupakan modul yang diadopsi dari Aplikasi SAKPA.

Dalam pengerjaan tugas akhir ini, penulis membatasi pengembangan pada pengembangan aplikasi SIAPKAN khususnya pada modul Administrasi. Modul Administrasi diharapkan dapat membantu seorang administrator/admin dalam mengelola administrasi dan menjaga agar data / informasi yang terdapat dalam aplikasi SIAPKAN merupakan data yang benar dan valid, manajemen pengguna baik fungsi dan hak aksesnya, serta manajemen referensi yang terkait dengan modul lainnya.

SIAPKAN merupakan aplikasi berbasis web yang dikembangkan dengan menggunakan bahasa pemrograman Java (servlet dan JSP) dan database PostgreSQL. Java dipilih karena keunggulan yang dimiliki antara lain menggunakan paradigma *Object Oriented Programming (OOP)* sehingga lebih efisien dalam pemrograman, pemeliharaan, dan pengembangan, bersifat open source, serta dapat berjalan pada platform yang berbeda-beda

sedangkan PostgreSQL dipilih karena selain bersifat open source juga memiliki keunggulan dalam hal kecepatan update record. Dalam Tugas akhir ini dibahas pengembangan Aplikasi SIAPKAN Modul Administrasi, yaitu modul yang digunakan untuk melakukan pengelolaan referensi tiap modul, pengaturan pengguna, waktu kerja, bantuan dan informasi tentang aplikasi SIAPKAN.

## 1.2. Perumusan Masalah

Rumusan masalah yang akan diselesaikan dalam pengerjaan tugas akhir ini meliputi:

1. Bagaimana mendesain dan membangun modul Administrasi agar sesuai dengan modul-modul lain pada aplikasi SIAPKAN?
2. Bagaimana proses integrasi database modul Administrasi dengan database modul lainnya?
3. Bagaimana membuat skrip/kode bahasa pemrograman yang tepat untuk modul Administrasi?
4. Bagaimana menguji dan mengimplementasikan modul Administrasi?

## 1.3. Batasan Masalah

Batasan permasalahan dalam tugas akhir ini adalah :

1. Pengguna pada modul Administrasi adalah seorang Administrator/Admin yang diserah-tugaskan langsung oleh pejabat satuan kerja berwenang.
2. Username dan Password pengguna modul Administrasi ditentukan sesuai dengan kode *role* setiap modul.
3. Perancangan fungsionalitas modul Administrasi dibatasi pada:
  - a. Pengelolaan Referensi seluruh modul
  - b. Pengelolaan Pengguna/Pengguna Management
  - c. Pesan Galat/Error Message
  - d. Dokumentasi bantuan

- e. Backup data
4. Pengelolaan referensi modul di SIAPKAN dibatasi pada:
  - a. Referensi modul anggaran,
  - b. Referensi modul komitmen,
  - c. Referensi modul bendahara,
  - d. Referensi modul pembayaran,
  - e. Referensi modul persediaan,
  - f. Referensi modul aset tetap dan,
  - g. Referensi umum,
5. Pengelolaan pengaturan dibatasi pada:
  - a. Mengelola pengguna modul, antara lain:
    - Mengelola profil pengguna,
    - Mengelola role pengguna-modul,
    - Mengelola validasi pengguna saat login, dan
    - Mengelola *right access* per modul.
  - b. Melakukan backup database.

#### **1.4. Tujuan**

Tujuan pengerjaan tugas akhir ini adalah pembuatan modul Administrasi sebagai bagian dari aplikasi SIAPKAN yang akan menjadi aplikasi tunggal pengelolaan pencairan dana APBN dan penggunaan uang negara di lingkup satuan kerja Direktorat Jenderal Perbendaharaan Negara.

#### **1.5. Relevansi atau Manfaat**

Tugas akhir ini diharapkan dapat memberikan manfaat antara lain:

1. Membantu tugas administrator dalam mengelola aplikasi SIAPKAN secara *back-end process*.
2. Mengurangi duplikasi data pasca rekonsiliasi data dari aplikasi-aplikasi Satker yang saling terpisah.

3. Memperlancar proses pelaporan dari Satker di lingkup DJPBN hingga Kantor Pusat Direktorat Jenderal Perbendaharaan.
4. Mengintegrasikan data jurnal dari berbagai modul terkait untuk menjaga integritas laporan keuangan yang dihasilkan oleh satuan kerja.
5. Mempermudah kerja satuan kerja/satker dalam melakukan rutinitaspengelolaan keuangan negara.

#### **1.6. Sistematika Penulisan**

Sistematika penulisan buku tugas akhir dibagi menjadi 6 bab sebagai berikut.

#### **BAB IPENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan permasalahan, batasan masalah/ruang lingkup, tujuan, manfaat dan sistematika penulisan.

#### **BAB IITINJAUAN PUSTAKA**

Bab ini menjelaskan teori-teori dari berbagai pustaka yang digunakan sebagai acuan dalam penulisan buku tugas akhir ini.

#### **BAB III**

#### **METODOLOGI**

Bab ini membahas alur pengerjaan tugas akhir dari awal sampai selesainya tugas akhir.

#### **BAB IV**

#### **PERANCANGAN DESAIN APLIKASI**



Bab ini menjelaskan rancangan desain aplikasi yang dibuat berdasarkan kebutuhan sistem. Desain tersebut digunakan untuk pembangunan aplikasi pada tugas akhir ini.

## **BAB V**

### **IMPLEMENTASI DAN UJI COBA SISTEM**

Bab ini menjelaskan pembangunan aplikasi yang sesuai dengan desain. Selain itu, dijelaskan pula uji coba sistem untuk memastikan fungsionalitas sistem sudah sesuai dengan desain yang telah dibuat.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan yang dapat diambil dari tugas akhir ini dan saran untuk kelanjutan sistem.

## BAB II TINJAUAN PUSTAKA

Sebelum melakukan penelitian tugas akhir, penulis melakukan tinjauan terhadap tulisan dari beberapa penelitian sebelumnya yang sesuai dengan tema yang diambil. Hasil tinjauan tersebut adalah sebagai berikut:

### 2.1. Administrator

*Definisi administrator antara lain:*

Pengguna pada modul Administrasi adalah:

1. Administrator/Admin tingkat Satuan Kerja (Satker), terdiri atas satu orang setiap tingkat Satker, ditunjuk langsung berdasarkan surat penugasan resmi oleh Kuasa Pengguna Anggaran (KPA) atau pejabat tertinggi tingkat Satker. Admin tingkat Satker dapat melakukan fungsi pengelolaan modul Admin secara *back end process*, yaitu mengelola referensi, pengaturan dan bantuan yang terhubung dengan modul-modul lainnya pada aplikasi SIAPKAN, atau,
2. Administrator/Admin pusat, terdiri atas satu orang atau lebih sesuai penunjukkan atau penugasan dari pejabat tertinggi berdasarkan surat keterangan resmi pada tingkat Kementerian/Lembaga (K/L) yang membawahi Satker-satker yang bersangkutan. Sebagai contoh K/L A memiliki 100 Satuan Kerja, maka pejabat tertinggi atau yang setingkat K/L A dalam hal ini Menteri, Direktur Jenderal, atau pejabat Plt., dapat menunjuk seseorang atau lebih sebagai Administrator Pusat. Admin Pusat memiliki fungsionalitas yang sama dengan Admin tingkat Satker, namun dapat mengelola database

berdasarkan fungsi “Create, Read, Update dan Delete” baik secara *back end* maupun pengelolaan basis datanya.

Sumber: <http://id.shvoong.com/social-sciences/communication-media-studies/2068222-pengertian-software-aplikasi/#ixzz1pSALzCe>, diambil pada 23 Maret 2012.

## 2.2. Satuan Kerja

### 2.2.1. Definisi Satker

Satuan Kerja (Satker) adalah Kuasa Pengguna Anggaran/Kuasa Pengguna Barang yang merupakan bagian dari suatu unit organisasi pada Kementerian Negara/Lembaga yang melaksanakan satu atau beberapa kegiatan dari suatu program. [3]

### 2.2.2. Pengelompokan Satker

Satker dapat dikelompokkan menjadi enam antara lain:

1. Satker Pusat, yaitu adalah satker yang kewenangan dan tanggung jawabnya melakukan kegiatan pengelolaan anggaran dalam rangka pelaksanaan tugas pokok dan fungsi kantor pusat Kementerian Negara/Lembaga yang lokasinya dapat berada di pusat dan atau di daerah.
2. Satker/Unit Pelaksana Teknis (UPT) Kementerian Negara/Lembaga, yaitu instansi vertikal di daerah yang kewenangan dan tanggung jawabnya melakukan kegiatan pengelolaan anggaran dalam rangka pelaksanaan tugas pokok dan fungsi yang berasal dari kantor pusat.
3. Satker khusus, yaitu satker yang ditetapkan untuk melaksanakan tugas pokok dan fungsi dalam melaksanakan program/kegiatan yang dibiayai dari Bagian Anggaran Pembiayaan dan Perhitungan. Contoh:
  - a. KONI untuk membantu pembiayaan kegiatan-kegiatan keolahragaan yang bersifat nasional dan internasional.
  - b. Dekopin untuk membantu pembiayaan operasional Dewan Koperasi Indonesia.
4. Satker Perangkat Daerah (SKPD), yaitu satker di provinsi yang melaksanakan tugas dekonsentrasi dan satker di

- provinsi/kabupaten/kota/desa yang melaksanakan tugas pembantuan.
5. Satker Non-Vertikal Tertentu (SNVT), yaitu satker yang bukan merupakan instansi vertikal Kementerian Negara/Lembaga yang melakukan kegiatan yang dibiayai dari alokasi anggaran Kementerian Negara/Lembaga yang bersangkutan. Sebagai contoh, PT Perusahaan Listrik Negara yang melaksanakan kegiatan pembangunan listrik pedesaan.
  6. Satker Sementara (SKS), yaitu satker di luar pengertian butir 2 sampai dengan 5, yang ditetapkan untuk melakukan kegiatan yang dibiayai dari alokasi anggaran Kementerian Negara/Lembaga yang kewenangan dan tanggung jawabnya berasal dari Kementerian Negara/Lembaga yang bersangkutan. [2]

### 2.3. Java

Java merupakan bahasa yang dikembangkan oleh SUN Microsystem. Desain java bertujuan untuk menyediakan kode untuk program yang dapat berjalan pada berbagai platform yang berbeda, menghasilkan program yang kuat (robust), yaitu program yang memiliki bug seminimal mungkin serta menciptakan bahasa pemrograman yang mudah dipelajari dan digunakan. [1]

Pada awalnya Java dikembangkan untuk pemrograman device kecil sehingga memiliki karakteristik berukuran kecil, efisien, dan portable. Karakteristik ini sangat sesuai dengan pengembangan internet sehingga dalam beberapa tahun belakangan ini Java telah menjadi primadona untuk pemrograman berbasis internet.[5]

Pengembangan aplikasi berbasis web dalam pemrograman Java menggunakan Servlet dan Java Server Pages (JSP). Servlet berfungsi untuk memperluas fungsionalitas sebuah server (server web, server aplikasi, server HTTP). Servlet adalah program Java yang diintegrasikan di dalam webserver untuk melakukan fungsi

server-side, yang dijalankan untuk menanggapi permintaan dari client yang berupa web browser. JSP merupakan perluasan dari teknologi servlet. Tujuan dari JSP adalah untuk lebih menyederhanakan penulisan servlet. Sebelum dijalankan oleh server, JSP akan dikompilasi terlebih dahulu menjadi servlet. Dibandingkan dengan servlet, kode yang ditulis untuk JSP relatif lebih ringkas dan proses deployment lebih mudah.[5]

#### 2.4. Servlets

Servlets adalah suatu kode aplikasi berbasis Java yang berjalan di web server. Servlets merupakan aplikasi yang berfungsi menjembatani antara client dengan server, dalam artian servlets meneruskan permintaan (request) dari web browser atau dari HTTP client menuju aplikasi atau database yang ada di server [4]. Pada saat meneruskan request, servlets mampu melakukan beberapa fungsi antara lain :

1. Membaca informasi yang dimasukkan oleh *pengguna*. Misalkan informasi login, informasi profil dan lain-lain.
2. Mengambil informasi tambahan yang disisipkan dalam HTTP *request*. Informasi ini bisa berupa *cookies*, kapasitas browser, nama host dan informasi lainnya.
3. Menghasilkan output. Servlets bisa berkomunikasi dengan database atau aplikasi di server misal mengirimkan query yang nanti akan menghasilkan informasi untuk diteruskan kembali ke client.
4. Mengatur hasil di dalam sebuah dokumen. Misalnya memberi informasi tambahan pada halaman HTML.
5. Menentukan parameter HTTP *response*. Berguna bagi browser karena bisa mengetahui dokumen apa yang dikirimkan, parameter caching, cookies pengaturan dan informasi lainnya
6. Mengirimkan dokumen ke client. Dokumen yang dikirimkan bisa berupa format HTML, gambar gif, bahkan dokumen yang dikompres dalam format zip.

Servlets juga berfungsi memperluas fungsionalitas dari server. Setelah request diterima, servlets menjalankan pengolahan permintaan dari sisi server (server-side). Karena fungsi pengolahan ada di sisi server maka output yang dihasilkan bisa bermacam-macam misal dalam format HTML, XHTML atau XML. Dengan demikian output yang dihasilkan servlets bersifat dinamis, dalam arti permintaan dari client yang berbeda akan menghasilkan content yang berbeda pula [10].

Servlets mempunyai banyak kelebihan dibandingkan dengan teknologi aplikasi web sebelumnya CGI. Keunggulan ini antara lain :

1. Efisien.

*Pada CGI, setiap HTTP request akan diproses satu per satu. Tiap satu proses itu akan menempati memori secara bergantian setelah proses sebelumnya selesai. Berbeda dengan servlets, ketika banyak HTTP request diterima meskipun ada banyak thread namun hanya satu servlet class yang ditempatkan di memori sehingga sangat menghemat memori.*

2. Tepat dan lengkap

Servlets mempunyai infrastruktu yang lengkap baik untuk fungsi parsing dan decoding data HTML, membaca dan mengatur header, mengatur cookies, session tracking dan banyak sekali utilitas yang mampu dikerjakan servlets.

3. Handal

Servlets mampu berkomunikasi langsung dengan web server. *Dibandingkan dengan CGI yang membutuhkan sebuah API server.* Servlets juga mampu menjaga informasi dari request satu ke request lainnya, menyederhanakan teknik session tracking dan menyembunyikan penghitungan lama.

4. Portabel

Servlets ditulis dalam bahasa pemrograman Java dan mengikuti standar API. Maka dari itu servlets dapat diaplikasikan ke dalam banyak web server misalnya Microsoft Internet Information Server, IBM WebSphere dan lain sebagainya.

#### 5. Aman

Bahasa pemrograman yang digunakan dalam CGI mempunyai kelemahan bahwa ada beberapa karakter yang diperlakukan khusus oleh server misalkan titik koma (;). Karakter ini bisa mengakibatkan gangguan jika disertakan dalam HTTP request dengan sengaja oleh client. Servlets telah terbebas dari masalah ini.

#### 6. Murah

Ada banyak sekali web server pada saat ini yang murah bahkan gratis yang kompatibel dengan servlets.

### 2.5. Java Server Pages

Teknologi Java Server Pages (JSP) adalah suatu halaman yang mampu menggabungkan halaman web yang regular, statik dan dinamis yang dihasilkan dari Servlets [5]. Perbedaan JSP dengan Servlets terletak pada kegunaannya, pada Servlets lebih menitikberatkan pada layer aplikasi atau logika proses dibandingkan dengan layer presentasi. Sedangkan pada JSP lebih menitikberatkan pada sisi presentasi kepada pengguna, bagaimana *suatu aplikasi ditampilkan kepada pengguna*.

JSP merupakan perluasan dari Servlets. Tujuan dari JSP adalah meringkas penulisan Servlets. Sebuah kode JSP sebelum dijalankan oleh server akan dikonversi ke Servlets terlebih dahulu. Sebuah halaman JSP dapat terdiri dari elemen khusus yang memungkinkan *server* memasukkan konten dinamis pada halaman tersebut. *Aplikasi yang dikembangkan menggunakan arsitektur client/server Model 2 yang memisahkan antara konten dengan logika yang digunakan untuk memanipulasi konten tersebut. Model 2 terdiri dari tiga proses yang terpisah [7], yaitu:*

1. *Browser*, berada pada sisi *client* yang menampilkan konten dinamis.
2. *Application Server*, menggunakan Servlet sebagai *controller* yang bekerja untuk menangani proses permintaan dari JSP lalu berkomunikasi dengan sumber data lalu menyatukannya

sebagai konten dinamis agar dapat diakses oleh JSP dan ditampilkan pada *Browser*.

3. *Data Source*, merupakan sistem manajemen data yang diperlukan untuk mengakses data dari *Application Server*.

Keunggulan JSP dibandingkan dengan Servlets adalah yang pertama penulisannya yang ringkas, kedua adalah JSP diperlakukan sama dengan file HTML sehingga proses deployment lebih cepat. Dalam sisi penggunaan JavaBeans, penulisan dalam format JSP akan lebih memudahkan programmer. Karena JSP mampu menggabungkan kode java dan konten seperti teks statik, kode HTML, DHTML, XHTML, XML dan kode lainnya dalam bentuk skrip atau tag [10]. Jadi konten tidak dihasilkan dari kode java namun bisa langsung dipakai seperti HTML. Berbeda dengan Servlets yang harus menuliskan konten dalam bahasa Java terlebih dahulu.

Secara umum JSP akan digunakan oleh programmer ketika sebagian besar dari suatu halaman membutuhkan banyak konten statis yang tidak berasal dari program Java misal HTML, XHTML atau XML. Sedangkan Servlets akan dipakai jika sebagian besar isi dari halaman merupakan output dari program Java atau output bersifat dinamik.

## 2.6. PostgreSQL

PostgreSQL merupakan salah satu sistem manajemen basis data yang berlisensi open source. PostgreSQL menggabungkan model relasional sebagai basis data dan mendukung penggunaan SQL query language. PostgreSQL mempunyai karakteristik performa yang baik dan dapat dijalankan pada berbagai platform serta mendukung berbagai penggunaan bahasa pemrograman.[8]

PostgreSQL mempunyai sistem otorisasi yang baik serta fleksibel sehingga membuatnya aman untuk digunakan. Selain itu dengan dukungan teknis serta komunitas yang besar membuat PostgreSQL stabil dan minim dari bug. PostgreSQL dapat



diandalkan untuk menampung data biner yang besar seperti foto atau video.

## 2.7. JDBC

Kelas-kelas Java Database Connectivity (JDBC) merupakan kelas-kelas dari Java yang memungkinkan sebuah aplikasi untuk mengirimkan pernyataan SQL ke sebuah Database Management System (DBMS) dan mengembalikan (retrieve) hasil. JDBC memiliki fungsi yang sama dengan Open Database Connectivity (ODBC). Salah satu keunggulan JDBC adalah interoperabilitas, yaitu seorang developer dapat membuat sebuah aplikasi tanpa menargetkan DBMS tertentu. Pengguna dapat menggunakan driver JDBC tertentu untuk menentukan target database secara spesifik.[10]

Antarmuka JDBC menyediakan aplikasi dengan seperangkat metode yang memungkinkan koneksi database, query, dan pengembalian hasil. Antarmuka JDBC adalah antarmuka antara driver database tertentu dan pengguna aplikasi Java.

JDBC terdiri atas beberapa komponen, yaitu :

- a. *Aplikasi* : Aplikasi pengguna memanggil metode JDBC untuk mengirim pernyataan SQL ke database dan mengambil hasil.
- b. *Driver Manager* : Tujuan utamanya adalah memuat driver khusus untuk aplikasi pengguna.
- c. *Driver* : Driver memproses pemanggilan method-method JDBC, mengirimkan pernyataan SQL ke sumber data tertentu, dan mengembalikan hasil ke aplikasi. Jika diperlukan, driver menerjemahkan dan / atau mengoptimalkan *request* sehingga *request* sesuai dengan sintaks yang didukung oleh suatu DBMS.

## 2.8. Arsitektur Model Controller View (MVC)

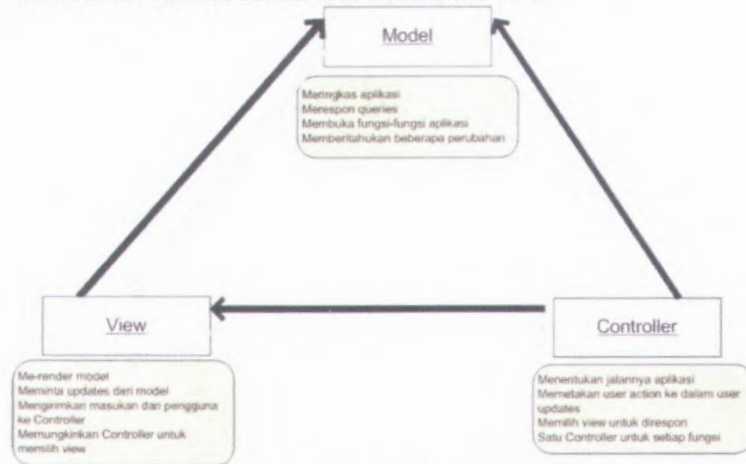
Arsitektur MVC adalah arsitektur yang memisahkan data aplikasi (terdapat dalam Model) dari komponen tampilan grafis (View) dan logika pemroses input (Controller). Controller

mengimplementasikan logika untuk memproses masukan dari pengguna. Model berisi data aplikasi dan view menghasilkan tampilan data yang disimpan dalam model (Deitel, Deitel, & Santry, 2001).

MVC memiliki beberapa keunggulan yaitu :

- MVC tidak membatasi sebuah aplikasi untuk sebuah *view* tunggal dan sebuah *controller* tunggal.
- Pengembang dapat memodifikasi tiap-tiap komponen secara sendiri-sendiri tanpa mengubah komponen lain.

Gambar 2.1 berikut adalah alur arsitektur MVC.



Gambar 2. 1-Arsitektur MVC

## **BAB III METODOLOGI**

Pada bagian ini diuraikan metode yang digunakan dalam penelitian tugas akhir. Metode ini digunakan sebagai panduan agar tahapan pengerjaan tugas akhir ini berjalan terarah dan sistematis. Tahapan metode penelitian dalam penyusunan tugas akhir ini yaitu:

### **3.1. Studi Literatur**

Studi literatur dilakukan untuk memahami konsep, teori dan teknologi yang akan digunakan. Pemahaman tersebut mengenai berkaitan dengan Undang Undang serta Peraturan yang terkait dalam pembuatan aplikasi. Selain itu juga dipelajari bahasa pemrograman serta perangkat lain yang mendukung pengembangan aplikasi. Literatur didapatkan melalui media buku, *halaman web, serta kumpulan peraturan perundangan.*

### **3.2. Analisis Kebutuhan**

Setelah memahami konsep dan teori yang mendukung pengembangan modul Admin untuk aplikasi SIAPKAN, maka dilakukan analisis kebutuhan sistem. Analisis kebutuhan dilakukan berdasarkan hasil pemahaman terhadap penelitian sebelumnya. Dari hasil analisis tersebut didapatkan gambaran yang jelas tentang modul Administrasi SIAPKAN yang dibutuhkan. Untuk lebih jelasnya, analisis kebutuhan dijelaskan pada bab 4 buku tugas akhir ini.

### **3.3. Perancangan Aplikasi**

Pada tahap ini dilakukan perancangan dan desain dari sistem aplikasi yang akan dikembangkan. Perancangan dan desain sistem mencakup proses bisnis serta mengacu pada integrasi dengan modul lain. Perancangan dan desain dari sisi Graphical Pengguna Interface/GUI akan menggunakan bantuan perangkat Photoshop dan GUI Desain Studio, sedangkan untuk desain sistem secara

model seperti Use Case, Class Model, dan Sequence Model akan digunakan Unified Modeling Language (UML). UML adalah bahasa standar yang digunakan untuk proses dokumentasi, pembuatan spesifikasi dan membangun sebuah sistem perangkat lunak.

### **3.4. Pembuatan Aplikasi/Coding**

Setelah penyelesaian tahap perancangan aplikasi, tahap selanjutnya adalah melakukan proses pembuatan aplikasi atau tahap pemrograman untuk modul Administrasi. Tahap ini akan menggunakan platform Java Server Pages dan PostgreSQL sebagai manajemen basis data. Metode scripting menggunakan MVC atau Model View dan Controller, tanpa bantuan *Framework*. Model berisi file *Daftar\*.java*, View berisi file *.jsp* dan Controller berisi file *controller\*.java* dan *\*Servlet.java*.

### **3.5. Integrasi Aplikasi**

Pada tahap ini proses integrasi terhadap Modul Administrasi dengan keseluruhan modul yang telah dibangun dalam aplikasi SIAPKAN. Proses integrasi dipermudah dengan adanya fitur *subversion* pada editor NetBeans 7.1.1. dan *code.google.com*.

### **3.6. Evaluasi/Uji Coba**

Untuk menguji kelayakan aplikasi SIAPKAN modul Administrasi, tahap selanjutnya adalah mengevaluasi dan menguji coba kinerja sistem yang berjalan didalamnya. Evaluasi ini dilakukan dengan melakukan uji coba menggunakan Java test unit serta output nilai. Output nilai yang dihasilkan aplikasi dibandingkan dengan penilaian manual (*human rater*).

Evaluasi ini bertujuan untuk melihat kesesuaian dengan desain yang telah dibuat serta mengetahui apakah terdapat kesalahan-kesalahan, baik secara tampilan informasi maupun kode program. Apabila ditemukan, maka akan dilakukan perbaikan pada aplikasi tersebut. *Proses ini akan terus dilaksanakan hingga mencapai kondisi program sesuai dengan standar kebutuhan dan desain awal.*

Tiga jenis uji coba yang akan dilakukan, yaitu:

1. Pengujian unit (unit test), merupakan pengujian yang dilakukan terhadap modul bendahara.
2. Pengujian integrasi (integration test), merupakan pengujian yang dilakukan untuk memastikan semua modul pada aplikasi SIAPKAN berjalan dengan baik.
3. Stress test, merupakan pengujian terhadap kemampuan aplikasi SIAPKAN ketika terjadi penggunaan dalam frekuensi dan volume besar.

### **3.7. Migrasi Data**

Setelah modul Administrasi dan tahap integrasi antar modul telah selesai dikerjakan dengan optimal, tahapan selanjutnya adalah *bagaimana memindahkan atau migrasi data dari sistem database aplikasi-aplikasi yang ada sebelumnya ataupun yang sedang existing kedalam aplikasi SIAPKAN sesuai dengan modul masing-masing*. Migrasi data pada aplikasi SIAPKAN khususnya modul Administrasi dilakukan pada dua kemungkinan:

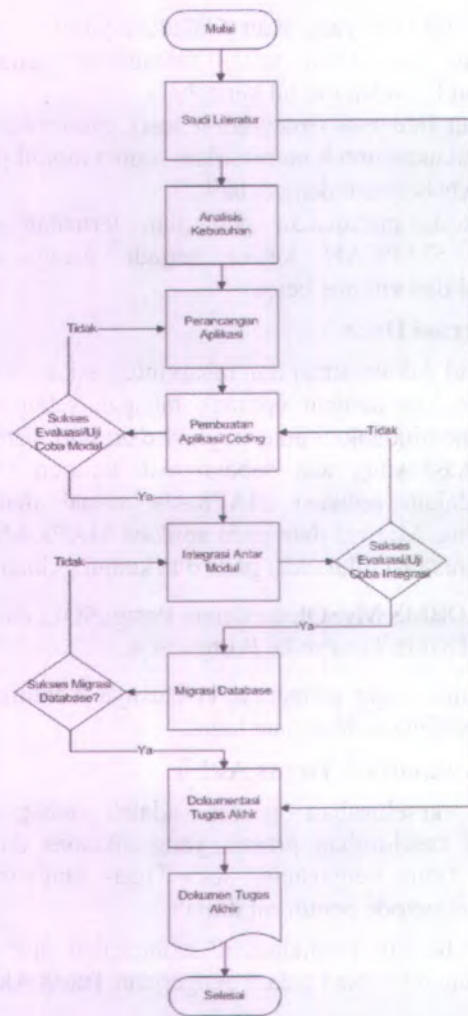
- a. Migrasi DBMS MySQL ke dalam PostgreSQL, dan kedua
- b. Migrasi DBMS FoxPro ke PostgreSQL.

Hal ini bertujuan agar pengguna, khususnya Administrator tidak perlu lagi menginput ulang data lama.

### **3.8. Dokumentasi Tugas Akhir**

Akhir dari keseluruhan proses adalah tahap pembuatan dokumentasi keseluruhan proses, yang dikemas dalam sebuah buku Tugas Akhir. Penyusunan buku Tugas Akhir ini dilakukan sesuai dengan metode penulisan ilmiah.

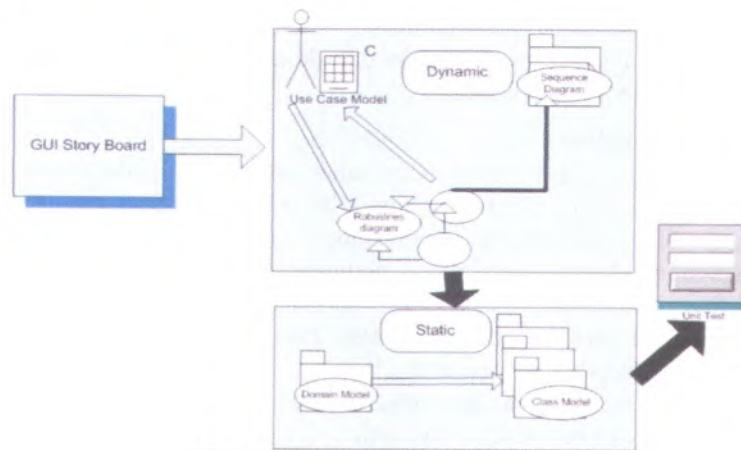
Gambar 3.1 berikut menjelaskan serangkaian alur metodologi penelitian yang dilakukan dalam pengerjaan Tugas Akhir ini.



Gambar 3. 1 Diagram alur metodologi penelitian tugas akhir

## BAB IV PERANCANGAN PERANGKAT LUNAK

Pada bab ini dijelaskan mengenai rancangan sistem aplikasi yang akan dibangun pada tugas akhir ini. Rancangan tersebut menggunakan referensi ICONIX process dengan alur pembuatan skema database, GUI(Graphical Pengguna Interface) story board, domain model, use case diagram serta deskripsi, robustness diagram, sequence diagram, class diagram, dan test case. Hal ini tergambar dalam Gambar 4.1 berikut.



Gambar 4. 1-Gambar Iconix Process

Desain sistem dibuat dengan mengacu pada kebutuhan aplikasi yang dibagi menjadi 2, yaitu fungsionalitas dan non fungsionalitas. Kebutuhan fungsionalitas aplikasi didefinisikan berdasarkan komponen-komponen yang dibuat, yaitu :

- a. Proses Login
- b. Interaksi dengan GUI Modul Administrasi
- c. Pengaturan Referensi
- d. Pengaturan Profil Pengguna

- e. Pengaturan Waktu Kerja/*Yulian Date*
- f. Bantuan
- g. Tentang SIAPKAN
- h. Proses Logout

#### 4.1. Analisis Kebutuhan

Untuk melakukan identifikasi kebutuhan maka dipergunakan Requirement Model (RQM). Dengan mempergunakan RQM akan dirinci secara detil beberapa kebutuhan sistem informasi. Rincian kebutuhan berdasar RQM meliputi :

- a. Kebutuhan yang dilengkapi dengan sekumpulan atribut yang terkait dengan kebutuhan tersebut (*Requirementsdocumentviews*)
- b. Hubungan antara kebutuhan dan desain obyek, file eksternal dan kebutuhan lainnya.

##### 4.1.1. Pengguna

Modul Administrasi merupakan salah satu modul pada aplikasi SIAPKAN yang digunakan untuk menyediakan data referensi bagi kebutuhan data dari modul-modul lainnya, serta melakukan pengaturan/konfigurasi yang berhubungan dengan pengguna setiap modul, *role*, waktu kerja, dan *backup* data. Pengguna dari modul Administrasi disebut sebagai Administrator/Admin, yang memiliki hak-hak khusus antara lain:

1. Pengelolaan data referensi, dengan proses tayang, rekam, ubah, dan hapus data referensi,
2. Pengelolaan profil pengguna, dengan proses tayang, rekam, ubah, dan hapus data pengguna,
3. Pengelolaan *role* pengguna, *role* inilah yang menentukan hak akses pengguna tiap modul, apakah dia sebagai Admin (*role* 1) hingga seorang kuasa pengguna Anggaran (*role* 9),
4. Pengelolaan *rightaccess*, yang artinya beberapa proses perekaman data membutuhkan hak dan batasan tertentu, misalnya untuk menandatangani sebuah laporan, maka jabatan adalah "pejabat" dan golongan minimum adalah "IIIb - Penata",



5. Seorang Administrator dapat melakukan fungsi *Backup* database, jika memang diperlukan sebagai bentuk manajemen resiko.

#### 4.1.2. Kebutuhan fungsional/Proses

Kebutuhan fungsional/proses pada modul Administrasi, secara lengkap dijelaskan pada daftar Use Case Modul Administrasi, namun secara garis besar kebutuhan fungsional/proses dari Modul Administrasi antara lain:

1. Proses penayangan data referensi (Umum, Aset, Bendahara, Persediaan, Anggaran, dan Pembayaran),
2. Proses pencarian data referensi (Umum, Aset, Bendahara, Persediaan, Anggaran, dan Pembayaran),
3. Proses perekaman/ubah data referensi (Umum, Aset, Bendahara, Persediaan, Anggaran, dan Pembayaran),
4. Proses hapus data referensi (Umum, Aset, Bendahara, Persediaan, Anggaran, dan Pembayaran),
5. Proses penayangan data pengaturan (Profil User dan Waktu Kerja),
6. Proses pencarian data pengaturan (Profil User dan Waktu Kerja),
7. Proses perekaman/ubah data pengaturan (Profil User dan Waktu Kerja),
8. Proses hapus data pengaturan (Profil User dan Waktu Kerja),
9. Proses penayangan data bantuan (Peraturan dan Konten),
10. Proses perekaman/ubah data bantuan (Peraturan),
11. Proses hapus data bantuan (Peraturan), dan
12. Proses Backup database.

Analisis hubungan antara kebutuhan dan pengguna (Admin) yang terlibat dalam kebutuhan tersebut, dapat dilihat pada tabel 4.1 Berikut ini.

**Tabel 4.1 Analisis Hubungan Kebutuhan dan Pengguna**

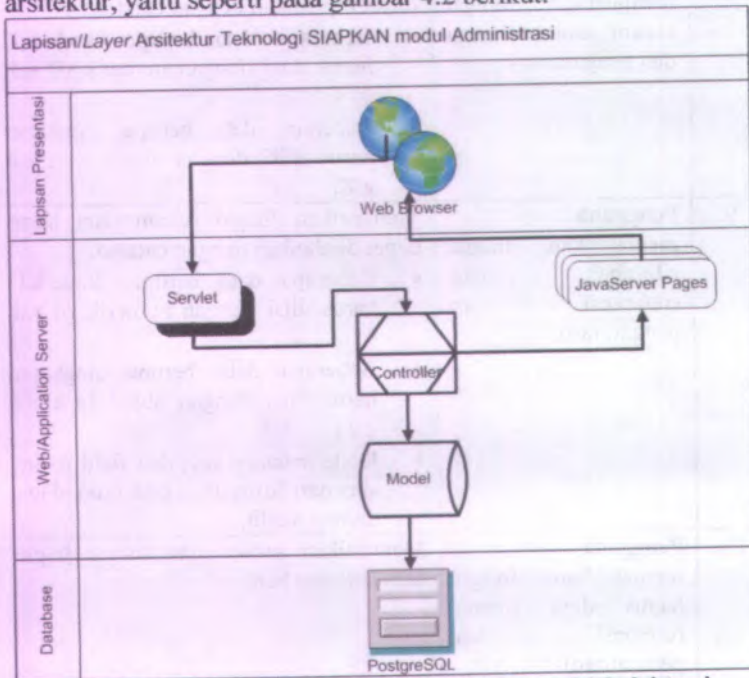
Analisis Kebutuhan		
No.	Implementasi	Analisis Kebutuhan

1.	Kebutuhan minimum perangkat keras	<ul style="list-style-type: none"> <li>• PC/Notebook/Netbook/Macbook berprosesor intel atau AMD,</li> <li>• Memory RAM/SODIMM minimal 1GB.</li> <li>• Memory penyimpanan minimal 10GB.</li> </ul>
2.	Kebutuhan minimum perangkat lunak	<ul style="list-style-type: none"> <li>• Sistem Operasi Windows XP/Vista/7 dan MacOS 8/9/X</li> <li>• <i>Browser</i> Firefox Mozilla ver.3 up., Safari, Chrome, Opera, Internet Explorer ver.6 up.</li> <li>• Mengizinkan java script pada <i>browser</i>, <i>cookies</i> dan histori.</li> <li>• Aplikasi DBMS PostgreSQL: Bitnami Wappstack dan PhpgMyadmin.</li> </ul>
3.	Pengguna menjalankan aplikasi SIAPKAN.	Memastikan kebutuhan minimum perangkat lunak dan perangkat keras tercapai.
4.	Pengguna menjalankan fungsi login.	<p>Memastikan bahwa:</p> <ul style="list-style-type: none"> <li>• Pengguna harus sudah memiliki username dan password yang terdaftar pada database.</li> <li>• Username dan password harus diisi.</li> <li>• Username dan password harus sesuai dengan data pada database.</li> <li>• Username yang diisi tidak boleh <i>berkarakter diluar angka dan atau abjad</i>.</li> <li>• Username dan password masuk ke dalam grup "role" tertentu.</li> </ul>
4.	Pengguna mengakses menu-menu modul Administrasi	Memastikan kebutuhan link dari tombol menu ke halaman dan proses yang dibutuhkan telah tersedia.
6.	Pengguna mengakses sub menu dari menu modul Administrasi	Memastikan kebutuhan link dari tombol ke halaman sub menu dan proses yang dibutuhkan telah tersedia.

7.	Pengguna menjalankan fungsi tayang (referensi, pengaturan, bantuan).	Memastikan fungsi tayang harus dapat dijalankan.
8.	Pengguna menjalankan fungsi rekam data (referensi dan pengaturan).	Memastikan fungsi rekam data harus dapat dijalankan dengan catatan: <ul style="list-style-type: none"> <li>• Beberapa data bertipe "kode/kd" harus diisi dengan numerik (0 s.d. 9)</li> <li>• Beberapa data berupa singkatan harus diisi dengan abjad (a/A s.d. z/Z)</li> </ul>
9.	Pengguna menjalankan fungsi edit/ubah data (referensi dan pengaturan).	Memastikan fungsi rekam data harus dapat dijalankan dengan catatan: <ul style="list-style-type: none"> <li>• Beberapa data bertipe "kode/kd" harus diisi dengan numerik (0 s.d. 9)</li> <li>• Beberapa data berupa singkatan harus diisi dengan abjad (a/A s.d. z/Z)</li> <li>• Kode <i>primary key</i>, dan field paling atas dari form ubah/edit tidak dapat diubah/diedit.</li> </ul>
10.	Pengguna menjalankan fungsi hapus data (menu referensi dan pengaturan)	Memastikan proses dan fungsi hapus data berjalan baik.
11.	Pengguna menjalankan proses backup data	Memastikan proses dan fungsi backup data dari postgre ke Java dan direktori berjalan baik.
12.	Pengguna melakukan Logout dari sistem	Memastikan proses <i>destroy session</i> berjalan baik.

## 4.2. Arsitektur Teknologi

Berdasarkan dokumen Use Case Driven Object Modeling with UML, desain arsitektur teknologi pada aplikasi SIAPKAN khususnya modul Administrasi dibagi menjadi tiga *layer/lapisan* arsitektur, yaitu seperti pada gambar 4.2 berikut.



Gambar 4.2 Arsitektur Teknologi SIAPKAN modul Administrasi

Dari gambar 4.2 diatas, pembagian komponen arsitektur aplikasi SIAPKAN dapat dijelaskan sebagai berikut:

### 4.2.1. Lapisan presentasi

Lapisan presentasi merupakan lapisan yang menghasilkan tampilan sebagai sarana interaksi antara pengguna dengan aplikasi. Karena Aplikasi SIAPKAN merupakan aplikasi berbasis web, lapisan presentasi yang digunakan adalah peramban / *web browser*.

#### 4.2.2. Lapisan web server / server aplikasi

Lapisan web server merupakan lapisan yang menghubungkan antara lapisan database dan lapisan presentasi. Lapisan ini terdiri atas JSP, servlet, controller, dan model.

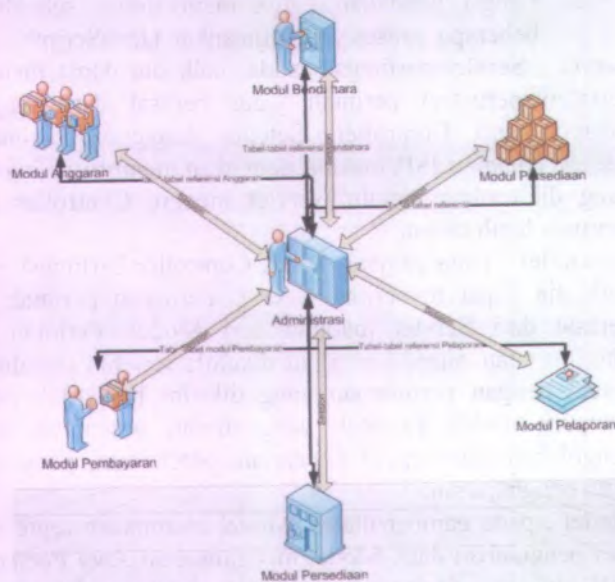
- a. Java Server Pages/JSP : halaman JSP digunakan sebagai tampilan yang berinteraksi langsung dengan pengguna aplikasi SIAPKAN (pada Modul Administrasi disebut Admin). Interaksi dapat berupa :
  - a. Form input/isian yang mengharuskan pengguna untuk menginput data
  - b. Daftar yang berisi informasi yang dilihat oleh pengguna dan dapat digunakan untuk penghubung untuk mengubah atau menghapus data yang ada pada daftar tersebut.
  - c. Fungsi tambahan untuk memvalidasi sebuah atau beberapa proses yang dijalankan (JavaScript).
- b. Servlet : Servlet berfungsi ganda, baik dia dapat menerima serta meneruskan perintah yang berasal dari pengguna maupun dari Controller. Setelah pengguna berinteraksi dengan halaman JSP, maka sistem akan memproses informasi yang dikirimkan melalui servlet menuju Controller untuk diproses lebih lanjut.
- c. Controller : pada gambar diatas, Controller berfungsi ganda, baik dia dapat menerima serta meneruskan perintah yang berasal dari Servlet maupun dari Model. Perintah yang diterima atau diteruskan akan dikelola terlebih dahulu agar sesuai dengan permintaan yang dikirim baik oleh Servlet ataupun Model. Perintah yang diolah umumnya berupa pengolahan data seperti pencarian, perekaman, pengubahan serta penghapusan.
- d. Model : pada gambar diatas, Model merupakan representasi dari pengolahan data. Model menggunakan *Java Persistence API* sehingga *Query* yang dikelola lebih mudah. Hal ini mempercepat permintaan yang diterima atau dikirim melalui Controller.

#### 4.2.3. Lapisan database

Database : pada gambar diatas, database merupakan gudang bagi seluruh data pada aplikasi. Data dipanggil dan ditransmisikan ketika ada permintaan dari model yang berupa *Query*. Pada pengembangan aplikasi SIAPKAN, teknologi database yang digunakan adalah PostgreSQL dengan perangkat lunak Bitnami Wappstack sebagai *interface*-nya.

#### 4.3. Hubungan Modul Administrasi dengan Modul Lain

Aplikasi SIAPKAN merupakan aplikasi *web based* yang dibagi menjadi tujuh buah modul. Keenam modul lainnya, selain modul Administrasi, memiliki keterkaitan dengan modul Administrasi yang dapat dilihat pada gambar 4.3 berikut.



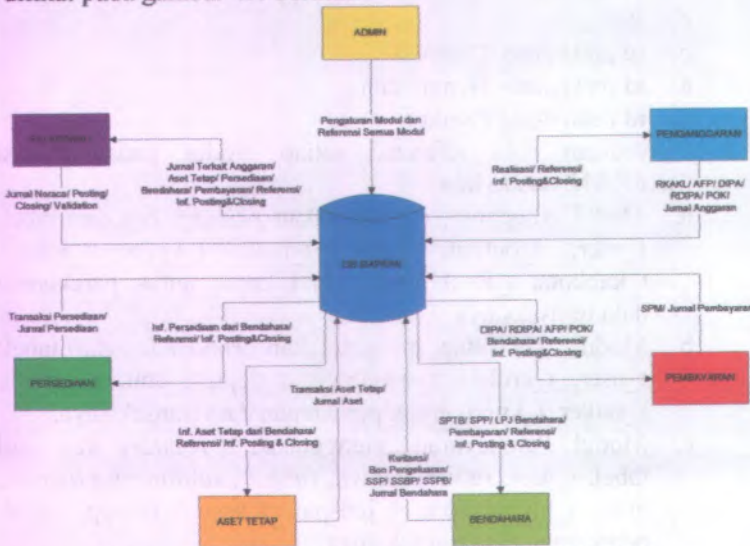
Gambar 4.3 Keterkaitan modul Administrasi dengan modul-modul lainnya  
Penjelasan untuk gambar 4.3 diatas adalah:

1. Keseluruhan modul pada aplikasi SIAPKAN menggunakan metode session yang nilainya khusus dari modul Administrasi pada saat proses login. Nilai session tersebut antara lain:
  - a. Username,
  - b. Password,
  - c. Role,
  - d. **id** pada class TKanwil
  - e. **id** pada class TKppn, dan
  - f. **id** pada class TSatker.
2. Penyediaan data referensi setiap modul pada aplikasi SIAPKAN, antara lain:
  - a. Modul Anggaran, menggunakan *primary key* dari tabel: *t\_user*, *t\_output*, *t\_giat*, *t\_mak*, *t\_kppn*, *t\_lokasi*, *t\_kabkota*, *t\_ib*, *t\_tema*, dan *t\_kppn*, untuk perekaman data transaksinya.
  - b. Modul Aset Tetap, menggunakan *primary key* dari tabel: *t\_user*, *t\_croleh*, *t\_sskelbmn*, *t\_dept*, *t\_unit*, *t\_kanwil*, *t\_satker*, *t\_kppn*, untuk perekaman data transaksinya.
  - c. Modul Pembayaran, menggunakan *primary key* dari tabel: *t\_user*, *satker*, *kanwil*, *fungsi*, *subfungsi*, *program*, *mak*, *t\_kppn*, *t\_spm*, *t\_jenspm*, *t\_spp*, *t\_jenspp*, untuk perekaman data transaksinya.
  - d. Modul Persediaan, menggunakan *primary key* dari tabel: *t\_dept*, *t\_unit*, *t\_wilayah*, *t\_satker*, *t\_barang*, *t\_sskelbmn*, *t\_kanwil*, *t\_perkiraan*, *t\_kppn*, untuk perekaman data transaksinya.
  - e. Modul Bendahara, menggunakan *primary key* dari tabel: *t\_bendahara*, *t\_mak*, *t\_bkpk*, *t\_gbkpk*, *t\_satker*, *t\_dept*, *t\_unit*, *t\_kppn*, untuk perekaman data transaksinya.

#### 4.4. Hubungan Antar Modul SIAPKAN

Pengembangan aplikasi Sistem Informasi Pengelolaan Keuangan Negara (SIAPKAN) merupakan pengintegrasian dari beberapa aplikasi yang telah dikembangkan oleh Direktorat Jenderal Perbendaharaan Negara (DJPBN) bagi para satuan kerjanya.

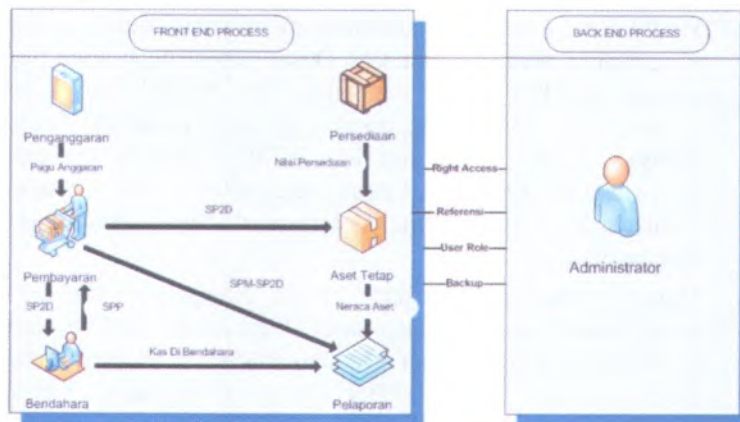
Pengembangan aplikasi SIAPKAN dibagi atas tujuh modul, yaitu Modul Administrasi, Modul Anggaran, Modul Aset Tetap, Modul Bendahara, Modul Pembayaran, dan Modul Persediaan. Untuk desain hubungan antar modul pada aplikasi SIAPKAN dapat dilihat pada gambar 4.4 berikut.



**Gambar 4. 4-Desain Hubungan Antar Modul SIAPKAN**

Untuk melihat keterkaitan antar modul dilihat dari sisi penggunaan data, dapat digambarkan seperti pada gambar 4.5 berikut.





Gambar 4. 5-Diagram keterkaitan antar modul

Hubungan atau keterkaitan antar modul dalam aplikasi SIAPKAN dapat dijelaskan sebagai berikut:

1. Modul Administrasi berfungsi untuk menangani kebutuhan referensi data untuk modul-modul lainnya. Selain itu Modul Administrasi berfungsi untuk menatausahakan pengguna dan *role*-nya sesuai dengan kebutuhan masing-masing Satuan Kerja. Output dari Modul Administrasi berupa referensi Anggaran, Aset, Persediaan, Bendahara dan Pembayaran serta manajemen pengaturan (pengguna dan waktu kerja) dan Backup data.
2. Modul Anggaran berfungsi untuk menyusun kertas kerja Rencana Kerja Anggaran Kementerian Lembaga (RKA-KL) di setiap Satuan Kerja (Satker). Modul Anggaran mendapatkan referensi dari Modul Administrasi sebagai dasar untuk pembuatan Kertas Kerja Pedoman Operasional Kinerja (POK) dan Daftar Isian Pelaksanaan Anggaran (DIPA). Output yang dihasilkan yaitu POK digunakan oleh Modul Bendahara sebagai dasar untuk menyusun transaksi pengeluaran. Sedangkan DIPA digunakan oleh Modul Pembayaran sebagai dasar pencairan dana APBN.

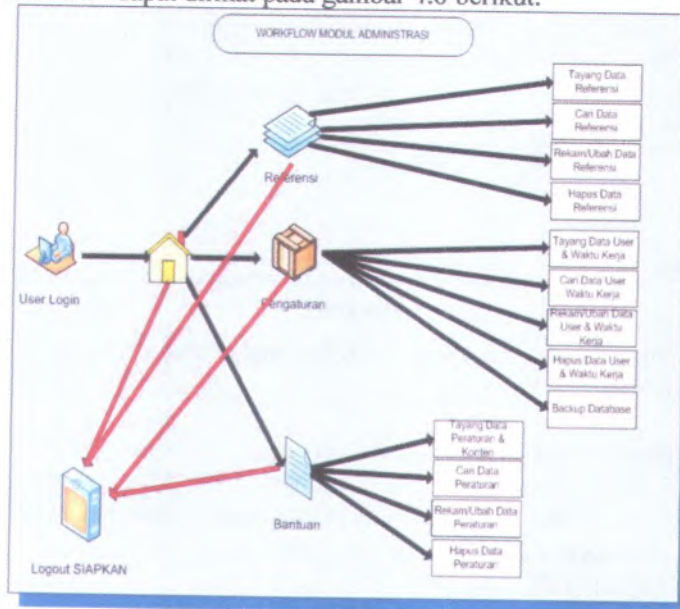
3. Modul Bendahara berfungsi untuk menatausahakan transaksi pengeluaran serta penerimaan. Dasar penyusunan transaksi berasal dari POK yang dihasilkan oleh Modul Anggaran. Output dari Modul Bendahara berupa Kuitansi, Surat Pernyataan Tanggung Jawab Belanja (SPTB) dan Pembukuan Bendahara. Output tersebut digunakan oleh Modul Pembayaran sebagai dasar pembuatan Surat Permintaan Pembayaran.
4. Modul Pembayaran berfungsi untuk memproses dokumen pembayaran yang berupa Surat Permintaan Pembayaran (SPP) dan Surat Perintah Membayar (SPM) sebagai dasar permintaan pembayaran APBN masing-masing Satuan Kerja. Dasar pencairan dana berupa DIPA berasal dari Modul Anggaran, sementara input yang digunakan untuk membuat SPP dan SPM berasal dari Modul Bendahara yang berupa Kuitansi dan Surat Pernyataan Tanggung Jawab Belanja. Selain itu Modul Pembayaran menggunakan input referensi yang berasal dari Modul Administrasi. Output SPP dan SPM digunakan oleh Modul Aset Tetap sebagai dasar pencatatan transaksi pembelian aset. Sedangkan untuk Modul Pelaporan digunakan sebagai dasar untuk menyusun jurnal, buku besar serta laporan keuangan tingkat Satuan Kerja.
5. Modul Aset Tetap berfungsi untuk menatausahakan aset tetap pada masing-masing Satuan Kerja (Satker). Modul Aset Tetap menggunakan data SPM yang telah tercatat nomor Surat Perintah Pencairan Dana (SP2D) untuk pencatatan transaksi pembelian serta mengambil nilai saldo persediaan dari Modul Persediaan untuk menyusun laporan posisi Barang Milik Negara (BMN).
6. Modul Persediaan berfungsi untuk menatausahakan barang habis pakai (persediaan) pada setiap satuan kerja. Modul persediaan menggunakan referensi kode barang, jenis transaksi, dan unit pengguna dari Modul Administrasi serta

menyediakan data nilai saldo persediaan bagi Modul Aset Tetap dalam penyusunan laporan posisi BMN.

7. Modul Pelaporan menerima posting jurnal dari modul-modul lain dan menampungnya dalam general ledger untuk digunakan menyusun laporan keuangan dengan didukung data referensi dari Modul Administrasi. Setiap periode, Modul Laporan mengirim feedback kepada modul lain setelah melakukan closing, bahwa data tidak boleh di edit lagi.

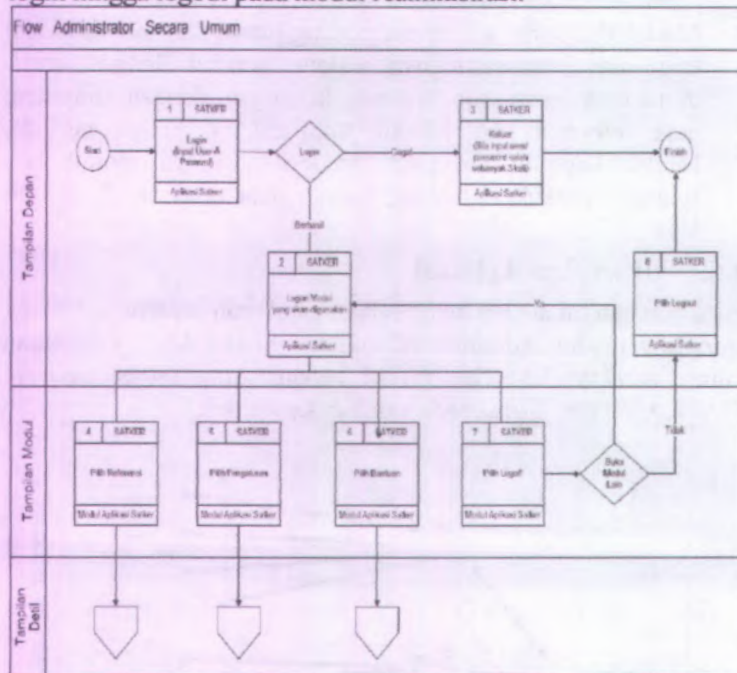
#### 4.5. Workflow Aplikasi

Pada sub bab ini adalah memberikan gambaran sederhana tentang *workflow* modul Administrasi aplikasi SIAPKAN. Gambaran umum *workflow*/alur kerja dari Modul Administrasi aplikasi SIAPKAN dapat dilihat pada gambar 4.6 berikut.



Gambar 4. 6-Workflow Modul Administrasi

Gambar 4.7 berikut memperlihatkan *workflow* proses user saat login hingga logout pada modul Administrasi.



Gambar 4. 7-Workflow User pada modul Administrator pada Modul Administrasi

Penjelasan untuk diagram *workflow* pada gambar 4.6 dan 4.7 diatas antara lain:

1. Login.  
proses masuknya pengguna ke dalam Aplikasi SIAPKAN. Setelah masuk ke dalam aplikasi, pengguna membawa *session* yang diperoleh pada proses login, terutama username, password, dan *role*.
2. Tayang data  
Pengguna dapat melakukan proses tayang data antara lain dari menu referensi, menu pengaturan, dan menu bantuan.

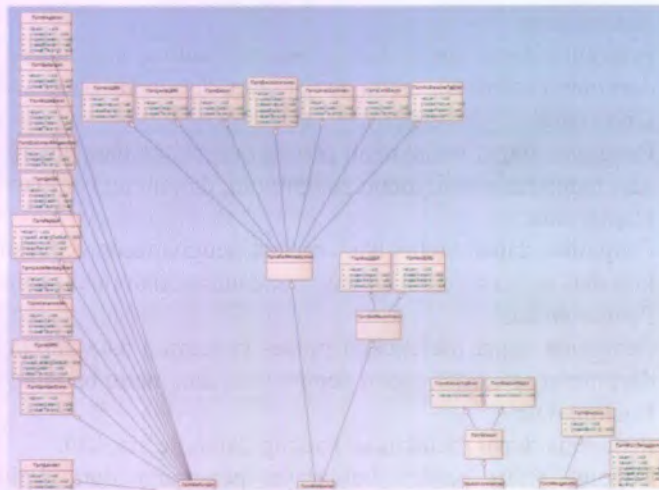
3. Rekam data  
Pegguna dapat melakukan proses perekaman data antara lain dari menu referensi, menu pengaturan, dan menu bantuan
4. Ubah data  
Pegguna dapat melakukan proses perubahan data antara lain dari menu referensi, menu pengaturan, dan menu bantuan
5. Hapus data  
Pegguna dapat melakukan proses penghapusan data antara lain dari menu referensi, menu pengaturan, dan menu bantuan
6. Pencarian data  
Pegguna dapat melakukan proses pencarian data antara lain dari menu referensi, menu pengaturan, dan menu bantuan
7. Backup data  
Pegguna dapat melakukan backup database siapkan.
8. Logout, yaitu proses keluarnya pegguna dari Aplikasi SIAPKAN. Proses logout akan menghapus *session* yang diperoleh pada proses login.

#### 4.6. Domain Model

*Domain model* merupakan komponen yang sangat penting karena domain model menggambarkan obyek-obyek utama yang akan digunakan pada tugas akhir. Domain model ini dapat berubah seiring pengembangan berupa penambahan obyek-obyek yang digambarkan domain model dari desain dan aplikasi agar sesuai dengan alur dari sistem.

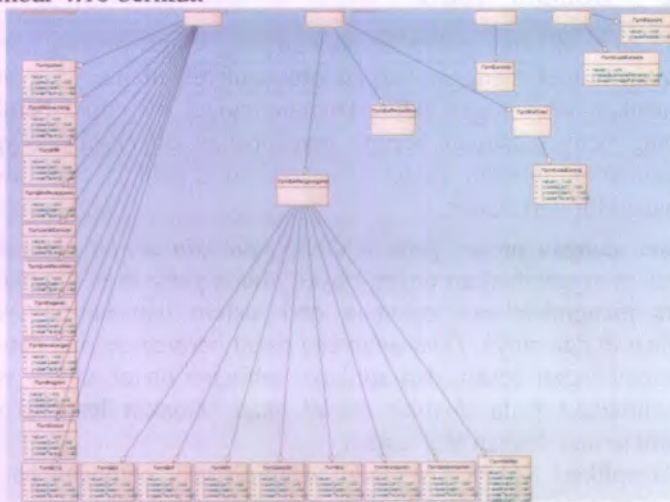
Sesuai dengan proses pada ICONIX, *domain model* digunakan untuk menggambarkan obyek-obyek utama yang akan digunakan serta menginisialisasi cakupan dari sistem dan hal-hal yang terlibat di dalamnya. *Domain model* dapat berubah seiring dengan pengembangan desain dan aplikasi, sehingga obyek-obyek yang digambarkan pada *domain model* akan semakin lengkap dan akurat sesuai dengan alur sistem.

Pada aplikasi SIAPKAN modul Administrasi, potongan bagian atas *domain model* dapat dilihat pada Gambar 4.8.



**Gambar 4. 8-Domain Model Modul Administrasi (bagian atas)**

Dan domain model bagian bawah dapat dilihat seperti pada gambar 4.10 berikut.



**Gambar 4. 9-Domain Model Modul Administrasi (bagian bawah)**

Beberapa *form* yang di inisiasikan oleh *domain model* pada gambar 4.8 dan 4.9 diatas antara lain:

1. Form Referensi Umum  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Referensi Umum dapat di sub kategorikan lagi menjadi form SATKER, BANK, CARA BAYAR, CARA TARIK, GOL & PANGKAT, KEGIATAN, KEWENANGAN, KPKNL, KPP, KPPN, LOKASI, MAK, OUTPUT, PEJABAT, PROGRAM, dan SUMBER DANA.
2. Form Referensi Aset  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Referensi Aset dapat di sub kategorikan lagi menjadi form CARA PEROLEHAN, ASET TETAP, dan SSKEL BARANG.
3. Form Referensi Anggaran  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Referensi Anggaran dapat di sub kategorikan lagi menjadi form AKUN, BLOKIR, DEPARTEMEN, GBKPK, UNIT, dan IN BUDGETING.
4. Form Referensi Bendahara  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Referensi Bendahara dapat di sub kategorikan lagi menjadi form BENDAHARA, FUNGSI, SUB FUNGSI, BEBAN, dan JENIS DOKUMEN
5. Form Referensi Persediaan.  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Referensi Persediaan dapat di sub kategorikan lagi menjadi form MAP BRG, BARANG, dan KANWIL.
6. Form Referensi Pembayaran  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Referensi Pembayaran dapat di sub kategorikan lagi menjadi form JENIS SPM, JENIS SPP, dan JENIS BEBAN

7. Form Pengaturan  
Merupakan salah satu komponen menu pada Modul Administrasi. Di dalam form Form Pengaturan dapat di sub kategorikan lagi menjadi form Profil User, Waktu Kerja, dan Backup Data.
8. Form Backup data  
Merupakan salah satu komponen menu pada Modul Administrasi. Form Backup data berfungsi men-*dump* data dari postgreSQL ke direktori C:\
9. Form Bantuan  
Merupakan salah satu komponen menu pada Modul Administrasi. Form Bantuan di sub kategorikan menjadi form Daftar Peraturan, dan Konten Modul.

#### 4.7. Manajemen Session

Pada sub bab Manajemen Session, secara garis besar membahas tentang penerapan metode *session*. Berikut adalah beberapa proses yang akan menggunakan *session* pada aplikasi SIAPKAN khususnya modul Administrasi.

##### 4.7.1. Manajemen Session Pengguna

1. *Pengguna* login menggunakan **username** dan **password** untuk mengakses aplikasi *web*.
2. Jika **username** dan **password** benar, atribut *session* akan di set pada *session* yang ada. Kemudian *pengguna* akan dilempar ke halaman utama. (dalam hal ini halaman utama, berbeda untuk setiap modul dan *pengguna* yang mengaksesnya.)
3. Setiap halaman yang akan diakses oleh *pengguna* harus divalidasi oleh *session*, jika ada *pengguna* yang mencoba mengakses sebuah halaman tanpa validasi, maka *pengguna* akan langsung dilempar ke halaman *login*.
4. Jika *pengguna* ingin keluar dari *session* yang ada, *pengguna* bisa mengakses sebuah halaman dimana didalam halaman tsb, *session* yang dipegang oleh aplikasi *web*, dihancurkan.



5. Beberapa file yang berkaitan dengan proses tersebut antara lain DaftarPegguna.java (model), TPengguna.java (entity), index.jsp, ceklogin.jsp, LoginServlet.java, LoginController.java dan web.xml. (tidak memerlukan file login.jsp)

#### 4.7.2. Manajemen Invoke Database

Pada Java, *session* didapatkan dari invokasi dari instans sebuah *servlet* karena setiap *request* ditangani oleh sebuah *servlet instance*. Selain itu biasanya beberapa melakukan koneksi *database* pada *method* di *servlet* dan melakukan pengecekan langsung dengan *query* ke *database*, seperti pada query berikut:

```
"SELECT COUNT (o) FROM o.TPegguna WHERE o.nmpengguna = 'sakti'
AND o.password = 'bersiap' GROUP BY o.username";
```

#### 4.7.3. Manajemen Logout

Untuk proses logout dapat dilakukan dengan menghilangkan *session*, yaitu cukup dengan method *invalidate* atau *removeAttribute* pada file java Servlet. Sebagai contoh *scriptsederhana* berikut.

```
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
```

```
throws ServletException, IOException {
    request.getSession().invalidate();
    response.sendRedirect("login");
}
```

#### 4.7.4. Manajemen Session Atribut

Pada pengelolaan profil pengguna, dibutuhkan *session* dari atribut-atribut pada tabel pengguna yang akan dibawa dan dipakai bersama pada pengguna setiap modul dan atau *role*-nya. Berikut ini adalah atribut yang dibawa oleh *session* pada saat Pengguna sukses melakukan proses login:

- Username,
- Password,
- Role,
- kdsatker



- tahun anggaran,
- kanwil, dan
- kppn

#### 4.8. GUI Story Board

Dalam sub bab GUI Story Board ini, akan memuat tampilan dan alur kerja aplikasi Modul Administrasi SIAPKAN pada saat dioperasikan. GUI Story Board dalam aplikasi ini memuat beberapa tampilan statis berupa tampilan menu-menu atau form-form yang disediakan di dalam modul Administrasi nantinya.

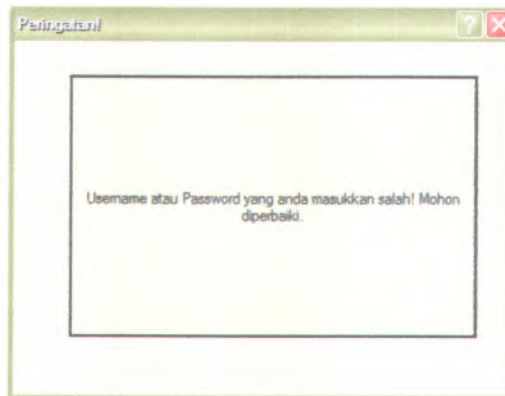
Berikut adalah desain tampilan pada menu-menu atau form-form modul Administrasi SIAPKAN dengan menggunakan aplikasi GUI Design Studio Professional.

Gambar 4.10 berikut ini menunjukkan tampilan form Login, yang terdiri dari field Username, Password, dan tombol Simpan.

**Gambar 4. 10-Tampilan form Login**

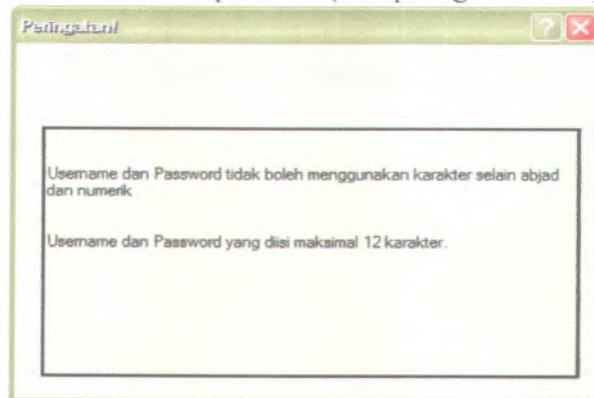
Ketika pengguna mengisi field Username dan Password, terdapat beberapa skenario hingga pengguna dapat memasuki halaman utama/Admin Home.

- Skenario login 1. Pengguna salah mengisi Username/ Password (lihat pada gambar 4.11)



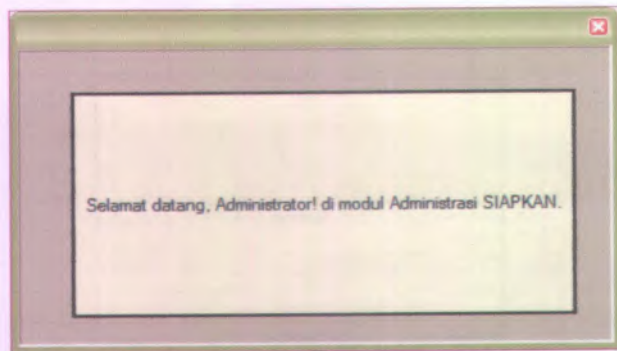
**Gambar 4. 11-Peringatan salah mengisikan username/password**

- Skenario login 2. Pengguna salah mengisikan tipe karakter pada username dan password (lihat pada gambar 4.12)



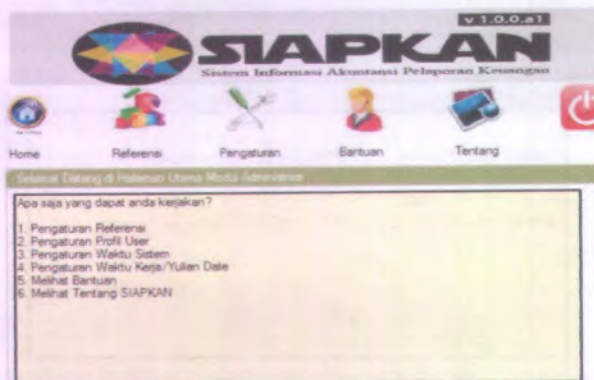
**Gambar 4. 12-Salah menginputkan karakter pada username dan password**

- Skenario login 3. Pengguna sukses mengisikan username dan password (lihat pada gambar 4.13)



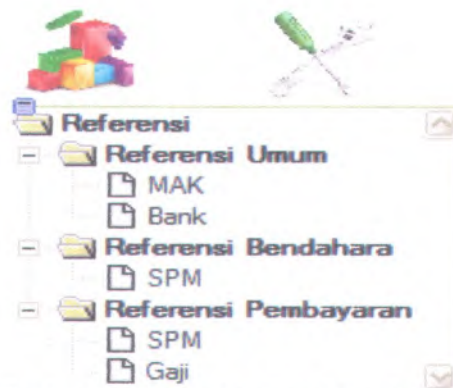
**Gambar 4. 13-Pengguna berhasil login**

Setelah Pengguna berhasil login, pengguna akan diarahkan secara otomatis ke halaman utama/Home Admin, seperti pada gambar 4.14 berikut.



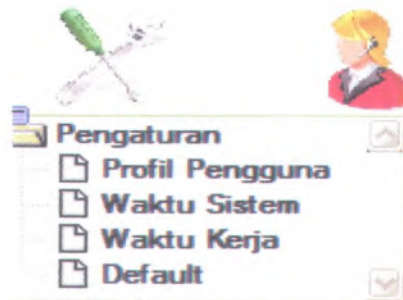
**Gambar 4. 14-Masuk pada halaman Utama Administrator**

Pada saat pengguna mengklik tombol "Referensi", pengguna akan diarahkan pada sub menu Referensi Umum, Referensi Bendahara, Referensi Persediaan, Referensi Anggaran, Referensi Aset Tetap, dan Referensi Pembayaran. Sebagai contoh dapat dilihat pada gambar 4.15 berikut.



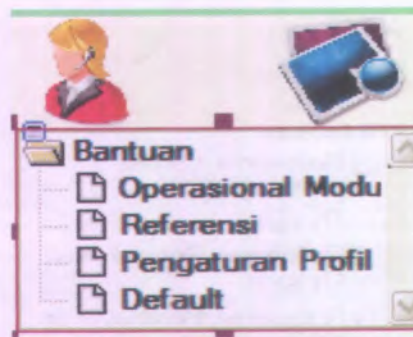
Gambar 4. 15-Desain Dropdown Menu Referensi

Apabila pengguna mengklik tombol Pengaturan, pengguna akan diarahkan pada fungsi sub menu Profil Pengguna, Waktu Sistem, Waktu Kerja, dan Default, sebagai contohnya pada gambar 4.16 berikut.



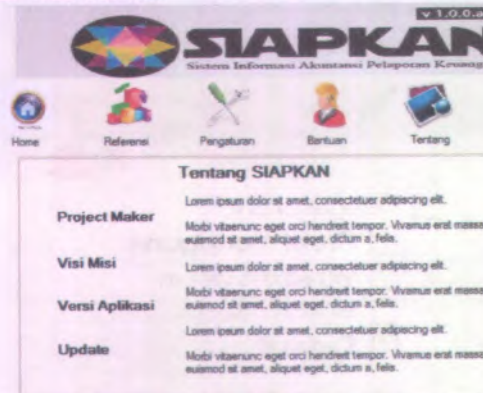
Gambar 4. 16-Desain Dropdown Menu Pengaturan

Pengguna dapat melihat berbagai jenis bantuan pada menu Bantuan yang terletak di sisi kanan menu Pengaturan seperti pada gambar 4.17 berikut.



Gambar 4. 17-Desain Dropdown Menu Bantuan

Selanjutnya untuk Menu terakhir adalah menu Tentang SIAPKAN, berisikan seputar informasi aplikasi SIAPKAN, antara lain: Project Maker, Visi Misi, Versi, dan Update. Seperti pada gambar 4.18 berikut.



Gambar 4. 18-Desain Tampilan Menu Tentang

Pada bagian ini akan dijelaskan tentang beberapa form tampilan proses rekam, ubah, dan hapus (RUH), mewakili keseluruhan form yang ada.

Pada saat pengguna mengklik tombol Referensi Umum → Bank, yang tampil pertama kali adalah form tayang dari Menu Referensi → Referensi Umum → Bank, seperti pada gambar 4.19 berikut.



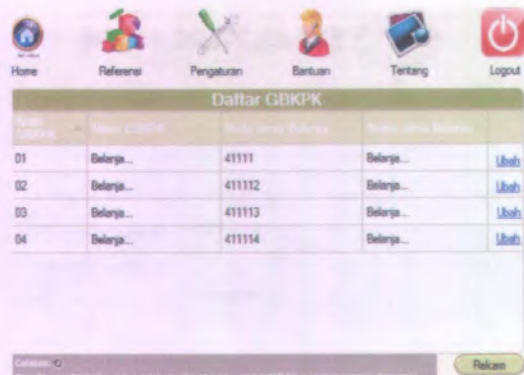
**Gambar 4. 19-Desain Tayang Referensi Umum – Bank**

Selanjutnya pengguna akan diarahkan kepada form Rekam Bank jika menekan tombol **Rekam** atau form Ubah jika menekan link **Ubah**, seperti pada gambar 4.20 berikut ini.



**Gambar 4. 20-Desain form Rekam/Ubah Bank**

Untuk Form Tayang pada saat pengguna mengakses menu Referensi GBKPK dapat dilihat pada gambar 4.21 berikut ini.



**Gambar 4. 21-Desain Tayang GBKPK**

Sedangkan Form rekam pada data referensi GBKPK dapat dilihat pada gambar 4.22 berikut.



**Gambar 4. 22-Desain Rekam/Ubah GBKPK**

Selanjutnya adalah desain GUI untuk Menu Pengaturan → Profil Pengguna, baik fungsi tayang maupun rekam, ubah dan hapus data, secara berurutan dapat dilihat pada gambar 4.23 dan 4.24 berikut.

Gambar 4.23 memperlihatkan desain proses tayang data User.





ID	Username	Password	Email	Role	Actions
01	saki	bersiap	411111	administrator	<a href="#">Ubah</a> <a href="#">Hapus</a>
02	admin	admin	411111	administrator	<a href="#">Ubah</a> <a href="#">Hapus</a>
03	bendahara	bendahara	411111	bendahara	<a href="#">Ubah</a> <a href="#">Hapus</a>
04	sesta	sesta	411111	sesta	<a href="#">Ubah</a> <a href="#">Hapus</a>

Daftar User Rekam

**Gambar 4. 23-Desain Tayang dan Hapus Pengguna**

Gambar 4.24 memperlihatkan desain proses ubah data User.



Rekam/Ubah User Simpan

Kode User: 01

Username: saki

Password: bersiap

Kode Satker: 41111

Role: Administrator

**Gambar 4. 24-Desain Rekam/Ubah Pengguna**

Selanjutnya, pengguna sebagai seorang administrator perlu untuk melakukan pengaturan waktu sistem dan waktu kerja/yulian date. Gambar 4.25 memperlihatkan desain setting data Pengaturan Waktu Kerja.

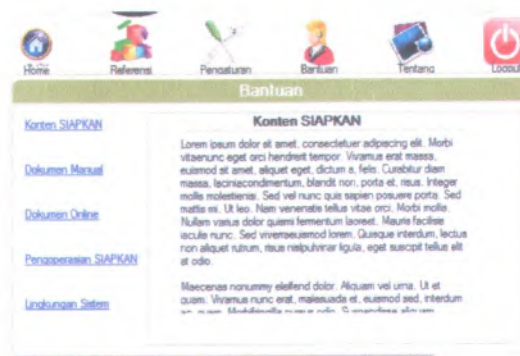
**Gambar 4. 25-Desain Pengaturan Waktu Kerja/Yulian Date**

Gambar 4.26 memperlihatkan desain tayang data Pengaturan Waktu Kerja.

No. Urut	Hari	Tanggal	Status	Aksi
01	Senin	01/01/2012	K	Ubah Hapus
02	Selasa	01/01/2012	K	Ubah Hapus
03	Rabu	01/01/2012	K	Ubah Hapus
04	Kamis	01/01/2012	K	Ubah Hapus
05	Jumat	01/01/2012	K	Ubah Hapus
06	Sabtu	01/01/2012	L	Ubah Hapus
07	Minggu	01/01/2012	L	Ubah Hapus
...	...	...	...	Ubah Hapus

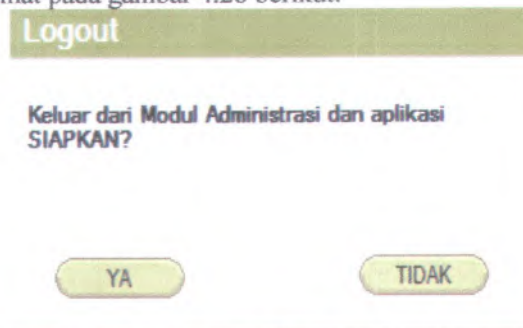
**Gambar 4. 26-Desain Daftar Waktu Kerja**

Bagi pengguna baru yang ingin mengetahui tata cara menjalankan aplikasi SIAPKAN, telah disediakan menu Bantuan, berikut desainnya tersaji pada gambar 4.27 berikut.



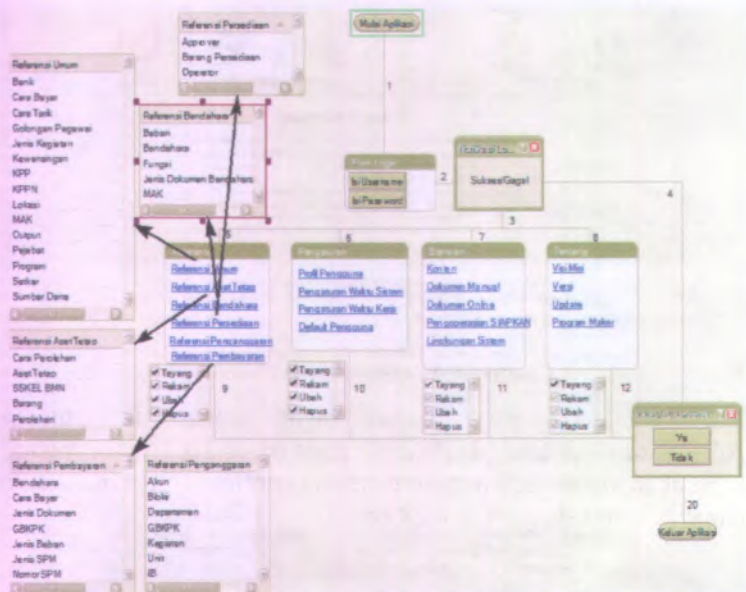
Gambar 4. 27-Desain Menu Bantuan

Bagi pengguna yang sudah ingin keluar dari modul Administrasi/aplikasi SIAPKAN, dapat mengklik tombol logout, yang akan memanggil pop-up windows konfirmasi logout. Desain dapat dilihat pada gambar 4.28 berikut.



Gambar 4. 28-Desain pop-up notifikasi logout.

Dari tampilan-tampilan desain menu modul Administrasi diatas, maka alur aplikasi dapat dibuat dengan menambahkan hubungan antar desain tampilan yang dapat diilustrasikan pada gambar 4.29 berikut.

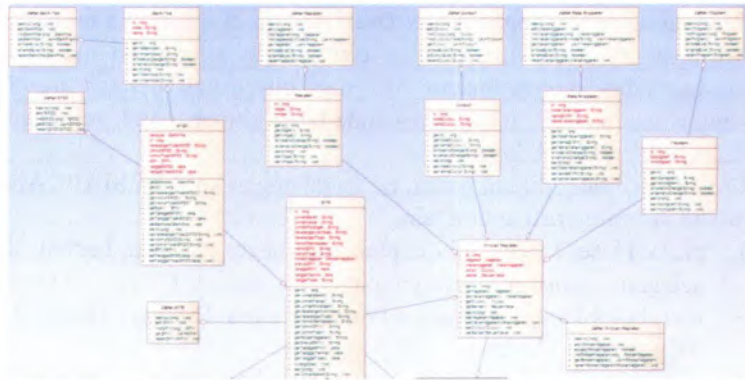


Gambar 4. 29-GUI StoryBoard

#### 4.9. Class Diagram

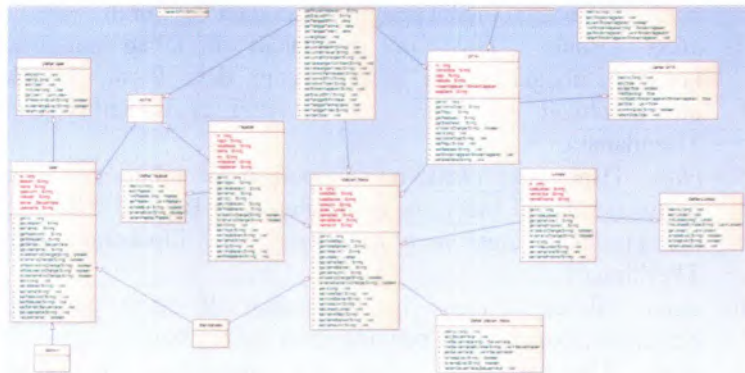
*Class Diagram* merupakan penjabaran dari *class* yang ada pada Java dan digunakan pada aplikasi SIAPKAN. *Class diagram* meliputi atribut serta operasi yang ada di dalam setiap *class* tersebut.

Untuk gambaran potongan bagian atas class diagram pada modul Administrasi dapat dilihat pada gambar 4.30 berikut.



**Gambar 4. 30-Potongan class diagram Modul Administrasi (bagian atas)**

Sedangkan gambaran potongan bagian bawah class diagram pada modul Administrasi dapat dilihat pada gambar 4.31 berikut.



**Gambar 4. 31-Potongan class diagram Modul Administrasi (bagian bawah)**

Pengembangan aplikasi SIAPKAN modul Administrasi menggunakan teknologi JPA, yang berarti pengembangan aplikasi SIAPKAN dapat menggunakan database tapi tidak perlu berurusan dengan struktur table ataupun SQL. Table dalam database akan di generate oleh JPA berdasarkan Entity class yang telah definisikan.

Pada tahap perancangan *Class Diagram* ini, di dalamnya terdapat perancangan *Database*, dengan ciri khusus penamaan masing-masing tabel menggunakan "T" dan diteruskan dengan huruf kapital setelah "T", sebagai penanda tabel khusus untuk referensi data pada modul Administrasi .

*Class-class* yang dibuat dalam pengembangan aplikasi SIAPKAN modul Administrasi antara lain:

1. *class* TUser : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data User. TUser membutuhkan relasi *many to one* dari *class* TSatker, TKanwil, TKppn, dan TLokasi.
2. *class* TAKun : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Akun. TAKun membutuhkan relasi *many to one* dari *class* TJenbel, Tgbkpk, dan TBkpk
3. *class* TAturan : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Peraturan.
4. *class* TBank : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Bank. TBank membutuhkan relasi *many to one* dari *class* TSatker dan TBendahara.
5. *class* TBarang : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Barang. TBarang membutuhkan relasi *many to one* dari tabel TJnsbarang, dan TPerkiraan.
6. *class* TBeban : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Beban.
7. *class* TBendahara : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Bendahara. TBendahara membutuhkan relasi *many to one* dari *class* TSatker.
8. *class* TBiaya : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Biaya.
9. *class* TBidang : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Bidang.

10. *class* TBkpk : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Matakuliah, *class* ini membutuhkan *class* BKPK.
11. *class* TBlokir : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Blokir.
12. *class* TCrbayar : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Cara Pembayaran.
13. *class* TCroleh : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Cara Perolehan.
14. *class* TCrtarik : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Cara Penarikan
15. *class* TDekon : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Dekonsentrasi
16. *class* TDept : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Departemen.
17. *class* TEntity : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Entitas Anggaran. TEntity membutuhkan relasi *many to onedari class* TSatker.
18. *class* TFuncsi : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Fungsi Anggaran.
19. *class* TGBkpk : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data GBKPK. TGBkpk membutuhkan relasi *many to onedari class* TJenbel.
20. *class* TGiat : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Kegiatan. TGiat membutuhkan relasi *many to onedari class* TUnit, TProgram, TFuncsi, dan TFuncsi.
21. *class* TGolbmn : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Golongan Barang Milik Negara.
22. *class* TGolpangkat : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Golongan Pangkat
23. *class* THarikerja : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Hari Kerja.

24. *class* Tfb : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Index Budgeting.
25. *class* TJenbel : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis Belanja
26. *class* TJendok : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis Dokumen
27. *class* TJenspm : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis SPM.
28. *class* TJenspp : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis SPP.
29. *class* TJenbeban : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis Beban. TJnsbeban membutuhkan relasi *many to onedari class* TSdana.
30. *class* TJnsgiat : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis Kegiatan.
31. *class* TJnsprogram : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Jenis Program
32. *class* TKabkota : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Kabupaten Kota. TKabkota membutuhkan relasi *many to onedari class* TLokasi.
33. *class* Tkanwil : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Kantor Wilayah. Tkanwil membutuhkan relasi *many to onedari class* TPebin.
34. *class* TKbkpk : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data KBKPK.
35. *class* TKetakun : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Keterangan Akun Anggaran. TKetakun membutuhkan relasi *many to onedari class* Tgbkpk.
36. *class* TKondisi : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Kondisi Barang.



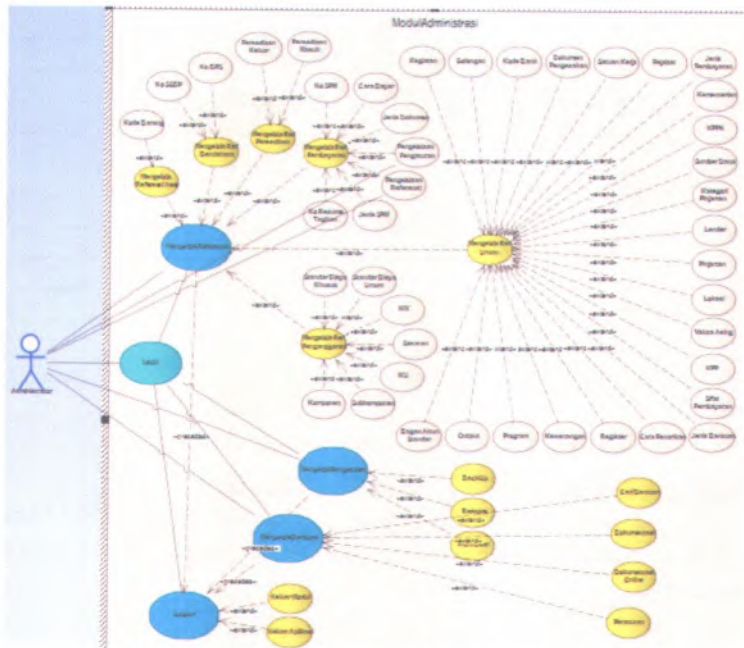
37. *class* TKpknl : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data KPKNL. TKpknl memiliki *foreign key* dari *class* TLokasi.
38. *class* TKpp : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data KPP. TKpp memiliki *foreign key* dari *class* TLokasi, TSatker, dan TKanwil.
39. *class* TKppn : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data KPPN. TKppn memiliki *foreign key* dari *class* TLokasi, TSatker, dan TKanwil.
40. *class* TLokasi : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Lokasi.
41. *class* TMak : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data MAK.
42. *class* TMapbrg : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data MAP Barang.
43. *class* TOutput : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Output. TOutput membutuhkan relasi *many to one* dari *class* TGiat.
44. *class* TPbi : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data PBI.
45. *class* TPebin : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data PEBIN.
46. *class* TPejabat : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Pejabat.
47. *class* TPeraturan : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Peraturan.
48. *class* TProgram : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Program.
49. *class* TSatker : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Satuan Kerja. TSatker membutuhkan relasi *many to one* dari *class* TKabkota, TUnit, TDept, TKppn, TKanwil, dan TLokasi.
50. *class* TSdana : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Sumber Dana.

51. *class* TSedia : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Persediaan.
52. *class* TSfungsi : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Sub Fungsi.
53. *class* TSpm : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data SPM.
54. *class* TSskel : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Sub-sub Kelompok Barang.
55. *class* TUnit : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Unit.
56. *class* TUniti : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Unit I.
57. *class* TUnitii : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Unit II.
58. *class* TUnitiii : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Unit III.
59. *class* TWilayah : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Wilayah.
60. *class* TYulian : *class* ini diciptakan di tahap pertama, berfungsi sebagai media penyimpanan data Waktu Kerja/YulianDate.

Beberapa dari daftar *class entity* diatas, membutuhkan *primary key* dari tabel lain yang kemudian menjadi *foreign key* di tabelnya, karena berhubungan dengan nilai atribut session yang akan dibawa oleh modul lainnya.

#### **4.10. Use Case Diagram**

*Use case diagram* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case diagram* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Gambar 4.32 dibawah adalah gambaran kesefuruhan *Use Case Diagram* modul Administrasi.

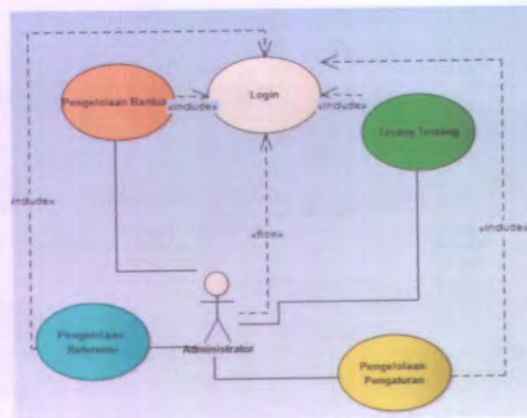


Gambar 4.32 Use Case Diagram modul Administrasi

*Use case diagram* yang dirancang harus memenuhi kebutuhan aplikasi Modul Administrasi yang telah disebutkan sebelumnya, terutama dalam kebutuhan fungsional. Dalam modul Administrasi, terdapat seorang aktor yang berinteraksi dengan system yaitu Administrator/Admin.

#### 4.10.1. Use Case level 1 modul Administrasi

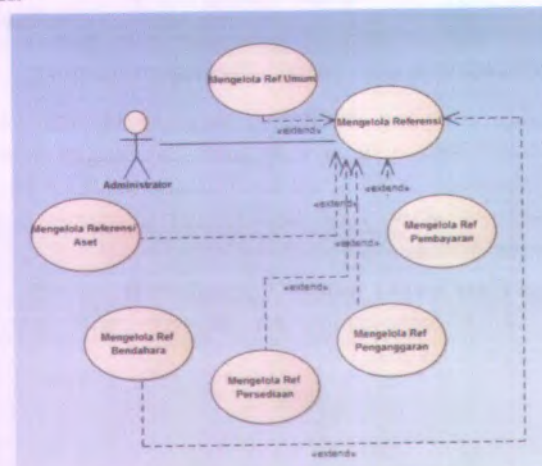
Gambar 4.33 menggambarkan diagram use case modul administrasi pada level 1.



Gambar 4. 33-Use Case level 1 modul Administrasi

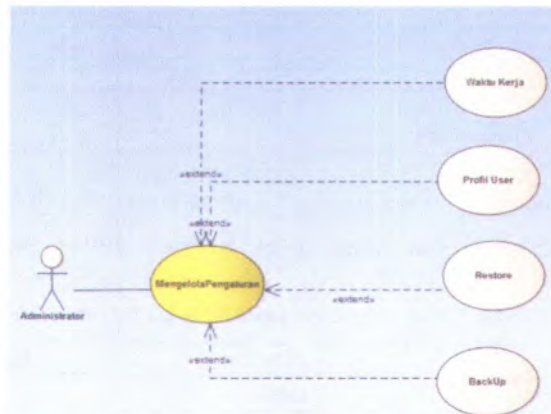
#### 4.10.2. Use Case level 2 modul Administrasi

Gambar 4.34 berikut menayangkan use case diagram level kedua aplikasi SIAPKAN modul Administrasi menu Pengelolaan Referensi.



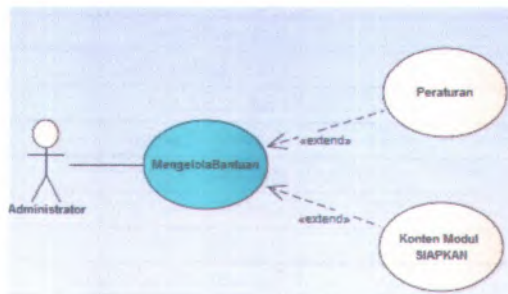
Gambar 4. 34-Use Case Diagram level 2: Pengelolaan Referensi

Gambar 4.35 berikut menayangkan use case diagram level kedua aplikasi SIAPKAN modul Administrasi menu Pengelolaan Pengaturan.



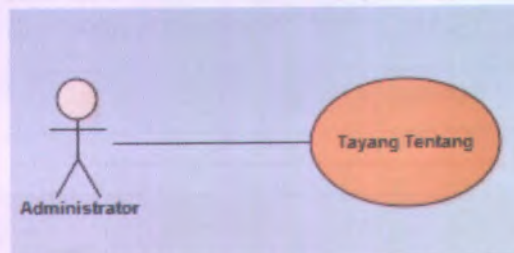
**Gambar 4. 35-Use Case Diagram level 2: Pengelolaan Pengaturan**

Gambar 4.35 berikut menayangkan use case diagram level kedua aplikasi SIAPKAN modul Administrasi menu Pengelolaan Bantuan.



**Gambar 4. 36-Use Case level 2: Pengelolaan Bantuan**

Gambar 4.34 berikut menayangkan use case diagram level kedua aplikasi SIAPKAN modul Administrasi menu Pengelolaan Tentang.



Gambar 4. 37-Use Case level 2: Tayang Tentang SIAPKAN

Untuk deskripsi Use Case level 3 dapat dilihat pada tabel 4.3berikut.

Tabel 4. 2Desain Use Case Level 3 Modul Administrasi

No.	Menu	Sub Menu	Kode UC
1.	<b>Login</b>	Login	UC-1
2.	<b>Referensi Bendahara</b>	Tayang Bendahara	UC-2
		Rekam Bendahara	UC-3
		Ubah Bendahara	UC-4
		Hapus Bendahara	UC-5
		Tayang Beban	UC-6
		Rekam Beban	UC-7
		Ubah Beban	UC-8
		Hapus Beban	UC-9
		Tayang Fungsi	UC-10
		Rekam Fungsi	UC-11
		Ubah Fungsi	UC-12
		Hapus Fungsi	UC-13
		Tayang jenis dokumen	UC-14
		Rekam Jenis Dokumen	UC-15
	Ubah Jenis Dokumen	UC-16	
	Hapus Jenis Dokumen	UC-17	
3.	<b>Referensi Pembayaran</b>	Tayang Mata Anggaran	UC-18
		Rekam Mata Anggaran	UC-19

No.	Menu	Sub Menu	Kode UC
		(MAK)	
		Tayang No. SPM	UC-20
		Rekam No. SPM	UC-21
		Ubah No. SPM	UC-22
		Hapus No. SPM	UC-23
		Tayang Cara Bayar	UC-24
		Rekam Cara Bayar	UC-25
		Ubah Cara Bayar	UC-26
		Hapus Cara Bayar	UC-27
		Tayang Jenis Dokumen	UC-28
		Rekam Jenis Dokumen	UC-29
		Ubah Jenis Dokumen	UC-30
		Hapus Jenis Dokumen	UC-31
		Tayang Bendahara	UC-32
		Rekam Bendahara	UC-33
		Ubah Bendahara	UC-34
		Hapus Bendahara	UC-35
		Tayang Jenis Beban	UC-36
		Rekam Jenis Beban	UC-37
		Ubah Jenis Beban	UC-38
		Hapus Jenis Beban	UC-39
		Tayang Jenis SPM	UC-30
		Rekam Jenis SPM	UC-31
		Ubah Jenis SPM	UC-32
		Hapus Jenis SPM	UC-33
<b>4.</b>	<b>Referensi Aset Tetap</b>	Tayang Aset Tetap	UC-34
		Rekam Aset Tetap	UC-35
		Ubah Aset Tetap	UC-36
		Hapus Aset Tetap	UC-37
		Tayang Cara Perolehan	UC-38
		Rekam Cara Perolehan	UC-39

No.	Menu	Sub Menu	Kode UC
		Ubah Cara Perolehan	UC-40
		Hapus Cara Perolehan	UC-41
		Tayang SSKel BMN	UC-42
		Rekam SSKel BMN	UC-43
		Ubah SSKel BMN	UC-44
		Hapus SSKel BMN	UC-45
<b>5.</b>	<b>Referensi Persediaan</b>	Tayang Map Barang	UC-46
		Rekam Map Barang	UC-47
		Ubah Map Barang	UC-48
		Hapus Map Barang	UC-49
		Tayang Kanwil	UC-50
		Rekam Kanwil	UC-51
		Ubah Kanwil	UC-52
		Hapus Kanwil	UC-53
<b>6.</b>	<b>Referensi Anggaran</b>	Tayang Akun	UC-54
		Rekam Akun	UC-55
		Ubah Akun	UC-56
		Hapus Akun	UC-57
		Tayang Blokir	UC-58
		Rekam Blokir	UC-59
		Ubah Blokir	UC-60
		Hapus Blokir	UC-61
		Tayang Departemen	UC-62
		Rekam Departemen	UC-63
		Ubah Departemen	UC-64
		Hapus Departemen	UC-65
		Tayang GBKPK	UC-66
		Rekam GBKPK	UC-67
		Ubah GBKPK	UC-68
		Hapus GBKPK	UC-69
		Tayang Kegiatan	UC-70



No.	Menu	Sub Menu	Kode UC
		Rekam Kegiatan	UC-71
		Ubah Kegiatan	UC-72
		Hapus Kegiatan	UC-73
		Tayang Unit	UC-74
		Rekam Unit	UC-75
		Ubah Unit	UC-76
		Hapus Unit	UC-77
		Tayang IB	UC-78
		Rekam IB	UC-79
		Ubah IB	UC-80
		Hapus IB	UC-81
		Tayang Output	UC-82
		Rekam Output	UC-83
		Ubah Output	UC-84
		Hapus Output	UC-85
<b>7.</b>	<b>Pengaturan</b>	Tayang Profil Pengguna	UC-86
		Rekam Profil Pengguna	UC-87
		Ubah Profil Pengguna	UC-88
		Hapus Profil Pengguna	UC-89
		Tayang Waktu Kerja	UC-90
		Rekam Waktu Kerja	UC-96
		Rekam Waktu Kerja	UC-97
		Hapus Waktu Kerja	UC-98
		Backup database	UC-99
<b>8.</b>	<b>Bantuan</b>	Tayang Konten	UC-100
		Tayang Peraturan	UC-101
		Rekam Peraturan	UC-102
		Ubah Peraturan	UC-103
		Hapus Peraturan	UC-104
<b>9.</b>	<b>Tentang</b>	Tayang Tentang Siapkan	UC-105
<b>10.</b>	<b>Logout</b>	Logout Aplikasi	UC-106

Selain dari Use Case diatas, terdapat pula beberapa kebutuhan non-fungsional seperti berikut :

- *Software* Pendukung

Beberapa *software* pendukung tentunya dibutuhkan dalam menjalankan aplikasi ini nantinya, antara lain:

1. Browser (Mozilla Firefox, Opera, Google Chrome, Safari),
2. Bitnami Wappstack PostgreSQL atau DBMS PostgreSQL lain.

Untuk rancangan diagram dan deskripsi *use case* dapat dilihat pada lampiran A.

#### 4.11. Robustness Diagram

*Robustness diagram* merupakan penjelasan alur dari *use case* yang dirancang. *Robustness diagram* dibuat di dalam seluruh *use case* yang ada dan mengacu pada diagram *use case* sebelumnya. *Robustness diagram* dapat dilihat pada Lampiran B.

#### 4.12. Sequence Diagram

*Sequence diagram* memuat alur dalam *use case* dengan pendeskripsian yang mengarah pada pemrograman aplikasi. Sehingga sebelum merancang *sequence diagram* terlebih dahulu harus mengerti tentang teknologi yang akan diterapkan. Rancangan *sequence diagram* dapat dilihat pada lampiran C.

#### 4.13. Test Case

*Test case* dirancang untuk menjaga performa aplikasi agar sesuai dengan desain yang dibuat. Dalam hal ini, *test case* akan dijalankan dengan beberapa skenario yang sesuai dengan rancangan pada diagram *use case* pada lampiran A. Untuk rancangan skenario dan *test* dapat dilihat pada lampiran D. *Test case* nantinya akan diuji coba berupa *unit test*.

## BAB V IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini akan dijelaskan mengenai implementasi dan uji coba pembuatan aplikasi berdasarkan rancangan yang telah disebutkan sebelumnya.

### 5.1. Lingkungan Implementasi

Pengembangan aplikasi SIAPKAN modul Administrasi ini dikembangkan di sebuah Laptop yang juga menjadi prototip *client* dari aplikasi tersebut dengan spesifikasi seperti pada tabel 5.1 berikut.

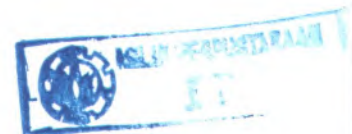
Tabel 5.1 Spesifikasi client

Spesifikasi
Prosesor : Intel® Core™i3-2350 CPU @2.30GHz 2.30 GHz
RAM/SODIM terinstal 2.00 GB
VGA: Intel Onboard
Sistem Operasi : Windows 7 Ultimate 64-bit (6.1,Build 7601)

Beberapa perangkat lunak juga dibutuhkan untuk mengimplementasikan aplikasi SIAPKAN modul Administrasi. Berikut pada tabel 5.2 adalah spesifikasi perangkat lunak yang digunakan.

Tabel 5.2 Spesifikasi Perangkat Lunak

Perangkat Lunak	
Bahasa pemrograman	Java (servlet dan JSP)
Development kit	JDK 1.6.0_06
Editor	Netbeans IDE 7.1.1
Web server	Apache Tomcat 7.0.22
ORM	JPA Persistence



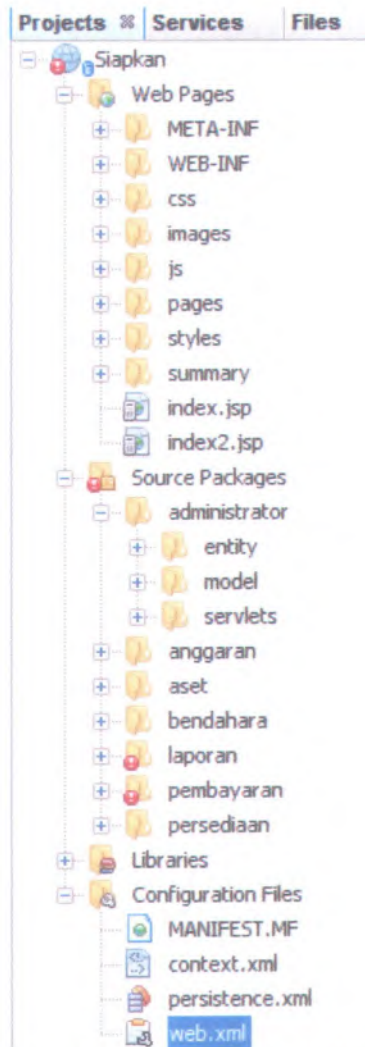
Framework	-
atabase	PostgreSQL 9.1
Tools lain	Bitnami wappstack, Full Converter Enterprise
Browser	Mozilla Firefox Versi 10

Aplikasi SIAPKAN merupakan aplikasi yang dikerjakan oleh beberapa orang dalam satu tim. Untuk mempermudah pembuatan program oleh tim, digunakan *subversion hosting* sebagai *repository* kode program yang dibuat. *Subversion hosting* yang digunakan dalam pengembangan Aplikasi SIAPKAN adalah google code dengan alamat <http://code.google.com/p/sistem-informasi-akuntansi-pelaporan-keuangan/>. Untuk username sesuai dengan email masing-masing

## 5.2. Struktur Project Directory

*Project directory* dari aplikasi SIAPKAN terdiri atas tiga direktori, yaitu direktori *nbproject*, *src*, dan *web*. Direktori *nbproject* berisi semua metadata mengenai proyek yang dibuat, yaitu *ant-deploy.xml*, *build-impl.xml*, *genfiles.properties*, *project.properties*, dan *project.xml*. Direktori *src* terdiri atas direktori *conf* untuk mengatur konfigurasi *persistence* dan direktori *java* yang berisi kode program yang dibuat. Direktori *java* kemudian dibagi berdasarkan modul-modul yang dibuat dalam Aplikasi SIAPKAN, yaitu *administrator*, *anggaran*, *aset*, *bendahara*, *laporan*, *pembayaran*, dan *persediaan*. Pembagian *project directory* SIAPKAN ini juga akan membantu memudahkan tim dalam melakukan migrasi dan integrasi data atau program. Untuk gambaran *Project directory* dapat dilihat pada gambar 5.1 berikut.



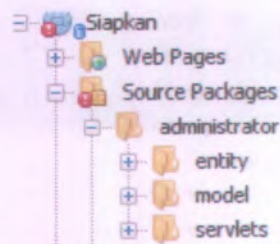


**Gambar 5. 1-Project directory SIAPKAN**

Untuk mendukung metode pemrograman MVC atau Model View Controller, maka setiap direktori Java Modul Administrasi dibagi

menjadi tiga sub direktori, yaitu **entity**, **model**, dan **servlet**. (lihat pada gambar 5.2)

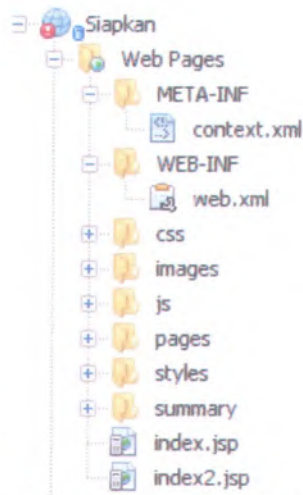
- a. Direktori **entity** berisi kode program *entityclass* tiap-tiap modul. Dari direktori **entity** inilah yang menjadi tabel-tabel dalam database PostgreSQL.
- b. Direktori model berisi kode program untuk proses *create*, *read*, *update*, *delete* (CRUD) dari setiap *entity class*. Model berisi eksekusi perintah dari program ke database.
- c. Direktori servlet berisi kode program untuk menampilkan halaman jsp dan implementasi proses CRUD pada menu-menu aplikasi. Servlet kemudian dibagi menjadi dua tipe yaitu Servlet, yang mengatur hasil atau output tampilan yang akan dihasilkan, dan Controller, yang mengatur proses apa saja yang akan dijalankan oleh Model sebelum memerintahkan database.



Gambar 5. 2-Pembagian direktori berdasarkan metode MVC

Selanjutnya adalah *web directory* yang berisi file-file yang digunakan untuk tampilan Aplikasi SIAPKAN berupa file \*.jsp, \*.css, dan javascript. Direktori ini dibagi lagi menjadi beberapa sub direktori yaitu meta-inf, web-inf, css, images, js, dan pages. (Lihat pada gambar 5.3)

Berikut ini adalah penjelasan dari masing-masing sub direktori dari *Web Directory*:



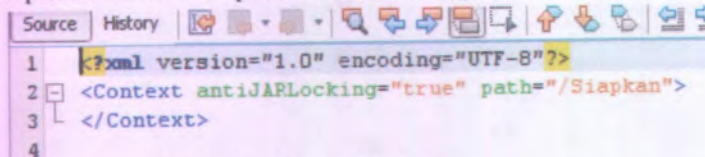
**Gambar 5. 3-Sub Direktori dari Web Directory**

- a. Direktori meta-inf berisi file context.xml yang berfungsi untuk menentukan alamat/url dari Aplikasi SIAPKAN.
- b. Direktori web-inf berisi file web.xml, yaitu file yang digunakan untuk mendefinisikan servlet beserta mapping-nya.
- c. Direktori css berisi file-file css untuk mengatur tampilan dari Aplikasi SIAPKAN.
- d. Direktori js berisi file-file javascript dengan fungsi terutama untuk menangani validasi pada *client-side* dan memberikan efek-efek khusus pada tiap-tiap halaman dalam Aplikasi SIAPKAN.
- e. Direktori pages berisi file jsp dari tiap-tiap modul. Direktori ini dibagi lagi menjadi sub-sub direktori sesuai dengan modul yang ada di dalam Aplikasi SIAPKAN, yaitu administrator, anggaran, aset, bendahara, laporan, pembayaran, dan persediaan.

### 5.3. File Konfigurasi

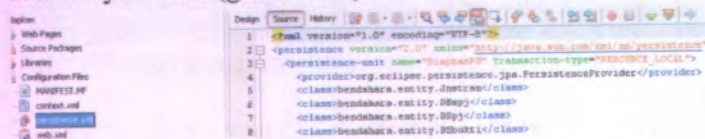
File konfigurasi pada proyek SIAPKAN terdiri atas context.xml, persistence.xml dan web.xml.

1. File Context.xml digunakan untuk mengatur alamat url Aplikasi SIAPKAN pada web browser.(gambar 5.4)



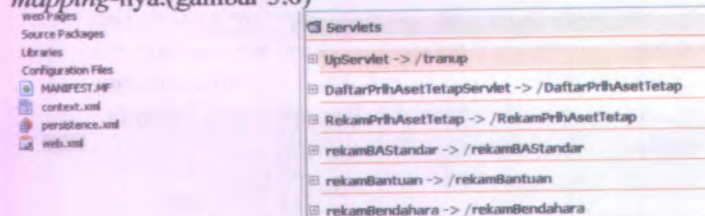
Gambar 5. 4-File Konfigurasi

2. File Persistence.xml digunakan untuk konfigurasi koneksi ke database PostgreSQL dan menampung semua kelas entitas yang ada pada proyek SIAPKAN. Persistence.xml akan *force* kelas entitas yang ada di dalamnya untuk membentuk sebuah tabel dalam database pada saat program dijalankan, sesuai dengan perintah yang dipilih, None, Create and Drop, atau hanya Create.(gambar 5.5)



Gambar 5. 5-File Persistence.xml

3. File Web.xml digunakan untuk mendefinisikan servlet beserta *mapping*-nya.(gambar 5.6)



Gambar 5. 6-File Web.xml

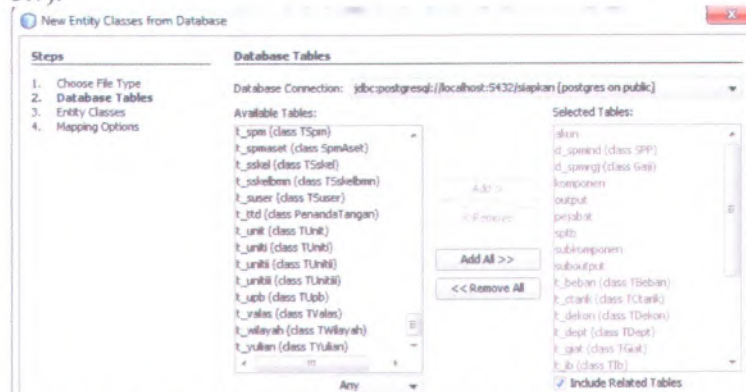


#### 5.4. Program Scripting

*Scripting* atau penulisan kode program dilakukan dengan menggunakan editor Open Source Netbeans IDE 7.1.1. *Scripting* kode program dilakukan pada source packages. Untuk memudahkan tim serta menata kode, maka *source packages* dibagi menjadi beberapa *sub package* berdasarkan modul-modul yang ada pada Aplikasi SIAPKAN, yaitu administrator, anggaran, aset tetap, bendahara, laporan, pembayaran, dan persediaan. Sesuai metode MVC, maka setiap package tiap-tiap modul dibagi lagi menjadi entity, model, dan servlet. Dalam modul Administrasi, sub package dibagi menjadi tiga bagian:

##### 1. Package administrator.entity

Package ini berisi *entity class* yang merepresentasikan data yang digunakan dalam Aplikasi SIAPKAN Modul Administrasi. *Entity class* juga merepresentasikan tabel-tabel yang ada dalam database siapkan. Terdapat dua metode dalam membuat *entity class*, yaitu scripting manual atau create entity baru dari database (gambar 5.7).



Gambar 5. 7-Membuat Entity Class dari database yang ada.

Salah satu contoh *entity class* modul administrasi yang berhasil dibuat dapat dilihat pada gambar 5.8.

```

@Entity
@Table(name = "t_bank")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "TBank.findAll", query = "SEL
    @NamedQuery(name = "TBank.findByKdbank", query =
    @NamedQuery(name = "TBank.findByNmbukubank", que:
    @NamedQuery(name = "TBank.findByThang", query =
    @NamedQuery(name = "TBank.findByKdbenda", query :
    @NamedQuery(name = "TBank.findByKdsatker", query
    })
public class TBank implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "kdbank")
    private String kdbank;
    @Basic(optional = false)
    @Column(name = "nmbukubank")
    private String nmbukubank;

```

Gambar 5. 8-Potongan entity class "TBank" diambil dari tabel t\_bank

## 2. Package administrator.model

Package ini berisi kelas-kelas JPA *controller* untuk setiap *entity class*. Kelas-kelas dalam package administrator.model digunakan untuk proses Tayang, Rekam, Ubah dan Hapus data (*create, read, update, delete / CRUD*) dan validasi pada *server-side*. Penamaan standar pada setiap kela package ini adalah "Daftar" diikuti dengan nama setiap *entity class*-nya. Sebagai contoh untuk TBank.java, maka *controller class*-nya adalah DaftarBank.java. Sebagai contoh dari package administrator.model adalah gambar 5.9 yang merupakan potongan DaftarBank.java

```

package administrator.model;

import administrator.entity.TBank;
import java.util.ArrayList;
import java.util.List;
import javax.persistence.*;

/**
 *
 * @author Momoncha
 */
public class DaftarBank {
    public DaftarBank() {
        emf = Persistence.createEntityManagerFacto.
    }
    private EntityManagerFactory emf = null;

    public EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    public List<TBank> getBank() {
        List<TBank> bank = new ArrayList<TBank>();
    }
}

```

Gambar 5. 9-Potongan file DaftarBank.java

### 3. Package administrator.servlets

Package ini berisi file *servlet* dan *controller* yang digunakan pada Aplikasi SIAPKAN Modul Administrasi. Pada package administrator.model ditambahkan beberapa class pendukung implementasi method-method didalamnya. Kemudian *servlet* memanggil method yang ada pada kelas pendukung ini sesuai dengan perintah yang akan dieksekusi.

Gambar 5.10 menunjukkan *servlet* untuk proses rekam, ubah, hapus (RUH) Referensi Bank, gambar 5.11 menunjukkan potongan kode pada kelas pendukung untuk *servlet* tersebut, dan gambar 5.12 menampilkan proses *controller* rekam data bank.

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException, NonexistentEntityException {
    String pageref = request.getParameter("pageref");
    HttpSession session = request.getSession();
    TUser user = (TUser) session.getAttribute("user");
    if (session.getAttribute("logged_in") == null || (Boolean) session.getAttribute("logged_in") == false) {
        request.getRequestDispatcher(redirect()).forward(request, response);
    } else if (user != null && !user.getRole().equals("U")) {
        request.getRequestDispatcher(forbidden()).forward(request, response);
    } else if (pageref.equals("rekamBank")) {
        session.removeAttribute("error");
        session.removeAttribute("sukses");
        session.removeAttribute("tempinput");
        request.getRequestDispatcher("/pages/administrator/Referensi_Duan/rekam_bank.jsp").
    } else if (pageref.equals("proses-rekamBank")) {
        session.removeAttribute("error");
        session.removeAttribute("sukses");
        session.removeAttribute("tempinput");
        BankController bankController = new BankController(request);
        request.getRequestDispatcher(bankController.prosesrekamBank()).forward(request, response);
    } else if (pageref.equals("ubahBank")) {
        session.removeAttribute("error");
        session.removeAttribute("sukses");
        session.removeAttribute("tempinput");
        //request.getRequestDispatcher("/pages/administrator/Referensi_Duan/edit_bank.jsp");
        BankController bankController = new BankController(request);
        request.getRequestDispatcher(bankController.ubahBank()).forward(request, response);
    }
}

```

Gambar 5. 10-Potongan file BankServlet

```

public void rekamBank(TBank bank) {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        em.persist(bank);
        em.getTransaction().commit();
    } finally {
        if (em != null) {
            em.close();
        }
    }
}

```

Gambar 5. 11-Potongan file DaftarBank untuk rekam data

```

public String prosesrekamBank() {
    TBank bank = new TBank();
    HttpSession session = request.getSession();
    String kodebank = request.getParameter("kodebank");
    String nama = request.getParameter("membukubank");
    String tahun = request.getParameter("tahun");
    String bendabera = request.getParameter("bendabera");
    String setker = request.getParameter("setker");

    if (kodebank != null) {
        bank.setKdBank(kodebank);
        bank.setBukubank(nama);
        bank.setThang(tahun);
        bank.setKdbenda(bendabera);
        bank.setKdsetker(setker);
        session.setAttribute("tempOutput", bank);
    } else {
        session.setAttribute("error", "Data Bank Gagal Disimpan.");
        return "pages/administrator/Pelaporan/Umum/validasi_bank.jsp";
    }
    if (kodebank.equals("") || nama.equals("")) {
        session.setAttribute("error", "Kode Bank atau Nama Bank Tidak Boleh Kosong.");
        return "pages/administrator/Pelaporan/Umum/validasi_bank.jsp";
    }
}

```

Gambar 5. 12-Potongan file BankController untuk proses rekam data bank

### 5.5. ImplementasiTampilan

Tampilan pada Aplikasi SIAPKAN khususnya modul Administrasi menggunakan Java Server Pages (JSP) dan CSS yang berfungsi untuk memperindah tampilan. Sedangkan untuk penambahan fitur atau validasi dari sisi *client* digunakan bantuan Javascript.

Implementasi tampilan halaman indeks pada modul Administrasi dapat dilihat pada gambar 5.12 berikut.



Gambar 5. 13-Tampilan form login/ index.jsp

Implementasi tampilan halaman *Home* pada modul Administrasi dapat dilihat pada gambar 5.14 berikut.



Gambar 5. 14 Halaman Utama Modul Administrasi

Implementasi tampilan halaman tayang referensi Bank pada modul Administrasi dapat dilihat pada gambar 5.15 berikut.



Gambar 5. 15-Halaman Tayang Referensi Bank

Implementasi tampilan halaman utama menu Pengaturan pada modul Administrasi dapat dilihat pada gambar 5.16 berikut.



**Gambar 5. 16-Halaman utama menu Pengaturan**

Implementasi tampilan halaman tayang Peraturan pada modul Administrasi dapat dilihat pada gambar 5.17 berikut.



**Gambar 5. 17-Halaman Bantuan tayang Peraturan**

Implementasi tampilan halaman tayang Tentang SIAPKAN pada modul Administrasi dapat dilihat pada gambar 5.18 berikut.

**SEKILAS SIAPKAN**

SIAPKAN merupakan singkatan dari Sistem Aplikasi Keuangan dan Pengelolaan Keuangan. SIAPKAN dirancang sebagai bagian dari Tugas Akhir para Pegawai Tugas Belajar DPRN pada ITS Surabaya ingkahan 2009. Terlepasnya database aplikasi pengelolaan keuangan karena melanda kebutuhan sebuah aplikasi pengelolaan keuangan yang terintegrasi, suatu aplikasi sistem yang terintegrasi yang dibangun guna mendukung implementasi Sistem Persendanaan dan Anggaran Negara (SPAN). Saat ini aplikasi SIAPKAN sudah hingga tahap rilis versi V1.5.

Direncanakan pada pertengahan bulan Juli 2012 aplikasi SIAPKAN sudah siap untuk dirilis dan diujicobakan untuk instansi/lembaga kerja DPRN. Kami berharap ada tahap pengembangan lebih lanjut demi penyempurnaan dari aplikasi SIAPKAN ini.

**TIM SIAPKAN**

Ahmad Sahrial	Pengembang dan Programmer Modul Aset Tetap	unuckiers@gmail.com
Ali Nasrun	Pengembang dan Programmer Modul Pelaporan	kalanda.mister@gmail.com
Alfa Satria Hidayat	Pengembang dan Programmer Modul Bendahara	alfahid@gmail.com
Harli Rahmanto Zauhar	Pengembang dan Programmer Modul Penganggaran	harlez.ro@gmail.com
Novan Andre Valen T.	Pengembang dan Programmer Modul Pembayaran	accioerry@gmail.com
Wahyudiono	Pengembang dan Programmer Modul Administrasi	kuromomoncha@gmail.com
Yoga Anief Priwanto	Pengembang dan Programmer Modul Persediaan	yeeriatpa4@gmail.com

**KEBUTUHAN MINIMUM SIAPKAN**

Sistem Operasi: Windows 95/98/NT/2000/XP/Vista/7 dan Mac OS 8/9/X

Gambar 5.18-Halaman Tentang

Implementasi tampilan halaman tayang Profil User pada modul Administrasi dapat dilihat pada gambar 5.19 berikut.

**DAFTAR PROFIL USER**

Revisikan Nama atau Data Profil User berikut:

Username	Role	Nama Lengkap	IDP	Priban
admin	1	Wahyudiono	081113708	<input type="checkbox"/>
appm	3	Novan Andre Valen	342525248	<input type="checkbox"/>
awa	2	Ahmad Sahrial	081004529	<input type="checkbox"/>
bendahara	5	Alfa Satria Hidayat	0810299887	<input type="checkbox"/>
glpa	6	Harli Rahmanto Zauhar	435348348	<input type="checkbox"/>
lpa	8	Kusna Pengguna Anggaran	23812825	<input type="checkbox"/>
pelaporan	8	Ali Nasrun	342525223	<input type="checkbox"/>
persediaan	7	Yoga Anief Priwanto	334534834	<input type="checkbox"/>
sdn	1	Wahyudiono	081113708	<input type="checkbox"/>
spm	4	Novan Andre Valen	34854834743	<input type="checkbox"/>

Gambar 5.19-Halaman Tayang Pengguna

Implementasi tampilan halaman rekam Profil User pada modul Administrasi dapat dilihat pada gambar 5.20 berikut.



**Halaman Rekam Profil User**

Tanggal Data:

Username:

Password:

Rekam Revisi: 00000 - BPK RI PERWAKILAN PROVINSI DAERAH ISTIMEWA YOGYAKARTA

Kantor Wilayah: Kantor Wilayah DJP Nangroe Aceh Darussalam

KORP: 00101 - BANDA ACEH

NIK:

Nama Lengkap:

Entitas Pengantar: U/s Juru Mudi

Jabatan User:  Admin  CR  Operator  Applier

Tahun Anggaran: 2011

**Gambar 5. 20-Halaman Rekam Profil Pengguna**

Implementasi tampilan halaman ubah Profil User pada modul Administrasi dapat dilihat pada gambar 5.21 berikut.

**Halaman Edit Profil User**

Uraian Data Utama:

Username: admin

Password: \*\*\*\*\*

Rekam Revisi: 00000 - BPK RI PERWAKILAN PROVINSI DAERAH ISTIMEWA YOGYAKARTA

Kantor Wilayah: Kantor Wilayah DJP Nangroe Aceh Darussalam

KORP: 00101 - BANDA ACEH

NIK: 000110700

Nama Lengkap: Wahyudin

Entitas Pengantar: U/s Juru Mudi

Jabatan User:  Admin  CR  Operator  Applier

Tahun Anggaran: 2011

**Gambar 5. 21-Halaman Edit Profil Pengguna**

## 5.6. Pembuatan Aplikasi

Dalam sub bab ini berisi penjelasan, metodologi, serta standarisasi mengenai pembuatan aplikasi SIAPKAN mulai dari pembuatan form login/index, halaman home Admin, halaman Referensi, halaman dengan proses Tayang, Rekam, Ubah dan Hapus data, hingga proses logout dari aplikasi.

### 5.6.1. Halaman Index/form login

Halaman index/form login dibuat dengan menggunakan JSP dan CSS untuk memperindah tampilan, dengan tambahan JavaScript sebagai validator *client-side*. Konsepnya adalah sebuah halaman indeks yang terkesan klasik, namun elegan, maka digunakanlah *template* dengan warna utama perak/silver.

Untuk potongan bagian atas script JSP pada index.jsp dapat dilihat pada gambar 5.22 berikut.

```

<%@page import="administrator.model.DaftarUser"%>
<%@page import="administrator.model.DaftarKppn"%>
<%DaftarUser dUser = new DaftarUser();%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>SIAPKAN v1.0</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<script type="text/javascript">

function validateForm() {
    var a=document.forms["login"]["username"].value;
    var b=document.forms["login"]["password"].value;
    if (a=="||b==" ) {
        alert("Username dan Password Tidak Boleh Kosong.");
        return false;
    }
}

function isNumberKey(evt) {
    var charCode = (evt.which) ? evt.which : event.keyCode
    if (charCode > 31 && (charCode < 48 || charCode > 57) ) {
        return false;
    }
}

return true;

```

Gambar 5. 22-bagian atas (head) script index.jsp

Sedangkan untuk potongan bagian atas script JSP pada index.jsp dapat dilihat pada gambar 5.23 berikut.

```

<form method="post" action="login" style="width: 100%; border: 1px solid #ccc; padding: 10px; margin: 0 auto; text-align: center;">
<table border="1" style="width: 100%; border-collapse: collapse;">
<tr>
<td>
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;">
<input type="text" value="username" style="width: 95%; height: 20px; border: none; border-bottom: 1px solid #ccc; margin-bottom: 5px;"/>
<input type="password" value="password" style="width: 95%; height: 20px; border: none; border-bottom: 1px solid #ccc; margin-bottom: 5px;"/>
<input type="submit" value="Login" style="width: 100%; height: 20px; border: none; border-top: 1px solid #ccc; margin-top: 5px;"/>

```

Gambar 5. 23-bagian bawah (body) index.jsp

Setelah project SIAPKAN *running* akan menggenerate file CSS, JSP dan JavaScript di dalam file index.jsp dengan default url laman adalah /Siapkan, seperti pada gambar 5.24 berikut.



Gambar 5. 24-Index.jsp setelah running project SIAPKAN

### 5.6.2. Manajemen Login

Untuk manajemen login, perlu memperhatikan beberapa aturan validasi seperti berikut:

1. Input username tidak diizinkan menggunakan karakter diluar aA – zZ atau 0 – 9, sedangkan password diizinkan,
2. Setiap pengguna hanya dapat masuk ke modul sesuai dengan *role* modulnya masing-masing
3. Input username dan password tidak boleh kosong,
4. Username dan password tidak boleh salah atau ada di dalam database,
5. Username dan password adalah case sensitive, artinya jika di dalam database tersimpan “pengguna”, maka input “Pengguna” tidak diizinkan untuk login.
6. Tidak diizinkan mengakses halaman modul sebelum login

Untuk pengaturan validasi diatas maka perlu dibuat beberapa metode pengkodean yang menggunakan session. Session membantu membawa atribut dari satu objek yang akan dipanggil/digunakan oleh objek/method lain.

### 5.6.3. Implementasi Login

Untuk implementasi login adalah dengan membuat file servlet bernama LoginServlet yang berisi:

1. Request parameter dari isian username dan password, dan membawa method check dari model.DaftarPengguna, seperti pada gambar 5.25 dan 5.28 berikut:

Gambar 5.25 sebagai potongan class LoginServlet

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException, NonexistentEntityException {
    PrintWriter out = response.getWriter();
    String username = request.getParameter("username");
    String password = request.getParameter("password");
    DaftarUser users = new DaftarUser();
    //TUser user = users.getUserLogin(username, password);
    boolean resultCheck = users.check(username, password);
    HttpSession session = request.getSession(true);
```

Gambar 5. 25-Potongan LoginServlet, request parameter username dan password

Gambar 5.26 sebagai potongan class DaftarPengguna.

```

public boolean check(String username, String password) {
    boolean result = false;
    EntityManager em = getEntityManager();
    try {
        Query q = em.createQuery("SELECT COUNT(*) FROM TPengguna WHERE username = :username AND password = :password");
        q.setParameter("username", username);
        q.setParameter("password", password);
        int jumlahUser = ((Long) q.getSingleResult()).intValue();
        if (jumlahUser == 1) {
            result = true;
        }
    } finally {
        em.close();
    }
    return result;
}

```

**Gambar 5. 26-Potongan DaftarPengguna.Java, method boolean check**

Dengan menggunakan method boolean check, Username dan Password di dalam entity TPengguna akan di cek jumlahnya harus 1, jika benar, maka akan dilanjutkan ke dalam request session `HttpSession session = request.getSession(true);`, jika hasilnya "true", maka akan dilanjutkan pada iterasi selanjutnya.

2. Membuat iterasi validator jika username atau password kosong/null. Dan membuat/set session baru bernama "logged\_in. dapat dilihat pada script berikut.

```

if (username.isEmpty() || password.isEmpty()) {
    request.setAttribute("error", "Username atau Password Tidak Boleh Kosong.");
    RequestDispatcher rdp =
    request.getRequestDispatcher("index.jsp");
    rdp.forward(request, response);
    session.setAttribute("logged_in", false);
}

```

3. Membuat iterasi lanjutan "else if" yang mengembalikan hasil ke `penggunas.check` dengan logika lolos validasi jika username dan password ada dan sesuai dengan database. Diambil dari method boolean check pada `DaftarPengguna.java`. dapat dilihat pada *script* berikut.

```

} else if (penggunas.check(username, password) == false) {
    session.setAttribute("logged_in", false);
    request.setAttribute("error", "Username atau Password salah.");
    RequestDispatcher rdp =
    request.getRequestDispatcher("index.jsp");
    rdp.forward(request, response);
}

```

4. Membuat iterasi else untuk melanjutkan validasi apakah username dan password ada dan sesuai dengan role masing-masing modul, serta menetapkan/set session bernama "logged\_in", seperti pada script berikut.

```

else {
    TPenggunapengguna = penggunas.getPenggunaLogin(username,
password);
    session.setAttribute("logged_in", true);
    if (pengguna.getRole().equals("1")) {
        //admin
        session.setAttribute("pengguna", pengguna);
        session.setAttribute("username", username);
        session.setAttribute("role", pengguna.getRole());
        RequestDispatcher rdp =
request.getRequestDispatcher("home_admin");
        rdp.forward(request, response);
        session.removeAttribute("error");
    } else if (pengguna.getRole().equals("2")) {
        //aset tetap
        session.setAttribute("pengguna", pengguna);
        session.setAttribute("username", username);
        session.setAttribute("role", pengguna.getRole());
        session.setAttribute("logged_in", true);
        RequestDispatcher rdp =
request.getRequestDispatcher("homeaset?pageref=list");
        rdp.forward(request, response);
        session.removeAttribute("error");
    }
}
// dan selanjutnya.

```

5. Untuk penetapan set role pada file LoginServlet dibagi menjadi 9 tipe role yaitu : 1 (administrator), 2 (aset tetap), 3 (bendahara) , 4 (pembayaran), 5 (aprover pembayaran), 6 (anggaran), 7( persediaan), 8 (pelaporan), dan 9 (kpa). Analoginya jika pengguna lolos validasi diatas, maka pengguna akan masuk ke halaman utama masing-masing modul sesuai *role*-nya dengan menggunakan request dispatcher page. Dapat dilihat pada *script* berikut.

```

if (pengguna.getRole().equals("1")) {

```

```

        session.setAttribute("pengguna", pengguna);
        session.setAttribute("username", username);
        session.setAttribute("role", pengguna.getRole());
        RequestDispatcher rdp =
request.getRequestDispatcher("home_admin");
        rdp.forward(request, response);
        session.removeAttribute("error");
    }

```

#### 5.6.4. Fungsi Validasi dan DatePicker dengan JavaScript

Berapa fungsi validasi yang digunakan dengan bantuan Javascript antara lain.

1. Fungsi agar field input tidak boleh kosong. Lihat *script* berikut:

```

function validateForm() {
    var a=document.forms["login"]["username"].value;
    var b=document.forms["login"]["password"].value;
    if (a=="" || b=="") {
        alert("Username dan Password Tidak Boleh Kosong.");
        return false;
    }
}

```

2. Fungsi agar field input tidak boleh diisi oleh karakter diluar aA – zZ atau 0 – 9. Lihat pada *script* berikut.

```

function isCharAneh(evt) {
    var charCode = (evt.which) ? evt.which : event.keyCode
    if (charCode > 31 && (charCode < 48 || charCode > 57)&&
charCode < 65 && (charCode < 91 || charCode > 96)) {
        return false;
    }
    return true;
}

```

3. Fungsi larangan menggunakan klik kanan pada field inputan.

```

function whichButton()
{

```

```

    if (event.button==2)//For right click
    {
        alert("Dilarang Klik Kanan (Larangan Fungsi Paste) !");
    }
}

```

4. Fungsi larangan anti kopi paste/larangan penggunaan tombol ctrl. Lihat pada script berikut.

```

function noCTRL(evt)
{
    var code = (document.all) ? event.keyCode:evt.which;

    var msg = "Dilarang Menekan Tombol 'Ctrl' (Larangan Fungsi
Paste) !";
    if (parseInt(code)==17) // This is the Key code for CTRL key
    {
        alert(msg);
        window.event.returnValue = false;
    }
}

```

5. Fungsi DatePicker, yaitu memunculkan kalender secara otomatis pada saat field input di klik dengan menambahkan file beberapa css. Lihat pada *script* berikut.

```

<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.min.js"></
script>
<link rel="stylesheet"
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.7.3/themes/south-
street/ui.all.css" type="text/css">
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.6/jquery-
ui.min.js"></script>
<script type="text/javascript" src="./scripts/jquery.validate.js"></script>
<script type="text/javascript"
src="./scripts/jquery.ui.datepicker.validation.js"></script>
<link rel="stylesheet" href="css/scroll-tab.css" />

```



### 5.6.5. Proses Tayang Data

Dengan menggunakan metode Model View Controller/MVC, untuk implementasi proses tayang data pada modul Administrasi antara lain:

1. Menggunakan text pada file \*.jsp saja tanpa menggunakan fungsi MVC, seperti pada daftar\_konten.jsp, home\_admin.jsp, pengaturan.jsp, referensi.jsp, bantuan.jsp, dan tentang.jsp. beberapa file \*.jsp tersebut umumnya hanya digunakan sebagai halaman depan/home dari masing-masing menu yang diakses.
2. Menggunakan metode MVC, yaitu dengan menghubungkan class entity, model, dan servlet menjadi sebuah fungsi tayang. Sebagai contoh untuk tayang data referensi bank dapat dilihat pada gambar 5.27 class entity TBank

```
public class TBank implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "kdbank")
    private String kdbank;
    @Basic(optional = false)
    @Column(name = "nmbukubank")
    private String nmbukubank;
    @Column(name = "thang")
    private String thang;
    @Basic(optional = false)
    @Column(name = "kdbenda")
    private String kdbenda;
    @Basic(optional = false)
    @Column(name = "kdsatker")
    private String kdsatker;

    public TBank() {
    }

    public TBank(String kdbank) {
        this.kdbank = kdbank;
    }
}
```

Gambar 5. 27-class entity TBank

Untuk method tayang data referensi bank dapat dilihat pada gambar 5.28 class DaftarBank.

```

public class DaftarBank {
    public DaftarBank() {
        emf = Persistence.createEntityManagerFactory("SlapkaanPU");
    }
    private EntityManagerFactory emf = null;

    public EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    public List<TBank> getBank() {
        List<TBank> bank = new ArrayList<TBank>();

        EntityManager em = getEntityManager();
        try {
            Query q = em.createQuery("SELECT a FROM TBank a ORDER BY a.kdBank ASC");
            bank = q.getResultList();
        } finally {
            em.close();
        }
        return bank;
    }
}

```

**Gambar 5. 28-Potongan method getBank pada model DaftarBank**

Untuk method tayang data referensi bank dapat dilihat pada gambar 5.29 class BankServlet.

```

} else if (pageref.equals("listbank")) {
    session.removeAttribute("session");
    session.removeAttribute("tempInput");
    BankController bankController = new BankController(request);
    request.getRequestDispatcher(bankController.listBank()).forward(request, response);
}

```

**Gambar 5. 29-Potongan iterasi session pageref untuk url listbank pada BankServlet**

Untuk method listBank referensi bank dapat dilihat pada gambar 5.30 class BankController.

```

public BankController(HttpServletRequest request) {
    this.request = request;
}
int offset;
int length;
DaftarBank daftarBank = new DaftarBank();
List<TBank> listBank = daftarBank.getBank();

public String listBank() {
    int maxEntriesPerPage = 20;
    int page = 1;

    String pageNumberValue = request.getParameter("pageNumber");
    if (pageNumberValue != null) {
        try {
            page = Integer.parseInt(pageNumberValue);
            System.out.println("Page Number:" + page);
        } catch (NumberFormatException e) {
            e.printStackTrace();
        }
    }
    int offset = maxEntriesPerPage * (page - 1);
    TestList(offset, maxEntriesPerPage);
    request.setAttribute("pages", getPages());
    request.setAttribute("listbank", getListByOffsetAndLength());
    return "/pages/administrator/Referensi Umum/daftar_bank.jsp";
}

```

**Gambar 5. 30-Potongan method listBank pada potongan kelas BankController untuk menayangkan data Bank**

### 3. Hasil implementasi Tayang (Referensi Bank)

Berikut adalah hasil implementasi tayang data, khususnya untuk tayang Referensi Bank.(Lihat gambar 5.31)



Gambar 5. 31-Implementasi tayang data (referensi bank)

#### 5.6.6. Proses Rekam Data

Dengan menggunakan metode Model View Controller/MVC, untuk implementasi proses rekam data pada modul Administrasi dapat dilihat dari beberapa potongan script berikut.

Gambar 5.32, method rekamBank pada class DaftarBank.

```
public void rekamBank(TBank bank) {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        em.persist(bank);
        em.getTransaction().commit();
    } finally {
        if (em != null) {
            em.close();
        }
    }
}
```

Gambar 5. 32-Potongan Method rekamBank pada class DaftarBank

Gambar 5.33, method prosesrekamBank pada class BankController.

```
public String prosesrekamBank() {
    TBank bank = new TBank();
    HttpSession session = request.getSession();
    String kodebank = request.getParameter("kdbank");
    String nama = request.getParameter("nmbukubank");
    String tahun = request.getParameter("thang");
    String bendahara = request.getParameter("kdbenda");
    String satker = request.getParameter("kdsatker");

    if (kodebank != null) {
        bank.setKdbank(kodebank);
        bank.setNmbukubank(nama);
        bank.setThang(tahun);
        bank.setKdbenda(bendahara);
        bank.setKdsatker(satker);
        session.setAttribute("tempinput", bank);
    } else {
        session.setAttribute("error", "Data Bank Gagal Disimpan ");
        return "/pages/administrator/Referensi Umum/rekam_bank.jsp";
    }
}
```

**Gambar 5. 33-Potongan method prosesrekamBank pada kelas BankController**

Gambar 5.34, iterator pada class BankServlet.

```
else if (pageref.equals("listbank")) {
    session.removeAttribute("error");
    session.removeAttribute("tempinput");
    BankController bankController = new BankController(request);
    request.getRequestDispatcher(bankController.listBank()).forward(request, response);
}
```

**Gambar 5. 34-Iterator else if membawa session pageref url listbank pada kelas BankServlet**

Gambar 5.35, implementasi tayang Rekam data Bank.

Gambar 5. 35-Implementasi Rekam data Bank

### 5.6.7. Proses Edit Data

Dengan menggunakan metode Model View Controller/MVC, untuk implementasi proses edit data pada modul Administrasi dapat dilihat dari beberapa potongan script berikut.

Gambar 5.36, potongan method editBank.

```
public void editBank(TBank bank) {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        bank = em.merge(bank);
        em.getTransaction().commit();
    } finally {
        if (em != null) {
            em.close();
        }
    }
}
```

Gambar 5. 36-Potongan metbhod editBank

Gambar 5.37, potongan method ubahBank & prosesUbahBank pada *class* BankController.

```

public String ubahBank() {

    HttpSession session = request.getSession();
    DaftarBank daftarBank = new DaftarBank();
    List<TBank> listBank = daftarBank.getBank();
    request.setAttribute("listBank", listBank);

    String editBank = request.getParameter("idBank");
    TBank bank = daftarBank.findBank(editBank);
    session.setAttribute("editBank", bank);
    return "pages/administrasi/Referensi/Ubah/edit_bank.jsp"
}

public String prosesUbahBank() {
    TBank bank = new TBank();
    DaftarBank daftarBank = new DaftarBank();
    HttpSession session = request.getSession();
    String nama = request.getParameter("nama");
    String tahun = request.getParameter("tahun");
    String bendahara = request.getParameter("bendahara");
    String satker = request.getParameter("satker");
    bank = (TBank) session.getAttribute("editBank");

    if (nama != null) {
        bank.setNamaBank(nama);
        bank.setThang(tahun);
        bank.setKdbenda(bendahara);
        bank.setKdsatker(satker);
    }
}

```

**Gambar 5. 37-Potongan method ubahBank & prosesUbahBank pada kelas BankController**

Gambar 5.38, potongan iterasi pada *class* BankServlet.

```

else if (pageref.equals("ubahbank")) {
    session.removeAttribute("error");
    session.removeAttribute("sukses");
    session.removeAttribute("tempinput");
    request.setAttribute("url", "pages/administrasi/Referensi/Ubah/edit_bank.jsp");
    BankController bankController = new BankController(request);
    request.getRequestDispatcher(bankController.ubahBank()).forward(request, response);
} else if (pageref.equals("proses-ubahbank")) {
    session.removeAttribute("error");
    session.removeAttribute("sukses");
    session.removeAttribute("tempinput");
    BankController bankController = new BankController(request);
    request.getRequestDispatcher(bankController.prosesUbahBank()).forward(request, response);
}

```

**Gambar 5. 38-Potongan iterasi else if mengambil sesion pageref url ubahbank pada kelas BankServlet**

### 5.6.8. Proses Pencarian Data

Dengan menggunakan metode Model View Controller/MVC, untuk implementasi proses Pencarian data pada modul Administrasi dapat dilihat dari beberapa potongan script berikut.

Gambar 5.39, Potongan method findBankbyNama pada kelas DaftarBank.

```

public List<TBank> findBankByNama(String nmBank) {
    EntityManager em = getEntityManager();
    List<TBank> bank = null;
    try {
        Query q = em.createQuery("SELECT a FROM TBank AS a where a.mmbukubank like
        q.setParameter("nmBank", "%" + nmBank + "%");
        bank = q.getResultList();
    } finally {
        em.close();
    }
    return bank;
}
}

```

Gambar 5. 39-Potongan method findBankbyNama pada kelas DaftarBank

Gambar 5.40 potongan method cariBank pada kelas BankController.

```

public String cariBank() {
    String nama = request.getParameter("nmbank");
    DaftarBank daftarbank = new DaftarBank();
    List<TBank> listBank = daftarbank.findBankByNama(nama);
    request.setAttribute("listbank", listBank);
    request.setAttribute("pages", getPages());
    // request.setAttribute("listbank", getListByOffsetAndLength());
    return "/pages/administrator/ReferensiUmum/daftar_bank.jsp";
}
}

```

Gambar 5. 40 method cariBank pada kelas BankController

Gambar 5.41 Potongan iterasi else if dengan session pageref url caribank pada kelas BankServlet.

```

} else if (pageref.equals("caribank")) {
    //session.removeAttribute("ctrl");
    session.removeAttribute("tempinput");
    BankController bankController = new BankController(request);
    request.getRequestDispatcher(bankController.cariBank()).forward(request, response);
}
}

```

Gambar 5. 41-Potongan iterasi else if dengan session pageref url caribank pada kelas BankServlet

### 5.6.9. Proses Backup Data

Dengan menggunakan metode Model View Controller/MVC, untuk implementasi proses Backup data pada modul Administrasi dapat dilihat dari beberapa potongan script berikut.

Gambar 5.42 Potongan method tesBackup pada kelas DaftarBackup.



```

public DaftarBackup() {
    emf = Persistence.createEntityManagerFactory("Siapbac20");
}
private EntityManagerFactory emf = null;

public EntityManager getEntityManager() {
    return emf.createEntityManager();
}

public void tesBackup(String TUser, String TBank, String TCroleh, String Tbakket) {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();

        Query q = em.createQuery("COPY db.backup to C.");
        q.setParameter("db.backup", "db.backup");
        q.setParameter("password", "password");
        q.setParameter("file", "file");
        q.setParameter("admin", "admin");
        q.executeUpdate();

        em.getTransaction().commit();
    } finally {

```

Gambar 5. 42 Potongan method tesBackup pada kelas DaftarBackup

Gambar 5.43 Potongan method listBackup pada kelas BackupController

```

public String backupData() {
    request.setAttribute("ruang", thang);
    return "/pages/administrator/Pengaturan/backup_data.jsp";
}
public String listBackup() {
    int maxEntriesPerPage = 5;
    int page = 1;

    String pageNumberValue = request.getParameter("pageNumber");
    if (pageNumberValue != null) {
        try {
            page = Integer.parseInt(pageNumberValue);
            System.out.println("Page Number:" + page);
        } catch (NumberFormatException e) {
            e.printStackTrace();
        }
    }

    int offset = maxEntriesPerPage * (page - 1);
    request.setAttribute("listbackup", getListByOffsetAndLength());
    return "/pages/administrator/Pengaturan/backup_data.jsp";
}

```

Gambar 5. 43 Potongan method listBackup pada kelas BackupController

Gambar 5.44 Potongan iterasi else if session pageref url listbackup pada kelas BackupServlet

```

} else if (pageref.equals("listbackup")) {
    session.removeAttribute("error");
    session.removeAttribute("tempinput");
    BackupController backupController = new BackupController(request);
    request.getRequestDispatcher(backupController.listBackup()).forward(request, response);
}

```

**Gambar 5. 44 Potonganiterasi else if session pageref url listbackup pada kelas BackupServlet**

### 5.6.10. Proses Logout

Proses logout memiliki arti yaitu menghilangkan/*removing* seluruh nilai session yang masih dibawa atau terdaftar (username, password, role, tempinput, pageref, dll) saat aplikasi dijalankan hingga pengguna diarahkan kembali ke halaman index.jsp. Pada modul Administrasi ini, proses logout dijelaskan di dalam file LogoutServlet, seperti pada gambar 5.45 berikut.

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession session = request.getSession(true);
    session.removeAttribute("logged_in");
    session.invalidate();
    session = null;
    RequestDispatcher rdp = request.getRequestDispatcher("index.jsp");
    rdp.forward(request, response);
}

```

**Gambar 5. 45-Potonganproses remove session pada LogoutServlet**

Sebagai asumsi, jika pengguna mengklik tombol ber-url "logout" maka seluruh session `logged_in` akan dihapus, kemudian mengosongkan semua session, dengan menginisiasi session = null.

## 5.7. Integrasi Modul

Integrasi antar modul pada aplikasi SIAPKAN sudah dilakukan sejak awal pengerjaan tugas akhir. Dengan bantuan subversion (svn) google code, dan menu subversion pada editor Netbeans 7.1.1, seluruh modul-modul yang ada dapat dijadikan satu kemasan *Project* SIAPKAN. Seperti telah disebutkan pada bab sebelumnya setiap modul dibuat dalam package tersendiri.

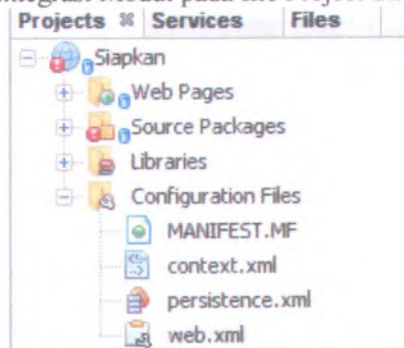
Modul Administrasi yang berfungsi sebagai pusat data referensi dari modul lainnya, dengan menggunakan *subversion*, mengizinkan modul lain mudah mengakses atau menggunakan *primarykey*, atribut atau tabel dari modul Administrasi. Untuk

contoh subversion google code dan project SIAPKAN dapat dilihat pada gambar 5.46 dan 5.47 berikut.



Gambar 5. 46-Subversion google code untuk integrasi modul

Gambar 5.47 Integrasi Modul pada file Project SIAPKAN



Gambar 5. 47-Integrasi Modul pada Project SIAPKAN

Proses integrasi antar modul melalui modul Administrasi secara garis besar antara lain:

1. Pemakaian atribut bersama.  
Atribut-atribut seperti *username*, *password*, dan *role* digunakan untuk akses ke setiap modul melalui proses login. Dengan ketiga atribut tersebut, *username* dipilah pada saat proses login agar masing-masing pengguna dapat masuk ke halaman utama modulnya. Masing-masing pengguna modul harus ditentukan dahulu *username*, *password*, dan *role*-nya
2. Penyediaan data referensi.  
Modul Administrasi menjadi sumber data referensi bagi modul lainnya, dimana Primary Key pada tabel-tabel referensi digunakan sebagai dasar perekaman bagi modul lainnya. Sebagai contoh **kdsatker** pada tabel **TSatker** digunakan oleh setiap modul, karena data **TSatker** dibutuhkan pada banyak transaksi perekaman data pada setiap modul.
3. Penyeragaman nama proses.  
Pada tahap ini dilakukan penyeragaman penamaan proses. Sebagai contoh, untuk penayangan data menggunakan format kata depan "list\_\*", untuk proses rekam adalah "rekam\_\*", proses ubah data adalah "ubah\_\*", dan proses hapus adalah "hapus\_\*" (\**pengganti kata yang mengikuti kata depan penamaan fungsi/proses*).
4. Penyeragaman nama file.  
Pada tahap ini dilakukan penyeragaman nama file, baik file bertipe Model, penamaan diawali oleh "Daftar" kemudian dilanjutkan oleh nama *class entity*-nya. Pada file bertipe Controller dan Servlet, kata depan adalah nama *class entity*-nya sedangkan kata belakang ditambah Controller atau Servlet, misal *SatkerController* atau *SatkerServlet*.
5. Integrasi pada database.  
Pada tahap ini seluruh tabel yang digunakan oleh masing-masing modul dijadikan menjadi satu kesatuan database bernama "siapkan". Dengan user "postgres" dan password "bersiap".

### **5.8. Migrasi data**

Migrasi data digunakan untuk memindahkan data pada database yang digunakan sistem saat ini yang menggunakan MySQL dan FoxPro ke dalam database Aplikasi SIAPKAN yang menggunakan DBMS PostgreSQL. Proses migrasi data dilakukan dengan manual, dan dengan menggunakan software Full Convert 5.13 Enterprise.

Beberapa tahapan migrasi data saat implementasi tugas akhir ini antara lain:

1. Menganalisis tabel-tabel dari database sumber dari 5 aplikasi yang berbeda yaitu aplikasi SPM, SISKAK, SIMAK BMN, DIPAK/RKAKL, dan SAKPA.
2. Tabel-tabel tersebut memiliki dua tipe file, yaitu .sql dari MySQL dan .dbf dari DBMS FoxPro
3. Konversi database bertipe .dbf (Foxpro) ke dalam .sql
4. Konversi database bertipe .sql (MySQL) ke dalam tipe .sql (Postgre),
5. Setelah mendapatkan tabel sumber baru sebanyak 247 tabel, kemudian di sortir sesuai kebutuhan modul Administrator menjadi 70 tabel/ yang kemudian menjadi *Entity Class* baru di package modul Administrasi.

Hasil Migrasi data dapat dilihat pada gambar 5.48.

### **5.9. Uji Coba dan Evaluasi**

Subbab ini berisi bagian uji coba dan evaluasi hasil dari implementasi aplikasi SIAPKAN modul Administrasi. Di dalamnya dibahas mengenai dua tipe uji coba, yaitu uji coba interaksi dan uji coba fungsional.

<input type="checkbox"/>	t_bank	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_banking	saft	4	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_belanja	saft	8	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_bendahara	saft	26	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_biaya	saft	20	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_bidang	saft	10	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_bidmkn	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_eksp	saft	250	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_ekstir	saft	8	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_dukubani	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_emfip_bend	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_orbayer	saft	6	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_orleh	saft	52	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_bank	saft	5	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_banor	saft	6	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_dact	saft	39	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_dil	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_dir	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_emity	saft	80	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_es2	saft	2794	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_fungil	saft	11	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_gbrp	saft	30	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_gbrt	saft	1024	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_gbrt	saft	0	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex
<input type="checkbox"/>	t_gbrmn	saft	8	Browse	Select	Insert	Empty	Alter	Drop	Vacuum	Analyze	Reindex

Gambar 5. 48-Hasil Migrasi data

### 5.9.1. Uji Coba Interaksi

Uji coba interaksi dilakukan untuk mengetahui apakah interaksi dari pengguna ke sistem aplikasi SIAPKAN pada modul Administrasi dapat berjalan baik. Hasil uji coba interaksi dapat dilihat pada tabel 5.3 berikut ini.

Tabel 5. 3 Uji Coba Interaksi

No.	Interaksi	Hasil
1.	Tayang Form Login/halaman indeks	Tampilan halaman indeks ada
2.	Isi field username	Inputusername berhasil
3.	Isi field password	Input password berhasil
4.	Klik tombol Login	Berhasil memproses request (error/sukses login)
5.	Tayang halaman Home Admin	Tampilan home admin ada

6.	Klik Referensi Umum	Berhasil menampilkan halaman Referensi Umum
7.	Klik Referensi Persediaan	Berhasil menampilkan halaman Referensi Persediaan
8.	Klik Referensi Anggaran	Berhasil menampilkan halaman Referensi Anggaran
9.	Klik Referensi Pembayaran	Berhasil menampilkan halaman Referensi Pembayaran
10.	Klik Referensi Bendahara	Berhasil menampilkan halaman Referensi Bendahara
11.	Klik Referensi Aset Tetap	Berhasil menampilkan halaman Referensi Aset Tetap
12.	Klik Profil Pengguna	Berhasil menampilkan halaman Profil Pengguna
13.	Klik Waktu Kerja	Berhasil menampilkan halaman Waktu Kerja
14.	Klik Bantuan	Berhasil menampilkan halaman Bantuan
15.	Klik Tentang	Berhasil menampilkan halaman Tentang SIAPKAN
16.	Input Cari Data	Hasil pencarian berhasil ditampilkan
17.	Klik Rekam Data	Berhasil menampilkan Halaman Rekam Data
18.	Klik Ubah Data	Berhasil menampilkan Halaman Ubah Data
19.	Klik Hapus Data	Berhasil menghapus Data
20.	Klik Backup Data	Suspended.
20.	Klik Logout	Berhasil keluar dari aplikasi

### 5.9.2. Uji Coba Fungsional

Tahap uji coba fungsional ini memastikan apakah kebutuhan dan proses dari setiap Use Case telah terpenuhi. Dengan menggunakan unit test dari rancangan yang telah dirancang pada lampiran D. setiap skenario pada test case dijalankan dan hasil

yang ada pada test case dibandingkan dengan hasil aplikasi. Tabel 5.4 berikut adalah kesimpulan dari hasil unit test.

**Tabel 5. 4 Hasil Unit Test**

<b>Sub Menu</b>	<b>Kode UC</b>	<b>Hasil</b>
Login	UC-1	Berhasil
Tayang Bendahara	UC-2	Berhasil
Rekam Bendahara	UC-3	Berhasil
Ubah Bendahara	UC-4	Berhasil
Hapus Bendahara	UC-5	Berhasil
Tayang Beban	UC-6	Berhasil
Rekam Beban	UC-7	Berhasil
Ubah Beban	UC-8	Berhasil
Hapus Beban	UC-9	Berhasil
Tayang Fungsi	UC-10	Berhasil
Rekam Fungsi	UC-11	Berhasil
Ubah Fungsi	UC-12	Berhasil
Hapus Fungsi	UC-13	Berhasil
Tayang dokumen	UC-14	Berhasil
Rekam Dokumen	UC-15	Berhasil
Ubah Dokumen	UC-16	Berhasil
Hapus Dokumen	UC-17	Berhasil
Tayang MAK	UC-18	Berhasil
Rekam MAK	UC-19	Berhasil
Ubah MAK	UC-20	Berhasil
Hapus MAK	UC-21	Berhasil
Tayang Jenis SPM	UC-22	Berhasil
Rekam Jenis SPM	UC-23	Berhasil
Ubah Jenis SPM	UC-24	Berhasil
Hapus Jenis SPM	UC-25	Berhasil
Tayang Cara Bayar	UC-26	Berhasil
Rekam Cara Bayar	UC-27	Berhasil
Ubah Cara Bayar	UC-28	Berhasil
Hapus Cara Bayar	UC-29	Berhasil
Tayang Jenis Dokumen	UC-30	Berhasil
Rekam Jenis Dokumen	UC-31	Berhasil



<i>Sub Menu</i>	<i>Kode UC</i>	<i>Hasil</i>
Ubah Jenis Dokumen	UC-32	Berhasil
Hapus Jenis Dokumen	UC-33	Berhasil
Tayang Bendahara	UC-34	Berhasil
Rekam Bendahara	UC-35	Berhasil
Ubah Bendahara	UC-36	Berhasil
Hapus Bendahara	UC-37	Berhasil
Tayang Jenis Beban	UC-38	Berhasil
Rekam Jenis Beban	UC-39	Berhasil
Ubah Jenis Beban	UC-40	Berhasil
Hapus Jenis Beban	UC-41	Berhasil
Tayang Jenis SPM	UC-42	Berhasil
Rekam Jenis SPM	UC-43	Berhasil
Ubah Jenis SPM	UC-44	Berhasil
Hapus Jenis SPM	UC-45	Berhasil
Tayang Aset Tetap	UC-46	Berhasil
Rekam Aset Tetap	UC-47	Berhasil
Ubah Aset Tetap	UC-48	Berhasil
Hapus Aset Tetap	UC-49	Berhasil
Tayang Cara Perolehan	UC-50	Berhasil
Rekam Cara Perolehan	UC-51	Berhasil
Ubah Cara Perolehan	UC-52	Berhasil
Hapus Cara Perolehan	UC-53	Berhasil
Tayang SSKel BMN	UC-54	Berhasil
Rekam SSKel BMN	UC-55	Berhasil
Ubah SSKel BMN	UC-56	Berhasil
Hapus SSKel BMN	UC-57	Berhasil
Tayang Map Barang	UC-58	Berhasil
Rekam Map Barang	UC-59	Berhasil
Ubah Map Barang	UC-60	Berhasil
Hapus Map Barang	UC-61	Berhasil
Tayang Kanwil	UC-62	Berhasil
Rekam Kanwil	UC-63	Berhasil
Ubah Kanwil	UC-64	Berhasil
Hapus Kanwil	UC-65	Berhasil
Tayang Akun	UC-66	Berhasil

<b>Sub Menu</b>	<b>Kode UC</b>	<b>Hasil</b>
Rekam Akun	UC-67	Berhasil
Ubah Akun	UC-68	Berhasil
Hapus Akun	UC-69	Berhasil
Tayang Blokir	UC-70	Berhasil
Rekam Blokir	UC-71	Berhasil
Ubah Blokir	UC-72	Berhasil
Hapus Blokir	UC-73	Berhasil
Tayang Departemen	UC-74	Berhasil
Rekam Departemen	UC-75	Berhasil
Ubah Departemen	UC-76	Berhasil
Hapus Departemen	UC-77	Berhasil
Tayang GBKPK	UC-78	Berhasil
Rekam GBKPK	UC-79	Berhasil
Ubah GBKPK	UC-80	Berhasil
Hapus GBKPK	UC-81	Berhasil
Tayang Kegiatan	UC-82	Berhasil
Rekam Kegiatan	UC-83	Berhasil
Ubah Kegiatan	UC-84	Berhasil
Hapus Kegiatan	UC-85	Berhasil
Tayang Unit	UC-86	Berhasil
Rekam Unit	UC-87	Berhasil
Ubah Unit	UC-88	Berhasil
Hapus Unit	UC-89	Berhasil
Tayang IB	UC-90	Berhasil
Rekam IB	UC-91	Berhasil
Ubah IB	UC-92	Berhasil
Hapus IB	UC-93	Berhasil
Tayang Output	UC-94	Berhasil
Rekam Output	UC-95	Berhasil
Ubah Output	UC-96	Berhasil
Hapus Output	UC-97	Berhasil
<i>Tayang Profil Pengguna</i>	UC-98	<i>Berhasil</i>
Rekam Profil Pengguna	UC-99	Berhasil
Ubah Profil Pengguna	UC-100	Berhasil

<b>Sub Menu</b>	<b>Kode UC</b>	<b>Hasil</b>
Hapus Profil Pengguna	UC-101	Berhasil
Tayang Waktu Kerja	UC-102	Berhasil
Rekam Waktu Kerja	UC-103	Berhasil
Hapus Waktu Kerja	UC-104	Berhasil
Backup database	UC-105	Berhasil
Tayang Konten	UC-106	Berhasil
Tayang Dokumen Peraturan	UC-107	Berhasil
Rekam Dokumen Peraturan	UC-108	Berhasil
Ubah Dokumen Peraturan	UC-109	Berhasil
Hapus Dokumen Peraturan	UC-110	Berhasil
Tayang Tentang Siapkan	UC-111	Berhasil
Backup Data	UC-112	Suspended
Logout Aplikasi	UC-113	Berhasil

## BAB VI KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan dan saran

### 6.1. Kesimpulan

Setelah melakukan proses pengerjaan tugas akhir ini, penulis dapat menyimpulkan beberapa poin yang terdapat pada saat pengerjaan yaitu:

1. Integrasi aplikasi yang digunakan dalam aplikasi SIAPKAN bertujuan untuk mengurangi duplikasi proses dan data yang pada saat ini terjadi, ataupun dampak efisiensi dari segi waktu. Pada Aplikasi SIAPKAN Modul Administrasi, integrasi membuat pengguna masing-masing modul dapat dengan mudah mendapatkan data referensi yang sama dengan modul lainnya, hingga tidak perlu khawatir terjadinya perbedaan kode referensi atau terjadinya duplikasi data.
2. Pemrograman bermetode Model, View, dan Controller, sangat membantu penulis dalam mempermudah kodifikasi/*programming*, khususnya dalam mengembangkan sebuah aplikasi berskala besar. Karena sifatnya yang tidak menumpuk fungsi kedalam sebuah file, jika terdapat sebuah kesalahan dalam method dapat dengan mudah dilacak dan diperbaiki.
3. Menu pengelolaan referensi tersedia di modul Administrasi karena fungsinya sebagai penyedia referensi data dari modul-modul lainnya. Fungsi lainnyatersedia pada menu pengaturan untuk profil pengguna, *role* pengguna dan waktu kerja sistem, dan menu Backup untuk *mem-backup* database keseluruhan SIAPKAN.
4. Metode *session* sangat membantu penulis khususnya dalam pelacakan data dan obyek modul Administrasi pada saat digunakan bersama oleh modul lainnya.

5. Dengan menggunakan fitur subversion baik dari code.google.com, maupun dari editor Netbeans, dapat mempermudah tim SIAPKAN dalam pengintegrasian pekerjaan masing-masing anggota, karena sifatnya yang selalu dapat cepat di-update dan di-commit setiap saat ada perubahan. Jika pun terjadi kejadian insidental yang mengharuskan programming dari awal, sudah tersedia fitur *checkout* untuk mengembalikan struktur dan isi program seperti sedia kala.

## 6.2. Saran

Pengembangan aplikasi SIAPKAN modul Administrasi ini ternyata tidak luput dari kekurangan dan kesalahan di dalam pengerjaannya. Beberapa saran yang dapat penulis berikan untuk pengembangan aplikasi SIAPKAN dengan tipe Java Web, antara lain:

1. Aplikasi SIAPKAN dapat dilakukan uji coba/trial secara langsung pada proses bisnis pengelolaan transaksi keuangan negara satuan kerja dalam ruang lingkup Direktorat Jenderal Perbendaharaan Negara.
2. Perlunya pengembangan lebih lanjut pada fitur modul Administrasi aplikasi SIAPKAN, khususnya untuk penambahan data referensi, menu Backup dan Restore data.
3. Untuk memenuhi kebutuhan manajemen rekanan/pihak ketiga, maka diperlukan sebuah modul tambahan bernama Modul Komitmen.
4. Pengembangan aplikasi berskala besar seperti SIAPKAN, dapat menggunakan *framework*, karena proses pemrograman lebih terstruktur dan tertata dibandingkan dengan non-*framework*.
5. Diperlukan sebuah *templateview* JSP berupa drop down list, yang dapat menampung semua *header* menu modul aplikasi SIAPKAN. Karena dapat lebih menghemat proses, kode, dan waktu.

## DAFTAR PUSTAKA

- [1] ---- . 1998. *ZDU Student Manual : Java Programming Part I*. ZD University
- [2] Andie Megantara dkk (2008) - *Manajemen Perbendaharaan Pemerintahan, Aplikasi di Indonesia*, halaman 31.
- [3] Direktur Jenderal Perbendaharaan. 2009. *Peraturan Direktur Jenderal Perbendaharaan Nomor PER-47/PB/2009 tentang Petunjuk Pelaksanaan Penatausahaan dan Penyusunan Laporan Pertanggungjawaban Bendahara Kementerian Negara/Lembaga/ Kantor/Satuan Kerja*
- [4] Hall, Marty. 2000. *Core Servlets and JavaServer Pages*. Upper Saddle River : Prentice-Hall, Inc, 2000.
- [5] Hermawan, Benny. 2004. *Menguasai Java Programming 2 dan Object Oriented Programming*. Yogyakarta : Penerbit Andi
- [6] Kementerian Keuangan. 2007. *Peraturan Menteri Keuangan Nomor 120/PMK.06/2007 tentang Penatausahaan Barang Milik Negara*.
- [7] Seshadri, Govind. 1999. *Server-side Java: Understanding JavaServer Pages Model 2 architecture*. Javaworld. [Online] 1999.
- [8] Suharto, B. Herry & Soesilo Wijono. 2006. *Membangun Aplikasi Menggunakan Qt Designer dengan Database PostgreSQL/MySQL*. Yogyakarta : Penerbit Andi
- [9] Van Haecke, Bernard. 2002. *JDBC3.0 Java Database Connectivity*. New York : Hungry Minds, Inc, 2002.
- [10] Wijono, Sri Hartati, B. Herry Suharto, & Matius Soesilo Wijono. 2007. *Pemrograman Java Servlet dan JSP dengan Netbeans*. Yogyakarta : Penerbit Andi
- [11] Zeis, C., Ruel, C., & Wessler, M. (2009). *Oracle 11g For Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc.

**LAMPIRAN A**  
**DESKRIPSI USE CASE**

### A.1. Deskripsi Use Case Proses Login

Tabel A. 1 Deskripsi Use Case Proses Login

UC-1– Proses Login	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman index/form login	
<b>Triggers:</b> Admin menginput field username dan password, Admin mengklik tombol “Enter”	
<b>Basic course:</b> Sistem menampilkan Halaman indeks/form login. Admin mengisi field username dan password lalu menekan tombol Enter.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan masukan ke dalam Daftar Pengguna, kemudian Admin masuk ke halaman Utama modul Administrasi	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Username dan Password tidak boleh kosong".</li> <li>• Jika input Username dan Password adalah salah, maka system menampilkan pesan "Username atau Password salah."</li> <li>• Field input username tidak dapat diisi dengan karakter diluar abjad aA – zZ atau numerik 0 - 9</li> <li>• Jika role telah sesuai dengan role pada modul Administrasi, yaitu "1", maka Admin berhasil masuk ke halaman modul Administrasi.</li> <li>• Jika sebelum login, pengguna mengetik url halaman modul pada browser, akan dikembalikan ke halaman unauthorized.</li> </ul>	



## A.2. Deskripsi Use Case Tayang Bendahara

Tabel A. 2 Deskripsi Use Case Tayang Bendahara

UC-2– Proses Tayang Bendahara	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Bendahara	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Bendahara.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Bendahara	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Bendahara, sesuai dengan tabel t_bendahara	

## A.3. Deskripsi Use Case Rekam Bendahara

Tabel A. 3 deskripsi Use Case Rekam Bendahara

UC-3–Proses Rekam Bendahara	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Bendahara	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Bendahara. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan	

masuk ke dalam Daftar Bendahara, dan menampilkan pesan sukses pada Halaman Rekam Bendahara.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi."
- Jika NIP tidak berupa angka atau NIP tidak tepat 18 digit, sistem menampilkan pesan "NIP harus berupa 18 digit angka".
- Jika kode bendahara sudah pernah direkam, sistem menampilkan pesan "Kode bendahara sudah ada".

#### A.4. Deskripsi Use Case Ubah Bendahara

Tabel A. 4 Deskripsi Use Case Ubah Bendahara

<b>UC-4 –Proses Ubah Bendahara</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Bendahara	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data bendahara yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Bendahara dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Bendahara dan menampilkan pesan sukses pada Halaman Daftar Bendahara	
<b>Alternate courses:</b>	
<ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika NIP tidak berupa angka atau NIP tidak tepat 18 digit, sistem menampilkan pesan "gagal merekam data!NIP harus</li> </ul>	

<p>berupa 18 digit angka (0-9)".</p> <ul style="list-style-type: none"> <li>• Jika NIP sudah pernah direkam, maka sistem gagal menyimpan data, memunculkan pesan "gagal merekam! NIP sudah ada"</li> </ul>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### A.5. Deskripsi Use Case Hapus Bendahara

Tabel A. 5 Deskripsi Use Case Hapus Bendahara

<b>UC-5 –ProsesHapus Bendahara</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Bendahara	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Bendahara yang dipilih dari Daftar Bendahara dan menampilkan Halaman Daftar Bendahara terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Bendahara tanpa melakukan apapun.	

### A.6. Deskripsi Use Case Tayang Beban

Tabel A. 6 Deskripsi Use Case Tayang Beban

<b>UC-6– Proses Tayang Beban</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum,	

Admin mengklik tombol Beban
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Beban.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Beban
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Beban, sesuai dengan tabel beban</li> </ul>

## A.7. Deskripsi Use Case Rekam Beban

Tabel A. 7 Deskripsi Use Case Rekam Beban

UC-7-Proses Rekam Beban	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Beban	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Beban. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Beban, dan menampilkan pesan sukses pada Halaman Rekam Beban.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi."</li> <li>• Jika kode beban sudah pernah direkam, sistem menampilkan pesan "Kode beban sudah pernah direkam".</li> <li>• Jika kode beban direkam selain menggunakan karakter abjad aA - zZ, pesan "gagal merekam data"</li> </ul>	

### A.8. Deskripsi Use Case Ubah Beban

Tabel A. 8 Deskripsi Use Case Ubah Beban

UC-8 –Proses Ubah Bendahara	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Bendahara	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Beban yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Bebandan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Bebandan menampilkan pesan sukses pada Halaman Daftar Beban	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode beban sudah pernah direkam, sistem menampilkan pesan "Kode beban sudah pernah direkam".</li> <li>• Jika kode beban direkam selain menggunakan karakter abjad aA – zZ, pesan "gagal merekam data"</li> </ul>	

### A.9. Deskripsi Use Case Hapus Beban

Tabel A. 9 Deskripsi Use Case Hapus Beban

UC-9 –Proses Hapus Beban	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b>	

Admin berada pada halaman Daftar Beban
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.
<b>Post-conditions:</b> Sistem menghapus data Beban yang dipilih dari Daftar Bebandan menampilkan Halaman Daftar Bebanterbaru.
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Perubahan Kondisi tanpa melakukan apapun.

### A.10. Deskripsi Use Case Tayang Fungsi

Tabel A. 10 Deskripsi Use Case Tayang Fungsi

UC-10- Proses Tayang Fungsi	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Bendahara, Admin mengklik tombol Fungsi	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Fungsi.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Fungsi	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Fungsi, sesuai dengan tabel t fungsi	

### A.11. eskripsi Use Case Rekam Fungsi

Tabel A. 11 Deskripsi Use Case Rekam Fungsi

UC-11—Proses Rekam Fungsi	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Fungsi	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Fungsi. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Fungsi, dan menampilkan pesan sukses pada Halaman Rekam Ruangan.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi."</li> <li>• Jika kode Fungsi sudah pernah direkam, sistem menampilkan pesan "Kode ruangan sudah pernah direkam".</li> <li>• Jika kode fungsi diinput selain numeric (0 – 9), pesan "gagal merekam"</li> </ul>	

### A.12. Deskripsi Use Case Ubah Fungsi

Tabel A. 12 Deskripsi Use Case Ubah Fungsi

UC-12 –Proses Ubah Fungsi	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Fungsi	
<b>Triggers:</b>	

Admin mengklik tombol ubah pada baris data Fungsi yang akan diubah.

**Basic course:**

Sistem menampilkan Halaman Ubah Fungsi dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.

**Post-conditions:**

Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Fungsi dan menampilkan pesan sukses pada Halaman Daftar Fungsi

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika kode Fungsi sudah pernah direkam, sistem menampilkan pesan "Kode ruangan sudah pernah direkam".
- Jika kode fungsi diinput selain numeric (0 – 9), pesan "gagal menyimpan"

### A.13. Deskripsi Use Case Hapus Fungsi

Tabel A. 13 Deskripsi Use Case Hapus Fungsi

UC-13 –Proses Hapus Fungsi	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level: 3</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Fungsi	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Fungsi yang dipilih dari Daftar Fungsi dan menampilkan Halaman Daftar Fungsi terbaru.	



**Alternate courses:**

Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Fungsitanpa melakukan apapun.

**A.14. Deskripsi Use Case Tayang Jenis Dokumen**

Tabel A. 14 Deskripsi Use Case Tayang Jenis Dokumen

<b>UC-14</b> Proses Tayang Jenis Dokumen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Bendahara, Admin mengklik tombol Jenis Dokumen	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Jendok.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Jendok	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Jendok, sesuai dengan tabel t_jendok	

**A.15. Deskripsi Use Case Rekam Jenis Dokumen**

Tabel A. 15 Deskripsi Use Case Rekam Jenis Dokumen

<b>UC-15</b> Proses Rekam Jenis Dokumen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jendok	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b>	

Sistem menampilkan Halaman Rekam Jendok. Admin mengisi field-field yang ada lalu menekan tombol Simpan.

**Post-conditions:**

Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Jendok, dan menampilkan pesan sukses pada Halaman Rekam Ruangan.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi."
- Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".
- Jika kode bendahara sudah pernah direkam, sistem menampilkan pesan "Kode Jendok sudah pernah direkam".

### A.16. Deskripsi Use Case Ubah Jenis Dokumen

Tabel A. 16 Deskripsi Use Case Ubah Jenis Dokumen

UC-16 –Proses Ubah Jenis Dokumen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jendok	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Jendok yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Jendok dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Bendahara dan menampilkan pesan sukses pada Halaman Daftar Bendahara	
<b>Alternate courses:</b>	

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".
- Jika kode Jendok sudah pernah direkam, sistem menampilkan pesan "Kode Jendoksudah pernah direkam".

### A.17. Deskripsi Use Case Hapus Jenis Dokumen

Tabel A. 17 Deskripsi Use Case Hapus Jenis Dokumen

UC-17 –ProsesHapus Jenis Dokumen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jendok	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Jendok yang dipilih dari Daftar Jendok dan menampilkan Halaman Daftar Jenis Dokumen terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Jenis Dokumen tanpa melakukan apapun.	

### A.18. Deskripsi Use Case Tayang MAK

Tabel A. 18 Deskripsi Use Case Tayang MAK

UC-18– Proses Tayang MAK	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b>	

<i>Admin berada pada halaman Referensi</i>
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol MAK
<b>Basic course:</b> Sistem menampilkan Halaman Daftar .
<b>Post-conditions:</b> Sistem menampilkan daftar referensi MAK
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi MAK, sesuai dengan tabel t_mak

### A.19. Deskripsi Use Case Rekam MAK

Tabel A. 19 Deskripsi Use Case Rekam MAK

UC-19—Proses Rekam MAK	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar MAK	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam MAK. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar MAK, dan menampilkan pesan sukses pada Halaman Rekam MAK.	
<b>Alternate courses:</b> • Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi." • Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".	

- Jika kode MAK sudah pernah direkam, sistem menampilkan pesan "Kode MAK sudah pernah direkam".

## A.20. Deskripsi Use Case Ubah MAK

Tabel A. 20 Deskripsi Use Case Ubah MAK

<b>UC-20 –Proses Ubah MAK</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Bendahara	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data MAK yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah MAK dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar MAK dan menampilkan pesan sukses pada Halaman Daftar MAK	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode MAK sudah pernah direkam, sistem menampilkan pesan "Kode MAK sudah pernah direkam".</li> </ul>	

## A.21. Deskripsi Use Case Hapus MAK

Tabel A. 21 Deskripsi Use Case Hapus MAK

<b>UC-21 –Proses Hapus MAK</b>	
<b>Primary Actor:</b>	<b>Level:</b>

Administrator (Admin)	Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar MAK	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data MAK yang dipilih dari Daftar MAK dan menampilkan Halaman Daftar Bendahara terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar MAK tanpa melakukan apapun.	

## A.22. Deskripsi Use Case Tayang Jenis SPM

Tabel A. 22 Deskripsi Use Case Tayang Jenis SPM

UC-22– Proses Tayang Jenis SPM	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Pembayaran, Admin mengklik tombol Jenis SPM	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Jenis SPM.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Jenis SPM	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Jenis SPM, sesuai dengan tabel t_jenspm	

### A.23. Deskripsi Use Case Rekam Jenis SPM

Tabel A. 23 Deskripsi Use Case Rekam Jenis SPM

UC-23—Proses Rekam Jenis SPM	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis SPM	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Jenis SPM. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Jenis SPM, dan menampilkan pesan sukses pada Halaman Rekam Jenis SPM	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Jenis SPM sudah pernah direkam, sistem menampilkan pesan "Kode Jenis SPM sudah pernah direkam".</li> </ul>	

### A.24. Deskripsi Use Case Ubah Jenis SPM

Tabel A. 24 Deskripsi Use Case Ubah Jenis SPM

UC-24 —Proses Ubah Jenis SPM	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis SPM	

<p><b>Triggers:</b> Admin mengklik tombol ubah pada baris data Jenis SPM yang akan diubah.</p>
<p><b>Basic course:</b> Sistem menampilkan Halaman Ubah Jenis SPM dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.</p>
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Jenis SPM dan menampilkan pesan sukses pada Halaman Daftar Jenis SPM</p>
<p><b>Alternate courses:</b></p> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Jenis SPM sudah pernah direkam, sistem menampilkan pesan "Kode Jenis SPM sudah pernah direkam".</li> </ul>

## A.25. Deskripsi Use Case Hapus Jenis SPM

Tabel A. 25 Deskripsi Use Case Hapus Jenis SPM

UC-25 –Proses Hapus Jenis SPM	
<p><b>Primary Actor:</b> Administrator (Admin)</p>	<p><b>Level:</b> Pengguna Goal</p>
<p><b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis SPM</p>	
<p><b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.</p>	
<p><b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.</p>	
<p><b>Post-conditions:</b> Sistem menghapus data Jenis SPM yang dipilih dari Daftar Jenis</p>	



SPM dan menampilkan Halaman Daftar Jenis SPM terbaru.
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Jenis SPM tanpa melakukan apapun.

## A.26. Deskripsi Use Case Tayang Cara Bayar

Tabel A. 26 Deskripsi Use Case Tayang Cara Bayar

UC-26– Proses Tayang Cara Bayar	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Cara Bayar	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Cara Bayar.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Cara Bayar	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Cara Bayar, sesuai dengan tabel t_crbayar	

## A.27. Deskripsi Use Case Rekam Cara Bayar

Tabel A. 27 Deskripsi Use Case Rekam Cara Bayar

UC-27– Proses Rekam Cara Bayar	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Cara Bayar	
<b>Triggers:</b> Admin mengklik link Rekam Data	

<p><b>Basic course:</b> Sistem menampilkan Halaman Rekam Cara Bayar. Admin mengisi field-field yang ada lalu menekan tombol Simpan.</p>
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Cara Bayar, dan menampilkan pesan sukses pada Halaman Rekam Cara Bayar.</p>
<p><b>Alternate courses:</b></p> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Cara Bayar sudah pernah direkam, sistem menampilkan pesan "Kode Cara Bayarsudah ada".</li> </ul>

## A.28. Deskripsi Use Case Ubah Cara Bayar

Tabel A. 28 Deskripsi Use Case Ubah Cara Bayar

UC-28 –Proses Ubah Bendahara	
<p><b>Primary Actor:</b> Administrator (Admin)</p>	<p><b>Level:</b> Pengguna Goal</p>
<p><b>Pre-conditions:</b> Admin berada pada halaman Daftar Cara Bayar</p>	
<p><b>Triggers:</b> Admin mengklik tombol ubah pada baris data Cara Bayar yang akan diubah.</p>	
<p><b>Basic course:</b> Sistem menampilkan Halaman Ubah Cara Bayar dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.</p>	
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Cara Bayar dan menampilkan pesan sukses pada Halaman Daftar Cara Bayar</p>	

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".
- Jika kode Cara Bayar sudah pernah direkam, sistem menampilkan pesan "Kode Cara Bayarsudah ada".

**A.29. Deskripsi Use Case Hapus Cara Bayar**

Tabel A. 29 Deskripsi Use Case Hapus Cara Bayar

<b>UC-29 –ProsesHapus Cara Bayar</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Cara Bayar	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Cara Bayaryang dipilih dari Daftar Cara Bayardan menampilkan Halaman Daftar Cara Bayarterbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Cara Bayartanpa melakukan apapun.	

**A.30. Deskripsi Use Case Tayang Jenis Dokumen**

Tabel A. 30 Deskripsi Use Case Tayang Jenis Dokumen

<b>UC-30– Proses Tayang Jenis Dokumen</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b>	

<i>Admin berada pada halaman Referensi</i>
<b>Triggers:</b> Admin mengklik tombol Referensi Bendahara, Admin mengklik tombol Jenis Dokumen
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Jenis Dokumen.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Jenis Dokumen
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Jenis Dokumen, sesuai dengan tabel t_jendok

### A.31. Deskripsi Use Case RekamJenis Dokumen

Tabel A. 31 Deskripsi Use Case Rekam Jenis Dokumen

UC-31—Proses Rekam Jenis Dokumen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis Dokumen	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Jenis Dokumen. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Jenis Dokumen, dan menampilkan pesan sukses pada Halaman Rekam Jenis Dokumen.	
<b>Alternate courses:</b> • Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi". • Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".	

- Jika kode Jenis Dokumen sudah pernah direkam, sistem menampilkan pesan "Kode Jenis Dokumen sudah ada".

### A.32. Deskripsi Use Case Ubah Jenis Dokumen

Tabel A. 32 Deskripsi Use Case Ubah Jenis Dokumen

<b>UC-32 –Proses Ubah Jenis Dokumen</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis Dokumen	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Jenis Dokumen yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Jenis Dokumen dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Jenis Dokumen dan menampilkan pesan sukses pada Halaman Daftar Jenis Dokumen	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Jenis Dokumen sudah pernah direkam, sistem menampilkan pesan "Kode Jenis Dokumen sudah ada".</li> </ul>	

### A.33. Deskripsi Use Case Hapus Jenis Dokumen

Tabel A. 33 Deskripsi Use Case Hapus Jenis Dokumen

**UC-33 –Proses Hapus Bendahara**

<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis Dokumen	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Jenis Dokumen yang dipilih dari Daftar Jenis Dokumen dan menampilkan Halaman Daftar Jenis Dokumen terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Jenis Dokumen tanpa melakukan apapun.	

### A.34. Deskripsi Use Case Tayang Sumber Dana

Tabel A. 34 Deskripsi Use Case Tayang Sumber Dana

<b>UC-34</b> – Proses Tayang Sumber Dana	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Sumber Dana	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Sumber Dana.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Sumber Dana	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Sumber Dana, sesuai	

dengan tabel t\_sdana

### A.35. Deskripsi Use Case Rekam Sumber Dana

Tabel A. 35 Deskripsi Use Case Rekam Sumber Dana

UC-2.2–Proses Rekam Sumber Dana	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Sumber Dana	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Sumber Dana. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Sumber Dana, dan menampilkan pesan sukses pada Halaman Rekam Sumber Dana	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Sumber Dana sudah pernah direkam, sistem menampilkan pesan "Kode Sumber Dana sudah ada".</li> </ul>	

### A.36. Deskripsi Use Case Ubah Sumber Dana

Tabel A. 36 Deskripsi Use Case Ubah Sumber Dana

UC-36 –Proses Ubah Sumber Dana	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b>	

<i>Admin berada pada halaman Daftar Sumber Dana</i>
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Sumber Danayang akan diubah.
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Sumber Danadan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Sumber Danadan menampilkan pesan sukses pada Halaman Daftar Sumber Dana
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Sumber Danasudah pernah direkam, sistem menampilkan pesan "Kode Sumber Dana sudah ada".</li> </ul>

### A.37. Deskripsi Use Case HapusSumber Dana

Tabel A. 37Deskripsi Use Case Hapus Sumber Dana

UC-37 –ProsesHapus Sumber Dana	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Sumber Dana	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b>	



Sistem menghapus data Bendahara yang dipilih dari Daftar Sumber Danadan menampilkan Halaman Daftar Sumber Danaterbaru.

**Alternate courses:**

Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Sumber Danatanpa melakukan apapun.

### A.38. Deskripsi Use Case Tayang Jenis SPM

Tabel A. 38 Deskripsi Use Case Tayang Jenis SPM

<b>UC-38– Proses Tayang Jenis SPM</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Pembayaran, Admin mengklik tombol Jenis SPM	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Jenis SPM.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Jenis SPM	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Jenis SPM, sesuai dengan tabel t_jenspm</li> </ul>	

### A.39. Deskripsi Use Case Rekam Jenis SPM

Tabel A. 39 Deskripsi Use Case Rekam Jenis SPM

<b>UC-39– Proses Rekam Jenis SPM</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis SPM	

<b>Triggers:</b> Admin mengklik link Rekam Data
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Jenis SPM. Admin mengisi field-field yang ada lalu menekan tombol Simpan.
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Jenis SPM, dan menampilkan pesan sukses pada Halaman Rekam Jenis SPM
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kodedan jam selesai tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".</li> <li>• Jika kode Jenis SPM sudah pernah direkam, sistem menampilkan pesan "Kode Jenis SPM sudah ada".</li> </ul>

#### A.40. Deskripsi Use Case Ubah Jenis SPM

Tabel A. 40 Deskripsi Use Case Ubah Jenis SPM

UC-40 –Proses Ubah Jenis SPM	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis SPM	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Jenis SPM yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Jenis SPM dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b>	

Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Jenis SPM dan menampilkan pesan sukses pada Halaman Daftar Jenis SPM

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika kodedan jam selesai tidak berupa angka, sistem menampilkan pesan "gagal menyimpan data harus berupa angka 0 - 9".
- Jika kode Jenis SPM sudah pernah direkam, sistem menampilkan pesan "Kode Jenis SPM sudah ada".

#### A.41. Deskripsi Use Case Hapus Jenis SPM

Tabel A. 41 Deskripsi Use Case Hapus Jenis SPM

UC-41 –Proses Hapus Jenis SPM	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Jenis SPM	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Jenis SPM yang dipilih dari Daftar Jenis SPM dan menampilkan Halaman Daftar Bendahara terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Jenis SPM tanpa melakukan apapun.	

### A.42. Deskripsi Use Case Tayang Aset Tetap

Tabel A. 42 Deskripsi Use Case Tayang Aset Tetap

UC-42– Proses Tayang Aset Tetap	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Aset, Admin mengklik tombol Aset Tetap	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Aset Tetap.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Aset Tetap	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Aset Tetap, sesuai dengan tabel t_asetap	

### A.43. Deskripsi Use Case Rekam Aset Tetap

Tabel A. 43 Deskripsi Use Case Rekam Aset Tetap

UC-43–Proses Rekam Aset Tetap	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Aset Tetap	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Aset Tetap. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan	

masuk ke dalam Daftar Aset Tetap, dan menampilkan pesan sukses pada Halaman Rekam Aset Tetap.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Aset Tetap sudah pernah direkam, sistem menampilkan pesan "Kode Aset Tetap sudah ada".

#### A.44. Deskripsi Use Case Ubah Aset Tetap

Tabel A. 44 Deskripsi Use Case Ubah Aset Tetap

<b>UC-44 – Proses Ubah Aset Tetap</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Aset Tetap	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Aset Tetap yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Aset Tetap dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Aset Tetap dan menampilkan pesan sukses pada Halaman Daftar Aset Tetap	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> </ul>	

- Jika kode Aset Tetap sudah pernah direkam, sistem menampilkan pesan "Kode Aset Tetap sudah ada".

#### A.45. Deskripsi Use Case Hapus Aset Tetap

Tabel A. 45 Deskripsi Use Case Hapus Aset Tetap

<b>UC-45 – Proses Hapus Aset Tetap</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Aset Tetap	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Aset Tetap menghapus data Aset Tetap yang dipilih dari Daftar Aset Tetap dan menampilkan Halaman Daftar Aset Tetap terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Aset Tetap tanpa melakukan apapun.	

#### A.46. Deskripsi Use Case Tayang Cara Perolehan

Tabel A. 46 Deskripsi Use Case Tayang Cara Perolehan

<b>UC-46 – Proses Tayang Cara Perolehan</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Aset, Admin mengklik tombol Cara Perolehan	

<b>Basic course:</b> Sistem menampilkan Halaman Daftar Cara Perolehan.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Cara Perolehan
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Cara Perolehan, sesuai dengan tabel t_croleh</li> </ul>

#### A.47. Deskripsi Use Case Rekam Cara Perolehan

Tabel A. 47 Deskripsi Use Case Rekam Cara Perolehan

UC-47--Proses Rekam Cara Perolehan	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Cara Perolehan	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Cara Perolehan. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Cara Perolehan, dan menampilkan pesan sukses pada Halaman Rekam Cara Perolehan.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode Cara Perolehan tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Aset Tetap sudah pernah direkam, sistem menampilkan pesan "Kode Cara Perolehan sudah ada".</li> </ul>	



#### A.48. Deskripsi Use Case Ubah Cara Perolehan

Tabel A. 48 Deskripsi Use Case Ubah Cara Perolehan

<b>UC-48 –Proses Ubah Cara Perolehan</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Cara Perolehan	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Cara Perolehan yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Cara Perolehan dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Cara Perolehan dan menampilkan pesan sukses pada Halaman Daftar Cara Perolehan	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika kode Cara Perolehan tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Kode Cara Perolehan sudah pernah direkam, sistem menampilkan pesan "Kode Cara Perolehan sudah ada".</li> </ul>	

#### A.49. Deskripsi Use Case Hapus Cara Perolehan

Tabel A. 49 Deskripsi Use Case Hapus Cara Perolehan

<b>UC-49 –Proses Hapus Cara Perolehan</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal



<b>Pre-conditions:</b> Admin berada pada halaman Daftar Cara Perolehan
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.
<b>Post-conditions:</b> Sistem menghapus data Cara Perolehanyang dipilih dari Daftar Cara Perolehandan menampilkan Halaman Daftar Cara Perolehanterbaru.
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Cara Perolehantanpa melakukan apapun.

#### A.50. Deskripsi Use Case Tayang SSKel Barang

Tabel A. 50 Deskripsi Use Case Tayang SSKel Barang

UC-50– Proses Tayang SSKel Barang	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Aset, Admin mengklik tombol SSKel Barang	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Bendahara.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Bendahara	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Bendahara, sesuai dengan tabel t bendahara	

### A.51. Deskripsi Use Case RekamSSKel Barang

Tabel A. 51 Deskripsi Use Case Rekam SSKel Barang

UC-51-Proses Rekam SSKel Barang	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar SSKel Barang	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam SSKel Barang. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar SSKel Barang, dan menampilkan pesan sukses pada Halaman Rekam SSKel Barang.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode SSKel Barangsudah pernah direkam, sistem menampilkan pesan "Kode SSKel Barang sudah ada"."</li> </ul>	

### A.52. Deskripsi Use Case UbahSSKelBarang

Tabel A. 52 Deskripsi Use Case Ubah SSKel Barang

UC-52 -Proses Ubah SSKel Barang	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar SSKel Barang	

<p><b>Triggers:</b> Admin mengklik tombol ubah pada baris data SSKel Barang yang akan diubah.</p>
<p><b>Basic course:</b> Sistem menampilkan Halaman Ubah SSKel Barang dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.</p>
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar SSKel Barang dan menampilkan pesan sukses pada Halaman Daftar SSKel Barang.</p>
<p><b>Alternate courses:</b></p> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode SSKel Barang sudah pernah direkam, sistem menampilkan pesan "Kode SSKel Barang sudah ada".</li> </ul>

### A.53. Deskripsi Use Case Hapus SSKel BMN

Tabel A. 53 Deskripsi Use Case Hapus SSKel BMN

UC-53 –Proses Hapus SSKel BMN	
<p><b>Primary Actor:</b> Administrator (Admin)</p>	<p><b>Level:</b> Pengguna Goal</p>
<p><b>Pre-conditions:</b> Admin berada pada halaman Daftar SSKel BMN</p>	
<p><b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.</p>	
<p><b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.</p>	
<p><b>Post-conditions:</b></p>	

Sistem menghapus data SSKel BMN yang dipilih dari Daftar SSKel BMN dan menampilkan Halaman Daftar SSKel BMN terbaru.

**Alternate courses:**

Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar SSKel BMN tanpa melakukan apapun.

#### A.54. Deskripsi Use Case Tayang Map Barang

Tabel A. 54 Deskripsi Use Case Tayang Map Barang

<b>UC-54</b> – Proses Tayang Bendahara	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Persediaan, Admin mengklik tombol Map Barang	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Map Barang.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Map Barang	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Map Barang, sesuai dengan tabel t_mapbrg	

#### A.55. Deskripsi Use Case Rekam Map Barang

Tabel A. 55 Deskripsi Use Case Rekam Map Barang

<b>UC-56</b> – Proses Rekam Map Barang	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Map Barang	

<b>Triggers:</b> Admin mengklik link Rekam Data
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Map Barang. Admin mengisi field-field yang ada lalu menekan tombol Simpan.
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Map Barang, dan menampilkan pesan sukses pada Halaman Rekam Map Barang
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Map Barangsudah pernah direkam, sistem menampilkan pesan "Kode Map Barangsudah ada"."</li> </ul>

### A.56. Deskripsi Use Case UbahMap Barang

Tabel A. 56Deskripsi Use Case Ubah Map Barang

<b>UC-56 –Proses Ubah Map Barang</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Map Barang	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Map Barangyang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Map Barangdan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b>	

Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Map Barang dan menampilkan pesan sukses pada Halaman Daftar Map Barang

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Map Barang sudah pernah direkam, sistem menampilkan pesan "Kode Map Barang sudah ada".

### A.57. Deskripsi Use Case Hapus Map Barang

Tabel A. 57 Deskripsi Use Case Hapus Map Barang

UC-57 –ProsesHapus Map Barang	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Map Barang	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Bendahara yang dipilih dari Daftar Map Barang dan menampilkan Halaman Daftar Map Barangterbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Map Barangtanpa melakukan apapun.	

### A.58. Deskripsi Use Case Tayang Kanwil

Tabel A. 58 Deskripsi Use Case Tayang Kanwil

UC-58– Proses Tayang Kanwil	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Persediaan, Admin mengklik tombol Kanwil	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Kanwil.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Kanwil	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Kanwil, sesuai dengan tabel t_kanwil</li> </ul>	

### A.59. Deskripsi Use Case Rekam Kanwil

Tabel A. 59 Deskripsi Use Case Rekam Kanwil

UC-59–Proses Rekam Kanwil	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Kanwil	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Kanwil. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan	

masuk ke dalam Daftar Kanwil, dan menampilkan pesan sukses pada Halaman Rekam Kanwil.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Kanwil sudah pernah direkam, sistem menampilkan pesan "Kode Kanwil sudah ada".

### A.60. Deskripsi Use Case Ubah Kanwil

Tabel A. 60 Deskripsi Use Case Ubah Kanwil

<b>UC-60</b> –Proses Ubah Kanwil	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Kanwil	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Kanwil yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Kanwil dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Kanwil dan menampilkan pesan sukses pada Halaman Daftar Kanwil	
<b>Alternate courses:</b>	
<ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem</li> </ul>	



menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".

- Jika kode Kanwil sudah pernah direkam, sistem menampilkan pesan "Kode Kanwil sudah ada".

### A.61. Deskripsi Use Case Hapus Kanwil

Tabel A. 61 Deskripsi Use Case Hapus Kanwil

UC-61 – Proses Hapus Kanwil	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Kanwil	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Kanwil yang dipilih dari Daftar Kanwil dan menampilkan Halaman Daftar Kanwil terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Kanwil tanpa melakukan apapun.	

### A.62. Deskripsi Use Case Tayang Akun

Tabel A. 62 Deskripsi Use Case Tayang Akun

UC-62 – Proses Tayang Akun	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b>	

Admin mengklik tombol Referensi Anggaran, Admin mengklik tombol Akun
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Akun.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Akun
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Akun, sesuai dengan tabel t akun

### A.63. Deskripsi Use Case Rekam Akun

Tabel A. 63 Deskripsi Use Case Rekam Akun

UC-63-Proses Rekam Akun	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Akun	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Akun. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Akun, dan menampilkan pesan sukses pada Halaman Rekam Akun.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Akun sudah pernah direkam, sistem menampilkan pesan "Kode Akun sudah ada".</li> </ul>	

#### A.64. Deskripsi Use Case UbahAkun

Tabel A. 64 Deskripsi Use Case Ubah Akun

UC-64 –Proses Ubah Akun	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Akun	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Akun yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Akun dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Akun dan menampilkan pesan sukses pada Halaman Daftar Akun	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Akun sudah pernah direkam, sistem menampilkan pesan "Kode Akun sudah ada".</li> </ul>	

#### A.65. Deskripsi Use Case HapusAkun

Tabel A. 65 Deskripsi Use Case Hapus Akun

UC-65 –Proses Hapus Akun	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b>	

Admin berada pada halaman Daftar Akun
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin Ya.
<b>Post-conditions:</b> Sistem menghapus data Bendahara yang dipilih dari Daftar Akundan menampilkan Halaman Daftar Akunterbaru.
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Akun tanpa melakukan apapun.

### A.66. Deskripsi Use Case Tayang Blokir

Tabel A. 66 Deskripsi Use Case Tayang Blokir

UC-66– Proses Tayang Blokir	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Anggaran, Admin mengklik tombol Blokir	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Blokir.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Blokir	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Blokir, sesuai dengan tabel t_blokir	

### A.67. Deskripsi Use Case Rekam Blokir

Tabel A. 67 Deskripsi Use Case Rekam Blokir

UC-67– Proses Rekam Blokir
----------------------------

<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Blokir	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Blokir. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Blokir, dan menampilkan pesan sukses pada Halaman Rekam Blokir.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Blokirsudah pernah direkam, sistem menampilkan pesan "Kode Blokirsudah ada".</li> </ul>	

### A.68. Deskripsi Use Case UbahBlokir

Tabel A. 68Deskripsi Use Case Ubah Blokir

UC-68 –Proses Ubah Blokir	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Blokir	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Blokiryang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Blokirdan memuat data yang	

akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.

**Post-conditions:**

Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Blokir dan menampilkan pesan sukses pada Halaman Daftar Blokir

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Blokir sudah pernah direkam, sistem menampilkan pesan "Kode Blokir sudah ada".

### A.69. Deskripsi Use Case Hapus Blokir

Tabel A. 69 Deskripsi Use Case Hapus Blokir

UC-69 – Proses Hapus Blokir	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Blokir	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Blokir yang dipilih dari Daftar Blokir dan menampilkan Halaman Daftar Blokir terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Blokir tanpa melakukan apapun.	

### A.70. Deskripsi Use Case Tayang Departemen

Tabel A. 70 Deskripsi Use Case Tayang Departemen

UC-70– Proses Tayang Departemen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Anggaran, Admin mengklik tombol Departemen	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Departemen.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Departemen	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Departemen, sesuai dengan tabel t_dept</li> </ul>	

### A.71. Deskripsi Use Case Rekam Departemen

Tabel A. 71 Deskripsi Use Case Rekam Departemen

UC-71–Proses Rekam Departemen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Departemen	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Departemen. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Departemen, dan menampilkan pesan	

sukses pada Halaman Rekam Departemen.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Departemen sudah pernah direkam, sistem menampilkan pesan "Kode Departemen sudah ada".

## A.72. Deskripsi Use Case UbahDepartemen

Tabel A. 72 Deskripsi Use Case Ubah Departemen

UC-72 –Proses Ubah Departemen	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Departemen	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Departemen yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Departemen dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Departemen dan menampilkan pesan sukses pada Halaman Daftar Departemen	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> </ul>	



- Jika kode Departemen sudah pernah direkam, sistem menampilkan pesan "Kode Departemen sudah ada".

### A.73. Deskripsi Use Case Hapus Departemen

Tabel A. 73 Deskripsi Use Case Hapus Departemen

<b>UC-73 –Proses Hapus Departemen</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Departemen	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Departemen yang dipilih dari Daftar Departemen dan menampilkan Halaman Daftar Departemen terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Departemen tanpa melakukan apapun.	

### A.74. Deskripsi Use Case Tayang GBKPK

Tabel A. 74 Deskripsi Use Case Tayang GBKPK

<b>UC-74 – Proses Tayang GBKPK</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Anggaran	

<b>Basic course:</b> Sistem menampilkan Halaman Daftar GBKPK.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi GBKPK
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi GBKPK, sesuai dengan tabel t_gbkpk</li> </ul>

### A.75. Deskripsi Use Case RekamGBKPK

Tabel A. 75 Deskripsi Use Case Rekam GBKPK

UC-75-Proses Rekam GBKPK	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar GBKPK	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam GBKPK. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar GBKPK, dan menampilkan pesan sukses pada Halaman Rekam GBKPK.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode GBKPKsudah pernah direkam, sistem menampilkan pesan "Kode GBKPKsudah ada"."</li> </ul>	

### A.76. Deskripsi Use Case UbahGBKPK

Tabel A. 76 Deskripsi Use Case Ubah GBKPK

<b>UC-76 –Proses Ubah GBKPK</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar GBKPK	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data GBKPK yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah GBKPK dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar GBKPK dan menampilkan pesan sukses pada Halaman Daftar GBKPK	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode GBKPK sudah pernah direkam, sistem menampilkan pesan "Kode GBKPK sudah ada".</li> </ul>	

### A.77. Deskripsi Use Case HapusGBKPK

Tabel A. 77 Deskripsi Use Case Hapus GBKPK

<b>UC-2.4 –Proses Hapus GBKPK</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b>	

Admin berada pada halaman Daftar GBKPK
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.
<b>Post-conditions:</b> Sistem menghapus data GBKPK yang dipilih dari Daftar GBKPK dan menampilkan Halaman Daftar GBKPK terbaru.
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar GBKPK tanpa melakukan apapun.

### A.78. Deskripsi Use Case Tayang Kegiatan

Tabel A. 78 Deskripsi Use Case Tayang Kegiatan

UC-78– Proses Tayang Kegiatan	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Kegiatan	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Kegiatan.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Kegiatan	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Kegiatan, sesuai dengan tabel t giat	

### A.79. Deskripsi Use Case RekamKegiatan

Tabel A. 79 Deskripsi Use Case Rekam Kegiatan

UC-79–Proses Rekam Kegiatan	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Kegiatan	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Kegiatan. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Kegiatan, dan menampilkan pesan sukses pada Halaman Rekam Kegiatan.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Kegiatan sudah pernah direkam, sistem menampilkan pesan "Kode Kegiatan sudah ada".</li> </ul>	

### A.80. Deskripsi Use Case UbahKegiatan

Tabel A. 80 Deskripsi Use Case Ubah Kegiatan

UC-80 –Proses Ubah Kegiatan	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Kegiatan	
<b>Triggers:</b>	

Admin mengklik tombol ubah pada baris data Kegiatan yang akan diubah.
<p><b>Basic course:</b> Sistem menampilkan Halaman Ubah Kegiatan dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.</p>
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Kegiatan dan menampilkan pesan sukses pada Halaman Daftar Kegiatan</p>
<p><b>Alternate courses:</b></p> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Kegiatan sudah pernah direkam, sistem menampilkan pesan "Kode Kegiatan sudah ada".</li> </ul>

### A.81. Deskripsi Use Case Hapus Kegiatan

Tabel A. 81 Deskripsi Use Case Hapus Kegiatan

UC-81 –Proses Hapus Kegiatan	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Kegiatan	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Kegiatan yang dipilih dari Daftar Kegiatan dan menampilkan Halaman Daftar Bendahara terbaru.	

**Alternate courses:**

Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Kegiatan tanpa melakukan apapun.

**A.82. Deskripsi Use Case Tayang Unit**

Tabel A. 82 Deskripsi Use Case Tayang Unit

<b>UC-82– Proses Tayang Unit</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Anggaran, Admin mengklik tombol Unit	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Unit.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Unit	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Unit, sesuai dengan tabel unit	

**A.83. Deskripsi Use Case Rekam Unit**

Tabel A. 83 Deskripsi Use Case Rekam Unit

<b>UC-83– Proses Rekam Unit</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Unit	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Unit. Admin mengisi field-	

field yang ada lalu menekan tombol Simpan.
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Unit, dan menampilkan pesan sukses pada Halaman Rekam Unit.</p>
<p><b>Alternate courses:</b></p> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Unit sudah pernah direkam, sistem menampilkan pesan "Kode Unit sudah ada"."</li> </ul>

#### A.84. Deskripsi Use Case Ubah Unit

Tabel A. 84 Deskripsi Use Case Ubah Unit

UC-84 –Proses Ubah Unit	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Unit	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Unit yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Unit dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Unit dan menampilkan pesan sukses pada Halaman Daftar Unit	
<b>Alternate courses:</b>	
<ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan</li> </ul>	



"Semua field harus diisi".

- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Unit sudah pernah direkam, sistem menampilkan pesan "Kode Unit sudah ada".

### A.85. Deskripsi Use Case HapusUnit

Tabel A. 85Deskripsi Use Case Hapus Unit

UC-85 –ProsesHapus Unit	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Unit	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Unityang dipilih dari Daftar Unitdan menampilkan Halaman Daftar Unitterbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Unittanpa melakukan apapun.	

### A.86. Deskripsi Use Case Tayang IB

Tabel A. 86Deskripsi Use Case Tayang IB

UC-86– Proses Tayang IB	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	

<b>Triggers:</b> Admin mengklik tombol Referensi Anggaran, Admin mengklik tombol IB
<b>Basic course:</b> Sistem menampilkan Halaman Daftar IB.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi IB
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi IB, sesuai dengan tabel t ib

### A.87. Deskripsi Use Case RekamIB

Tabel A. 87 Deskripsi Use Case Rekam IB

UC-87-Proses Rekam IB	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar IB	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam IB. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan masukan ke dalam Daftar IB, dan menampilkan pesan sukses pada Halaman Rekam IB.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode IB sudah pernah direkam, sistem menampilkan</li> </ul>	

pesan "Kode IB sudah ada"."
-----------------------------

### A.88. Deskripsi Use Case UbahIB

Tabel A. 88 Deskripsi Use Case Ubah IB

UC-88 –Proses Ubah IB	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar IB	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data IB yang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah IB dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar IB dan menampilkan pesan sukses pada Halaman Daftar IB	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode IB sudah pernah direkam, sistem menampilkan pesan "Kode IB sudah ada"."</li> </ul>	

### A.89. Deskripsi Use Case HapusIB

Tabel A. 89 Deskripsi Use Case Hapus IB

UC-89 –Proses Hapus IB	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal

<b>Pre-conditions:</b> Admin berada pada halaman Daftar IB
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.
<b>Post-conditions:</b> Sistem menghapus data IB yang dipilih dari Daftar IB dan menampilkan Halaman Daftar IB terbaru.
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar IB tanpa melakukan apapun.

## A.90. Deskripsi Use Case Tayang Output

Tabel A. 90 Deskripsi Use Case Tayang Output

UC-90– Proses Tayang Output	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Output	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Output.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Output	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Output, sesuai dengan tabel t_output	

### A.91. Deskripsi Use Case RekamOutput

Tabel A. 91 Deskripsi Use Case Rekam Output

<b>UC-91</b> –Proses Rekam Output	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Output	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Output. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Output, dan menampilkan pesan sukses pada Halaman Rekam Output.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Output sudah pernah direkam, sistem menampilkan pesan "Kode Output sudah ada".</li> </ul>	

### A.92. Deskripsi Use Case UbahOutput

Tabel A. 92 Deskripsi Use Case Ubah Output

<b>UC-92</b> –Proses Ubah Output	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Output	
<b>Triggers:</b>	

Admin mengklik tombol ubah pada baris data Output yang akan diubah.
<p><b>Basic course:</b> Sistem menampilkan Halaman Ubah Output dan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.</p>
<p><b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Output dan menampilkan pesan sukses pada Halaman Daftar Output</p>
<p><b>Alternate courses:</b></p> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kode tidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Output sudah pernah direkam, sistem menampilkan pesan "Kode Output sudah ada".</li> </ul>

### A.93. Deskripsi Use Case Hapus Output

Tabel A. 93 Deskripsi Use Case Hapus Output

UC-93 –Proses Hapus Output	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Output	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Output yang dipilih dari Daftar Output dan menampilkan Halaman Daftar Output terbaru.	

**Alternate courses:**

Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Outputtanpa melakukan apapun.

**A.94. Deskripsi Use Case Tayang Profil Pengguna**

Tabel A. 94 Deskripsi Use Case Tayang Profil Pengguna

<b>UC-94</b> —Proses Tayang Profil Pengguna	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Pengaturan	
<b>Triggers:</b> Admin mengklik tombol Pengaturan, Admin mengklik tombol Profil Pengguna	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Profil Pengguna.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Profil Pengguna	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Profil Pengguna, sesuai dengan tabel t_pengguna	

**A.95. Deskripsi Use Case Rekam Profil Pengguna**

Tabel A. 95 Deskripsi Use Case Rekam Profil Pengguna

<b>UC-95</b> —Proses Rekam Profil Pengguna	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Profil Pengguna	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Profil Pengguna. Admin	

mengisi field-field yang ada lalu menekan tombol Simpan.

**Post-conditions:**

Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Profil Pengguna, dan menampilkan pesan sukses pada Halaman Rekam Profil Pengguna.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Penggunasudah pernah direkam, sistem menampilkan pesan "Kode Penggunasudah ada."
- Input khusus satker, kanwil dan KPPN diambil dari tabel t\_satker, t\_kanwil, dan t\_kppn.

### A.96. Deskripsi Use Case UbahProfil Pengguna

Tabel A. 96Deskripsi Use Case Ubah Profil Pengguna

UC-96 –Proses Ubah Profil Pengguna	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Profil Pengguna	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Profil Penggunayang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Profil Penggunadan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Profil Penggunadan menampilkan pesan sukses pada Halaman Daftar Profil Pengguna.	



**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Penggunasudah pernah direkam, sistem menampilkan pesan "Kode Penggunasudah ada".
- Input khusus satker, kanwil dan KPPN diambil dari tabel t\_satker, t\_kanwil, dan t\_kppn.

**A.97. Deskripsi Use Case HapusProfil Pengguna**

Tabel A. 97 Deskripsi Use Case Hapus Profil Pengguna

<b>UC-97 –ProsesHapus Profil Pengguna</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Profil Pengguna	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Profil Penggunayang dipilih dari Daftar Profil Penggunadan menampilkan Halaman Daftar Profil Penggunaterbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Profil Penggunatanpa melakukan apapun.	

### A.98. Deskripsi Use Case Tayang Waktu Kerja

Tabel A. 98 Deskripsi Use Case Tayang Waktu Kerja

UC-98– Proses Tayang Waktu Kerja	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Pengaturan	
<b>Triggers:</b> Admin mengklik tombol Pengaturan, Admin mengklik tombol Waktu Kerja	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Waktu Kerja.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Waktu Kerja	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Waktu Kerja, sesuai dengan tabel t_yulian	

### A.99. Deskripsi Use Case Rekam Waktu Kerja

Tabel A. 99 Deskripsi Use Case Rekam Waktu Kerja

UC-99–Proses Rekam Bendahara	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Waktu Kerja	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Waktu Kerja. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Waktu Kerja, dan menampilkan pesan	

sukses pada Halaman Rekam Waktu Kerja.

**Alternate courses:**

- Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".
- Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".
- Jika kode Waktu Kerjasudah pernah direkam, sistem menampilkan pesan "Kode Waktu Kerjasudah ada".

**A.100. Deskripsi Use Case Ubah Waktu Kerja**

Tabel A. 100 Deskripsi Use Case Ubah Waktu Kerja

<b>UC-100 –Proses Ubah Waktu Kerja</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Waktu Kerja	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Waktu Kerjayang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Waktu Kerjadan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Waktu Kerjadan menampilkan pesan sukses pada Halaman Daftar Waktu Kerja	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> </ul>	

- Jika kode Waktu Kerja sudah pernah direkam, sistem menampilkan pesan "Kode Waktu Kerja sudah ada".

### A.101. Deskripsi Use Case Hapus Waktu Kerja

Tabel A. 101 Deskripsi Use Case Hapus Waktu Kerja

<b>UC-101 –Proses Hapus Waktu Kerja</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Waktu Kerja	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin Tetap memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Bendahara yang dipilih dari Daftar Waktu Kerjadan menampilkan Halaman Daftar Waktu Kerjaterbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Waktu Kerjatanpa melakukan apapun.	

### A.102. Deskripsi Use Case Backup Data

Tabel A. 102 Deskripsi Use Case Backup Data

<b>UC-102– Proses Backup Data</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Pengaturan	
<b>Triggers:</b> Admin mengklik tombol Pengaturan, Admin mengklik tombol Backup.	

<b>Basic course:</b> Sistem menampilkan Halaman Backup Datadan memuat tombol untuk mem-backup data.
<b>Post-conditions:</b> Saat Admin mengklik tombol “Backup” database akan di- <i>dump</i>
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Direktori dump adalah C:\</li> <li>• Database yang di backup adalah keseluruhan database siapkan.</li> </ul>

### A.103. Deskripsi Use Case Tayang Konten

Tabel A. 103 Deskripsi Use Case Tayang Konten

<b>UC-103– Proses Tayang Konten</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Bantuan	
<b>Triggers:</b> Admin mengklik tombol Bantuan, Admin mengklik tombol Konten	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Konten.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Konten	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Konten, sesuai dengan text pada file konten.jsp.</li> </ul>	

### A.104. Deskripsi Use Case Tayang Peraturan

Tabel A. 104 Deskripsi Use Case Tayang Peraturan

<b>UC-104– Proses Tayang Peraturan</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal



<b>Pre-conditions:</b> Admin berada pada halaman Bantuan
<b>Triggers:</b> Admin mengklik tombol Bantuan, Admin mengklik tombol Peraturan
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Peraturan.
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Peraturan
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Sistem menampilkan daftar referensi Peraturan, sesuai dengan tabel t_peraturan</li> </ul>

### A.105. Deskripsi Use Case Rekam Peraturan

Tabel A. 105 Deskripsi Use Case Rekam Peraturan

UC-105-Proses Rekam Peraturan	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Peraturan	
<b>Triggers:</b> Admin mengklik link Rekam Data	
<b>Basic course:</b> Sistem menampilkan Halaman Rekam Peraturan. Admin mengisi field-field yang ada lalu menekan tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin . Sistem menyimpan masukan ke dalam Daftar Peraturan, dan menampilkan pesan sukses pada Halaman Rekam Peraturan.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa</li> </ul>	

- |                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>angka 0 - 9".</p> <ul style="list-style-type: none"> <li>• Jika kode Peraturansudah pernah direkam, sistem menampilkan pesan "Kode Peraturan sudah ada"."</li> </ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### A.106. Deskripsi Use Case UbahPeraturan

Tabel A. 106Deskripsi Use Case Ubah Peraturan

<b>UC-106 –Proses Ubah Peraturan</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Peraturan	
<b>Triggers:</b> Admin mengklik tombol ubah pada baris data Peraturanyang akan diubah.	
<b>Basic course:</b> Sistem menampilkan Halaman Ubah Peraturandan memuat data yang akan diubah pada field-field yang sesuai. Admin mengubah isi field kemudian mengklik tombol Simpan.	
<b>Post-conditions:</b> Sistem memvalidasi masukan dari Admin. Sistem menyimpan perubahan ke dalam Daftar Peraturandan menampilkan pesan sukses pada Halaman Daftar Peraturan	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Jika ada field yang kosong, sistem menampilkan pesan "Semua field harus diisi".</li> <li>• Jika data berawalan kodetidak berupa angka, sistem menampilkan pesan "gagal menyimpan! data harus berupa angka 0 - 9".</li> <li>• Jika kode Peraturansudah pernah direkam, sistem menampilkan pesan "Kode Peraturan sudah ada"."</li> </ul>	

### A.107. Deskripsi Use Case HapusPeraturan

Tabel A. 107Deskripsi Use Case Hapus Peraturan

<b>UC-107 –ProsesHapus Peraturan</b>
--------------------------------------

<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Daftar Peraturan	
<b>Triggers:</b> Admin mengklik link Hapus pada data yang akan dihapus.	
<b>Basic course:</b> Sistem menampilkan konfirmasi penghapusan. Admin memilih Ya.	
<b>Post-conditions:</b> Sistem menghapus data Peraturan yang dipilih dari Daftar Peraturan dan menampilkan Halaman Daftar Peraturan terbaru.	
<b>Alternate courses:</b> Jika Admin memilih 'Tidak', sistem kembali menampilkan Halaman Daftar Peraturan tanpa melakukan apapun.	

### A.108. Deskripsi Use Case Tayang Tentang

Tabel A. 108 Deskripsi Use Case Tayang Tentang

<b>UC-108– Proses Tayang Bendahara</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pengguna Goal
<b>Pre-conditions:</b> Admin berada pada halaman Referensi	
<b>Triggers:</b> Admin mengklik tombol Referensi Umum, Admin mengklik tombol Bendahara	
<b>Basic course:</b> Sistem menampilkan Halaman Daftar Bendahara.	
<b>Post-conditions:</b> Sistem menampilkan daftar referensi Bendahara	
<b>Alternate courses:</b> • Sistem menampilkan daftar referensi Bendahara, sesuai dengan teks pada file tentang.jsp	



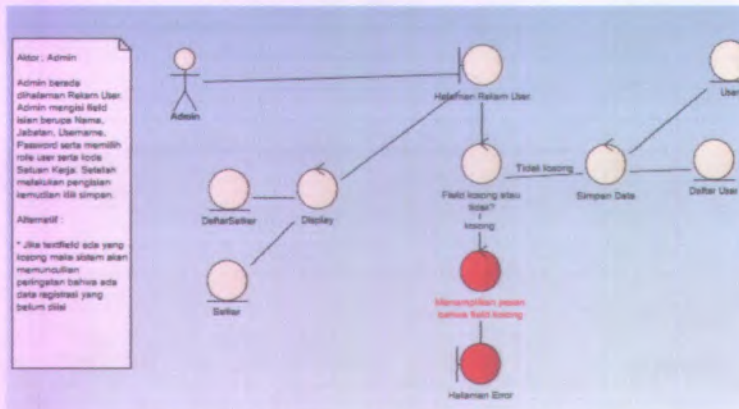
### A.109. Deskripsi Use Case Proses Logout

Tabel A. 109 Deskripsi Use Case Proses Logout

<b>UC-109– Proses Logout</b>	
<b>Primary Actor:</b> Administrator (Admin)	<b>Level:</b> Pegawai
<b>Pre-conditions:</b> Admin berada pada halaman menu apapun	
<b>Triggers:</b> Admin mengklik tombol “logout” atau link “keluar”	
<b>Basic course:</b> Pegawai dikembalikan ke halaman indeks.	
<b>Post-conditions:</b> Sistem me-remove seluruh nilai session yang terpakai saat aplikasi SIAPKAN berjalan.	
<b>Alternate courses:</b> <ul style="list-style-type: none"> <li>• Seluruh nilai session telah ter-remove, pegawai tidak dapat mengakses halaman sebelum login.</li> </ul>	

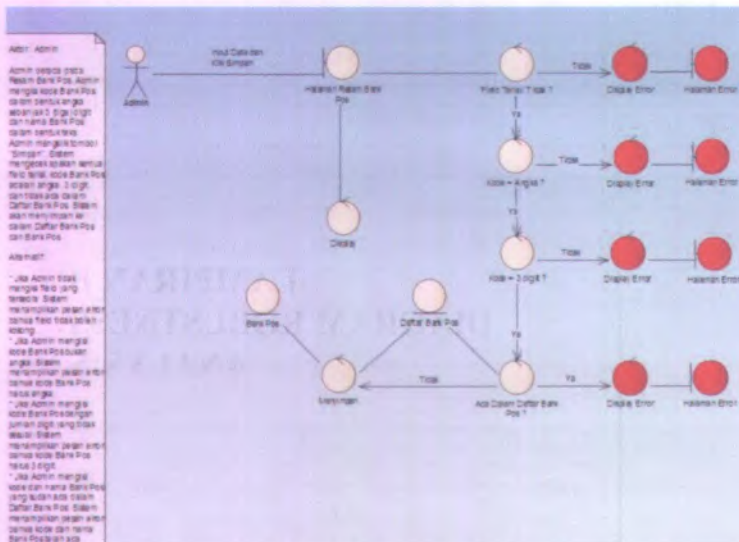
**LAMPIRAN B**  
**DIAGRAM ROBUSTNESS**  
**ANALYSIS**

### B.1. Diagram Robustness Rekam User



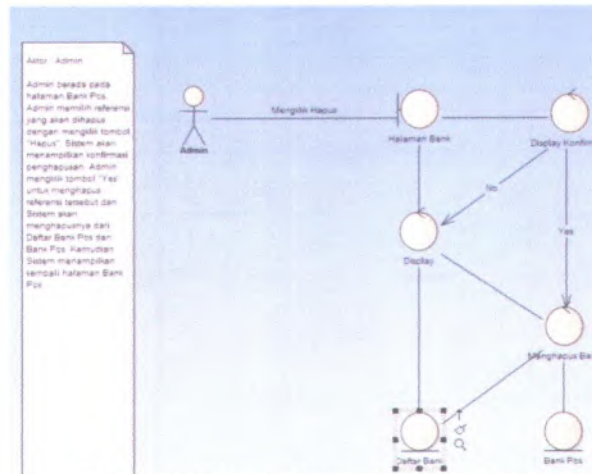
Gambar B. 1 Diagram Robustness Rekam User

### B.2. Diagram Robustness Rekam Bank



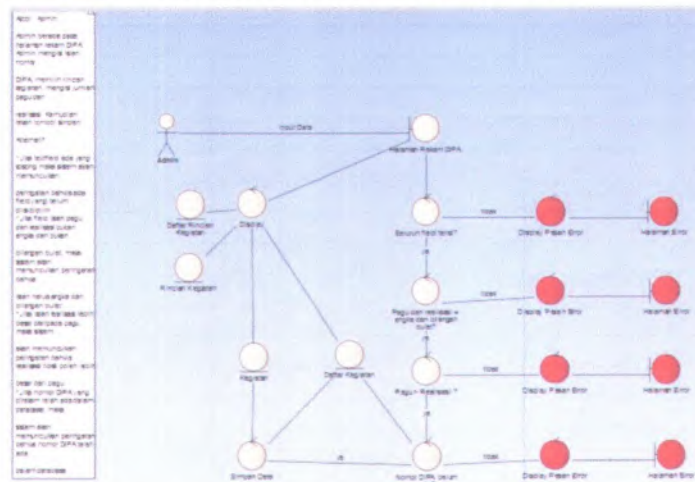
Gambar B. 2 Diagram Robustness Rekam Bank

### B.3. Diagram Robustness Hapus Bank



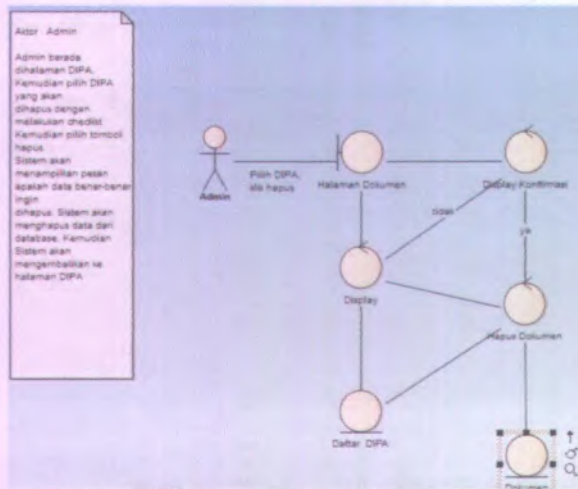
Gambar B. 3 Diagram Robustness Hapus Bank

### B.4. Diagram Robustness Rekam Dokumen



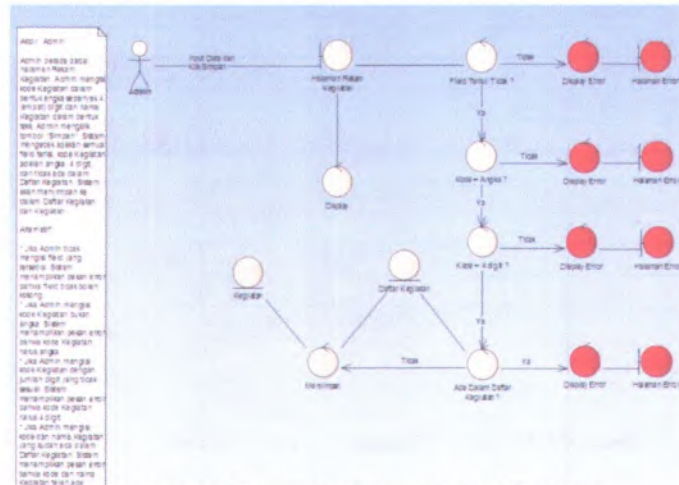
Gambar B. 4 Diagram Robustness Rekam Dokumen

## B.5. Diagram Robustness Hapus Dokumen



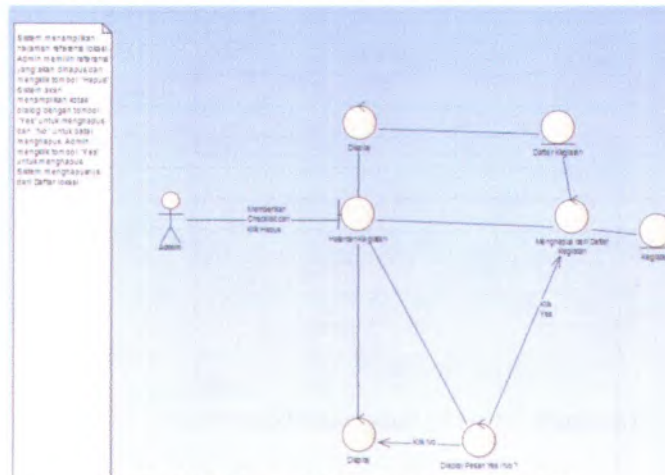
Gambar B. 5 Diagram Robustness Hapus Dokumen

### B.6. Diagram Robustness Rekam Kegiatan



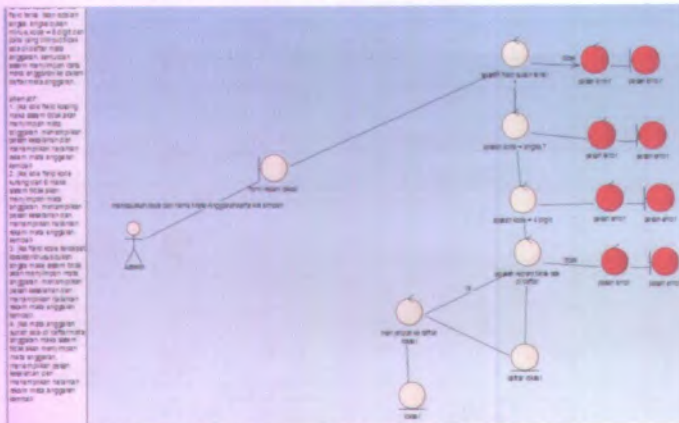
Gambar B. 6 Diagram Robustness Rekam Kegiatan

### B.7. Diagram Robustness Hapus Kegiatan



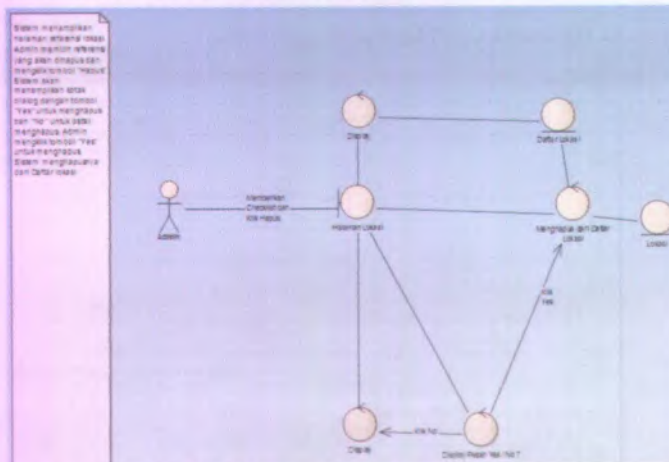
Gambar B. 7 Diagram Robustness Hapus Kegiatan

**B.8. Diagram Robustness Rekam Lokasi**



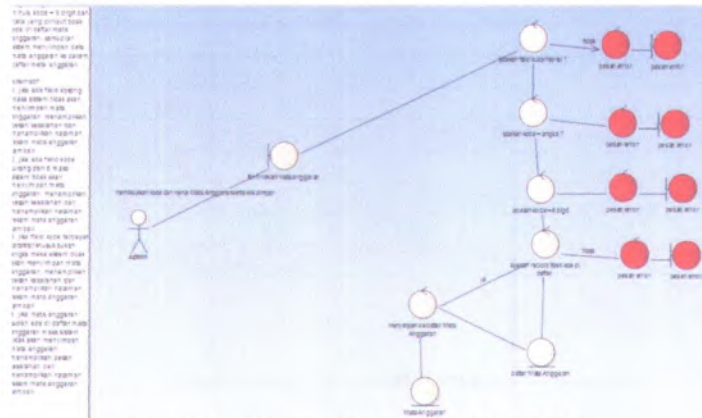
**Gambar B. 8**Diagram Robustness Rekam Lokasi

**B.9. Diagram Robustness Hapus Lokasi**



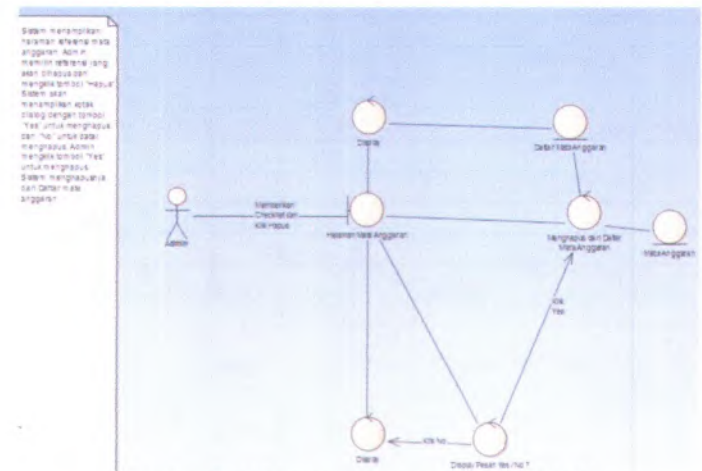
**Gambar B. 9**Diagram Robustness Hapus Lokasi

### B.10. Diagram Robustness Rekam MAK



Gambar B. 10 Diagram Robustness Rekam MAK

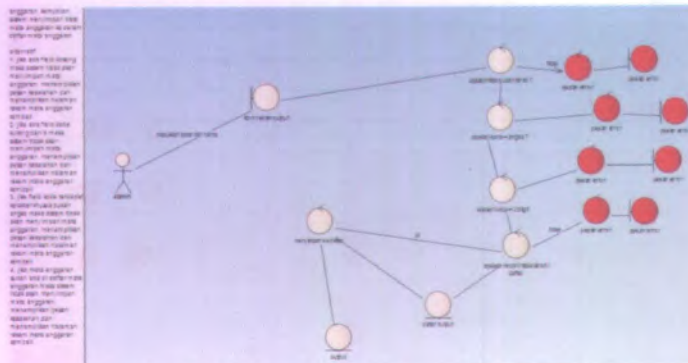
### B.11. Diagram Robustness Hapus MAK



Gambar B. 11 Diagram Robustness Hapus MAK

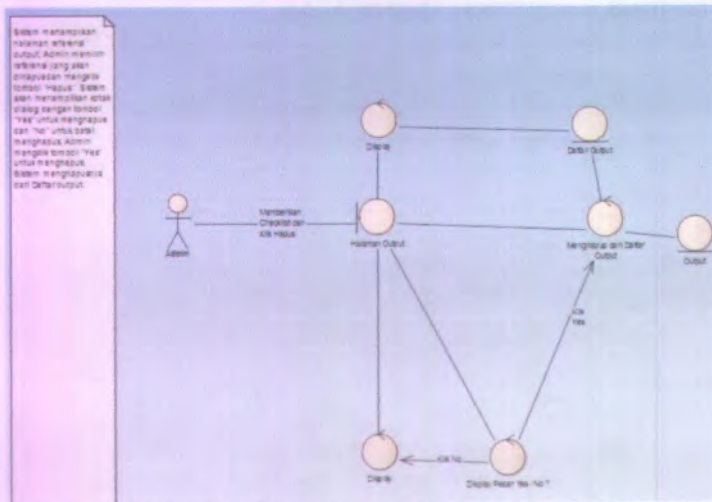


**B.12. Diagram Robustness Rekam Output**



**Gambar B. 12**Diagram Robustness Rekam Output

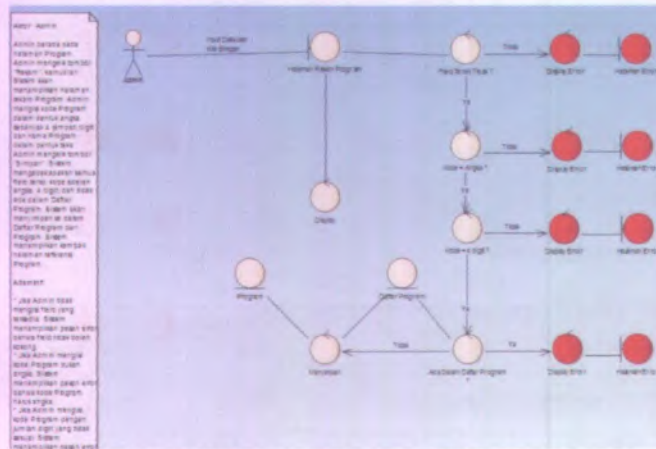
**B.13. Diagram Robustness Hapus Output**



**Gambar B. 13**Diagram Robustness Hapus Output

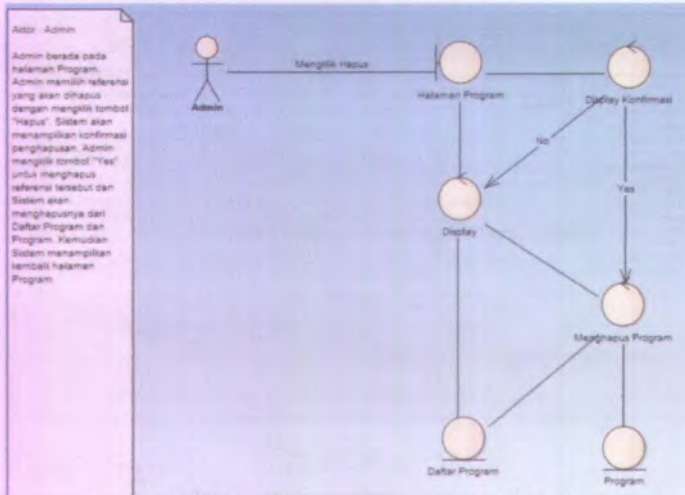


**B.16. Diagram Robustness Rekam Program**



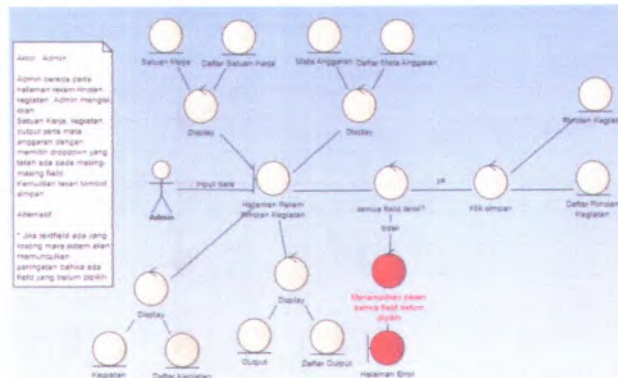
**Gambar B. 16**Diagram Robustness Rekam Program

**B.17. Diagram Robustness Hapus Program**



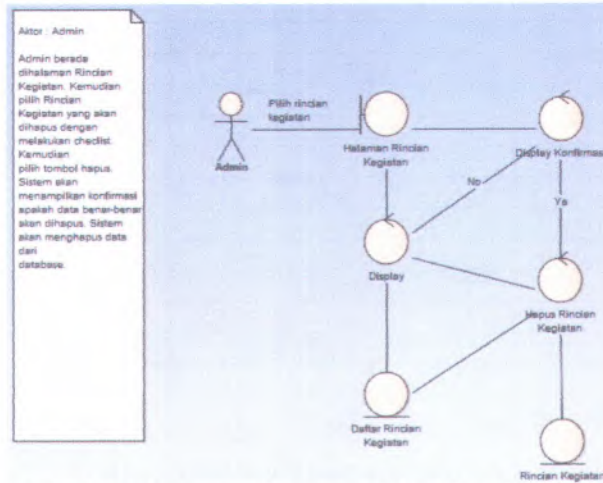
**Gambar B. 17**Diagram Robustness Hapus Program

**B.18. Diagram Robustness Rekam Jenis Kegiatan**



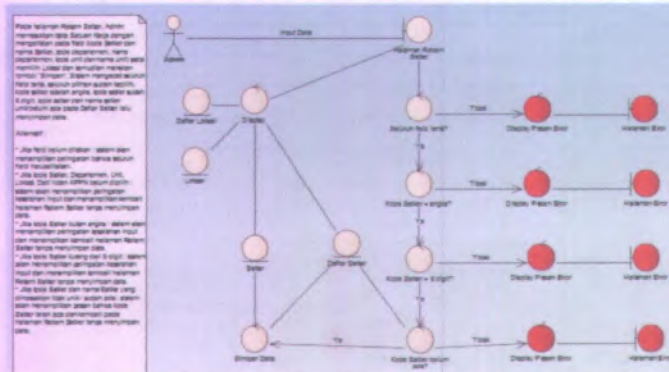
**Gambar B. 18**Diagram Robustness Rekam Jenis Kegiatan

**B.19. Diagram Robustness Hapus Jenis Kegiatan**



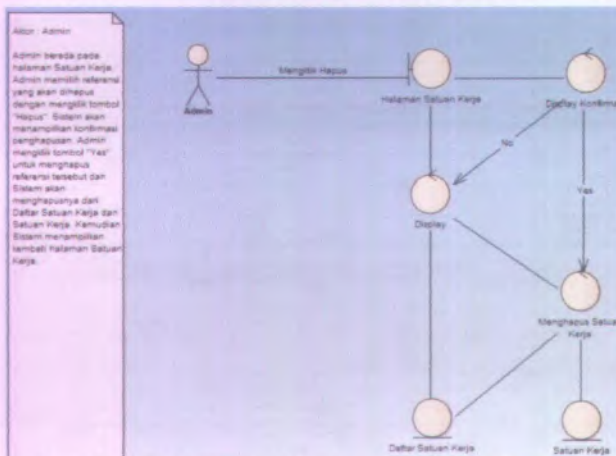
**Gambar B. 19**Diagram Robustness Hapus Jenis Kegiatan

### B.20. Diagram Robustness Rekam Satuan Kerja



Gambar B. 20 Diagram Robustness Rekam Satuan Kerja

### B.21. Diagram Robustness Hapus Satuan Kerja

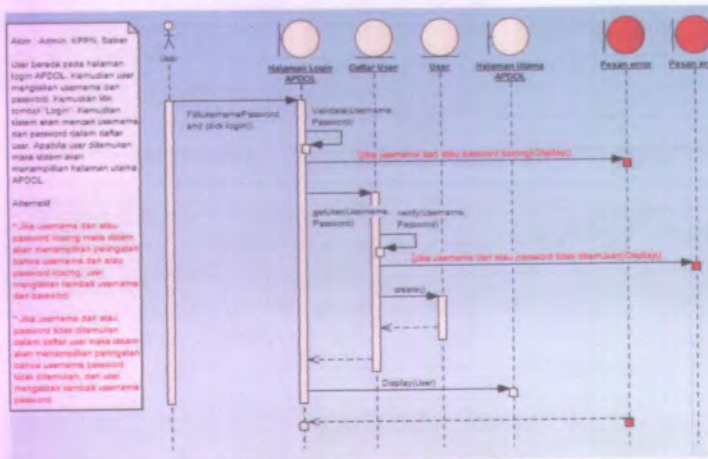


Gambar B. 21 Diagram Robustness Hapus Satuan Kerja

**LAMPIRAN C**  
**SEQUENCE DIAGRAM**

C-1

### C.1. Diagram Sequence Login

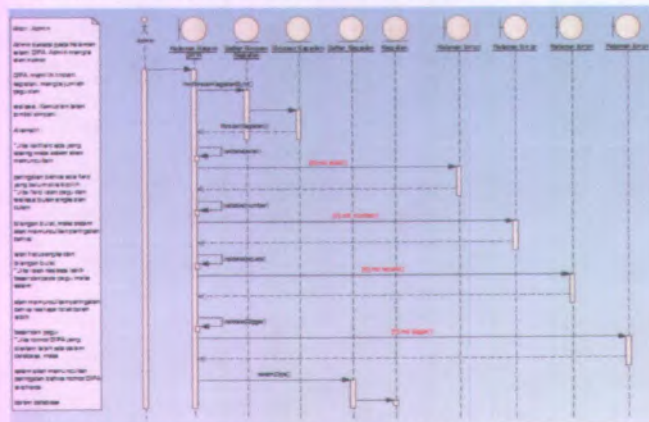


Gambar C. 1 Diagram Sequence Login



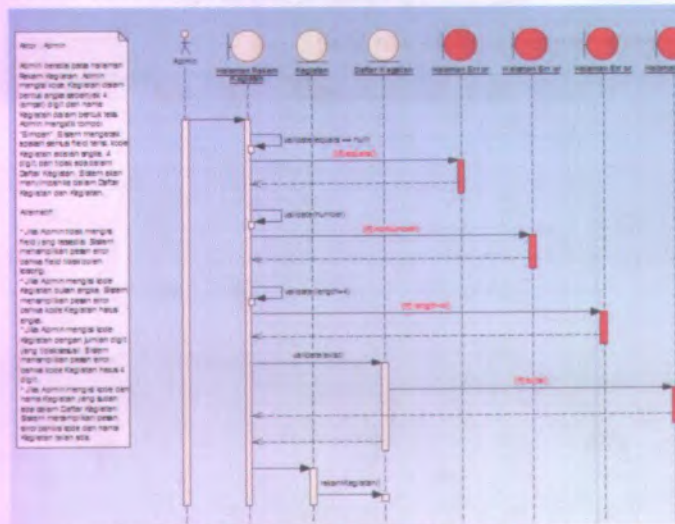


### C.4. Diagram Sequence Rekam Dokumen



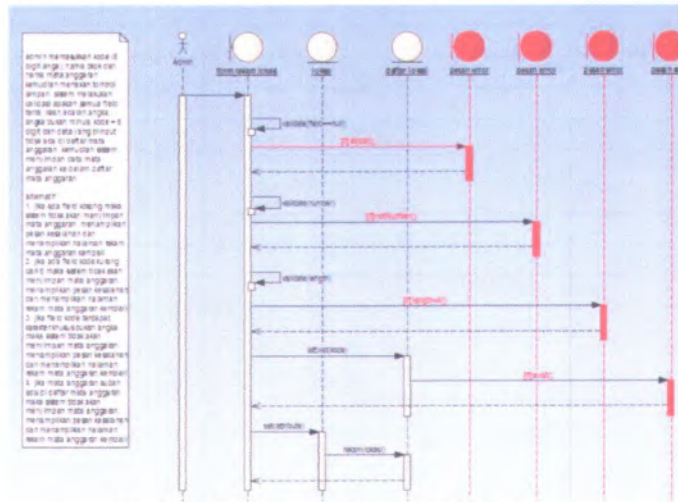
Gambar C. 4 Diagram Sequence Rekam Dokumen

### C.5. Diagram Sequence Rekam Kegiatan



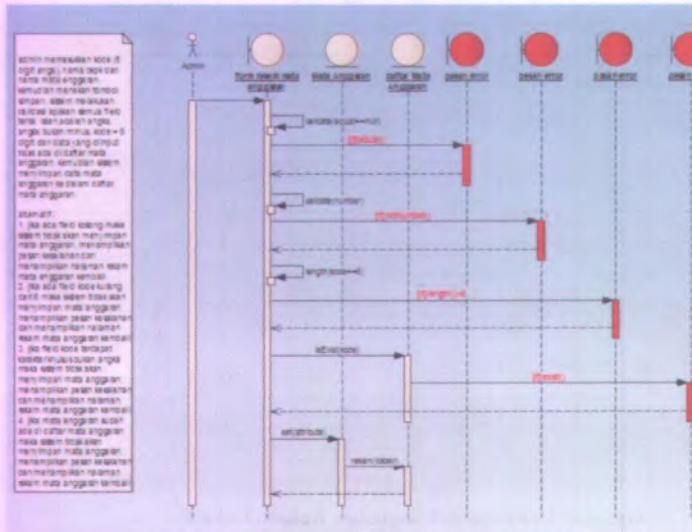
Gambar C. 5 Diagram Sequence Rekam Kegiatan

### C.6. Diagram Sequence Rekam Lokasi



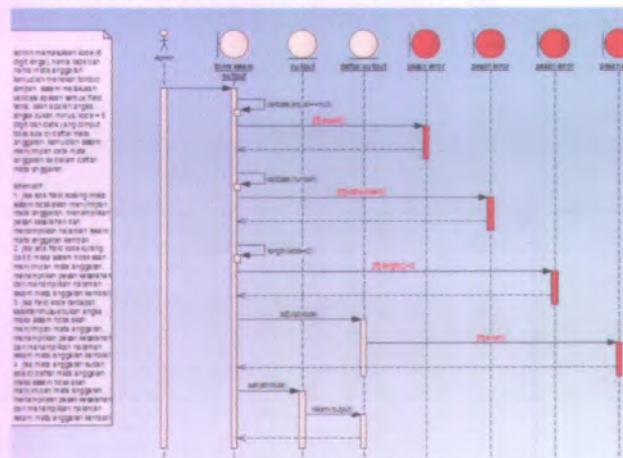
Gambar C. 6 Diagram Sequence Rekam Lokasi

### C.7. Diagram Sequence Rekam MAK



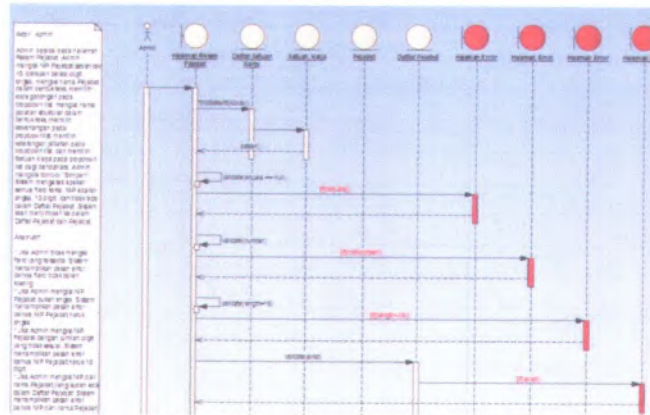
Gambar C. 7 Diagram Sequence Rekam MAK

### C.8. Diagram Sequence Rekam Output



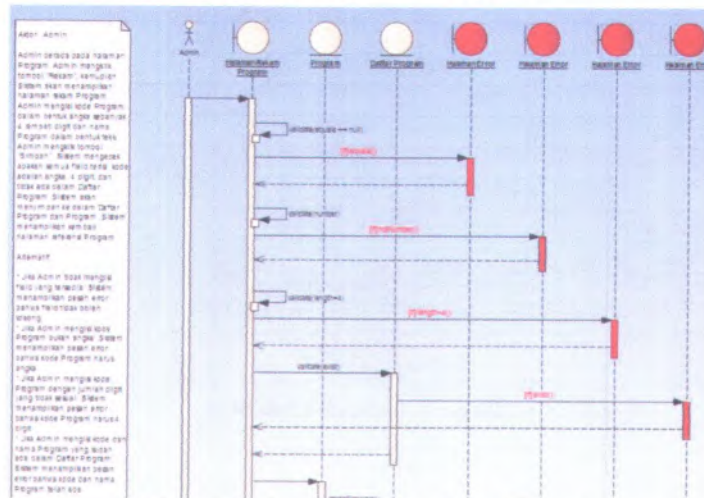
Gambar C. 8 Diagram Sequence Rekam Output

### C.9. Diagram Sequence Rekam Pejabat



Gambar C. 9 Diagram Sequence Rekam Pejabat

### C.10. Diagram Sequence Rekam Program

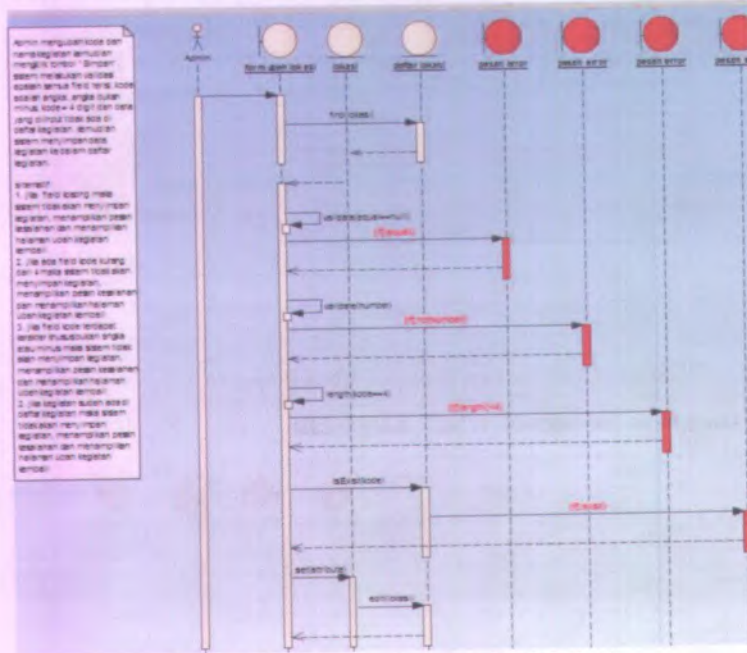


Gambar C. 10 Diagram Sequence Rekam Program



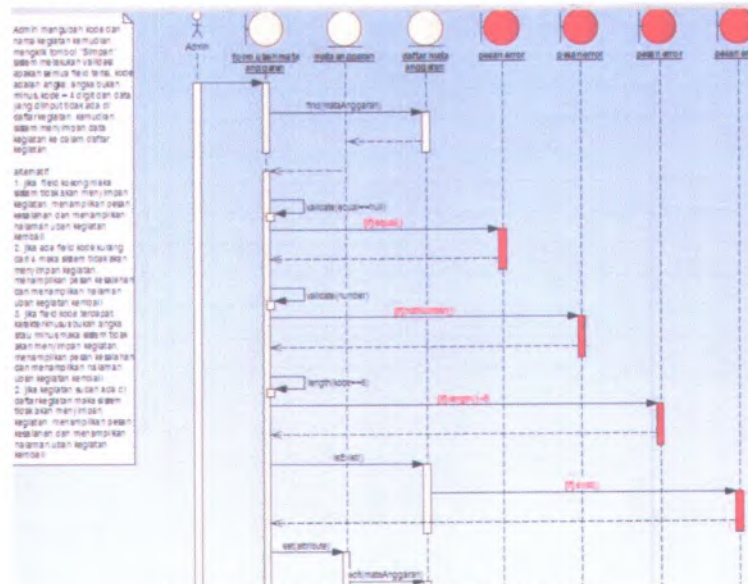


### C.15. Diagram Sequence Ubah Lokasi



Gambar C. 15 Diagram Sequence Ubah Lokasi

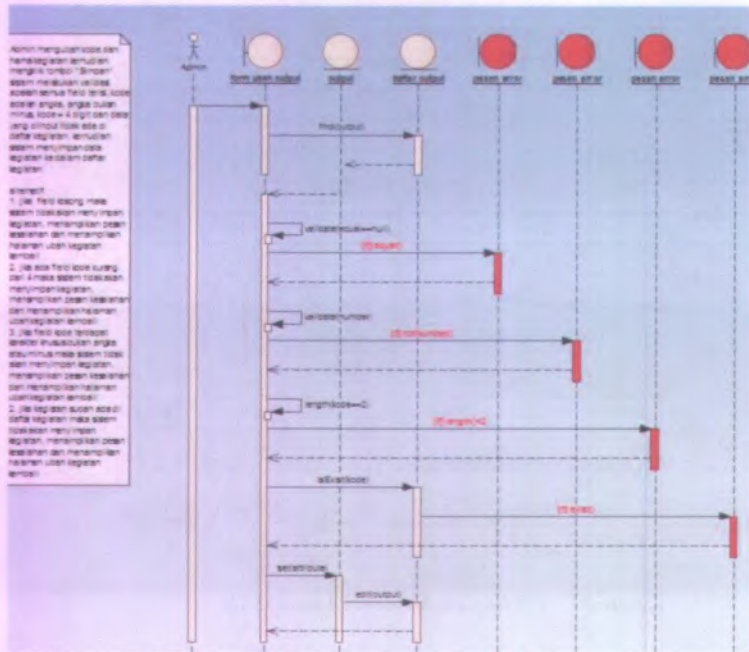
### C.16. Diagram Sequence Ubah MAK



Gambar C. 16 Diagram Sequence Ubah MAK



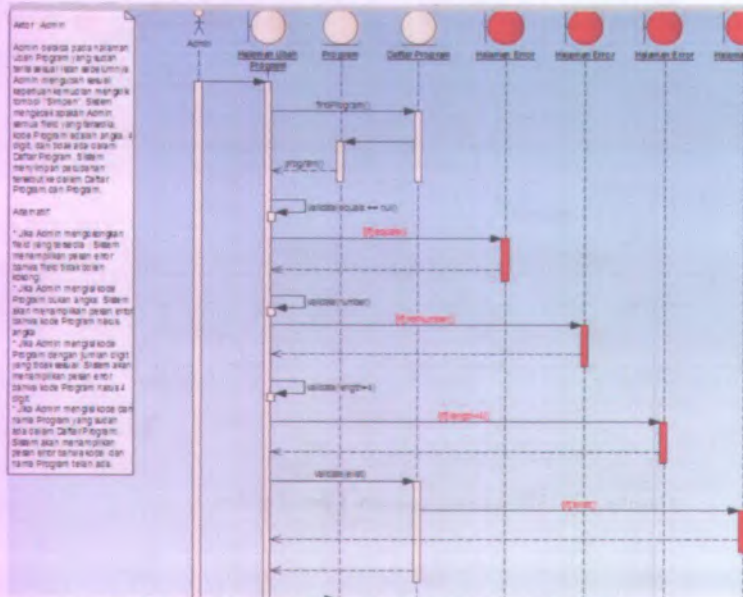
C.17. Diagram Sequence Ubah Output



Gambar C. 17 Diagram Sequence Ubah Output



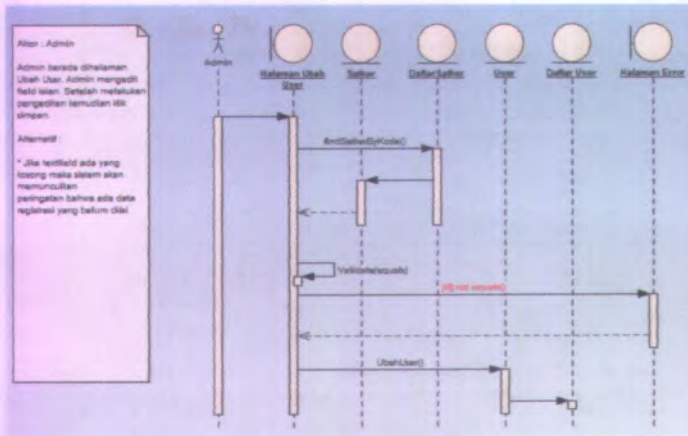
### C.19. Diagram Sequence Ubah Program



Gambar C. 19 Diagram Sequence Ubah Program

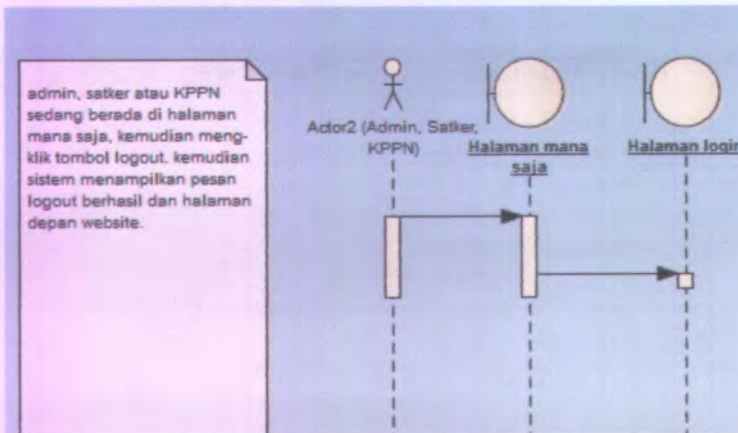


### C.21. Diagram Sequence Ubah Pengguna



Gambar C. 21 Diagram Sequence Ubah Pengguna

### C.22. Diagram Sequence Logout



Gambar C. 22 Diagram Sequence Logout

**LAMPIRAN D**  
**TEST CASE**

D-1



### D.1. Test Case Login

Tabel D. 1 Test Case Login

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Load halaman index.jsp saat run aplikasi SIAPKAN	Menekan tombol run project	Halaman index.jsp tampil, form login tampil.	sukses
TC02	Salah satu field tidak terisi	Menekan Tombol Enter	Gagal login, pesan, username atau password tidak boleh kosong.	sukses
TC03	Input username salah	Menekan Tombol Enter	Gagal login, pesan, username atau password salah.	sukses
TC04	Input password salah	Menekan Tombol Enter	Gagal login, pesan, username atau password salah.	sukses
TC05	Mengisi field	Menekan Tombol Enter	Field username tidak	sukses

	usernamenon abjad dan angka		dapat diinputkan.	
TC06	Menempelkan/mengetik url halaman modul sebelum login	Menempelkan/mengetik url di browser	Diarahkan ke halaman error.	sukses
TC07	Input Username dan Password benar	Menekan Tombol Enter	Masuk ke halaman Utama Modul Administrasi.	sukses

## D.2. Test Case Rekam Satker

Tabel D. 2 Test Case Rekam Satker

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Satker diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan 6 digit angka 0 - 9	sukses
TC02	Mengisi Field Kode Kabkota diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Mengisi Field Kode Unit diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC04	Mengisi Field Nomor	Menekan	Pesan eror, kode harus	sukses



D-4

	SP diluar angka 0 - 9	Tombol Simpan	diinput dengan angka 0 - 9	
TC05	Mengisi Field Kode Departemen diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC06	Mengisi Field Kode KPPN diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC07	Mengisi Field Kode Lokasi diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 -	sukses
TC08	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC09	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.3. Test Case Ubah Satker

Tabel D. 3 Test Case Ubah Satker

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Masuk ke halaman Ubah Satker	Menekan icon ubah	Sistem akan menampilkan halaman Ubah Satker	sukses
TC02	Kode Satker terkunci	Mengklik field	Kode Satker terkunci	sukses

		kode Satker		
TC03	Pengguna menekan tombol No pada konfirmasi penghapusan	Menekan Tombol No pada Konfirmasi		
TC04	Mengisi Field Kode Kabkota diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC05	Mengisi Field Kode Unit diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC06	Mengisi Field Nomor SP diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC07	Mengisi Field Kode Departemen diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC08	Mengisi Field Kode KPPN diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC09	Mengisi Field Kode Lokasi diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 -	sukses
TC10	Salah satu field input	Menekan	Pesan eror, pastikan semua	sukses

D-6

	kosong.	Tombol Simpan	field terisi.	
TC09	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil menyimpan data	sukses

#### D.4. Test Case Hapus Satker

Tabel D. 4 Test Case Hapus Satker

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Satker	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Satker	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Satker terbaru	Menekan "ya hapus data"	Kembali ke daftar Satker terbaru	sukses

#### D.5. Test Case Rekam KPP

Tabel D. 5 Test Case Rekam KPP

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode	Menekan	Pesan eror, kode harus	sukses

	KPP diluar angka 0 - 9	Tombol Simpan	diinput dengan 6 digit angka 0 - 9	
TC02	Mengisi Field Kode Dati 2 diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Mengisi Field Kode Kanwil diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC04	Mengisi Kode Lokasi diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC05	Mengisi Field Kode Satker diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC06	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC07	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.6. Test Case Ubah KPP

Tabel D. 6 Test Case Ubah KPP

ID	Skenario	Aksi	Hasil	Sukses/Gagal
----	----------	------	-------	--------------

D-8

TC01	Kode KPP terkunci	Mengisi Kode KPP	Field kode KPP terkunci	sukses
TC02	Mengisi Field Kode Dati 2 diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Mengisi Field Kode Kanwil diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC04	Mengisi Kode Lokasi diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC05	Mengisi Field Kode Satker diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC06	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC07	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.7. Test Case Hapus KPP

Tabel D. 7 Test Case Hapus KPP

ID	Skenario	Aksi	Hasil	Sukses/Gagal
----	----------	------	-------	--------------

TC01	Menghapus salah satu daftar KPP	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar KPP	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar KPP terbaru	Menekan "ya hapus data"	Kembali ke daftar KPP terbaru	sukses

#### D.8. Test Case RekamBank

Tabel D. 8 Test Case Rekam Bank

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Bank diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses



D-10

### D.9. Test Case Ubah Bank

Tabel D. 9 Test Case Ubah Bank

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Bank terkunci	Mengisi Kode Bank	Field kode Bank terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.10. Test Case Hapus Bank

Tabel D. 10 Test Case Hapus Bank

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Bank	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Bank	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Bank terbaru	Menekan "ya hapus data"	Kembali ke daftar Bank terbaru	sukses

### D.11. Test Case Rekam KPPN

Tabel D. 11 Test Case Rekam KPPN

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode KPPN diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode Dati 2 diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.12. Test Case Ubah KPPN

Tabel D. 12 Test Case Ubah KPPN

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode KPPN terkunci	Mengisi Kode	Field kode KPPN terkunci	sukses



D-12

		KPPN		
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.13. Test Case Hapus KPPN

Tabel D. 13 Test Case Hapus KPPN

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar KPPN	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar KPPN	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar KPPN terbaru	Menekan "ya hapus data"	Kembali ke daftar KPPN terbaru	sukses

#### D.14. Test Case Rekam Cara Bayar

Tabel D. 14 Test Case Rekam Cara Bayar

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Cara Bayar diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.15. Test Case Ubah Cara Bayar

Tabel D. 15 Test Case Ubah Cara Bayar

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Cara Bayar terkunci	Mengisi Kode Cara Bayar	Field kode Cara Bayar terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.16. Test Case Hapus Cara Bayar**

Tabel D. 16 Test Case Hapus Cara Bayar

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Cara Bayar	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Cara Bayar	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Cara Bayar terbaru	Menekan "ya hapus data"	Kembali ke daftar Cara Bayar terbaru	sukses

**D.17. Test Case Rekam Lokasi**

Tabel D. 17 Test Case Rekam Lokasi

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Lokasi diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses

	9			
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.18. Test Case Ubah Lokasi

Tabel D. 18 Test Case Ubah Lokasi

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Lokasi terkunci	Mengisi Kode Lokasi	Field kode Lokasi terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

D-16

### D.19. Test Case Hapus Lokasi

Tabel D. 19 Test Case Hapus Lokasi

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Lokasi	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Lokasi	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Lokasi terbaru	Menekan "ya hapus data"	Kembali ke daftar Lokasi terbaru	sukses

### D.20. Test Case Rekam Cara Tarik

Tabel D. 20 Test Case Rekam Cara Tarik

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Cara Tarik diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses

TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses
------	-------------------------------	-----------------------	-------------------------------------	--------

### D.21. Test Case Ubah Cara Tarik

Tabel D. 21 Test Case Ubah Cara Tarik

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Cara Tarik terkunci	Mengisi Kode Cara Tarik	Field kode Cara Tarik terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.22. Test Case Hapus Cara Tarik

Tabel D. 22 Test Case Hapus Cara Tarik

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Cara Tarik	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu	Menekan "ya	Data terhapus	sukses

	daftar Cara Tarik	hapus data”		
TC03	Kembali ke daftar Cara Tarik terbaru	Menekan “ya hapus data	Kembali ke daftar Cara Tarik `terbaru	sukses

### D.23. Test Case Rekam MAK

Tabel D. 23 Test Case Rekam MAK

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode MAK diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.24. Test Case Ubah MAK

Tabel D. 24 Test Case Ubah MAK

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode MAK terkunci	Mengisi Kode KPP	Field kode MAK terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.25. Test Case Hapus MAK

Tabel D. 25 Test Case Hapus MAK

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar MAK	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar MAK	Menekan "ya hapus data"	Data terhapus	sukses



D-20

TC03	Kembali ke daftar MAK terbaru	Menekan "ya hapus data	Kembali ke daftar MAK terbaru	sukses
------	-------------------------------	------------------------	-------------------------------	--------

### D.26. Test Case Rekam Gol&Pangkat

Tabel D. 26 Test Case Rekam Gol&Pangkat

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Gol&Pangkat diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.27. Test Case Ubah Gol&Pangkat

Tabel D. 27 Test Case Ubah Gol&Pangkat

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Gol&Pangkat terkunci	Mengisi Kode Gol&Pangkat	Field kode Gol&Pangkat terkunci	sukses

TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.28. Test Case Hapus Gol&Pangkat

Tabel D. 28 Test Case Hapus Gol&Pangkat

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Gol&Pangkat	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Gol&Pangkat	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Gol&Pangkat terbaru	Menekan "ya hapus data"	Kembali ke daftar Gol&Pangkat `terbaru	sukses

### D.29. Test Case Rekam Output

Tabel D. 29 Test Case Rekam Output

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode	Menekan	Pesan eror, kode harus	sukses

D-22

	Output diluar angka 0 - 9	Tombol Simpan	diinput dengan angka 0 - 9	
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.30. Test Case Ubah Output

Tabel D. 30 Test Case Ubah Output

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Output terkunci	Mengisi Kode Output	Field kode Output terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai	Menekan	Pesan sukses, berhasil	sukses

	aturan	Tombol Simpan	merekam data	
--	--------	---------------	--------------	--

### D.31. Test Case Hapus Output

Tabel D. 31 Test Case Hapus Output

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Output	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Output	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Output terbaru	Menekan "ya hapus data"	Kembali ke daftar Output `terbaru	sukses

### D.32. Test Case Rekam Kegiatan

Tabel D. 32 Test Case Rekam Kegiatan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Kegiatan diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode	Menekan	Pesan eror, kode harus	sukses

D-24

	lainnya diluar angka 0 - 9	Tombol Simpan	diinput dengan angka 0 - 9	
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.33. Test Case Ubah Kegiatan

Tabel D. 33 Test Case Ubah Kegiatan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Kegiatan terkunci	Mengisi Kode Kegiatan	Field kode Kegiatan terkunci	sukses
TC02	Mengisi Field Kode Dati 2 diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.34. Test Case Hapus Kegiatan

Tabel D. 34 Test Case Hapus Kegiatan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Kegiatan	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Kegiatan	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Kegiatan terbaru	Menekan "ya hapus data"	Kembali ke daftar Kegiatan `terbaru	sukses

### D.35. Test Case Rekam Pejabat

Tabel D. 35 Test Case Rekam Pejabat

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Pejabat diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode NIP diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Mengisi Field Kode lainnya diluar angka 0 -	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses

	9			
TC04	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC05	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.36. Test Case Ubah Pejabat

Tabel D. 36 Test Case Ubah Pejabat

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Pejabat terkunci	Mengisi Kode Pejabat	Field kode Pejabat terkunci	sukses
TC02	Mengisi Field NIP diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC04	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC05	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.37. Test Case Hapus Pejabat

Tabel D. 37 Test Case Hapus Pejabat

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Pejabat	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Pejabat	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Pejabat terbaru	Menekan "ya hapus data"	Kembali ke daftar Pejabat terbaru	sukses

### D.38. Test Case Rekam Kewenangan

Tabel D. 38 Test Case Rekam Kewenangan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Kewenangan diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses





**D.39. Test Case Ubah Kewenangan**

Tabel D. 39 Test Case Ubah Kewenangan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Kewenangan terkunci	Mengisi Kode Kewenangan	Field kode Kewenangan terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.40. Test Case Hapus Kewenangan**

Tabel D. 40 Test Case Hapus Kewenangan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Kewenangan	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Kewenangan	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar	Menekan "ya	Kembali ke daftar	sukses

	Kewenangan terbaru	hapus data	Kewenangan `terbaru	
--	--------------------	------------	---------------------	--

#### D.41. Test Case Rekam Program

Tabel D. 41 Test Case Rekam Program

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Program diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.42. Test Case Ubah Program

Tabel D. 42 Test Case Ubah Program

ID	Skenario	Aksi	Hasil	Sukses/Gagal
----	----------	------	-------	--------------

TC01	Kode Program terkunci	Mengisi Kode Program	Field kode Program terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.43. Test Case Hapus Program

Tabel D. 43 Test Case Hapus Program

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Program	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Program	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Program terbaru	Menekan "ya hapus data"	Kembali ke daftar Program terbaru	sukses

#### D.44. Test Case Rekam KPKNL

Tabel D. 44 Test Case Rekam KPKNL

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode KPKNL diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.45. Test Case Ubah KPKNL

Tabel D. 45 Test Case Ubah KPKNL

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode KPKNL terkunci	Mengisi Kode KPKNL	Field kode KPKNL terkunci	sukses
TC02	Mengisi Field Kode	Menekan	Pesan error, kode harus	sukses

	lainnya diluar angka 0 - 9	Tombol Simpan	diinput dengan angka 0 - 9	
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.46. Test Case Hapus KPKNL

Tabel D. 46 Test Case Hapus KPKNL

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar KPKNL	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar KPKNL	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar KPKNL terbaru	Menekan "ya hapus data"	Kembali ke daftar KPKNL `terbaru	sukses

#### D.47. Test Case Rekam Sumber Dana

Tabel D. 47 Test Case Rekam Sumber Dana

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Sumber Dana diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.48. Test Case Ubah Sumber Dana

Tabel D. 48 Test Case Ubah Sumber Dana

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Sumber Dana terkunci	Mengisi Kode Sumber Dana	Field kode Sumber Dana terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.49. Test Case Hapus Sumber Dana**

Tabel D. 49 Test Case Hapus Sumber Dana

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Sumber Dana	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Sumber Dana	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Sumber Dana terbaru	Menekan "ya hapus data"	Kembali ke daftar Sumber Dana `terbaru	sukses

**D.50. Test Case Rekam Cara Perolehan**

Tabel D. 50 Test Case Rekam Cara Perolehan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Cara Perolehan diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses

TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses
------	-------------------------------	-----------------------	-------------------------------------	--------

#### D.51. Test Case Ubah Cara Perolehan

Tabel D. 51 Test Case Ubah Cara Perolehan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Cara Perolehan terkunci	Mengisi Kode Cara Perolehan	Field kode Cara Perolehan terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.52. Test Case Hapus Cara Perolehan

Tabel D. 52 Test Case Hapus Cara Perolehan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Cara Perolehan	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu	Menekan "ya	Data terhapus	sukses



D-36

	daftar Cara Perolehan	hapus data”		
TC03	Kembali ke daftar Cara Perolehan terbaru	Menekan “ya hapus data	Kembali ke daftar Cara Perolehan `terbaru	sukses

### D.53. Test Case Rekam Aset Tetap

Tabel D. 53 Test Case Rekam Aset Tetap

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Aset Tetap diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.54. Test Case Ubah Aset Tetap

Tabel D. 54 Test Case Ubah Aset Tetap

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Aset Tetap terkunci	Mengisi Kode Aset Tetap	Field kode Aset Tetap terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.55. Test Case Hapus Aset Tetap

Tabel D. 55 Test Case Hapus Aset Tetap

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Aset Tetap	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Aset Tetap	Menekan "ya hapus data"	Data terhapus	sukses

D-38

TC03	Kembali ke daftar Aset Tetap terbaru	Menekan "ya hapus data	Kembali ke daftar Aset Tetap `terbaru	sukses
------	--------------------------------------	------------------------	---------------------------------------	--------

### D.56. Test Case Rekam SSKel Barang

Tabel D. 56 Test Case Rekam SSKel Barang

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode SSKel Barang diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan 6 digit angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.57. Test Case Ubah SSKel Barang**

Tabel D. 57 Test Case Ubah SSKel Barang

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode SSKel Barang terkunci	Mengisi Kode SSKel Barang	Field kode SSKel Barang terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.58. Test Case Hapus SSKel Barang**

Tabel D. 58 Test Case Hapus SSKel Barang

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar SSKel Barang	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar SSKel Barang	Menekan "ya hapus data"	Data terhapus	sukses



D-40

TC03	Kembali ke daftar SSKel Barang terbaru	Menekan "ya hapus data	Kembali ke daftar SSKel Barang `terbaru	sukses
------	----------------------------------------	------------------------	-----------------------------------------	--------

#### D.59. Test Case Rekam Bendahara

Tabel D. 59 Test Case Rekam Bendahara

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Bendahara diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

	Sukses/Gagal
rkunci	sukses
us a 0 - 9	sukses
semua	sukses
il	sukses

	Sukses/Gagal
	sukses
	sukses

### 1. Test Case Ubah Bendahara

Tabel D. 60 Test Case Ubah Bendahara

	Skenario	Aksi	Hasil	Sukses/Gagal
1	Kode Bendahara terkunci	Mengisi Kode Bendahara	Field kode Bendahara terkunci	sukses
2	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
3	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
4	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### 1. Test Case Hapus Bendahara

Tabel D. 61 Test Case Hapus Bendahara

	Skenario	Aksi	Hasil	Sukses/Gagal
1)	Menghapus salah satu daftar Bendahara	Menekan icon hapus	Pesan, "hapus data?"	sukses
2)	Menghapus salah satu daftar Bendahara	Menekan "ya hapus data"	Data terhapus	sukses

ungsi	sukses
-------	--------

### D.62. Test Case Rekam Fungsi

Tabel D. 62 Test Case Rekam Fungsi

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Fungsi diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

erkunci	Sukses/Ga sukses
---------	---------------------

		Beban		
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.67. Test Case Hapus Beban

Tabel D. 67 Test Case Hapus Beban

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Beban	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Beban	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Beban terbaru	Menekan "ya hapus data"	Kembali ke daftar Beban `terbaru	sukses

#### D.68. Test Case Rekam Jenis Dokumen

Tabel D. 68 Test Case Rekam Jenis Dokumen

ID	Skenario	Aksi	Hasil	Sukses/Gagal
----	----------	------	-------	--------------



TC01	Mengisi Field Kode Jenis Dokumen diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.69. Test Case Ubah Jenis Dokumen

Tabel D. 69 Test Case Ubah Jenis Dokumen

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Jenis Dokumen terkunci	Mengisi Kode Jenis Dokumen	Field kode Jenis Dokumen terkunci	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.70. Test Case Hapus Jenis Dokumen

Tabel D. 70 Test Case Hapus Jenis Dokumen

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Jenis Dokumen	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Jenis Dokumen	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Jenis Dokumen terbaru	Menekan "ya hapus data"	Kembali ke daftar Jenis Dokumen `terbaru	sukses

### D.71. Test Case Rekam MAP Barang

Tabel D. 71 Test Case Rekam MAP Barang

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode MAP Barang diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input	Menekan	Pesan error, pastikan semua	sukses

D-48

	kosong.	Tombol Simpan	field terisi.	
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.72. Test Case Ubah MAP Barang

Tabel D. 72 Test Case Ubah MAP Barang

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode MAP Barang terkunci	Mengisi Kode MAP Barang	Field kode MAP Barang terkunci	sukses
TC02	Mengisi Field Kode lainya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.73. Test Case Hapus MAP Barang

Tabel D. 73 Test Case Hapus MAP Barang

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar MAP Barang	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar MAP Barang	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar MAP Barang terbaru	Menekan "ya hapus data"	Kembali ke daftar MAP Barang `terbaru	sukses

### D.74. Test Case Rekam Kanwil

Tabel D. 74 Test Case Rekam Kanwil

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Kanwil diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input	Menekan	Pesan error, pastikan semua	sukses

D-50

	kosong.	Tombol Simpan	field terisi.	
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.75. Test Case Ubah Kanwil

Tabel D. 75 Test Case Ubah Kanwil

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Kanwil terkunci	Mengisi Kode Kanwil	Field kode Kanwil terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.76. Test Case Hapus Kanwil

Tabel D. 76 Test Case Hapus Kanwil

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Kanwil	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Kanwil	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Kanwil terbaru	Menekan "ya hapus data"	Kembali ke daftar Kanwil terbaru	sukses

### D.77. Test Case Rekam Jenis SPM

Tabel D. 77 Test Case Rekam Jenis SPM

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Jenis SPM diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input	Menekan	Pesan eror, pastikan semua	sukses

	kosong.	Tombol Simpan	field terisi.	
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.78. Test Case Ubah Jenis SPM

Tabel D. 78 Test Case Ubah Jenis SPM

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Jenis SPM terkunci	Mengisi Kode Jenis SPM	Field kode Jenis SPM terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.79. Test Case Hapus Jenis SPM

Tabel D. 79 Test Case Hapus Jenis SPM

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar SPM	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar SPM	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar SPM terbaru	Menekan "ya hapus data"	Kembali ke daftar SPM terbaru	sukses

### D.80. Test Case Rekam Jenis SPP

Tabel D. 80 Test Case Rekam Jenis SPP

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Jenis SPP diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input	Menekan	Pesan error, pastikan semua	sukses



	kosong.	Tombol Simpan	field terisi.	
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.81. Test Case Ubah Jenis SPP

Tabel D. 81 Test Case Ubah Jenis SPP

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Jenis SPP terkunci	Mengisi Kode Jenis SPP	Field kode Jenis SPP terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.82. Test Case Hapus Jenis SPP

Tabel D. 82 Test Case Hapus Jenis SPP

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar SPP	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar SPP	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar SPP terbaru	Menekan "ya hapus data"	Kembali ke daftar SPP `terbaru	sukses

### D.83. Test Case Rekam Jenis Beban

Tabel D. 83 Test Case Rekam Jenis Beban

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Jenis Beban diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.84. Test Case Ubah Jenis Beban**

Tabel D. 84 Test Case Ubah Jenis Beban

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Jenis Beban terkunci	Mengisi Kode Jenis Beban	Field kode Jenis Beban terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.85. Test Case Hapus Jenis Beban**

Tabel D. 85 Test Case Hapus Jenis Beban

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Jenis Beban	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Jenis Beban	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Jenis	Menekan "ya	Kembali ke daftar Jenis	sukses

	Beban terbaru	hapus data	Beban `terbaru	
--	---------------	------------	----------------	--

### D.86. Test Case Rekam Akun

Tabel D. 86 Test Case Rekam Akun

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Akun diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.87. Test Case Ubah Akun

Tabel D. 87 Test Case Ubah Akun

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Akun terkunci	Mengisi Kode	Field kode Akun terkunci	sukses

D-58

		Akun		
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.88. Test Case Hapus Akun

Tabel D. 88 Test Case Hapus Akun

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Akun	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Akun	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Akun terbaru	Menekan "ya hapus data"	Kembali ke daftar Akun terbaru	sukses

**D.89. Test Case Rekam Blokir**

Tabel D. 89 Test Case Rekam Blokir

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Blokir diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.90. Test Case Ubah Blokir**

Tabel D. 90 Test Case Ubah Blokir

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Blokir terkunci	Mengisi Kode Blokir	Field kode Blokir terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.91. Test Case Hapus Blokir**

Tabel D. 91 Test Case Hapus Blokir

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Blokir	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Blokir	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Blokir terbaru	Menekan "ya hapus data"	Kembali ke daftar Blokir terbaru	sukses

**D.92. Test Case Rekam Departemen**

Tabel D. 92 Test Case Rekam Departemen

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Departemen diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 -	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses

	9			
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.93. Test Case Ubah Departemen

Tabel D. 93 Test Case Ubah Departemen

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Departemen terkunci	Mengisi Kode Departemen	Field kode Departemen terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses



D-62

#### D.94. Test Case Hapus Departemen

Tabel D. 94 Test Case Hapus Departemen

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Departemen	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Departemen	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Departemen terbaru	Menekan "ya hapus data"	Kembali ke daftar Departemen `terbaru	sukses

#### D.95. Test Case Rekam GBKPK

Tabel D. 95 Test Case Rekam GBKPK

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode GBKPK diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input	Menekan	Pesan eror, pastikan semua	sukses

	kosong.	Tombol Simpan	field terisi.	
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.96. Test Case Ubah GBKPK

Tabel D. 96 Test Case Ubah GBKPK

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode GBKPK terkunci	Mengisi Kode GBKPK	Field kode GBKPK terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.97. Test Case Hapus GBKPK**

Tabel D. 97 Test Case Hapus GBKPK

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar GBKPK	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar GBKPK	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar GBKPK terbaru	Menekan "ya hapus data"	Kembali ke daftar GBKPK `terbaru	sukses

**D.98. Test Case Rekam Unit**

Tabel D. 98 Test Case Rekam Unit

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Unit diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses

TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses
------	-------------------------------	-----------------------	-------------------------------------	--------

### D.99. Test Case Ubah Unit

Tabel D. 99 Test Case Ubah Unit

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Unit terkunci	Mengisi Kode Unit	Field kode Unit terkunci	sukses
TC02	Mengisi Field Kode lainnya diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.100. Test Case Hapus Unit

Tabel D. 100 Test Case Hapus Unit

ID	Skenario	Aksi	Hasil	Sukses/Gagal
----	----------	------	-------	--------------

TC01	Menghapus salah satu daftar Unit	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Unit	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Unit terbaru	Menekan "ya hapus data"	Kembali ke daftar Unit `terbaru	sukses

#### D.101. Test Case Rekam IB

Tabel D. 101 Test Case Rekam IB

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode IB diluar angka 0 - 9	Menekan Tombol Simpan	Pesan error, kode harus diinput dengan 6 digit angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan error, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.102. Test Case Ubah IB**

Tabel D. 102 Test Case Ubah IB

<b>ID</b>	<b>Skenario</b>	<b>Aksi</b>	<b>Hasil</b>	<b>Sukses/Gagal</b>
TC01	Kode IB terkunci	Mengisi Kode IB	Field kode IB terkunci	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.103. Test Case Hapus IB**

Tabel D. 103 Test Case Hapus IB

<b>ID</b>	<b>Skenario</b>	<b>Aksi</b>	<b>Hasil</b>	<b>Sukses/Gagal</b>
TC01	Menghapus salah satu daftar IB	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar IB	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar IB terbaru	Menekan "ya hapus data"	Kembali ke daftar IB `terbaru	sukses

**D.104. Test Case Rekam Profil Pengguna**

Tabel D. 104 Test Case Rekam Profil Pengguna

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Profil Penggunadiluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Mengisi Field Kode lain diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.105. Test Case Ubah Profil Pengguna**

Tabel D. 105 Test Case Ubah Profil Pengguna

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Profil Penggunaterkunci	Mengisi Kode KPP	Field kode KPP terkunci	sukses
TC02	Mengisi Field Kode diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses

TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.106. Test Case Hapus Profil Pengguna

Tabel D. 106 Test Case Ubah Profil Pengguna

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Profil Pengguna	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Profil Pengguna	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Profil Pengguna terbaru	Menekan "ya hapus data"	Kembali ke daftar Profil Pengguna`terbaru	sukses

#### D.107. Test Case Rekam Waktu Kerja

Tabel D. 107 Test Case Rekam Waktu Kerja

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode	Menekan	Pesan eror, kode harus	sukses



D-70

	Waktu Kerja diluar angka 0 - 9	Tombol Simpan	diinput dengan angka 0 - 9	
TC06	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC07	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

#### D.108. Test Case Ubah Waktu Kerja

Tabel D. 108 Test Case Ubah Waktu Kerja

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Waktu Kerja terkunci	Mengisi Kode Waktu Kerja	Field kode Waktu Kerja terkunci	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

**D.109. Test Case Hapus Waktu Kerja**

Tabel D. 109 Test Case Hapus Waktu Kerja

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Waktu Kerja	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Waktu Kerja	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Waktu Kerja terbaru	Menekan "ya hapus data"	Kembali ke daftar Waktu Kerja `terbaru	sukses

**D.110. Test Case Backup Data**

Tabel D. 110 Test Case Backup Data

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menjalankan fungsi backup data	Mengklik tombol "Backup"	Data dari database siapkan ter backup/dumped ke direktori C:\	suspended

D-72

### D.111. Test Case Rekam Peraturan

Tabel D. 111 Test Case Rekam Peraturan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Mengisi Field Kode Peraturan diluar angka 0 - 9	Menekan Tombol Simpan	Pesan eror, kode harus diinput dengan angka 0 - 9	sukses
TC02	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC03	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.112. Test Case Ubah Peraturan

Tabel D. 112 Test Case Ubah Peraturan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Kode Peraturan terkunci	Mengisi Kode Peraturan	Field kode Peraturan terkunci	sukses
TC03	Salah satu field input kosong.	Menekan Tombol Simpan	Pesan eror, pastikan semua field terisi.	sukses
TC04	Seluruh inputan sesuai aturan	Menekan Tombol Simpan	Pesan sukses, berhasil merekam data	sukses

### D.113. Test Case Hapus Peraturan

Tabel D. 113 Test Case Hapus Peraturan

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Menghapus salah satu daftar Peraturan	Menekan icon hapus	Pesan, "hapus data?"	sukses
TC02	Menghapus salah satu daftar Peraturan	Menekan "ya hapus data"	Data terhapus	sukses
TC03	Kembali ke daftar Peraturan terbaru	Menekan "ya hapus data"	Kembali ke daftar Peraturan terbaru	sukses

### D.114. Test Case Logout

Tabel D. 114 Test Case Logout

ID	Skenario	Aksi	Hasil	Sukses/Gagal
TC01	Logout dari aplikasi	Mengklik tombol Logout atau link "keluar"	Keluar dari aplikasi	sukses

## BIODATA PENULIS



Penulis dilahirkan di Jakarta, 17 Maret 1985, merupakan anak pertama dari 3 bersaudara, memiliki ibu bernama Sri Tirahaniatun dan ayah bernama Syawal. Menempuh pendidikan formal yaitu TK Wawan Kemanggisan Jakarta, SDN05 Pagi Kemanggisan Jakarta, SMPN 101 Palmerah Jakarta. Dan SMAN 78 Kemanggisan Jakarta. Setelah lulus dari SMA pada tahun 2002, Penulis mengikuti SPMB dan USM STAN kemudian diterima di Spesialisasi Kebendaharaan

Negara Sekolah Tinggi Akuntansi Negara (STAN). Lulus dari STAN pada tahun 2005, penulis ditempatkan di Bangko, Jambi hingga 1,5 tahun, kemudian berpindah penempatan ke Pekanbaru, Riau. Sesaat kemudian penulis menikah dengan seorang wanita bernama Nidaan Khafian, S.Sos., pada tanggal 5 Januari 2008 di Jakarta.

Kegiatan penulis banyak aktif dalam operasional kantor, khususnya pelayanan pada Kantor Pelayanan Perbendaharaan Negara baik di KPPN Bangko maupun di KPPN Pekanbaru.

Memiliki nama panggilan "Momoncha" dengan hobi sebagai seorang "Otaku" yaitu sifat seseorang yang menggemari hal-hal yang berisi animasi, komik, film, musik, dan atau budaya Jepang. Pada tanggal 9 September 2009, dinyatakan lulus dari tes Beasiswa Internal Direktorat Jenderal Perbendaharaan, dan akhirnya berkuliah di kampus Jurusan Sistem Informasi angkatan 2009 dengan NRP 5209 108 718.

