

48-273 / H / 12



RSSI
005.3
Zau
p-1

2012

TUGAS AKHIR - KS09 1336

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

HARI RAKHMANTO ZAUHAR
NRP 5209 108 714

Dosen Pembimbing
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom
Siswanto, S.Sos, M.M

| PERPUSTAKAAN ITS | |
|---------------------|------------|
| Tgl. Terima | 20-07-2012 |
| Terima Dari | H |
| No Agenda Prp. | - |

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



FINAL PROJECT - KS09 1336

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

HARI RAKHMANTO ZAUHAR
NRP 5209 108 714

Supervisor

Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom
Siswanto, S.Sos, M.M

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

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

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

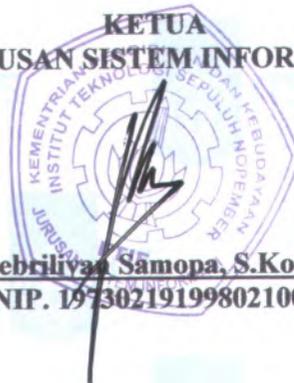
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

HARI RAKHMANTO ZAUHAR
5209 108 714

Surabaya, Juli 2012

**KETUA
JURUSAN SISTEM INFORMASI**



Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom
NIP. 197302191998021001

**PENGEMBANGAN SISTEM INFORMASI
PENGELOLAAN KEUANGAN NEGARA MODUL
PENGANGGARAN 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:

HARI RAKHMANTO ZAUHAR
5209 108 714

Disetujui Tim Penguji:

Tanggal Ujian : 16 Juli 2012
Periode Wisuda : September 2012

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

Siswanto, S.Sos., MM.


(Pembimbing II)

Mudjahiddin, S.T., M.T.


(Penguji I)

Radityo Prasetianto W., S.Kom., M.Kom.


(Penguji II)

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

Nama Mahasiswa : Hari Rakhmanto Zauhar
NRP : 5209 108 714
Jurusan : Sistem Informasi FTIf-ITS
Dosen Pembimbing : Dr. Eng. Febriliyan Samopa, S.Kom,
M.Kom
Siswanto, S.sos, M.M

Abstrak

Banyaknya fungsi yang dijalankan di satuan kerja pemerintah telah didukung oleh teknologi informasi yaitu berupa aplikasi-aplikasi yang banyak pula. Namun banyaknya aplikasi tersebut menyebabkan munculnya permasalahan baru yaitu sistem database yang masih terpisah. Saat ini akan dikembangkan suatu aplikasi yang bernama Sistem Informasi Pengelolaan Keuangan Negara yang akan mengintegrasikan semua aplikasi operasional di satuan kerja. Aplikasi ini berbasis web dan menggunakan teknologi Java dan PostgreSQL

Salah satu fungsi yang wajib ikut diintegrasikan adalah fungsi penganggaran yang saat ini menggunakan aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran. Aplikasi ini sangat penting karena merupakan awal dari proses pengelolaan keuangan. Jika aplikasi ini tidak ikut diintegrasikan maka akan berpotensi tidak sinkronnya data Daftar Isian Pelaksanaan Anggaran dari aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran dengan aplikasi Surat Perintah Membayar. Terpisahnya database dari aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran dengan aplikasi Surat Perintah

Membayar juga dapat menimbulkan potensi tindakan kriminal manipulasi data Daftar Isian Pelaksanaan Anggaran

Dengan diintegrasikannya aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran ke dalam Modul Penganggaran aplikasi Sistem Informasi Pengelolaan Keuangan Negara maka dihasilkan data Kertas Kerja Rencana Kerja Anggaran Kemeterian Lembaga secara on-line yang secara langsung dapat diakses oleh Modul Bendahara. Selain itu Modul Penganggaran juga menghasilkan data Daftar Isian Pelaksanaan Anggaran yang dibutuhkan oleh Modul Pembayaran untuk mengawasi pagu anggaran, realisasi pencairan dana dan sisa pagu anggaran. Dari hasil tersebut maka Modul Penganggaran Sistem Informasi Pengelolaan Keuangan Negara dapat menggantikan aplikasi Rencana Kerja Anggaran Kementerian Negara Lembaga dan Daftar Isian Pelaksanaan Anggaran

Kata Kunci : *Penganggaran, Aplikasi, Integrasi, Java Server Pages, PostgreSQL*

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

Name : Hari Rakhmanto Zauhar
NRP : 5209 108 714
Department : Information System FTIf-ITS
Supervisor : Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom
Siswanto, S.sos, M.M

Abstract-- Many functions that run in the satuan kerja of government has been supported by information technology that is a lot of applications as well. However, many such applications led to the emergence of new problems which database systems are still separate. Currently going developed an application called State Financial Management Information System (SIAPKAN) which will integrate all operational applications in the satuan kerja. This web-based applications uses Java Server Pages and PostgreSQL technology.

One of the functions that must be integrated is budgeting function that is currently using the application of Rencana Kerja Anggaran Kementerian Negara Lembaga Dipa. This application is very important because it is the beginning of the financial management cycle. If the application does not come then the data will not be synchronized between application of Rencana Kerja Anggaran Kementerian Negara Lembaga Dipa and application of Surat Perintah Membayar. Separation of the database from the application of these application can also pose potential criminal manipulation Dipa data.

With the integration application of Rencana Kerja Anggaran Kementerian Lembaga dan Daftar Isian Pelaksanaan Anggaran in the Budgeting Module State Financial Management Information System, the generated data of Kertas Kerja Rencana Kerja Anggaran Kementerian Lembaga can be on-line accessed by the

Treasurer module. In addition Budgeting module also generates Daftar Isian Pelaksanaan Anggaran data required by Payments Module to oversee the budget cap, actual disbursement of funds and the rest of the budget ceiling.

Keywords : Budgeting, application, integration, Java Server Pages, PostgreSQL.

KATA PENGANTAR

Alhamdulillah, segala puja dan puji syukur penulis ucapkan kepada Allah SWT yang telah memberikan islam, iman dan kesempatan sehingga penulis dapat menyelesaikan laporan tugas akhir ini yang berjudul "PENGEMBANGAN SISTEM INFORMASI PENGELOLAAN KEUANGAN NEGARA MODUL PENGANGGARAN MENGGUNAKAN TEKNOLOGI JAVA SERVER PAGES DAN POSTGRESQL." Tak lupa sholawat dan salam selalu dicurahkan kepada junjungan serta contoh terbaik Nabi Muhammad SAW yang telah menjadi figur teladan terbaik sepanjang masa yang menjadi salah satu unsur motivator bagi penulis.

Tugas akhir ini disusun demi memenuhi persyaratan kelulusan mahasiswa dan semoga dapat memberikan manfaat yang luas bagi banyak orang pada umumnya dan bagi Direktorat Jenderal Perbendaharaan pada khususnya.

Pada kesempatan yang baik ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak, Ibu yang selalu saya hormati dan sayangi, yang selalu sabar dan telaten membimbing penulis dari kecil hingga menginjak masa perkuliahan ini serta mba Herlin yang senantiasa mendukung penulis, sungguh jasa mereka tak akan pernah penulis balas selama-lamanya. Semoga Allah SWT senantiasa melindungi, menyayangi dan membimbing mereka menuju ke arah kebaikan.
2. Istri dan si kecil kakak Tazkia yang senantiasa menanti kepulangan dari kuliah dan terus menyemangati penulis dalam menyelesaikan tugas akhir ini, semoga Allah SWT senantiasa menaungi mereka dalam kebaikan-Nya. Ibu, Bleim, Blendung, Blenyaq, Blerip, dan Bledheg, serta keluarga penulis lainnya yang selalu memberikan dukungan dan kepercayaan. Terima kasih atas doa dan dukungan kalian yang tiada henti.

3. *Bapak Febriliyan Samopa, selaku dosen pembimbing I, dan Bapak Siswanto, selaku dosen pembimbing II, yang telah meluangkan waktu dan pikiran di tengah kesibukan beliau untuk membimbing dan mengarahkan penulis dalam mengerjakan tugas akhir ini.*
4. *Seluruh anggota tim tugas akhir SIAPKAN yang telah berjuang bersama, Novan, Alfie, Rizal, Yoga, Wahyudiono serta Mas ali.*
5. *Segenap keluarga Bendits yang telah berjuang bersama-sama serta saling mendukung selama masa tugas belajar ini.*
6. *Direktorat Jenderal Perbendaharaan yang telah memberikan penulis kepercayaan besar dalam menjalankan tugas belajar selama ini.*
7. *Jurusan Sistem Informasi ITS Surabaya yang selalu memberikan dukungan serta pembelajaran.*
8. *Pihak-pihak lain yang telah mendukung adanya tugas akhir ini.*

Penulis menyadari bahwa tugas akhir ini masih belum sempurna. Karena itu penulis mohon maaf atas segala kekurangan dan kekeliruan yang ada di dalam tugas akhir ini. Penulis membuka pintu selebar-lebarnya bagi pihak-pihak yang ingin memberikan kritik dan saran bagi penulis untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi seluruh pembaca.

Surabaya, Juli 2012

Penulis

DAFTAR ISI

| | | |
|---|---|-----|
| 1 | Abstrak | iii |
| | Kata Pengantar | vii |
| | Daftar Isi..... | ix |
| | Daftar Gambar..... | xv |
| | Daftar Tabel..... | xix |
| 1 | BAB I PENDAHULUAN | 1 |
| | 1.1. Latar Belakang Masalah | 1 |
| | 1.2. Perumusan Masalah | 4 |
| | 1.3. Batasan Masalah | 4 |
| | 1.4. Tujuan | 5 |
| | 1.5. Relevansi atau Manfaat..... | 5 |
| | 1.6. Sistematika Penulisan | 5 |
| 2 | BAB II TINJAUAN PUSTAKA | 7 |
| | 2.1. Penganggaran Tingkat Satuan Kerja..... | 7 |
| | 2.2. Servlets..... | 8 |
| | 2.3. JavaServer Pages..... | 10 |
| | 2.4. Model View and Controller (MVC) | 11 |
| | 2.5. PostgreSQL..... | 12 |
| | 2.6. Java Database Connectivity (JDBC)..... | 13 |
| 3 | BAB III METODOLOGI | 15 |
| | 3.1. Studi Literatur | 16 |
| | 3.2. Identifikasi Kebutuhan..... | 16 |
| | 3.3. Desain Aplikasi..... | 16 |
| | 3.4. Pemrograman aplikasi..... | 16 |
| | 3.4. Integrasi Modul..... | 17 |
| | 3.5. Uji Coba Aplikasi | 17 |

| | |
|--|-----------|
| 3.6. Migrasi Data | 17 |
| 3.7. Penyusunan Buku Tugas Akhir | 17 |
| 4 BAB IV ANALISIS KEBUTUHAN DAN DESAIN | |
| APLIKASI..... | 19 |
| 4.1. Analisis Kebutuhan Aplikasi | 19 |
| 4.1.1. Dokumen Dasar Hukum | 20 |
| 4.1.2. Pengguna | 20 |
| 4.1.3. Fitur aplikasi..... | 21 |
| 4.1.4. Workflow Aplikasi | 22 |
| 4.1.5. Penyusunan Rencana Belanja..... | 22 |
| 4.1.6. Penyusunan Rencana pendapatan..... | 23 |
| 4.1.7. Penyampaian data Kertas Kerja RKA-K/L | 23 |
| 4.1.8. Pencetakan Konsep Dipa..... | 23 |
| 4.1.9. Transfer data DIPA | 24 |
| 4.1.10. Penyusunan data rencana penarikan dana | 24 |
| 4.2. Arsitektur Aplikasi..... | 26 |
| 4.3. Hubungan Antar Modul | 28 |
| 4.4. Domain Model | 33 |
| 4.5. Class Diagram..... | 35 |
| 4.6. Use Case Diagram..... | 35 |
| 4.7. Robustness Diagram | 36 |
| 4.8. Sequence Diagram | 36 |
| 4.9. Skenario Test Case..... | 36 |
| 4.10. GUI Story Board | 36 |
| 5 BAB V IMPLEMENTASI DAN UJI COBA SISTEM..... | 37 |
| 5.1. Lingkungan Implementasi | 37 |
| 5.2. Struktur Direktori Aplikasi | 38 |
| 5.3. Konfigurasi Aplikasi Siapkan | 40 |
| 5.4. Penulisan Kode Program..... | 41 |

| | |
|--|----|
| 5.5. Pembuatan Tampilan | 45 |
| 5.6. Integrasi Aplikasi | 46 |
| 5.7. Uji coba aplikasi | 50 |
| 6. BAB VI KESIMPULAN DAN SARAN | 52 |
| 6.1 Kesimpulan | 52 |
| 6.2 Saran | 52 |
| 7. DAFTAR PUSTAKA | 54 |
| 8. LAMPIRAN A CLASS MODEL | 1 |
| A.1. Class Mode | 2 |
| 9. Lampiran B DIAGRAM DAN DESKRIPSI USE CASE | 1 |
| B.1. Diagram Use Case | 2 |
| B.2. Deskripsi Use Case Rekam Detil | 2 |
| B.3. Deskripsi Use Case Rekam Afp | 3 |
| B.4. Deskripsi Use Case Rekam Komponen | 4 |
| B.5. Deskripsi Use Case Rekam Output | 5 |
| B.6. Deskripsi Use Case Rekam Sub Komponen | 6 |
| B.7. Deskripsi Use Case Rekam Sub Output | 6 |
| B.8. Deskripsi Use Case Ubah Afp | 7 |
| B.9. Deskripsi Use Case Approve Afp | 8 |
| B.10. Deskripsi Use Case Hapus Akun | 9 |
| B.11. Deskripsi Hapus Detil | 9 |
| B.12. Deskripsi Use Case Hapus Komponen | 10 |
| B.13. Deskripsi Use Case Hapus Output | 10 |
| B.14. Deskripsi Use Case Hapus Pendapatan | 11 |
| B.15. Deskripsi Use Case Hapus Sub Komponen | 12 |
| B.16. Deskripsi Use Case Hapus Sub Output | 12 |
| B.17. Deskripsi Use Case Rekam Akun | 13 |
| B.18. Deskripsi Use Case Rekam Pendapatan | 14 |
| B.19. Deskripsi Use Case Tayang DIPA | 14 |
| B.20. Deskripsi Use Case Ubah Akun | 15 |
| B.21. Deskripsi Use Case Ubah Komponen | 15 |
| B.22. Deskripsi Use Case Ubah Output | 16 |
| B.23. Deskripsi Use Case Ubah Pendapatan | 17 |

| | |
|--|----|
| B.24. Deskripsi Use Case Ubah Sub Komponen | 18 |
| B.25. Deskripsi Use Case Ubah Sub Output | 18 |
| B.26. Deskripsi Use Case Ubah Detil | 19 |
| B.27. Deskripsi Use Case Cetak Dipa | 20 |
| 10. Lampiran C ROBUSTNESS ANALYSIS..... | 1 |
| C.1. Rekam Detil Robustness | 2 |
| C.2. Rekam Afp Robustness..... | 2 |
| C.3. Rekam Komponen Robustness | 2 |
| C.4. Rekam Output Robustness..... | 3 |
| C.5. Rekam Sub Komponen Robustness | 4 |
| C.6. Rekam Sub Output Robustness..... | 4 |
| C.7. Ubah Afp Robustness | 5 |
| C.8. Approval Afp Robustness..... | 5 |
| C.9. Hapus Akun Robustness | 5 |
| C.10. Hapus Detil Robustness..... | 6 |
| C.11. Hapus Komponen Robustness | 6 |
| C.12. Hapus Output Robustness..... | 7 |
| C.13. Hapus Pendapatan Robustness | 8 |
| C.14. Hapus Sub Komponen Robustness | 8 |
| C.15. Hapus Sub Output Robustness..... | 8 |
| C.16. Rekam Akun Robustness | 9 |
| C.17. Rekam Pendapatan Robustness | 9 |
| C.18. Tayang Dipa Robustness | 10 |
| C.19. Ubah Akun Robustness..... | 10 |
| C.20. Ubah Komponen Robustness..... | 11 |
| C.21. Ubah Output Robustness | 12 |
| C.22. Ubah Pendapatan Robustness | 12 |
| C.23. Ubah Sub Komponen Robustness | 13 |
| C.24. Ubah Sub Output Robustness | 13 |
| C.25. Ubah Detil Robustness | 14 |
| C.26. Cetak Robustness..... | 14 |
| 11. Lampiran D SEQUENCE DIAGRAM..... | 1 |
| D.1. Rekam Detil Sequence..... | 2 |
| D.2. Rekam Afp Sequence..... | 2 |

| | |
|---|----|
| D.3. Rekam Komponen Sequence | 3 |
| D.4. Rekam Output Sequence | 3 |
| D.5. Rekam Sub Komponen Sequence | 4 |
| D.6. Rekam Sub Output Sequence | 4 |
| D.7. Ubah Afp Sequence | 5 |
| D.8. Approval Afp Sequence | 5 |
| D.9. Hapus Akun Sequence | 6 |
| D.10. Hapus Detil Sequence | 7 |
| D.11. Hapus Komponen Sequence | 8 |
| D.12. Hapus Output Sequence | 9 |
| D.13. Hapus Pendapatan Sequence | 10 |
| D.14. Hapus Sub Komponen Sequence | 11 |
| D.15. Hapus Sub Output Sequence | 12 |
| D.16. Rekam Akun Sequence | 13 |
| D.17. Rekam Pendapatan Sequence | 13 |
| D.18. Tayang Dipa Sequence | 14 |
| D.19. Ubah Akun Sequence | 14 |
| D.20. Ubah Komponen Sequence | 15 |
| D.21. Ubah Output Sequence | 16 |
| D.22. Ubah Pendapatan Sequence | 17 |
| D.23. Ubah Sub Komponen Sequence | 18 |
| D.24. Ubah Sub Output Sequence | 19 |
| D.25. Ubah Detil Sequence | 20 |
| D.26. Cetak Dipa Sequence | 21 |
| 12 | 22 |
| 13. <i>Lampiran E TEST CASE</i> | 1 |
| E.1. Test Case 1: Rekam / Ubah output | 2 |
| E.2. Test Case 2: Rekam / Ubah Sub Output | 3 |
| E.3. Test Case 3: Rekam / Ubah Komponen | 5 |
| E.4. Test Case 4: Rekam / Ubah Sub Komponen | 6 |
| E.5. Test Case 5: Rekam / Ubah akun | 8 |
| E.6. Test Case 6: Rekam / Ubah detil | 9 |
| E.7. Test Case 7: Rekam / Ubah AFP | 10 |
| E.8. Test Case 8: Hapus Output | 12 |



| | |
|---|----|
| E.9. Test Case 9: Hapus Sub Output | 13 |
| E.10. Test Case 10: hapus Komponen..... | 14 |
| E.11. Test Case 11: Hapus Sub Komponen..... | 15 |
| E.12. Test Case 12: Hapus Akun..... | 16 |
| E.13. Test Case 13: Hapus Detil..... | 17 |
| E.14. Test Case 14: Approval AFP | 18 |
| E.15. Test Case 15: Tayang Dipa..... | 19 |
| 14. Lampiran F GUI STORYBOARD | 1 |
| F.1. Pengesahan Afp GUI | 2 |
| F.2. Kertas Kerja GUI..... | 2 |
| F.3. Tayang KPJM GUI | 3 |
| F.4. DIPA GUI..... | 3 |
| F.5. Rekam Ubah Sub Output GUI | 4 |
| F.6. Rekam Ubah Komponen GUI..... | 4 |
| F.7. Rekam Ubah Output GUI | 5 |
| F.8. Rekam Ubah Sub Komponen GUI | 5 |
| F.9. Rekam Ubah Akun GUI..... | 6 |
| F.10. Rekam Ubah Afp GUI | 6 |
| F.11. Rekam Ubah Detil GUI..... | 7 |
| 15. Biodata Penulis..... | 7 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Alur penganggaran tingkat satker | 8 |
| Gambar 3.1 Tahap pengerjaan tugas akhir | 15 |
| Gambar 4.1 ICONIX Process | 19 |
| Gambar 4.2 Struktur Anggaran | 21 |
| Gambar 4.3 Workflow Aplikasi RKA-K/L | 25 |
| Gambar 4.4 Workflow Sistem Penganggaran Terintegrasi | 26 |
| Gambar 4.5 Arsitektur Aplikasi | 27 |
| Gambar 4.6 Desain Aplikasi SIAPKAN | 29 |
| Gambar 4.7 Interaksi Antar Modul | 32 |
| Gambar 4.8 Domain Model | 33 |
| Gambar 5.1 Sub Direktori Nbproject | 38 |
| Gambar 5.2 sub direktori src | 39 |
| Gambar 5.3 sub direktori web | 39 |
| Gambar 5.4 context.xml | 40 |
| Gambar 5.5 informasi koneksi | 40 |
| Gambar 5.6 informasi entitas | 41 |
| Gambar 5.7 alamat url dan kelas servletnya | 41 |
| Gambar 5.8 kelas entitas output | 42 |
| Gambar 5.9 kelas model DaftarOutput | 43 |
| Gambar 5.10 kelas PokServlet | 44 |
| Gambar 5.11 kelas OutputController | 45 |
| Gambar 5.12 tampilan list output | 45 |
| Gambar 5.13 Halaman Rekam Output | 46 |
| Gambar 5.14 Halaman Rekam Akun | 47 |
| Gambar 5.15 Halaman Rekam Pendapatan | 47 |
| Gambar 5.16 Data POK (Pedoman Operasional Kegiatan) | 48 |
| Gambar 5.17 Integrasi Dengan Modul Bendahara | 49 |
| Gambar 5.18 Data Dipa | 50 |
| Gambar 0.1 Class Diagram | 2 |
| Gambar 9.1 Diagram Use Case | 2 |
| Gambar 10.1 Rekam Detil Robustness | 2 |
| Gambar 10.2 Rekam Afp Robustness | 2 |
| Gambar 10.3 Rekam Komponen Robustness | 3 |

| | |
|--|----|
| <i>Gambar 10.4</i> Rekam Output Robustness..... | 3 |
| <i>Gambar 10.5</i> Rekam Sub Komponen Robustness..... | 4 |
| <i>Gambar 10.6</i> Rekam Sub Output Robustness | 4 |
| <i>Gambar 10.7</i> Ubah Afp Robustness | 5 |
| <i>Gambar 10.8</i> Approval Afp Robustness..... | 5 |
| <i>Gambar 10.9</i> Hapus Akun Robustness | 6 |
| <i>Gambar 10.10</i> Hapus Detil Robustness..... | 6 |
| <i>Gambar 10.11</i> Hapus Komponen Robustness | 7 |
| <i>Gambar 10.12</i> Hapus Output Robustness..... | 7 |
| <i>Gambar 10.13</i> Hapus Pendapatan Robustness | 8 |
| <i>Gambar 10.14</i> Hapus Sub Komponen Robustness..... | 8 |
| <i>Gambar 10.15</i> Hapus Sub Output Robustness | 9 |
| <i>Gambar 10.16</i> Rekam Akun Robustness..... | 9 |
| <i>Gambar 10.17</i> Rekam Pendapatan Robustness | 10 |
| <i>Gambar 10.18</i> Tayang Dipa Robustness | 10 |
| <i>Gambar 10.19</i> Ubah Akun Robustness | 11 |
| <i>Gambar 10.20</i> Ubah Komponen Robustness..... | 11 |
| <i>Gambar 10.21</i> Ubah Output Robustness | 12 |
| <i>Gambar 10.22</i> Ubah Pendapatan Robustness | 12 |
| <i>Gambar 10.23</i> Ubah Sub Komponen Robustness | 13 |
| <i>Gambar 10.24</i> Ubah Sub Output Robustness | 13 |
| <i>Gambar 10.25</i> Ubah Detil Robustness | 14 |
| <i>Gambar 10.26</i> Cetak Dipa Robustness..... | 14 |
| <i>Gambar 11.1</i> Rekam Detil Sequence..... | 2 |
| <i>Gambar 11.2</i> Rekam Afp Sequence | 2 |
| <i>Gambar 11.3</i> Rekam Komponen Sequence..... | 3 |
| <i>Gambar 11.4</i> Rekam Output Sequence | 3 |
| <i>Gambar 11.5</i> Rekam Sub Komponen Sequence..... | 4 |
| <i>Gambar 11.6</i> Rekam Sub Output Sequence | 4 |
| <i>Gambar 11.7</i> Ubah Afp Sequence..... | 5 |
| <i>Gambar 11.8</i> Approval Afp Sequence..... | 5 |
| <i>Gambar 11.9</i> Hapus Akun Sequence..... | 6 |
| <i>Gambar 11.10</i> Hapus Detil Sequence..... | 7 |
| <i>Gambar 11.11</i> Hapus Komponen Sequence..... | 8 |
| <i>Gambar 11.12</i> Hapus Output Sequence..... | 9 |

| | |
|---|----|
| <i>Gambar 11.13 Hapus Pendapatan Sequence</i> | 10 |
| <i>Gambar 11.14 Hapus Sub Komponen Sequence</i> | 11 |
| <i>Gambar 11.15 Hapus Sub Output Sequence</i> | 12 |
| <i>Gambar 11.16 Rekam Akun Sequence</i> | 13 |
| <i>Gambar 11.17 Rekam Pendapatan Sequence</i> | 13 |
| <i>Gambar 11.18 Tayang Dipa Sequence</i> | 14 |
| <i>Gambar 11.19 Ubah Akun Sequence</i> | 14 |
| <i>Gambar 11.20 Ubah Komponen Sequence</i> | 15 |
| <i>Gambar 11.21 Ubah Output Sequence</i> | 16 |
| <i>Gambar 11.22 Ubah Pendapatan Sequence</i> | 17 |
| <i>Gambar 11.23 Ubah Sub Komponen Sequence</i> | 18 |
| <i>Gambar 11.24 Ubah Sub Output Sequence</i> | 19 |
| <i>Gambar 11.25 Ubah Detil Sequence</i> | 20 |
| <i>Gambar 11.26 Cetak Dipa Sequence</i> | 21 |
| <i>Gambar 14.1 Pengesahan Afp GUI</i> | 2 |
| <i>Gambar 14.2 Kertas Kerja GUI</i> | 2 |
| <i>Gambar 14.3 Tayang KPJM GUI</i> | 3 |
| <i>Gambar 14.4 DIPA GUI</i> | 3 |
| <i>Gambar 14.5 Rekam Ubah Sub Output GUI</i> | 4 |
| <i>Gambar 14.6 Rekam Ubah Komponen GUI</i> | 4 |
| <i>Gambar 14.7 Rekam Ubah Output GUI</i> | 5 |
| <i>Gambar 14.8 Rekam Ubah Sub Komponen GUI</i> | 5 |
| <i>Gambar 14.9 Rekam Ubah Akun GUI</i> | 6 |
| <i>Gambar 14.10 Rekam Ubah Afp GUI</i> | 6 |
| <i>Gambar 14.11 Rekam Ubah Detil GUI</i> | 7 |
| <i>Gambar 14.12 POK GU</i> | 7 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 5.1 Tool Yang Digunakan | 37 |
| Tabel 5.2 Tabel Use Case..... | 51 |
| Tabel 9.1 Deskripsi Use Case Rekam Detil | 2 |
| Tabel 9.2 Deskripsi Use Case Rekam Afp | 3 |
| Tabel 9.3 Deskripsi Use Case Rekam Komponen..... | 4 |
| Tabel 9.4 Deskripsi Use Case Rekam Output | 5 |
| Tabel 9.5 Deskripsi Use Case Rekam Sub Komponen | 6 |
| Tabel 9.6 Deskripsi Use Case Rekam Sub Output | 6 |
| Tabel 9.7 Deskripsi Use Case Ubah Afp..... | 7 |
| Tabel 9.8 Deskripsi Use Case Approve Afp | 8 |
| Tabel 9.9 Deskripsi Use Case Hapus Akun..... | 9 |
| Tabel 9.10 Deskripsi Hapus Detil | 9 |
| Tabel 9.11 Deskripsi Use Case Hapus Komponen..... | 10 |
| Tabel 9.12 Deskripsi Use Case Hapus Output | 10 |
| Tabel 9.13 Deskripsi Use Case Hapus Pendapatan | 11 |
| Tabel 9.14 Deskripsi Use Case Hapus Sub Komponen | 12 |
| Tabel 9.15 Deskripsi Use Case Hapus Sub Output | 12 |
| Tabel 9.16 Deskripsi Use Case Rekam Akun | 13 |
| Tabel 9.17 Deskripsi Use Case Rekam Pendapatan..... | 14 |
| Tabel 9.18 Deskripsi Use Case Tayang Dipa..... | 14 |
| Tabel 9.19 Deskripsi Use Case Ubah Akun | 15 |
| Tabel 9.20 Deskripsi Use Case Ubah Komponen | 15 |
| Tabel 9.21 Deskripsi Use Case Ubah Output..... | 16 |
| Tabel 9.22 Deskripsi Use Case Ubah Pendapatan..... | 17 |
| Tabel 9.23 Deskripsi Use Case Ubah Sub Komponen | 18 |
| Tabel 9.24 Deskripsi Use Case Ubah Sub Output..... | 18 |
| Tabel 9.25 Deskripsi Use Case Ubah Detil | 19 |
| Tabel 9.26 Deskripsi Use Case Cetak Dipa..... | 20 |

BAB I

PENDAHULUAN

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

1.1. Latar Belakang Masalah

Satuan kerja (satker) adalah instansi pemerintah yang merupakan bagian dari struktur kementerian di pemerintahan. Setiap satu satker mempunyai kepala satker (kepala dinas, kepala kantor dsb.), bendahara satker, pejabat pembuat komitmen, pejabat penanda tangan Surat Perintah Membayar (SPM), pembuat daftar gaji (PDG), dan pegawai yang bekerja di instansi tersebut (Republik Indonesia, 2004). Beberapa contoh dari satker adalah Dinas Perhubungan Kota Surabaya, Kantor Kejaksaan Tinggi Surabaya, Kantor Departemen Agama Kota Surabaya dan lain-lain .

Satker mendapatkan dana dari Anggaran Pendapatan dan Belanja Negara (APBN) untuk melaksanakan kegiatannya. Satker sebagai organisasi pemerintah menjalankan fungsi akuntansi dan pengelolaan keuangan atas kegiatan yang dilakukannya. Dalam menjalankan fungsinya, satker telah menerapkan teknologi informasi untuk meningkatkan kinerja dan kualitas laporan keuangannya. Fungsi tersebut antara lain pembukuan pengeluaran dana dari APBN yang ditunjang dengan aplikasi SPM, pembukuan bendahara memakai aplikasi Sistem Informasi Keuangan (SISKA), Penganggaran satker yang dibantu oleh aplikasi Rencana Kerja Anggaran Kementerian/Lembaga dan Daftar Isian Pelaksanaan Anggaran (RKA-K/L DIPA), akuntansi kantor satker yang menggunakan Sistem Akuntansi Kuasa Pengguna Anggaran (SAKPA) dan penatausahaan barang-barang inventarisasi satker dengan memakai aplikasi Persediaan dan Sistem Informasi Manajemen Akuntansi Barang Milik Negara

(SIMAK BMN). Semua aplikasi tersebut dibangun secara terpisah-pisah berdasarkan urgensi kebutuhan satker.

Banyaknya aplikasi pendukung yang digunakan pada mulanya cukup bermanfaat dan menghemat banyak waktu. Namun hal ini menimbulkan masalah di kemudian hari karena *database* yang digunakan terpisah di masing-masing aplikasi. Sinkronisasi data antar aplikasi dibutuhkan karena masing-masing aplikasi saling berhubungan erat, misalnya aplikasi SPM berhubungan dengan SIMAK BMN untuk mencocokkan data pengeluaran dengan jumlah barang yang benar-benar telah dibeli. Aplikasi DIPA harus sesuai jumlah pagu dananya dengan aplikasi SPM dan masih banyak lagi hubungan antar aplikasi tersebut.

Jika *database* masih terpisah maka akan ada potensi perbedaan data antar aplikasi, misalnya aplikasi RKA-K/L DIPA dengan aplikasi SPM. Data pagu anggaran pada aplikasi RKA-K/L DIPA harus sama dengan aplikasi SPM. Jika ada pengurangan pagu anggaran pada aplikasi RKA-K/L DIPA sedangkan data DIPA tidak ditransfer ke aplikasi SPM maka satker mempunyai informasi yang salah tentang pagu anggarannya

Terpisahnya *database* juga bisa berpotensi memicu tindakan kriminal manipulasi informasi anggaran. Seseorang bisa saja menambah data pagu anggaran di aplikasi SPM untuk dicairkan dananya tanpa mengubah data pagu anggaran di aplikasi RKA-K/L DIPA.

Untuk mengatasi masalah di atas, saat ini akan dibangun satu aplikasi yang bernama Sistem Informasi Keuangan Negara (SIAPKAN) yang akan mengintegrasikan semua fungsi satker. Aplikasi ini akan menggunakan sistem *single database* sehingga data yang disajikan akan selalu relevan jika terjadi perubahan data pada satu sisi fungsi. Aplikasi SIAPKAN terdiri dari 7 modul:

1. Modul Penganggaran
2. Modul Pembayaran
3. Modul Bendahara

4. Modul Persediaan
5. Modul Aset Tetap
6. Modul Administrasi
7. Modul Pelaporan

Untuk modul Penganggaran memerlukan input berupa dokumen Alokasi Pagu Anggaran per Satker. Dokumen input ini akan digunakan oleh aplikasi untuk menghasilkan dokumen data Kertas Kerja RKA-K/L. Input yang kedua adalah Surat Penetapan RKA-K/L yang didapat dari Direktorat Jenderal Anggaran (DJA) Kementerian Keuangan (Kemenkeu) dimana input ini akan menjadi konsep DIPA. Input yang ketiga adalah Surat Pengesahan DIPA (SP DIPA) yang didapat dari Kantor Wilayah Direktorat Jenderal Perbendaharaan (Kanwil DJPB) Kemenkeu dimana dokumen ini akan menjadi dasar bahwa DIPA telah diterbitkan dan bisa mulai dicairkan dananya. Data dari DIPA yang telah mendapatkan pengesahan akan menjadi input bagi modul Pembayaran.

Fungsi perencanaan penganggaran dimana satker menggunakan aplikasi RKA-K/L DIPA memegang peranan sangat penting dalam pengelolaan keuangan karena merupakan titik awal proses penyusunan APBN. Aplikasi RKA-K/L DIPA digunakan satker untuk menyusun Kertas Kerja RKA-K/L. Dari Kertas Kerja ini akan dibahas di tingkat Kementerian/Lembaga dengan Dewan Perwakilan Rakyat (DPR) untuk mendapat persetujuan. Setelah disetujui oleh DPR, data Kertas Kerja RKA-K/L yang ada di aplikasi akan menjadi data DIPA yang berfungsi sebagai dasar pengeluaran dana APBN. Dari data DIPA tersebut satker akan dapat mencairkan dana dengan menggunakan aplikasi SPM.

Selama ini, sebelum satker bisa mulai menggunakan aplikasi SPM, satker harus melakukan transfer data DIPA terlebih dahulu dari aplikasi RKA-K/L DIPA ke aplikasi SPM dengan menggunakan media *disk* yang berupa *flash disk*. Hal ini yang menjadi alasan utama kenapa aplikasi RKA-K/L DIPA perlu untuk diintegrasikan ke aplikasi SIAPKAN. Data pagu anggaran

harus selalu sama antara aplikasi RKA-K/L DIPA dengan aplikasi SPM.

Dengan terintegrasinya aplikasi RKA-K/L DIPA ke dalam aplikasi SIAPKAN, maka satker hanya membutuhkan satu aplikasi saja untuk mengakses baik fungsi penganggaran dan pelaksanaan anggaran (pencairan dana APBN). Hal ini akan memudahkan satker jika saja ada perubahan parameter referensi, maka tidak perlu melakukan perubahan di dua aplikasi, tapi hanya akan merubah di aplikasi SIAPKAN saja. Aplikasi SIAPKAN akan dibangun menggunakan bahasa pemrograman Java yaitu Servlets dan JavaServer Pages.

1.2. Perumusan Masalah

Perumusan masalah dari pemaparan latar belakang di atas adalah sebagai berikut :

1. Bagaimana membangun Aplikasi SIAPKAN Modul Penganggaran yang sesuai dengan kebutuhan satker (aplikasi RKA-K/L DIPA)
2. Bagaimana mengintegrasikan Modul Penganggaran dengan modul-modul lainnya dalam aplikasi SIAPKAN

1.3. Batasan Masalah

Dari uraian permasalahan di atas, perlu dicantumkan ruang lingkup penulisan tugas akhir :

1. Satker adalah instansi pemerintah pusat, bukan instansi pemerintah daerah
2. Proses penganggaran tingkat satker bukan di kantor pusat
3. Fitur yang diambil dari aplikasi RKA-K/L mencakup :
 - a. Penyusunan data belanja Kertas Kerja RKA-K/L
 - b. Penyusunan data Rencana Penarikan Dana
 - c. Penyusunan data rencana pendapatan
 - d. Monitoring DIPA

1.4. Tujuan

Membuat Modul Penganggaran yang merupakan bagian dari aplikasi integrasi SIAPKAN untuk menjadi aplikasi tunggal pengelolaan keuangan dan akuntansi di satker.

1.5. Relevansi atau Manfaat

Manfaat yang bisa diambil tugas akhir ini antara lain :

1. Mempercepat proses bisnis baik operasional maupun pelaporan di satker
2. Meningkatkan akurasi data laporan
3. Meningkatkan akuntabilitas laporan pertanggungjawaban

1.6. Sistematika Penulisan

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

1. Bab I Pendahuluan
Bab ini menjelaskan latar belakang, rumusan masalah, batasan masalah/ruang lingkup, tujuan, manfaat dan sistematika penulisan.
2. Bab II Tinjauan Pustaka
Bab ini menjelaskan teori-teori dari berbagai pustaka yang digunakan sebagai acuan dalam penulisan buku tugas akhir ini.
3. Bab III Metodologi
Bab ini membahas metode dan langkah yang digunakan dalam pengerjaan tugas akhir.
4. 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.
5. Bab V Implementasi Dan Uji Coba Sistem
Bab ini menjelaskan pembangunan aplikasi yang sesuai dengan desain. Selain itu, dijelaskan pula uji coba sistem dalam menjaga performa aplikasi.
6. 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



2.1. Penganggaran Tingkat Satuan Kerja

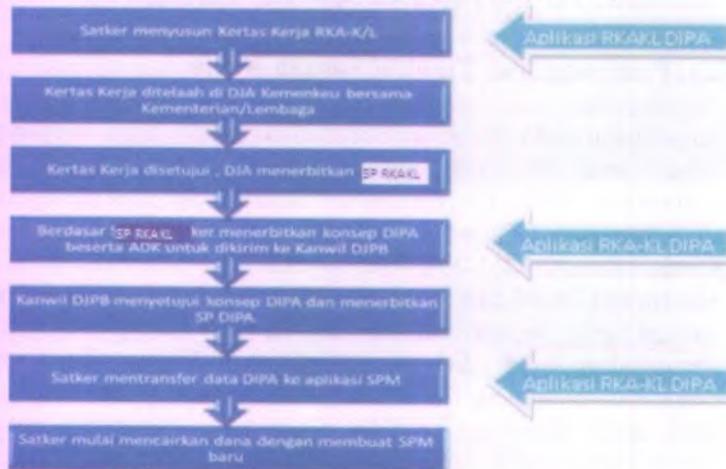
Penganggaran adalah proses merencanakan penerimaan dan pengeluaran suatu instansi pemerintah untuk satu tahun anggaran. Proses awal dari penganggaran tingkat satker dimulai dari penyusunan Kertas Kerja RKA-K/L berdasarkan pagu dana yang telah ditetapkan Kementerian/Lembaga dengan DPR sebelumnya. Kertas Kerja RKA-K/L berisi rencana pengeluaran dan penerimaan dana dari dan ke APBN berdasarkan klasifikasi tertentu dan terinci sampai ke jumlah barang/jasa yang dibelanjakan atau diterima. Pembuatan Kertas Kerja ini menggunakan aplikasi RKA-K/L DIPA, hard copy dan soft copy nanti akan dikirimkan ke DJA Kementerian Keuangan untuk diproses lebih lanjut.

Pada tingkat Kementerian/Lembaga, DJA bersama Kementerian/Lembaga melakukan proses penelaahan terhadap Kertas Kerja RKA-K/L dengan mempertimbangkan banyak aspek baik administratif dan substantif. Jika Kertas Kerja telah disetujui maka DJA akan menerbitkan SP RKA-K/L yang akan menjadi dasar penerbitan konsep DIPA (Peraturan Menteri Keuangan Republik Indonesia, 2012).

Dengan berdasar SP RKA-K/L, satker menerbitkan konsep DIPA beserta Arsip Data Komputer (ADK) di dalam flash disk untuk dikirimkan ke Kanwil DJPB. Penerbitan konsep DIPA bersama ADK dilakukan di aplikasi RKA-K/L DIPA (Peraturan Menteri Keuangan Republik Indonesia, 2012).

Setelah DIPA disetujui, Kanwil DJPB menerbitkan Surat Pengesahan DIPA (SP DIPA) pertanda bahwa DIPA dapat digunakan sebagai dasar pencairan dana APBN. Kemudian Satker mentransfer data DIPA dari aplikasi RKA-K/L DIPA ke aplikasi SPM. Setelah terisi, maka satker dapat membuat SPM atas belanja tertentu dan mencairkan dana dari APBN (Peraturan Menteri Keuangan Republik Indonesia, 2012).

Berikut disajikan alur proses bisnis penganggaran sampai dengan proses pencairan dana siap dijalankan (lihat gambar 1) :



Gambar 2.1 Alur penganggaran tingkat satker

2.2. 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 (Hall, 2000). Pada saat meneruskan request, servlets mampu melakukan beberapa fungsi antara lain :

1. Membaca informasi yang dimasukkan oleh user. 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 (Wijono, 2007).

Servlets mempunyai banyak kelebihan dibandingkan dengan teknologi aplikasi web sebelumnya CGI. Keunggulan ini antara lain (Hall, 2000):

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, ketikan 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.

2.3. JavaServer Pages

Teknologi JavaServer Pages (JSP) adalah suatu halaman yang mampu menggabungkan halaman web yang regular, statik dan dinamis yang dihasilkan dari Servlets (Hall, 2000). 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 user, bagaimana suatu aplikasi ditampilkan kepada user.

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.

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 (Wijono, 2007). 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 Sevlets akan dipakai jika sebagian besar isi dari halaman merupakan output dari program Java atau output bersifar dinamik.

2.4. Model View and Controller (MVC)

MVC adalah suatu pola penulisan kode pemrograman yang memisahkan logika bisnis dari sisi presentasi (Basham, Sierra, & Bates, 2008). Sisi logika bisnis di sini disebut Model, dimana kode Model tidak memuat sama sekali bagaimana menampilkan presentasi ke user. Model merupakan pemrograman Java yang konvensional (berisi *method*, objek dan lain-lain). Sedangkan *View* adalah sisi kode presentasi, berisi bahasa pemrograman berbasis HTML sehingga terfokus pada bagaimana menampilkan

sesuatu ke user. *View* yang dimaksud di sini adalah JSP itu sendiri. Sedangkan *Controller* adalah kode Java yang merupakan penghubung antara *View* dengan *Model*. *Controller* meneruskan perintah-perintah yang diinputkan user di *View*, kemudian menggunakan *method* yang ada di *Model* menghasilkan output yang diteruskan kembali ke *View*.

Keuntungan pola pemrograman MVC adalah logika bisnis yang terdapat pada *Model* dapat digunakan berulang-ulang meski memakai *Servlet* yang berbeda. Konsep ini merupakan pencerminan sifat Java yang *Object Oriented*. Keuntungan lain adalah masing-masing sisi *Model* dan *View* dapat dikonsentrasikan sesuai fungsi masing-masing. Pada sisi *Model*, *programmer* harus bisa menuangkan logika bisnis dalam bentuk kode, sedangkan dalam sisi *View*, *programmer* harus bisa menciptakan layer presentasi yang komunikatif dan informatif.

2.5. PostgreSQL

PostgreSQL adalah sistem basis data relasional yang dikembangkan dari sistem basis data Postgress. PostgreSQL memiliki lisensi BSD klasik yaitu bebas diambil, diubah, dipakai dan didistribusikan dengan tanpa biaya. PostgreSQL dikembangkan pertama kali oleh Andrew Yu dan Jolly Chen (Momjian, 2001).

Secara umum kapasitas PostgreSQL tidak terbatas artinya hanya dibatasi oleh kemampuan memori komputer. Untuk batas maksimal tabel adalah maksimum 32 Terabyte, batas maksimal baris adalah 1,5 Terabyte, batas maksimal satu field adalah 1 Gigabyte. Jumlah maksimal kolom dalam PostgreSQL adalah 250 sampai dengan 1600 kolom tergantung dari tipe data dari kolom. Sedangkan jumlah baris dan indeks dalam satu tabel tidak dibatasi.

Salah satu keunggulan PostgreSQL adalah berupa RDBMS sehingga mampu menjaga integritas referensial dari *foreign-key*. PostgreSQL sering disebut sebagai alternatif bagi sistem database Oracle yang bersifat komersial (Blum, 2007).

2.6. Java Database Connectivity (JDBC)

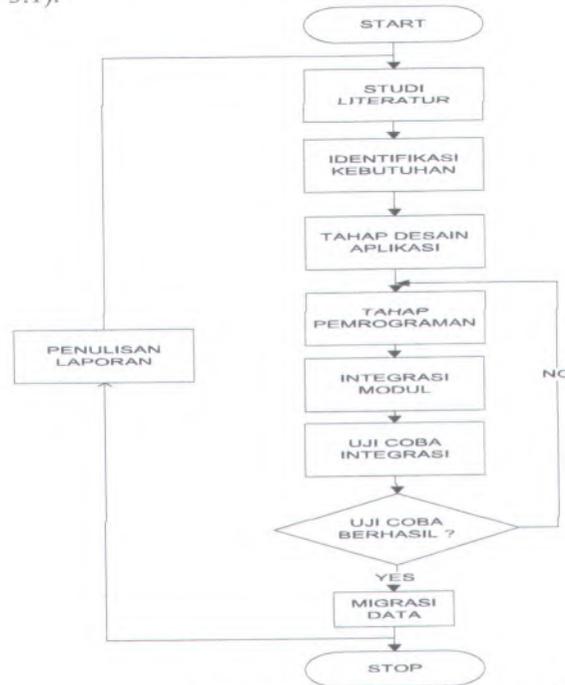
JDBC adalah suatu kelas-kelas Java yang memungkinkan suatu aplikasi dapat mengirimkan pernyataan Structured Query Language (SQL) ke dalam suatu Database Management System (DBMS). SQL adalah bahasa pemrograman khusus untuk pengelolaan data-data dalam bentuk tabel. JDBC mempunyai keunggulan yaitu sifatnya yang interoperabilitas. Maksudnya adalah pengembang aplikasi dapat membuat aplikasi tanpa menargetkan ke DBMS tertentu. Pengembang dapat menggunakan JDBC untuk menargetkan ke DBMS yang ingin dituju (Reese, 2000).

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.

BAB III METODOLOGI

Bagian ini akan menjelaskan langkah-langkah yang akan dilakukan beserta metode yang dipakai. Penulis pertama kali akan melakukan studi literatur, mengidentifikasi kebutuhan, mendesain aplikasi, memrogram aplikasi, melakukan ujicoba aplikasi serta penyusunan buku tugas akhir.

Untuk mempermudah memahami tahap pengerjaan tugas akhir ini, berikut disertakan gambar (flowchart) tahap-tahap yang akan dikerjakan penulis selama pengerjaan tugas akhir (lihat gambar 3.1).



Gambar 3.1 Tahap pengerjaan tugas akhir

3.1. Studi Literatur

Tahap ini dilakukan penulis untuk mendapatkan pengetahuan lebih lanjut bagaimana membuat aplikasi web menggunakan teknologi Servlets, JavaServer Pages dan database PostgreSQL. Sumber-sumber yang dimaksud antara lain buku-buku yang membahas tentang bagaimana mendesain dan membuat aplikasi menggunakan teknologi Servlets, JavaServer Pages dan database PostgreSQL. Diharapkan dengan memahami dengan baik pemrograman ini, aplikasi yang dihasilkan dapat memenuhi kebutuhan pengguna dan memudahkan pengguna dalam menggunakannya.

3.2. Identifikasi Kebutuhan

Langkah yang harus diambil untuk mendefinisikan kebutuhan baik kebutuhan dari pengguna dan sistem. Kebutuhan dari pengguna adalah apa saja fitur yang diinginkan pengguna yang nanti akan ada dalam aplikasi. Kebutuhan pengguna mencakup kebutuhan pengguna dari sisi operator dari aplikasi RKA-K/L DIPA. Kebutuhan dari sistem adalah teknologi dan data apa saja yang diperlukan untuk membangun aplikasi.

3.3. Desain Aplikasi

Pada tahap ini akan dilakukan perancangan aplikasi dengan menggunakan metode Use Case Driven Object by Unified Modeling Language. Di dalam bahasa pemrograman Java penulisan kode harus menggunakan metode Object Oriented Programming maka dari itu di dalam metode ini akan ditentukan obyek apa saja yang akan dibuat, proses apa saja yang akan terjadi dalam aplikasi dan kemungkinan kesalahan dalam aplikasi.

3.4. Pemrograman aplikasi

Setelah didapatkan desain aplikasi maka langkah selanjutnya adalah mulai mengembangkan aplikasi. Pada tahap ini akan

digunakan metode *incremental* karena dimungkinkan akan terjadi perubahan pada sisi desain sampai ditemukan desain dan aplikasi yang baik. Bahasa pemrograman yang akan dipakai adalah Java, teknologinya menggunakan Servlets, JavaServer Pages dan database PostgreSQL.

3.4. Integrasi Modul

Pada tahap ini semua modul telah melewati semua tahap uji coba dan telah sempurna berjalan. Integrasi yang dimaksud adalah integrasi kode aplikasi dan *database* masing-masing modul.

3.5. Uji Coba Aplikasi

Uji coba aplikasi yang telah terintegrasi wajib dilakukan untuk menjamin keberhasilan hubungan antar modul. Uji coba yang dilakukan mencakup *unit test*, *case test*, uji coba integrasi dan *stress test* yaitu bagaimana ketahanan aplikasi dalam menerima *request* dari banyak user. Dalam tahap ini masih dimungkinkan untuk perubahan kode program untuk kesesuaian kode antar modul.

3.6. Migrasi Data

Migrasi data diperlukan bagi modul tertentu dikarenakan data tersebut selalu ada meskipun tahun anggaran berganti, misalnya data nilai bangunan, nilai kendaraan, saldo neraca dan lain-lain. Modul Penganggaran tidak memerlukan migrasi data karena data RKA-KL DIPA akan selalu baru setiap pergantian tahun anggaran.

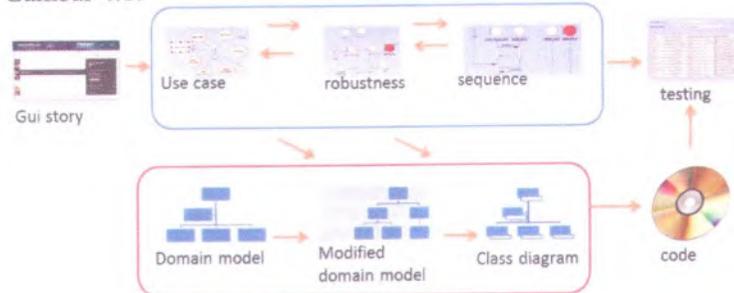
3.7. Penyusunan Buku Tugas Akhir

Tahapan terakhir adalah penulis akan menyusun buku tugas akhir semua proses yang dilakukan dalam pembuatan aplikasi ini. Dokumen ini diharapkan dapat menjadi literatur yang berguna dalam mengembangkan aplikasi sejenis menjadi lebih baik dan bermanfaat.

BAB IV

ANALISIS KEBUTUHAN DAN DESAIN APLIKASI

Pada bab ini akan dijelaskan analisis kebutuhan, arsitektur dan desain aplikasi Siapkan Modul Penganggaran yang disusun berdasarkan *ICONIX proses*. *ICONIX process* adalah suatu proses perancangan aplikasi dengan menterjemahkan proses bisnis aplikasi ke dalam bentuk diagram sehingga memudahkan *programmer* menterjemahkan proses bisnis. *ICONIX process* terdiri dari perancangan *GUI (Graphical User Interface) storyboard*, *domain model*, *use case diagram* serta deskripsi, *robustness diagram* dan *sequence*. Hal ini tergambar dalam Gambar 4.1.



Gambar 4.1 ICONIX Process

Untuk melengkapi desain aplikasi ini, akan disertakan skenario uji coba aplikasi yang mencakup *Unit dan Case Test*.

4.1. Analisis Kebutuhan Aplikasi

Kebutuhan Modul Penganggaran sama dengan aplikasi RKA-K/L. Kebutuhan terdiri dari dokumen sebagai dasar untuk menyusun data, user yang terlibat dalam aplikasi, dan kebutuhan infrastruktur aplikasi.

4.1.1. Dokumen Dasar Hukum

Aplikasi RKA-K/L digunakan untuk menyusun rencana belanja dan pendapatan. Dalam menyusun rencana belanja dibutuhkan dokumen yang menjadi dasar perekaman data rencana belanja dan pendapatan. Dokumen tersebut antara lain :

- Dokumen SP RKA-KL yang berisi pagu alokasi anggaran per satker
- Peraturan Menteri Keuangan Nomor 112/PMK.02/2012 Tentang Petunjuk Penyusunan Dan Penelaahan Rencana Kerja Dan Anggaran Kementerian Negara/Lembaga
- Dokumen peraturan kebijakan pemerintah, misalnya Peraturan Presiden Nomor 22 Tahun 2009 tentang Kebijakan Penganekaragaman Konsumsi Pangan Berbasis Sumber Daya Lokal. Dokumen ini digunakan untuk menentukan arah kebijakan belanja satker yang harus searah dengan kebijakan pemerintah.
- Dokumen lainnya yang mengatur kebijakan pengeluaran satker berkaitan.

4.1.2. Pengguna

Aplikasi RKA-K/L digunakan oleh beberapa instansi dengan level yang berbeda. Jika digunakan di satker maka aplikasi ini berfungsi untuk menyusun rencana belanja dan pendapatan. Jika di tingkat kanwil maka aplikasi ini berfungsi memonitor laporan rencana belanja dan pendapatan hasil dari satker di bawah wilayah kanwil bersangkutan. Untuk tingkat satker memerlukan dua pengguna yang berkepentingan dalam proses penganggaran. Dua pengguna tersebut adalah :

- Operator
Operator adalah pegawai yang bertugas menyusun Kertas Kerja RKA-K/L menggunakan aplikasi untuk

disampaikan ke pada Kuasa Pengguna Anggaran (KPA) untuk disahkan.

- KPA adalah pejabat yang berwenang dalam mengesahkan Kertas Kerja RKA-K/L dan Rencana Penarikan dana.

Kedua user di atas akan menjadi user dalam aplikasi Siapkan Modul Penganggaran.

4.1.3. Fitur aplikasi

Sesuai dengan tujuan tugas akhir ini, adalah membuat aplikasi terintegrasi yang dapat menggantikan beberapa aplikasi lama yang masih terpisah. Dengan melihat aplikasi RKA-K/L maka dapat dirumuskan fitur-fitur apa saja yang wajib diikutsertakan dalam aplikasi Siapkan Modul Penganggaran. Fitur-fitur tersebut antara lain :

- Penyusunan rencana belanja

Fitur ini memuat penyusunan rencana belanja dengan berdasarkan pada struktur anggaran yang telah ditetapkan dalam Peraturan Menteri Keuangan. Struktur anggaran tersebut terdiri dari Program, Kegiatan, Output, Sub Output, Komponen, Sub Komponen, Akun dan detail belanja (lihat gambar 4.2).



Gambar 4.2 Struktur Anggaran

- *Penyusunan rencana pendapatan*
Penyusunan rencana pendapatan harus sesuai dengan Peraturan Menteri Keuangan Nomor 112/PMK.02/2012 dimana harus menyerakan klasifikasi kegiatan dan akun belanja.

4.1.4. Workflow Aplikasi

Modul penganggaran disusun berdasarkan aplikasi yang sudah ada sebelumnya yaitu aplikasi RKA-K/L. Pada dasarnya aplikasi RKA-K/L mempunyai fungsi utama dalam menyusun Kertas Kerja RKA-K/L, kemudian data Kertas Kerja tersebut akan dikirimkan ke kantor pusat masing-masing satker untuk mendapat persetujuan dengan DPR. Aplikasi ini juga mempunyai fungsi tambahan monitoring data DIPA, data Rencana Penarikan Dana. Berikut akan ditunjukkan alur kerja aplikasi RKA-K/L yang nantinya akan menjadi dasar pembuatan aplikasi SIAPKAN Modul Penganggaran.

4.1.5. Penyusunan Rencana Belanja

Proses yang pertama kali dilakukan adalah penyusunan rencanabelanja. Rencana belanja disusun berdasarkan struktur anggaran Output, Sub Output, Komponen, Sub Komponen, Akun dan detil belanja.

- Output adalah hasil capaian dari suatu kegiatan yang akan dilaksanakan
- Sub Output adalah klasifikasi turunan dari output
- Komponen adalah unsur biaya yang terkandung dalam suatu kegiatan, misalnya dalam suatu kegiatan atau proyek terdapat komponen pembayaran gaji, komponen belanja operasional, komponen belanja modal dan lain-lain.
- *Sub Komponen adalah sub klasifikasi komponen yang dirinci lebih kecil lagi.*

- Akun adalah klasifikasi anggaran terkecil yang nantinya akan digunakan dalam system pembayaran.
- Detil belanja adalah penjelasan belanja apa saja, jumlahnya berapa dan berapa harga satuan dari barang atau jasa yang dibeli.

4.1.6. Penyusunan Rencana pendapatan

Rencana pendapatan memuat target pendapatan selama lima tahun ke depan. Rencana ini terdiri dari dua jenis pendapatan:

- Pendapatan Perpajakan
Pendapatan jenis ini hanya ada di instansi Kantor Pelayanan Perpajakan saja
- Pendapatan Bukan Pajak
Pendapatan ini bias berada di semua instansi. Contoh dari pendapatan ini adalah pendapatan sewa rumah Negara, pendapatan biaya pembuatan STNK dan lain-lain.

4.1.7. Penyampaian data Kertas Kerja RKA-K/L

Setelah semua data terbentuk, kemudian dilakukan pencetakan dan kepala kantor sebagai KPA menandatangani sebagai tanda bahwa kertas kerja tersebut telah disetujui. Data berupa file juga disertakan dalam Compact Disk supaya di kantor pusat masing-masing satker dapat diinputkan ke aplikasi RKA-K/L tingkat Kementerian Lembaga.

Namun demikian alur ini nantinya tidak akan disertakan dalam aplikasi SIAPKAN dikarenakan aplikasi SIAPKAN telah menggunakan teknologi *web-based* yang memungkinkan komunikasi antara data aplikasi satker yang berada di daerah kabupaten kota dengan aplikasi kementerian lembaga yang berada di kantor pusat.

4.1.8. Pencetakan Konsep Dipa

Konsep DIPA merupakan suatu konsep dokumen dasar pencairan dana yang dibuat oleh satker untuk mendapatkan

persejuaan dari Direktorat Jenderal Perbendaharaan (DJPB). Konsep DIPA berisi :

- Surat Pengesahan DIPA
- DIPA Halaman I sampai dengan IV

4.1.9. Transfer data DIPA

Konsep DIPA yang telah disetujui nanti akan dikembalikan ke satker dimana DIPA telah dilengkapi nomor DIPA dan tanggal pengesahan DIPA. Selain surat pengesahan, satker juga akan menerima data DIPA yang telah disetujui dan harus dilakukan *import* data DIPA ke aplikasi RKA-K/L. *Import* data ini akan berguna sebagai alat pengawasan terhadap pagu anggaran.

Namun untuk alur **d** dan **c** tidak disertakan dalam aplikasi SIAPKAN karena aplikasi SIAPKAN adalah berbasis web maka dari itu Direktorat Jenderal Perbendaharaan dapat melakukan akses langsung ke aplikasi.

4.1.10. Penyusunan data rencana penarikan dana

Penyusunan rencana penarikan dana berfungsi untuk memperkirakan kebutuhan dana pada suatu periode tertentu sehingga pemanfaatan dana benar-benar efisien (Peraturan Direktur Jenderal Perbendaharaan, 2012).

Penyusunan rencana ini dilakukan per detil belanja dan disusun per bulan. Maka dari itu setiap bulan satker harus sudah mempunyai jumlah perkiraan dana yang akan dicairkan dari pagu yang telah disetujui (lihat gambar 4.3).

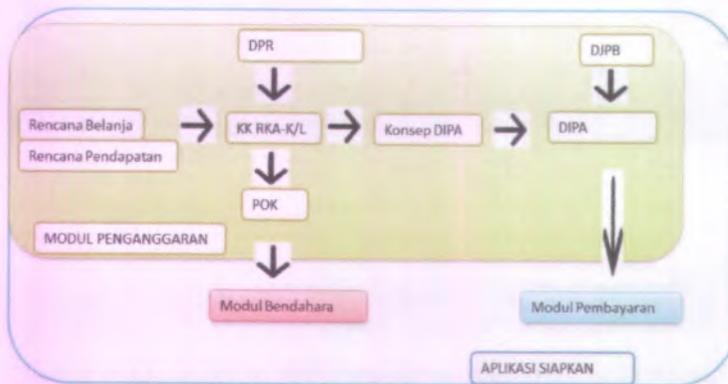


Gambar 4.3 Workflow Aplikasi RKA-K/L

Dari gambar diatas dapat dilihat alur aplikasi RKA-K/L DIPA, sedangkan alur aplikasi SIAPKAN pada dasarnya sama dengan gambar di atas, yang membedakan adalah:

- media komunikasi data antara satker dengan instansi kantor pusat masing-masing dan satker dengan kantor DJPB. Jika aplikasi RKA-K/L menggunakan media *disk* untuk meneruskan data komputer, maka dengan aplikasi SIAPKAN menggunakan data jaringan internet sebagai komunikasinya.
- Setelah DIPA disetujui DJPB, bendahara bisa mulai memulai menyusun transaksi belanjanya. Bendahara menggunakan Modul Bendahara mengakses POK sebagai dasar membuat transaksi belanja.
- Kemudian setelah transaksi dilakukan, untuk pencairan dananya, Modul Pembayaran dapat membayarkan tagihan sebesar jumlah transaksi.

Berikut adalah gambar mengenai *workflow* aplikasi SIAPKAN yang telah mengintegrasikan seluruh sistem baik yang ada di tingkat satker maupu di tingkat kementerian lembaga (lihat gambar 4.4).

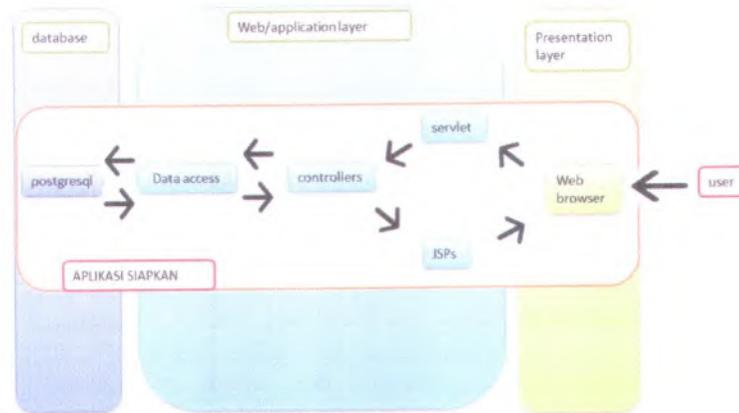


Gambar 4.4 Workflow Sistem Penganggaran Terintegrasi

Perbedaannya terletak pada proses persetujuan dari kantor pusat satker masing-masing dengan dari DJPB telah masuk pada sistem aplikasi SIAPKAN, maka dari itu tidak diperlukan lagi media penyimpanan *disk* untuk mengirim data computer.

4.2. Arsitektur Aplikasi

Di bawah ini merupakan desain arsitektur aplikasi yang didasarkan pada metode *Model, View, and Controller* (lihat gambar 4.5).



Gambar 4.5 Arsitektur Aplikasi

Dari gambar di atas dapat ditentukan bahwa unsur-unsur arsitektur aplikasi SIAPKAN adalah :

1. Jsp : File jsp adalah tampilan aplikasi yang berhubungan langsung dengan user. Jsp bisa berupa:
 - Form isian, form dimana user harus memberikan input data kepada system, misalnya input teks, angka atau tanggal.
 - Daftar, jsp bisa berupa daftar informasi dimana user hanya bisa melihat atau menghapus isi dari daftar tersebut, misalnya daftar akun dimana user dapat menghapus isi dari daftar user tersebut.
2. Servlet : servlet adalah kelas java yang berfungsi meneruskan perintah dari input user kepada kelas *controller* yang tepat.
3. *Controller* : adalah kelas java yang berisi logika aplikasi yang merupakan inti pemroses dari aplikasi. *Controller* mengatur segala aliran perintah dan informasi dari dan ke semua bagian aplikasi.
4. *Data Access/Model*: adalah kelas java yang berfungsi mengolah data di database. Dalam arsitektur SIAPKAN, model yang dipakai menggunakan *Java Persistence API*.

5. *Database* : database adalah penampung data aplikasi.

Setelah user berinteraksi dengan jsp, maka sistem akan memproses aksi apa yang diinputkan oleh user. Berikut adalah penjelasan dari proses yang terjadi di arsitektur aplikasi:

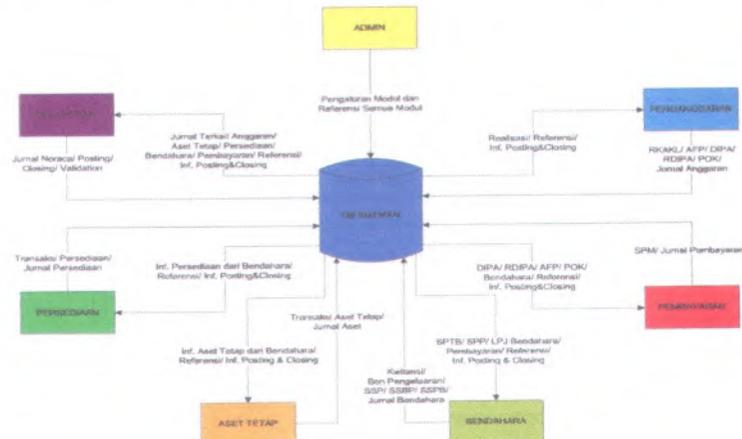
1. Jsp meneruskan perintah dalam bentuk *url* (*Uniform Resource Locator*) dan data yang diinputkan oleh user.
2. Servlet menerima *url* dan meneruskan ke kelas *controller* yang tepat.
3. *Controller* menerima perintah dari servlet kemudian mengambil data yang dikirim dari jsp. *Controller* akan memanggil *Data Access/Model* untuk memproses perintah tersebut dengan mengolah data dimana dalam hal ini bisa berupa mengambil, merekam, mengubah ataupun menghapus data di database.
4. Setelah database diolah maka *Data Access/Modul* akan meneruskan data hasil olahan kepada *controller*. Kemudian *controller* akan meneruskan umpan balik langsung menuju ke jsp untuk ditampilkan hasil proses yang baru saja diinputkan oleh user.

Proses pembuatan arsitektur ini melewati beberapa tahap. Tahap pertama yang dilakukan adalah pembuatan jsp sebagai tampilan. Alasan kenapa jsp dibuat pertama kali adalah untuk menyesuaikan form standar berdasarkan aplikasi RKA-K/L DIPA. Tahap kedua adalah pembuatan struktur database. Dikarenakan sistem pembuatan tabel telah dikendalikan oleh *Java Persistence API* maka penulis hanya membentuk kelas *entity* beserta atribut pembentuknya.

4.3. Hubungan Antar Modul

Aplikasi Sistem Informasi Pelaporan Keuangan Negara (SIAPKAN) yang akan dikembangkan merupakan integrasi dari beberapa aplikasi yang sebelumnya telah digunakan pada Satuan Kerja di lingkungan Direktorat Jenderal Perbendaharaan. Pengembangan aplikasi ini terbagi menjadi tujuh

modul, yaitu Modul Administrasi, Modul Anggaran, Modul Bendahara, Modul Pembayaran, Modul Aset Tetap dan Modul Persediaan. Gambar 4.6 menunjukkan hubungan antar modul.



Gambar 4.6 Desain Aplikasi SIAPKAN

Hubungan serta keterkaitan antar modul 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 aplikasi sesuai dengan kebutuhan masing-masing Satuan Kerja. Output dari Modul Administrasi berupa referensi Anggaran, Aset, Persediaan, Bendahara dan Pembayaran serta manajemen pengguna.
2. Modul Penganggaran berfungsi untuk menyusun kertas kerja Rencana Kerja Anggaran Kementerian Lembaga (RKA-KL) di setiap Satuan Kerja (Satker). Modul Penganggaran 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 Penganggaran. 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 Penganggaran, 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.*

Modul Penganggaran berinteraksi dengan modul Administrasi, Bendahara serta Pembayaran. Keempat modul tersebut mempunyai proses bisnis yang berkesinambungan dalam organisasi.

Integrasi yang dilakukan antara modul Penganggaran dengan modul Administrasi adalah use case Login, Logout serta ambil data di table Referensi. User harus login terlebih dahulu sebelum masuk ke Modul Pembayaran. Begitu pula jika user telah logout, maka tidak dapat masuk lagi ke Modul Pembayaran sebelum melakukan login ulang (lihat gambar 4.7).



Gambar 4.7 Interaksi Antar Modul

Pada Modul Penganggaran, referensi yang dibutuhkan dari Modul Administrasi, antara lain :

1. Table output,
2. Table kegiatan,
3. Table akun, table KPPN,
4. Table lokasi.
5. Tabel tema.
6. Tabel baseline
7. Tabel user (login dan logout).

Modul Pembayaran membutuhkan tabel *t_user* yang diambil melalui *session* ketika pengguna masuk ke dalam aplikasi melalui halaman Login. Data yang diambil dari *session* antara lain username, kode satker, kode KPPN, kode kanwil, dank kode departemen.

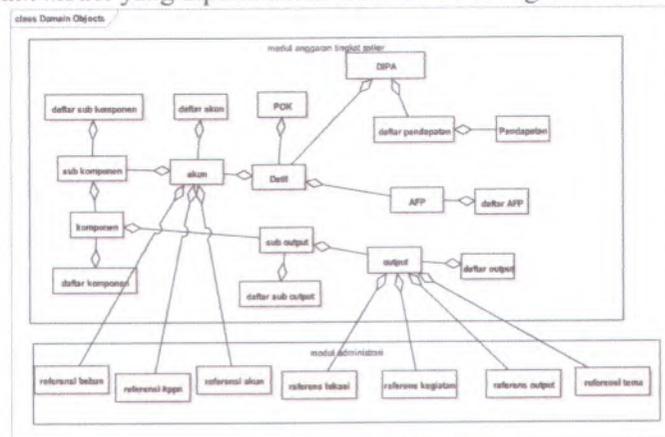
Integrasi Modul Penganggaran dengan Modul Pembayaran mencakup komunikasi data DIPA. Informasi yang diambil dari DIPA adalah *d_smpind* yang menyimpan realisasi pencairan dana.

Sedangkan interaksi dengan Modul Bendahara adalah informasi POK. POK berisi data detil belanja yang memuat pagu dana maksimal yang boleh dibelanjakan oleh bendahara. Tabel yang diambil oleh Modul Bendahara dari Modul Penganggaran adalah :

1. Tabel belanja output
2. Tabel komponen
3. Tabel belanja akun
4. Tabel detail belanja

4.4. Domain Model

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 yang dikembangkan pada Tugas Akhir ini, penulis menginisialisasi *domain model* yang diperlihatkan Gambar 4.8 sebagai berikut.



Gambar 4.8 Domain Model

Berdasarkan *domain model* pada Gambar 4.8, terdapat beberapa *class* yang diinisialisasi, yaitu :

1. **Kertas Kerja RKA-KL**, daftar terperinci kegiatan pemerintah yang mencakup kegiatan belanja dan pendapatan negara dalam satu tahun anggaran.

2. **DIPA**, merupakan Kertas Kerja RKA-KL yang telah mendapatkan pengesahan. Setelah DIPA ini diterbitkan, satker dapat mulai melaksanakan kegiatannya seperti belanja barang atau memungut penerimaan.
3. **POK**, daftar terperinci rencana belanja satuan kerjayang terdiri dari banyak komponen yaitu output, komponen, akun dan rencana penyerapan anggaran. (kumpulan dari detil)
4. **Detil**, form rencana belanja.
5. **Akun**, adalah mata anggaran yang digunakan untuk belanja
6. **Sub Komponen**, adalah penggolongan dari belanja per mata anggaran
7. **Komponen**, adalah klasifikasi belanja dilihat dari jenisnya
8. **Sub Output**, pengklasifikasian output
9. **Output**, adalah sesuatu yang dihasilkan dari kegiatan pemerintah.
10. **Afp**, rincian rencana penarikan dana anggaran dalam satu tahun
11. **KPJM**, kerangka pengeluaran jangka menengah, adalah proyeksi belanja selama 4 tahun ke depan.
12. **Pendapatan**, rencana/target pendapatan instansi selama satu tahun ke depan
13. **Daftar Output**, adalah sekumpulan dari output
14. **Daftar Sub Output**, adalah sekumpulan dari sub output
15. **Daftar Komponen**, adalah sekumpulan dari komponen
16. **Daftar Sub Komponen**, adalah sekumpulan dari sub komponen
17. **Daftar Akun**, adalah sekumpulan dari akun
18. **Daftar AFP**, adalah sekumpulan dari AFP

Terdapat juga beberapa *domain* yang berasal dari modul lain yang terintegrasi pada aplikasi SIAPKAN. *Domain* tersebut diperlukan untuk melengkapi kebutuhan yang diperlukan pada *domain* Modul Pembayaran.

4.5. Class Diagram

Class Diagram adalah penjelasan kelas-kelas beserta atribut dan method apa yang terdapat di dalamnya. Kelas-kelas ini yang akan dipakai dalam aplikasi dan dapat berubah sesuai dengan kebutuhan sewaktu menjalankan tahap pemrograman. Penjelasan lebih detail tentang *Class Diagram* terdapat di Lampiran A.

4.6. Use Case Diagram

Use Case Diagram menunjukkan hal-hal yang dapat dilakukan oleh masing-masing aktor. *Use case diagram* serta deskripsinya dapat dilihat pada Lampiran B. Berikut ini merupakan daftar *use case* yang ada pada Modul Penganggaran :

1. Use Case RUH (terdiri dari 3 use case yaitu rekam, ubah dan hapus) POK yang mencakup :
 - a. Use Case RUH Output
 - b. Use Case RUH Sub Output
 - c. Use Case RUH Komponen
 - d. Use Case RUH Sub Komponen
 - e. Use Case RUH Akun
 - f. Use Case RUH Detil
 - g. Use Case RUH AFP
2. Use Case Rekam Pendapatan
3. Use Case Ubah Pendapatan
4. Use Case Hapus Pendapatan
5. Use Case Tayang DIPA
6. Use Case Tayang KPJM
7. Use Case Approve AFP
8. Use Case Approve Kertas Kerja
9. Use Case Kirim Data AFP
10. Use Case Kirim Data Kertas Kerja
11. Use Case Cetak AFP
12. Use Case Cetak Kertas Kerja
13. Use Case Cetak Laporan

4.7. Robustness Diagram

Robustness diagram merupakan komponen-komponen apa yang terlibat pada saat *use case* berjalan. *Robustness diagram* menjelaskan apayang dilakukan user dan sistem dalam kegiatan tersebut. *Robustness diagram* dapat dilihat pada Lampiran C.

4.8. Sequence Diagram

Sequence diagram merupakan tahap-tahap aktivitas yang terjadi secara berurutan dari awal sampai akhir dari suatu *use case*. Perancangan *sequence diagram* dapat dilihat pada lampiran D.

4.9. Skenario Test Case

Test case merupakan proses yang dirancang untuk menjaga performa aplikasi agar sesuai dengan desain telah yang dibuat. *Test case* akan dijalankan dengan beberapa skenario yang sesuai dengan rancangan pada diagram *use case*. Perancangan skenario dan *test* dapat dilihat pada lampiran E. *Test case* nantinya akan diuji coba berupa *unit test*.

4.10. GUI Story Board

GUI Story Board digunakan sebagai gambaran awal atau bentuk dari setiap form yang ada pada sistem yang dikembangkan. Pada perancangan GUI untuk aplikasi SIAPKAN meliputi rancangan desain tampilan form rekam, ubah dan tayang untuk masing-masing Use Case serta desain tampilan error apabila terjadi kesalahan dalam pemrosesan data. Untuk lebih lengkapnya dapat dilihat lampiran F.

BAB V IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini akan dijelaskan proses pembuatan aplikasi SIAPKAN Modul Penganggaran. Proses ini terdiri dari pembuatan kode program dengan disertai tes skenario uji coba setelah aplikasi selesai dibuat.

5.1. Lingkungan Implementasi

Proses pembuatan aplikasi SIAPKAN ini dilakukan di komputer jinjing atau laptop dengan spesifikasi sebagai berikut :

- Prosesor : Intel® Core i3 2350M, 2.3 GHz
- Memori RAM : 4 GB
- Memori grafik : 2 GB
- Sistem Operasi : Windows 7

Sedangkan *tools* yang dipakai pembuatan desain dan kode aplikasi adalah sebagaimana tabel 5.1:

Tabel 5.1 Tool Yang Digunakan

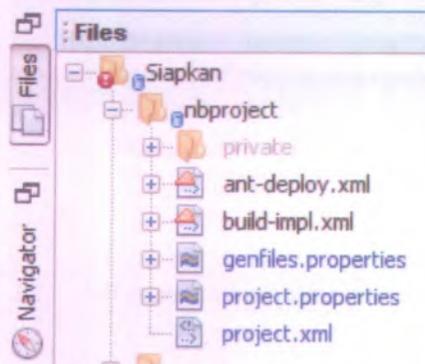
| Tools | |
|----------------------|--|
| Aplikasi Desain | Enterprise Architect |
| <i>Java Resource</i> | JDK1.6.0_06 |
| <i>Java Editor</i> | Netbeans IDE 7.1.1 |
| Web server | Apache Tomcat 7.0.22 |
| ORM | JPA Persistence |
| Sistem Database | PostgreSQL 9.1 |
| Editor Html | Adobe Dreamweaver CS5 |
| Tools lain | Bitnami wappstack, Full Converter Enterprise |

Modul anggaran adalah salah satu modul dari aplikasi SIAPKAN. Beberapa modul lain pada saat yang bersamaan sedang dikerjakan, maka dari itu untuk mendukung integrasi semua

modul, digunakan teknologi *subversion* yang memberikan fasilitas hosting untuk menempatkan *source* aplikasi sehingga setiap modul dapat langsung terintegrasi pada saat proses pembuatan. *Subversion* yang digunakan memakai jasa dari Google dengan alamat <http://code.google.com/p/sistem-informasi-akuntansi-pelaporan-keuangan/>.

5.2. Struktur Direktori Aplikasi

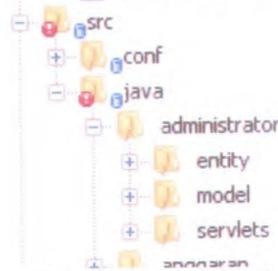
Sesuai dengan aplikasi web berbasis java pada umumnya, aplikasi SIAPKAN mempunyai struktur direktori yang terdiri dari tiga sub direktori, yaitu *nbproject*, *src* dan *web*. Sub direktori *nbproject* adalah direktori yang menyimpan metadata aplikasi seperti alamat database, informasi login database, informasi sesi dan lain-lain. Sub direktori ini juga memuat informasi konfigurasi aplikasi terkait dengan sistem database yang digunakan, dan lain-lain (lihat gambar 5.1).



Gambar 5.1 Sub Direktori Nbproject

Untuk sub direktori *src* adalah memuat kode kelas-kelas dan modul java yang menjadi inti dari aplikasi. Di dalam direktori ini terdapat pembagian sub sub direktori berdasarkan masing-masing modul aplikasi SIAPKAN. Kemudian dari masing-masing Dari sub sub direktori per modul ini dibagi kembali ke dalam 3 direktori kecil berupa *entity*, *model*, dan *serlet*. Tiga direktori tersebut

- Direktori entity berisi kelas-kelas java
- Direktori model berisi kode program yang berfungsi merekam, mengubah, menghapus dan semua logika yang berkaitan dengan akses ke database.
- Direktori servlet berisi logika pemrograman yang mengatur hubungan antara perintah dari *interface* aplikasi dengan model yang menangani database (lihat gambar 5.2).



Gambar 5.2 sub direktori src

Sedangkan sub direktori web berisi file berfungsi mengatur *interface* aplikasi. File ini dalam direktori ini mempunyai format jsp (JavaServer Pages) yang dalam penulisannya sama dengan format file html. Kemudian dalam membantu file jsp untuk meningkatkan kualitas tampilan, disertakan pula file gambar dalam format jpg dan *script* file css untuk mengatur detail dari jsp. Direktori ini hanya berfokus bagaimana memberikan tampilan halaman yang informatif, menarik dan mudah dicerna oleh pengguna jadi tidak terdapat logika pemrograman di dalamnya (lihat gambar 5.3).



Gambar 5.3 sub direktori web

5.3. Konfigurasi Aplikasi Siapkan

Aplikasi Siapkan merupakan aplikasi berbasis web, maka dari itu terdapat file konfigurasi yang berisi informasi metadata aplikasi. File-file tersebut antara lain context.xml, persistence.xml dan web.xml. penjelasan singkatnya adalah sebagai berikut :

- Context.xml berfungsi mengatur alamat url pangkal (*root*) aplikasi pada web browser (lihat gambar 5.4).

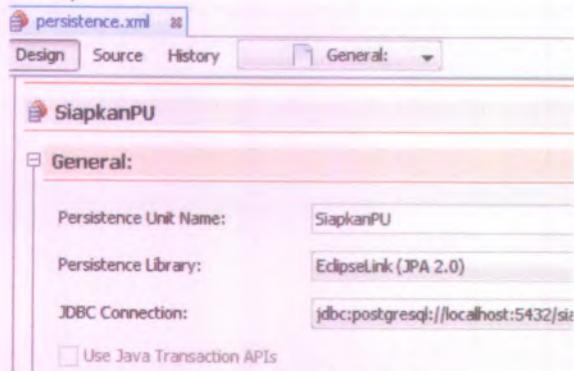
```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <Context antiJARLocking="true" path="/Siapkan">
3  </Context>
4

```

Gambar 5.4 context.xml

- Persistence.xml berfungsi mengatur konektivitas kode program untuk bisa mengakses ke database. Informasi yang terdapat di dalamnya antara lain :
 - Informasi konektivitas (*username* dan *password*)
 - Nama entitas aplikasi merupakan cerminan nama tabel dalam database (lihat gambar 5.5 dan gambar 5.6).



Gambar 5.5 informasi koneksi

Include All Entity Classes in "Siapkan" Module

Include Entity Classes:

```
bendahara.entity.Jnstran
bendahara.entity.DBspj
bendahara.entity.DSpj
bendahara.entity.DSbukti
bendahara.entity.DBukti
bendahara.entity.DSlpj
bendahara.entity.DSbuktiup
aset.entity.Kdp
```

Gambar 5.6 informasi entitas

- Web.xml digunakan untuk mendefinisikan alamat url *servlet* beserta alamat kelas *servlet* yang dituju (lihat gambar 5.7).

UpServlet -> /tranup

| | |
|---|--|
| Servlet Name: | UpServlet |
| Description: | |
| <input checked="" type="radio"/> Servlet Class: | bendahara.servlet.UpServlet |
| <input type="radio"/> JSP File: | |
| URL Pattern(s): | /tranup |
| | Use comma (,) to separate multiple patterns. |

Gambar 5.7 alamat url dan kelas servletnya

5.4. Penulisan Kode Program

Di dalam penulisan kode program, terdapat pembagian direktori yang disesuaikan dengan modul masing-masing, sehingga di dalam direktori src terdapat sub direktori untuk modul administrasi, anggaran, bendahara, pembayaran, persediaan, aset dan pelaporan. Secara berurutan, penulis menentukan terlebih dahulu entitas apa yang akan dibuat, kemudian membuat model,

mendesain tampilan dan terakhir membuat logika proses di servlet.

5.4.1. *Package* anggaran.entity

Di dalam paket ini terdapat beberapa entitas yang merupakan unsur pembentuk aplikasi RKA-KL/DIPA, antara lain klasifikasi anggaran yang terdiri dari program, kegiatan, output, komponen, akun. Satu kelas dalam modul ini merupakan bagian dari entitas referensi yang berada di modul administrasi. Dengan kata lain satu entitas Modul Penganggaran bersumber dari daftar entitas di Modul Penganggaran, misalnya output yang di definisikan di Modul Penganggaran diambil dari daftar output yang telah didefinisikan sebelumnya di modul administrasi. Kelas-kelas yang terdapat dalam paket ini adalah kelas Output.java, SubOutput.java, Komponen.java, SubKomponen.java, Akun.java, Detil.java, Afp.java, Dipa.java, Kpjm.java, KertasKerja.java, Kpjm.java dan Pendapatan.java (lihat gambar 5.8).

```

@Entity
public class Output implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    @ManyToOne
    private TDekon tDekon;
    @Basic(optional = false)
    @Column(name = "chang", length = 4)
    private String chang;
    @ManyToOne
    private TJendok tJendok;
    @ManyToOne
    private TOutput tOutput;
    @ManyToOne
    private TSatker tSatker;
}

```

Gambar 5.8 kelas entitas output

5.4.2. *Package anggaran.model*

Di dalam paket ini berisi dengan kode java yang berhubungan dengan manipulasi data dalam database. Semua entitas yang akan dijadikan table dibuatkan modelnya pada kelas ini. Misalnya terdapat *method* rekam, ubah, hapus dan cari. Kelas-kelas yang terdapat dalam paket ini adalah `DaftarOutput.java`, `DaftarSubOutput.java`, `DaftarKomponen.java`, `DaftarSubKomponen.java`, `DaftarAkun.java`, `Pok.java`, `DaftarAfp.java`, `DaftarPendapatan.java`, `DaftarKpjm.java` dan `DaftarKertasKerja.java` (lihat gambar 5.9).

```
public class DaftarOutput {

    public DaftarOutput() {
        emf = Persistence.createEntityManagerFactory("SiapkanPU");
    }

    private EntityManagerFactory emf = null;

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

public List<Output> getOutput() {
    List<Output> output = new ArrayList<Output>();

    EntityManager em = getEntityManager();
    try {
        Query q = em.createQuery("SELECT a FROM Output AS a");
        output = q.getResultList();

    } finally {
        em.close();
    }
    return output;
}

public Output findOutput(Long id) {
    EntityManager em = getEntityManager();
    try {
        return em.find(Output.class, id);
    }
}
```

Gambar 5.9 kelas model DaftarOutput

5.4.3. *Package anggaran.servlet*

Sedangkan di dalam paket servlet ini terdapat dua pemisahan fungsi yang bertujuan untuk meringkas kode program. Fungsi pertama yaitu kelas servlet yang mengendalikan perintah dari *interface* ke kelas *controller*, kemudian kelas *controller* yang berfungsi memproses perintah untuk dijalankan lebih lanjut. Sebagai contoh, ketika pengguna menekan rekam, maka kelas servlet akan meneruskan perintah tersebut ke kelas *controller* dengan *method* prosesRekam. Hal ini akan mempersingkat kode logika yang berada di kelas servlets karena terjadi pemisahan fungsi antara fungsi pengiriman perintah dengan fungsi logika. Kelas-kelas yang ada di paket ini adalah AfpController.java, AfpServlet.java, DipaServlet.java, PokServlet.java, OutputController.java, SubOutputController.java, KomponenController.java, SubKomponenController.java, AkunController.java, DetilController.java, KpjmServlet.java, KpjmServlet.java, KertasKerjaServlet.java dan KertasKerjaController.java (lihat gambar 5.10 dan gambar 5.11).

```
public class PokServlet extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
        throws ServletException, IOException, NonexistentEntityException {
        String pageref = request.getParameter("pageref");
        HttpSession session = request.getSession();

        if(pageref.equals("rekam_output")) {
            session.removeAttribute("error");
            session.removeAttribute("sukses");
            session.removeAttribute("tempinput");
            request.getRequestDispatcher("/pages/anggaran/rekam_output.jsp").forw
        } if(pageref.equals("rekam_sub_output")) {
            session.removeAttribute("error");
            session.removeAttribute("sukses");
            session.removeAttribute("tempinput");
            request.getRequestDispatcher("/pages/anggaran/rekam_sub_output.jsp").
```

Gambar 5.10 kelas PokServlet

```

OutputController.java
Source History
90     TKabkota < Kabkota >;
91     DaftarKabkota < DaftarKabkota > = new DaftarKabkota();
92     TTemp < Temp >;
93     DaftarTemp < DaftarTemp > = new DaftarTemp();
94
95     public String prosesRekamOutput() {
96
97         String kodekon = request.getParameter("kodekon");
98         String stroutput = request.getParameter("id_siswa");
99         String kdab = request.getParameter("kdab");
100        String strIdkabkota = request.getParameter("id_kab");
101        String volkpjms1Str = request.getParameter("volkpjms1");
102        String volkpjms2Str = request.getParameter("volkpjms2");
103        String volkpjms3Str = request.getParameter("volkpjms3");
104        String volkpjms4Str = request.getParameter("volkpjms4");
105        String thnMulaiStr = request.getParameter("thn_mulai");
106        String thnHentiStr = request.getParameter("thn_henti");
107        String kdtemp = request.getParameter("kdtemp");

```

Gambar 5.11 kelas OutputController

5.5.Pembuatan Tampilan

Dalam halaman *interface* aplikasi SIAPKAN, penulis menggunakan aplikasi Dreamweaver untuk mempercepat proses pembuatannya. *Script* css juga ikut disertakan dalam membantu memperbaiki tampilan misalnya dari sisi ukuran teks dan jenis *font* yang diinginkan. *File* jsp dan css semuanya berada di dalam direktori web. *File* jsp yang dibuat antara lain *afp.jsp*, *akun.jsp*, *dipa.jsp*, *dipa1.jsp*, *dipa1b.jsp*, *dipa2.jsp*, *dipa3.jsp*, *dipa4.jsp*, *spdipa.jsp*, *edit_akun.jsp*, *rekam_akun.jsp*, *edit_output.jsp*, *rekam_output.jsp*, *edit_sub_output.jsp*, *rekam_sub_output.jsp*, *edit_komponen.jsp*, *rekam_komponen.jsp*, *edit_sub_komponen.jsp*, *rekam_sub_komponen.jsp*, *edit_detil.jsp*, *rekam_detil.jsp*, *mak.jsp*, *menu_anggaran.jsp*, *output.jsp*, *pok.jsp* dan *style.css* (lihat gambar 5.12).

Data Output

| Program/Regulasi/Output | Sub-Output | Komponen | Sub-Komponen | Alokasi | Detail | Vol | Harga | Jumlah | Pribes |
|-------------------------|------------|----------|--------------|---------|--------|-----|-------|-------------|--------|
| | | | | | | | | 112.000.000 | |
| | | | | | | | | 40.000.000 | |
| | | | | | | | | 40.000.000 | |
| | | | | | | | | 40.000.000 | |
| | | | | | | | | 1.000 | |
| | | | | | | | | 1.000 | |
| | | | | | | | | 40.000.000 | |
| | | | | | | | | 40.000.000 | |
| | | | | | | | | 122.000.000 | |
| | | | | | | | | 20.000.000 | |
| | | | | | | | | 20.000.000 | |
| | | | | | | | | 20.000.000 | |

Gambar 5.12 tampilan list output

5.6. Integrasi Aplikasi

Tahapan ini dilakukan untuk mengintegrasikan seluruh modul ke dalam aplikasi SIAPKAN. Integrasi dilakukan menyesuaikan dengan proses bisnis organisasi. Modul Penganggaran berinteraksi dengan modul Administrasi, Bendahara serta Pembayaran. Keempat modul tersebut mempunyai proses bisnis yang berkesinambungan dalam organisasi.

Integrasi yang dilakukan antara modul Penganggaran dengan modul Administrasi adalah use case Login, Logout serta ambil data di table Referensi t_user. User harus login terlebih dahulu sebelum masuk ke Modul Pembayaran. Begitu pula jika user telah logout, maka tidak dapat masuk lagi ke Modul Pembayaran sebelum melakukan login ulang.

Pada Modul Penganggaran, referensi yang dibutuhkan dari Modul Administrasi, antara lain : table output, table kegiatan, tabel kewenangan, tabel lokasi dan kabupaten/kota dan tabel tema untuk perekaman transaksi output (lihat gambar 5.13).

Halaman Rekam Output

Rekam Sub Output

Kode Kegiatan/KodeOutput/Output
... 1004 / 01 /

Dukungan Pelaksanaan sidang MPR dan Alat Kelengkapannya

Kode IB: 00 / Base Line

Kode Propinsi Nama Propinsi: 01 / DKI JAKARTA

Kode Kab/Kota Nama Kabupaten/Kota: 51 KOTA JAKARTA PUSAT

Volume: (Ditaring menggunakan fungsi "paste" /)

Kewenangan: 1 / Kantor Pusat

Tahun Awal: 2012 Tahun Akhir: 2012

Volume KPJM: (Ditaring menggunakan fungsi "paste" /)

TAHUN 2011 TAHUN 2012 TAHUN 2013 TAHUN 2014 TAHUN 2015

0

Tema:
001 / Pemberantasan Kemiskinan dan Kelaparan (MDG's_1)

Gambar 5.13 Halaman Rekam Output

Kemudian Modul Penganggaran juga membutuhkan table akun, tabel KPPN, mata anggaran, sumber dana dan kegiatan dalam perekaman transaksi akun dan pendapatan (lihat gambar 5.14 dan 5.15).

Halaman Rekam Akun

Rekam Detil

Output / Sub Output / Komponen / Sub Komponen :
01 / 111 / 001/ A.sub Komponen 1 ▾

Mata Anggaran :
... 511125 /
Belanja Tunj. PPh PNS

KPPN :
001 / BANDA ACEH ▾

Sumber Dana : RUPIAH MURNI PLN LOCAL COST/RMP PNPB

Gambar 5.14 Halaman Rekam Akun

Halaman Rekam Pendapatan

List Pendapatan

Kegiatan :
... 1007 / Pembangunan, Pengadaan, Peningkatan dan Pengelolaan Sarana dan Prasarana MPR

Mata Anggaran :
... 421312 / Pendapatan Royalti

Jenis PRMBP : Umum Fungsional

Target Pendapatan:
TAHUN 2011 TAHUN 2012 TAHUN 2013 TAHUN 2014 TAHUN 2015

Gambar 5.15 Halaman Rekam Pendapatan

Untuk mengakses data referensi tersebut, maka Modul Penganggaran harus mengambil data yang terdapat pada modul Administrasi.

Sedangkan interaksi dengan Modul Bendahara adalah informasi POK. POK berisi data detil belanja yang memuat pagu dana

maksimal yang boleh dibelanjakan oleh bendahara. Bendahara dalam merekam transaksi tidak bisa sembarangan, karena harus berpedoman kepada data detil belanja dan tidak boleh menyimpang ke belanja lainnya. Transaksi yang direkam oleh Modul Bendahara berasal dari rencana belanja (POK) yang telah direkam sebelumnya oleh Modul Penganggaran. Gambar 5.16 menunjukkan rencana belanja (POK) yang telah direkam di Modul Penganggaran.

Data Pedoman Operasional Kegiatan

| Program / Kegiatan / Output / Sub Output /Komponen /Sub Komponen /Item / Detil | Vol | Harga | Jumlah | Priban |
|--|-----|---------------|----------------|--------|
| 111 sub output2 | | | 75.157.001,000 | |
| 001 - Komponen Pembayaran Gaji dan Tunjangan | | | 75.157.001,000 | |
| A sub komponen 1 | | | 75.157.001,000 | |
| 511111 - Akun Belanja Gaji Pokok PNS | | | 1.000.000,000 | |
| belanja gaji pokok | 1 | 1.000.000,000 | 1.000.000,000 | |
| 511119 - Akun Belanja Pembulanan Gaji PNS | | | 1.000,000 | |
| belanja gaji pembulanan | 1 | 1.000,000 | 1.000,000 | |
| 511121 - Akun Belanja Tunj. Suami/Istri PNS | | | 10.800.000,000 | |
| belanja tunjangan suami/istri | 1 | 5.000.000,000 | 10.000.000,000 | |
| 511122 - Akun Belanja Tunj. Anak PNS | | | 2.000.000,000 | |

Gambar 5.16 Data POK (Pedoman Operasional Kegiatan)

Integrasi Modul Penganggaran dengan Modul Pembayaran mencakup komunikasi data DIPA. Informasi yang diambil dari DIPA adalah berapa jumlah pagu dana yang masih tersedia yang bisa dilakukan pencairan dananya. Dalam data DIPA terdapat informasi pagu dana awal, jumlah realisasi sampai dengan pencairan terakhir serta sisa pagu dana yang bisa dicairkan. Dengan integrasi ini maka dari Modul Penganggaran dapat memantau berapa jumlah pencairan dana yang telah dilakukan dan berapa sisa dananya.

Data POK di atas menjadi dasar bagi Modul Bendahara untuk menyusun transaksi. Di Modul Bendahara ketika akan merekam transaksi akan mengambil data dari POK tersebut, kemudian menginputkan nilai transaksi dimana nilai transaksi tidak boleh melebihi nilai pagu rincian belanja di dalam POK. Gambar 5.17 menunjukkan menu rekam transaksi LS yang mengambil POK dari Modul Bendahara.

Penerima / Penyetor
 Bendahara Bendahara Pengeluaran

Uraian Transaksi
 Belanja Keperluan Kantor

Mata Anggaran

| | | | | |
|------|----|-----|--------|--|
| 1006 | 01 | 002 | 521111 | Belanja Keperluan Perkantoran |
| 1006 | 01 | 002 | 521111 | Belanja Keperluan Perkantoran |
| 1008 | 05 | 333 | 521111 | Belanja Keperluan Perkantoran |
| 1006 | 01 | 001 | 524111 | Belanja perjalanan biasa (DM) |
| 1006 | 01 | 001 | 531114 | Belanja Modal Pembuatan Sertifikat Tanah |
| 1006 | 01 | 002 | 532111 | Belanja Modal Peralatan dan Mesin |
| 1008 | 05 | 333 | 541111 | Belanja Pembayaran Bunga Surat Perbendaharaan Negara 7 Rupiah |
| 1008 | 05 | 333 | 551214 | Belanja Subsidi Minyak Bakar |
| 1008 | 05 | 333 | 581141 | Belanja TVRI (untuk menampung belanja dari ma subsidi TVRI -551414) |

Kembali Simpan

Gambar 5.17 Integrasi Dengan Modul Bendahara
 Di Modul Bendahara tidak dapat merekam transaksi selain yang telah diinputkan di POK Modul Penganggaran.
 Sedangkan integrasi dengan Modul Pembayaran adalah pengawasan pagu DIPA. Pengawasan ini mencakup monitoring realisasi pencairan dana beserta sisa pagu dana yang masih bisa dicairkan dananya. DIPA mengambil data realisasi dari transaksi yang telah dibayarkan di Modul Pembayaran. Gambar 5.18 menunjukkan data pagu DIPA, realisasi pencairan dana beserta sisa pagu dana yang masih bisa dicairkan dananya.

Data Dipa

| Kode / Nama MIPA Kegiatan | Pagu | Realisasi | Sisa Pagu |
|--|----------------|------------|----------------|
| (01) Belanja Pegawai | 37,143,000,000 | 0 | 37,143,000,000 |
| (011111) Belanja Gaji Pokok PNS | 1,000,000,000 | 0 | 1,000,000,000 |
| (011119) Belanja Pembulatan Gaji PNS | 1,000,000 | 0 | 1,000,000 |
| (011121) Belanja Tunj. Suami/Istri PNS | 10,000,000,000 | 0 | 10,000,000,000 |
| (011122) Belanja Tunj. Anak PNS | 2,000,000,000 | 0 | 2,000,000,000 |
| (011129) Belanja Tunj. Struktural PNS | 10,000,000,000 | 0 | 10,000,000,000 |
| (011214) Belanja Tunj. Fungsional PNS | 2,000,000,000 | 0 | 2,000,000,000 |
| (011215) Belanja Tunj. PPH PNS | 100,000,000 | 0 | 100,000,000 |
| (011126) Belanja Tunj. Baras PNS | 10,000,000,000 | 0 | 10,000,000,000 |
| (0111211) Belanja Tunj. Perbaikan Penghasilan PNS | 1,000 | 0 | 1,000 |
| (0111212) Belanja Tunj. Cacat PNS | 2,000,000 | 0 | 2,000,000 |
| (0111291) Belanja Tunj. Daerah Terpencil/Sangat Terpencil PNS | 40,000,000 | 0 | 40,000,000 |
| (011151) Belanja Tunjangan Umum PNS | 2,000,000,000 | 0 | 2,000,000,000 |
| (02) Belanja Barang | 20,521,200,000 | 0 | 20,521,200,000 |
| (021111) Belanja Keperluan Perbaikan | 121,200,000 | 0 | 121,200,000 |
| (024111) Belanja perjalanan dinas (DM) | 20,000,000,000 | 0 | 20,000,000,000 |
| (03) Belanja Modal | 100,000,000 | 0 | 100,000,000 |
| (031114) Belanja Modal Pembuatan Sertifikat Tanah | 0 | 0 | 0 |
| (032111) Belanja Modal Peralatan dan Mesin | 100,000,000 | 50,000,000 | 50,000,000 |
| (07) Belanja Bantuan Sosial | 1,000,000,000 | 50,000,000 | 950,000,000 |
| (041111) Belanja Pembayaran Bunga Surat Perbendaharaan Negara 7 Hapiah | 1,000,000,000 | 0 | 1,000,000,000 |
| (08) Belanja Lainnya | 20,000,000 | 7,000,000 | 13,000,000 |

Gambar 5.18 Data Dipa

5.7. Uji coba aplikasi

Dalam uji coba aplikasi ini menggunakan beberapa skenario untuk menguji baik validasi input user maupun stabilitas sistem. Skenario itu antara lain :

- Mengecek apakah user telah lengkap mengisi dalam masing-masing form.
 - Mengecek apakah input yang dimasukkan sesuai dengan format yang bisa diterima oleh sistem
 - Mengecek kesesuaian input tanggal, misalkan pada bulan Februari tidak boleh ada tanggal 30 dan 31.
 - Mengecek apakah ada input yang sama dalam satu daftar (tidak boleh ada duplikasi data)
 - Mengecek proses hapus ke daftar apakah data benar-benar terhapus dari daftar.
 - Mengecek proses *approve* apakah status data telah berubah.
- Dengan menggunakan dokumentasi tes skenario uji coba (lampiran E) maka didapatkanlah hasil seperti ditunjukkan tabel di bawah ini (lihat tabel 5.2):

Tabel 5.2 Tabel Use Case

| No. | Uraian | Hasil |
|-----|-----------------------------|----------|
| 1. | Tes rekam/ubah output | berhasil |
| 2. | Tes rekam/ubah sub output | berhasil |
| 3. | Tes rekam/ubah komponen | berhasil |
| 4. | Tes rekam/ubah sub komponen | berhasil |
| 5. | Tes rekam/ubah akun | berhasil |
| 6. | Tes rekam/ubah detil | berhasil |
| 7. | Tes rekam/ubah afp | berhasil |
| 8. | Tes hapus output | berhasil |
| 9. | Tes hapus sub output | berhasil |
| 10. | Tes hapus komponen | berhasil |
| 11. | Tes hapus sub komponen | berhasil |
| 12. | Tes hapus akun | berhasil |
| 13. | Tes hapus detil | berhasil |
| 14. | Tes approval afp | berhasil |
| 15. | Tes cetak AFP | berhasil |
| 16. | Tes tayang dipa | berhasil |
| 17. | Tes cetak Dipa | berhasil |

BAB VI

KESIMPULAN DAN SARAN

Bagian ini berisi kesimpulan dan saran setelah dikerjakannya aplikasi SIAPKAN beserta dokumentasinya.

6.1 Kesimpulan

Kesimpulan yang dapat ditarik setelah penulis menyelesaikan tugas akhir ini antara lain :

1. Aplikasi Siapkan Modul Penganggaran telah selesai dibangun dengan fitur yang sama dengan aplikasi RKA-K/L kecuali yang telah dijelaskan dalam batasan masalah. Hal ini
2. Modul Penganggaran telah berhasil diintegrasikan dengan Modul Administrasi ditunjukkan Modul Penganggaran dapat mengakses table referensi seperti Output, Lokasi, KPPN dan lain-lain. Jika table referensi tersebut diubah dari Modul Administrasi, maka data yang diterima oleh Modul Anggaran juga akan berubah.
3. Modul Penganggaran telah berhasil diintegrasikan dengan Modul Bendahara, ditunjukkan dengan Modul Bendahara dapat mengakses data POK dalam merekam transaksi.
4. Modul Penganggaran telah berhasil diintegrasikan dengan Modul Pembayaran, hal ini ditunjukkan oleh data pagu dana, realisasi sampai dengan pencairan dana terakhir dan sisa pagu dana yang termuat dalam DIPA dapat diakses dari Modul Pembayaran.
5. Aplikasi SIAPKAN telah mampu menghilangkan duplikasi data sehingga informasi keuangan di satker semakin valid.

6.2 Saran

Setelah menyelesaikan aplikasi SIAPKAN modul Penganggaran, saran yang penulis dapat kumpulkan untuk pengembangan lebih lanjut antara lain :

1. Untuk lebih mempercepat kinerja aplikasi dapat digunakan *framework* Spring yang telah mengaplikasikan metode Model View Control (MVC) dengan baik.
2. Pengembangan lebih lanjut dapat diintegrasikan sekaligus aplikasi instansi yang berada di atas level satker, misalnya aplikasi yang terdapat di masing-masing kantor wilayah satker bersangkutan.

DAFTAR PUSTAKA

- Basham, B., Sierra, K., & Bates, B. (2008). *Head Firsts Servlets and JSP Second Edition*. California: O'Reilly.
- Blum, R. (2007). *PostgreSQL 8 For Windows*. New York: Mc Graw Hill.
- Hall, M. (2000). *Core Servlets and JavaServer Pages*. Upper Saddle River: Prentice-Hall, Inc.
- Hermawan, B. (2004). *Menguasai Java Programming 2 dan Object Oriented Programming*. Yogyakarta: Penerbit Andi.
- Momjian, B. (2001). *PostgreSQL Introduction and Concepts*. Canada: Addison-Wesley.
- Peraturan Direktur Jenderal Perbendaharaan (2012) *Peraturan Ditjen Perbendaraan Nomor PER-03/PB/2010 tentang Perkiraan Penarikan Dana Harian Satuan Kerja dan Perkiraan Pencairan Dana Harian KPPN*, Jakarta : Sekretariat Direktur Jenderal Perbendaharaan.
- Peraturan Menteri Keuangan Republik Indonesia. (2012). *Peraturan Menteri Keuangan Nomor 112/PMK.02/2012 Tentang Petunjuk Penyusunan Dan Penelaahan Rencana Kerja Dan Anggaran Kementerian Negara/Lembaga*. Jakarta: Sekretariat Kementerian Keuangan.
- Reese, G. (2000). *Database Programming with JDBC and Java, Second Edition*. California: O'Reilly & Associates, Inc.
- Republik Indonesia. (2004). *Undang-Undang Republik Indonesia Nomor 1 Tahun 2004 Tentang Perbendaharaan Negara*. Jakarta: Sekretariat Negara.

Suharto, B. H. (2006). *Membangun Aplikasi Menggunakan Qt Designer dengan Database PostgreSQL/MySQL*. Yogyakarta: Penerbit Andi.

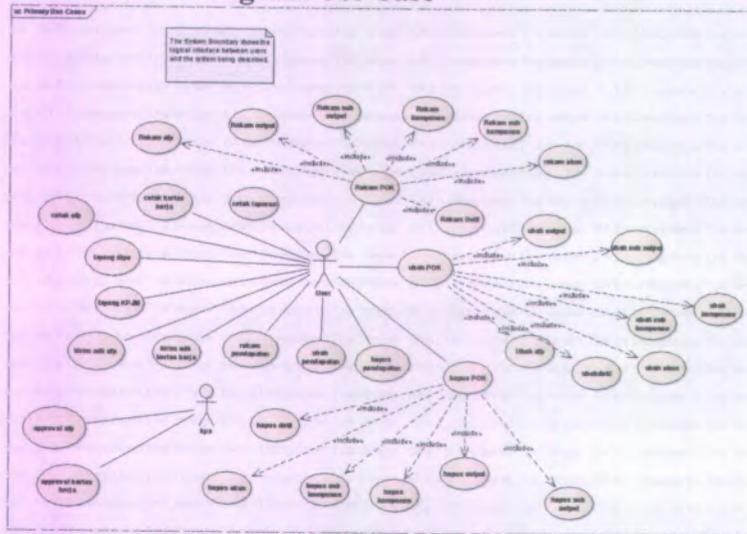
Wijono, S. H. (2007). *Pemrograman Java Servlet dan JSP dengan Netbeans*. Yogyakarta: C.V. ANDI OFFSET (Penerbit ANDI).



LAMPIRAN A
CLASS MODEL

LAMPIRAN B
DIAGRAM DAN DESKRIPSI USE CASE

B.1. Diagram Use Case



Gambar 9.1 Diagram Use Case

B.2. Deskripsi Use Case Rekam Detil

Tabel 9.1 Deskripsi Use Case Rekam Detil

| | |
|--|----------------------------|
| UC01 – Rekam Detil | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam detil. | |
| Triggers: • Operator Anggaran mengklik tombol Rekam | |
| Basic course: Sistem menampilkan form rekam detil. User memilih akun mengisi volume kegiatan, satuan volume kegiatan, harga satuannya, user menekan simpan | |
| Post-conditions: | |

Sistem mengecek apakah field kosong, input yang dimasukkan adalah angka dan bukan minus serta data detail dengan klasifikasi akun, sub komponen, komponen, sub output dan output telah ada dalam daftar detail. Sistem menyimpan ke daftar detail

Alternate courses:

1. jika field volume, satuan volume, dan harga satuan kosong, 3system menampilkan pesan kesalahan.
2. jika field volume atau harga dimasukkan bukan angka, maka 3system akan menampilkan peringatan kesalahan
3. jika field volume atau harga dimasukkan angka minus, maka 3system akan menampilkan pesan kesalahan
4. jika detail yang dipilih telah ada di daftar detail, 3system menampilkan pesan kesalahan

B.3. Deskripsi Use Case Rekam Afp

Tabel 9.2 Deskripsi Use Case Rekam Afp

| | |
|---|----------------------------|
| <i>UC02</i> – Rekam Afp | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam afp. | |
| Triggers: • Operator Anggaran mengklik tombol Rekam | |
| Basic course: 3system menampilkan form rekam afp dari detail belanja. User mengisikan nilai kontrak dan non kontrak, rencana penyerapan dana dari bulan januari sampai dengan desember. User menekan simpan | |
| Post-conditions: Sistem mengecek input yang dimasukkan adalah angka dan bukan minus. Sistem menyimpan ke daftar afp | |
| Alternate courses: 1. jika semua field yang diisi bukan angka, maka 3system akan menampilkan pesan error | |

2. jika semua field dimasukkan berupa angka minus, maka 4system akan menampilkan pesan error
3. jika rincian afp melebihi pagu POK, atau nilai kontrak melebihi nilai pagu maka 4system akan menampilkan pesan error.

B.4. Deskripsi Use Case Rekam Komponen

Tabel 9.3 Deskripsi Use Case Rekam Komponen

| | |
|--|----------------------------|
| <i>UC03</i> – Rekam Komponen | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam komponen. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Rekam | |
| Basic course: 4system menampilkan form rekam komponen. User memilih sub output, mengisikan kode, dan nama komponen, sifat biaya, tahun mulai dan akhir. User menekan simpan | |
| Post-conditions: Sistem mengecek apakah field kosong, input yang dimasukkan adalah angka dan bukan minus serta data komponen dengan klasifikasi sub output dan output belum ada dalam daftar komponen. Sistem menyimpan ke daftar komponen | |
| Alternate courses: <ol style="list-style-type: none"> 1. jika field kode, dan nama komponen, sifat biaya, tahun mulai dan akhir kosong, sistem menampilkan pesan kesalahan. 2. jika field kode, tahun mulai atau akhir dimasukkan bukan angka, maka sistem akan menampilkan peringatan kesalahan 3. jika field kode, tahun mulai atau akhir dimasukkan angka minus, maka sistem akan menampilkan pesan kesalahan 4. jika tahun akhir lebih awal dari tahun awal maka system menampilkan pesan kesalahan. 5. jika output yang dipilih telah ada di daftar output, sistem | |

| |
|-----------------------------|
| menampilkan pesan kesalahan |
|-----------------------------|

B.5. Deskripsi Use Case Rekam Output

Tabel 9.4 Deskripsi Use Case Rekam Output

| | |
|--|----------------------------|
| <i>UC04</i> – Rekam Output | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam output. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Rekam | |
| Basic course: System menampilkan form rekam output. User memilih referensi output, baseline, lokasi, kewenangan, volume, tahun mulai dan henti lalu mengisi volume kpbm . user menekan simpan | |
| Post-conditions: Sistem mengecek apakah field kosong, input yang dimasukkan adalah angka dan bukan minus serta data output telah ada dalam daftar output. Sistem menyimpan ke daftar output | |
| Alternate courses: <ol style="list-style-type: none"> 1. jika field volume kpbm kosong, sistem menampilkan pesan kesalahan. 2. jika field volume kpbm dimasukkan bukan angka, maka sistem akan menampilkan peringatan kesalahan 3. jika field volume kpbm dimasukkan angka minus, maka sistem akan menampilkan pesan kesalahan 4. jika tahun akhir lebih awal dari tahun awal maka system menampilkan pesan kesalahan. 5. jika output yang dipilih telah ada di daftar output, sistem menampilkan pesan kesalahan | |

B.6. Deskripsi Use Case Rekam Sub Komponen

Tabel 9.5 Deskripsi Use Case Rekam Sub Komponen

| | |
|---|----------------------------|
| <i>UC05</i> – Rekam Sub Komponen | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam sub komponen. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Rekam | |
| Basic course: System menampilkan form rekam sub komponen. User memilih komponen, mengisi kode dan uraian sub komponen . user menekan simpan | |
| Post-conditions: Sistem mengecek apakah field kosong, data sub komponen dengan klasifikasi komponen, sub output dan output telah ada dalam daftar sub komponen. Sistem menyimpan ke daftar sub komponen | |
| Alternate courses: <ol style="list-style-type: none"> 1. jika field kode dan nama sub komponen kosong, sistem menampilkan pesan kesalahan. 2. jika kode dan nama sub komponen yang diisi telah ada di daftar sub komponen, sistem menampilkan pesan kesalahan | |

B.7. Deskripsi Use Case Rekam Sub Output

Tabel 9.6 Deskripsi Use Case Rekam Sub Output

| | |
|---|----------------------------|
| <i>UC06</i> – Rekam Sub Output | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam sub output. | |

| |
|--|
| <p>Triggers:</p> <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Rekam |
| <p>Basic course:</p> <p>System menampilkan form rekam sub output. User memilih referensi output, mengisikan kode sub output , nama sub output dan nilai volume . user menekan simpan</p> |
| <p>Post-conditions:</p> <p>Sistem mengecek apakah field kosong, input yang dimasukkan adalah angka dan bukan minus serta data sub output dengan klasifikasi output telah ada dalam daftar sub output. Sistem menyimpan ke daftar output</p> |
| <p>Alternate courses:</p> <ol style="list-style-type: none"> 1. jika field kode, nama sub output dan volume kosong, sistem menampilkan pesan kesalahan. 2. jika field volume atau kode dimasukkan bukan angka, maka sistem akan menampilkan peringatan kesalahan 3. jika field volume atau kode dimasukkan angka minus, maka sistem akan menampilkan pesan kesalahan 4. jika output yang dipilih dan kode sub output telah ada di daftar sub output, sistem menampilkan pesan kesalahan |

B.8. Deskripsi Use Case Ubah Afp

Tabel 9.7 Deskripsi Use Case Ubah Afp

| | |
|--|------------------------------------|
| UC07 – Ubah Afp | |
| <p>Primary Actor: Operator Anggaran</p> | <p>Level: User Goal</p> |
| <p>Pre-conditions: Operator Anggaran berada di halaman ubah afp.</p> | |
| <p>Triggers:</p> <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| <p>Basic course:</p> <p>System menampilkan form ubah afp dari detail belanja. User mengubah nilai kontrak dan non kontrak, rencana penyerapan</p> | |

| |
|--|
| dana dari bulan januari sampai dengan desember. User menekan ubah |
| Post-conditions: Sistem mengecek input yang dimasukkan adalah angka, bukan minus dan perhitungannya. Sistem menyimpan ke daftar afp |
| Alternate courses: 1. jika semua field yang diisi bukan angka, maka system akan menampilkan pesan error 2. jika semua field dimasukkan berupa angka minus, maka system akan menampilkan pesan error 3. jika rincian afp melebihi pagu POK, maka system akan menampilkan pesan error. |

B.9. Deskripsi Use Case Approve Afp

Tabel 9.8 Deskripsi Use Case Approve Afp

| | |
|---|----------------------------|
| UC08 – Approve Afp | |
| Primary Actor: Kuasa Pengguna Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman approve Afp. | |
| Triggers: • Operator Anggaran mengklik tombol Approve | |
| Basic course: System menampilkan list afp. Kpa meneliti rincian afp. Kpa menekan approve, | |
| Post-conditions: System memberi status approval pada afp. | |
| Alternate courses: 1. kpa menekan tolak. Sistem memberi status tolak pada afp. | |

B.10. Deskripsi Use Case Hapus Akun

Tabel 9.9 Deskripsi Use Case Hapus Akun

| | |
|---|----------------------------|
| <i>UC12</i> – Hapus Akun | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman hapus akun. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol hapus | |
| Basic course: sistem menampilkan list POK. User memilih akun yang akan dihapus. User menekan hapus dan sistem menampilkan peringatan jika dihapus maka unsur yang ada di bawah akun akan ikut terhapus. User menekan hapus. | |
| Post-conditions: sistem menghapus akun dari daftar akun, detil dari daftar detil, dan afp dari daftar afp | |
| Alternate courses: - | |

B.11. Deskripsi Hapus Detil

Tabel 9.10 Deskripsi Hapus Detil

| | |
|---|----------------------------|
| <i>UC13</i> – Hapus Detil | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman hapus detil. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol hapus | |
| Basic course: system menampilkan list POK. User memilih detil yang akan dihapus. User menekan hapus system menampilkan peringatan | |

| |
|---|
| penghapusan. |
| Post-conditions: 10system menghapus detil dan afp dari daftar |
| Alternate courses: - |

B.12. Deskripsi Use Case Hapus Komponen

Tabel 9.11 Deskripsi Use Case Hapus Komponen

| | |
|---|----------------------------|
| UC14 – Hapus Komponen | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman hapus komponen. | |
| Triggers: • Operator Anggaran mengklik tombol hapus | |
| Basic course: sistem menampilkan daftar komponen. User memilih komponen yang akan dihapus. User menekan hapus, sistem menampilkan konfirmasi penghapusan . User menekan hapus | |
| Post-conditions: .sistem menghapus komponen dari daftar komponen, sub komponen, akun, detail, dan afp dari daftar | |
| Alternate courses: | |

B.13. Deskripsi Use Case Hapus Output

Tabel 9.12 Deskripsi Use Case Hapus Output

| | |
|---|----------------------------|
| UC15 – Hapus Output | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman hapus output. | |

| |
|---|
| Triggers: |
| <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol hapus |
| Basic course: |
| sistem menampilkan daftar output. User memilih output yang akan dihapus. User menekan hapus, sistem menampilkan konfirmasi penghapusan jika output dihapus maka sub output dan unsur yang berada di dalamnya akan terhapus juga. User menekan hapus |
| Post-conditions: |
| sistem menghapus output, sub output, komponen, sub komponen, , akun, detail dan. Afp dari daftar |
| Alternate courses: |
| - |

B.14. Deskripsi Use Case Hapus Pendapatan

Tabel 9.13 Deskripsi Use Case Hapus Pendapatan

| | |
|--|---------------|
| UC 16 – Hapus Pendapatan | |
| Primary Actor: | Level: |
| Operator Anggaran | User Goal |
| Pre-conditions: | |
| Operator Anggaran berada di halaman hapus pendapatan. | |
| Triggers: | |
| <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol pendapatan. | |
| Basic course: | |
| sistem menampilkan daftar pendapatan. User memilih pendapatan yang akan dihapus. User menekan tombol hapus | |
| Post-conditions: | |
| sistem menghapus pendapatan dari daftar pendapatan | |
| Alternate courses: | |
| - | |

B.15. Deskripsi Use Case Hapus Sub Komponen

Tabel 9.14 Deskripsi Use Case Hapus Sub Komponen

| | |
|--|----------------------------|
| <i>UC 17 – Hapus Sub Komponen</i> | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman hapus sub komponen. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol hapus. | |
| Basic course: sistem menampilkan daftar pok. User memilih sub komponen yang akan dihapus. User menekan hapus, sistem menampilkan konfirmasi penghapusan jika sub komponen dihapus maka akun dan unsur yang berada di bawahnya akan terhapus juga | |
| Post-conditions: sistem menghapus sub komponen, akun, detail, dan afp dari daftar | |
| Alternate courses: - | |

B.16. Deskripsi Use Case Hapus Sub Output

Tabel 9.15 Deskripsi Use Case Hapus Sub Output

| | |
|--|----------------------------|
| <i>UC 18 – Hapus Sub Output</i> | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman hapus sub output. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol hapus. | |
| Basic course: sistem menampilkan daftar POK. User memilih sub output yang akan dihapus. User menekan hapus | |

| |
|---|
| Post-conditions: sistem menampilkan konfirmasi penghapusan jika output dihapus maka sub output dan unsur yang berada di bawahnya akan terhapus juga. Sistem menghapus sub output, komponen, sub komponen, akun, detail, dan afp dari daftar |
| Alternate courses: - |

B.17. Deskripsi Use Case Rekam Akun

Tabel 9.16 Deskripsi Use Case Rekam Akun

| | |
|---|----------------------------|
| UC21 – Rekam Akun | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman rekam akun. | |
| Triggers: • Operator Anggaran mengklik tombol Rekam | |
| Basic course: 13system menampilkan form rekam akun. User memilih akun, kppn pembayaran dan sumber dana. User menekan simpan | |
| Post-conditions: 13system mengecek apakah akun dengan klasifikasi sub komponen, komponen, sub output dan output sudah ada dalam daftar akun. Kemudian 13system menyimpannya ke dalam daftar akun. | |
| Alternate courses: jika akun telah ada dalam daftar akun maka sistem akan menampilkan pesan kesalahan | |

B.18. Deskripsi Use Case Rekam Pendapatan

Tabel 9.17 Deskripsi Use Case Rekam Pendapatan

| | |
|---|----------------------------|
| <i>UC22</i> – Rekam Pendapatan | |
| <i>Primary Actor:</i> Operator Anggaran | <i>Level:</i> User Goal |
| <i>Pre-conditions:</i> Operator Anggaran berada di halaman rekam pendapatan. | |
| <i>Triggers:</i> <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Rekam | |
| <i>Basic course:</i> <p>14system menampilkan form rekam pendapatan. User memilih kegiatan, kode akun dan jenis PNBPN, serta mengisi nilai perencanaan pendapatan selama 4 tahun dari tahun berkenaan. user menekan rekam</p> | |
| <i>Post-conditions:</i> <p>14system melakukan verifikasi field. 14system menyimpan ke daftar pendapatan.</p> | |
| <i>Alternate courses:</i> <ol style="list-style-type: none"> 1. jika nilai pendapatan kosong, maka sistem menampilkan pesan kesalahan 2. jika nilai pendapatan bukan angka, maka sistem akan menampilkan pesan kesalahan 3. jika nilai pendapatan dimasukkan angka minus maka sistem akan menampilkan pesan kesalahan 4. jika kombinas kegiatan dana akun telah ada di daftar pendapatan, maka sistem menampilkan pesan kesalahan | |

B.19. Deskripsi Use Case Tayang DIPA

Tabel 9.18 Deskripsi Use Case Tayang Dipa

| | |
|--|----------------------------|
| <i>UC24</i> – Tayang DIPA | |
| <i>Primary Actor:</i> Operator Anggaran | <i>Level:</i> User Goal |
| <i>Pre-conditions:</i> | |

| |
|--|
| Operator Anggaran berada di halaman tayang dipa. |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol tayang |
| Basic course: sistem menampilkan halaman menu awal. User memilih menu tayang DIPA |
| Post-conditions: Sistem menayangkan DIPA |
| Alternate courses: |

B.20. Deskripsi Use Case Ubah Akun

Tabel 9.19 Deskripsi Use Case Ubah Akun

| | |
|--|----------------------------|
| UC25 – Ubah Akun | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah akun. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: 15system menampilkan form ubah akun. User hanya 15yst mengganti kode kppn dan sumber dana saja. User menekan simpan | |
| Post-conditions: 15system mengecek apakah akun dengan klasifikasi kppn belum ada di daftar akun. Lalu 15system menyimpannya ke daftar akun | |
| Alternate courses: jika akun telah ada dalam daftar akun maka sistem akan menampilkan pesan kesalahan | |

B.21. Deskripsi Use Case Ubah Komponen

Tabel 9.20 Deskripsi Use Case Ubah Komponen

| |
|-----------------------------|
| UC26 – Ubah Komponen |
|-----------------------------|

| | |
|--|----------------------------|
| | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah komponen. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: 16system menampilkan form ubah komponen yang menampilkan field- field yang telah terisi dengan isian yang lama. User hanya 16yst mengubah field uraian komponen, sifat biaya dan memilih ulang tahun awal serta akhir. User menekan simpan | |
| Post-conditions: 16system mengecek apakah ada isian yang masih kosong, kemudian mengubah komponen yang ada di daftar komponen | |
| Alternate courses: jika field uraian kosong, maka sistem menampilkan pesan kesalahan. | |

B.22. Deskripsi Use Case Ubah Output

Tabel 9.21 Deskripsi Use Case Ubah Output

| | |
|--|----------------------------|
| UC27 – Ubah Output | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah output. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: 16system menampilkan form ubah output yang menampilkan field- field yang telah terisi dengan isian yang lama. User hanya 16yst mengubah field volume kpm dan memilih ulang tahun awal serta akhir. User menekan simpan | |

Post-conditions:

Sistem mengecek apakah ada isian kosong, input yang dimasukkan sudah dalam angka, dan bukan minus, kemudian sistem mengubah output yang ada di daftar output

Alternate courses:

1. jika field volume kpmj kosong, maka sistem menampilkan pesan kesalahan.
2. jika field volume dimasukkan bukan angka, maka sistem akan menampilkan pesan kesalahan
2. jika field volume dimasukkan angka minus, maka sistem akan menampilkan pesan kesalahan

B.23. Deskripsi Use Case Ubah Pendapatan

Tabel 9.22 Deskripsi Use Case Ubah Pendapatan

| | |
|---|----------------------------|
| UC28 – Ubah Pendapatan | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah pendapatan. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: 17system menampilkan form ubah pendapatan yang telah terisi record yang lama. User hanya 17yst melakukan penggantian pada jenis PNBPN dan besar target pendapatan. User menekan simpan | |
| Post-conditions: Sistem mengecek apakah ada field kosong, isian adalah angka dan bukan minus, kemudian sistem menyimpannya di daftar pendapatan | |
| Alternate courses: <ol style="list-style-type: none"> 1. jika nilai pendapatan kosong, maka sistem menampilkan pesan kesalahan 2. jika nilai pendapatan bukan angka, maka sistem akan menampilkan pesan kesalahan | |

3. jika nilai pendapatan dimasukkan angka minus maka sistem akan menampilkan pesan kesalahan

B.24. Deskripsi Use Case Ubah Sub Komponen

Tabel 9.23 Deskripsi Use Case Ubah Sub Komponen

| | |
|---|----------------------------|
| <i>UC29</i> – Ubah Sub Komponen | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah sub komponen. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: 18system menampilkan form ubah sub komponen yang menampilkan field- field yang telah terisi dengan isian yang lama. User hanya 18syst mengubah uraian. User menekan simpan | |
| Post-conditions: Sistem mengecek apakah ada isian yang masih kosong, nama sub komponen belum ada di daftar sub komponen, kemudian 18system mengubah sub komponen yang ada di daftar sub komponen. | |
| Alternate courses: 1. jika field uraian kosong, maka sistem menampilkan pesan kesalahan. 2. jika uraian yang diisi telah ada dalam daftar sub komponen, sistem menampilkan pesan kesalahan | |

B.25. Deskripsi Use Case Ubah Sub Output

Tabel 9.24 Deskripsi Use Case Ubah Sub Output

| | |
|--|----------------------------|
| <i>UC30</i> – Ubah Sub Output | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: | |

| |
|---|
| Operator Anggaran berada di halaman ubah sub output. |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah |
| Basic course: 19system menampilkan form ubah sub output yang menampilkan field- field yang telah terisi dengan isian yang lama. User hanya 19yst mengubah field nama sub output dan volume . user menekan simpan |
| Post-conditions: Sistem mengecek apakah masih ada isian kosong, isian bukan angka dan minus, dan nama sub output belum ada di daftar sub output, kemudian 19system mengubah sub output yang ada di daftar sub output. |
| Alternate courses: 1. jika field volume dan nama sub output kosong, maka sistem menampilkan pesan kesalahan 2. jika field volume dimasukkan bukan angka, maka sistem akan menampilkan pesan kesalahan 3. jika field volume dimasukkan angka minus, maka sistem akan menampilkan pesan kesalahan 4. jika uraian sub output telah ada di daftar sub output, sistem menampilkan pesan kesalahan |

B.26. Deskripsi Use Case Ubah Detil

Tabel 9.25

Deskripsi Use Case Ubah Detil

| | |
|--|----------------------------|
| UC31 – Ubah Detil | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah detil. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: | |

20system menampilkan form ubah detil yang menampilkan field-field yang telah terisi dengan isian yang lama. User mengubah field volume kegiatan, satuan volume, harga satuan volume, dan mengisi kode blokir jika ada blokir. User menekan simpan

Post-conditions:

Sistem mengecek apakah masih ada isian yang kosong, isian bukan angka atau minus serta nama detil telah ada di pok, kemudian 20system mengubah detil yang ada di daftar detil

Alternate courses:

1. jika field header, volume, satuan volume, dan harga satuan kosong, sistem menampilkan pesan kesalahan.
2. jika field volume atau harga dimasukkan bukan angka, maka sistem akan menampilkan peringatan kesalahan
3. jika field volume atau harga dimasukkan angka minus, maka sistem akan menampilkan pesan kesalahan
4. jika uraian detil telah ada di daftar detil, sistem menampilkan pesan kesalahan

B.27. Deskripsi Use Case Cetak Dipa

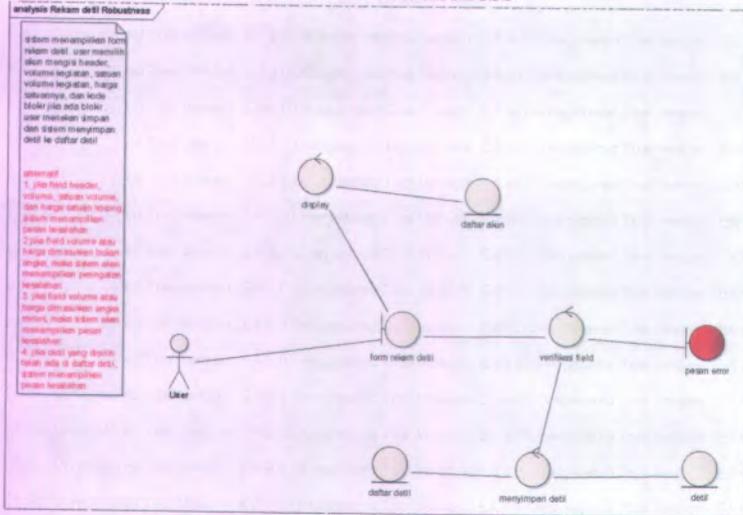
Tabel 9.26

Deskripsi Use Case Cetak Dipa

| | |
|---|----------------------------|
| UC31 – Ubah Detil | |
| Primary Actor: Operator Anggaran | Level: User Goal |
| Pre-conditions: Operator Anggaran berada di halaman ubah detil. | |
| Triggers: <ul style="list-style-type: none"> • Operator Anggaran mengklik tombol Ubah | |
| Basic course: 20system menampilkan halaman Dipa, user memilih halaman Dipa mana yang akan dicetak. User menekan cetak | |
| Post-conditions: Sistem mencetak Dipa | |
| Alternate courses: | |

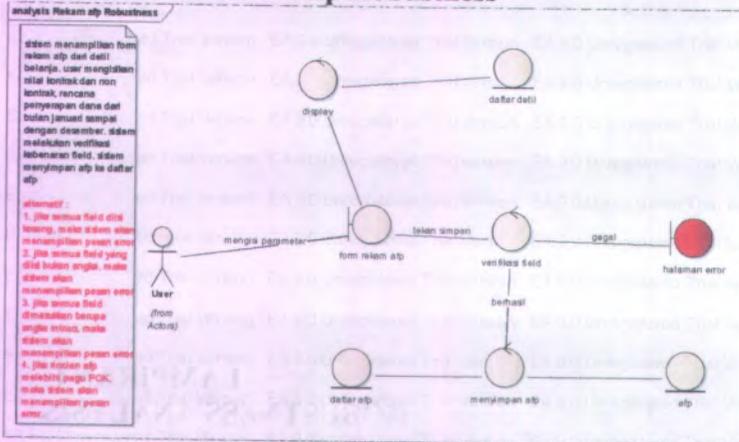
LAMPIRAN C
ROBUSTNESS ANALYSIS

C.1. Rekam Detil Robustness



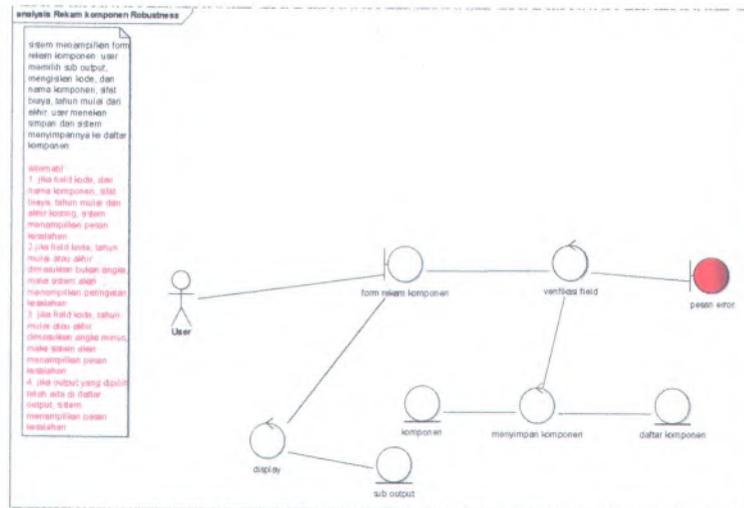
Gambar 10.1 Rekam Detil Robustness

C.2. Rekam Afp Robustness



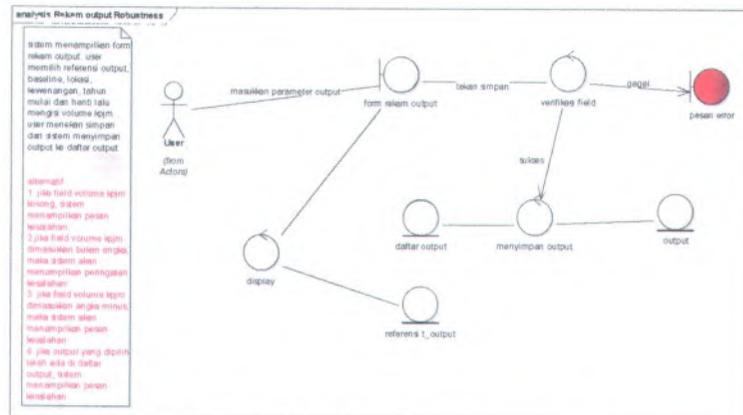
Gambar 10.2 Rekam Afp Robustness

C.3. Rekam Komponen Robustness



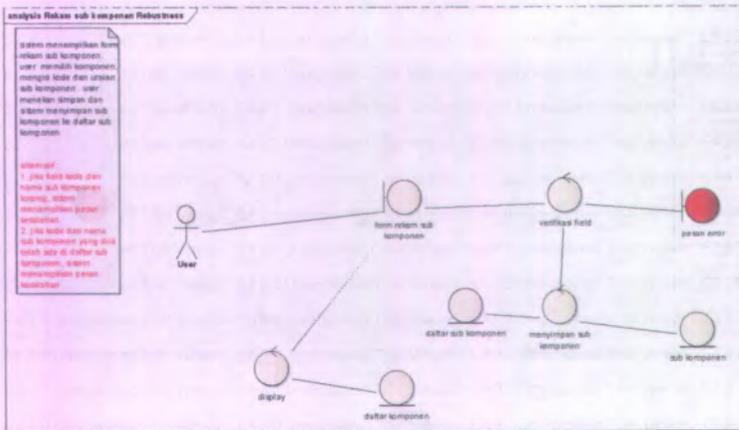
Gambar 10.3 Rekam Komponen Robustness

C.4. Rekam Output Robustness



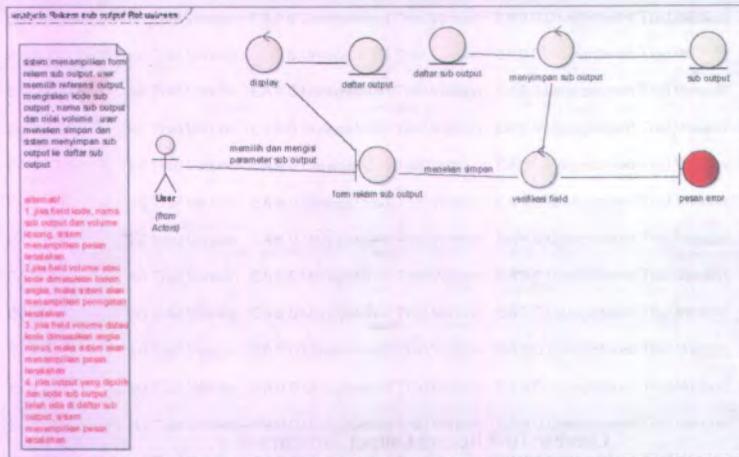
Gambar 10.4 Rekam Output Robustness

C.5. Rekam Sub Komponen Robustness



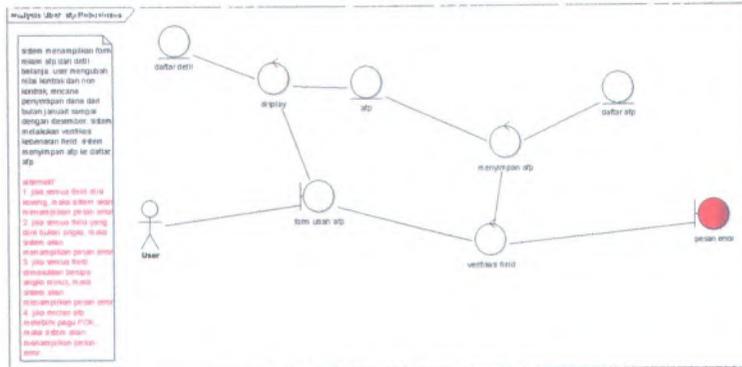
Gambar 10.5 Rekam Sub Komponen Robustness

C.6. Rekam Sub Output Robustness



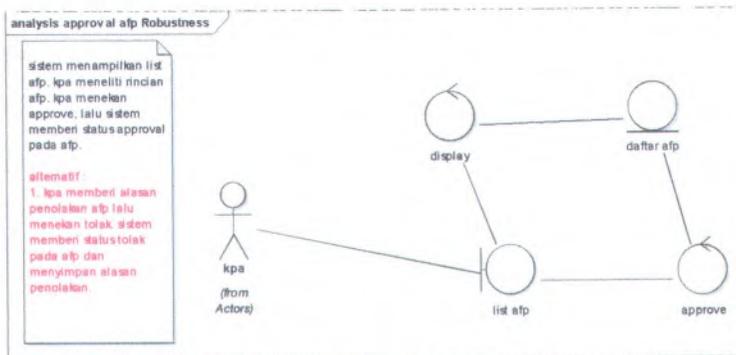
Gambar 10.6 Rekam Sub Output Robustness

C.7. Ubah Afp Robustness



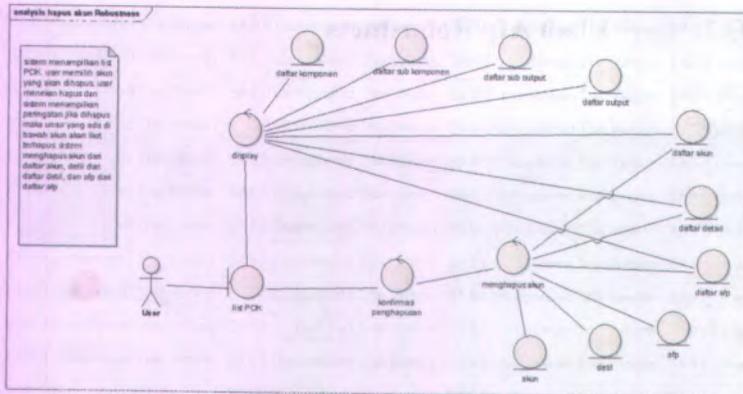
Gambar 10.7 Ubah Afp Robustness

C.8. Approval Afp Robustness



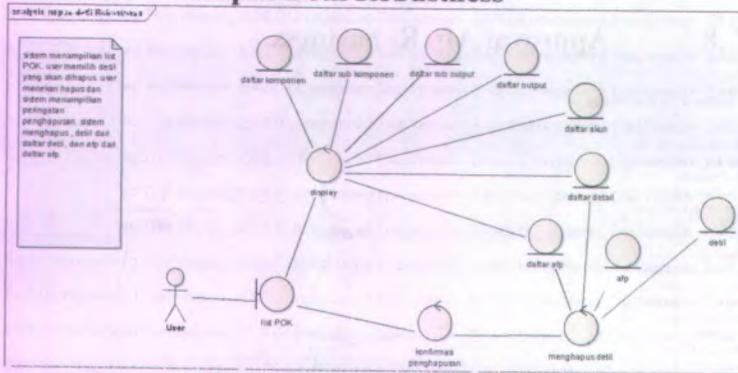
Gambar 10.8 Approval Afp Robustness

C.9. Hapus Akun Robustness



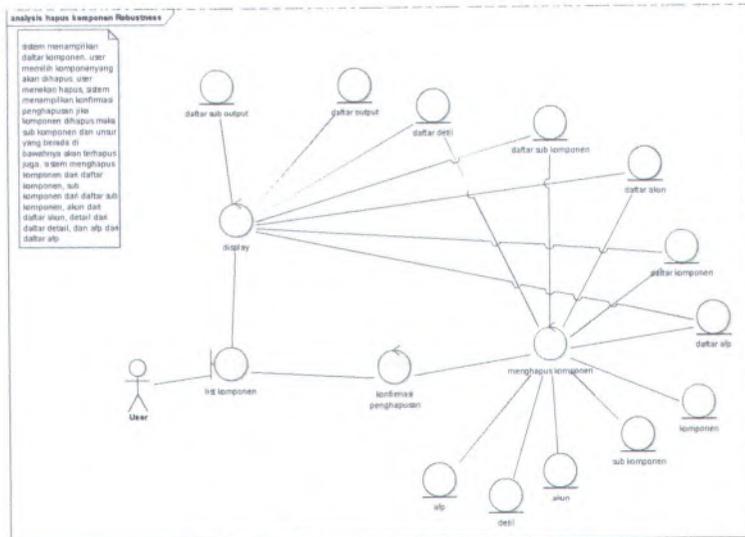
Gambar 10.9 Hapus Akun Robustness

C.10. Hapus Detil Robustness



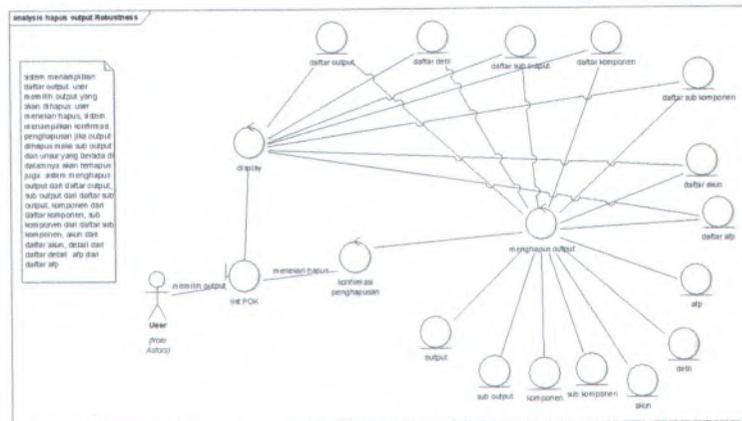
Gambar 10.10 Hapus Detil Robustness

C.11. Hapus Komponen Robustness



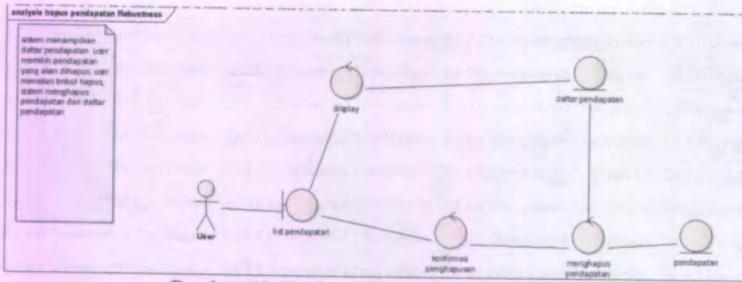
Gambar 10.11 Hapus Komponen Robustness

C.12. Hapus Output Robustness



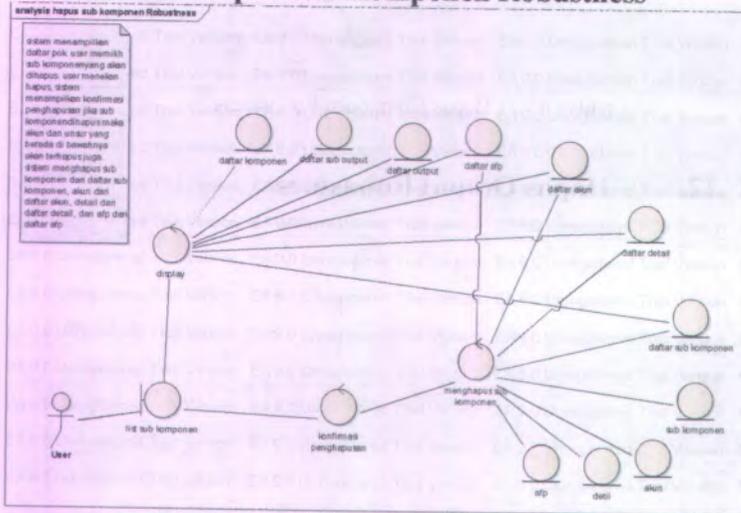
Gambar 10.12 Hapus Output Robustness

C.13. Hapus Pendapatan Robustness



Gambar 10.13 Hapus Pendapatan Robustness

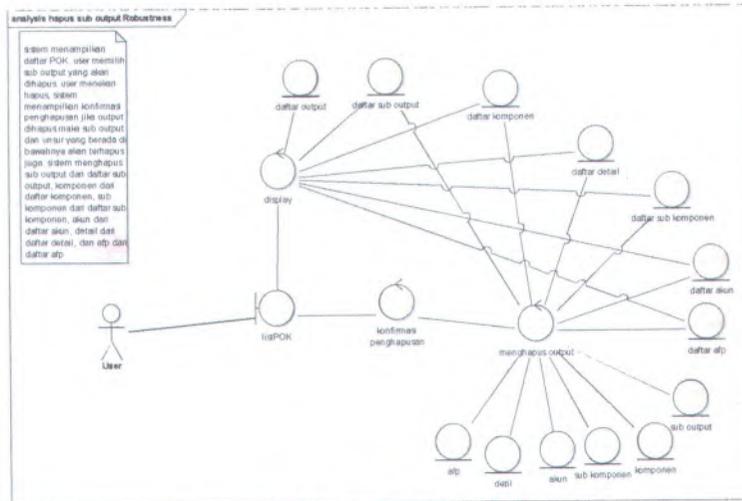
C.14. Hapus Sub Komponen Robustness



Gambar 10.14 Hapus Sub Komponen Robustness

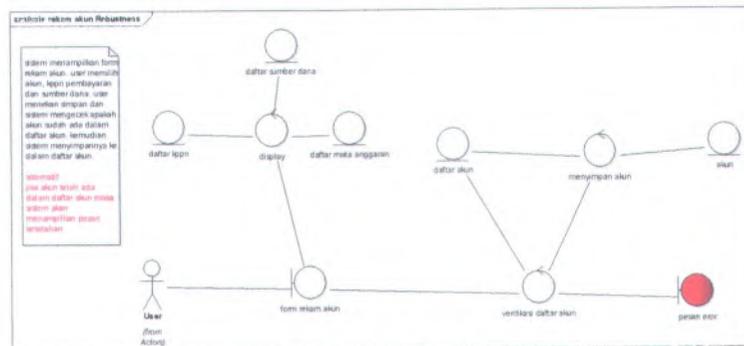
C.15. Hapus Sub Output Robustness

Handwritten notes and a blue stamp are present at the bottom left of the page. The stamp contains some illegible text, possibly a date or a signature.



Gambar 10.15 Hapus Sub Output Robustness

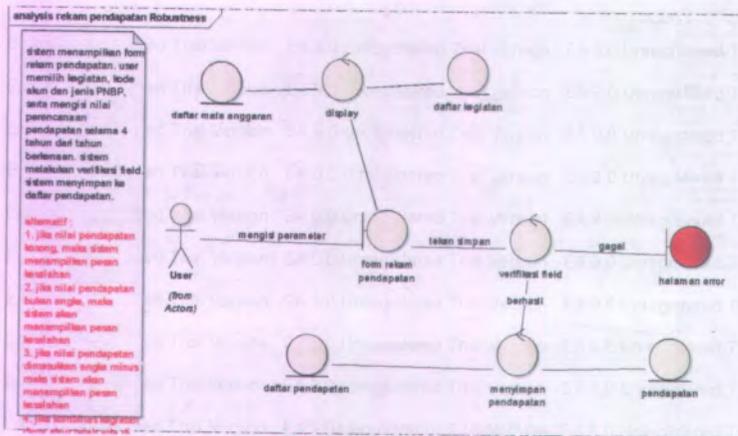
C.16. Rekam Akun Robustness



Gambar 10.16 Rekam Akun Robustness

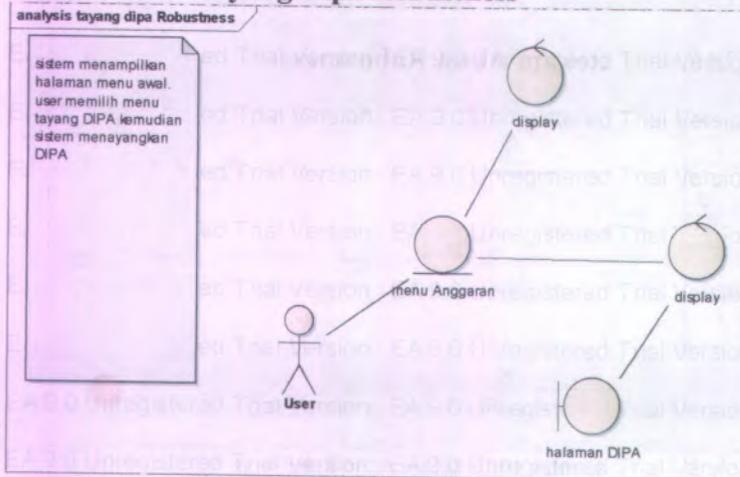
C.17. Rekam Pendapatan Robustness





Gambar 10.17 Rekam Pendapatan Robustness

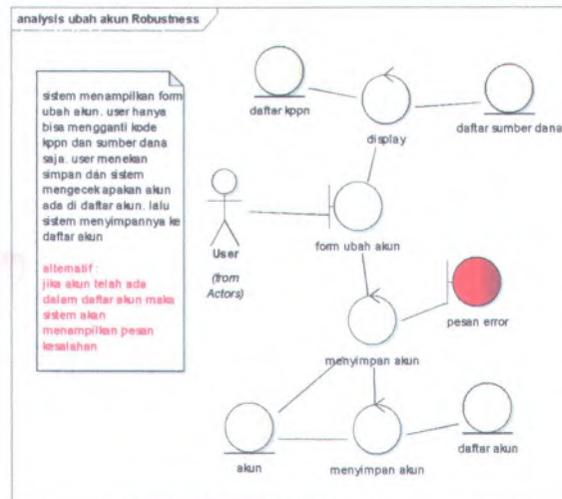
C.18. Tayang Dipa Robustness



Gambar 10.18 Tayang Dipa Robustness

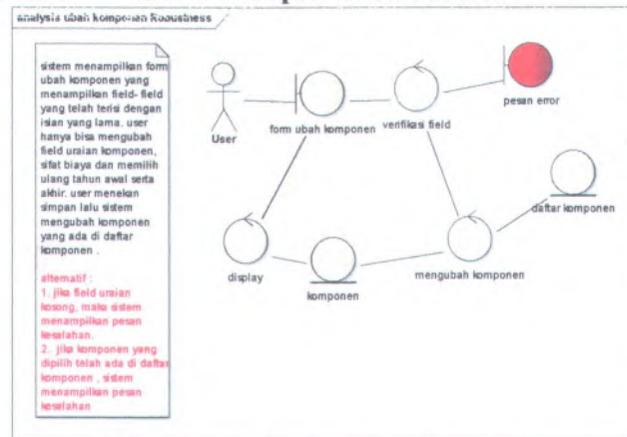
C.19. Ubah Akun Robustness

STAMPED AREA
27



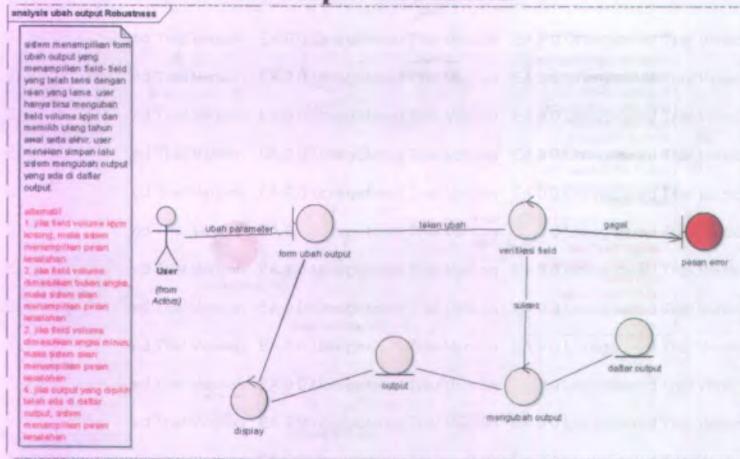
Gambar 10.19 Ubah Akun Robustness

C.20. Ubah Komponen Robustness



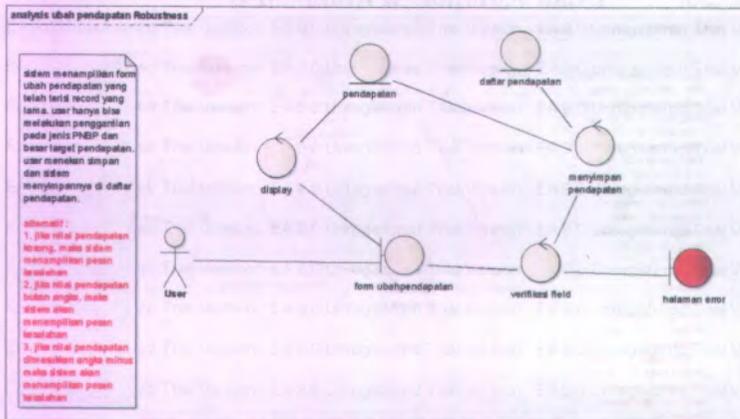
Gambar 10.20 Ubah Komponen Robustness

C.21. Ubah Output Robustness



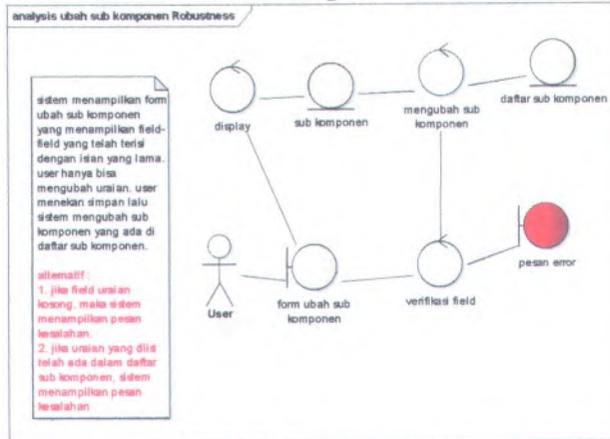
Gambar 10.21 Ubah Output Robustness

C.22. Ubah Pendapatan Robustness



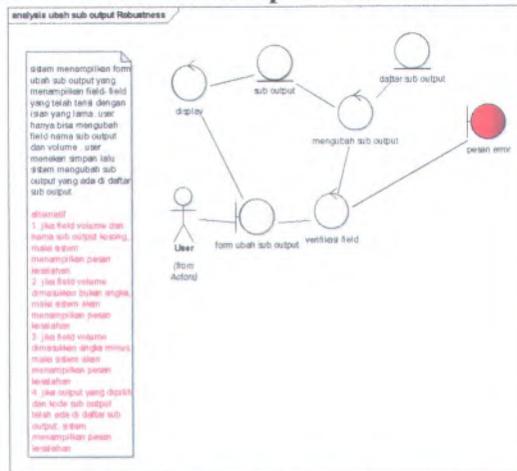
Gambar 10.22 Ubah Pendapatan Robustness

C.23. Ubah Sub Komponen Robustness



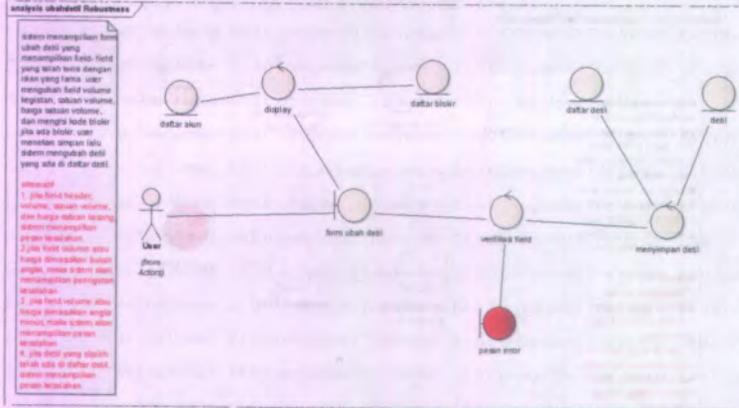
Gambar 10.23 Ubah Sub Komponen Robustness

C.24. Ubah Sub Output Robustness



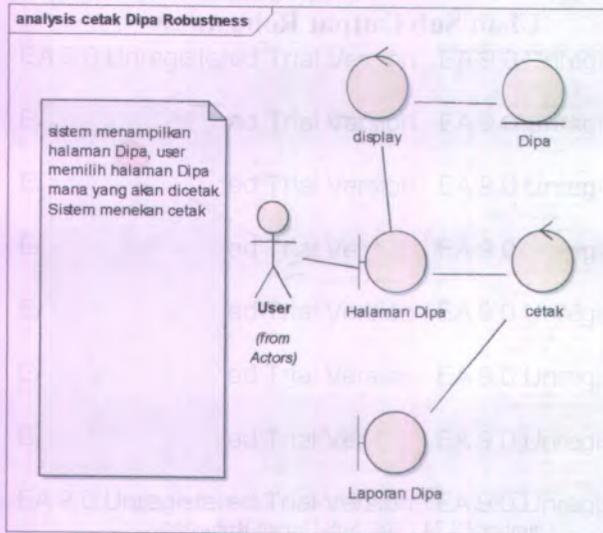
Gambar 10.24 Ubah Sub Output Robustness

C.25. Ubah Detil Robustness



Gambar 10.25 Ubah Detil Robustness

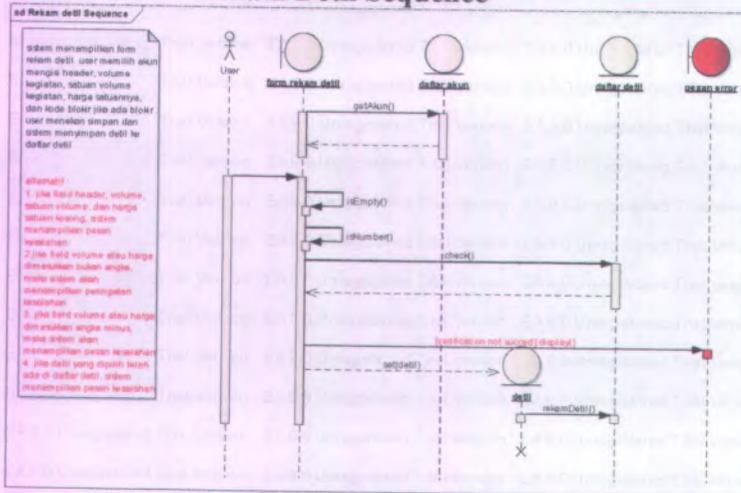
C.26. Cetak Robustness



Gambar 10.26 Cetak Dipa Robustness

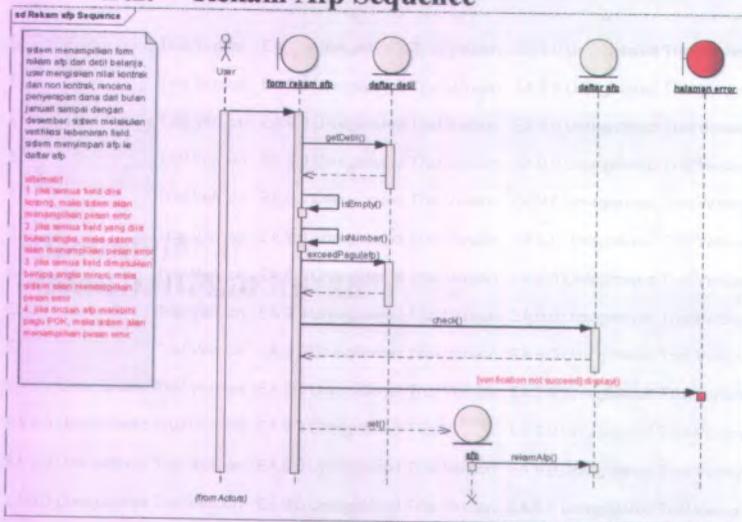
LANE BRAND
SEQUENCE DIAGRAM

D.1. Rekam Detil Sequence



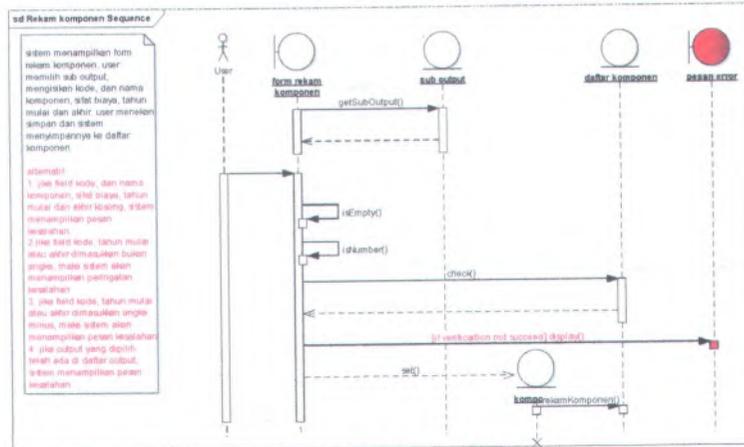
Gambar 11.1 Rekam Detil Sequence

D.2. Rekam Afp Sequence



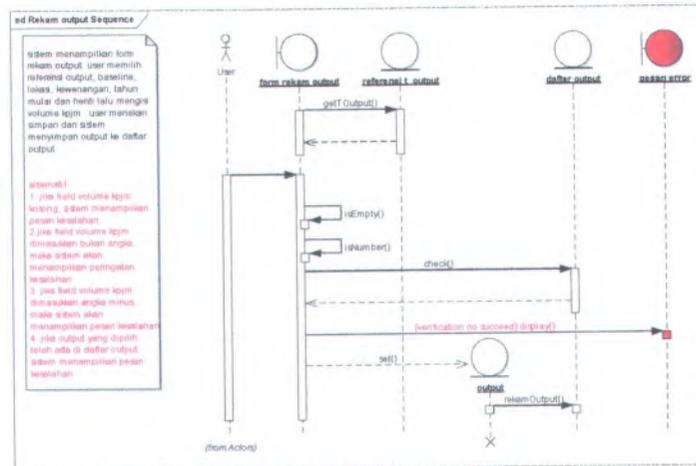
Gambar 11.2 Rekam Afp Sequence

D.3. Rekam Komponen Sequence



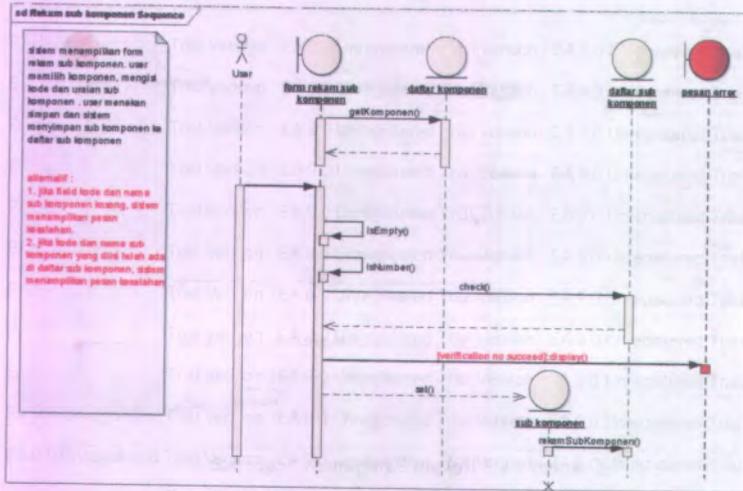
Gambar 11.3 Rekam Komponen Sequence

D.4. Rekam Output Sequence



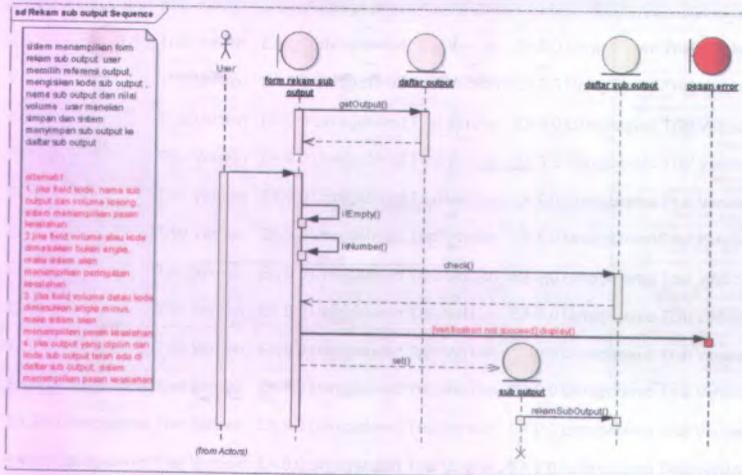
Gambar 11.4 Rekam Output Sequence

D.5. Rekam Sub Komponen Sequence



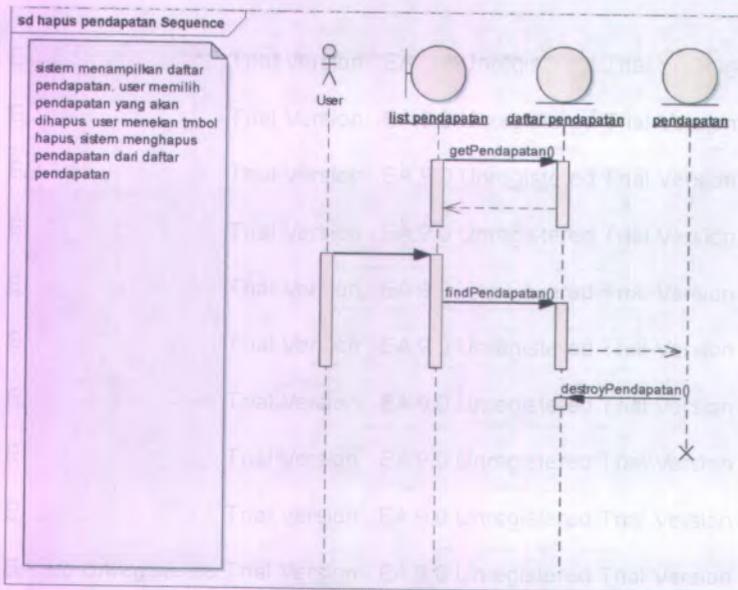
Gambar 11.5 Rekam Sub Komponen Sequence

D.6. Rekam Sub Output Sequence



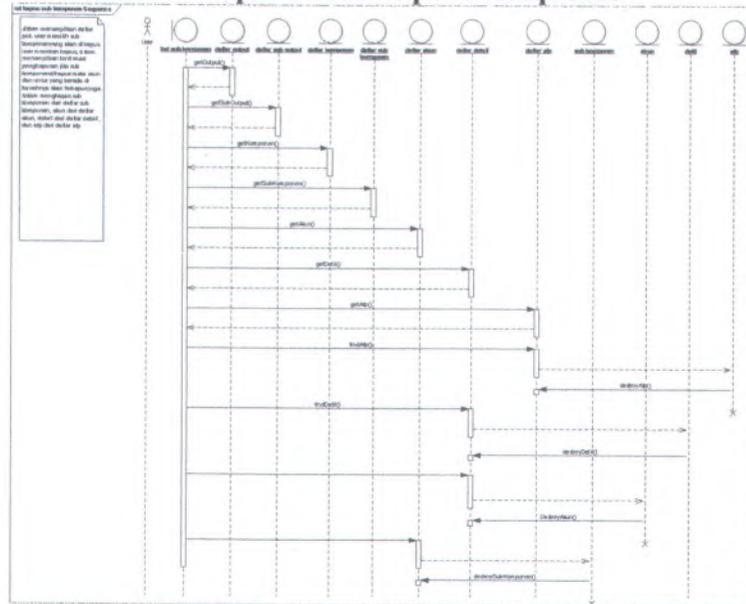
Gambar 11.6 Rekam Sub Output Sequence

D.13. Hapus Pendapatan Sequence



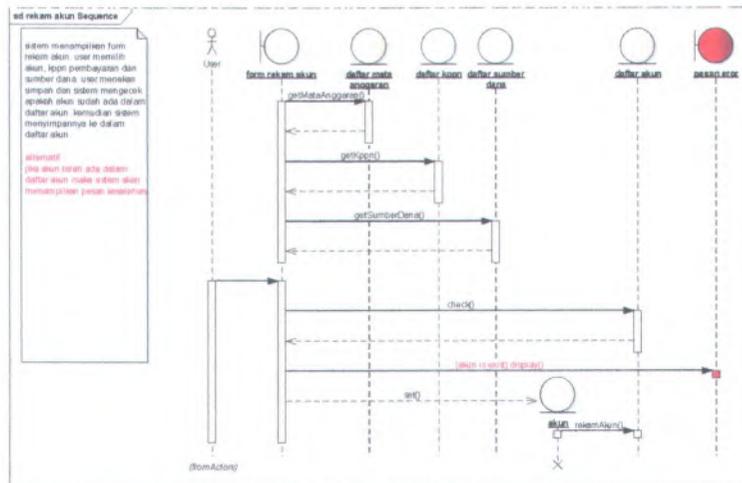
Gambar 11.13 Hapus Pendapatan Sequence

D.14. Hapus Sub Komponen Sequence



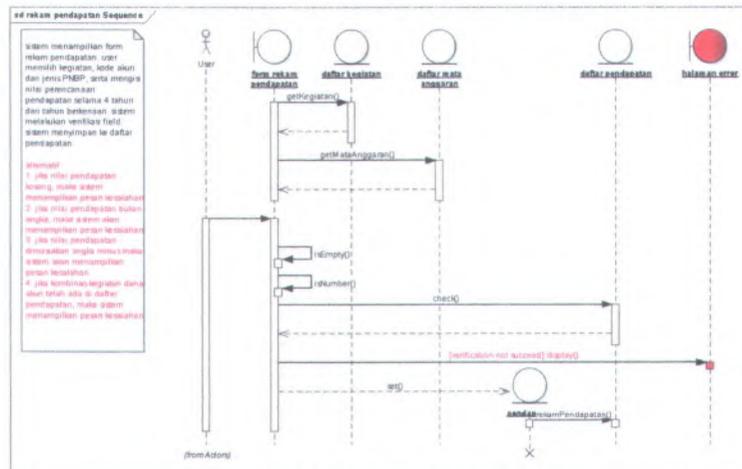
Gambar 11.14 Hapus Sub Komponen Sequence

D.16. Rekam Akun Sequence



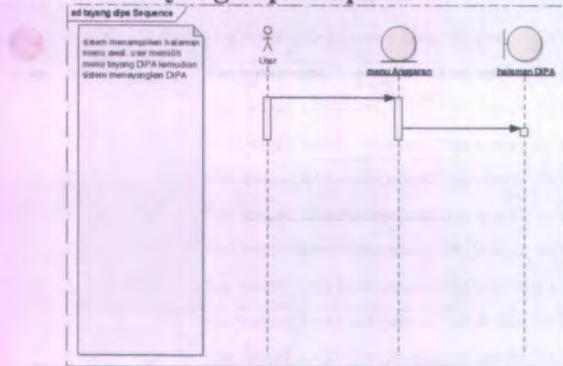
Gambar 11.16 Rekam Akun Sequence

D.17. Rekam Pendapatan Sequence



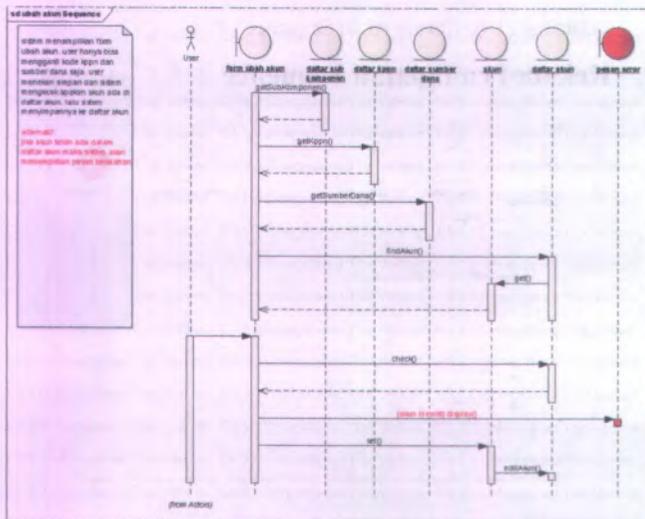
Gambar 11.17 Rekam Pendapatan Sequence

D.18. Tayang Dipa Sequence



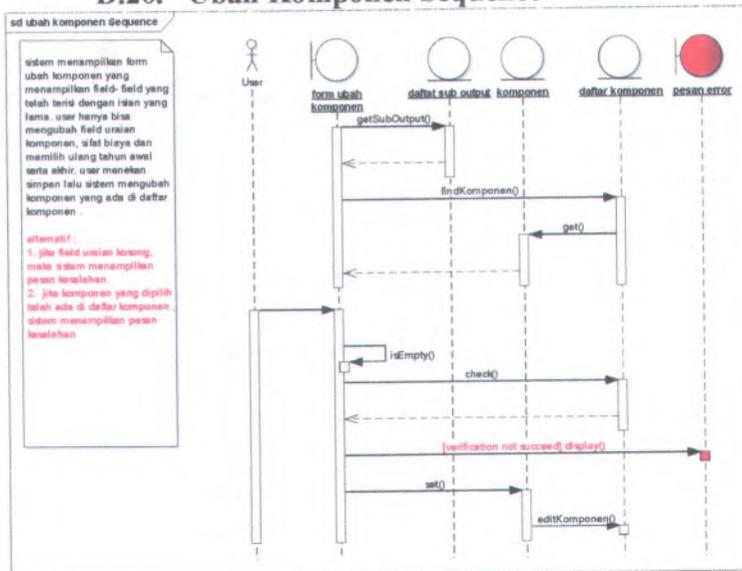
Gambar 11.18 Tayang Dipa Sequence

D.19. Ubah Akun Sequence



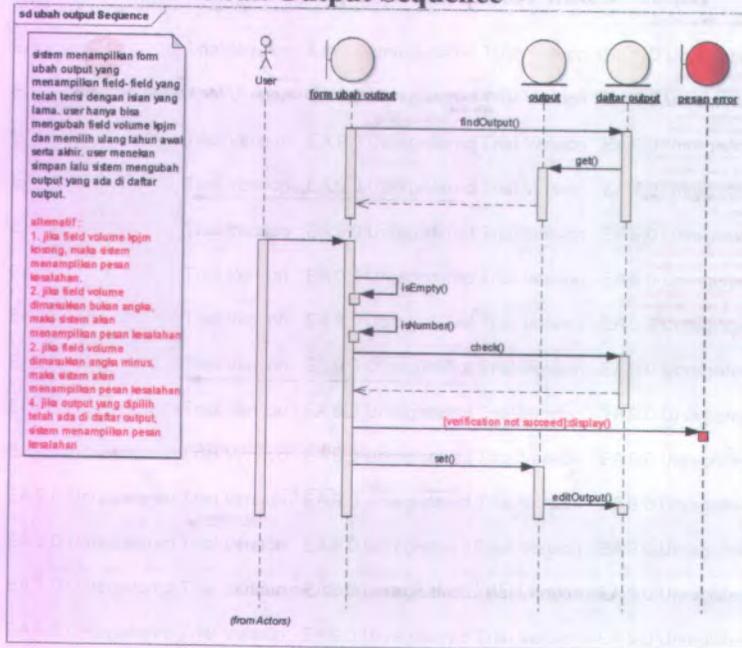
Gambar 11.19 Ubah Akun Sequence

D.20. Ubah Komponen Sequence



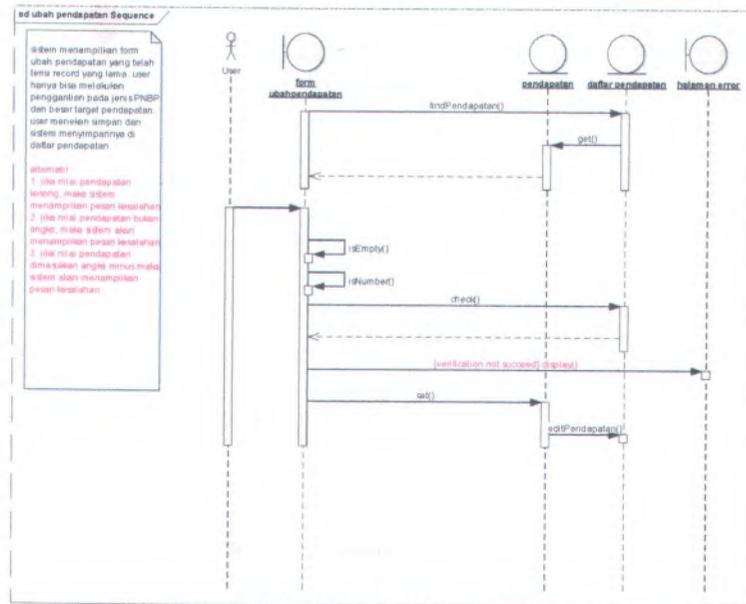
Gambar 11.20 Ubah Komponen Sequence

D.21. Ubah Output Sequence



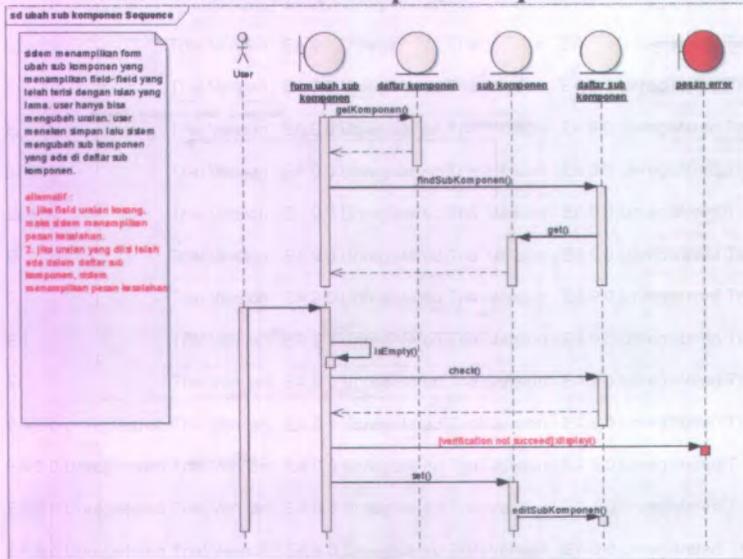
Gambar 11.21 Ubah Output Sequence

D.22. Ubah Pendapatan Sequence



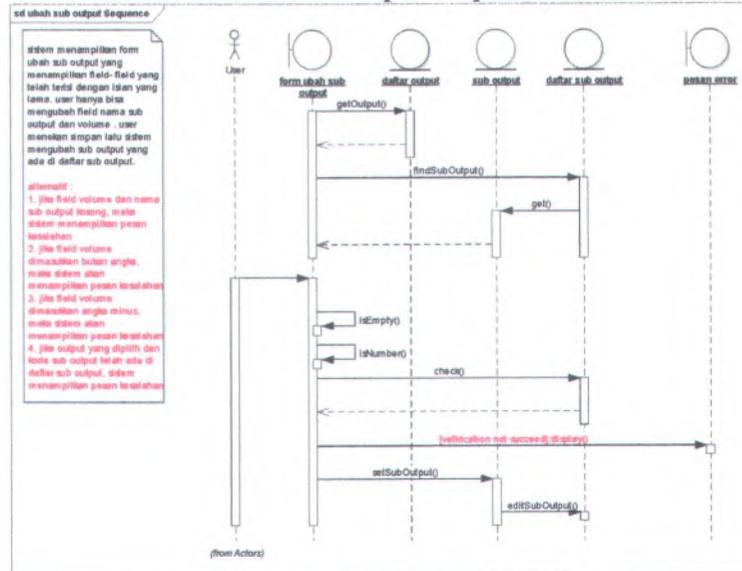
Gambar 11.22 Ubah Pendapatan Sequence

D.23. Ubah Sub Komponen Sequence



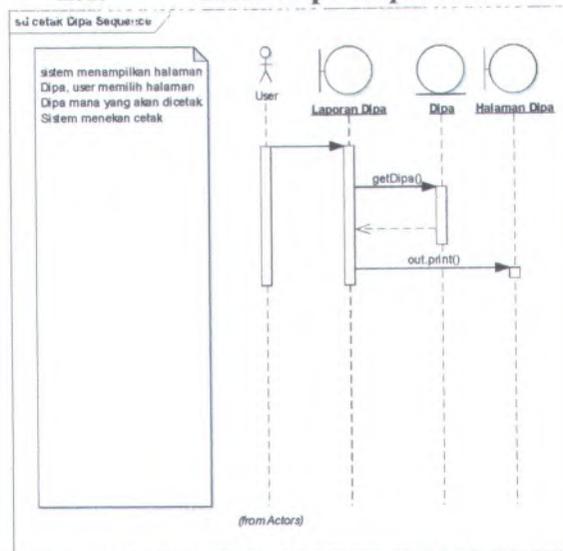
Gambar 11.23 Ubah Sub Komponen Sequence

D.24. Ubah Sub Output Sequence



Gambar 11.24 Ubah Sub Output Sequence

E.1. Cetak Dipa Sequence



Gambar 11.26 Cetak Dipa Sequence

LAMPIRAN E
TEST CASE

E-2

E.1. Test Case 1: Rekam / Ubah output

Deskripsi

Proses merekam atau mengubah output pada RKA-KL

Kondisi awal

Database kosong

Skenario

| Test Case | | | | | | |
|-----------|---|---|-----------------------|-----------------------------------|----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | Output,tahun 2012 dan 2013, volume : 5, 5, 5, 5 | Simpan data berhasil | | sukses | - |
| 2 | Khusus field tahun awal dikosongkan | Kosong | Pesan error | | sukses | - |
| 3 | Khusus field tahun akhir dikosongkan | Kosong | Pesan error | | sukses | - |
| 4 | Khusus masing-masing field volume kpm dikosongkan | Kosong | Pesan error | | sukses | - |
| 5 | Masukkan kombinasi field yang sama (record yang | Data sama dengan record | Pesan error | | sukses | - |

| | | | | | | |
|---|--|---|---|--|-------------------------|-----------------|
| | sama) | sebelumnya | | | | |
| 6 | Field tahun awal, tahun akhir , masing-masing volume kpm dimasukkan karakter bukan angka | A, a, -, 10a, “ “, “,” | Pesan error | | sukses | - |
| 7 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data dalam database) | | sukses | |
| 8 | Mengecek tahun akhir harus lebih besar daripada tahun awal | Tahun akhir dimasukkan lebih kecil dari pada tahun awal | Pesan error | | sukses | |
| 9 | Matikan fitur javascript browser, lakukan 7 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | - |
| | | | | | Test Case Status | <i>Berhasil</i> |

E.2. Test Case 2: Rekam / Ubah Sub Output

Deskripsi

Proses merekam atau mengubah sub output RKA-KL

Kondisi awal

Database kosong

Skenario

| Test Case | | | | | | |
|-------------------------|--|---------------------------------------|---|-----------------------------------|-----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | Output,002, sub output 002, 1000 | Simpan data berhasil | | sukses | - |
| 2 | Salah satu field dikosongkan | Misal : field sub output kosong | Pesan error | | sukses | - |
| 3 | Masukkan kombinasi field yang sama (record yang sama) | Data sama dengan record sebelumnya | Pesan error | | sukses | - |
| 4 | Field sub output, volume dimasukkan karakter bukan angka | A, a, -, 10a, " ", ":", | Pesan error | | sukses | |
| 5 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data dalam database) | | sukses | |
| 6 | Matikan fitur javascript browser, lakukan 5 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.3. Test Case 3: Rekam / Ubah Komponen

Deskripsi

Proses merekam atau mengubah komponen RKA-KL

Kondisi awal

Database kosong

Skenario

| Test Case | | | | | | |
|-----------|--|------------------------------------|-----------------------|-----------------------------------|----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | Output,002, sub output 002, 1000 | Simpan data berhasil | | sukses | - |
| 2 | Salah satu field dikosongkan | Misal : field uraian kosong | Pesan error | | sukses | - |
| 3 | Masukkan kombinasi field yang sama (record yang sama) | Data sama dengan record sebelumnya | Pesan error | | sukses | - |
| 4 | Field komponen, tahun mulai dan berhenti dimasukkan karakter bukan angka | A, a, -, 10a, "", ",", | Pesan error | | sukses | |

E-6

| | | | | | | |
|---|--|---------------------------------------|---|--|-------------------------|-----------------|
| 5 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data dalam database) | | sukses | |
| 6 | Matikan fitur javascript browser, lakukan 5 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| | | | | | Test Case Status | <i>Berhasil</i> |

E.4. Test Case 4: Rekam / Ubah Sub Komponen

Deskripsi

Proses merekam atau mengubah sub komponen RKA-KL

Kondisi awal

Database kosong

| Test Case | | | | | | |
|-----------|------------------------------|-------------------------------|-----------------------|-----------------------------------|----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | komponen, AA, sub komponen AA | Simpan data berhasil | | sukses | - |
| 2 | Salah satu field dikosongkan | Misal : field sub komponen | Pesan error | | sukses | - |

| | | | | | | |
|-------------------------|--|---------------------------------------|---|--|-----------------|---|
| | | kosong | | | | |
| 3 | Masukkan kombinasi field yang sama (record yang sama) | Data sama dengan record sebelumnya | Pesan error | | sukses | - |
| 4 | Field sub komponen diisi dengan angka dan karakter khusus | 1, 02, 33, “ “, “-”, “&&” | Pesan error | | sukses | |
| 5 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data dalam database) | | sukses | |
| 6 | Matikan fitur javascript browser, lakukan 5 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.5. Test Case 5: Rekam / Ubah akun

Deskripsi

Proses merekam atau mengubah akun RKA-KL

Kondisi awal

Database awal kosong

E-8

Skenario

| Test Case | | | | | | |
|-----------|--|---------------------------------------|---|-----------------------------------|-----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | 521111,038,RM | Simpan data berhasil | | sukses | - |
| 2 | Masukkan kombinasi field yang sama (record yang sama) | Data sama dengan record sebelumnya | Pesan error | | sukses | - |
| 3 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data dalam database) | | sukses | |
| 4 | Matikan fitur javascript browser, lakukan 3 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| | | | | Test Case Status | <i>Berhasil</i> | |

E.6. Test Case 6: Rekam / Ubah detail

Deskripsi

Proses rekam atau mengubah detail rincian belanja RKA-KL

Kondisi awal

Database awal kosong

Skenario

| Test Case | | | | | | |
|-----------|---|--|--|-----------------------------------|----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | Volkeg = 100, satker = oh, harga satuan 1000 | Simpan data berhasil | | sukses | - |
| 2 | Salah satu field dikosongkan | Misal : field volkeg kosong | Pesan error | | sukses | - |
| 3 | Masukkan kombinasi field yang sama (record yang sama) | Data sama dengan record sebelumnya | Pesan error | | sukses | - |
| 4 | Field volkeg, satkeg, vol1, vol2, vol3, vol4, sat1, sat2, sat3, sat4, harga dimasukkan karakter bukan angka | A, a, -, 10a, " ", " ; " | Pesan error | | sukses | |
| 5 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data) | | sukses | |

E-10

| | | | | | | |
|---|--|------|--------------------------------------|------------------|----------|--|
| | | | dalam database) | | | |
| 6 | Matikan fitur javascript browser, lakukan 5 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| | | | | Test Case Status | Berhasil | |

E.7. Test Case 7: Rekam / Ubah AFP

Deskripsi

Proses merekam atau mengubah AFP (Aplikasi forecasting penyerapan dana)

Kondisi awal

Database kosong

Skenario

| Test Case | | | | | | |
|-----------|-----------------------|---|-----------------------|-----------------------------------|----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Isi semua field | Volkeg = 100, satker = oh, harga satuan | Simpan data berhasil | | sukses | - |

| | | | | | | |
|---|---|---------------------------------------|---|--|-------------------------|-----------------|
| | | 1000 | | | | |
| 2 | Field volkeg, satkeg, vol1, vol2, vol3, vol4, sat1, sat2, sat3, sat4, harga dimasukkan karakter bukan angka | A, a, -, 10a, " ", “,” | Pesan error | | sukses | |
| 3 | Khusus ubah, field awal jangan dirubah dulu, lalu tekan simpan | Data lama di form ubah jangan diganti | Proses simpan sukses (tidak mengubah data dalam database) | | sukses | |
| 4 | Matikan fitur javascript browser, lakukan 3 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| | | | | | Test Case Status | <i>Berhasil</i> |

E.8. Test Case 8: Hapus Output

Deskripsi

Proses menghapus data Output

Kondisi awal

Database awal output telah terisi record output

Skenario

| |
|------------------|
| Test Case |
|------------------|

| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
|-------------------------|--|------|--------------------------------------|-----------------------------------|-----------------|-------------------------|
| 1 | Hapus output | - | Output sampai dengan detil terhapus | | sukses | - |
| 2 | Pada kotak dialog konfirmasi hapus, tekan no (tidak jadi menghapus) | - | Data tidak terhapus | | sukses | - |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.9. Test Case 9: Hapus Sub Output

Deskripsi

Proses menghapus sub Output RKA-KL

Kondisi awal

Database awal data sub output

Skenario

Test Case

| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
|----|--|------|--|-----------------------------------|-------------------------|-------------------------|
| 1 | Hapus sub output | - | Sub Output sampai dengan detail terhapus | | sukses | - |
| 2 | Pada kotak dialog konfirmasi hapus, tekan no (tidak jadi menghapus) | - | Data tidak terhapus | | sukses | - |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| | | | | | Test Case Status | <i>Berhasil</i> |

E.10. Test Case 10: hapus Komponen

Deskripsi

Proses menghapus Komponen RKA-KL

Kondisi awal

Database awal telah terisi

Skenario

| |
|------------------|
| Test Case |
|------------------|

E-14

| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
|-------------------------|--|------|---------------------------------------|-----------------------------------|-----------------|-------------------------|
| 1 | Hapus komponen | - | komponen sampai dengan detil terhapus | | sukses | - |
| 2 | Pada kotak dialog konfirmasi hapus, tekan no (tidak jadi menghapus) | - | Data tidak terhapus | | sukses | - |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.11. Test Case 11: Hapus Sub Komponen

Deskripsi

Proses menghapus sub komponen RKA-KL

Kondisi awal

Database awal telah terisi

Skenario

| Test Case | | | | | | |
|-------------------------|--|------|---|-----------------------------------|-----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Hapus sub komponen | - | Sub komponen sampai dengan detil terhapus | | sukses | - |
| 2 | Pada kotak dialog konfirmasi hapus, tekan no (tidak jadi menghapus) | - | Data tidak terhapus | | sukses | - |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.12. Test Case 12: Hapus Akun

Deskripsi

Proses menghapus Akun RKA-KL

Kondisi awal

Database awal telah terisi

Skenario

| Test Case |
|-----------|
|-----------|

E-16

| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
|-------------------------|--|------|--------------------------------------|-----------------------------------|-----------------|-------------------------|
| 1 | Hapus akun | - | Akun sampai dengan detil terhapus | | sukses | - |
| 2 | Pada kotak dialog konfirmasi hapus, tekan no (tidak jadi menghapus) | - | Data tidak terhapus | | sukses | - |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.13. Test Case 13: Hapus Detil

Deskripsi

Proses menghapus detil RKA-KL

Kondisi awal

Database awal telah terisi

Skenario

| Test Case | | | | | | |
|-----------|--|------|--------------------------------------|-----------------------------------|------------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Hapus detil | - | detil terhapus | | sukses | - |
| 2 | Pada kotak dialog konfirmasi hapus, tekan no (tidak jadi menghapus) | - | Data tidak terhapus | | sukses | - |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| | | | | | Test Case Status | Berhasil |

E.14. Test Case 14: Approval AFP

Deskripsi

Proses mengesahkan AFP

Kondisi awal

Database awal telah terisi

Skenario

| Test Case | | | | | | |
|-----------|-----------------------|------|-----------------------|---------------------------|----------------|------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai) | Sukses / Gagal | Log Number |

| | | | | harapan) | | (jika gagal) |
|-------------------------|---|--------|--------------------------------------|----------|-----------------|--------------|
| 1 | Mengesahkan AFP | AFP | Status AFP tersahkan | | sukses | - |
| 2 | Pada kotak dialog konfirmasi approve, tekan no (tidak jadi mengesahkan) | AFP | AFP tidak tersahkan | | sukses | - |
| 3 | Pada saat AFP kosong (belum direkam), tekan approve | Kosong | Pesan error, data tidak ada data | | sukses | |
| 4 | Matikan fitur javascript browser, lakukan 3 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

E.15. Test Case 15: Tayang Dipa

Deskripsi

Proses menayangkan DIPA

Kondisi awal

Database awal telah terisi

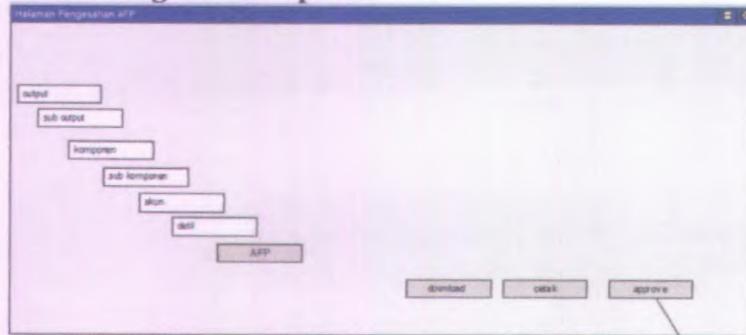
Skenario

| Test Case | | | | | | |
|-------------------------|--|--------|--------------------------------------|-----------------------------------|-----------------|-------------------------|
| No | Deskripsi Langkah Tes | Data | Hasil yang Diharapkan | Fakta (jika tidak sesuai harapan) | Sukses / Gagal | Log Number (jika gagal) |
| 1 | Menayangkan DIPA (skenario normal) | DIPA | DIPA tertayang dengan data akurat | | sukses | - |
| 2 | Pada saat POK kosong (belum direkam), tekan tayang | Kosong | Pesan error, data tidak ada data | | sukses | |
| 3 | Matikan fitur javascript browser, lakukan 2 test di atas sekali lagi | idem | Sesuai expected result masing-masing | | sukses | |
| Test Case Status | | | | | <i>Berhasil</i> | |

LAMPIRAN F
GUI STORYBOARD

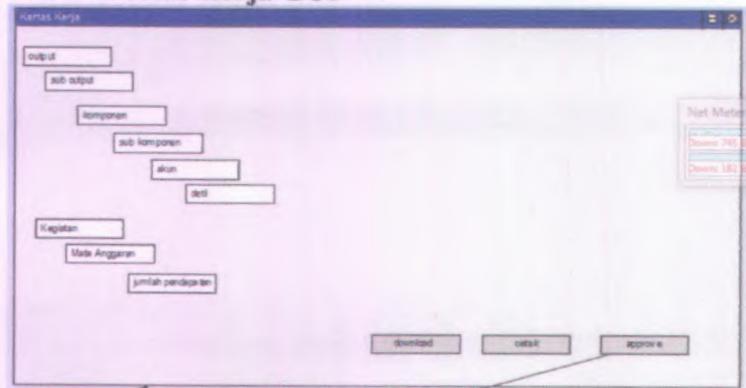
F-1

F.1. Pengesahan Afp GUI



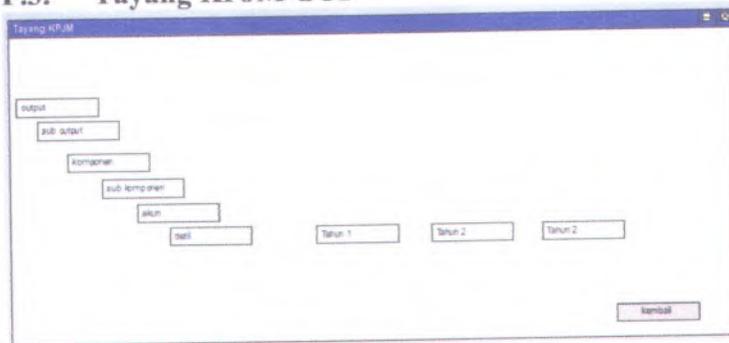
Gambar 13.1 Pengesahan Afp GUI

F.2. Kertas Kerja GUI



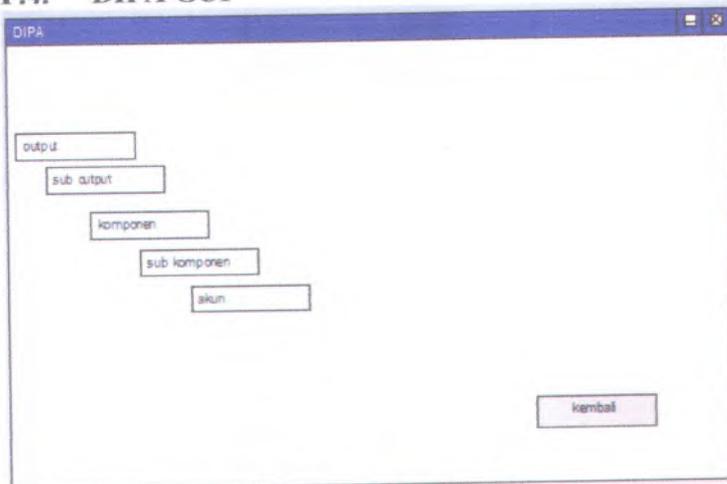
Gambar 13.2 Kertas Kerja GUI

F.3. Tayang KPJM GUI



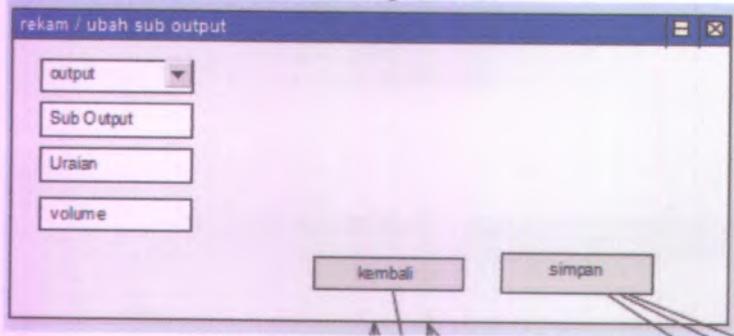
Gambar 13.3 Tayang KPJM GUI

F.4. DIPA GUI



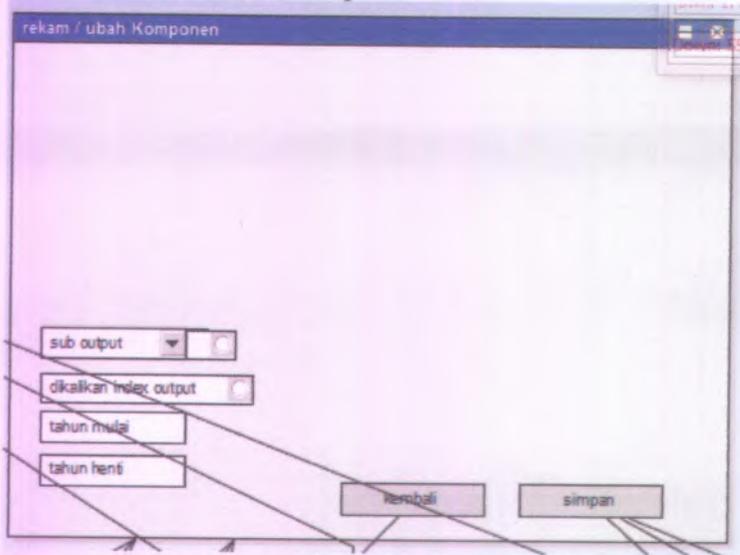
Gambar 13.4 DIPA GUI

F.5. Rekam Ubah Sub Output GUI



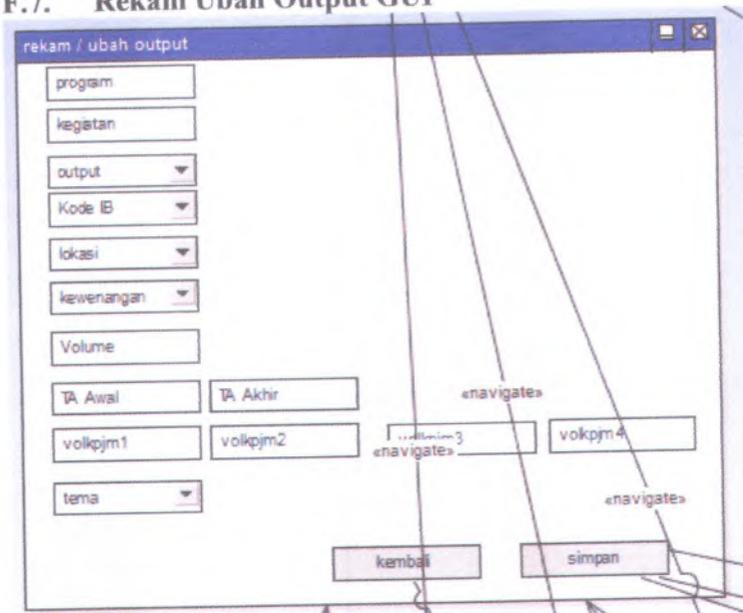
Gambar 13.5 Rekam Ubah Sub Output GUI

F.6. Rekam Ubah Komponen GUI



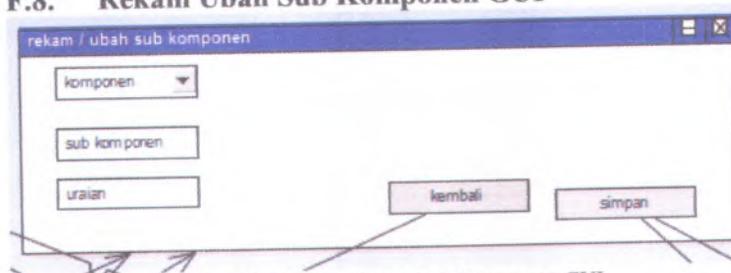
Gambar 13.6 Rekam Ubah Komponen GUI

F.7. Rekam Ubah Output GUI



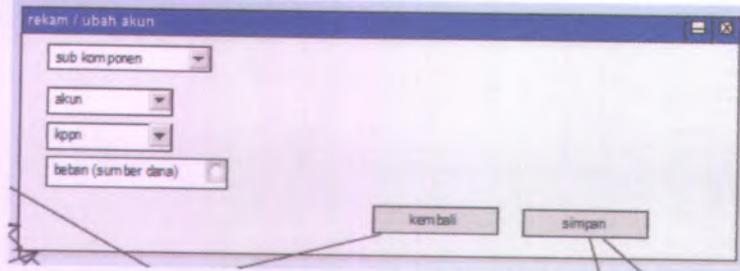
Gambar 13.7 Rekam Ubah Output GUI

F.8. Rekam Ubah Sub Komponen GUI



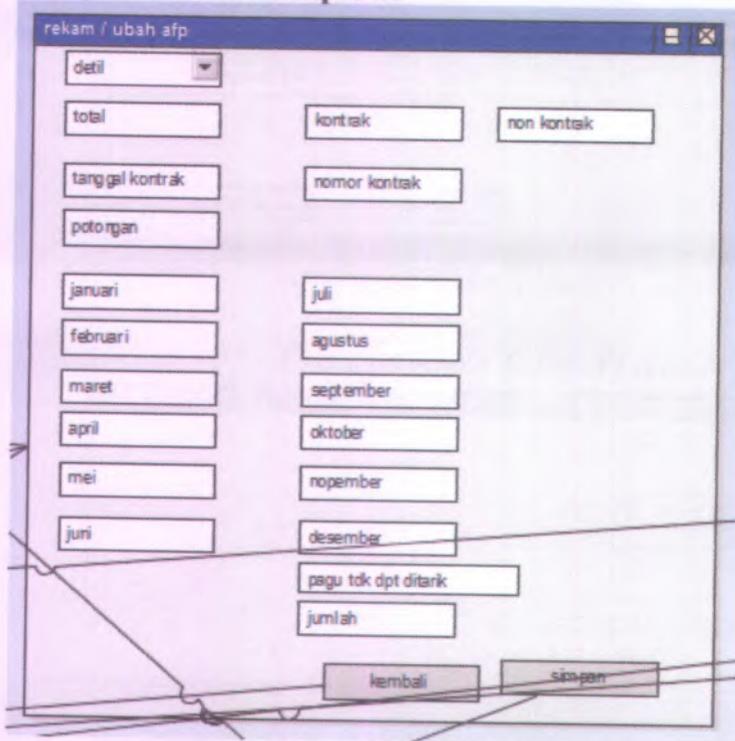
Gambar 13.8 Rekam Ubah Sub Komponen GUI

F.9. Rekam Ubah Akun GUI



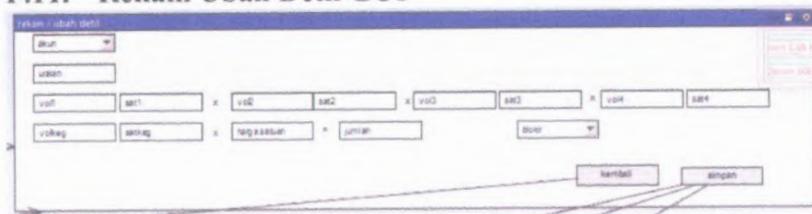
Gambar 13.9 Rekam Ubah Akun GUI

F.10. Rekam Ubah Afp GUI



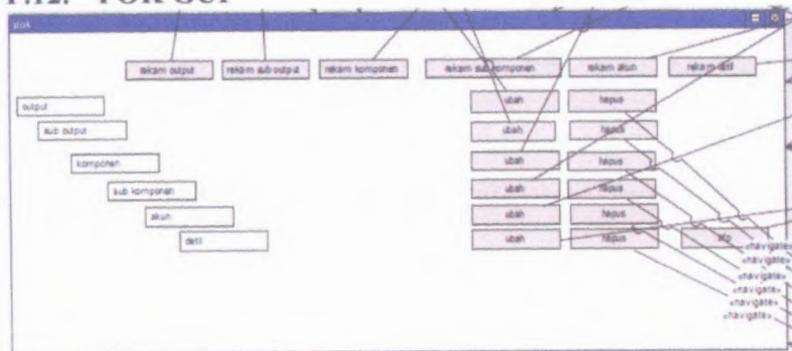
Gambar 13.10 Rekam Ubah Afp GUI

F.11. Rekam Ubah Detil GUI



Gambar 13.11 Rekam Ubah Detil GUI

F.12. POK GUI



Gambar 13.12 POK GU

BIODATA PENULIS



Penulis dilahirkan di Tulungagung, 14 April 1984, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal di SD Negeri Gondang 2 Tulungagung, lalu melanjutkan di SMP N 2 Gondang, dilanjutkan di SMUN Boyolangu dan Sekolah Tinggi Akuntansi Negara. Pada tahun 2009,

penulis mengajukan diri untuk mengikuti Program Beasiswa Internal Direktorat Jenderal Perbendaharaan, Departemen Keuangan, dan diterima di Jurusan Sistem Informasi pada tahun 2009 dengan NRP 5209108709.

Selama melaksanakan tugas belajar di Jurusan Sistem Informasi, penulis juga aktif membantu instansi bernaung dalam kegiatan pengembangan website untuk kantor-kantor daerah, dan juga aktif dalam menekuni hobi travelling. Untuk pengerjaan Tugas Akhir, penulis mengambil bidang minat e-Bisnis dengan topik tentang Integrasi Aplikasi bagi instansi Direktorat Jenderal Perbendaharaan.