ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS091336

RANCANG BANGUN APLIKASI PERENCANAAN PENCAIRAN DANA DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL

AGASTYA VITADHANI
NRP 5211 105 709

Dosen Pembimbing I
Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014

ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS091336

DEVELOPMENT OF DISBURSEMENT PLANNING APPLICATION USING JAVASERVER PAGES AND MYSQL TECHNOLOGY

AGASTYA VITADHANI
NRP 5211 105709

Academic Supervisor I
Dr. Eng. Febriliyan Samopa, S.Kom, M.Kom

DEPARTMENT OF INFORMATION SYSTEM
Faculty of Information Technology
Institute of Technology Sepuluh Nopember
Surabaya 2014

RANCANG BANGUN APLIKASI PERENCANAAN PENCAIRAN DANA DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL

Lembar Pengesahan
TUGAS AKHIR
Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
AGASTYA VITADHANI
5211 105 709

Surabaya, 2014

KETUA
JURUSAN SISTEM INFORMASI

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

RANCANG BANGUN APLIKASI PERENCANAAN PENCAIRAN DANA DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL

TUGAS AKHIR

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

Oleh:

AGASTYA VITADHANI
5211 105 709

Disetujui Tim Penguji:

Tanggal Ujian :
Periode Wisuda :

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

Andre Parvian Aristio, S.Kom, M.Sc (Penguji I)

Faizal Johan, S.Kom, M.T (Penguji II)

RANCANG BANGUN APLIKASI PERENCANAAN PENCAIRAN DANA DENGAN TEKNOLOGI JAVASERVER PAGES DAN MYSQL

Nama Mahasiswa : **Agastya Vitadhani**
NRP : **5211 105 709**
Jurusan : **Sistem Informasi FTiF-ITS**
Dosen Pembimbing : **Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom**

ABSTRAK

Salah satu tugas DJPBN adalah mengelola penerimaan dan pengeluaran negara yang dananya bersumber dari Anggaran Pendapatan dan Belanja Negara (APBN).Tugas DJPBN tersebut di atas tidak dapat dilaksanakan secara efektif karena setiap Kementerian Negara/Lembaga yang memiliki sumber pendanaan dari APBN, belum dapat merencanakan pengeluaran negara secara akurat. Di setiap awal tahun anggaran, Kementerian Keuangan melakukan peminjaman dana untuk menutup pengeluaran negara yang nilai dan waktunya tidak dapat diprediksi. Akibatnya, negara merugi karena harus menanggung bunga dari pinjaman yang belum tentu digunakan dananya.

Agar dana yang dibutuhkan dan yang mampu dihimpun oleh setiap satuan kerja (satker) Kementerian Negara/Lembaga dapat diketahui secara akurat, perlu diciptakan sebuah aplikasi. Aplikasi tersebut harus dapat memfasilitasi satker dalam melakukan perencanaan pengeluaran dana di setiap periode. Aplikasi tersebut juga harus dapat melakukan rekapitulasi perencanaan pengeluaran negara serta menyusun perencanaan kas. Penyajian perencanaan kas melalui aplikasi tersebut perlu untuk dapat ditampilkan secara real-time dengan menggunakan basis web.

Dengan adanya Aplikasi Perencanaan Pencairan Dana diharapkan perencanaan pengeluaran dan penerimaan negara dapat dilakukan seakurat mungkin, sehingga dapat memberikan informasi yang berkualitas bagi pengambilan keputusan dalam hal pengelolaan kas negara.

Kata kunci: *Aplikasi perencanaan pencairan dana, KPPN, Satker*

DEVELOPMENT OF DISBURSEMENT PLANNING APPLICATION USING JAVASERVER PAGES AND MYSQL TECHNOLOGY

Name : **Agastya Vitadhani**
NRP : **5211 105 709**
Department : **Information System FTiF-ITS**
Supervisor : **Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom**

ABSTRACT

One of DJPBN core business is to manage government revenue and expenditure where the funds come from state budget (APBN).the task can't be carried out effectively because every ministry/institution that has been funded by APBN, can not plan its own expenditure accurately. At the beginning of each year, ministry of finance was looking for a loan to cover goverment expenditure that its value and time can't be predicted. As the result, government loosing money because they have to bear the interest of the loan that has not been spent effectively.

In order for the ministry of finance be able to predict the funds that needed and collected by every budget user authority (satker) of ministry/institution accuratelly, an application is required. The application must be able to facilitate satker plan the expenditure in every period The application must also be able to recapitulate government spending plan and create cash forecasting. Presentation of disbursement planning through the application should be displayed real-time using web based.

With disbursement planning application, government revenue and expenditure planning is expected to be much more accurate, so that it can provide high-quality information for decision making in the management of state treasury.

Keywords: *Disbursement planning aplication, KPPN, Satker*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Pada bab ini dijelaskan latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir.

1.1. Latar Belakang

Sebagaimana disebutkan dalam Undang-Undang No. 1 tahun 2004 tentang Perbendaharaan Negara, Menteri Keuangan selaku Bendahara Umum Negara (BUN) berwenang mengusahakan dan mengatur dana yang diperlukan dalam pelaksanaan anggaran negara (Presiden Republik Indonesia, 2004). Pelaksanaan kewenangan tersebut diperinci pada Peraturan Menteri Keuangan (PMK) Nomor: 192/PMK.05/2009 tentang Perencanaan Kas, Menteri Keuangan selaku *Chief Financial Officer* (CFO) bertanggung jawab untuk membuat perencanaan kas (Menteri Keuangan, 2009).

Perencanaan kas adalah proyeksi penerimaan dan pengeluaran negara pada suatu periode (Menteri Keuangan, 2009). Perencanaan kas disusun dari perkiraan pencairan dan penerimaan dana dari KPPN dan unit eselon I Kementerian Keuangan yang mengelola penerimaan dan pengeluaran negara. Perkiraan pencairan dan penyetoran dana dari KPPN merupakan rekapitulasi perkiraan pencairan dan penyetoran dana satker di wilayah kerja KPPN tersebut.

Mekanisme perencanaan kas dilaksanakan dengan dua aplikasi:

1. Aplikasi Forecasting Satker (AFS) digunakan satker untuk mengelola perencanaan kas pada satker yang bersangkutan
2. Aplikasi Forecasting KPPN (AFK) digunakan KPPN untuk mengelola perencanaan kas seluruh satker di wilayah kerjanya.

Satker merencanakan pencairan dananya dengan AFS dalam interval bulan, minggu dan hari. Satker mengirimkan output AFS berupa arsip data komputer (ADK) ke kppn. Selanjutnya KPPN

menerima ADK dengan AFK. Mekanisme pencairan dana satker digambarkan dengan bagan alur pada gambar 1 sesuai dengan Peraturan Direktur Jenderal Perbendaharaan nomor 66 tahun 2005 (Direktur Jenderal Perbendaharaan, 2005).

Kesulitan mekanisme perencanaan kas muncul karena perencanaan kas harus sesuai dengan realisasinya. Banyaknya hambatan misalnya penundaan atau keterlambatan pencairan dana oleh satker dan penolakan pencairan dana oleh KPPN mengakibatkan proses update perencanaan kas menjadi aktifitas yang sering dilakukan. Hal ini berdampak pada proses pengiriman adk kppn yang juga menjadi sering dilakukan dan peningkatan volume pekerjaan klerikal.

Sampai saat ini sebagian besar satker masih belum mampu melakukan perencanaan atas pengeluaran negara. Mekanisme perencanaan kas dengan AFS dan AFK yang dimulai tahun 2010 tidak berjalan dengan baik, bahkah tidak dilaksanakan sama sekali sejak tahun 2013. Kesulitan yang muncul sebagaimana diuraikan sebelumnya mengakibatkan kurangnya minat dari satuan kerja untuk menggunakan AFS. Menurut Kumaradewi (Kumaradewi & Syafruddin, 2013) salah satu solusi dalam meningkatkan minat penggunaan AFS dengan memudahkan metode perekaman dan pengiriman data misalnya secara online.

Berdasarkan uraian di atas, dikemukakan rancangan aplikasi bernama Aplikasi Perencanaan Pencairan Dana (APPD) yang mengintegrasikan AFS dan AFK. APPD berbasis web dengan menggunakan bahasa pemrograman Java (JavaServer dan JSP) serta sistem manajemen basis data MySQL.

Pertimbangan membangun aplikasi berbasis web karena memiliki keunggulan yaitu :

1. Sifat web browser yang *obiquity* (ada dimana-mana).
2. Kenyamanan dalam penggunaan web browser sebagai *client*, misalnya tanpa penginstalan dan kemudahan update aplikasi.
3. Dapat dijalankan di berbagai sistem operasi.
4. Dapat dijalankan di perangkat *mobile* (misalnya handphone, tablet) yang memiliki *browser*.

Teknologi JSP dipilih karena mampu menciptakan web content yang bersifat dinamis dan statis dengan menggunakan bahasa pemrograman Java. Saat ini, PHP merupakan *web server system* yang paling populer, tetapi berdasarkan Scott Trent (Trent et al., 2008) untuk penggunaan dengan kinerja dan keluaran yang terbaik, JSP lebih disarankan daripada PHP. Sedangkan MySQL dipilih karena merupakan sistem manajemen basis data relasional yang populer digunakan untuk aplikasi berbasis web dan sifatnya yang *open source*. Selain pertimbangan tersebut, MySQL dipilih karena saat ini digunakan di setiap KPPN dan satker mitra kerjanya.

1.2. Perumusan Masalah

Permasalahan yang akan terselesaikan dalam tugas akhir ini adalah:

1. Bagaimana membangun APPD dengan menggunakan teknologi JavaServer Pages dan MySQL?
2. Bagaimana membangun dan mendesain APPD agar sesuai dengan proses bisnis dan memudahkan pengguna?

1.3. Batasan Masalah

Dari permasalahan yang telah disebutkan diatas, yang menjadi batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Satker yang dimaksud adalah satker yang menggunakan dana APBN dan dibayarkan melalui KPPN.
2. Satker yang dimaksud merupakan satker instansi pemerintah pusat, bukan satuan kerja perangkat daerah.
3. Aplikasi tidak mencakup perencanaan penerimaan negara.
4. Pengeluaran negara yang dimaksud hanya meliputi belanja negara.
5. Hasil akhir dari tugas ini adalah prototype APPD.

1.4. Tujuan

Tujuan akhir dari penulisan tugas akhir ini adalah membangun APPD.

1.5. Manfaat

Manfaat yang dapat diberikan dari tugas akhir ini adalah sebagai berikut:

1. Bagi organisasi
   a. APPD dapat mempermudah pengelolaan pengeluaran negara pada satker.
   b. APPD dapat membantu meningkatkan akurasi kebutuhan dana satker pada Kementerian Keuangan.
2. Bagi iptek

   Tugas akhir dapat menjadi referensi bagi pengembangan aplikasi yang sama di masa yang akan datang.
3. Bagi penulis
   a. Dapat mengetahui bagaimana proses membangun APPD.
   b. Dapat menerapkan rancang bangun perangkat lunak pada dunia kerja.

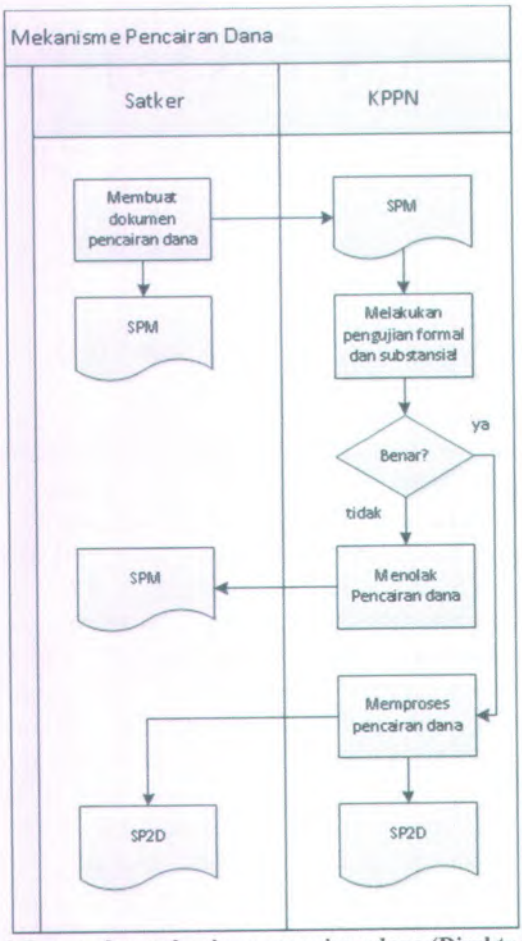
BAB II
LANDASAN TEORI

Pada bab ini dijelaskan literatur yang berhubungan dengan permasalahan tugas akhir.

2.1. Pencairan Dana

Pencairan dana yang dimaksud dalam tugas akhir ini adalah pencairan dana APBN yang dilakukan satker melalui KPPN sebagaimana ditetapkan dalam Peraturan Direktur Jenderal Perbendaharaan nomor PER-66/PB/2005 tentang Mekanisme Pelaksanaan Pembayaran Atas Beban Anggaran Pendapatan Dan Belanja Negara (Direktur Jenderal Perbendaharaan, 2005). Untuk mengajukan pencairan dana, satker menerbitkan surat perintah membayar (SPM) beserta lampirannya. SPM berisi identitas satker, kode pembebanan di APBN, rincian penggunaan dana dan informasi lainnya.

Satker mengajukan SPM ke KPPN untuk diuji kebenarannya secara formal dan substansial. Pengujian formal dilakukan dengan pencocokan tanda tangan, memeriksa pengisian jumlah uang dan kebenaran penulisan. Pengujian substansial dilakukan dengan menguji kebenaran perhitungan, menguji ketersediaan dana, menguji dokumen dasar penagihan, menguji surat pernyataan tanggung jawab (SPTB) dan menguji faktur pajak beserta surat setoran pajak (SSP). Pengujian ditindaklanjuti dengan penerbitan surat perintah pencairan dana (SP2D) apabila SPM memenuhi syarat. Apabila SPM yang diajukan tidak memenuhi syarat dilakukan pengembalian SPM kepada penerbit SPM. Mekanisme pencairan diilustrasikan dengan bagan alur pada gambar 1 sesuai dengan Peraturan Direktur Jenderal Perbendaharaan nomor 66 tahun 2005 (Direktur Jenderal Perbendaharaan, 2005).



Gambar 1 Bagan alur mekanisme pencairan dana (Direktur Jenderal Perbendaharaan, 2005)

2.2. Perencanaan Kas

Sebagaimana ditetapkan dalam PMK nomor: 195/PMK.05/2009 tentang Perencanaan Kas, perencanaan kas pemerintah pusat yang selanjutnya disebut perencanaan kas merupakan proyeksi penerimaan dan pengeluaran negara pada

periode tertentu dalam rangka pelaksanaan APBN (Menteri Keuangan, 2009). Ruang lingkup perencanaan kas meliputi perencanaan penerimaan negara, perencanaan pengeluaran negara dan perencanaan saldo rekening Kas Umum Negara (KUN) yang dilakukan secara periodik dalam rangka pelaksanaan APBN. Tujuan perencanaan kas adalah:

1. Kementerian Keuangan dapat memastikan ketersediaan dana guna memenuhi kewajiban negara;
2. Kementerian Keuangan dapat mengambil tindakan yang efektif dan efisien dalam rangka mengoptimalkan kelebihan kas atau menutupi kekurangan kas;
3. Kementerian negara/lembaga memperoleh dana senilai perkiraan pencairan dana dan/atau perkiraan penyetoran dana untuk membiayai kegiatan yang akan dilaksanakan;
4. Kementerian negara/lembaga memperoleh dana sesuai dengan waktu pelaksanaan kegiatan.

Pada PMK nomor: 195/PMK.05/2009 disebutkan perkiraan pencairan dana adalah daftar perkiraan kebutuhan dana untuk melaksanakan kegiatan yang dibuat oleh kantor/satker dan disampaikan ke KPPN untuk periode tertentu (Menteri Keuangan, 2009). Perkiraan pencairan dana dibuat secara periodik yaitu bulanan mingguan dan harian. Perkiraan penyetoran dana adalah daftar perkiraan penyetoran dana pada bank persepsi/Bendahara Umum Negara (BUN) yang dibuat oleh kantor/satker/instansi eselon I dan disampaikan ke KPPN atau kuasa BUN pusat untuk periode tertentu. perkiraan penyetoran dana dibuat secara periodik yaitu mingguan dan bulanan. Pelaksanaan perencanaan kas ditetapkan lebih spesifik dengan Keputusan Direktur Jenderal Perbendaharaan nomor Kep-163/PB/2011 tentang perubahan atas Keputusan Direktur Jenderal Perbendaharaan nomor Kep-185/PB/2010 tentang standar prosedur operasi/*standard operating procedures* di lingkungan instansi vertikal Direktorat Jenderal Perbendaharaan.

2.3. JavaServer Pages

Menurut Bergsten (Bergsten, 2002)JavaServer Pages adalah teknologiuntuk mengembangkan halaman web dengan memasukkan konten dinamis. Halaman JSP dapat merubah kontennya berdasarkan sejumlah variabel, termasuk identitas user, tipe browser user, informasi yang dimasukkan oleh user dan pilihan yang dibuat oleh user.

JSP merupakan cara baru dalam menggunakan servlet. Perbedaan JSP dengan Servlets terletak pada kegunaannya, pada Servlets lebih menitikberatkan pada layer aplikasi atau logika proses dibandingkan dengan layer presentasi. Sedangkan pada JSP lebih menitikberatkan pada sisi presentasi kepada pengguna, bagaimana suatu aplikasi ditampilkan kepada pengguna. Menurut Hunter (Hunter & Crawford, 1998) perbedaan utama JSP dengan servlet adalah penggunaan servet dengan menyisipkan tag <SERVLET> pada halaman HTML, sementara penggunaan JSP dengan mampu secara langsung menyisipkan potongan dari kode servlet. Setiap blok kode servlet (dinamakan scriptlet) didahului tag <% dan ditutup dengan tag %>.

Menurut Bergsten (Bergsten, 2002) keunggulan dari penggunaan teknologi JSP adalah:

1. JSP dibangun dari java servlet API, sehingga JSP terintegrasi dengan enterprise java API, termasuk di dalamnya:
    - *JDBC*
    - Remote Method Invocation (RMI) and OMG CORBA support
    - JNDI (Java Naming and Directory Interface)
    - Enterprise JavaBeans (EJB)
    - JMS (Java Message Service)
    - JTA (Java Transaction API)
    - JAXP (Java API for XML Processing)
    - JavaMail

2. JSP mendukung konten dinamis berbasis elemen dan scripting dan mendukung programmer untuk mengembangkan tag *libraries* secara kustom.
3. Halaman JSP dicompile untuk pemrosesan server secara efisien.
4. JSP page dapat digunakan dalam kombinasi dengan servlet yang menangani logika bisnis modelnya didukung oleh *java servlet templates engines*.

2.4. MySQL

MySQL merupakan salah satu sistem manajemen basis data *bersifat relasional yang berlisensi open source*. MySQL dikembangkan di tahun 1994 oleh Michael "Monty" Widenius dari perusahaan swedia TcX. MySQL diluncurkan ke internet di tahun 1995 oleh David Axmark.

Perusahaan TcX mengubah namanya menjadi MySQL AB setelah popularitas MySQL semakin meningkat. Di tahun 2008 *Sun Microsystem Inc mengakusisi MySQL AB. Kepemilikan* MySQL berganti lagi setelah Oracle Corporation mengakusisi Sun Microsystem Inc di tahun 2010.

Menurut DuBois (DuBois, 2009) Keunggulan MySQL dibandingkan dengan sistem manajemen database lainnya adalah:

1. MySQL adalah sistem database yang cepat.
2. MySQL adalah sistem database yang memiliki kinerja yang tinggi tetapi relatif sederhana.
3. MySQL menggunakan structured query language, bahasa standar seluruh sistem database modern
4. Server MySQL *multi-threaded*, sehingga banyak client yang dapat terkoneksi di waktu yang bersamaan.
5. MySQL dapat diakses menggunakan jaringan komputer bahkan melalui internet. Untuk melindungi keamanannya MySQL memiliki kontrol akses serta dapat melakukan koneksi terenkripsi menggunakan *secure socket layer protocol*.

6. MySQL dapat berjalan di berbagai sistem operasi.
7. MySQL memiliki ukuran distribusi yang relatif kecil.
8. MySQL tersedia dalam berbagai lisensi, baik *open source* maupun berbayar.
9. Distribusi dan source code MySQL mudah didapatkan.

2.5. Iconix Process

Iconic Process merupakan salah satu metodologi pengembangan perangkat lunak. Metode *Iconix Process* hanya menggunakan core subset dari Unified Modeling Language (UML) sehingga lebih sederhana. UML adalah bahasa standar yang digunakan untuk proses dokumentasi, pembuatan spesifikasi dan membangun sebuah sistem perangkat lunak.Menurut Rosenberg (Rosenberg & Stephens, 2007) metodologi *Iconix Process* terdiri atas:

1. Identifikasi kebutuhan, terdiri dari:
   a. Kebutuhan fungsional;
   b. Domain *model*;
   c. Kebutuhan behavioral (*GUI storyboard* dan *use cases*);
   d. Review kebutuhan;
2. Pembuatan desain pendahuluan, terdiri dari:
   a. *Robustness analysis;*
   b. Pembaharuan *domain model*;
   c. Identifikasi *controller*;
3. Review desain awal;
4. Pembuatan desain terperinci, terdiri dari:
   a. *Sequence diagram;*
   b. Pembaharuan *domain model*;
   c. *Class diagram*;
5. Review desain kritikal;
6. Implementasi, terdiri dari:
   a. Pengujian *code*;
   b. Integrasi dan pengujian skenario;
   c. Review *code* dan pembaharuan model;

BAB III
METODOLOGI PENELITIAN

Pada bagian ini akan dibahas mengenai langkah-langkah kegiatan penelitian yang akan dilakukan sebagaimana digambarkan pada gambar 2. Metodologi ini akan dibagi ke dalam beberapa tahap, yaitu:

Gambar 2 Alur metodologi penelitian

3.1. Studi Literatur

Studi literatur dalam pembuatan tugas akhir ini adalah dengan melakukan pembelajaran serta pemahaman pada literatur yang berkaitan dengan permasalahan yang ada. Literatur didapatkan melalui media buku, jurnal akademis, serta kumpulan peraturan perundangan.

3.2. Identifikasi Kebutuhan

Setelah memahami konsep dan teori yang mendukung pengembangan APPD, maka pada tahap ini dilakukan identifikasi kebutuhan sistem. Identifikasi ini dilakukan dengan menganalisis sistem yang telah ada sebelumnya dan isu-isu terbaru yang terkait dengan permasalahan. Dari identifikasi tersebut akan didapatkan gambaran jelas tentang APPD.

3.3. Desain Aplikasi

Pada tahap ini dilakukan desain dari sistem aplikasi yang akan dikembangkan dengan menyesuaikan proses bisnisnya. Pembuatan desain APPD menggunakan metode pengembangan perangkat lunak *Iconix Process* dengan menggunakan aplikasi Enterprise Architect dan MySQL Workbench.

3.4. Pembuatan Aplikasi

Setelah proses pembuatan desain selesai, dilanjutkan dengan pembuatan apikasi. Aplikasi dikembangkan dengan menggunakan *platform* JavaServer Pages dan MySQL sebagai basis data. Pembuatan aplikasi tersebut dilaksanakan dengan menggunakan aplikasi Netbean IDE, Xampp dan MySQL Workbench.

3.5. Uji Coba Aplikasi

Pada tahap ini akan dilakukan uji coba terhadap aplikasi yang telah dibangun. Uji coba dilakukan untuk mengevaluasi operasi dan ketahanan aplikasi.

Uji coba aplikasi dilakukan dengan metode uji fungsional dan uji struktural. Menurut Ratzmann (Ratzmann & De Young, 2003)secara tradisional uji fungsional disebutkan juga sebagai *black box test* dan uji struktural disebut juga sebagai *white box test*. Menurut Myers (Myers, 2004)*black box testing* atau *data-driven testing* atau *input/output testing* adalah pendekatan dalam melakukan pengujian dengan pertimbangan bahwa program merupakan suatu entitas yang komplet dengan tanpa mempertimbangkan struktur internal program. Sementara itu *white box testing* atau *logic-driven testing* adalah pengujian terhadap struktur internal program.

Black box testing pada tugas akhir dilakukan dengan :

1. Metode *security testing* yang dilaksanakan dengan menguji keamanan aplikasi menggunakan *tools* berupa Acunetix.
2. Metode *stress testing* yang dilaksanakan dengan menguji respon aplikasi dengan beban yang tinggi menggunakan *tools* berupa JMeter.

White box testing dilakukan metode *exhaustive input testing* yang dilaksanakan dengan memasukkan berbagai input di setiap *method* dengan mengunakan fasilitas JUnit pada java.

Halaman ini sengaja dikosongkan

BAB IV
DESAIN APLIKASI

Dalam bab ini akan dijelaskan mengenai pembuatan desain APPD. Pembuatan desain APPD menggunakan metode pengembangan perangkat lunak *Iconix Process*.

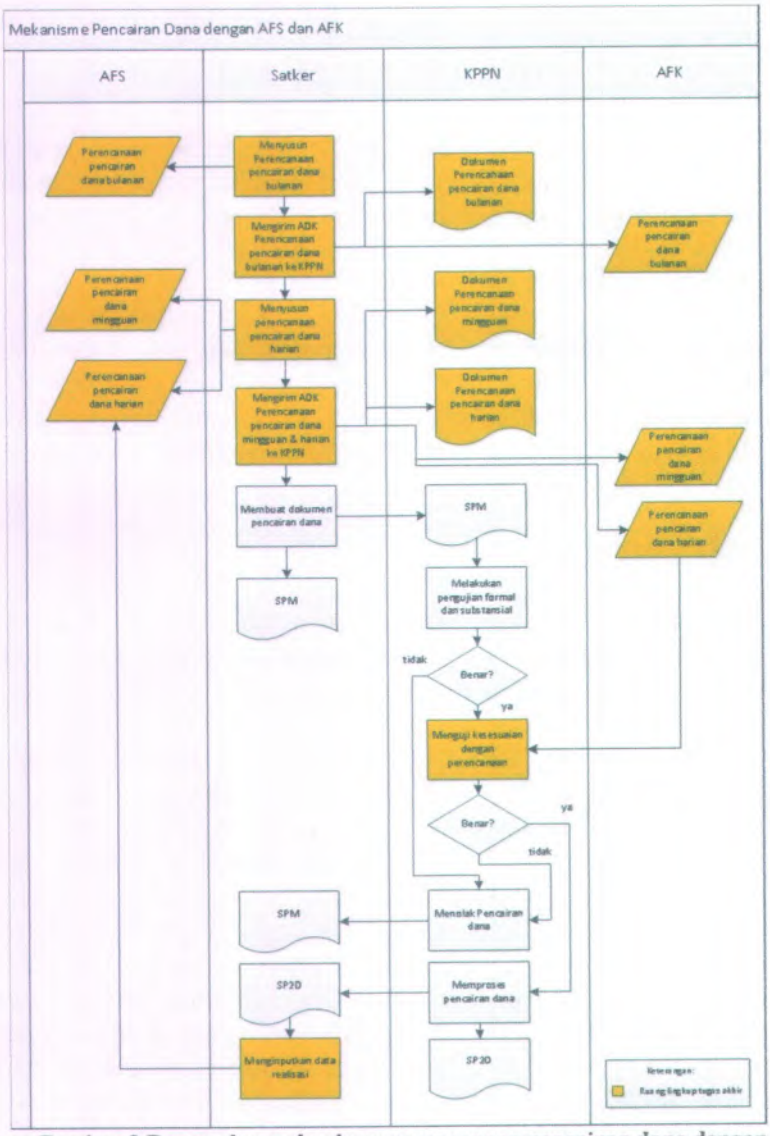
4.1. Identifikasi Kebutuhan

Identifikasi kebutuhan diperlukan untuk menentukan kebutuhan pengguna atas sistem, sehingga perangkat lunak yang dihasilkan sesuai dengan ekspektasi pengguna. Identifikasi kebutuhan dilakukan dengan mengumpulkan informasi dari pengguna dan dosen pembimbing.

4.1.1. Studi Lapangan

Sebagaimana telah diuraikan di bab I, konsep perencanaan kas saat ini dilaksanakan dengan mengunakan AFS an AFK. Kedua aplikasi tersebut merupakan aplikasi desktop dengan database yang berbeda.

Setiap awal tahun satker membuat perencanaan bulanan untuk satu tahun anggaran menggunakan AFS yang akan diupdate dan diserahkan ADK serta *hardcopy*-nya ke KPPN setiap bulan. Selanjutnya perencanaan bulanan tersebut akan menjadi dasar bagi pembuatan perencanaan harian. Perencanaan harian harus diserahkan ADK dan *hardcopy*-nya ke KPPN di akhir minggu kerja. Pegawai front office KPPN wajib mengawasi pencairan dana satker dengan data yang telah diterima melalui AFK. Perencanaan harian merupakan batas maksimal pencairan dana oleh satker. Satker yang telah merealisasikan pencairan dananya harus menginputkan data realisasi pencairan dana harian di AFS. Mekanisme perencanaan kas dengan AFS dan AFK ditunjukkan dengan gambar 3.



Gambar 3 Bagan alur mekanisme perencanaan pencairan dana dengan AFS dan AFK (Direktur Jenderal Perbendaharaan, 2005)

4.1.2. Kebutuhan Fungsional

Kebutuhan fungsional APPD disusun dengan berdasar pada peraturan yang berlaku mengenai perencanaan kas. Terdapat kebutuhan fungsional aplikasi yang tidak diatur dalam peraturan yang berlaku karena merupakan fitur pendukung dan keberadaannya tidak menyalahi peraturan yang berlaku.

Untuk memudahkan identifikasi kebutuhan fungsional, penulis mengklasifikasikannya berdasarkan jenis pengguna. Sesuai dengan proses bisnis perencanaan kas, penulis mengklasifikasikan pengguna ke dalam lima jenis, yang terdiri dari tiga aktor utama dan dua kombinasi :

1. Pengguna satker

Pengguna satker merupakan pengguna aplikasi yang berasal dari pihak satker dan bertugas menggunakan aplikasi sesuai dengan kewenangan satker.

2. Pengguna KPPN

Pengguna satker merupakan pengguna aplikasi yang berasal dari pihak KPPN dan bertugas menggunakan aplikasi sesuai dengan kewenangan KPPN.

3. Administrator.

Administrator merupakan pengguna aplikasi yang berasal dari pihak Direktorat Jenderal Perbendaharaan dan berperan sebagai administrator aplikasi.

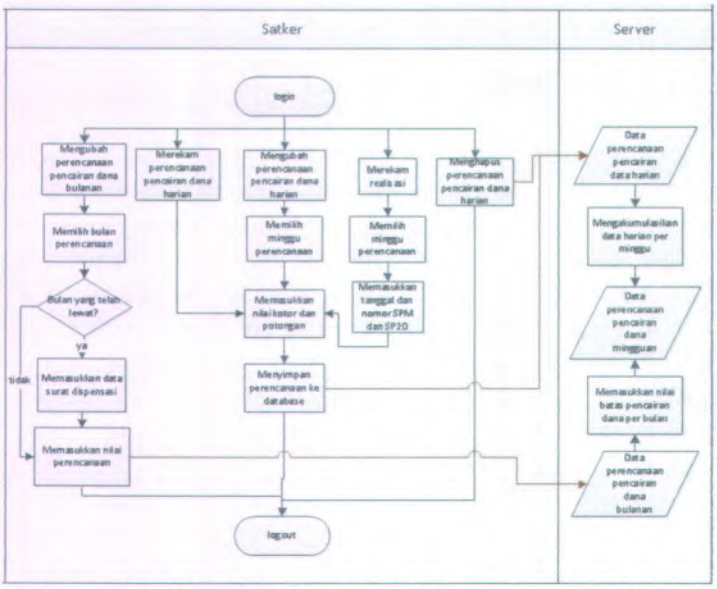
4. Pengguna umum

Pengguna umum merupakan pengguna aplikasi yang terdiri dari seluruh aktor. Keberadaan pengguna umum karena terdapat kesamaan fungsionalitas dari setiap aktor.

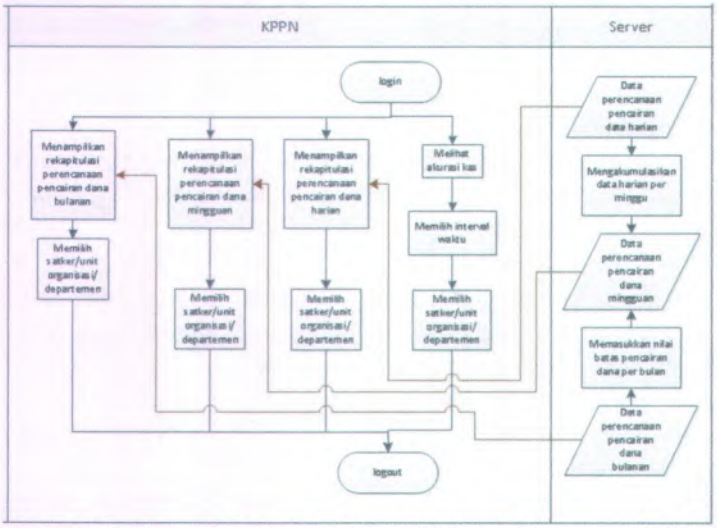
5. Pengguna non administrator

Pengguna non administrator terdiri dari operator satker dan operator kppn.

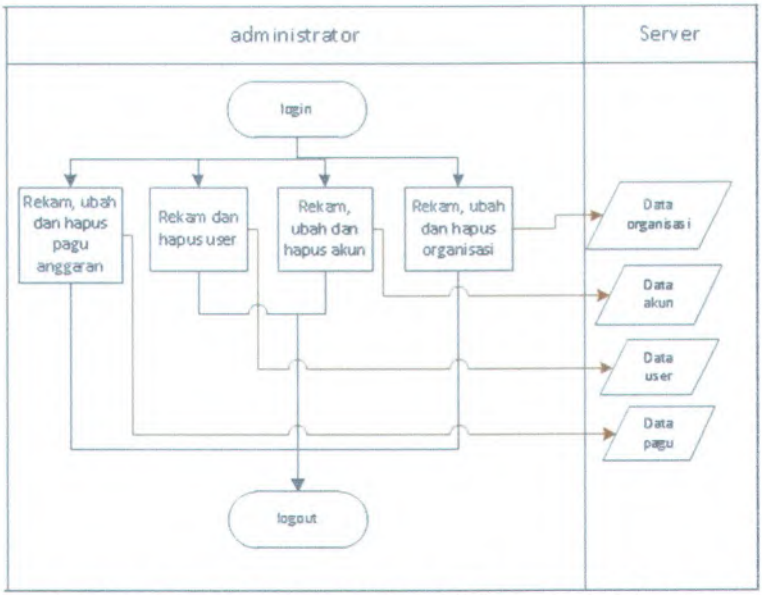
Fungsionalitas bagi aktor utama yang diharapkan dari APPD ditunjukkan pada gambar 4, 5 dan 6.



Gambar 4 Bagan alur fungsionalitas APPD untuk operator satker



Gambar 5 Bagan alur fungsionalitas APPD untuk operator KPPN



Gambar 6 Bagan alur fungsionalitas APPD untuk administrator

Dari ketiga jenis pengguna tersebut, diidentifikasikan kebutuhan fungsional aplikasi sebagai berikut:

1. Kebutuhan pengguna Satker:

Kebutuhan fungsional untuk pengguna dari kategori pengguna satker ditunjukkan pada tabel 1.

Tabel 1 Kebutuhan fungsional pengguna satker

Kode	Fungsional	Landasan
F1	Memungkinkan pengguna untuk mengubah perencanaan pencairan dana bulanan.	PMK 195/PMK/2009 pasal 6 (2), (3), (4) dan (6) pasal 8 (1) dan (2) pasal 12 (1) dan (2) pasal 13 (1) dan (2)
F2	Memungkinkan pengguna untuk merekam, mengubah dan menghapus perencanaan	PMK 195/PMK/2009 pasal 6 (2), (3), (4) dan (6) pasal 12 (5) dan (6) pasal

	pencairan dana harian.	13 (4) PER-03/PB/2010 pasal 1 (5) pasal 5, pasal 6, pasal 7 dan pasal 8
F3	Mengakumulasikan perencanaan dana harian ke perencanaan dana mingguan.	PMK 195/PMK/2009 pasal 6 (2), (3), (4) dan (6) pasal 12 (4) dan (5)
F4	Memungkinkan pengguna untuk merekam, mengubah dan menghapus data realisasi pada perencanaan pencairan dana harian.	-

2. Kebutuhan pengguna KPPN:

Kebutuhan fungsional untuk pengguna dari kategori pengguna KPPN ditunjukkan pada tabel 2.

Tabel 2 Kebutuhan fungsional pengguna KPPN

Kode	*Fungsional*	*Landasan*
F5	Menampilkan rekapitulasi perencanaan pencairan dana bulanan.	PMK 195/PMK/2009 Pasal 16 (1) dan (2)
F6	Menampilkan rekapitulasi perencanaan pencairan dana *mingguan*.	PMK 195/PMK/2009 Pasal 16 (1) dan (2)
F7	Menampilkan rekapitulasi perencanaan pencairan dana harian.	PMK 195/PMK/2009 Pasal 16 (1) dan (2) PER-03/PB/2010 pasal 10
F8	Memperhitungkan akurasi perencanaan pencairan dana.	PER-03/PB/2010 pasal 13

3. Kebutuhan administrator:

Kebutuhan fungsional untuk pengguna dari kategori administrator ditunjukkan pada tabel 3.

Tabel 3 Kebutuhan fungsional administrator

Kode	Fungsional	Landasan
F10	Memungkinkan *administrator untuk* merekam, mengubah dan menghapus pagu anggaran satker.	-
F11	Memungkinkan administrator untuk *merekam dan menghapus* pengguna.	-
F12	Memungkinkan administrator untuk merekam, mengubah dan menghapus satker.	-

4. *Kebutuhan pengguna umum*

Kebutuhan fungsional untuk pengguna dari kategori pengguna umum ditunjukkan pada tabel 4.

Tabel 4 Kebutuhan fungsional pengguna umum

Kode	Fungsional	Landasan
F13	login	-
F14	logout	-

5. Kebutuhan pengguna non administrator

Kebutuhan fungsional untuk pengguna dari kategori pengguna non administrator ditunjukkan pada tabel 5.

Tabel 5 Kebutuhan fungional pengguna non administrator

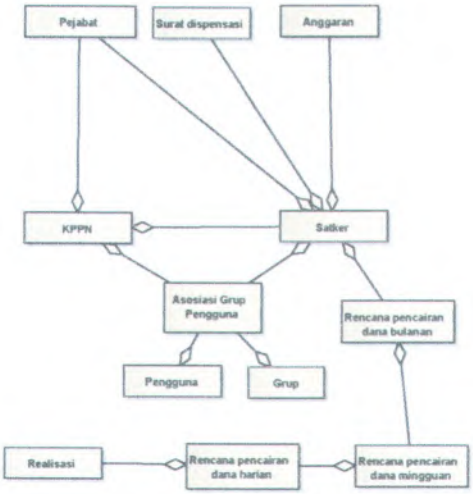
Kode	Fungsional	Landasan
F15	Merekam, mengubah, menghapus dan memilih pejabat kantor penanggung jawab	-

4.1.3. Domain Model

Domain model bentuk sederhana dari class diagram. Domain model digunakan untuk menampung seluruh objek yang digunakan pada rancang bangun aplikasi dan keterkaitan antar objek. Domain model membantu mengarahkan rancang bangun aplikasi dalam memenuhi prinsip *object-oriented*. Domain model akan terus berubah seiring berjalannya setiap tahapan metodologi tugas akhir, sehingga objek dan keterkaitannya akan semakin lengkap dan akurat. Domain model awal aplikasi diperlihatkan pada gambar 7.

Pada domain model tersebut memperlihatkan terdapat beberapa objek domain, yaitu:

1. KPPN;
2. Satker;
3. Pengguna, adalah kelas yang berisi data pengguna yang digunakan untuk login;
4. Grup, merupakan kelas yang berisi data yang mengklasifikasi pengguna;
5. Data Pengguna, adalah kelas yang berisi data terperinci mengenai pengguna aplikasi, yang merupakan composit dari kelas pengguna dan kelas grup;
6. Pejabat, adalah kelas yang berisi data pejabat yang berwenang atas perencanaan pencairan dana;
7. Surat dispensasi, adalah kleas yang berisi data mengenai surat dispensasi untuk perubahan perencanaan yang sudah kadaluarsa;
8. Anggaran, adalah kelas yang mewakili anggaran belanja satker dalam satu tahun anggaran;
9. Rencana pencairan dana bulanan, kelas yang berisi data rencana pencairan dana bulanan;
10. Rencana pencairan dana mingguan, kelas yang berisi data rencana pencairan dana mingguan;



Gambar 7 Domain Model

11. Rencana pencairan dana harian, kelas yang berisi data rencana pencairan dana harian;
12. Realisasi, adalah kelas yang mewakili realisasi pencairan dana satuan kerja yang ditunjukkan dengan SPM.;

Pada domain model tersebut memperlihatkan terdapat beberapa hubungan antar objek domain, yaitu:

1. Agregasi (satker → KPPN), menunjukkan setiap KPPN *memiliki beberapa satker di wilayah kerjanya.*
2. Agregasi (Kppn, satker → Pejabat) menunjukkan Pejabat bisa jadi bekerja di KPPN atau satker, tetapi tidak di keduanya.
3. Agregasi (surat dispensasi → satker) menunjukkan setiap satker memiliki beberapa surat dispensasi;
4. *Agregasi (Kppn, satker → Asosiasi grup pengguna)* menunjukkan setiap rincian Asosiasi grup pengguna menginformasikan bahwa pengguna bisa jadi bekerja di KPPN atau satker, tetapi tidak di keduanya.

5. Agregasi (anggaran → satker), menunjukkan setiap satker memiliki anggaran dana;
6. Agregasi (pengguna, grup → Asosiasi grup pengguna) menunjukkan asosiasi yang dihasilkan dari hubungan antara pengguna dengan grup, dimana satu grup dapat memiliki beberapa pengguna dan satu pengguna dapat memiliki beberapa grup.
7. Agregasi (Rencana pencairan dana bulanan → satker) menunjukkan setiap satker memiliki rencana pencairan dana bulanan;
8. Agregasi (Rencana pencairan dana mingguan → satker) menunjukkan setiap satker memiliki rencana pencairan dana *mingguan*;
9. Agregasi (Rencana pencairan dana harian → satker) menunjukkan setiap satker memiliki rencana pencairan dana harian;
10. Agregasi (Rencana pencairan dana mingguan → Rencana pencairan dana bulanan) menunjukkan setiap Rencana *pencairan dana bulanan memiliki rencana pencairan dana* mingguan;
11. Agregasi (Rencana pencairan dana harian → Rencana pencairan dana mingguan) menunjukkan setiap rencana penarikan dana mingguan memiliki rencana pencairan dana harian;
12. *Aggregasi (Realisasi → Rencana pencairan dana harian)* menunjukkan rencana pencairan dana harian memiliki realisasinya;

Rencana pencairan dana harian memiliki hubungan dengan rencana pencairan mingguan, tetapi rencana penarikan mingguan tidak memiliki hubungan dengan rencana penarikan dana bulanan. Hal ini dikarenakan rencana penarikan mingguan merupakan akumulasi dari rencana penarikan harian sementara rencana penarikan bulanan membutuhkan input tersendiri.

4.1.4. Kebutuhan behavioral

Identifikasi kebutuhan behavioral dimaksudkan untuk memahami bagaimana sistem dan aplikasi berinteraksi. Identifikasi kebutuhan behavioral dilakukan dengan menggunakan diagram *use case* dan GUI *storyboard*.

4.1.3.1.Diagram use case

Diagram *use case* berfungsi untuk menunjukkan hal-hal yang dapat dilakukan oleh aktor dalam lingkup aplikasi. Diagram *use case* disusun sesuai dengan kebutuhan fungsional aplikasi. Keterkaitan antara kebutuhan fungsional aplikasi dan *use case* ditampilkan pada tabel 6.

Tabel 6 Keterkaitan antara kebutuhan fungsional dan use case

Aktor	Kode fungsional	Kode *use case*
Satker	F1	UC07, UC08, UC09
	F2	UC10, UC11, UC12, UC14
	F3	UC15
	F4	UC13
KPPN	F5	UC16
	F6	UC17
	F7	UC18
	F8	UC19
Administrator	F9	UC25, UC26, UC27
	F10	UC20, UC21
	F11	UC22, UC23, UC24
Pengguna umum	F12	UC01
	F13	UC28
Pengguna non administrator	F14	UC02, UC03, UC04, UC05

Diagram *use case* dan deskripsi diagram *use case* APPD terlampir pada lampiran B, daftar *use case* secara ringkas ditampilkan pada gambar 8.

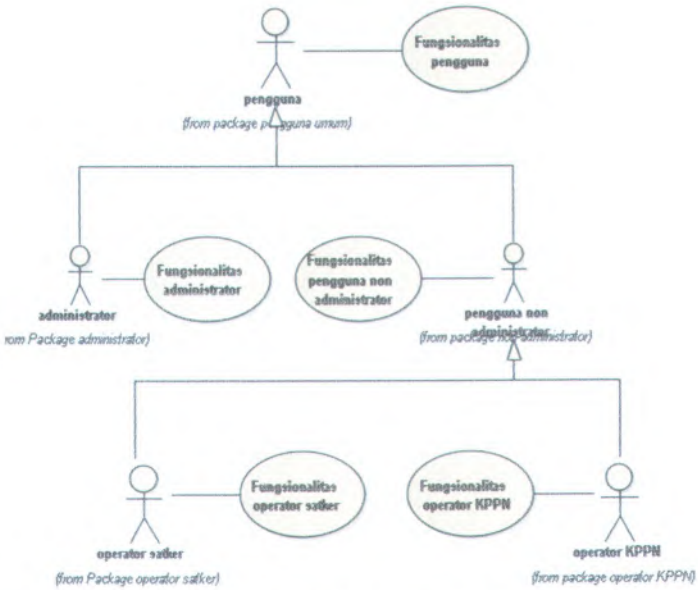
4.1.3.2.GUI storyboard

GUI *storyboard* merupakan gambaran atas form aplikasi yang ditunjukkan secara berurutan. GUI *storyboard* APPD terlampir pada lampiran A, sebagaimana dicontohkan pada gambar 9.

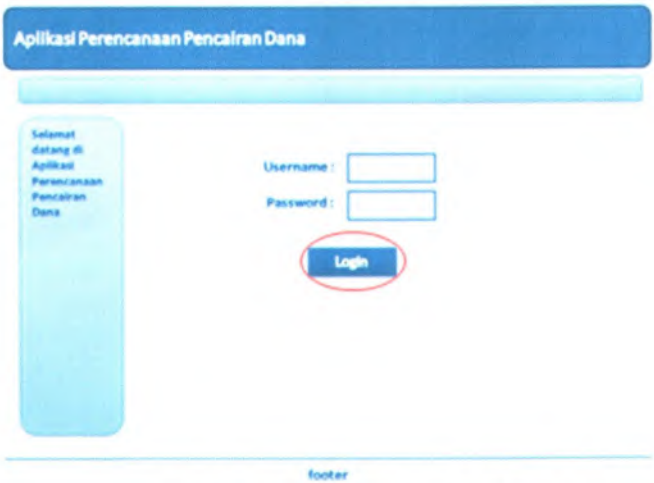
4.2. Desain Pendahuluan

4.2.1. Diagram Robustness

Diagram *robustness* merupakan diagram yang menunjukkan keterkaitan antara aktor, tampilan, proses dan objek. Fungsi diagram *robustness* menurut Rosenberg (Rosenberg & Stephens, 2007) adalah untuk memastikan *use case* ditulis dalam konteks domain model, objek yang ada pada domain model harus digunakan secara langsung pada diagram *robustness* sesuai dengan *use case*-nya. Diagram *robustness* APPD terlampir pada lampiran B, sebagaimana dicontohkan pada gambar 10.



Gambar 8 Use case APPD

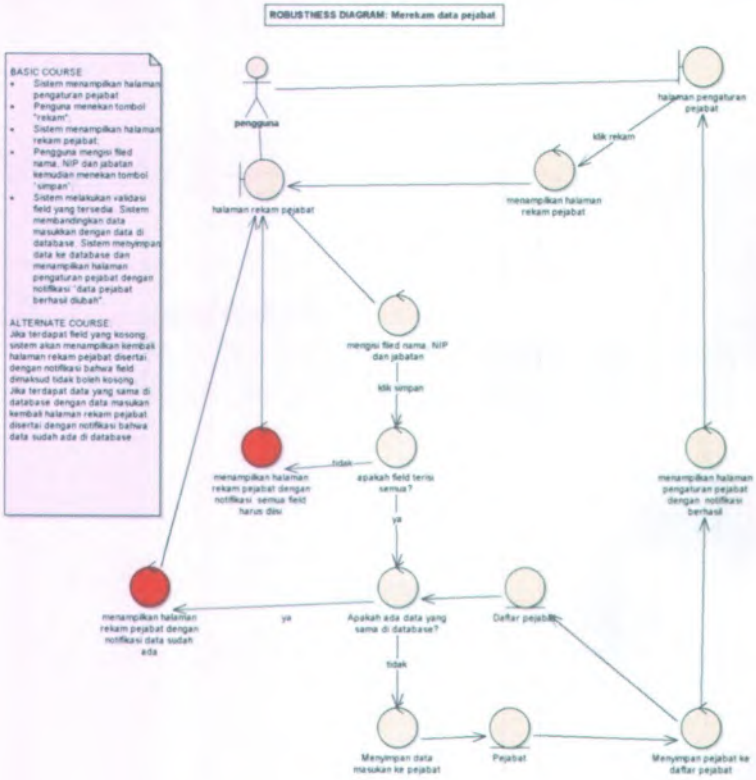


Gambar 9 Contoh Gambar GUI StoryBoard

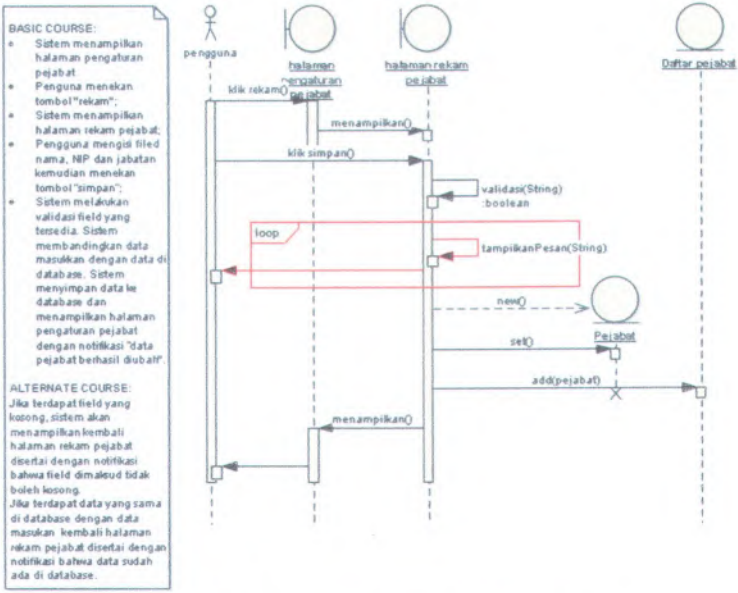
4.3. Desain Terperinci

4.3.1. Diagram Sequences

Diagram *sequences* merupakan diagram yang menggambarkan interaksi antara aktor dengan objek sesuai dengan *use case* dan diagram *robustness*. Diagram *sequences* APPD terlampir pada lampiran B, sebagaimana dicontohkan pada gambar 11.



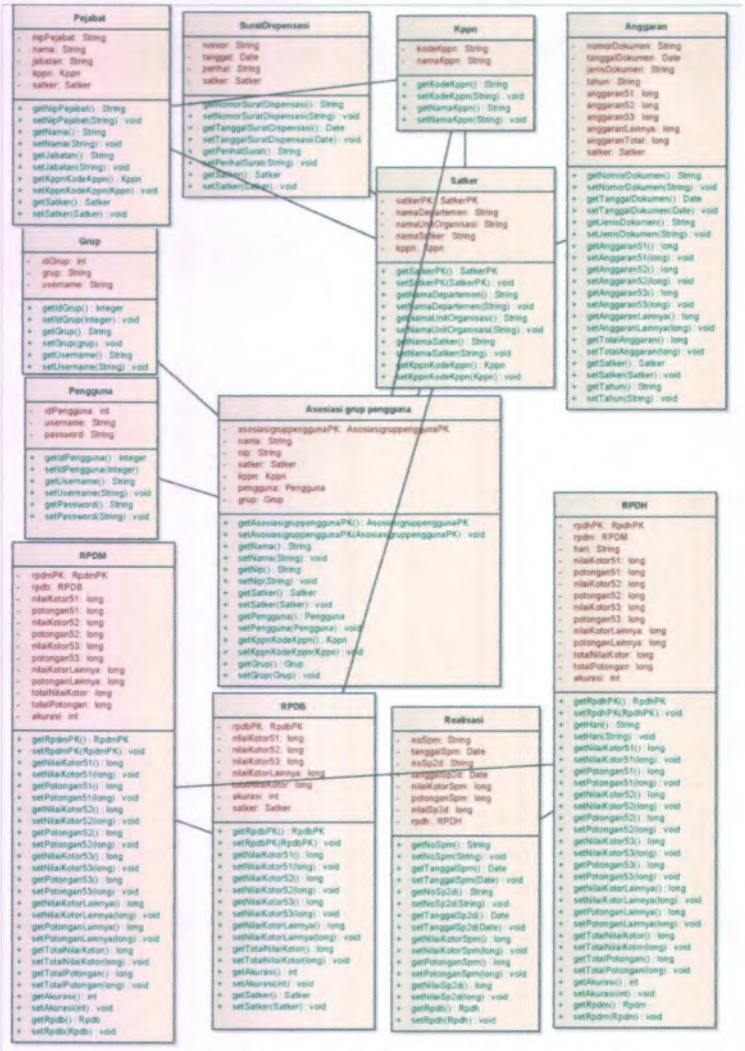
Gambar 10 Contoh Diagram Robustness



Gambar 11 Contoh Diagram Sequences

4.3.2. Diagram Class

Diagram class merupakan domain model yang telah disesuaikan dan diperinci melalui pembuatan diagram *robustness* dan diagram *sequences*. Diagram class mengambarkan kelas-kelas objek java yang akan dipakai dalam aplikasi. Diagram class APPD terlampir pada lampiran B, sebagaimana dicontohkan pada gambar 12.



Gambar 12 Diagram Class package entity

Keseluruhan pemetaan desain dinamis ditunjukkan pada tabel 7.

Tabel 7 Pemetaan keseluruhan desain dinamis

Kode fungsional	Kode *use case*	Kode diagram *robustness*	Kode diagram *sequence*
F1	UC07, UC08, UC09	RD07, RD08, RD09	SD07, SD08, SD09
F2	UC10, UC11, UC12, UC14	RD10, RD11, RD12, RD14	SD10, SD11, SD12, SD14
F3	UC15	RD15	SD15
F4	UC13	RD13	SD13
F5	UC16	RD16	SD16
F6	UC17	RD17	SD17
F7	UC18	RD18	SD18
F8	UC19	RD19	SD19
F9	UC25, UC26, UC27	RD25, RD26, RD27	SD25, SD26, SD27
F10	UC20, UC21	RD20, RD21	SD20, SD21
F11	UC22, UC23, UC24	RD22, RD23, RD24	SD22, SD23, SD24
F12	UC01	RD01	SD01
F13	UC28	RD28	SD28
F14	UC02, UC03, UC04, UC05	RD02, RD03, RD04, RD05	SD02, SD03, SD04, SD05

Halaman ini sengaja dikosngkan

BAB V
PEMBUATAN DAN UJI COBA APLIKASI

5.1. Pembuatan Aplikasi

5.1.1. Lingkungan Pembuatan Aplikasi

Lingkungan pembuatan APPD dibagi menjadi dua kelompok yaitu lingkungan perangkat keras dan lingkungan perangkat lunak.

1. Lingkungan perangkat lunak

Perangkat lunak yang digunakan untuk pembuatan APPD ditunjukkan pada tabel 8.

Tabel 8 Lingkungan perangkat lunak

Lingkungan perangkat lunak	
Sistem operasi	Window 7 Profesional 64-bit (6.1 Build 7601)
Bahasa pemrograman	Java 1.7.0_04
Server	GlassFish 3.1.2
Database	MySQL 5.5.27
Editor	NetBeans IDE 7.1.2
Tools lainnya	*MySQL Workbench 6.0, Navicat* 10.0.5
Browser	Mozilla Firefox 28.0

2. Lingkungan perangkat keras

Perangkat keras yang digunakan untuk pembuatan APPD memiliki spesifikasi sebagaimana ditunjukkan pada tabel 9.

5.1.2. Pembuatan Kelas

Pembuatan kelas APPD dilakukan berdasarkan desain aplikasi pada bab sebelumnya dan fungsi pendukungnya. Kelas-kelas tersebut dibuat dengan bahasa pemrograman Java dan disimpan pada file berekstensi .java. Tabel 10 menunjukkan daftar kelas yang terdapat pada APPD.

Tabel 9 Lingkungan perangkat keras

Lingkungan perangkat keras	
Processor	Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz 2.30GHz
RAM	4GB
VGA	Inter(R) HD Graphic Family

Tabel 10 Daftar kelas aplikasi

Package	Kelas	Nama file	Fungsi
Entity	Anggaran	Anggaran.java	Sebagai representasi tabel di database
	Asosiasi grup pengguna	Asosiasigruppenggu*na.java*	
	Grup	Grup.java	
	Kppn	Kppn.java	
	Pejabat	Pejabat.java	
	Pengguna	Pengguna.java	
	Realisasi	Realisasi.java	
	RPDB	Rpdb.java	
	RPDH	Rpdh.java	
	RPDM	Rpdm.java	
	Satker	Satker.java	
	Surat dispensasi	Suratdispensasi.java	
	ViewRPDB	Viewrpdb.java	Mengambil data dari database
	ViewRPDH	Viewrph.java	
	ViewRPDM	Viewrpdm.java	

			sesuai kebutuhan antar muka
Controller	ControllerServlet	ControllerServlet.java	sebagai servlet untuk grup operator satker
	ControllerServletAdmin	ControllerServletAdmin.java	Sebagai servlet untuk grup admin
	ControllerServletKPPN	ControllerServletKPPN.java	sebagai servlet untuk grup operator KPPN
	ServletAwal	ServletAwal.java	Mengarahkan setiap grup ke servletnya
Session	AnggaranFacade	AnggaranFacade.java	Menjalankan logika bisnis entitasnya
	AsosiasigruppenggunaFacade	AsosiasigruppenggunaFacade.java	
	GrupFacade	GrupFacade.java	
	KppnFacade	KppnFacade.java	
	PejabatFacade	PejabatFacade.java	
	PenggunaFacade	PenggunaFacade.java	
	RealisasiFacade	RealisasiFacade.java	
	RpdbFacade	RpdbFacade.java	
	RpdhFacade	RpdhFacade.java	
	RpdmFacade	RpdmFacade.java	

SatkerFacade	SatkerFacade.java	
SuratdispensasiFacade	SuratdispensasiFacade.java	

5.1.3. Pembuatan Antarmuka

Pembuatan antarmuka APPD berdasar pada *GUI storyboard* yang telah dibuat sebelumnya. Antarmuka dibuat dalam format html, disimpan dalam file dengan ekstensi .jsp dan dikelola oleh beberapa kelas servlet yang telah disebutkan sebelumnya. Antarmuka aplikasi dibentuk dari beberapa file yang ditunjukkan pada tabel 11.

Tabel 11 Daftar file antarmuka

Nama File	Keterangan
Footer.jsp	Segmen JSP yang menyimpan tampilan footer disetiap antarmuka
Header.jsp	Segmen JSP yang menyimpan tampilan header disetiap anatamuka
Template.jsp	Menyimpan tampilan antarmuka untuk grup operator satker
TemplateAdmin.jsp	Menyimpan tampilan antarmuka untuk grup admin
TemplateKPPN.jsp	Menyimpan tampilan antarmuka untuk grup operator KPPN

5.1.1. Pembuatan Fungsionalitas Aplikasi

Pembuatan aplikasi dilakukan Sebagaimana kebutuhan fungsional aplikasi yang telah disebutkan sebelumnya. Untuk melakukan fungsi-fungsi yang bersifat rekam, ubah dan hapus, digunakan method create(entity), edit(entity) dan remove (entity) pada kelas AbstractFacade.java serta entity yang dicakup pada kelas-kelas yang terdapat pada package entity. Sementara untuk fungsi yang bersifat menampilkan, digunakan method-method pada kelas-

kelas yang terdapat pada package session untuk melakukan query database.Berikut merupakan contoh beberapa pembuatan kode fungsi aplikasi.

1. Fungsi mengubah perencanaan pencairan dana bulanan.

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/ubahRencanaBulanan") sistem mengambil data yang dimasukan oleh pengguna untuk menemukan sebuah record pada tabel rpdb di database dengan menggunakan method find(entity) pada kelas entity Rpdb.java. record yang telah diambil ditampung sementara pada kelas entity Rpdb.java. Atribut-atribut entity yang telah ditemukan ditampilkan dengan field-field pada template.jsp, yang sekaligus digunakan untuk menampung nilai baru yang dimasukan pengguna. Pada kelas ControllerServlet.java pada percabangan userPath.equals("/ubahRB") sistem menyesuaikan entity dengan nilai atributnya yang baru dengan method set(atribut). Record tersebut diubah dengan menggunakan method edit(entity). Potongan kode pada ControllerServlet yang menangani fungsi tersebut ditunjukkan pada gambar 13.

```
rpdbPK = new RpdbPK(Integer.parseInt(request.getParameter("isianBulan")),
        tahun, param[0], param[1], param[2]);
rpdb = daftarRPDB.find(rpdbPK);
rpdb.setNilaiKotor51(belanja[0]);
rpdb.setNilaiKotor52(belanja[1]);
rpdb.setNilaiKotor53(belanja[2]);
rpdb.setNilaiKotorLainnya(belanja[3]);
rpdb.setTotalNilaiKotor(belanja[4]);
daftarRPDB.edit(rpdb);
daftarBulanan = daftarRPDB.temukanDgnOrganisasi(tahun, satker.getSatkerPK().getKodeDepartemen(),
        satker.getSatkerPK().getKodeUnitOrganisasi(), satker.getSatkerPK().getKodeSatker());
request.setAttribute("daftarBulanan", daftarBulanan);
cekMenu = 1;
request.setAttribute("cekMenu", cekMenu);
```

Gambar 13 Potongan kode fungsi ubah perencanaan pencairan dana bulanan

2. *Fungsi merekam perencanaan pencairan dana harian.*

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/rekamRH") sistem mendeklarasikan entity Rpdh dan RpdhPK. RpdhPK merupakan primary key dari entity Rpdh. Sistem mengambil data masukan pengguna dari template.jsp dan memasukannya ke atribut entity. Selanjutnya

sistem menyimpan entity sebagai record baru pada tabel rpdh pada database dengan method create(entity). Potongan kode pada ControllerServlet yang menangani fungsi tersebut ditunjukkan pada gambar 14.

```
}
rpdhPK = new RpdhPK(date, temukanBulan
rpdh = new Rpdh(rpdhPK);
rpdh.setNilaiKotor51(nilai[0]);
rpdh.setPotongan51(nilai[1]);
rpdh.setNilaiKotor52(nilai[2]);
rpdh.setPotongan52(nilai[3]);
rpdh.setNilaiKotor53(nilai[4]);
rpdh.setPotongan53(nilai[5]);
rpdh.setNilaiKotorLainnya(nilai[6]);
rpdh.setPotonganLainnya(nilai[7]);
rpdh.setTotalNilaiKotor(nilai[8]);
rpdh.setTotalPotongan(nilai[9]);
rpdh.setHari(temukanHari(date));
daftarRPDH.create(rpdh);
```

Gambar 14 Potongan kode fungsi rekam perencanaan pencairan dana harian

3. Fungsi mengubah perencanaan pencairan dana harian

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/ubahRencanaHarian") sistem mengambil record dari tabel rpdh pada database dengan mengunakan method find(entity) dengan parameter data masukan dari pengguna dan menampungnya pada kelas entity Rpdh.java. Kemudian sistem menampilkan atribut entity ke field-field pada template.jsp yang sekaligus digunakan untuk menampung nilai baru yang dimasukan pengguna

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/ubahRH") sistem mengubah nilai atribut entity dengan data masukan pengguna menggunakan method set(atribut). Sistem menyimpan kembali record dengan nilai baru ke database dengan method edit(entity). Potongan kode pada ControllerServlet yang menangani fungsi ubah perencanaan pencairan dana harian ditunjukkan pada gambar 15.

```
rpdhPK = new RpdhPK(date, temukanBulan(date), temukanMing
rpdh = daftarRPDH.find(rpdhPK);
rpdh.setNilaiKotor51(nilai[0]);
rpdh.setPotongan51(nilai[1]);
rpdh.setNilaiKotor52(nilai[2]);
rpdh.setPotongan52(nilai[3]);
rpdh.setNilaiKotor53(nilai[4]);
rpdh.setPotongan53(nilai[5]);
rpdh.setNilaiKotorLainnya(nilai[6]);
rpdh.setPotonganLainnya(nilai[7]);
rpdh.setTotalNilaiKotor(nilai[8]);
rpdh.setTotalPotongan(nilai[9]);
daftarRPDH.edit(rpdh);
```

Gambar 15 Potongan kode fungsi ubah perencanaan pencairan dana harian

4. *Fungsi menghapus perencanaan pencairan dana harian*

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/hapusRencanaHarian") sistem mengambil data masukan pengguna dan menggunakannya sebagai parameter method find(entity) untuk menemukan sebuah record pada tabel rpdh pada database dan menampungnya pada kelas entity *Rpdh.java. Sistem menghapus record tersebut dari database* dengan method remove(entity). Potongan kode pada ControllerServlet yang menangani fungsi ini ditunjukkan pada gambar 16.

```
rpdhPK = new RpdhPK(date, temukanBulan(date), temukanMingguKe(da
rpdh = daftarRPDH.find(rpdhPK);
daftarRPDH.remove(rpdh);
```

Gambar 16 Potongan kode fungsi hapus perencanaan pencairan dana harian

5. *Fungsi mengakumulasikan perencanaan pencairan dana* harian ke perencanaan dana mingguan.

Fungsi ini dijalankan pada kelas ControllerServlet.java pada percabangan yang merekam, mengubah dan menghapus entity Rpdh. Pada perekaman Rpdh dimana belum terdapat entity Rpdm, maka sistem akan merekam record pada tabel rpdm di *database dengan mengunakan method create(entity). Sementara* pada perekaman Rpdh dimana sudah terdapat Rpdm-nya atau

pada pengubahan dan penghapusan Rpdh, sistem akan mencari record pada tabel rpdm di database dan mengubahnya dengan method find(entity) dan edit(entity). Ketika seluruh Rpdh pada periode sebuah Rpdm dihapus, maka Rpdm tersebut tidak dihapus melainkan diubah seluruh atribut uangnya menjadi 0. Potongan kode pada ControllerServlet yang menangani fungsi ini ditunjukkan pada gambar 17.

```
rpdm = daftarRPDM.find(rpdmPK);
rpdm.setNilaiKotor51(rpdm.getNilaiKotor51() + nilai[0]);
rpdm.setPotongan51(rpdm.getPotongan51() + nilai[1]);
rpdm.setNilaiKotor52(rpdm.getNilaiKotor52() + nilai[2]);
rpdm.setPotongan52(rpdm.getPotongan52() + nilai[3]);
rpdm.setNilaiKotor53(rpdm.getNilaiKotor53() + nilai[4]);
rpdm.setPotongan53(rpdm.getPotongan53() + nilai[5]);
rpdm.setNilaiKotorLainnya(rpdm.getNilaiKotorLainnya() + nil
rpdm.setPotonganLainnya(rpdm.getPotonganLainnya() + nilai[7
rpdm.setTotalNilaiKotor(rpdm.getTotalNilaiKotor() + nilai[8
rpdm.setTotalPotongan(rpdm.getTotalPotongan() + nilai[9]);
daftarRPDM.edit(rpdm);
```

Gambar 17 Potongan kode fungsi mengakumulasikan perencanaan pencairan dana harian ke perencanaan dana mingguan

6. Fungsi merekam data realisasi pada perencanaan pencairan dana harian

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/konfirmRekamRealisasi") sistem mendeklarasikan entity Realisasi. Sistem mengambil data masukan pengguna dari template.jsp dan memasukannya ke atribut entity. Selanjutnya sistem menyimpan entity sebagai record baru pada tabel realisasi pada database dengan method create(entity). Potongan kode pada ControllerServlet yang menangani fungsi rekam realisasi ditunjukkan pada gambar 18.

```
if (!daftarHarian.isEmpty()) {
    rpdh = daftarHarian.get(0);
    if (rpdh.getTotalNilaiKotor() < sementara + Long.parseLong(request.getParameter("i
        //realisasi tidak boleh melebihi perencanaan.
        return;
    }
    realisasi = new Realisasi(request.getParameter("isianNoSPM"));
    realisasi.setRpdh(rpdh);
    realisasi.setTanggalSpm(date);
    sementara2 = request.getParameter("isianTglSP2D");
    d = format.parse(sementara2);
    date = new java.sql.Date(d.getTime());
    realisasi.setTanggalSp2d(date);
    realisasi.setNoSp2d(request.getParameter("isianNoSP2D"));
    realisasi.setNilaiKotorSpm(Long.parseLong(request.getParameter("isianNiKo")));
    realisasi.setPotonganSpm(Long.parseLong(request.getParameter("isianPot")));
    realisasi.setNilaiSp2d(realisasi.getNilaiKotorSpm() - realisasi.getPotonganSpm());
    daftarRealisasi.create(realisasi);
```

Gambar 18 Potongan kode fungsi rekam realisasi

7. Fungsi mengubah data realisasi pada perencanaan pencairan dana harian.

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/ubahRealisasi") sistem mengambil record dari tabel realisasi pada database dengan mengunakan method find(entity) dengan parameter data masukan dari pengguna dan menampungnya pada kelas entity Realisasi.java. Kemudian sistem menampilkan atribut entity ke field-field pada template.jsp yang sekaligus digunakan untuk menampung nilai baru yang dimasukan pengguna

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/konfirmUbahRealisasi") sistem mengubah nilai atribut entity dengan data masukan pengguna menggunakan method set(atribut). Sistem menyimpan kembali record dengan nilai baru ke database dengan method edit(entity). Potongan kode pada ControllerServlet yang menangani fungsi rekam realisasi ditunjukkan pada gambar 19.

```
if (rpdh.getTotalNilaiKotor() < sementara - nilaiLama + re
    //realisasi tidak boleh melebihi perencanaan.
    return;
}
daftarRealisasi.edit(realisasi);
```

Gambar 19 Potongan kode fungsi ubah realisasi

8. Fungsi menghapus data realisasi pada perencanaan pencairan dana harian

Pada kelas ControllerServlet.java pada percabangan userPath.equals("/hapusRealisasi") sistem mengambil data masukan pengguna dan menggunakannya sebagai parameter method find(entity) untuk menemukan sebuah record pada tabel realisasi pada database dan menampungnya pada kelas entity Realisasi.java. Sistem menghapus record tersebut dari database dengan method remove(entity). Potongan kode pada ControllerServlet yang menangani fungsi rekam realisasi ditunjukkan pada gambar 20.

```
) else if (userPath.equals("/hapusRealisasi")) {
    String queryString = request.getQueryString();
    realisasi = daftarRealisasi.find(queryString);
    rpdh = realisasi.getRpdh();
    daftarRealisasi.remove(realisasi);
```

Gambar 20 Potongan kode fungsi hapus realisasi

9. Fungsi menampilkan rekapitulasi perencanaan pencairan dana bulanan.

Fungsi tersebut dijalankan dari kelas ControllerServletKPPN.java pada percabangan userPath.equals("/cariRekapBlnanPerOrg") untuk menampilkan rekapitulasi satu tahun anggaran pada suatu organisasi dan userPath.equals("/cariRekapBlnanPerBln") untuk menampilkan rekapitulasi satu atau beberapa organisasi dalam satu bulan. Organisasi dimaksud dapat berupa departemen, unit organisasi maupun satker.

Untuk percabangan pertama, pada templateKPPN.jsp ditampilkan field departemen, unit organisasi dan satker. Data masukan pengguna akan menjadi parameter untuk pemanggilan method temukanDgnKppnDept(), temukanDgnKppnDeptUnOrg() atau temukanDgnKppnSatker() yang berfungsi untuk mengquery tabel rpdb pada database.

Untuk percabangan kedua, pada templateKPPN.jsp ditampilkan field departemen, unit organisasi, satker dan bulan. Data masukan pengguna akan menjadi parameter untuk pemanggilan method

temukanDgnKppnDeptBln(), temukanDgnKppnDeptUnOrgBln() atau temukanDgnKppnSatkerBln() yang berfungsi untuk mengquery tabel rpdb pada database. Potongan kode pada ControllerServletKPPN yang menangani fungsi tersebut ditunjukkan pada gambar 21.

```
String dept, unOrg, sat;
dept = request.getParameter("dept");
unOrg = request.getParameter("unitOrg");
sat = request.getParameter("satker");
long[][] total = new long[5][12];
if (!dept.equals("") && unOrg.equals("") && sat.equals("")) {
    daftarBulanan = daftarViewRPDB.temukanDgnKppnDept(tahun, kppn, dept);
    total[0] = jmlBlnanS1PerBln(daftarBulanan);
    total[1] = jmlBlnanS2PerBln(daftarBulanan);
    total[2] = jmlBlnanS3PerBln(daftarBulanan);
    total[3] = jmlBlnanLainPerBln(daftarBulanan);
    total[4] = jmlBlnanTotalPerBln(daftarBulanan);
} else if (!dept.equals("") && !unOrg.equals("") && sat.equals("")) {
    daftarBulanan = daftarViewRPDB.temukanDgnKppnDeptUnOrg(tahun, kppn, dept, unOrg);
    total[0] = jmlBlnanS1PerBln(daftarBulanan);
    total[1] = jmlBlnanS2PerBln(daftarBulanan);
    total[2] = jmlBlnanS3PerBln(daftarBulanan);
    total[3] = jmlBlnanLainPerBln(daftarBulanan);
    total[4] = jmlBlnanTotalPerBln(daftarBulanan);
} else if (!dept.equals("") && !unOrg.equals("") && !sat.equals("")) {
    satkerPK = new SatkerPK(dept, unOrg, sat);
    satker = new Satker(satkerPK);
    daftarBulanan = daftarViewRPDB.temukanDgnKppnSatker(tahun, kppn, satker);
    total[0] = jmlBlnanS1PerBln(daftarBulanan);
    total[1] = jmlBlnanS2PerBln(daftarBulanan);
    total[2] = jmlBlnanS3PerBln(daftarBulanan);
    total[3] = jmlBlnanLainPerBln(daftarBulanan);
    total[4] = jmlBlnanTotalPerBln(daftarBulanan);
}
request.setAttribute("total", total);
```

Gambar 21 Potongan kode fungsi menampilkan rekapitulasi perencanaan pencairan dana bulanan.

10. Fungsi menampilkan rekapitulasi perencanaan pencairan dana mingguan.

Fungsi tersebut dijalankan dari kelas ControllerServletKPPN.java pada percabangan userPath.equals("/cariRekapMggan") untuk menampilkan rekapitulasi suatu periode mingguan pada suatu organisasi.

Untuk percabangan pertama, pada templateKPPN.jsp ditampilkan field departemen, unit organisasi, satker, bulan dan minggu. Data masukan pengguna akan menjadi parameter untuk pemanggilan method temukanDgnKppnDeptBln(), temukanDgnKppnDeptUnOrgBln() atau

temukanDgnKppnSatkerBln() yang berfungsi untuk mengquery tabel rpdm pada database. Potongan kode pada ControllerServletKPPN yang menangani fungsi tersebut ditunjukkan pada gambar 22.

```
if (!dept.equals("") && unOrg.equals("") && sat.equals("") && mggKe == 0) {
    daftarMingguan = daftarViewRPDM.temukanDgnKppnDeptBln(tahun, kppn, dept, bln
    for (int i = 0; i < daftarMingguan.size(); i++) {
        for (int j = 1; j <= 4; j++) {
            if (daftarMingguan.get(i).getMingguKe() == j) {
                total[0][j - 1] += daftarMingguan.get(i).getNilaiKotor51();
                total[1][j - 1] += daftarMingguan.get(i).getPotongan51();
                total[2][j - 1] += daftarMingguan.get(i).getNilaiKotor52();
                total[3][j - 1] += daftarMingguan.get(i).getPotongan52();
                total[4][j - 1] += daftarMingguan.get(i).getNilaiKotor53();
                total[5][j - 1] += daftarMingguan.get(i).getPotongan53();
                total[6][j - 1] += daftarMingguan.get(i).getNilaiKotorLainnya();
                total[7][j - 1] += daftarMingguan.get(i).getPotonganLainnya();
                total[8][j - 1] += daftarMingguan.get(i).getTotalNilaiKotor();
                total[9][j - 1] += daftarMingguan.get(i).getTotalPotongan();
            }
        }
    }
    request.setAttribute("total", total);
```

Gambar 22 Potongan kode fungsi menampilkan rekapitulasi perencanaan pencairan dana mingguan

11. Fungsi menampilkan rekapitulasi perencanaan pencairan dana harian.

Fungsi tersebut dijalankan dari kelas ControllerServletKPPN.java pada percabangan userPath.equals("/cariRekapHranOrgPerMgg") untuk menampilkan rekapitulasi satu minggu kerja pada suatu organisasi dan userPath.equals("/cariRekapHranPerTgl") untuk menampilkan rekapitulasi satu atau beberapa organisasi dalam satu hari.

Untuk percabangan pertama, pada templateKPPN.jsp ditampilkan field departemen, unit organisasi, satker, bulan dan minggu kerja. Data masukan pengguna akan menjadi parameter untuk pemanggilan method temukanDgnKppnDeptTgl(), temukanDgnKppnDeptUnOrgTgl() atau temukanDgnKppnSatkerTgl() yang berfungsi untuk mengquery tabel rpdh pada database.

Untuk percabangan kedua, pada templateKPPN.jsp ditampilkan field departemen, unit organisasi, satker dan tanggal. Data masukan pengguna akan menjadi parameter untuk pemanggilan method temukanDgnKppnDeptTgl(),

temukanDgnKppnDeptUnOrgTgl() atau temukanDgnKppnSatkerTgl() yang berfungsi untuk mengquery tabel rpdh pada database. Potongan kode pada ControllerServletKPPN yang menangani fungsi tersebut ditunjukkan pada gambar 23.

```
if (!dept.equals("") && unOrg.equals("") && sat.equals("")) {
    daftarHarian = daftarViewRPDH.temukanDgnKppnDeptTgl(tahun, kppn, dept,
} else if (!dept.equals("") && !unOrg.equals("") && sat.equals("")) {
    daftarHarian = daftarViewRPDH.temukanDgnKppnDeptUnOrgTgl(tahun, kppn, d
} else if (!dept.equals("") && !unOrg.equals("") && !sat.equals("")) {
    satkerPK = new SatkerPK(dept, unOrg, sat);
    satker = new Satker(satkerPK);
    daftarHarian = daftarViewRPDH.temukanDgnKppnSatkerTgl(tahun, kppn, satk
}
request.setAttribute("daftarHarian", daftarHarian);
```

Gambar 23 Potongan kode fungsi menampilkan rekapitulasi perencanaan pencairan dana mingguan

12. Memperhitungkan akurasi perencanaan pencairan dana

Perhitungan akurasi dilakukan pada kelas ControllerServlet.java pada setiap terjadi rekam, ubah dan hapus realisasi, yaitu pada percabangan userPath.equals("/konfirmUbahRealisasi"), userPath.equals("/konfirmRekamRealisasi") dan userPath.equals("/hapusRealisasi"). Akurasi dinilai dengan persentase dari nilai realisasi dibandingkan perencanaan. Nilai kolom akurasi pada setiap tabel rpdb, rpdm dan rpdh yang terkait diubah sesuai dengan perubahan realisasi.

Untuk menampilkan nilai akurasi digunakan kelas ControllerServletKPPN.java pada percabangan userPath.equals("/akurasiPerencanaanBulanan"), userPath.equals("/akurasiPerencanaanMingguan") dan userPath.equals("/akurasiPerencanaanHarian"). Akurasi harian ditampilkan per hari kerja pada minggu yang dipilih, sementara akurasi mingguan ditampilkan per minggu pada bulan yang dipilih dan akurasi bulanan ditampilkan per bulan selama tahun anggaran berjalan. Potongan kode pada ControllerServletKPPN yang menangani fungsi tersebut ditunjukkan pada gambar 24.

```
for (int j = 0; j < daftarBulanan.size(); j++) {
    if (daftarBulanan.get(j).getBulan() == 1) {
        akurasiPerTahun.setJanuari(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 2) {
        akurasiPerTahun.setFebruari(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 3) {
        akurasiPerTahun.setMaret(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 4) {
        akurasiPerTahun.setApril(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 5) {
        akurasiPerTahun.setMei(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 6) {
        akurasiPerTahun.setJuni(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 7) {
        akurasiPerTahun.setJuli(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 8) {
        akurasiPerTahun.setAgustus(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 9) {
        akurasiPerTahun.setSeptember(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 10) {
        akurasiPerTahun.setOktober(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 11) {
        akurasiPerTahun.setNopember(daftarBulanan.get(j).getAkurasi());
    } else if (daftarBulanan.get(j).getBulan() == 12) {
        akurasiPerTahun.setDesember(daftarBulanan.get(j).getAkurasi());
    }
```

Gambar 24 Potongan kode fungsi memperhitungkan akurasi perencanaan pencairan dana bulanan

13. Memperhitungkan peramalan perencanaan pencairan dana.

Perhitungan peramalan berjalan pada kelas ControllerServletKPPN.java. Peramalan terbagi atas tiga metode yaitu: moving average, simple exponential smoothing dan holt.

Perhitungan metode moving average berjalan pada percabangan userPath.equals("/movingAverage"). Perhitungan dilakukan dengan menjumlahkan perencanaan pada bulan-bulan terakhir selama periode waktu yang dipilih dibagi dengan periodenya. Apabila nilai periode lebih besar jumlah bulan yang sudah terlewati pada tahun anggaran saat ini, maka digunakan data tahun anggaran sebelumnya.

Perhitungan metode simple exponential smooting berjalan pada percabangan userPath.equals("/simpleExpoSmoot"). Perhitungan metode ini menggunakan data perencanaan dari 2 tahun anggaran yang telah berlalu dan tahun anggaran berjalan. Estimasi awal level yang pada kode diwakili oleh atribut l0, dihitung dengan menjumlahkan perencanaan bulanan dari 2 tahun anggaran yang lalu hingga bulan berjalan dibagi dengan jumlah bulan. Nilai estimasi awal level tersebut akan menjadi peramalan bulan pertama. Untuk mengetahui peramalan selanjutnya digunakan perhitungan secara berulang dengan mengkalikan nilai atribut

alpha dengan perencanaan pada bulan yang dihitung ditambah dengan hasil perkalian 1-alpha dengan nilai atribut l0.

Perhitungan metode holt berjalan pada percabangan userPath.equals("/holtMethod"). Perhitungan dengan metode ini menggunakan data perencanaan dari 2 tahun anggaran yang telah berlalu dan tahun anggaran berjalan. Estimasi awal trend yang pada kode diwakili oleh atribut t0, merupakan nilai absolut dari pengurangan rata-rata perencanaan bulanan tahun anggaran dua tahun yang lalu degan rata-rata perencanaan bulanan tahun anggaran satu tahun yang lalu kemudian dibagi dua belas bulan. Estimasi awal level pada kode diwakili oleh atribut l0, dihitung dengan mengkalikan nilai estimasi awal trend dengan 6,5. Nilai *trend berikutnya dihitung dengan perkalian alpha dan* perencanaan ditambah dengan perkalian 1-alpha dan peramalan bulan sebelumnya. Nilai level berikutnya dihitung dengan perkalian beta dan 1-level bulan sebelumnya ditambahkan dengan perkalian 1-beta dan trend bulan sebelumnya. tren dan level pada bulan yang belum memiliki realisasi nilainya sama dengan trend *dan level pada bulan terakhir yang memiliki realisasi. Nilai* peramalan pada bulan yang telah memiliki realisasi merupakan penambahan dari trend dan level pada bulan yang diramalkan. Sementara untuk nilai peramalan pada bulan yang belum memiliki realisasi nilainya merupakan penambahan level dengan hasil perkalian trend dan jumlah bulan sejak bulan terakhir yang *memiliki realisasi. Potongan kode pada ControllerServletKPPN* yang menangani fungsi tersebut ditunjukkan pada gambar 25.

```
long hasil = 0L;
for (int i = 0; i < jmlPerBln.length; i++) {
    hasil += jmlPerBln[i];
}
hasil = hasil / jmlPerBln.length;
request.setAttribute("hasil", hasil);
```

Gambar 25 Potongan kode fungsi memperhitungkan peramalan dengan metode moving average

14. Memungkinkan administrator untuk merekam pagu anggaran satker.

Fungsi rekam anggaran berjalan kelas ControllerServletAdmin.java pada percabangan userPath.equals("/konfRekamAnggaran") sistem mendeklarasikan entity Anggaran. Sistem mengambil data masukan pengguna dari templateAdmin.jsp dan memasukannya ke atribut entity. Selanjutnya sistem menyimpan entity sebagai record baru pada tabel anggaran pada database dengan method create(entity). Potongan kode pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 26.

```
anggaran = new Anggaran(noDok);
anggaran.setSatker(satker);
anggaran.setTanggalDokumen(tglDok);
anggaran.setTahun(tahun);
anggaran.setJenisDokumen(jenDok);
anggaran.setAnggaran51(niKo51);
anggaran.setAnggaran52(niKo52);
anggaran.setAnggaran53(niKo53);
anggaran.setAnggaranLainnya(niKoLa);
long total = niKo51 + niKo52 + niKo53 +
anggaran.setTotalAnggaran(total);
daftarAnggaran.create(anggaran);
```

Gambar 26 Potongan kode fungsi merekam pagu anggaran

15. Memungkinkan administrator untuk mengubah pagu anggaran satker.

Pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/ubahAnggaran") sistem mengambil record dari tabel anggaran pada database dengan mengunakan method find(entity) dengan parameter data masukan dari pengguna dan menampungnya pada kelas entity Anggaran.java. Kemudian sistem menampilkan atribut entity ke field-field pada templateAdmin.jsp yang sekaligus digunakan untuk menampung nilai baru yang dimasukan pengguna

Pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/konfUbahAnggaran") sistem mengubah nilai atribut entity dengan data masukan pengguna menggunakan method set(atribut). Sistem menyimpan kembali record dengan nilai baru ke database dengan method edit(entity). Potongan kode

pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 27.

```
anggaran = daftarAnggaran.temukanDgnSatkerPKTahunNoDok(satkerPK,
long niKo51 = Long.parseLong(request.getParameter("niKo51"));
long niKo52 = Long.parseLong(request.getParameter("niKo52"));
long niKo53 = Long.parseLong(request.getParameter("niKo53"));
long niKoLa = Long.parseLong(request.getParameter("niKoLa"));
anggaran.setAnggaran51(niKo51);
anggaran.setAnggaran52(niKo52);
anggaran.setAnggaran53(niKo53);
anggaran.setAnggaranLainnya(niKoLa);
long total = niKo51 + niKo52 + niKo53 + niKoLa;
anggaran.setTotalAnggaran(total);
daftarAnggaran.edit(anggaran);
```

Gambar 27 Potongan kode fungsi ubah pagu anggaran

16. Memungkinkan administrator untuk menghapus pagu anggaran satker.

Pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/hapusAnggaran") sistem mengambil data masukan pengguna dan menggunakannya sebagai parameter method find(entity) untuk menemukan sebuah record pada tabel anggaran pada database dan menampungnya pada kelas entity Anggaran.java. Sistem menghapus record tersebut dari database dengan method remove(entity). Potongan kode pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 28.

```
anggaran = daftarAnggaran.temukanDgnSatkerPKTahunNoDok(satkerPK, tah
daftarAnggaran.remove(anggaran);
```

Gambar 28 Potongan kode fungsi hapus pagu anggaran

17. Memungkinkan administrator untuk merekam pengguna.

Fungsi ini berjalan pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/rekamPengguna"), userPath.equals("/konfRekamPengguna1") dan userPath.equals("/konfRekamPengguna2"). Sistem mendeklarasikan entity Pengguna dan Grup, mengkonversi password, men-set atribut entity sesuai dengan data masukkan dari pengguna dan membuat record baru pada tabel pengguna dan grup di database. Sistem kemudian mendeklarasikan AsosiasiGrupPengguna dengan primary key dari id pengguna dan

grup, men-set atribut entity sesuai dengan data masukkan dari pengguna dan membuat record baru pada tabel asosiasigruppengguna di database. Potongan kode pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 29.

```
for (int i = 0; i < daftarUserAppdLokal.size(); i++) {
    if (maksId < daftarUserAppdLokal.get(i).getIdPengguna()
        maksId = daftarUserAppdLokal.get(i).getIdPengguna()
    }
}
userAppd = new Pengguna(maksId + 1);
userAppd.setUsername(noNip);
userAppd.setPassword(pass2);
daftarUserAppd.create(userAppd);
//membuat grup
List<Grup> daftarGrupLokal = daftarGrup.findAll();
maksId = 0;
for (int i = 0; i < daftarGrupLokal.size(); i++) {
    if (maksId < daftarGrupLokal.get(i).getIdGrup()) {
        maksId = daftarGrupLokal.get(i).getIdGrup();
    }
}
grup = new Grup(maksId + 1);
grup.setGrup(peran);
grup.setUsername(noNip);
daftarGrup.create(grup);
```

Gambar 29 Potongan kode fungsi rekam pengguna

18. Memungkinkan administrator untuk menghapus pengguna. Fungsi ini berjalan pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/hapusPengguna"). Sistem mencari dan menghapus record pada tabel asosiasigruppengguna pada database sesuai id grup dan id pengguna yang dimasukkan pengguna. Kemudian sistem mencari dan menghapus record pada tabel pengguna dan grup sesuai dengan id grup dan id pengguna tersebut. Potongan kode pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 30.

```
//menghapus asosiasi
AsosiasigruppenggunaPK asosiasigruppenggunaPK =
Asosiasigruppengguna asosiasigruppengguna = daft
daftarPengguna.remove(asosiasigruppengguna);
//menghapus grup
grup = daftarGrup.find(idGrp);
daftarGrup.remove(grup);
//menghapus pengguna
userAppd = daftarUserAppd.find(idUsnm);
daftarUserAppd.remove(userAppd);
```

Gambar 30 Potongan kode fungsi hapus pengguna

19. Memungkinkan administrator untuk merekam satker.
Fungsi ini berjalan pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/konfRekamSatker"). Sistem mendeklarasikan kelas entity SatkerPK dan Satker, dimana SatkerPK merupakan primary key dari Satker. Sistem men-set atribut entity sesuai dengan data masukkan dari pengguna dan *membuat record baru pada tabel satker pada database. Potongan* kode pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 31.

```
SatkerPK satkerPK = new SatkerPK(koDept, koUnOrg, koSat);
satker = new Satker(satkerPK);
satker.setNamaDepartemen(naDept);
satker.setNamaUnitOrganisasi(naUnOrg);
satker.setNamaSatker(naSat);
kppn = daftarKppn.find(koKppn);
satker.setKppnKodeKppn(kppn);
daftarSatker.create(satker);
```

Gambar 31 Potongan kode fungsi rekam satker

20. Memungkinkan administrator untuk mengubah satker.
Fungsi ini berjalan pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/konfUbahSatker"). Sistem mencari record pada tabel satker di database dengan sesuai dengan data masukan dari pengguna kemudian menampungnya pada kelas entity Satker. Sistem men-set atribut entity sesuai dengan data masukkan dari pengguna dan menyimpan kembali *record pada tabel satker pada database. Potongan kode pada* ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 32.

```
satker = daftarSatker.find(satkerPK);
satker.setNamaDepartemen(naDept);
satker.setNamaUnitOrganisasi(naUnOrg);
satker.setNamaSatker(naSat);
daftarSatker.edit(satker);
```

Gambar 32 Potongan kode fungsi ubah satker

21. Memungkinkan administrator untuk menghapus satker.
Fungsi ini berjalan pada kelas ControllerServletAdmin.java pada percabangan userPath.equals("/hapusSatker"). Sistem mencari record pada tabel satker di database dengan sesuai dengan data

masukan dari pengguna dengan method find(entity) kemudian menghapus record tersebut dari database dengan method remove(entity). Potongan kode pada ControllerServletAdmin yang menangani fungsi tersebut ditunjukkan pada gambar 33.

```
//mengecek eksistensi realisasi, jika ekstis tidak boleh dihapus
daftarReal = daftarRealisasi.temukanDgnSatkerPKThn(satkerPK, tahun);
if (daftarReal.size() == 0) {
    daftarBulanan = daftarRPDB.temukanDgnSatker(tahun, satker);
    if (!daftarBulanan.isEmpty()) {
        for (int i = 0; i < daftarBulanan.size(); i++) {
            rpdb = daftarBulanan.get(i);
            daftarMingguan = daftarRPDM.temukanDgnRpdbPK(rpdb.getRpdbPK());
            if (!daftarMingguan.isEmpty()) {
                for (int j = 0; j < daftarMingguan.size(); j++) {
                    rpdm = daftarMingguan.get(j);
                    daftarHarian = daftarRPDH.temukanDgnRpdmPK(rpdm.getRpdmPK(
                    if (!daftarHarian.isEmpty()) {
                        for (int k = 0; k < daftarHarian.size(); k++) {
                            rpdh = daftarHarian.get(k);
                            daftarRPDH.remove(rpdh);
                        }
                    }
                    daftarRPDM.remove(rpdm);
                }
            }
            daftarRPDB.remove(rpdb);
        }
    }
    daftarSatker.remove(satker);
```

Gambar 33 Potongan kode fungsi ubah satker

5.2. Uji Coba Aplikasi

Uji coba dilakukan untuk mengevaluasi operasi dan ketahanan aplikasi. Lingkungan uji coba APPD sama dengan lingkungan pembuatan aplikasinya.

5.2.1. Blackbox testing

Blackbox testing merupakan pengujian terhadap fungsional aplikasi. Terdapat beberapa *tool* yang dapat digunakan untuk melakukan pengujian yang bersifat blackbox testing, misalnya Jmeter Kit dan Acunetix.

1. Uji Coba dengan Jmeter Kit

Pengujian dengan Jmeter Kit digunakan sebagai *black box* testing. Pengujian pertama dilakukan dengan menjalankan setiap usecase pada aplikasi dibawah pantauan Jmeter Kit. Tujuan pengujian ini

adalah sebagai uji fungsional aplikasi. Hasil pengujian usecase ditampilkan pada table 12.

Tabel 12 Hasil pengujian use case dengan Jmeter Kit via localhost

Usecase	Sample	Error	Throughput	Kb/sec	Avg byte
Seluruh UC pada package operator satker	154	0%	88,8/sec	589,64	6798,6
Seluruh UC pada package operator KPPN	200	0%	139,4/sec	1021,86	7507,8
Seluruh UC pada package admin	90	0%	46,2/secs	309,59	6858,1

Pada tabel 10, throughput adalah nilai request per satuan waktu, kb/sec adalah nilai throughput dalam kbps, dan avg byte adalah rata-rata ukuran response sample dalam byte. Hubungan ketiganya adalah $kb/sec = (throughput \times avg\ byte)/1024$. Dari tabel tersebut dapat diketahui bahwa pengujian pada ke tiga package memiliki tingkat error 0%. Hal itu menunjukkan bahwa tidak ditemukan kesalahan dalam aplikasi. Nilai throughput yang paling tinggi menunjukkan performa aplikasi pada package operator kppn merupakan yang terbaik.

Pengujian kedua dilakukan dengan metode *stress testing*. Pengujian dengan metode ini dilakukan dalam dua tahap, tahap pertama dengan menjalankan aplikasi pada JMeter dengan skenario digunakan dalam 99 pengguna secara bersamaan via localhost. Tujuan pengujian ini adalah mengetahui ketahanan aplikasi terhadap penggunaan atau volume kerja yang sangat besar. Hasil *stress testing* pertama dengan Jmeter Kit ditampilkan pada table 13.

Tabel 13 Hasil stress testing Jmeter Kit dengan 99 pengguna via localhost

Usecase	users	sample	error	throughput	kb/sec	avg byte
Seluruh UC pada package operator satker	99	6930	0%	244,9/sec	7148,36	29895,5
Seluruh UC pada package operator KPPN	99	11682	0%	153,2/sec	4695,00	31379,6
Seluruh UC pada package admin	99	12870	0%	95,1/sec	2932,96	31579,3

Dari tabel 13 dapat diketahui bahwa pengujian pada ke tiga package memiliki tingkat error 0%. Hal itu menunjukkan bahwa penggunaan aplikasi oleh 99 pengguna secara bersamaan tidak mengakibatkan munculnya kesalahan dalam aplikasi. Pada penggunaan secara masal, performa aplikasi pada package operator satker merupakan yang terbaik.

Pengujian metode *stress testing* tahap kedua dilakukan dengan mengakses server aplikasi dengan JMeter pada lima komputer klien via LAN. Pengujian dilaksanakan dengan menggunakan 5 buah komputer dan LAN pada Laboratorium Pemrograman Sistem Informasi ITS. Spesifikasi komputer yang digunakan dalam pengujian ditampikan pada tabel 14.

Tabel 14 Spesifikasi Komputer Klien

	Sistem Operasi	Proseccor	RAM	VGA
Server	Window 7 Profesional	Intel(R) Core(TM) i5-2410 @2.30GHz	4096MB	1760MB
Klien 1	Window XP Profesional	Intel Core i5-3570 @3,40GHz	3278MB	1024MB
Klien 2	Window XP Profesional	Intel Core i5-3570 @3,40GHz	3278MB	1024MB
Klien 3	Window XP Profesional	Intel Core i5-3570 @3,40GHz	3278MB	1024MB
Klien 4	Window XP Profesional	Intel Core i5-3570 @3,40GHz	3278MB	1024MB
Klien 5	Window XP Profesional	Intel Core i5-2400 @3,10GHz	3494MB	1024MB

Dilakukan pengujian melalui seluruh komputer klien dengan Jmeter secara bersamaan dan berkelanjutan. Jumlah treat untuk setiap klien adalah 1 dan di iterasi sebanyak 500 kali. Tujuan pengujian ini adalah mengetahui pemerataan akses aplikasi dari setiap klien. Hasil *stress testing* kedua dengan Jmeter Kit ditampilkan pada table 15.

Dari tabel 15 dapat diketahui bahwa ketika aplikasi diakses secara bersamaan, kb/sec yang didapatkan dari setiap klien hanya sedikit berbeda. Ketika dijalankan dengan 5 klien aplikasi tidak mengalami error.

Tabel 15 Hasil stress testing Jmeter manual via LAN

	sample	error	kb/sec
Klien 1	500	0,00%	5,81
Klien 2	500	0,00%	5,56
Klien 3	500	0,00%	5,48
Klien 4	*500*	*0,00%*	*6,19*
Klien 5	500	0,00%	6,01

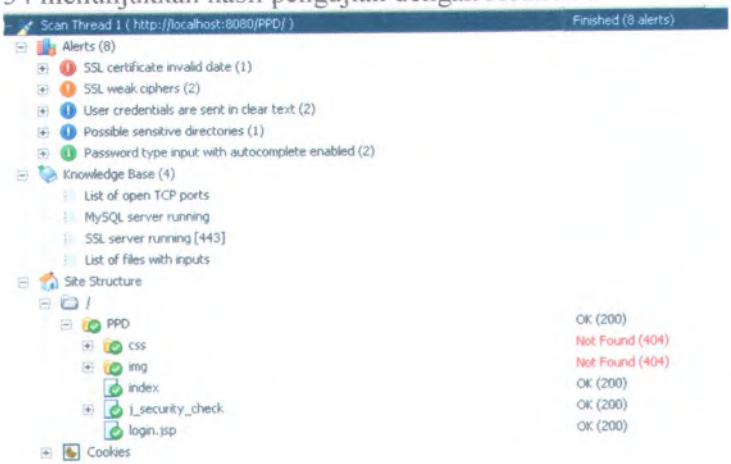
Selanjutnya dilakukan *distributed stress testing* dengan klien 2, klien 3 dan klien 5 menggunakan Jmeter dan Java Remote Method Invocation (RMI). Uji coba menggunakan sejumlah tread sekaligus dan tidak diiterasi. Tujuan pengujian ini adalah untuk mengetahui batas ketahanan maksimal aplikasi. Hasil uji coba dengan *distributed stress testing* ditampilkan pada tabel 16.
Dari tabel 16 ditunjukkan bahwa error muncul mulai dari 510 sample per klien yaitu sebesar 14,58%.

Tabel 16 hasil stress testing Jmeter dengan RMI via LAN

Sample/klien	Total sample	Error	Kb/sec
450	1350	0,00%	1,43
500	1500	0,00%	1,58
505	1515	0,00%	1,60
507	1521	0,00%	1,61
508	1524	0,00%	1,61
509	1527	0,00%	1,61
510	1530	14,58%	1,60
600	1800	41,17%	1,83

2. Uji coba dengan Acunetix

Pengujian dengan Acunetix digunakan sebagai *black box* testing. Yang sifatnya security testing. Pengujian ini bertujuan untuk mengetahui ketahanan aplikasi dari penetrasi eksternal. Gambar 34 menunjukkan hasil pengujian dengan Acunetix.



Gambar 34 Hasil pengujian dengan Acunetix

Dari gambar 34 dapat disimpulkan bahwa tidak ditemukan kemungkinan bahwa aplikasi dapat dipenetrasi dengan metode digunakan saat ini, misalnya XSS, SQL injection, RFI dan LFI.

Hasil scanning pada site structure menunjukkan bahwa website tidak dapat diakses sebelum melakukan login. Salah satu keunggulan APPD dalam hal ini adalah tidak terdapat *method* doGet yang biasanya merupakan akses penetrasi, sebelum pengguna melakukan login.

5.2.2. *Whitebox testing*

Whitebox testing merupakan pengujian terhadap struktur aplikasi. Terdapat beberapa *tool* yang dapat digunakan untuk melakukan pengujian yang bersifat white box testing, misalnya JUnit.

Uji Coba dengan JUnit.
Pengujian dengan JUnit dilakukan dengan menguji setiap *method* pada setiap kelas pada APPD. Tujuannnya untuk memastikan kebenaran kode disetiap *method*. Gambar 35 menunjukkan salah satu tampilan hasil pengujian dengan JUnit.
Tidak semua kelas dapat diuji dengan JUnit, mekanisme pada *java bean membuat kelas-kelasnya sulit untuk diuji dengan JUnit.* Keseluruhan hasil pengujian dengan JUnit ditunjukkan dalam table 17.
Dari tabel 17 ditunjukan bahwa seluruh kelas yang terdapat pada tabel lolos pengujian dengan JUnit.

Gambar 35 Contoh tampilan hasil pengujian dengan JUnit

Tabel 17 Hasil pengujian dengan JUnit

Kelas	Passed	Kelas	Passed
Anggaran.java	100%	Asosiasigruppenggguna.java	100%
Asosiasigruppengguna PK.java	100%	Grup	100%
Kppn	100%	Pejabat	100%
Pengguna	100%	Realisasi	100%
Rpdb	100%	RpdbPK	100%
Rpdh	100%	RpdhPK	100%
Rpdm	100%	RpdmPK	100%
Satker	100%	SatkerPK	100%
Suratdispensasi	100%	Viewrpdb	100%
Viewrpdm	100%	Viewrpdh	100%

Halaman ini sengaja dikosongkan

BAB VI
KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dalam menyusun tugas akhir ini, penulis menyimpulkan beberapa hal:

1. APPD telah berjalan dengan baik hal ini ditunjukan dengan keberhasilan seluruh uji coba aplikasi, baik berupa *white box testing* maupun *black box testing* sebagaimana telah dijelaskan pada Bab V Pembuatan dan Uji Coba Aplikasi Sub Bab Uji Coba Aplikasi.
2. Desain APPD sesuai dengan proses bisnis perencanaan kas, karena *use case* dan kebutuhan fungsional aplikasi disusun sesuai dengan peraturan yang berlaku mengenai perencanaan kas sebagaimana dijelaskan pada Bab IV Desain Aplikasi Sub Bab Identifikasi Kebutuhan.
3. APPD dapat memudahkan pengguna aplikasi, karena seluruh penghitungan dilakukan secara otomatis oleh aplikasi. Selain itu, APPD juga memudahkan pengguna karena menghilangkan pekerjaan klerikal, proses bertemu pihak satker dan kppn dan redundansi data karena berjalan secara online dengan satu database.

6.2. Saran

Beberapa saran yang dapat dipertimbangkan untuk penelitian lebih lanjut adalah sebagai berikut:

1. Diperlukan pengembangan APPD agar dapat mengakomodir pengeluaran negara di luar belanja negara, misalnya: dana bagi hasil (DBH), dana alokasi umum (DAU) dan Dana alokasi khusus (DAK).

2. Diperlukan pengembangan *APPD* agar dapat mengakomodir penerimaan negara baik dari sektor pajak maupun penerimaan negara bukan pajak (PNBP).

DAFTAR PUSTAKA

Bergsten, H., 2002. *JavaServer Pages*. Sebastopol: O'Reilly & Associates, Inc.

Deitel, P.J. & Deitel, H.M., 2007. *Java How to Program*. New Jersey: Pearson Education, Inc.

Direktur Jenderal Perbendaharaan, 2005. Peraturan Direktur Jenderal Perbendaharaan Nomor PER-66/PB/2005 tentang Mekanisme Pembayaran Atas Beban Anggaran Pendapatan Dan Belanja Negara., 2005.

Direktur Jenderal Perbendaharaan, 2010. Peraturan Direktur Jenderal Perbendaharaan Nomor PER-03/PB/2010 Tentang Perkiraan Penarikan Dana Harian Satuan Kerja dan Perkiraan Pencairan Dana Harian Kantor Pelayanan Perbendaharaan Negara., 2010.

Direktur Jenderal Perbendaharaan, 2011. Keputusan Direktur Jenderal Perbendaharaan nomor Kep-163/PB/2011 tentang perubahan atas Kep-185/PB/2010 tentang standar prosedur operasi/standard operating procedures di lingkungan instansi vertikal Direktorat Jenderal Perbendaharaan., 2011.

DuBois, P., 2009. *MySQL*. New Jersey: Pearson Education, Inc.

Hunter, J. & Crawford, W., 1998. *Java Servlet Programming*. Sebastopol: O'Reilly & Associates, Inc.

Kumaradewi, K.A. & Syafruddin, M., 2013. Analisis Minat Penggunaan Aplikasi Forecasting Satker Dalam Rangka Perencanaan Kas. *Diponegoro Journal of Acounting*, 2.

Menteri Keuangan, 2009. Peraturan Menteri Keuangan Nomor 195/PMK/2009 Tentang Perencanaan Kas., 2009.

Myers, G.J., 2004. *The Art of Software Testing*. New Jersey: John Wiley & Sons, Inc.

Presiden Republik Indonesia, 2004. Undang-undang Nomor 1 Tahun 2004 tentang Perbendaharaan Negara., 2004.

Ratzmann, M. & De Young, C., 2003. *Galileo Computing Software Testing and Internationalization*. Salt Lake City: Lemoine International, inc.

Rosenberg, D. & Stephens, M., 2007. *Use Case Driven Object Modeling with UML: Theory and Practice*. Berkeley: Apress.

Schwartz, B. et al., 2008. *High Performance MySQL*. Sebastopol: O'Reilly Media, Inc.

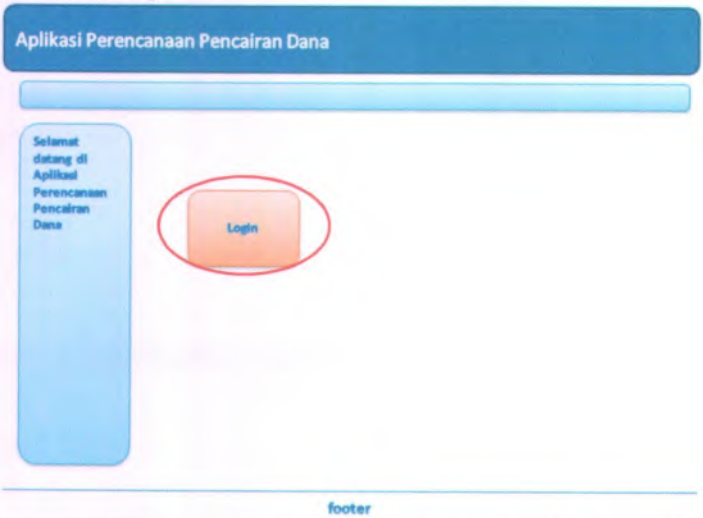
Trent, S. et al., 2008. Performance Comparison of PHP and JSP as Server-Side Scripting Languages. *Middleware 2008*.

LAMPIRAN A
GUI STORYBOARD

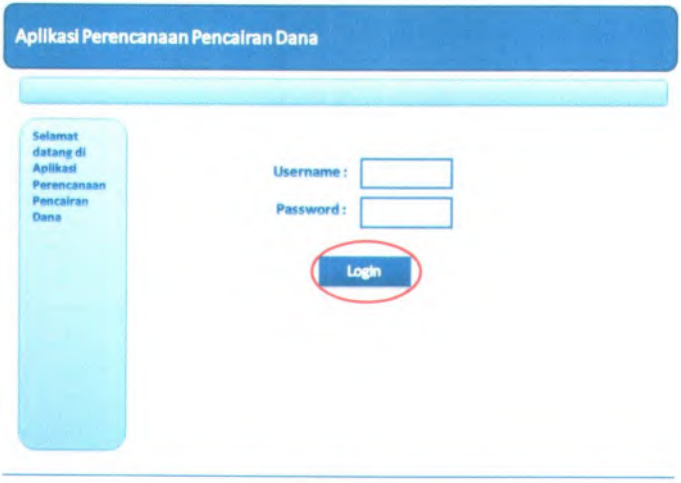
Catatan: dalam seluruh GUI Storyboard berikut, aktivitas menampilkan kotak konfirmasi pada setiap transaksi yang membutuhkannya dihilangkan, dengan tujuan mempersingkat storyboard.

A. Storyboard operator satker : Login

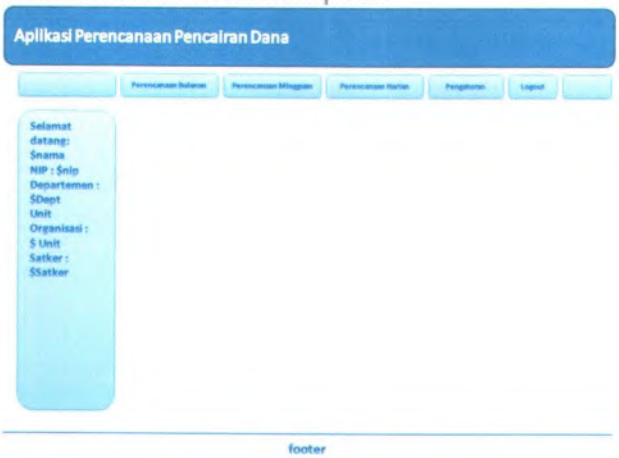
1. Pengguna membuka aplikasi di web browser dan menekan tombol login



2. Pengguna memasukkan username dan password dan menekan tombol login

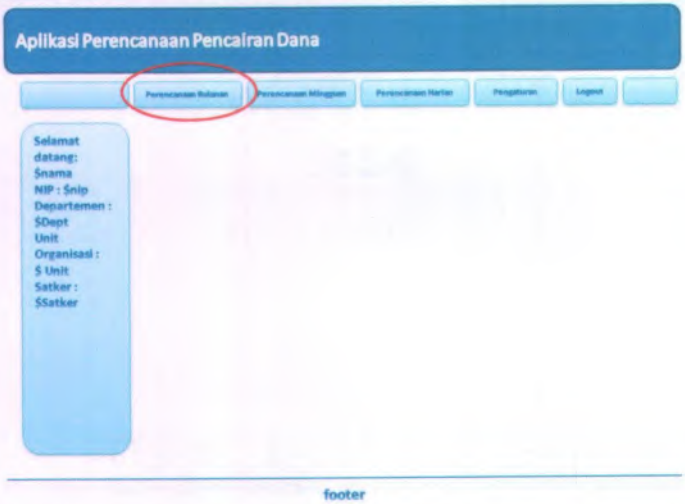


3. Halaman utama ditampilkan

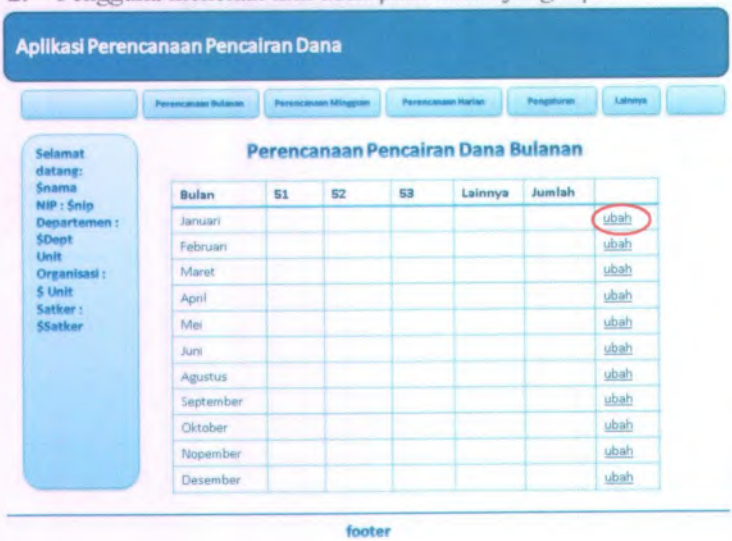


B. Storyboard operator satker : Mengubah perencanaan bulanan

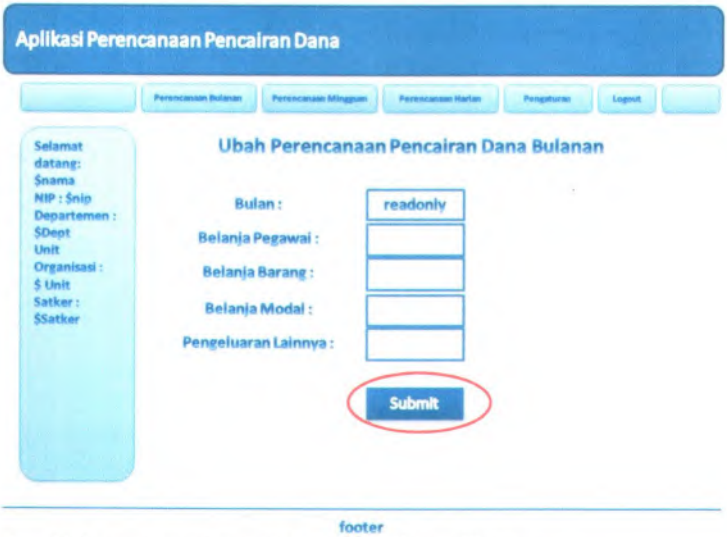
1. Pengguna menekan link perencanaan bulanan



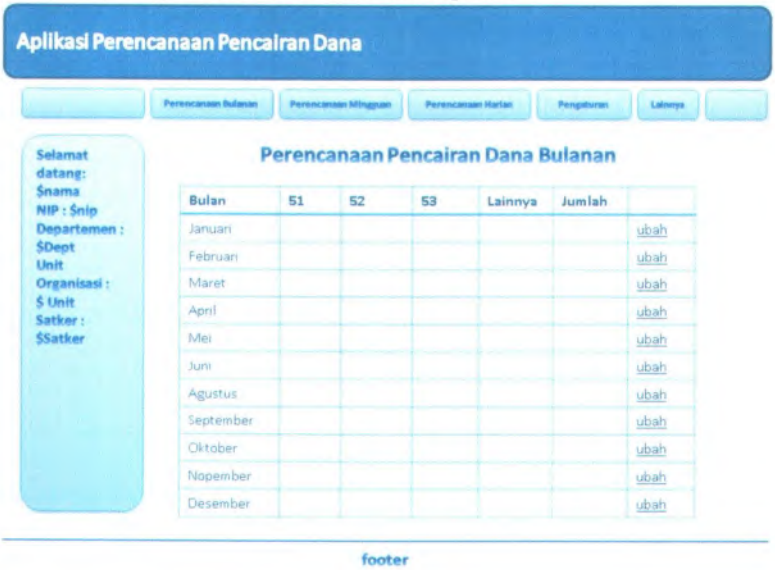
2. Pengguna menekan link ubah pada bulan yang dipilih



3. *Pengguna mengubah nilai perencanaan dan menekan tombol submit*

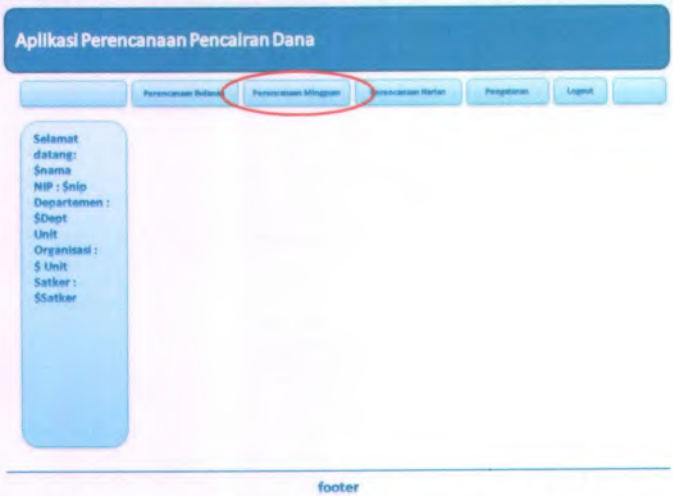


4. Halaman perencanaan bulanan ditampilkan

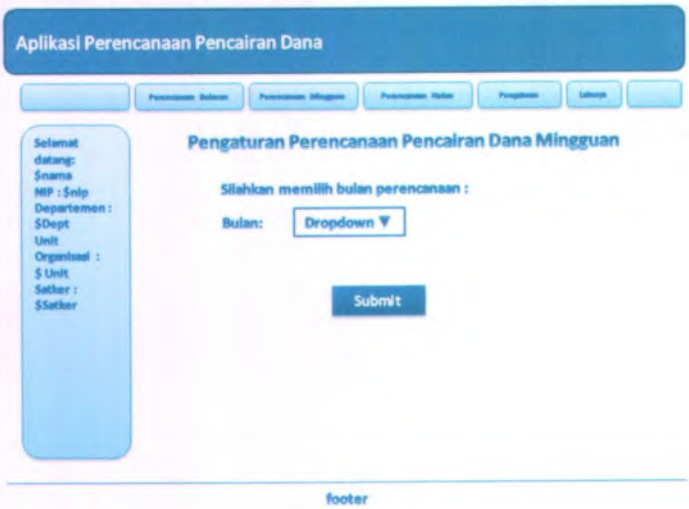


C. Storyboard operator satker : Menampilkan perencanaan mingguan

1. Pengguna menekan link perencanaan mingguan



2. Pengguna memilih bulan perencanaan kemudian menekan tombol submit



3. Halaman perencanaan mingguan ditampilkan

naan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Perencanaan Pencairan Dana Mingguan

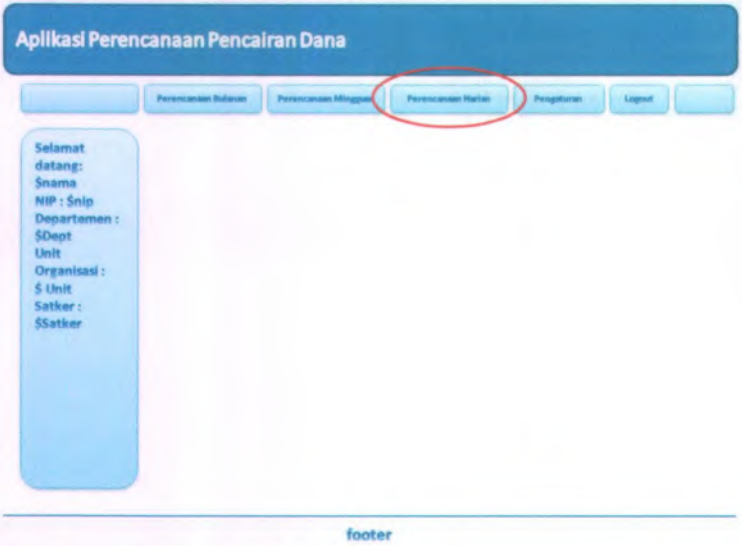
Minggu Ke	51		52		53		Lainnya		Jumlah	
	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan
1										
2										
3										
4										

Nilai Perencanaan Bulanan	
Total Perencanaan Mingguan	

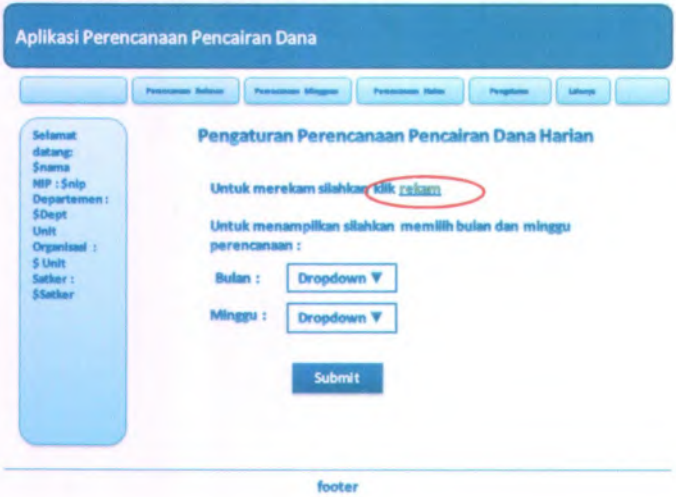
footer

D. Storyboard operator satker : Merekam perencanaan harian

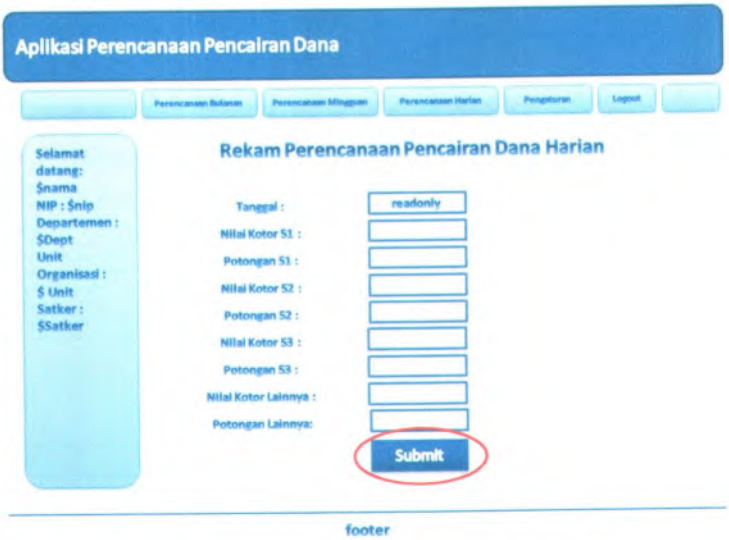
1. Pengguna menekan link perencanaan harian



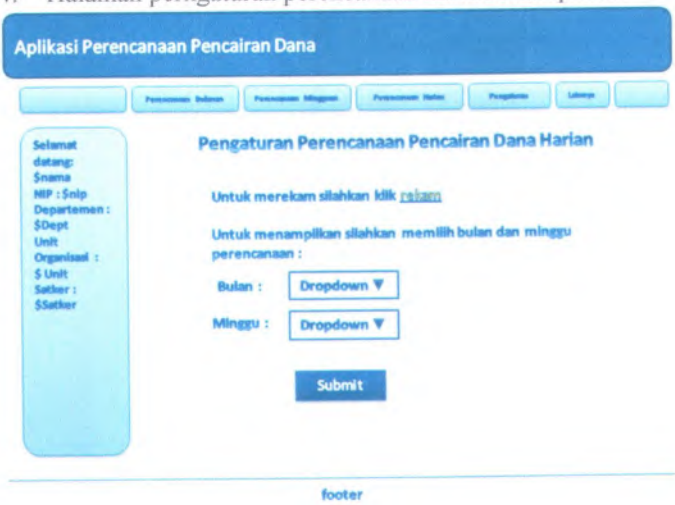
2. Pengguna menekan link rekam



3. Pengguna mengisi data perencanaan dan menekan tombol submit

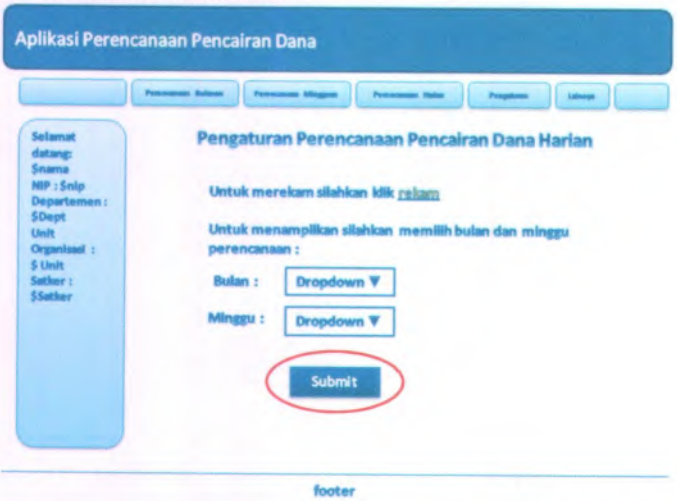


4. Halaman perngaturan perencanaan harian ditampilkan

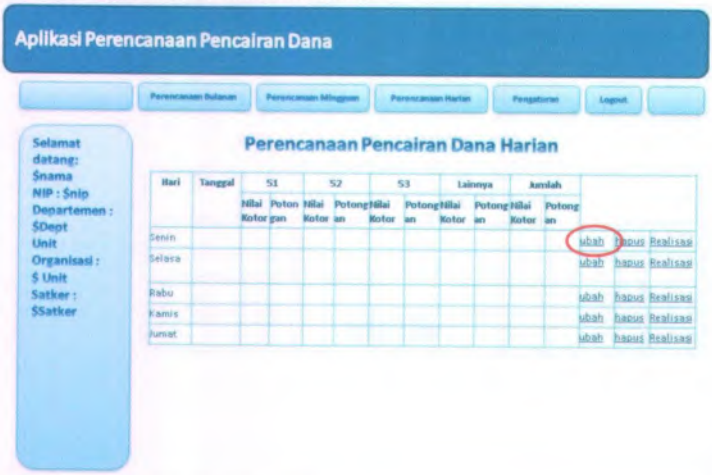


E. Storyboard operator satker : Mengubah perencanaan harian

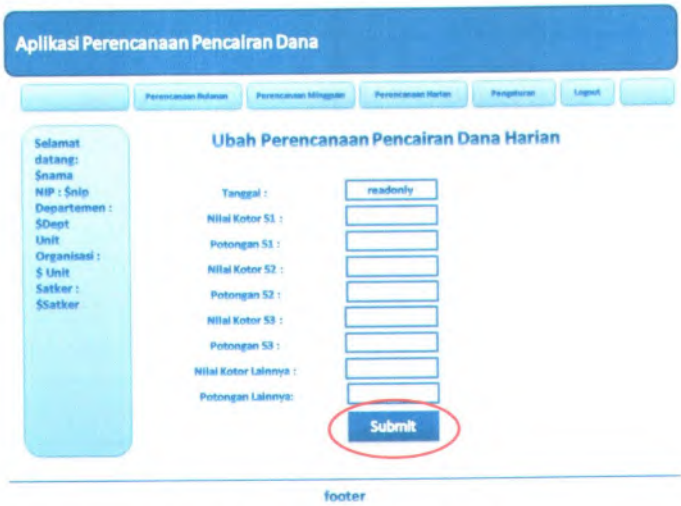
1. Pengguna memilih bulan dan minggu perencanaan dan menekan tombol submit



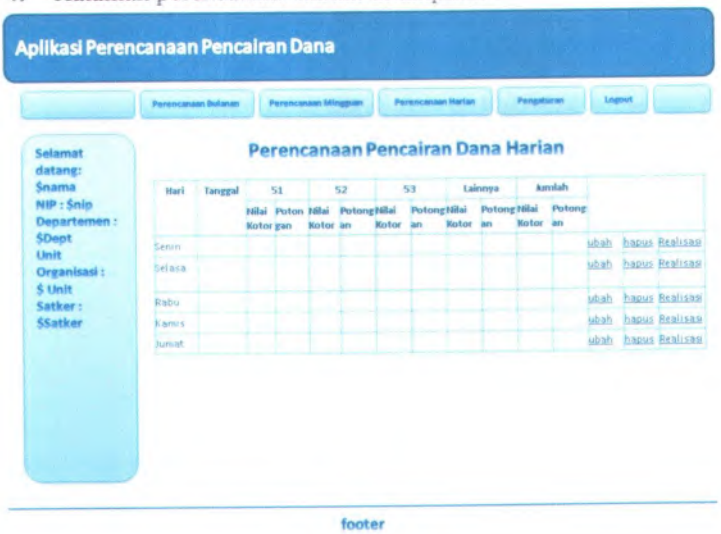
2. Pengguna menekan link ubah



3. Pengguna mengisi data perencanaan dan menekan tombol submit

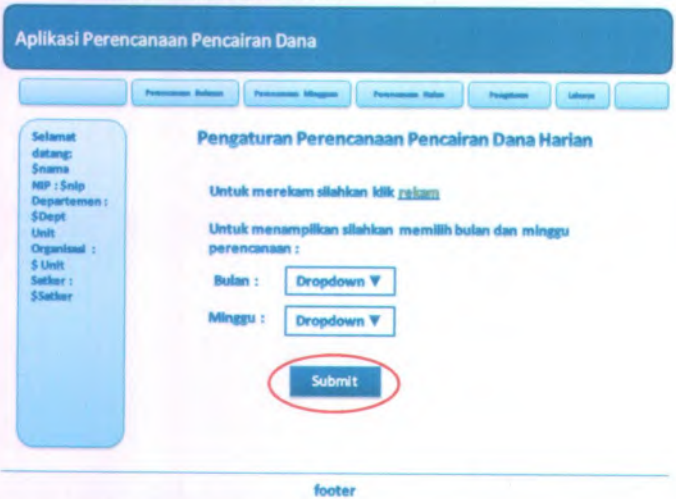


4. Halaman perencanaan harian ditampilkan



F. Storyboard operator satker : Menghapus perencanaan harian

1. Pengguna memilih bulan dan minggu perencanaan dan menekan tombol submit



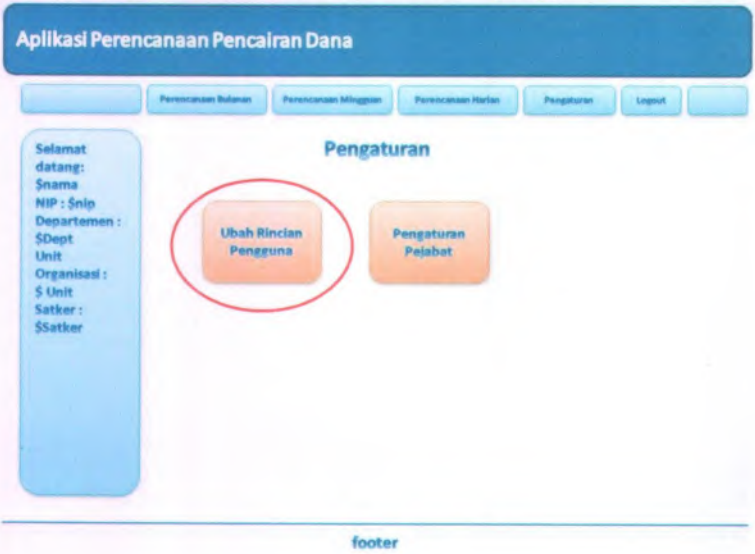
2. Pengguna menekan link hapus

3. Halaman perencanaan harian ditampilkan

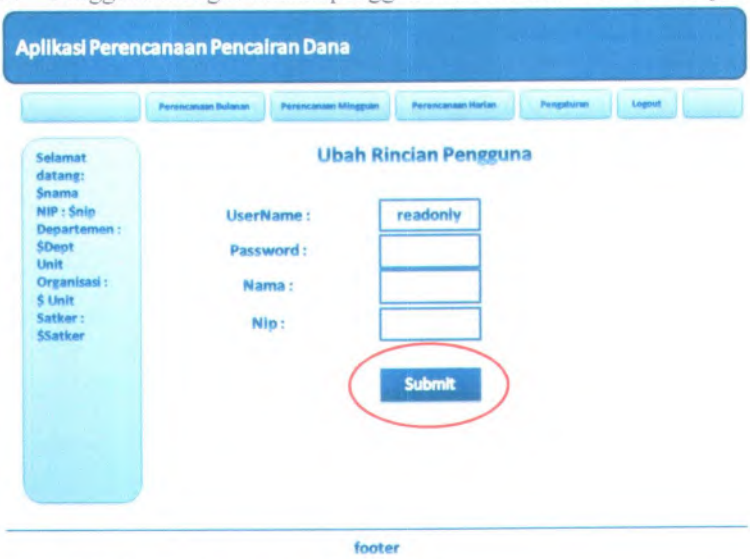
G. Storyboard umum : Mengubah rincian pengguna

1. Pengguna menekan link pengaturan

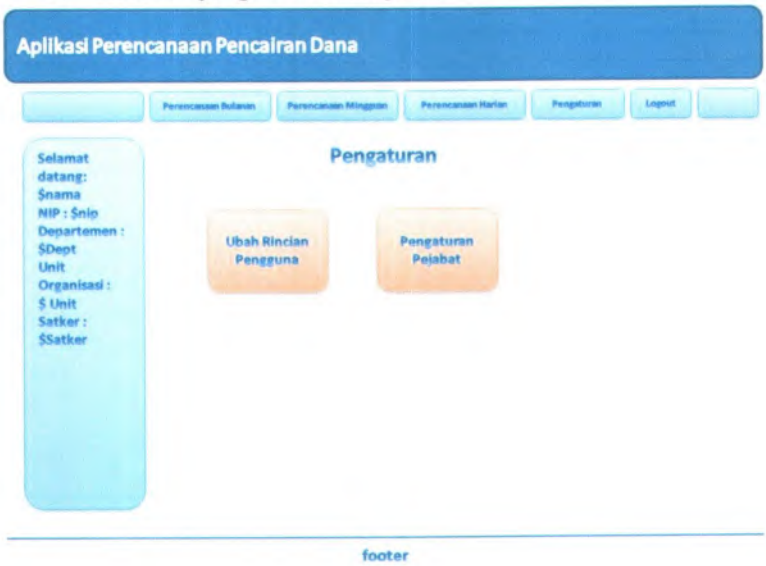
2. Pengguna menekan tombol ubah rincian pengguna



3. Pengguna mengubah data pengguna dan menekan tombol simpan

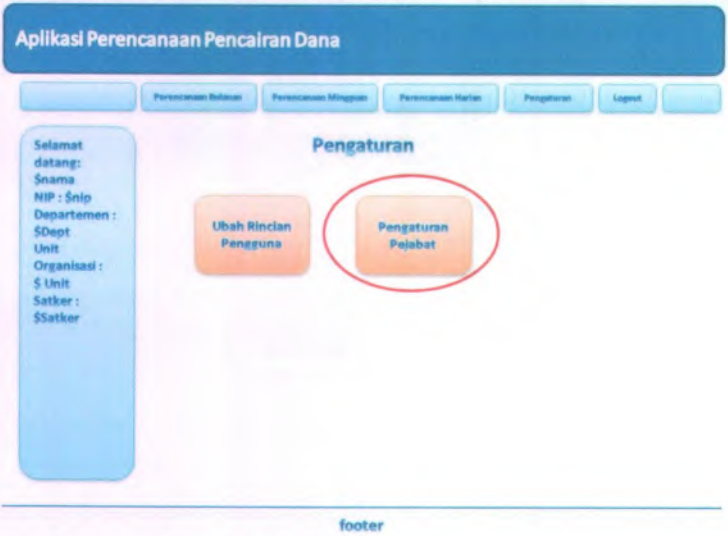


4. Halaman pengaturan ditampilkan

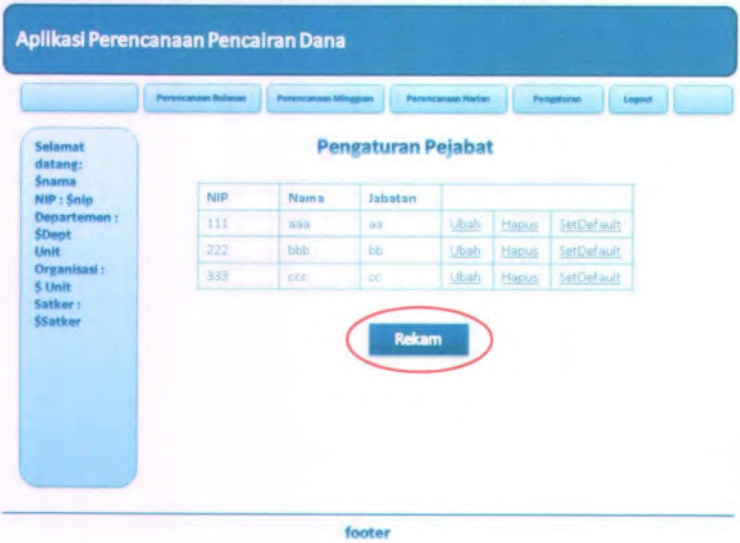


H. Storyboard umum : Merekam data pejabat

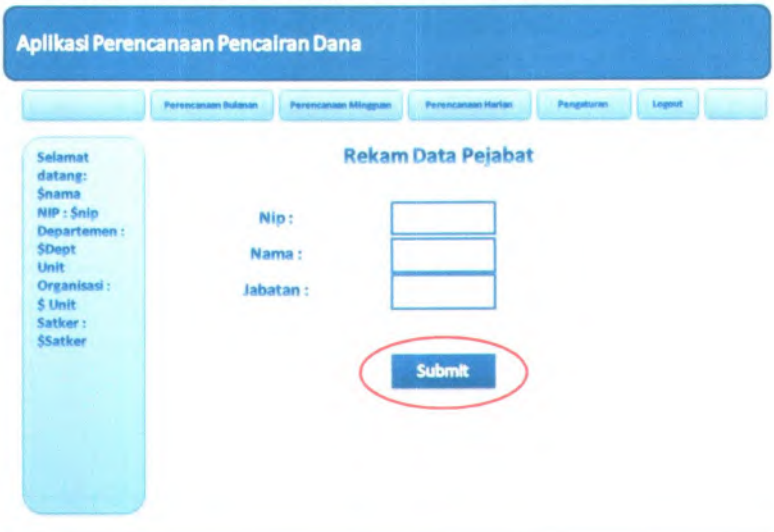
1. Pengguna menekan tombol pengaturan pejabat



2. Pengguna menekan tombol rekam



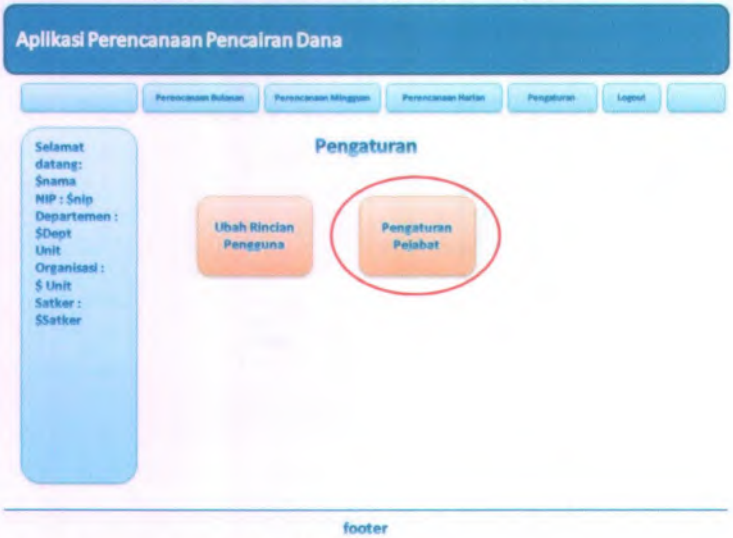
3. Pengguna mengisi data pejabat dan menekan tombol submit



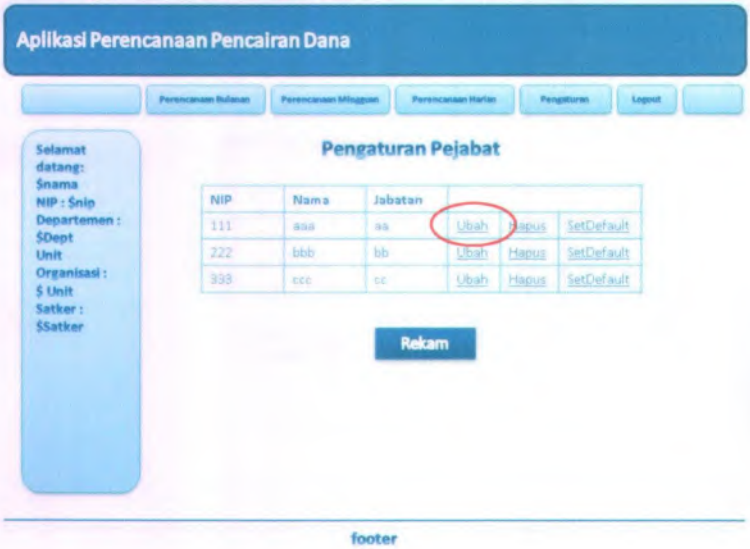
4. Halaman pengaturan pejabat ditampilkan

I. Storyboard umum : Mengubah data pejabat

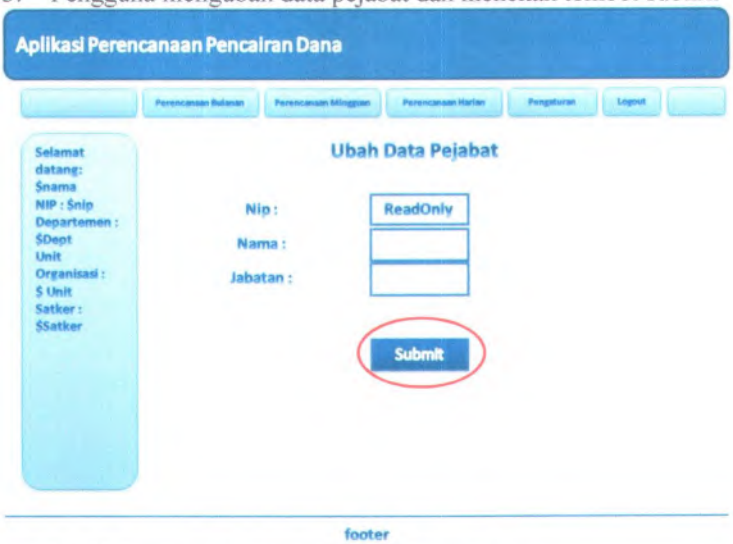
1. Pengguna menekan tombol pengaturan pejabat



2. Pengguna menekan link ubah



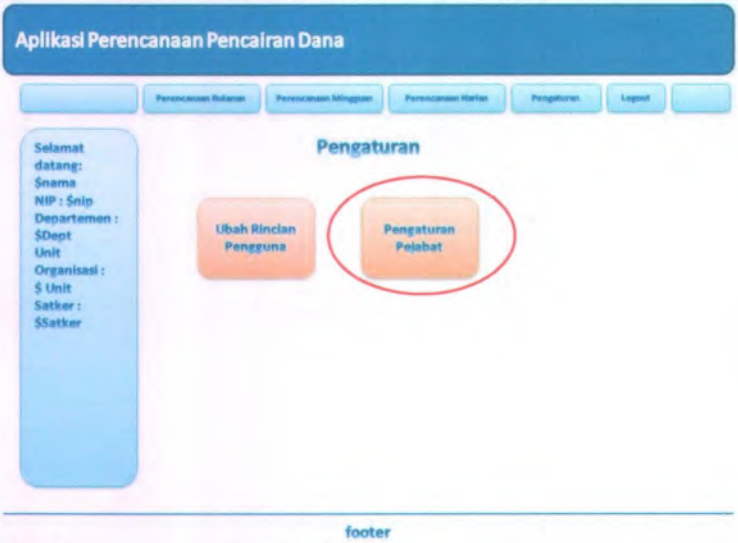
3. Pengguna mengubah data pejabat dan menekan tombol submit



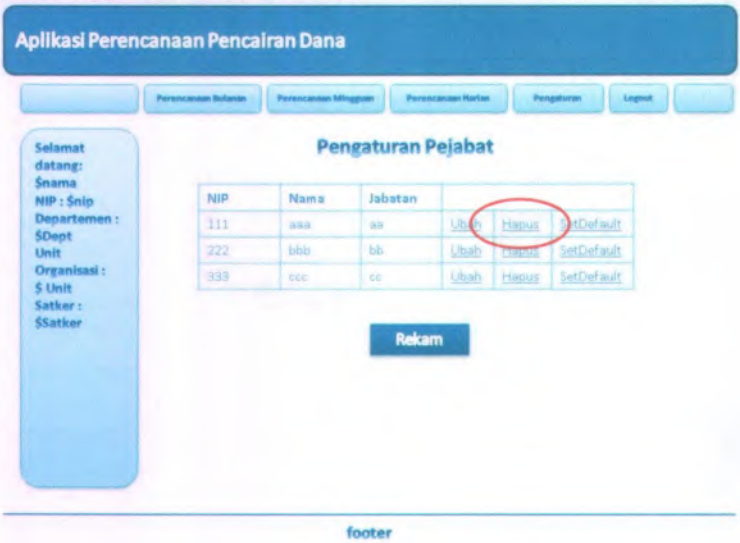
4. Halaman pengaturan pejabat ditampilkan

J. Storyboard umum : menghapus data pejabat

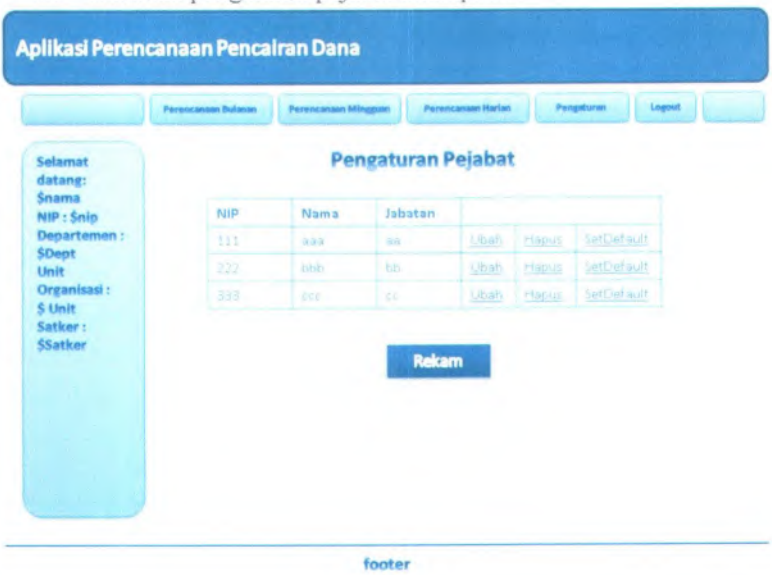
1. Pengguna menekan tombol pengaturan pejabat



2. Pengguna menekan link hapus



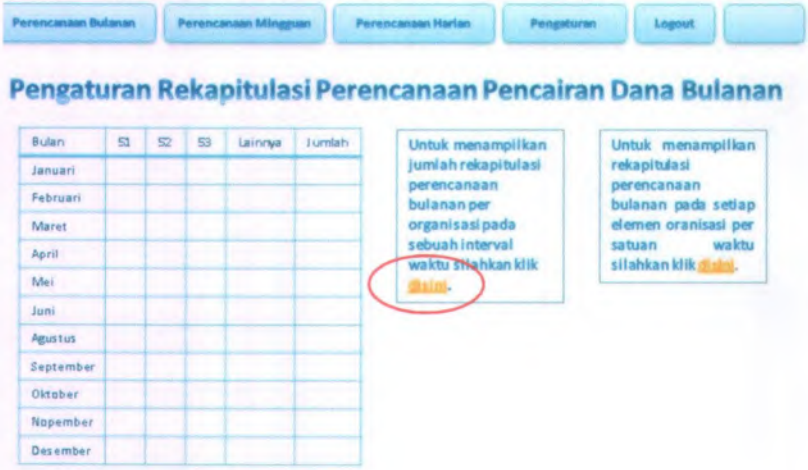
3. Halaman pengaturan pejabat ditampilkan



K. Storyboard operator kppn : menampilkan rekapitulasi perencanaan bulanan per departemen

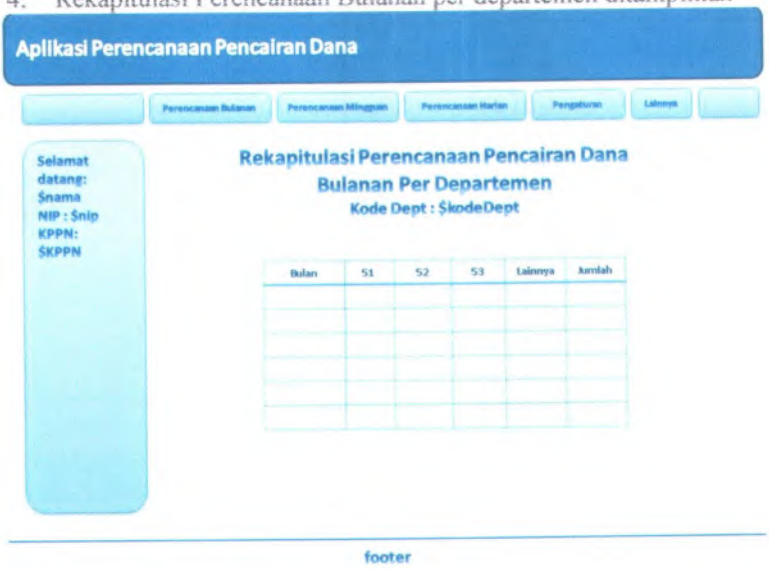
1. Pengguna menekan link perencanaan bulanan

2. Pengguna menekan link "disini" di kolom sebelah kiri



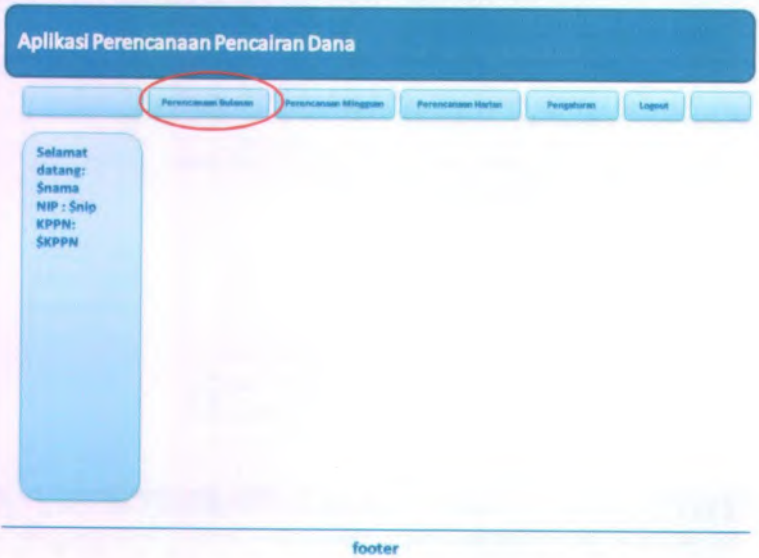
3. Pengguna mengisi kode departemen dan menekan tombol submit

4. Rekapitulasi Perencanaan Bulanan per departemen ditampilkan

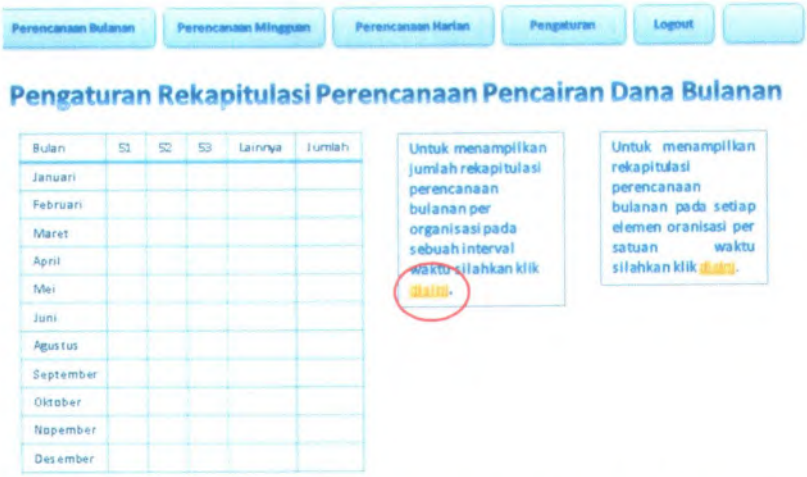


L. Storyboard operator kppn : menampilkan rekapitulasi perencanaan bulanan per departemen dan unit organisasi

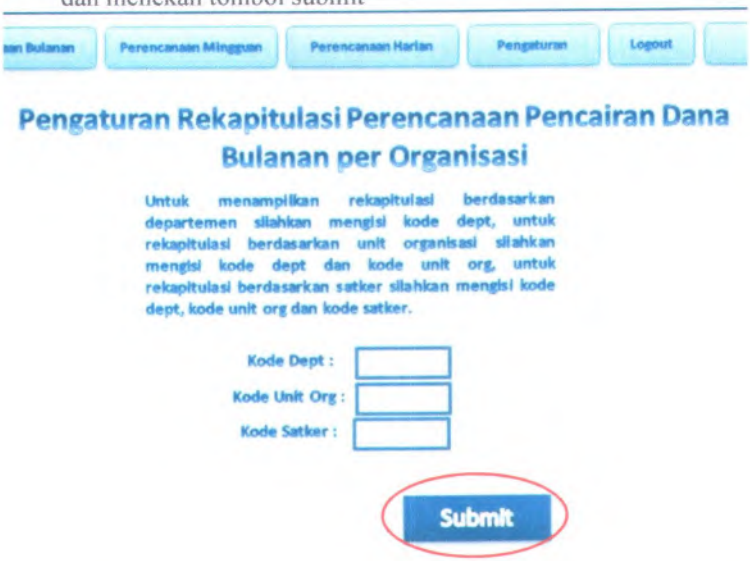
1. Pengguna menekan link perencanaan bulanan



2. Pengguna menekan link "disini" di kolom sebelah kiri



3. Pengguna memasukkan kode departemen dan kode unit organisasi dan menekan tombol submit



4. Rekapitulasi Perencanaan Bulanan per unit organisasi ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

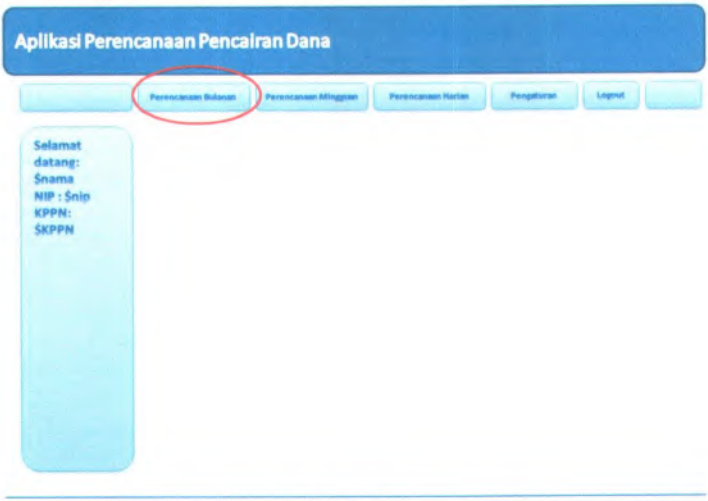
Rekapitulasi Perencanaan Pencairan Dana Bulanan Per Unit Organisasi

Kode Dept : $kodeDept
Kode Unit Org: $kodeUnitOrg

Bulan	51	52	53	Lainnya	Jumlah

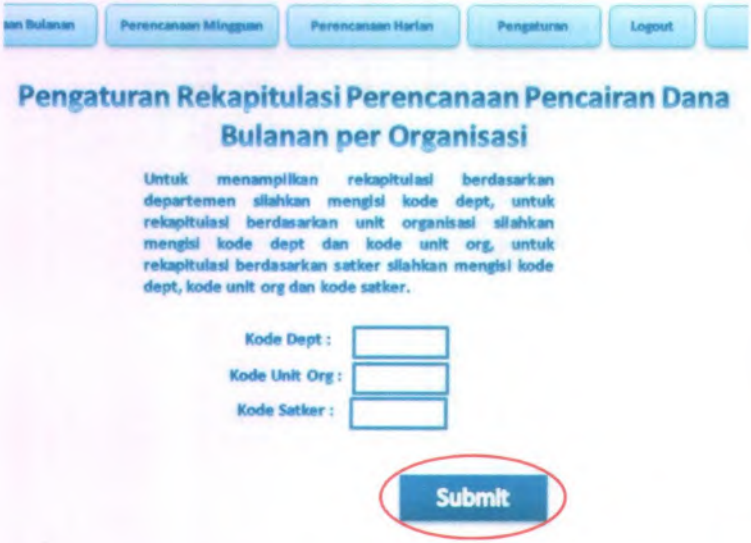
M. Storyboard operator kppn : menampilkan rekapitulasi perencanaan bulanan per departemen, unit organisasi dan satker

1. Pengguna menekan link perencanaan bulanan

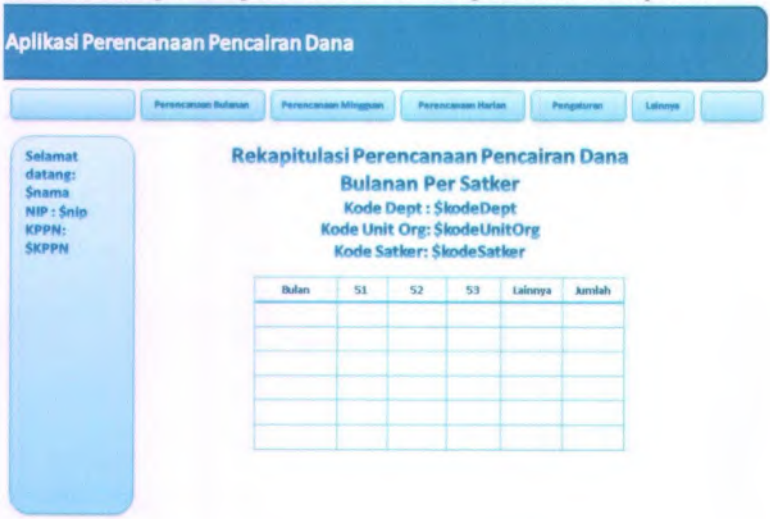


2. Pengguna menekan link "disini" di kolom sebelah kiri

3. *Pengguna memasukkan kode departemen , unit organisasi dan satker dan menekan tombol submit*

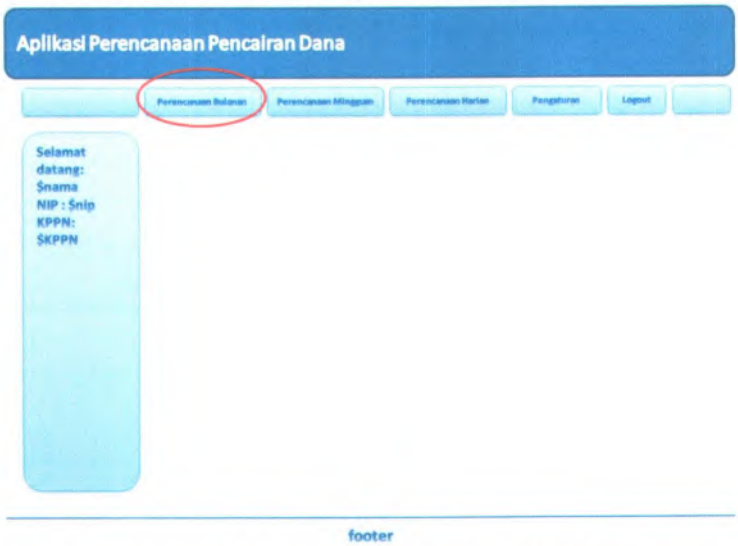


4. Rekapitulasi perencanaan bulanan per satker ditampilkan



N. Storyboard operator kppn : menampilkan rekapitulasi perencanaan bulanan per departemen per bulan

1. Pengguna menekan link perencanaan bulanan



2. Pengguna menekan link "disini" di kolom sebelah kanan

3. *Pengguna memasukkan kode departemen , memilih bulan* perencanaan dan menekan tombol submit

Pengaturan Rekapitulasi Perencanaan Pencairan Dana Bulanan per Satuan Waktu

Untuk menampilkan rekapitulasi berdasarkan departemen silahkan mengisi kode dept, untuk rekapitulasi berdasarkan unit organisasi silahkan mengisi kode dept dan kode unit org, untuk rekapitulasi berdasarkan satker silahkan mengisi kode dept, kode unit org dan kode satker.

Kode Dept : Bulan : Dropdown ▼

Kode Unit Org :

Kode Satker :

Submit

4. Rekapitulasi perencanaan bulanan per departemen per bulan ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

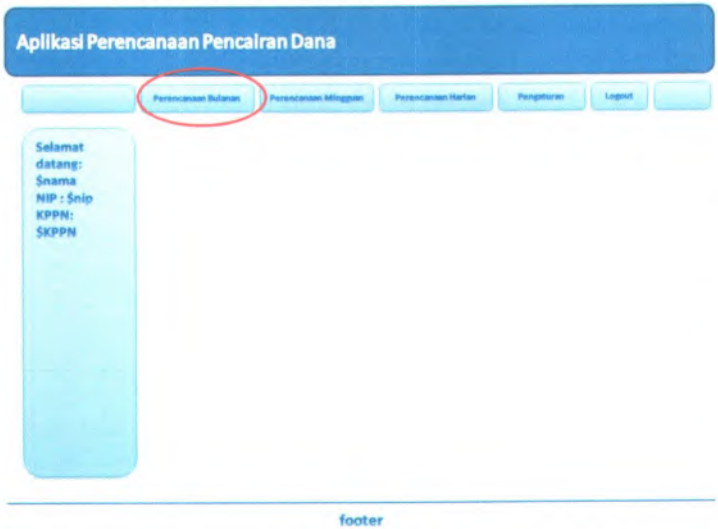
Rekapitulasi Perencanaan Pencairan Dana Bulanan per Departemen Per Bulan

Kode Dept : $kodeDept
Bulan : $bulan

Kode Unit Organisasi	Kode Satker	S1	S2	S3	Lainnya	Jumlah

O. Storyboard operator kppn : menampilkan rekapitulasi perencanaan bulanan per unit organisasi per bulan

1. Pengguna menekan link perencanaan bulanan



2. Pengguna menekan link "disini" di kolom sebelah kanan

3. *Pengguna memasukkan kode departemen dan kode unit organisasi,* memilih bulan perencanaan dan menekan tombol submit

Pengaturan Rekapitulasi Perencanaan Pencairan Dana Bulanan per Satuan Waktu

Untuk menampilkan rekapitulasi berdasarkan departemen silahkan mengisi kode dept, untuk rekapitulasi berdasarkan unit organisasi silahkan mengisi kode dept dan kode unit org, untuk rekapitulasi berdasarkan satker silahkan mengisi kode dept, kode unit org dan kode satker.

Kode Dept : Bulan : Dropdown ▼

Kode Unit Org :

Kode Satker :

Submit

4. Rekapitulasi perencanaan bulanan per departemen per bulan ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

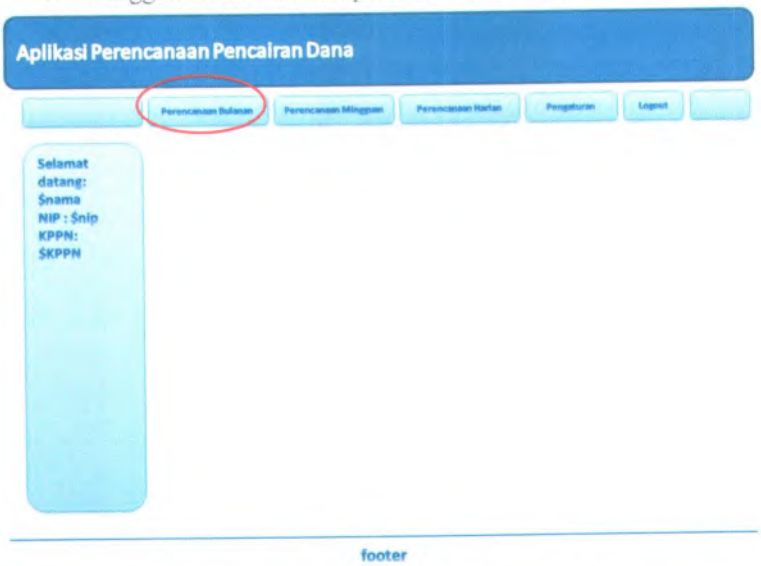
Rekapitulasi Perencanaan Pencairan Dana Bulanan per Unit Org per Bulan

Kode Dept : $kodeDept
Kode Unit Org : $kodeUnitOrg
Bulan : $bulan

Kode Satker	51	52	53	Lainnya	Jumlah

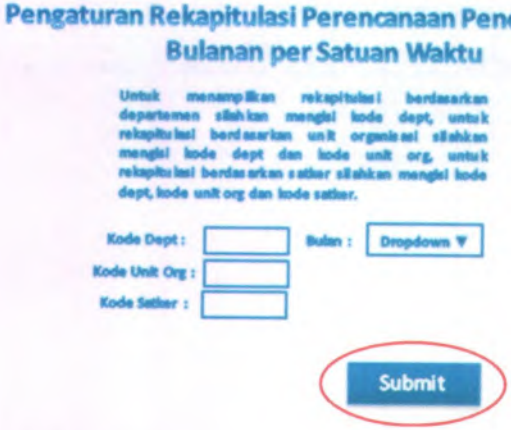
P. Storyboard operator kppn : menampilkan rekapitulasi perencanaan bulanan per satker per bulan

1. Pengguna menekan link perencanaan bulanan

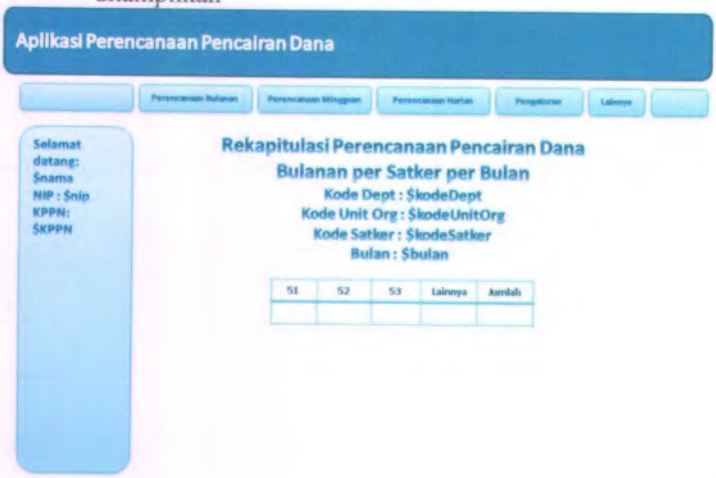


2. Pengguna menekan link "disini" di kolom sebelah kanan

3. *Pengguna memasukkan kode departemen, kode unit organisasi dan* kode satker, memilih bulan perencanaan dan menekan tombol submit

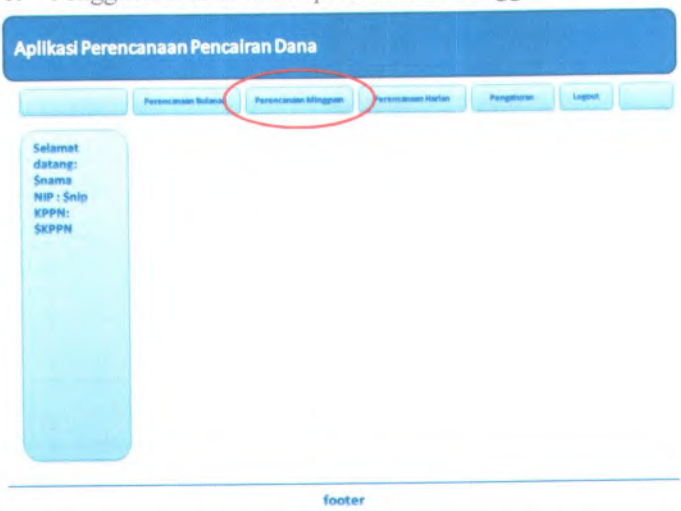


4. Rekapitulasi perencanaan bulanan per departemen per bulan ditampilkan



Q. Storyboard operator kppn : menampilkan rekapitulasi perencanaan mingguan per departemen

1. Pengguna menekan link perencanaan mingguan



2. Pengguna memasukkan kode departemen serta memilih bulan dan minggu perencanaan kemudian menekan tombol submit

3. Halaman rekapitulasi perencanaan mingguan ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

Rekapitulasi Perencanaan Pencairan Dana Mingguan

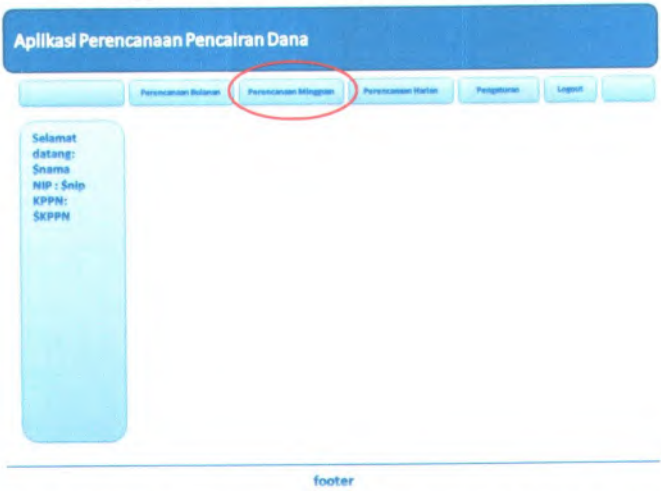
Kode Dept : $kodeDept
Bulan: $bulan
Minggu ke : $mingguKe

Kode Unit Organisasi	Kode Satker	51		52		53		Lainnya		Jumlah	
		Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan

footer

R. Storyboard operator kppn : menampilkan rekapitulasi perencanaan mingguan per departemen dan unit organisasi

1. Pengguna menekan link perencanaan mingguan



2. Pengguna memasukkan kode departemen dan unit organisasi serta memilih bulan dan minggu perencanaan kemudian menekan tombol submit

3. Halaman rekapitulasi perencanaan mingguan ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

Rekapitulasi Perencanaan Pencairan Dana Mingguan

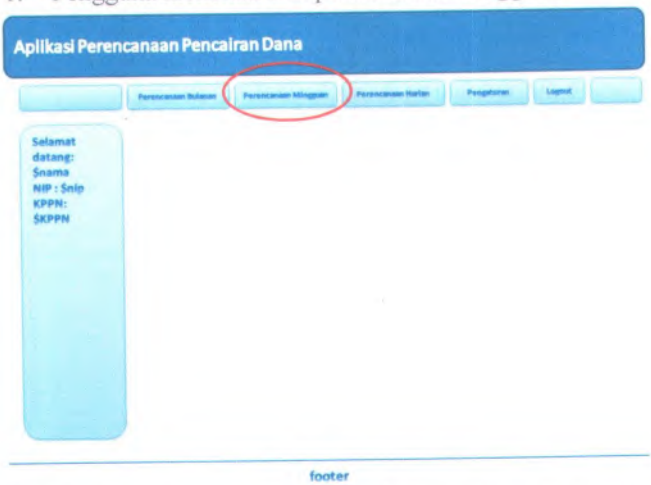
Kode Dept : $kodeDept
Kode Unit Organisasi: $kodeUnitOrg
Bulan: $bulan
Minggu ke : $mingguKe

Kode Satker	51		52		53		Lainnya		Jumlah	
	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan

footer

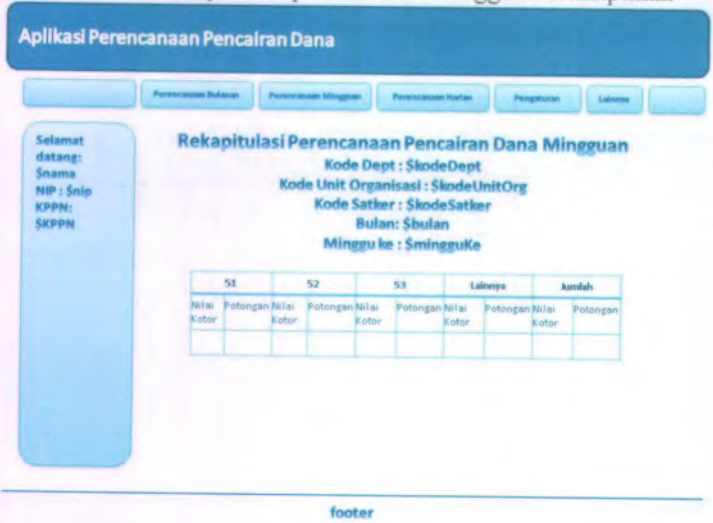
S. Storyboard operator kppn : menampilkan rekapitulasi perencanaan mingguan per departemen, unit organisasi dan satker

1. Pengguna menekan link perencanaan mingguan



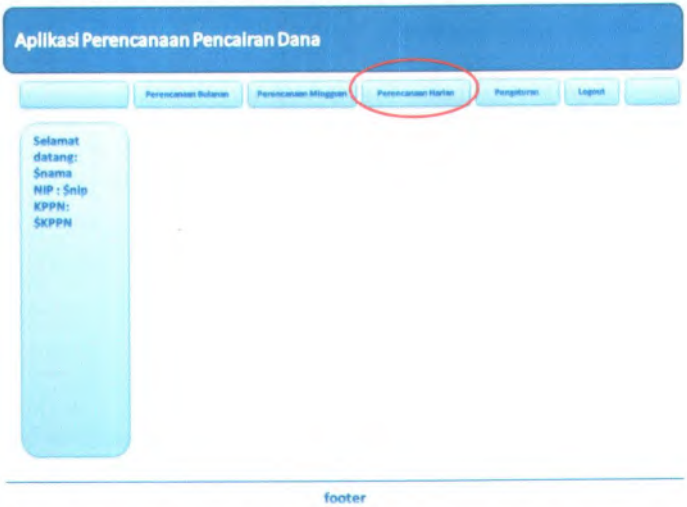
2. Pengguna memasukkan kode departemen, unit organisasi dan satker serta memilih bulan dan minggu perencanaan kemudian menekan tombol submit

3. Halaman rekapitulasi perencanaan mingguan ditampilkan

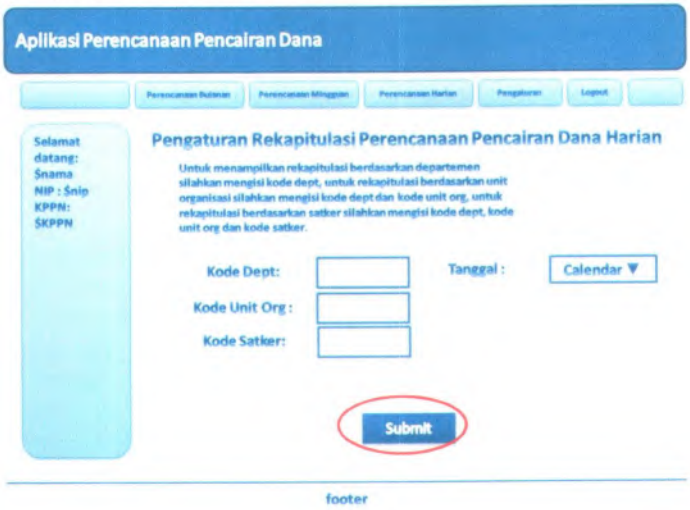


T. Storyboard operator kppn : menampilkan rekapitulasi perencanaan harian per departemen

1. Pengguna menekan link perencanaan harian



2. Pengguna memasukkan kode departemen serta memilih tanggal perencanaan kemudian menekan tombol submit



3. Halaman rekapitulasi perencanaan harian ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

Rekapitulasi Perencanaan Pencairan Dana Harian

Kode Dept : $kodeDept
Tanggal : $tanggal
Hari : $hari

Kode Unit Organisasi	Kode Satker	51		52		53		Lainnya		Jumlah	
		Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan

footer

U. Storyboard operator kppn : menampilkan rekapitulasi perencanaan harian per departemen dan unit organisasi

1. Pengguna menekan link perencanaan harian

2. Pengguna memasukkan kode departemen dan unit organisasi serta memilih tanggal perencanaan kemudian menekan tombol submit

3. Halaman rekapitulasi perencanaan harian ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

Rekapitulasi Perencanaan Pencairan Dana Harian

Kode Dept : $kodeDept
Kode Unit Org : $kodeUnitOrg
Tanggal : $tanggal
Hari : $hari

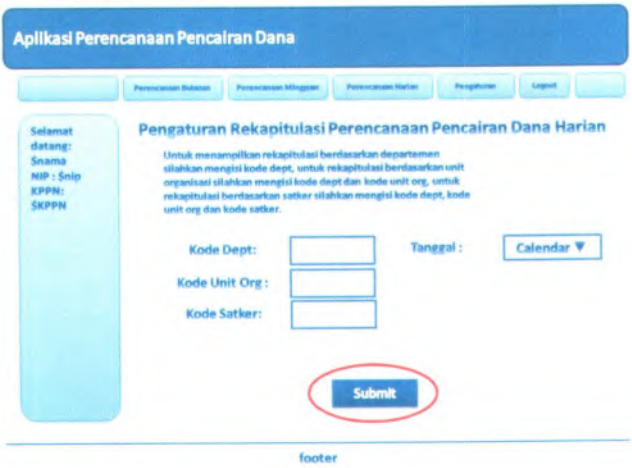
Kode Satker	51		52		53		Lainnya		Jumlah	
	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan

footer

V. Storyboard operator kppn : menampilkan rekapitulasi perencanaan harian per departemen, unit organisasi dan satker

1. Pengguna menekan link perencanaan harian

2. Pengguna memasukkan kode departemen, unit organisasi dan satker serta memilih tanggal perencanaan kemudian menekan tombol submit



3. Halaman rekapitulasi perencanaan harian ditampilkan

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

Rekapitulasi Perencanaan Pencairan Dana Harian

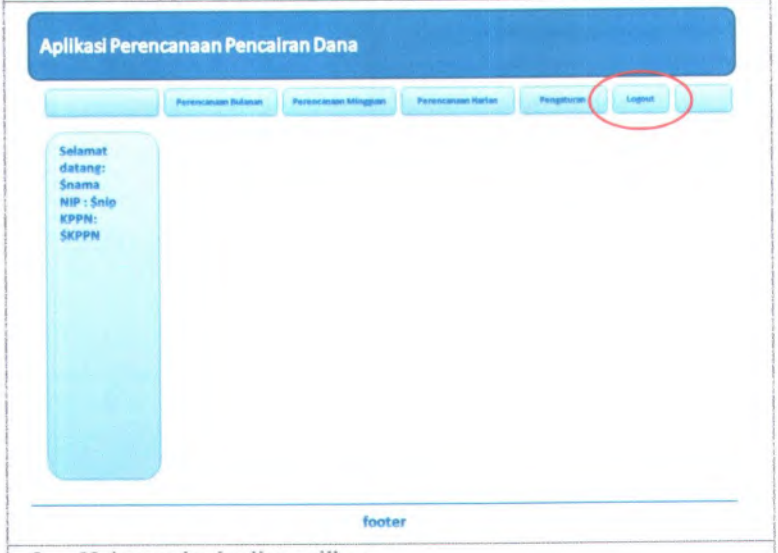
Kode Dept : $kodeDept
Kode Unit Org : $kodeUnitOrg
Kode Satker : $kodeSatker
Tanggal : $tanggal
Hari : $hari

51		52		53		Lainnya		Jumlah	
Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan	Nilai Kotor	Potongan

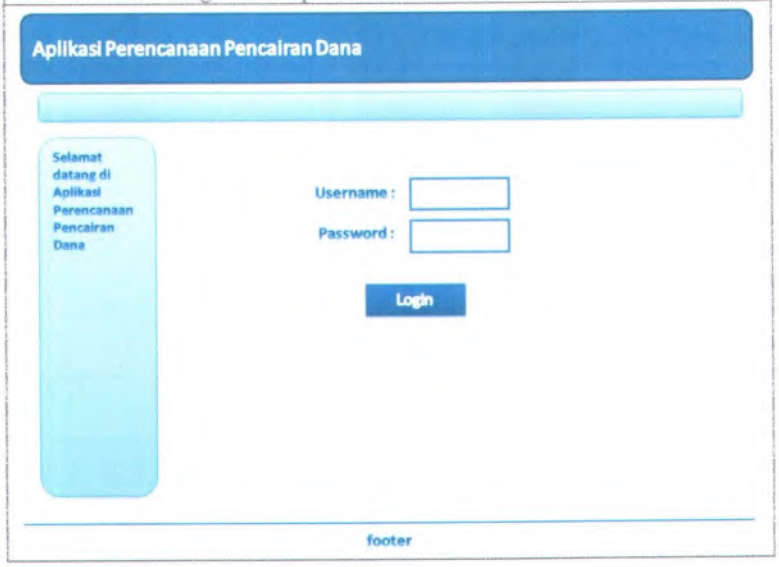
footer

W. Storyboard : logout

1. Pengguna menekan link logout

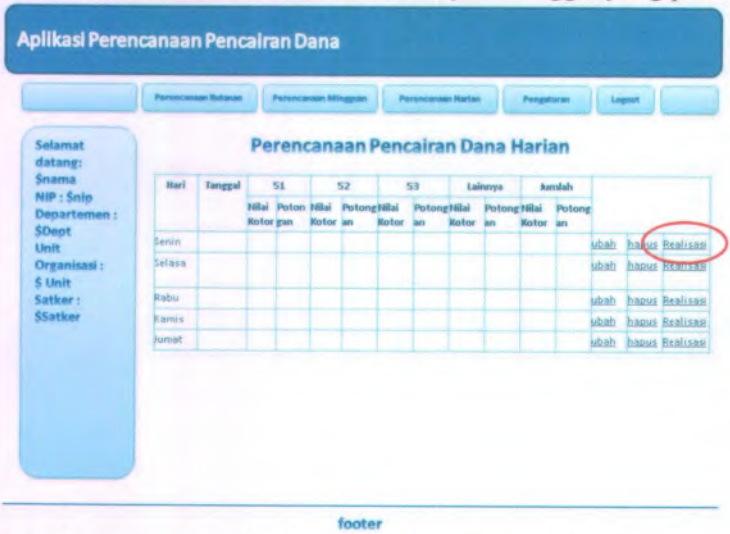


2. Halaman login ditampilkan

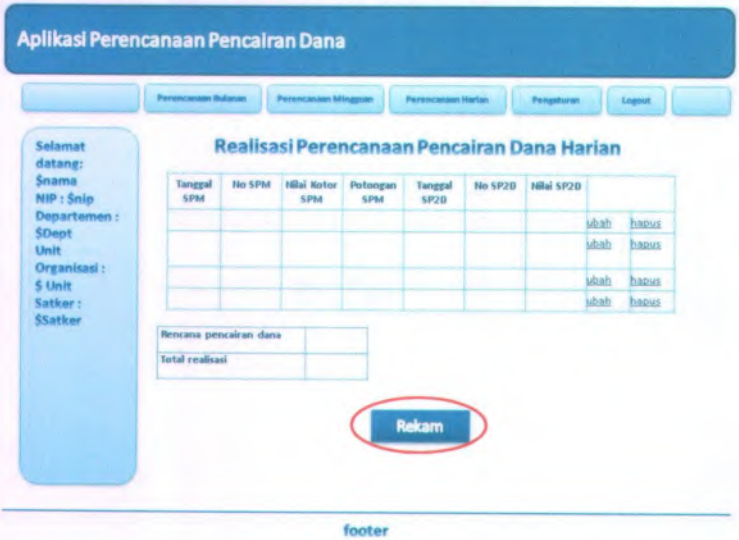


X. Storyboard Operator Satker : rekam realisasi

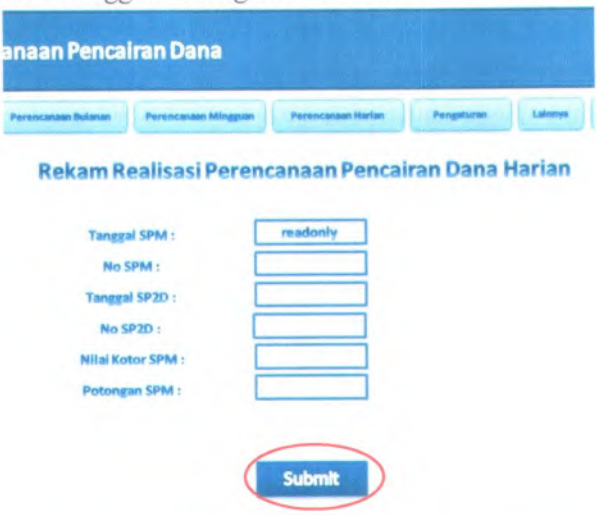
1. Pengguna menekan link realisasi pada tanggal yang pilih



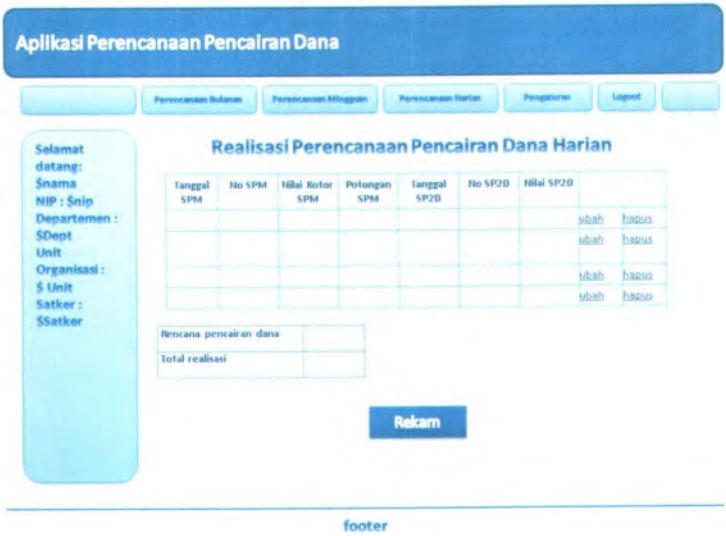
2. Pengguna menekan tombol rekam pada halaman realisasi



3. Pengguna mengisi data realisasi dan menekan tombol submit

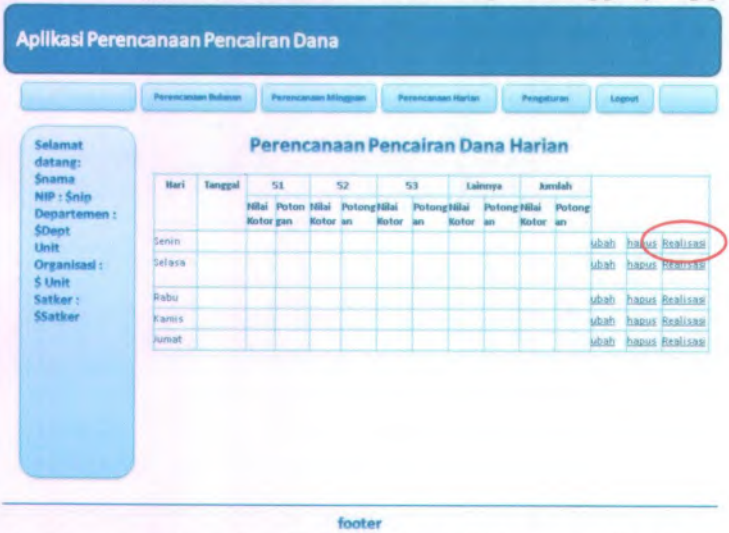


4. Halaman realisasi ditampilkan

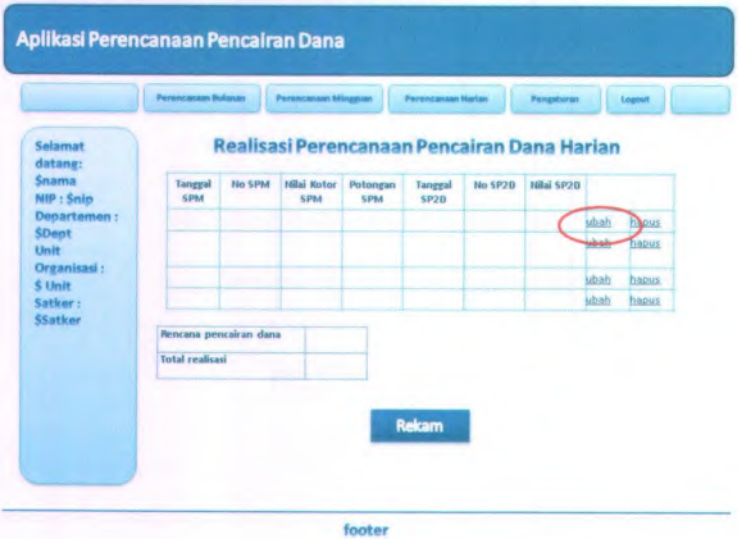


Y. Storyboard Operator Satker : Ubah realisasi

1. Pengguna menekan link realisasi pada tanggal yang pilih



2. Pengguna menekan link ubah pada halaman realisasi

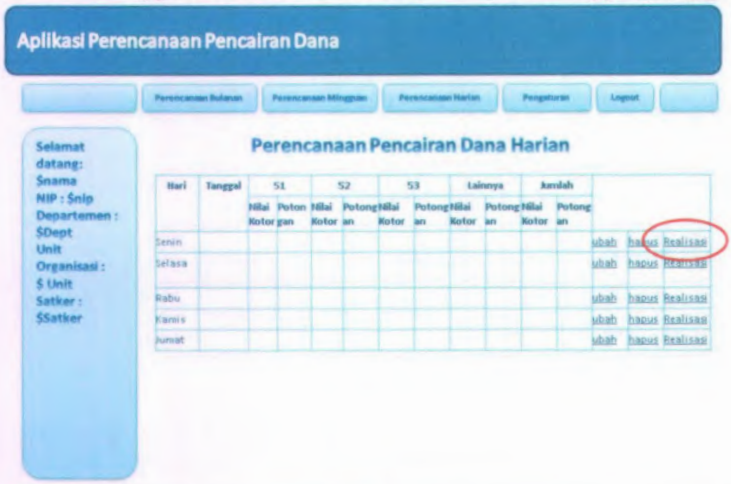


3. Pengguna mengubah data realisasi dan menekan tombol submit

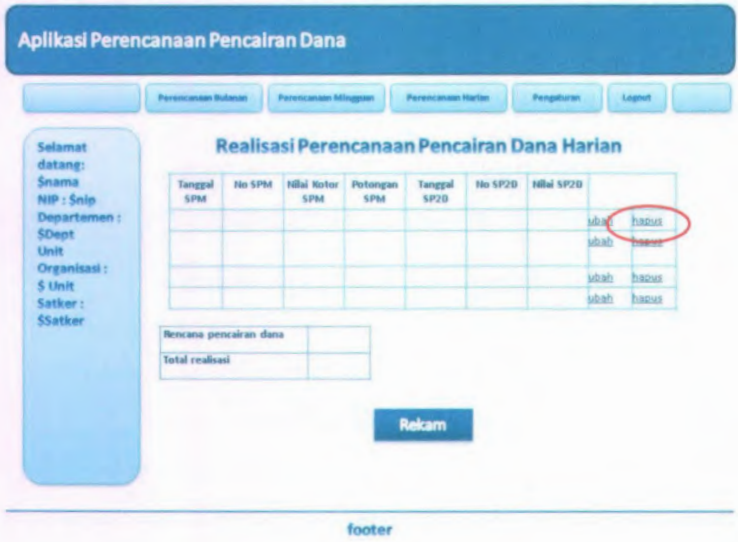
4. Halaman realisasi ditampilkan

Z. Storyboard Operator Satker : Hapus realisasi

1. Pengguna menekan link realisasi pada tanggal yang pilih



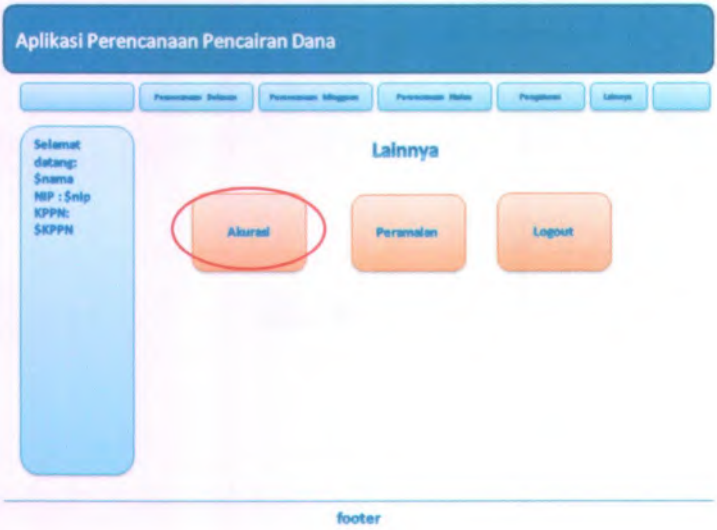
2. Pengguna menekan link hapus pada halaman realisasi



3. Halaman realisasi ditampilkan

AA. Storyboard Operator KPPN : menampilkan akurasi bulanan

1. Pada halaman lainnya, pengguna menekan tombol akurasi



2. Pada halaman pengaturan akurasi, pengguna menekan link "disini" pada pilihan akurasi perencanaan pencairan dana bulanan.

Pengaturan Akurasi

AKURASI PERENCANAAN PENCAIRAN DANA BULANAN

Untuk menampilkan akurasi rekapitulasi perencanaan pencairan dana bulanan, silahkan klik disini

AKURASI PERENCANAAN PENCAIRAN DANA MINGGUAN

Untuk menampilkan akurasi rekapitulasi perencanaan pencairan dana mingguan, silahkan pilih bulan kemudian tekan "proses"

Januari ▼ Proses

AKURASI PERENCANAAN PENCAIRAN DANA HARIAN

Untuk menampilkan akurasi rekapitulasi perencanaan pencairan dana harian, silahkan pilih bulan dan minggu kerja kemudian tekan "proses"

Januari ▼

1 ▼ Proses

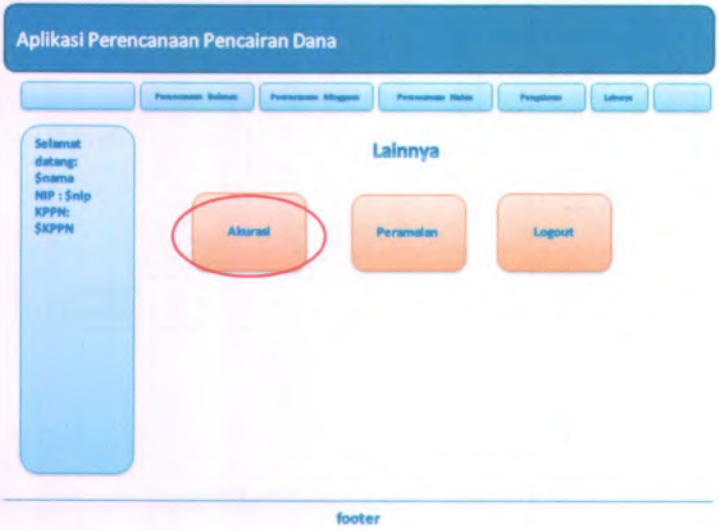
3. Halaman akurasi perencanaan pencairan dana bulanan ditampilan

Akurasi Perencanaan Pencairan Dana Bulanan

Dept	Unit org	Setler	Jan	Feb	Mar	April	Mei	Juni	Juli	Agus	Sept	Okt	Nop	Des

BB. Storyboard Operator KPPN : menampilkan akurasi mingguan

1. Pada halaman lainnya, pengguna menekan tombol akurasi



2. Pada halaman pengaturan akurasi, pengguna memilih bulan dan menekan tombol proses pada pilihan akurasi perencanaan pencairan dana mingguan.

Pengaturan Akurasi

AKURASI PERENCANAAN PENCAIRAN DANA BULANAN

Untuk menampilkan akurasi rekapitulasi perencanaan pencairan dana bulanan, silahkan klik disini

AKURASI PERENCANAAN PENCAIRAN DANA MINGGUAN

Untuk menampilkan akurasi rekapitulasi perencanaan pencairan dana mingguan, silahkan pilih bulan kemudian tekan "proses"

Januari ▼ Proses

AKURASI PERENCANAAN PENCAIRAN DANA HARIAN

Untuk menampilkan akurasi rekapitulasi perencanaan pencairan dana harian, silahkan pilih bulan dan minggu kerja kemudian tekan "proses"

Januari ▼

1 ▼ Proses

Halaman akurasi perencanaan pencairan dana mingguan ditampilkan.

Aplikasi Perencanaan Pencairan Dana

Perencanaan Bulanan | Perencanaan Mingguan | Perencanaan Harian | Pengaturan | Lainnya

Selamat datang:
$nama
NIP : $nip
KPPN:
$KPPN

Akurasi Perencanaan Pencairan Dana Mingguan

Dept	Unit org	Setler	Minggu ke-1	Minggu ke-2	Minggu ke-3	Minggu ke-4

footer

CC. Storyboard Operator KPPN : menampilkan akurasi harian

1. Pada halaman lainnya, pengguna menekan tombol akurasi

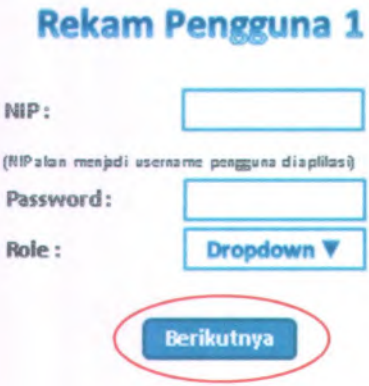
2. Pada halaman pengaturan akurasi, pengguna memilih bulan dan minggu kemudian menekan tombol proses pada pilihan akurasi perencanaan pencairan dana harian.

Halaman akurasi perencanaan pencairan dana harian ditampilkan.

DD. Storyboard Admin: rekam pengguna

1 Pada halaman pengaturan pengguna admin menekan tombol "rekam"

2 Pada halaman rekam pengguna 1, admin memasukan NIP dan password serta memilih role pengguna yang direkam kemudian menekan tombol "berikutnya".



3 Pada halaman rekam pengguna 2, jika role yang dipilih "operator kppn" admin harus menginputkan kode kppn, sementara jika role yang dipilih "operator satker" admin harus menginputkan kode departemen, unit organisasi dan satker, sementara jika role yang dipilih "admin" admin tidak menginputkan apapun, field yang tidak diperlukan akan ter-*disable*, kemudian admin menekan tombol "simpan".

4 Halaman pengaturan pengguna kembali ditampilan.

EE. Storyboard Admin: hapus pengguna

1 Pada halaman pengaturan pengguna admin menekan link "hapus"

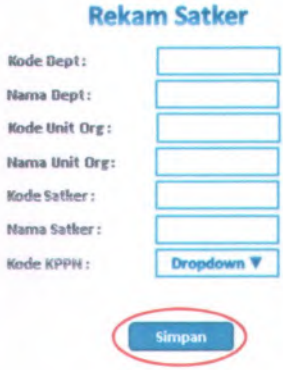
2 Halaman pengaturan pengguna kembali ditampilkan


Aplikasi Perencanaan Pencairan Dana
Selamat datang Admin : $nama NIP : $nip
Pengaturan Pengguna
NIP
Role
Kode Kppn
Kode Dept
Kode Unit Org
Kode Satker
123456 78910
Operator Satker
025
04
554155
hapus
123456 78911
Operator KPPN
145
hapus
123456 78912
Admin
hapus
rekam

FF. Storyboard Admin: rekam satker

1 Pada halaman pengaturan satker admin menekan tombol "rekam".

2 Pada halaman rekam satker admin mengisi seluruh field kemudian menekan tombol "simpan".



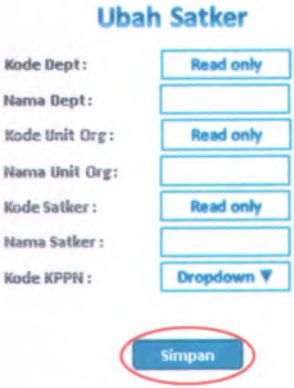
3 Halaman pengaturan sater kembali ditampilkan


Aplikasi Perencanaan Pencairan Dana
Selamat datang Admin : $nama
NIP : $nip
Pengaturan Satker
Kode Dept
Nama Dept
Kode Unit Org
Nama Unit Org
Kode Satker
Nama Satker
Kode KPPN
025
Kemen ag
04
Ditjen Pend. Islam
554155
MAN Krui
145
ubah
hapus
025
Kemen ag
03
Ditjen Bimas Islam
547513
Kemen ag Kab. Lambar
145
ubah
hapus
004
BPK
01
Sekjen
450774
BPK Prov Aceh
001
ubah
hapus
rekam

GG. Storyboard Admin: ubah satker

1 Pada halaman pengaturan satker admin menekan link "ubah".

2 Pada halaman ubah satker admin mengisi seluruh field kemudian menekan tombol "simpan".



3 Halaman pengaturan sater kembali ditampilkan

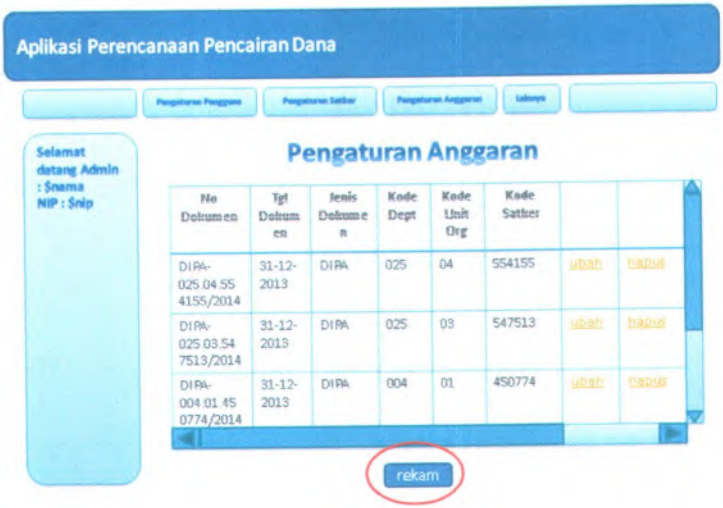
HH. Storyboard Admin: hapus satker

1 Pada halaman pengaturan satker admin menekan link "hapus".

2 Halaman pengaturan satker ditampilkan kembali.

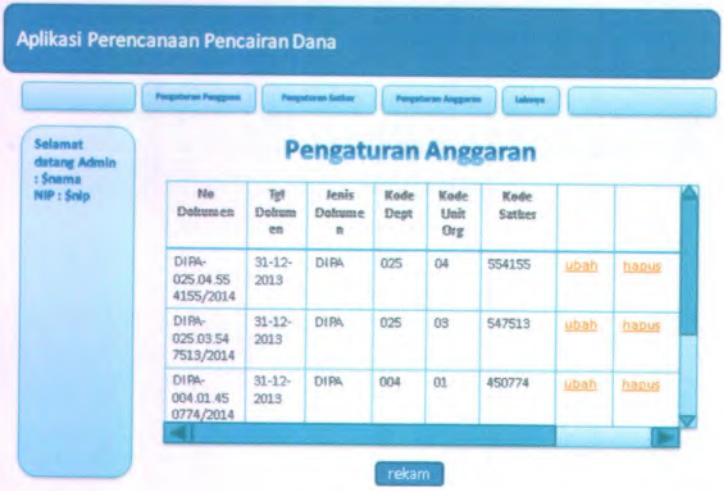
II. Storyboard Admin: rekam anggaran

1 Pada halaman pengaturan anggaran admin menekan tombol "rekam"



2 Pada halaman rekam anggaran admin mengisi seluruh field dan menekan tombol "simpan"

3 Halaman pengaturan anggaran kembali ditampilkan



JJ. Storyboard Admin: ubah anggaran

1 Pada halaman pengaturan anggaran admin menekan link "ubah"

2 Pada halaman ubah anggaran admin mengisi seluruh field dan menekan tombol "simpan"

3 Halaman pengaturan anggaran kembali ditampilkan

KK. Storyboard Admin: hapus anggaran

1 Pada halaman pengaturan anggaran admin menekan link "hapus"

2 Halaman pengaturan anggaran kembali ditampilkan

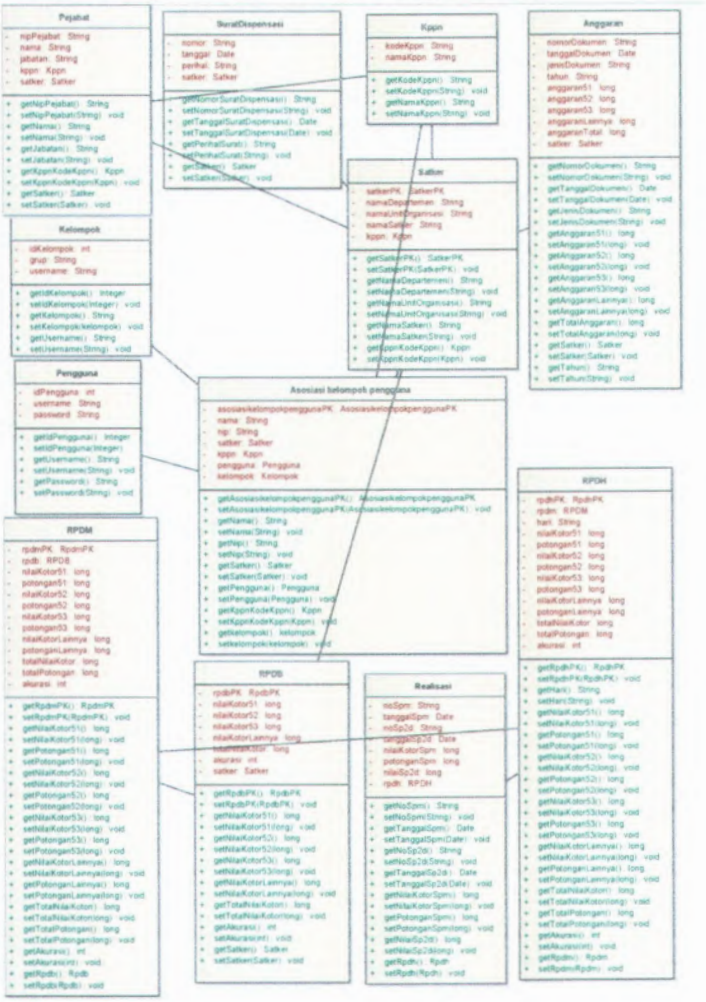
LAMPIRAN B
ANALISIS MODEL

Lampiran B terdiri dari 5 bagian, yaitu:

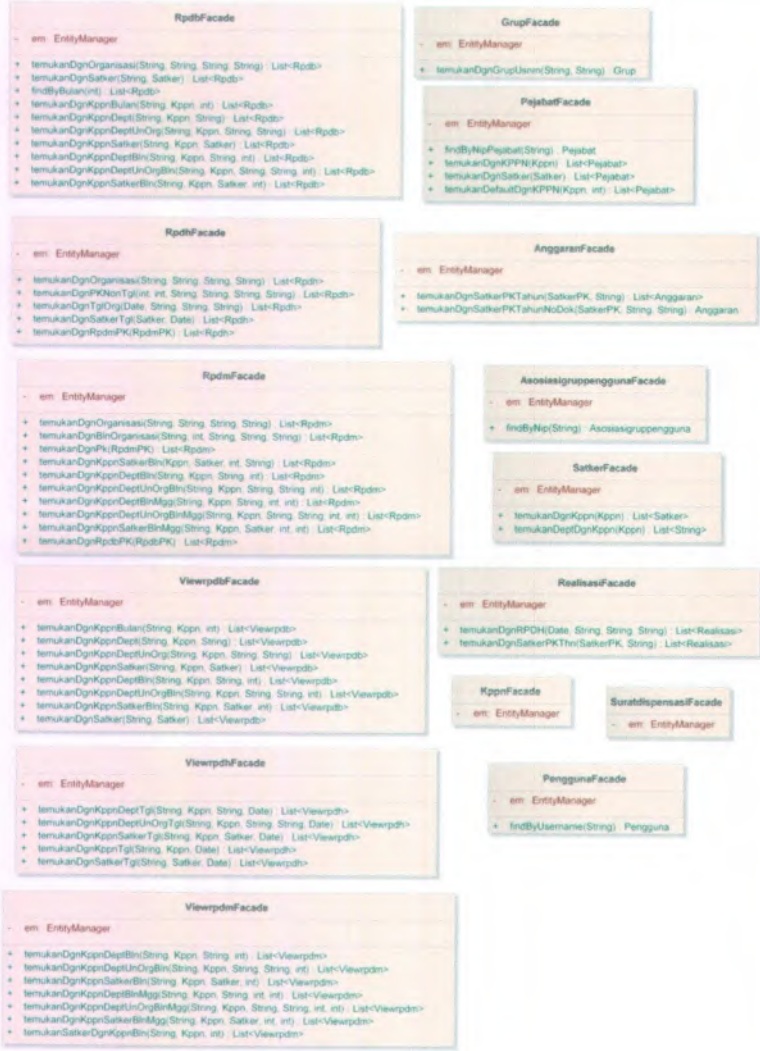
1. Diagram Class;
2. Diagram Use Case;
3. Deskripsi Diagram Use Case;
4. *Diagram Robustness;*
5. Diagram Sequences.

1. CLASS DIAGRAMS

Class Diagram pada Package Entity



Class Diagram pada Package Session



Class Diagram pada Package Controller

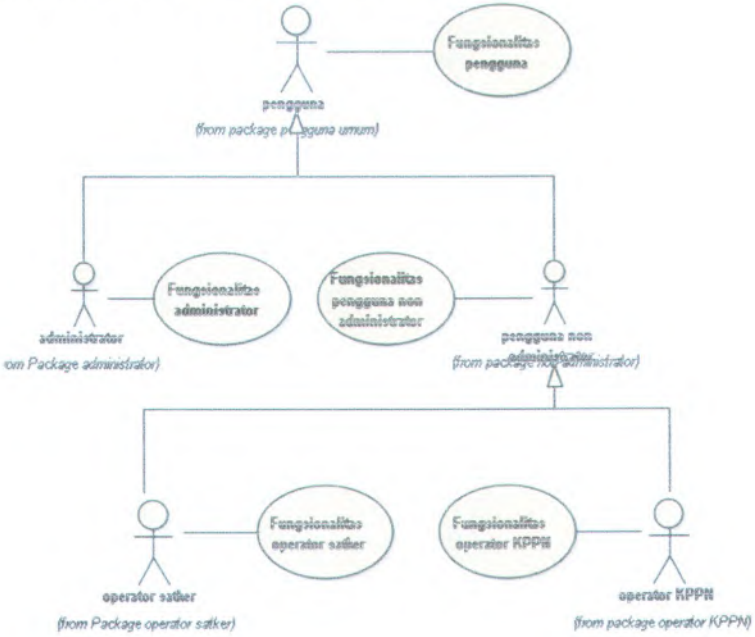
Class Diagram pada Package Validasi

Validator

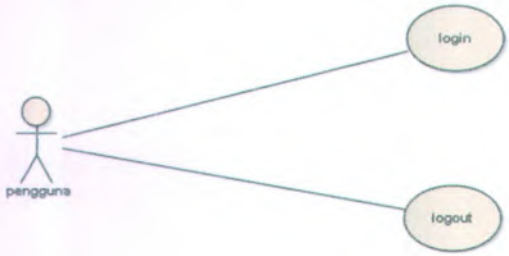
- validasiFormUbahRB(String, String, String, String, HttpServletRequest) : boolean
- validasiFormRekamRH(String, String, String, String, String, String, String, String, HttpServletRequest) : boolean
- validasiFormUbahPengguna(String, String, HttpServletRequest) : boolean
- validasiFormRekamPejabat(String, String, String, HttpServletRequest) : boolean
- isNumber(String) : boolean
- validasiFormOrg(String, String, String, HttpServletRequest) : boolean
- isNumberOrNull(String) : boolean
- validasiFormOrgTgl(String, String, String, String, HttpServletRequest) : boolean
- isTanggal(String) : boolean
- validasiFormMovingAverage(String, String, String, String, HttpServletRequest) : boolean
- validasiFormSimpleExpoSmooth(String, String, String, String, HttpServletRequest) : boolean
- isPecahan(String) : boolean
- validasiFormHolt(String, String, String, String, String, HttpServletRequest) : boolean
- validasiFormRekamAnggaran(String, String, String, String, String, String, HttpServletRequest) : boolean
- validasiFormRekamSatker(String, String, String, String, String, String, HttpServletRequest) : boolean
- validasiFormRekamPengguna(String, String, String, HttpServletRequest) : boolean
- validasiFormRekamPenggunaKppn(String, String, HttpServletRequest) : boolean
- validasiFormRekamPenggunaSatker(String, String, String, String, HttpServletRequest) : boolean
- validasiFormRekamPenggunaAdmin(String, HttpServletRequest) : boolean
- validasiFormRekamRealisasi(String, String, String, String, String, String, HttpServletRequest) : boolean
- validasiFormRekamSurat(String, String, String, HttpServletRequest) : boolean

2. USECASE DIAGRAMS

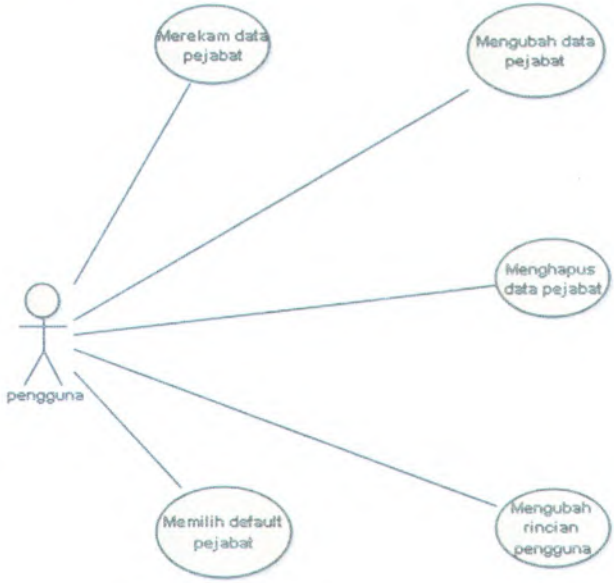
Keseluruhan Diagram Use Case



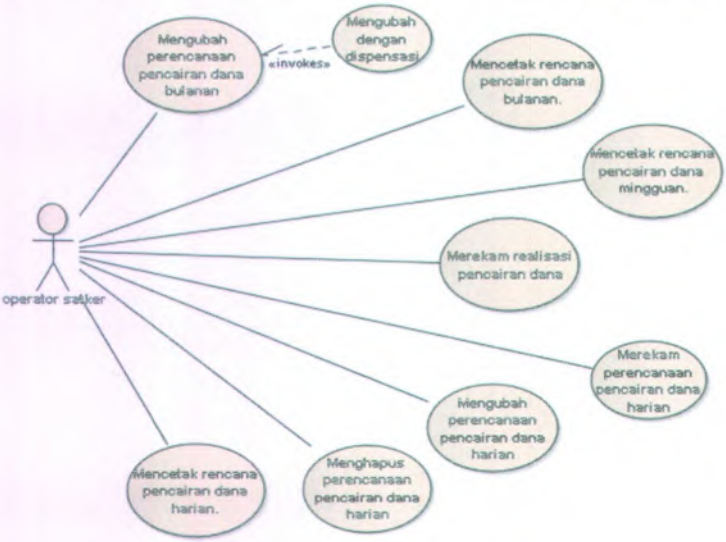
Usecase Diagram untuk Pengguna Umum



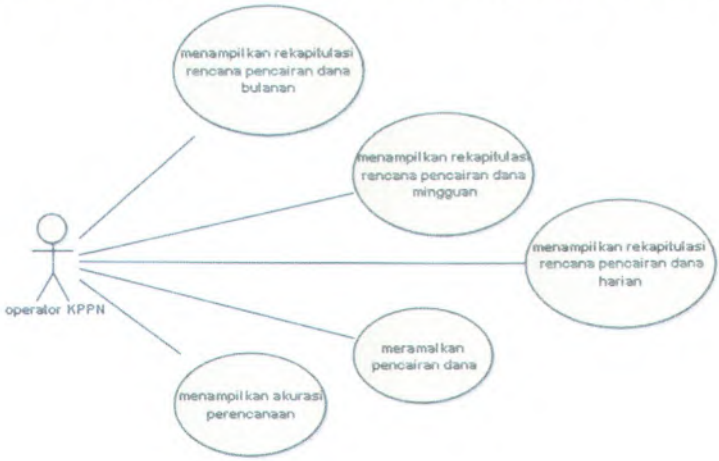
Usecase Diagram untuk Pengguna Non Administrator



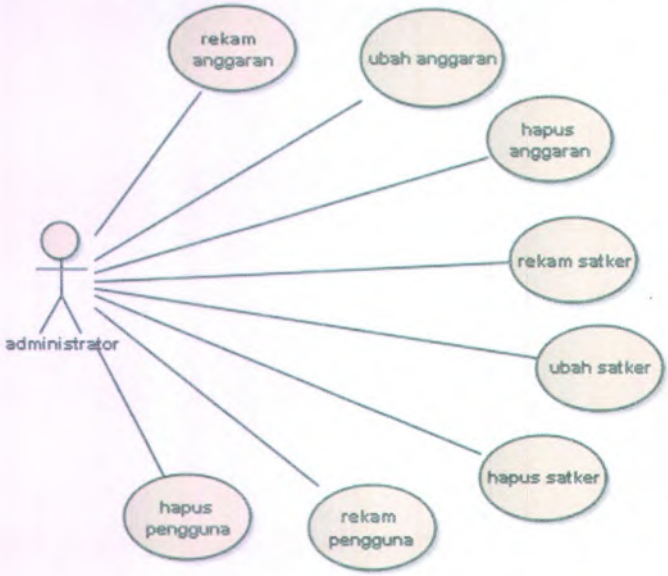
Usecase Diagram dengan Aktor Operator Satker



Usecase Diagram dengan Aktor Operator KPPN



Usecase Diagram dengan Aktor Administrator



3. DESKRIPSI USE CASE

Use Case Melakukan Login (UC01)

Use case name: melakukan login	*ID*: UC01
Primary actor: semua aktor	
Brief description: usecase ini digunakan untuk memverifikasi NIP dan kata sandi yang dimasukkan pengguna yang ingin mengunakan aplikasi.	
Pre-condition: pengguna berada pada halaman index	
Trigger: pengguna mengisi field NIP dan field kata sandi pada form login dan menekan tombol "masuk"; *Basic course*: 1. Sistem melakukan validasi atas data yang dimasukkan; 2. Sistem membandingkan NIP dan kata sandi yang dimasukan dengan NIP dan kata sandi yang tersedia pada database; 3. Sistem memulai sesi pengguna.	
Subflow: -	
Alternate Course: 1 : Jika terdapat field yang kosong, sistem akan menampilkan kembali halaman utama disertai dengan notifikasi bahwa field dimaksud tidak boleh kosong. 2 : Jika NIP dan kata sandi yang dimasukkan tidak sama dengan NIP dan kata sandi yang terdapat pada database, sistem akan menampilkan kembali halaman utama disertai dengan notifikasi "NIP atau kata sandi anda belum benar".	
Post Condition: Sistem menampilkan halaman utama sesuai dengan hak akses.	

Use Case Merekam Data Pejabat (UC02)

Use case name: merekam data pejabat	*ID*: UC02
Primary actor: Operator satker dan operator KPPN	
Brief description: use case ini menjelaskan tentang bagaimana pengguna merekam pejabat yang berwenang ke aplikasi.	
Pre-condition: pengguna berada bada halaman pengaturan pejabat.	
Trigger: pengguna menekan tombol "rekam".	
Basic course: 1. Sistem menampilkan halaman rekam pejabat; 2. Pengguna mengisi filed nama, NIP dan jabatan kemudian menekan tombol "simpan"; 3. Sistem melakukan validasi; 4. Sistem membandingkan data masukkan dengan data di database; 5. Sistem menyimpan data ke database;	
Subflow: -	
Alternate course: 3: jika terdapat field yang kosong, sistem akan menampilkan kembali halaman rekam pejabat disertai dengan notifikasi bahwa field dimaksud tidak boleh kosong. 4: jika terdapat data yang sama di database dengan data masukan kembali halaman rekam pejabat disertai dengan notifikasi bahwa data sudah ada di database.	
Post Condition: sistem menampilkan halaman pengaturan pejabat dengan notifikasi "pejabat berhasil direkam".	

Use Case Mengubah Data Pejabat (UC03)

Use case name: mengubah data pejabat	*ID*: UC03
Primary actor: Operator satker dan operator KPPN	
Brief description: use case ini menjelaskan tentang bagaimana pengguna mengubah pejabat yang berwenang yang telah direkam sebelumnya.	
Pre-condition: pengguna berada pada halaman pengaturan pejabat.	
Trigger: pengguna memilih pejabat dan menekan tombol "ubah".	
Basic course: 1. Sistem menampilkan halaman ubah pejabat dengan field nama, NIP dan jabatan terisi sesuai dengan data pejabat yang dipilih. 2. Pengguna mengubah filed nama, NIP atau jabatan kemudian menekan tombol "simpan". 3. Sistem melakukan validasi atas data yang dimasukkan. 4. Sistem membandingkan data masukkan dengan data di database; 5. Sistem mengganti data lama pejabat di database dengan data yang baru.	
Subflow:	
Alternate course: 3: jika terdapat field yang kosong, sistem akan menampilkan kembali halaman utama disertai dengan notifikasi bahwa field dimaksud tidak boleh kosong. 4: jika terdapat data yang sama di database dengan data masukan kembali halaman rekam pejabat disertai dengan notifikasi bahwa data sudah ada di database.	

Post Condition: sistem menampilkan halaman pengaturan pejabat dengan notifikasi "pejabat berhasil diubah".

Use Case Menghapus Data Pejabat (UC04)

Use case name: menghapus data pejabat	*ID*: UC04
Primary actor: Operator satker dan operator KPPN	
Brief description: use case ini menjelaskan tentang bagaimana pengguna menghapus pejabat yang berwenang yang telah direkam sebelumnya	
Pre-condition: pengguna berada pada halaman pengaturan pejabat.	
Trigger: pengguna memilih pejabat dan menekan tombol "hapus".	
Basic course: 1. Sistem menampilkan kotak konfirmasi penghapusan data pejabat yang berisi tombol "hapus" dan "batal" 2. Pengguna menekan tombol "hapus" pada kotak konfirmasi. 3. Sistem menghapus data pejabat yang bersangkutan pada database.	
Subflow:	
Alternate course: 2 : jika pengguna menekan tombol "batal" pada kotak konfirmasi., sistem akan menutup kotak konfirmasi.	
Post Condition: sistem menampilkan halaman pengaturan pejabat dengan notifikasi "pejabat berhasil dihapus".	

Use Case Memilih Default Pejabat (UC05)

Use case name: memilih default pejabat	*ID*: UC05
Primary actor: Operator satker dan operator KPPN.	
Brief description: use case ini menjelaskan tentang bagaimana pengguna memilih pejabat yang berwenang untuk ditampilkan pada pencetakan laporan.	
Pre-condition: pengguna berada pada halaman pengaturan pejabat.	
Trigger: pengguna memilih pejabat dan menekan tombol "pilih default".	
Basic course: 1. Sistem mengubah default pejabat pada database, dan meyesuaikan default pejabat pada tabel daftar pejabat.	
Subflow:	
Alternate course:	
Post Condition: sistem menampilkan halaman pengaturan pejabat	

Use Case Mengubah Rincian Pengguna (UC06)

Use case name: mengubah rincian pengguna	*ID*: UC06
Primary actor: operator satker dan operator KPPN	
Brief description: use case ini menjelaskan tentang bagaimana pengguna mengubah rincian informasi mengenai dirinya.	
Pre-condition: pengguna berada pada halaman pengaturan.	
Trigger: pengguna menekan tombol "atur pengguna"	
Basic course: 1. Sistem menampilkan halaman pengaturan pengguna yang terdapat field nama dan jabatan pengguna. 2. Pengguna mengisi field yang disediakan kemudian menekan tombol "simpan" 3. Sistem menyimpan data yang dimasukkan ke database.	
Subflow:	
Alternate course: 2: jika pengguna pada halaman pengaturan pengguna menekan tombol "batal" maka sistem akan menampilkan kembali halaman pengaturan.	
Post Condition: sistem menampilkan halaman pengaturan	

Use Case Mengubah Perencanaan Pencairan Dana Bulanan (UC07)

Use case name: mengubah perencanaan pencairan dana bulanan	*ID*: UC07
Primary actor: operator satker	
Brief description: user case ini digunakan untuk mengelola perencanaan pencairan dana bulanan, tidak terdapat use case rekam dan hapus karena dibuat untuk 12 bulan perencanaan dengan nilai default nol (0).	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana bulanan	
Trigger: pengguna menekan ikon "ubah" pada salah satu jenis belanja di tabel perencanaan pencairan dana bulanan.	
Basic course: 1. Sistem menampilkan halaman ubah perencanaan pencairan dana bulanan, dengan field setiap bulan perencanaan terisi dengan data dipilih. 2. Pengguna mengubah nilai yang terdapat pada satu atau lebih field pada bulan setelah bulan berjalan, kemudian menekan tombol "simpan"; 3. Sistem memvalidasi field yang tersedia, memastikan perencanaan tidak melebihi anggaran dan mengganti data perencanaan pencairan dana bulanan di database dengan data yang baru.	
Subflow: 2: Jika pengguna ingin mengubah field yang terdapat pada bulan berjalan atau sebelumnya, maka harus menggunakan usecase mengubah dengan dispensasi.	
Alternate course:	

3: jika terdapat field yang kosong maka sistem akan menampilkan halaman ubah perencanaan pencairan dana bulanan dengan notifikasi field tidak boleh kosong 3: jika perencanaan melebihi anggaran maka sistem akan menampilkan halaman ubah perencanaan *pencairan dana bulanan dengan notifikasi perencanaan* tidak boleh melebihi anggaran
Post Condition: sistem menampilkan kembali halaman perencanaan pencairan dana.

Use Case Mengubah Dengan Dispensasi (UC08)

Use case name: mengubah dengan dispensasi	*ID*: UC08
Primary actor: operator satker	
Brief description: user case ini digunakan untuk mengubah perencanaan pencairan dana bulanan pada bulan berjalan atau sebelumnya.	
Pre-condition: pengguna berada pada halaman ubah perencanaan pencairan dana bulanan.	
Trigger: pengguna menekan tomboi "dispensasi"	
Basic course: 1. Sistem akan menampilkan halaman dispensasi yang terdapat field nomor dan tanggal dokumen 2. Pengguna mengisi field yang disediakan kemudian menekan tombol "simpan". 3. Sistem melakukan validasi field dan menyimpan data yang dimasukan ke database.	
Subflow:	
Alternate course: 3: jika field kosong maka sistem akan menampilkan	

halaman dispensasi dengan notifikasi field tidak boleh kosong.
Post Condition: Sistem menampilkan kembali halaman ubah perencanaan pencairan dana bulanan dengan field yang telah terisi sebelumnya dan mengatur agar field yang sebelumnya tidak dapat diisi menjadi dapat diisi

Post Condition: sistem menampilkan kembali halaman perencanaan pencairan dana bulanan.

Use Case Mencetak Rencana Pencairan Dana Bulanan (UC09)

Use case name: mencetak rencana pencairan dana bulanan.	*ID*: UC09
Primary actor: operator satker.	
Brief description: use case digunakan untuk mencetak perencanaan pencairan dana bulanan.	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana bulanan.	
Trigger: pengguna menekan tombol "cetak"	
Basic course: Sistem mengirimkan paket data ke printer untuk mencetak perencanaan pencairan dana bulanan.	
Subflow: *Alternate course*:	
Post Condition: sistem menampilkan kembali halaman perencanaan pencairan dana.	

Use Case Merekam Perencanaan Pecairan Dana Harian (UC10)

Use case name: merekam perencanaan pecairan dana harian	*ID*: UC10
Primary actor: operator satker	
Brief description: use case ini digunakan untuk menambahkan perencanaan pencairan dana harian.	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana harian	
Trigger: pengguna menekan tombol "rekam"	
Basic course: 1. Sistem menampilkan halaman rekam perencanaan pencairan dana harian, yang terdapat field nilai kotor dan potongan di setiap jenis belanja pada minggu pertama setelah tanggal perencanaan yang belum direkam. 2. Pengguna mengisi field yang disediakan, kemudian menekan tombol "simpan"; 3. Sistem melakukan validasi field, membandingkan dengan batas maksimal, menambahkan data yang dimasukkan ke database	
Subflow:	
Alternate course: 3a: jika terdapat pengisian field yang tidak benar maka sistem akan menampilkan halaman rekam perencanaan pencairan dana harian dengan notifikasi pengisian belum benar 3b: jika perencanaan harian melebihi perencanaan bulanan maka sistem akan menampilkan halaman	

rekam perencanaan pencairan dana harian dengan notifikasi tidak boleh melebihi perencanaan bulanan
Post Condition: sistem menampilkan halaman perencanaan pencairan dana harian

Use Case Mengubah Perencanaan Pencairan Dana Harian (UC11)

Use case name: mengubah perencanaan pencairan dana harian.	*ID*: UC11
Primary actor: operator satker.	
Brief description: use case ini menjelaskan tentang bagaimana mengubah perencanaan pencairan dana harian yang telah dibuat sebelumnya.	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana harian.	
Trigger: pengguna memilih minggu perencanaan yang akan diubah, kemudian menekan ikon "ubah".	
Basic course: 1. Sistem menampilkan halaman ubah perencanaan pencairan dana harian, yang terdapat field nilai kotor dan potongan di setiap jenis belanja pada minggu yang dipilih; 2. Pengguna mengubah satu atau lebih field yang disediakan untuk pencairan pada tanggal setelah tanggal pengubahan, kemudian menekan tombol "simpan"; 3. Sistem melakukan validasi field, membandingkan data dengan perencanaan bulanan, mengganti data	

yang lama di database dengan data yang baru, kemudian menampilkan halaman perencanaan pencairan dana harian.
Subflow:
Alternate course: 3a: jika terdapat pengisian field yang tidak benar maka sistem akan menampilkan halaman ubah perencanaan pencairan dana harian dengan notifikasi pengisian belum benar 3b: jika perencanaan harian melebihi perencanaan bulanan maka sistem akan menampilkan halaman ubah perencanaan pencairan dana harian dengan notifikasi tidak boleh melebihi perencanaan bulanan
Post Condition: : sistem menampilkan halaman perencanaan pencairan dana harian

Use Case Menghapus Perencanaan Pencairan Dana Harian (UC12)

Use case name: menghapus perencanaan pencairan dana harian.	*ID*: UC12
Primary actor: operator satker	
Brief description: use case ini menjelaskan tentang bagaimana menghapus perencanaan pencairan dana harian yang telah dibuat sebelumnya.	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana harian.	
Trigger: pengguna memilih minggu perencanaan yang akan dihapus, kemudian menekan tombol "hapus".	

Basic course: 1. Sistem menampilkan kotak konfirmasi penghapusan; 2. Pengguna menekan tombol "proses"; 3. Sistem menghapus data minggu yang dipilih dari database.
Subflow:
Alternate course:
Post Condition: sistem menampilkan halaman perencanaan pencairan dana harian.

Use Case Merekam Realisasi Pencairan Dana (UC13)

Use case name: merekam realisasi pencairan dana	*ID*: UC13
Primary actor: operator satker	
Brief description: use case ini digunakan untuk menambahkan realisasi ke data perencanaan pencairan dana harian yang telah dibuat sebelumnya.	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana harian.	
Trigger: pengguna memilih minggu perencanaan yang akan dimasukkan realisasinya, kemudian menekan ikon "realisasi".	
Basic course: 1. Sistem menampilkan halaman rekam realisasi pencairan dana, yang terdapat field nilai kotor dan potongan di setiap jenis belanja pada minggu yang	

dipilih; 2. Pengguna mengisi satu atau lebih field yang disediakan pada tanggal sebelum tanggal perekaman, kemudian menekan tombol "simpan"; 3. Sistem memvalidasi field, membandingkan realisasi dengan perencanaan dan akan menyimpan data yang dimasukkan ke database.kemudian menampilkan halaman perencanaan pencairan dana harian.
Subflow:
Alternate course: 3a: jika terdapat pengisian field yang tidak benar maka sistem akan menampilkan rekam realisasi pencairan dana dengan notifikasi pengisian belum benar 3b: jika perencanaan harian melebihi perencanaan bulanan maka sistem akan menampilkan halaman rekam realisasi pencairan dana dengan notifikasi tidak boleh melebihi perencanaan harian.
Post Condition: sistem menampilkan halaman perencanaan pencairan dana harian

Use Case Mencetak Rencana Pencairan Dana Harian (UC14)

Use case name: mencetak rencana pencairan dana harian.	*ID*: UC14
Primary actor: operator satker. *Brief description*: use case digunakan untuk mencetak perencanaan pencairan dana harian pada minggu yang dipilih.	

Pre-condition: pengguna berada pada halaman perencanaan pencairan dana harian.
Trigger: pengguna memilih minggu perencanaan dan menekan tombol "cetak".
Basic course: Sistem mengirimkan paket data ke printer untuk mencetak perencanaan pencairan dana harian.
Subflow:
Alternate course: *Post Condition*: sistem menampilkan halaman perencanaan pencairan dana harian.

Use Case Mencetak Rencana Pencairan Dana Mingguan (UC15)

Use case name: mencetak rencana pencairan dana mingguan.	*ID*: UC15
Primary actor: operator satker	
Brief description: use case digunakan untuk mencetak perencanaan pencairan dana harian pada paket bulan yang dipilih (satu paket bulan adalah satu bulan ganjil dan bulan genap pertama setelahnya).	
Pre-condition: pengguna berada pada halaman perencanaan pencairan dana mingguan.	
Trigger: pengguna memilih paket bulan perencanaan dan menekan tombol "cetak".	
Basic course: Sistem mengirimkan paket data ke printer untuk mencetak	

perencanaan pencairan dana mingguan.
Subflow:
Alternate course:
Post Condition: sistem menampilkan halaman perencanaan pencairan dana mingguan.

Use Case Mencetak Rekapitulasi Rencana Pencairan Dana Bulanan (UC16)

Use case name: mencetak rekapitulasi rencana pencairan dana bulanan.	*ID*: UC16
Primary actor: operator KPPN	
Brief description: use case ini menjelaskan bagaimana pengguna mencetak rekapitulasi rencana pencairan dana bulanan	
Pre-condition: pengguna berada pada halaman rekapitulasi perencanaan pencairan dana bulanan.	
Trigger: pengguna memilih satu atau lebih satker, unit organisasi atau departemen kemudian menekan tombol "cetak"	
Basic course: Sistem mengirimkan paket data ke printer untuk mencetak rekapitulasi perencanaan pencairan dana bulanan.	
Subflow:	
Alternate course:	
Post Condition: sistem akan menampilkan halaman rekapitulasi perencanaan pencairan dana bulanan.	

Use Case Mencetak Rekapitulasi Rencana Pencairan Dana Mingguan (UC17)

Use case name: mencetak rekapitulasi rencana pencairan dana mingguan.	*ID*: UC17
Primary actor: operator KPPN	
Brief description: use case ini menjelaskan bagaimana pengguna mencetak rekapitulasi rencana pencairan dana mingguan	
Pre-condition: pengguna berada pada halaman rekapitulasi perencanaan pencairan dana mingguan.	
Trigger: pengguna memilih satu atau lebih satker, unit organisasi atau departemen kemudian menekan tombol "cetak"	
Basic course: 1. Sistem akan menampilkan kotak konfirmasi cetak; 2. Pengguna menekan tombol "proses". 3. Sistem mengirimkan paket data ke printer untuk mencetak rekapitulasi perencanaan pencairan dana mingguan.	
Subflow:	
Alternate course: 2: jika pengguna menekan tombol "batal" pada kotak konfirmasi, sistem akan menutup kotak konfirmasi. *Post Condition*: sistem akan menampilkan halaman rekapitulasi perencanaan pencairan dana mingguan	

Use Case Mencetak Rekapitulasi Rencana Pencairan Dana Harian (UC18)

Use case name: mencetak rekapitulasi rencana pencairan dana harian.	*ID*: UC18
Primary actor: operator KPPN	
Brief description: use case ini menjelaskan bagaimana pengguna mencetak rekapitulasi rencana pencairan dana harian	
Pre-condition: pengguna berada pada halaman rekapitulasi perencanaan pencairan dana harian.	
Trigger: pengguna memilih satu atau lebih satker, unit organisasi atau departemen kemudian menekan tombol "cetak"	
Basic course: 1. Sistem akan menampilkan kotak konfirmasi cetak; 2. Pengguna menekan tombol "proses". 3. Sistem mengirimkan paket data ke printer untuk mencetak rekapitulasi perencanaan pencairan dana harian.	
Subflow:	
Alternate course: 2: jika pengguna menekan tombol "batal" pada kotak konfirmasi, sistem akan menutup kotak konfirmasi. *Post Condition*: sistem akan menampilkan halaman rekapitulasi perencanaan pencairan dana harian	

Use Case Menampilkan Akurasi Perencanaan (UC19)

Use case name: menampilkan akurasi perencanaan	*ID*: UC19
Primary actor: operator KPPN	
Brief description: use case ini menjelaskan bagaimana pengguna menampilkan akurasi perencanaan pecairan dana.	
Pre-condition: pengguna berada di halaman utama	
Trigger: pengguna menekan menu akurasi perencanaan	
Basic course: 1. Sistem menampilkan halaman pengaturan akurasi perencanaan 2. Pengguna memilih jenis perencanaan (bulanan, mingguan atau harian), kemudian menekan tombol "proses" 3. sistem menghitung prosentase realisasi atas perencanaan dan menampilkan hasilnya dengan tabel pada halaman akurasi perencanaan bulanan, mingguan atau harian.	
Subflow:	
Alternate course:	
Post Condition: Sistem menampilkan halaman akurasi perencanaan bulanan, mingguan atau harian	

Use Case Merekam Pengguna (UC20)

Use case name: merekam pengguna	*ID*: UC20
Primary actor: administrator	
Brief description: use case ini digunakan untuk mendaftarkan pengguna aplikasi.	
Pre-condition: administrator berada di halaman pengaturan pengguna.	
Trigger: administrator menekan tombol "rekam"	
Basic course: 1. Sistem menampilkan halaman perekaman pengguna 1 yang terdapat beberapa field NIP, password dan hak akses; 2. Administrator field yang disediakan, kemudian menekan tombol "berikutnya"; 3. Sistem memvalidasi field kemudian menampilkan halaman perekaman pengguna 2 sesuai dengan pilihan hak akses: a. Jika operator satker, maka sistem menampilkan halaman perekaman pengguna 2 dengan field kode departemen, unit organisasi dan satker b. Jika operator KPPN, maka sistem menampilkan halaman perekaman pengguna 2 dengan field kode kppn c. Jika administrator maka, sistem menampilkan halaman perekaman pengguna 2 *dengan field nama dan jabatan* 4. Administrator mengisi field yang disediakan, kemudian menekan tombol "simpan"; 5. Sistem memvalidasi field kemudian menampilkan	

kotak konfirmasi perekaman; 6. Administrator menekan tombol "proses"; 7. Sistem menyimpan data masukan ke database
Subflow:
Alternate course: 3: Jika terdapat field yang kosong, maka sistem akan menampilkan halaman perekaman pengguna 1 dengan notifikasi kesalahan 5: Jika terdapat field yang kosong, maka sistem akan menampilkan halaman perekaman pengguna 2 dengan notifikasi kesalahan
Post Condition: sistem akan menampilkan halaman pengaturan pengguna.

Use Case Menghapus Pengguna (UC21)

Use case name: menghapus pengguna	*ID*: UC21
Primary actor: administrator	
Brief description: use case ini digunakan untuk menghapus pengguna aplikasi	
Pre-condition: administrator berada di halaman pengaturan pengguna.	
Trigger: administrator memilih pengguna dan menekan tombol "hapus"	
Basic course: 1. Sistem akan menampilkan kotak konfirmasi penghapusan; 2. Administrator menekan tombol "proses"; 3. Sistem akan menghapus pengguna yang dipilih dari database dan menyesuaikan tabel pengguna pada	

halaman pengaturan pengguna.
Subflow:
Alternate course:
Post Condition: sistem menampilkan halaman pengaturan pengguna.

Use Case Merekam Satker (UC22)

Use case name: merekam satker	*ID*: UC22
Primary actor: administrator	
Brief description: use case ini digunakan untuk menambahkan satker pada aplikasi.	
Pre-condition: administrator berada di halaman pengaturan satker.	
Trigger: administrator menekan tombol "rekam"	
Basic course: 1. Sistem menampilkan halaman perekaman satker dengan field kode dan nama departemen, unit organisasi dan satker; 2. Administrator mengisi field yang disediakan kemudian menekan tombol "simpan"; 3. Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi perekaman; 4. Administrator menekan tombol "proses"; 5. Sistem menyimpan data masukan ke database dan menyesuaikan tabel satker di halaman pengaturan satker.	
Subflow:	

Alternate course: 3a: jika administrator belum mengisi satu atau lebih field yang disediakan, maka sistem akan menampilkan halaman perekaman satker dengan notifikasi field harus diisi seluruhnya. 3b: *jika terdapat kombinasi kode departemen, unit* organisasi dan satker yang sama di database maka sistem akan menampilkan halaman perekaman satker dengan notifikasi data yang dimasukkan sudah ada di database.
Post Condition: sistem menampilkan halaman pengaturan satker.

Use Case Mengubah Satker (UC23)

Use case name: mengubah satker.	*ID*: UC23
Primary actor: administrator.	
Brief description: use case ini digunakan untuk mengubah satker pada aplikasi.	
Pre-condition: administrator berada di halaman pengaturan satker.	
Trigger: administrator menekan ikon ubah pada salah satu satker di tabel satker.	
Basic course: 1. Sistem akan menampilan halaman pengubahan satker dengan field kode dan nama departemen, unit organisasi dan satker; 2. Administrator mengubah satu atau lebih field yang tersedia kemudian menekan tombol "simpan"; 3. Sistem memvalidasi field,membandingkan data	

masukan dengan database dan menampilkan kotak konfirmasi pengubahan. 4. Administrator menekan tombol "proses"; 5. Sistem akan menggantikan data satker yang dipilih pada database dengan data masukan dan menyesuaikan tabel satker pada halaman pengaturan satker.
Subflow:
Alternate course: 3a: jika administrator mengosongi satu atau lebih field yang disediakan, maka sistem akan menampilkan halaman perekaman satker dengan notifikasi field harus diisi seluruhnya. 3b: jika terdapat kombinasi kode departemen, unit organisasi dan satker yang sama di database maka sistem akan menampilkan halaman perekaman satker dengan notifikasi data yang dimasukkan sudah ada di database.
Post Condition: sistem menampilkan halaman pengaturan satker.

Use Case Menghapus Satker (UC24)

Use case name: menghapus satker	*ID*: UC24
Primary actor: administrator.	
Brief description: use case ini digunakan untuk menghapus satker pada aplikasi.	
Pre-condition: administrator berada di halaman pengaturan satker.	
Trigger: administrator menekan ikon hapus pada salah satu satker di tabel satker.	
Basic course: 1. Sistem akan menampilkan kotak konfirmasi penghapusan; 2. Administrator menekan tombol "proses"; 3. Sistem akan menghapus satker yang dipilih dari database	
Subflow:	
Alternate course:	
Post Condition: sistem akan menampilkan halaman pengaturan satker.	

Use Case Merekam Anggaran (UC25)

Use case name: merekam anggaran	*ID*: UC25
Primary actor: administrator	
Brief description: use case ini digunakan untuk menambahkan anggaran salah satu satker pada aplikasi.	
Pre-condition: administrator berada di halaman pengaturan anggaran.	
Trigger: administrator menekan tombol "rekam"	
Basic course: 1. Sistem menampilkan halaman perekaman anggaran dengan field kode dan nama departemen, unit organisasi dan satker, jenis, nomor dan tanggal dokumen dan tabel anggaran dengan kolom kode program, fungsi, sub fungsi, kegiatan, sub kegiatan dan jenis belanja serta jumlah dana; 2. Administrator mengisi field yand disediakan, mengisi tabel anggaran dan menekan tombol "simpan"; 3. Sistem memvalidasi field, membandingkan data masukan dengan database dan menampilkan kotak konfirmasi perekaman; 4. Administrator menekan tombol "proses"; 5. Sistem menyimpan data masukan ke database.	
Subflow:	
Alternate course: 3a: jika administrator tidak mengisi satu atau lebih field yang ada dan tidak mengisi tabel sama sekali maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi field dan tabel harus diisi. 3b: jika terdapat nomor dokumen yang sama di	

database maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi data yang dimasukkan sudah ada di database.
Post Condition: sistem menampilkan halaman pengaturan anggaran..

Use Case Mengubah Anggaran (UC26)

Use case name: mengubah anggaran	*ID*: UC26
Primary actor: administrator	
Brief description: use case ini digunakan untuk mengubah anggaran salah satu satker pada aplikasi.	
Pre-condition: administrator berada di halaman pengaturan anggaran.	
Trigger: administrator menekan ikon "ubah" pada salah satu satker pada tabel daftar anggaran	
Basic course: 1. Sistem menampilkan halaman pengubahan anggaran dengan field kode dan nama departemen, unit organisasi dan satker, jenis, nomor dan tanggal dokumen dan tabel anggaran dengan kolom kode program, fungsi, sub fungsi, kegiatan, sub kegiatan dan jenis belanja serta jumlah dana terisi data dari database. 2. Administrator mengubah isi field yang disediakan atau mengubah isi tabel anggaran dan menekan tomboi "simpan"; 3. Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi pengubahan;	

4. Administrator menekan tombol "proses"; 5. Sistem mengganti data lama pada database dengan data masukan.
Subflow:
Alternate course: 3a: jika administrator belum mengisi satu atau lebih field yang disediakan, maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi field harus diisi seluruhnya. 3b: jika terdapat nomor surat yang sama di database maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi data yang dimasukkan sudah ada di database
Post Condition: sistem menampilkan halaman pengaturan anggaran..

Use Case Menghapus Anggaran (UC27)

Use case name: menghapus anggaran	*ID*: UC27
Primary actor: administrator.	
Brief description: use case ini digunakan untuk menghapus anggaran salah satu satker pada aplikasi..	
Pre-condition: administrator berada di halaman pengaturan anggaran.	
Trigger: administrator menekan ikon "hapus" pada salah satu anggaran pada tabel daftar anggaran.	
Basic course: 1. Sistem akan menampilkan kotak konfirmasi penghapusan; 2. Administrator menekan tombol "proses";	

3. Sistem akan menghapus anggaran satker yang dipilih dari database
Subflow:
Alternate course:
Post Condition: sistem akan menampilkan halaman pengaturan anggaran.

Use Case Melakukan Logout (UC28)

Use case name: melakukan logout	*ID*: UC28
Primary actor: semua aktor	
Brief description: use case ini digunakan untuk mengakhiri penggunaan sistem dengan menghentikan hak akses pengguna	
Pre-condition: pengguna berada pada halaman utama	
Trigger: pengguna menekan tombol "logout"	
Basic course: 1. Sistem menghentikan hak akses pengguna;	
Subflow: -	
Alternate Course:	
Post Condition: Sistem menampilkan halaman index dengan form login.	

4. DIAGRAM ROBUSTNESS

Diagram Robustness Melakukan Login (RD01)

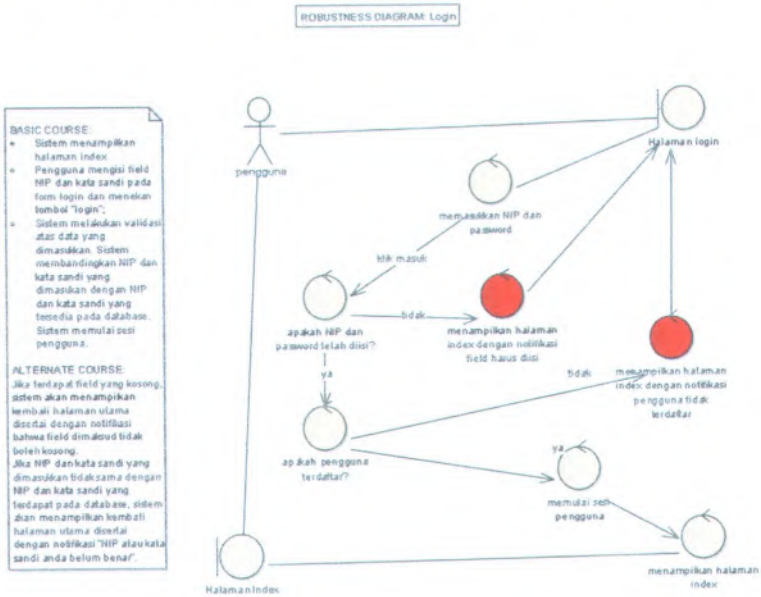


Diagram Robustness Merekam Data Pejabat (RD02)

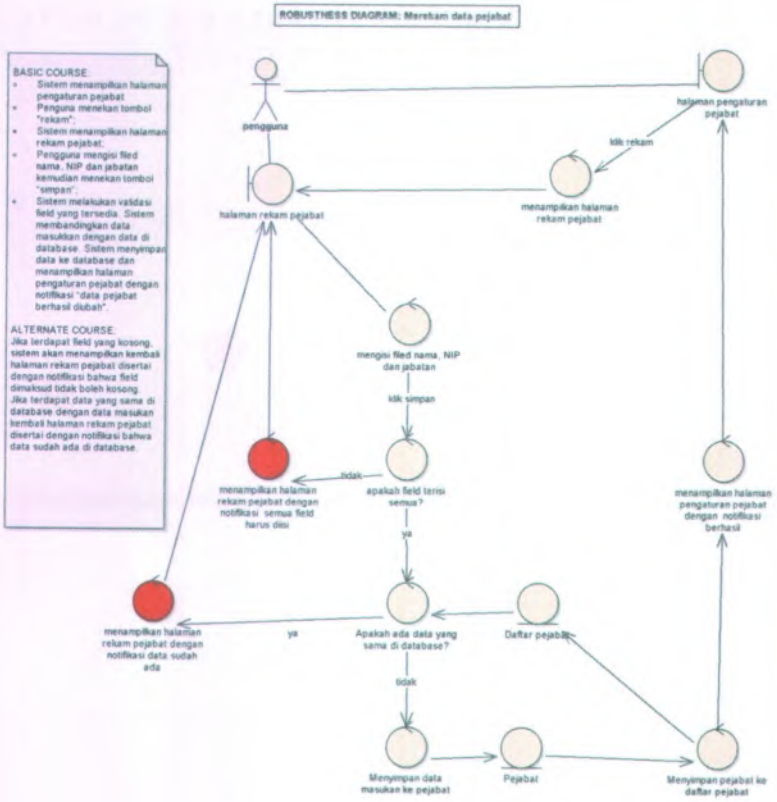


Diagram Robustness Mengubah Data Pejabat (RD03)

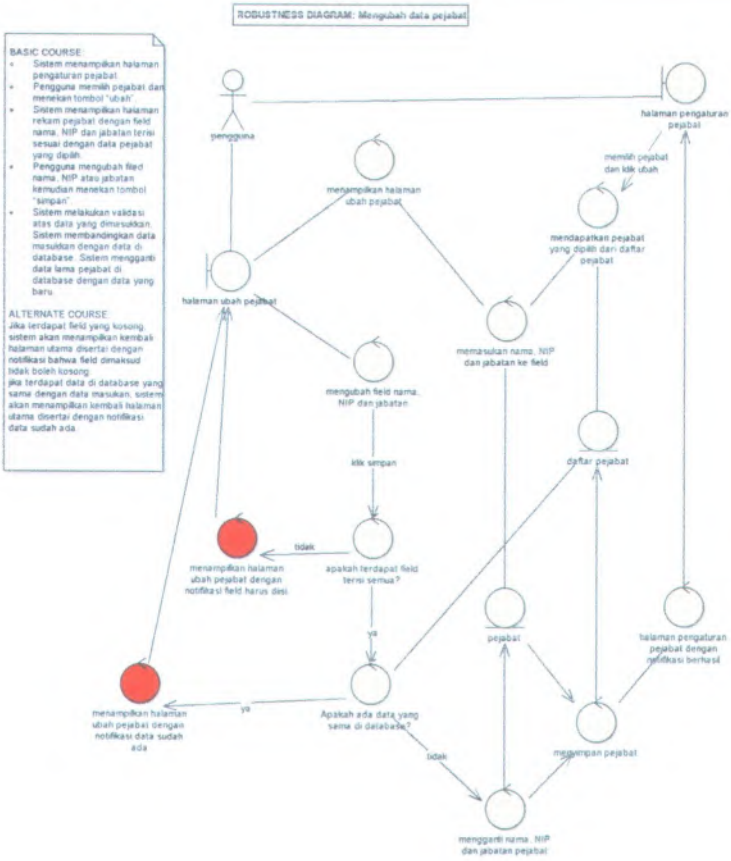


Diagram Robustness Menghapus Data Pejabat (RD04)

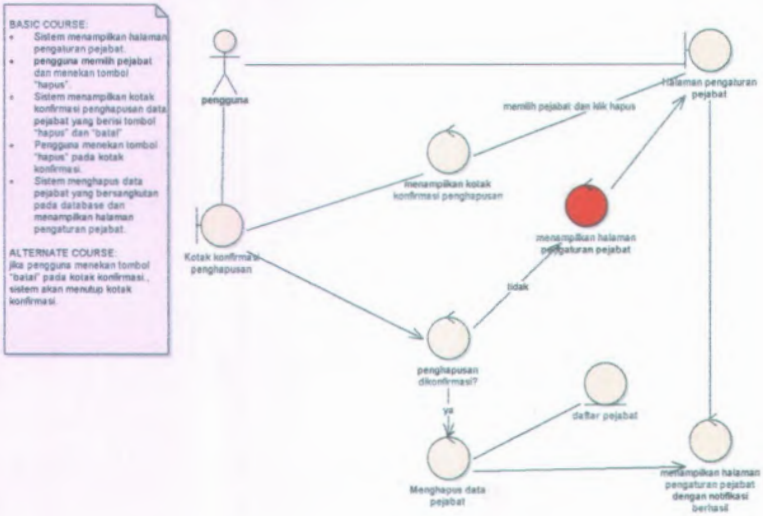


Diagram Robustness Memilih Default Pejabat (RD05)

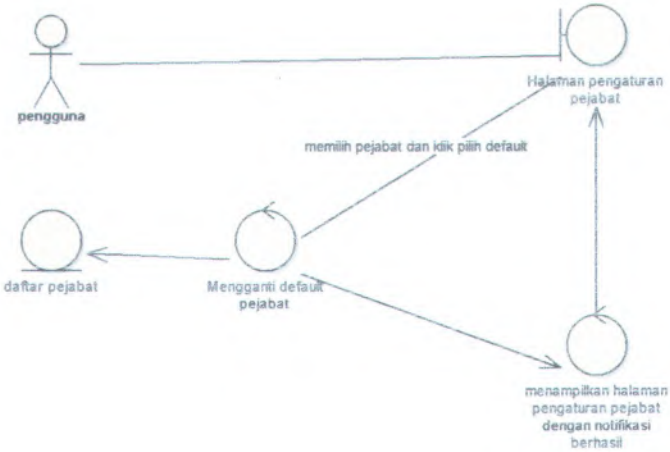


Diagram Robustness Mengubah Rincian Pengguna (RD06)

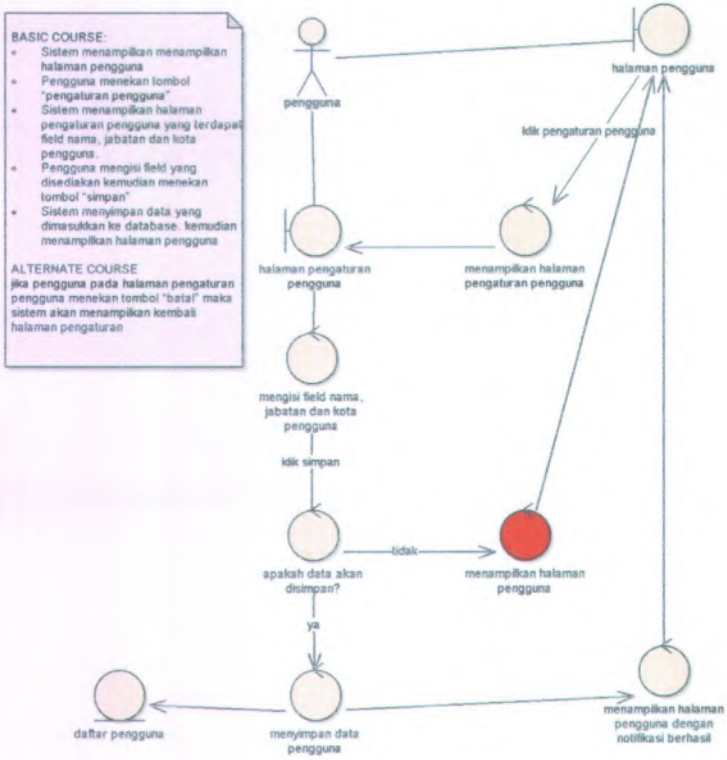


Diagram Robustness Mengubah Perencanaan Pencairan Dana Bulanan (RD07)

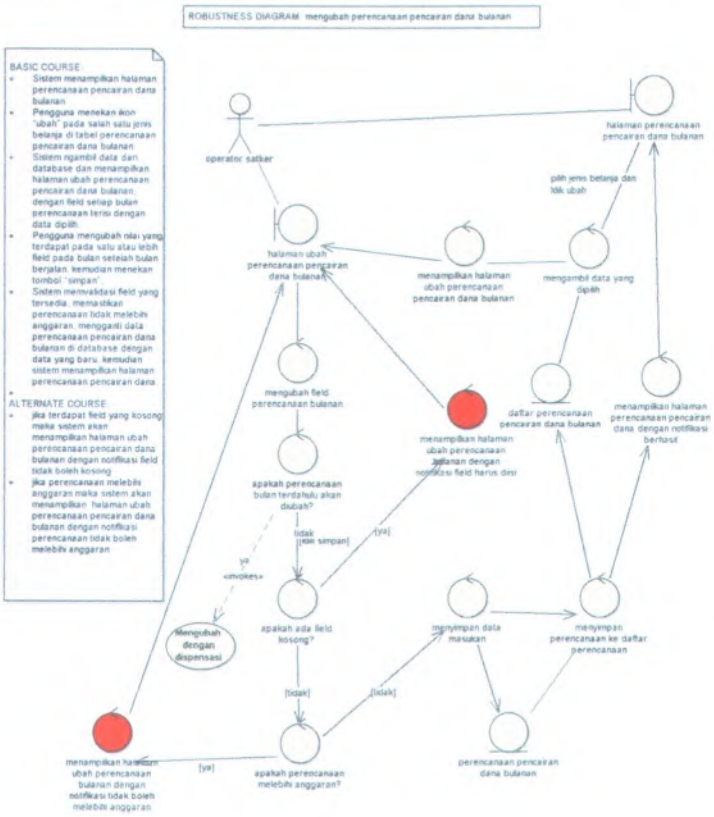


Diagram Robustness Mengubah Dengan Dispensasi (RD08)

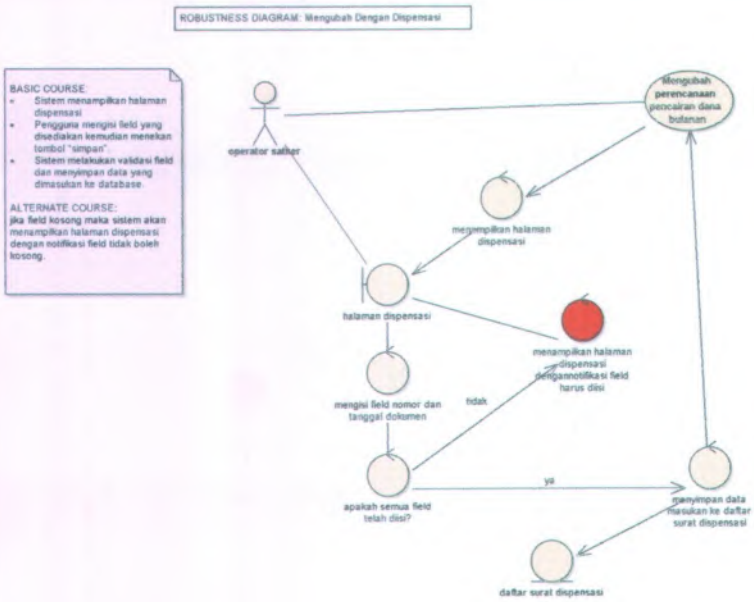


Diagram Robustness Mencetak Rencana Pencairan Dana Bulanan (RD09)

ROBUSTNESS DIAGRAM: mencetak perencanaan pencairan dana bulanan

BASIC COURSE:

- Sistem menampilkan pengaturan halaman perencanaan pencairan dana bulanan.
- pengguna memilih data dan menekan tombol proses
- Sistem menampilkan halaman perencanaan pencairan dana bulanan.
- Pengguna menekan tombol cetak
- Sistem mengirimkan paket data ke printer untuk mencetak perencanaan pencairan dana bulanan dan menampilkan halaman perencanaan pencairan data bulanan

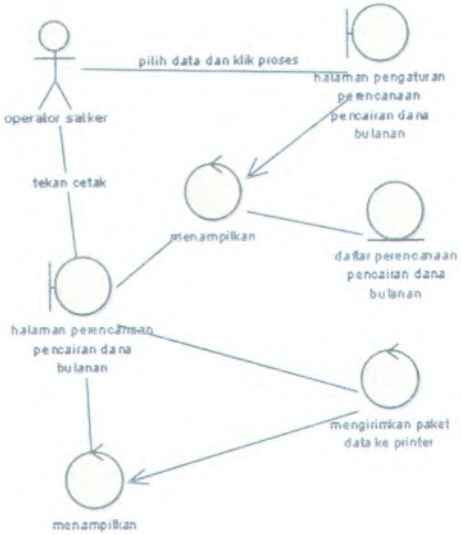


Diagram Robustness Merekam Perencanaan Pecairan Dana Harian (RD10)

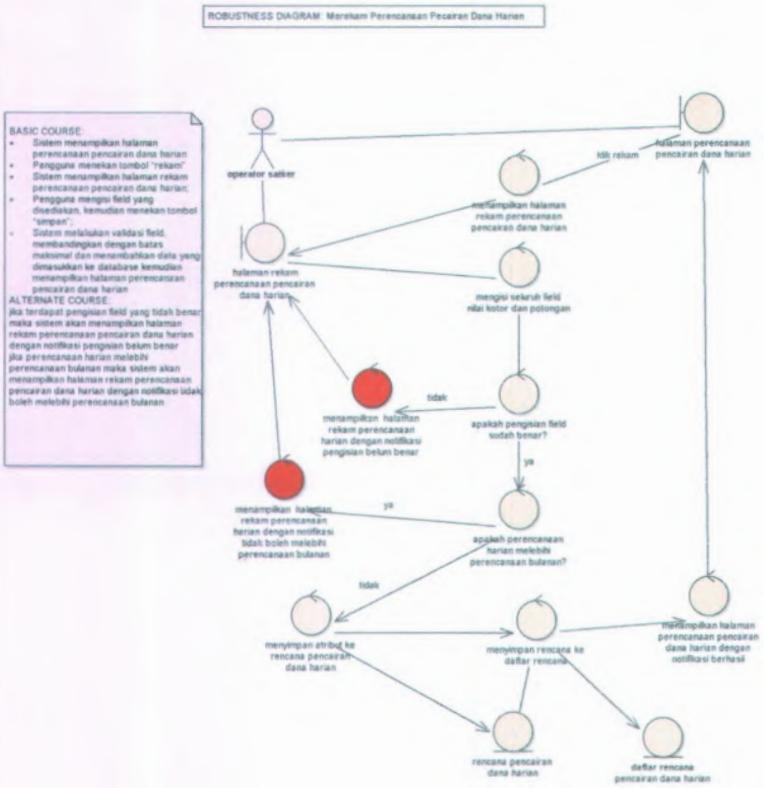


Diagram Robustness Mengubah Perencanaan Pencairan Dana Harian (RD11)

ROBUSTNESS DIAGRAM: Mengubah perencanaan pencairan dana harian

BASIC COURSE:

- sistem menampilkan halaman perencanaan pencairan dana harian.
- Pengguna memilih minggu perencanaan yang akan diubah, kemudian menekan ikon 'ubah'
- Sistem menampilkan halaman ubah perencanaan pencairan dana harian yang terdapat field nilai kotor dan potongan di setiap jenis belanja pada minggu yang dipilih.
- Pengguna mengubah satu atau lebih field yang disediakan untuk pencairan pada tanggal setelah tanggal pengubahan, kemudian menekan tombol 'simpan'
- Sistem melakukan validasi field membandingkan data dengan perencanaan bulanan, mengganti data yang lama di database dengan data yang baru, kemudian menampilkan halaman perencanaan pencairan dana harian

ALTERNATE COURSE:

jika terdapat pengisian field yang tidak benar maka sistem akan menampilkan halaman rekam perencanaan pencairan dana harian dengan notifikasi pengisian belum benar
jika perencanaan harian melebihi perencanaan bulanan maka sistem akan menampilkan halaman rekam perencanaan pencairan dana harian dengan notifikasi tidak boleh melebihi perencanaan bulanan

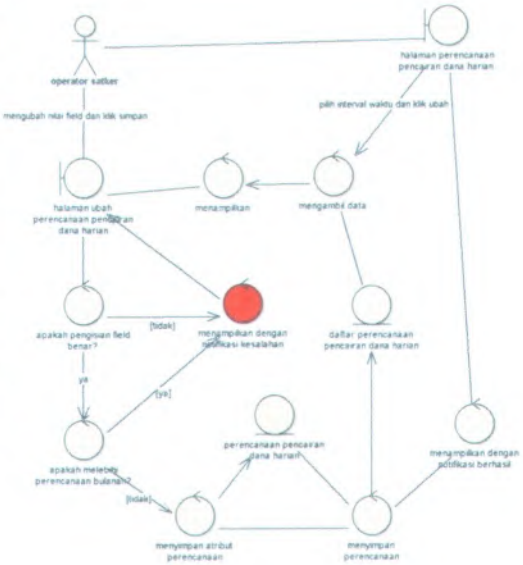


Diagram Robustness Menghapus Perencanaan Pencairan Dana Harian (RD12)

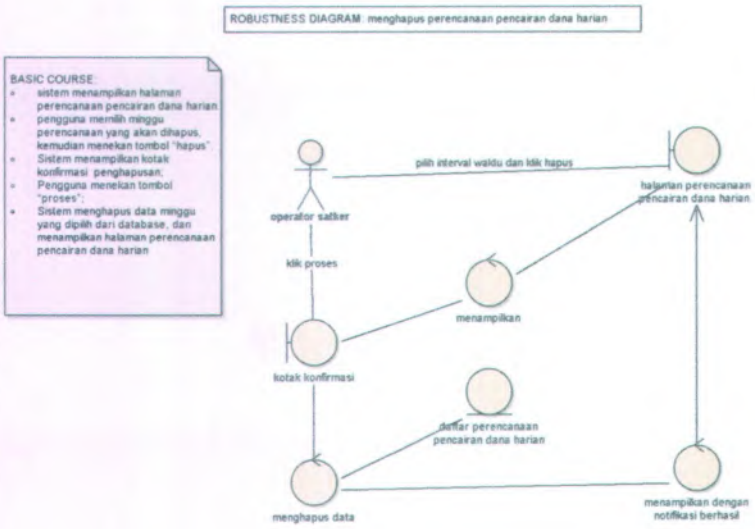


Diagram Robustness Merekam Realisasi Pencairan Dana (RD13)

ROBUSTNESS DIAGRAM: Merekam Realisasi Pencairan Dana

BASIC COURSE

- Sistem menampilkan halaman perencanaan pencairan dana harian
- pengguna memilih minggu perencanaan yang akan dimasukkan realisasinya, kemudian menekan ikon "realisasi".
- Sistem menampilkan halaman rekam realisasi pencairan dana, yang terdapat field nilai kotor dan potongan di setiap jenis belanja pada minggu yang dipilih.
- Pengguna mengisi satu atau lebih field yang disediakan pada tanggal sebelum tanggal perekaman, kemudian menekan tombol "simpan".
- sistem memvalidasi field, membandingkan realisasi dengan perencanaan dan akan menyimpan data yang dimasukkan ke database kemudian menampilkan halaman perencanaan pencairan dana harian

ALTERNATE COURSE:
jika terdapat pengisian field yang tidak benar maka sistem akan menampilkan rekam realisasi pencairan dana dengan notifikasi pengisian belum benar
jika perencanaan harian melebihi perencanaan bulanan maka sistem akan menampilkan halaman rekam realisasi pencairan dana dengan notifikasi tidak boleh melebihi perencanaan harian

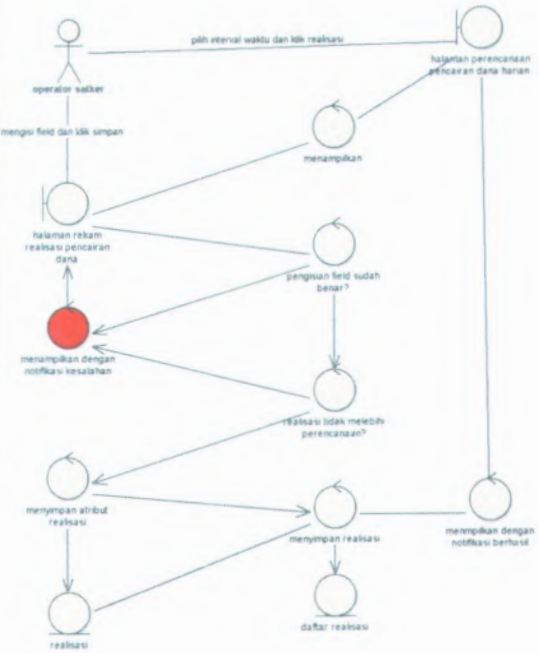


Diagram Robustness Mencetak Rencana Pencairan Dana Harian (RD14)

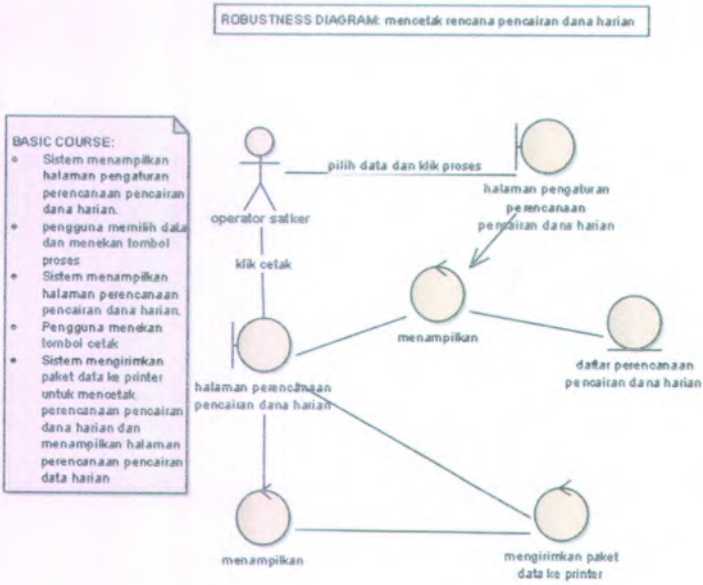


Diagram Robustness Mencetak Rencana Pencairan Dana Mingguan (RD15)

ROBUSTNESS DIAGRAM: mencetak perencanaan pencairan dana mingguan

BASIC COURSE:
- Sistem menampilkan halaman pengaturan perencanaan pencairan dana mingguan.
- pengguna memilih data dan menekan tombol proses
- Sistem menampilkan halaman perencanaan pencairan dana mingguan.
- Pengguna menekan tombol cetak
- Sistem mengirimkan paket data ke printer untuk mencetak perencanaan pencairan dana mingguan dan menampilkan halaman perencanaan pencairan data mingguan

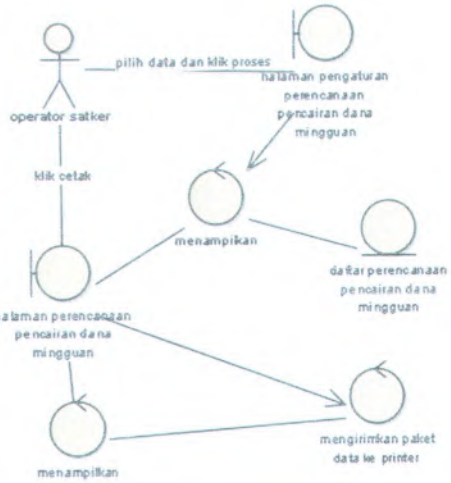


Diagram Robustness Mencetak Rekapitulasi Rencana Pencairan Dana Bulanan (RD16)

ROBUSTNESS DIAGRAM: Mencetak Rekapitulasi Rencana Pencairan Dana Bulanan

BASIC COURSE:
- Sistem menampilkan halaman pengaturan rekapitulasi perencanaan pencairan dana bulanan.
- pengguna memilih data dan klik proses
- Sistem menampilkan halaman rekapitulasi perencanaan pencairan dana bulanan.
- Pengguna menekan tombol cetak
- Sistem mengirimkan paket data ke printer untuk mencetak rekapitulasi perencanaan pencairan dana bulanan dan menampilkan halaman rekapitulasi perencanaan pencairan dana bulanan

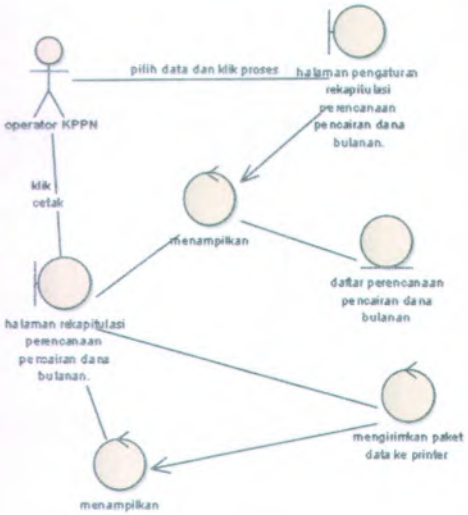


Diagram Robustness Mencetak Rekapitulasi Rencana Pencairan Dana Mingguan (RD17)

ROBUSTNESS DIAGRAM: Mencetak Rekapitulasi Rencana Pencairan Dana mingguan

BASIC COURSE:

- Sistem menampilkan halaman pengaturan rekapitulasi perencanaan pencairan dana mingguan
- pengguna memilih data dan klik proses
- Sistem menampilkan halaman rekapitulasi perencanaan pencairan dana mingguan.
- Pengguna menekan tombol cetak
- Sistem mengirimkan paket data ke printer untuk mencetak rekapitulasi perencanaan pencairan dana mingguan dan menampilkan halaman rekapitulasi perencanaan pencairan dana mingguan

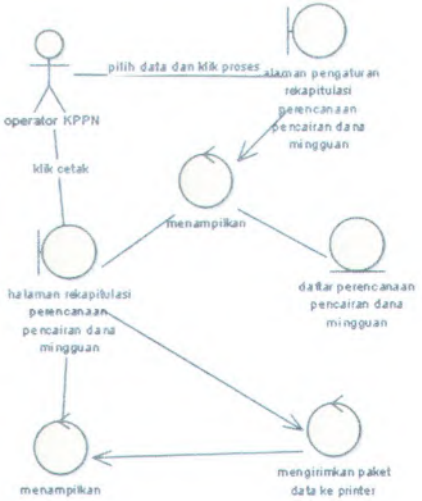


Diagram Robustness Mencetak Rekapitulasi Rencana Pencairan Dana Harian (RD18)

ROBUSTNESS DIAGRAM: Mencetak Rekapitulasi Rencana Pencairan Dana Harian

BASIC COURSE:

- Sistem menampilkan halaman pengaturan rekapitulasi perencanaan pencairan dana harian.
- pengguna memilih data dan klik proses
- Sistem menampilkan halaman rekapitulasi perencanaan pencairan dana harian.
- Pengguna menekan tombol cetak
- Sistem mengirimkan paket data ke printer untuk mencetak rekapitulasi perencanaan pencairan dana harian dan menampilkan halaman rekapitulasi perencanaan pencairan dana harian

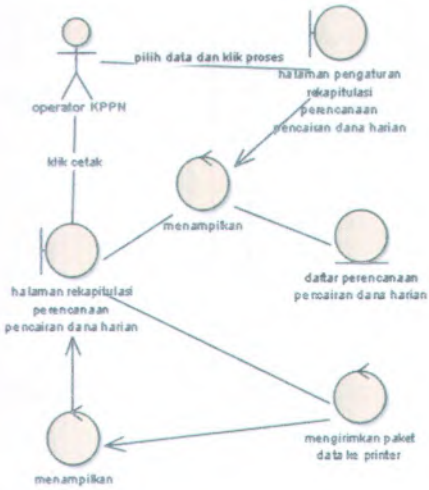


Diagram Robustness Menampilkan Akurasi Perencanaan (RD19)

ROBUSTNESS DIAGRAM: Menampilkan Akurasi Perencanaan

BASIC COURSE:
- Sistem menampilkan halaman pengaturan akurasi perencanaan
- Pengguna memilih jenis perencanaan (bulanan, mingguan atau harian), kemudian menekan tombol "proses"
- sistem menampilkan akurasi dengan tabel pada halaman akurasi perencanaan bulanan, mingguan atau harian

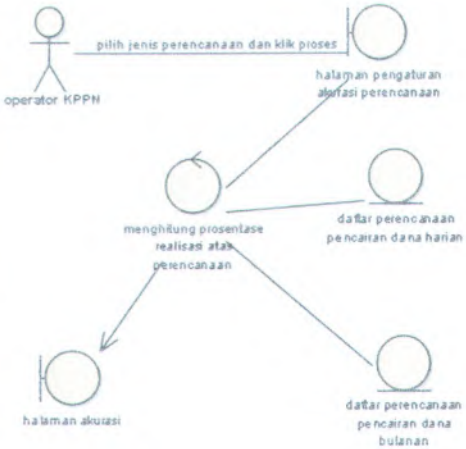


Diagram Robustness Merekam Pengguna (RD20)

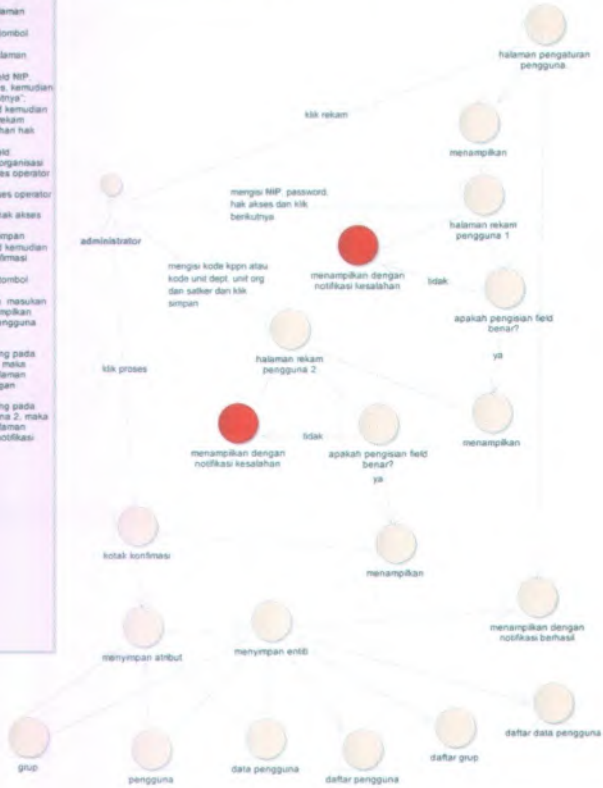
ROBUSTNESS DIAGRAM Merekam Pengguna

BASIC COURSE
- sistem menampilkan halaman pengaturan pengguna
- administrator menekan tombol "rekam"
- Sistem menampilkan halaman rekam pengguna 1
- Administrator mengisi field NIP, password dan hak akses, kemudian menekan tombol "berikutnya";
- Sistem memvalidasi field kemudian menampilkan halaman rekam pengguna 2, sesuai pilihan hak akses
- Administrator mengisi field:

1. kode departemen, unit organisasi dan satker, jika hak akses operator satker
2. kode KPPN jika hak akses operator KPPN
3. nama dan jabatan jika hak akses administrator

kemudian menekan tombol simpan
- Sistem memvalidasi field kemudian menampilkan kotak konfirmasi penyimpanan
- administrator menekan tombol "proses"
- Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan pengguna

ALTERNATE COURSE:
Jika terdapat field yang kosong pada halaman rekam pengguna 1, maka sistem akan menampilkan halaman perekaman pengguna 1 dengan notifikasi kesalahan
Jika terdapat field yang kosong pada halaman perekaman pengguna 2, maka sistem akan menampilkan halaman rekam pengguna 2 dengan notifikasi kesalahan
an



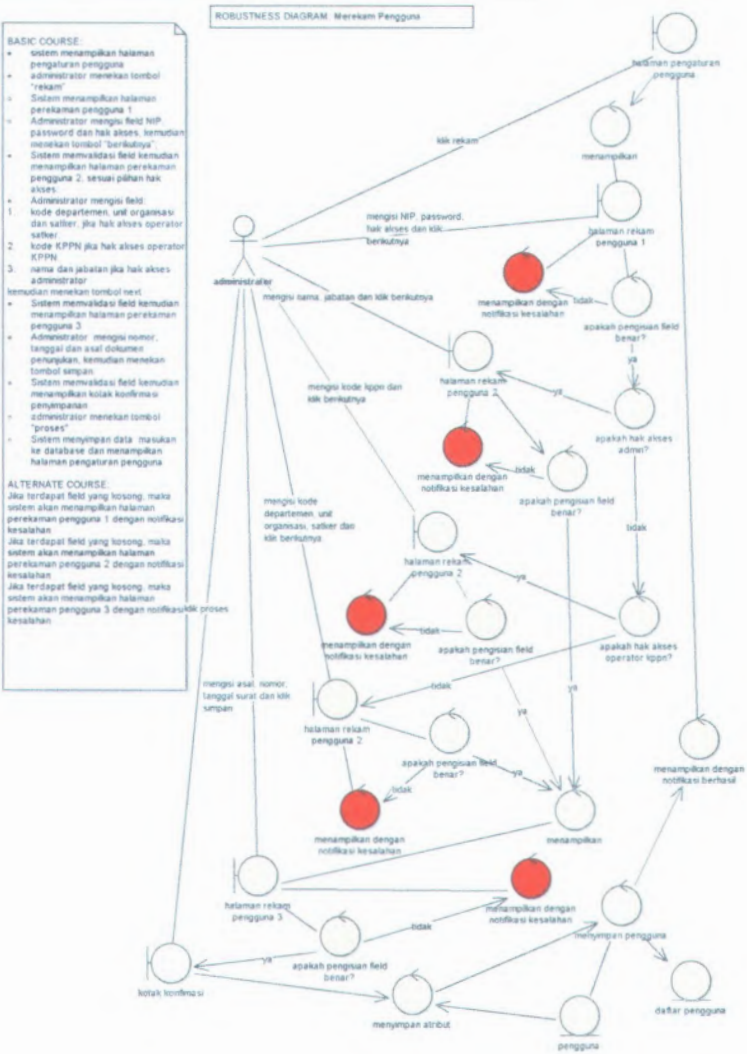

ROBUSTNESS DIAGRAM: Merekam Pengguna
BASIC COURSE:
• sistem menampilkan halaman pengaturan pengguna
• administrator menekan tombol "rekam"
• Sistem menampilkan halaman perekaman pengguna 1
• Administrator mengisi field NIP, password dan hak akses, kemudian menekan tombol "berikutnya"
• Sistem memvalidasi field kemudian menampilkan halaman perekaman pengguna 2, sesuai pilihan hak akses
• Administrator mengisi field:
1. kode departemen, unit organisasi dan satker, jika hak akses operator satker
2. kode KPPN jika hak akses operator KPPN
3. nama dan jabatan jika hak akses administrator
kemudian menekan tombol next
• Sistem memvalidasi field kemudian menampilkan halaman perekaman pengguna 3
• Administrator mengisi nomor, tanggal dan asal dokumen penunjukan, kemudian menekan tombol simpan
• Sistem memvalidasi field kemudian menampilkan kotak konfirmasi penyimpanan
• administrator menekan tombol "proses"
• Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan pengguna
ALTERNATE COURSE:
Jika terdapat field yang kosong, maka sistem akan menampilkan halaman perekaman pengguna 1 dengan notifikasi kesalahan
Jika terdapat field yang kosong, maka sistem akan menampilkan halaman perekaman pengguna 2 dengan notifikasi kesalahan
Jika terdapat field yang kosong, maka sistem akan menampilkan halaman perekaman pengguna 3 dengan notifikasi kesalahan
administrator
halaman pengaturan pengguna
klik rekam
menampilkan
halaman rekam pengguna 1
mengisi NIP, password, hak akses dan klik "berikutnya"
menampilkan dengan notifikasi kesalahan
tidak
apakah pengisian field benar?
ya
mengisi nama, jabatan dan klik berikutnya
halaman rekam pengguna 2
mengisi kode kppn dan klik berikutnya
apakah hak akses admin?
menampilkan dengan notifikasi kesalahan
apakah pengisian field benar?
mengisi kode departemen, unit organisasi, satker dan klik berikutnya
halaman rekam pengguna 2
menampilkan dengan notifikasi kesalahan
apakah pengisian field benar?
apakah hak akses operator kppn?
klik proses
mengisi asal, nomor, tanggal surat dan klik simpan
halaman rekam pengguna 2
apakah pengisian field benar?
menampilkan dengan notifikasi berhasil
menampilkan dengan notifikasi kesalahan
menampilkan
halaman rekam pengguna 3
menampilkan dengan notifikasi kesalahan
menyimpan pengguna
apakah pengisian field benar?
kotak konfirmasi
menyimpan atribut
daftar pengguna
pengguna

Diagram Robustness Menghapus Pengguna (RD21)

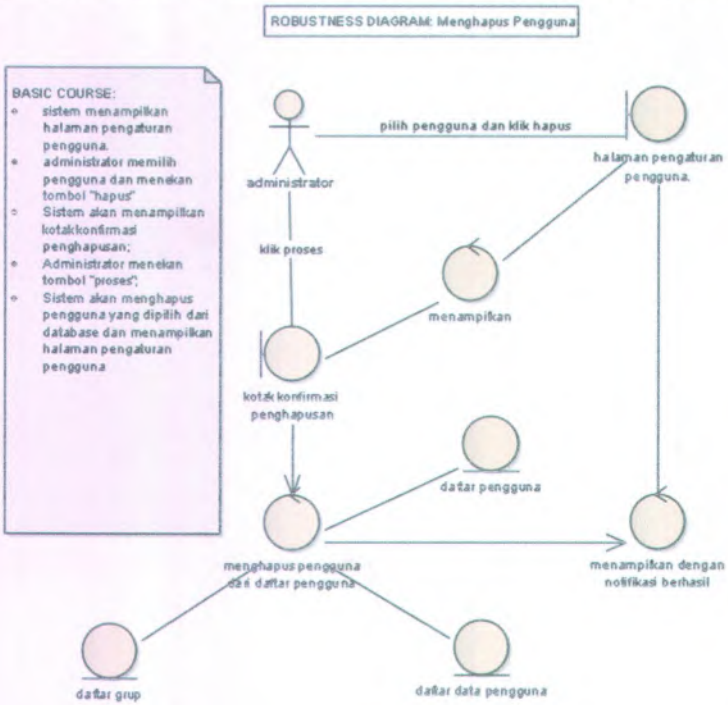


Diagram Robustness Merekam Satker (RD22)

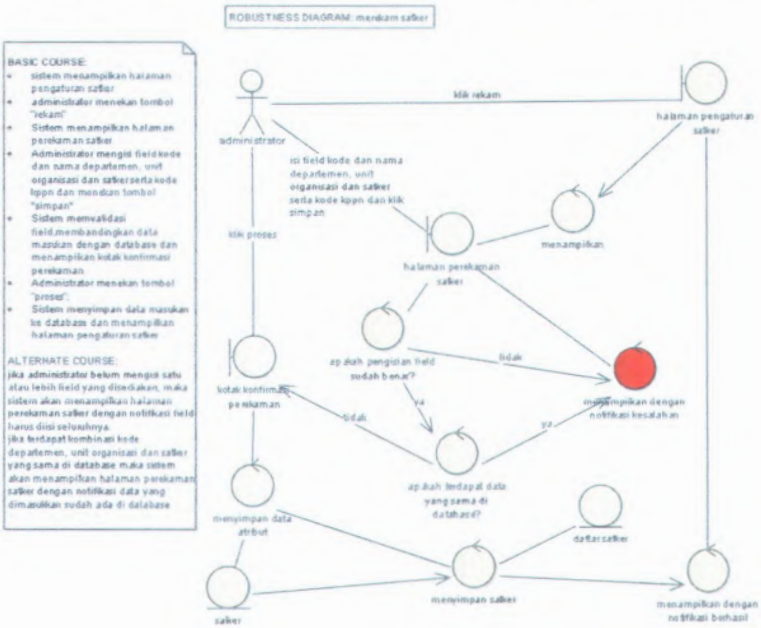


Diagram Robustness Mengubah Satker (RD23)

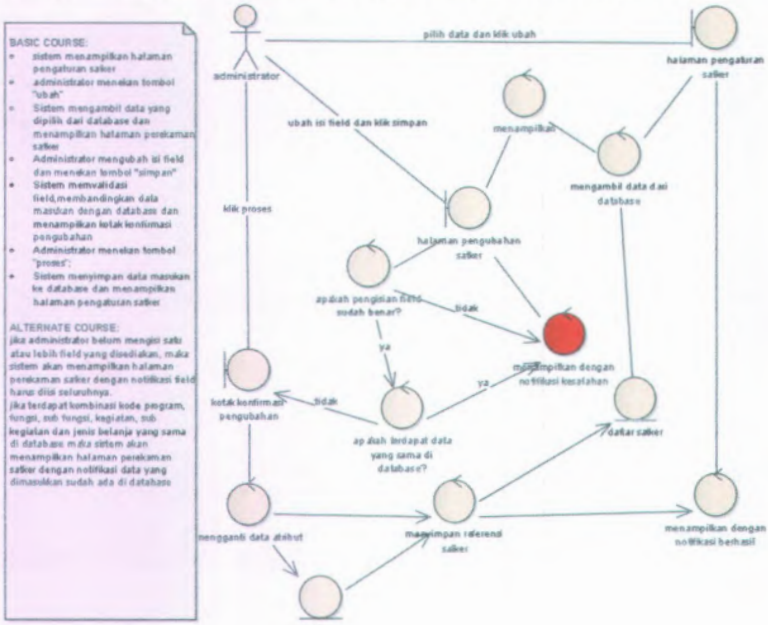


Diagram Robustness Menghapus Satker (RD24)

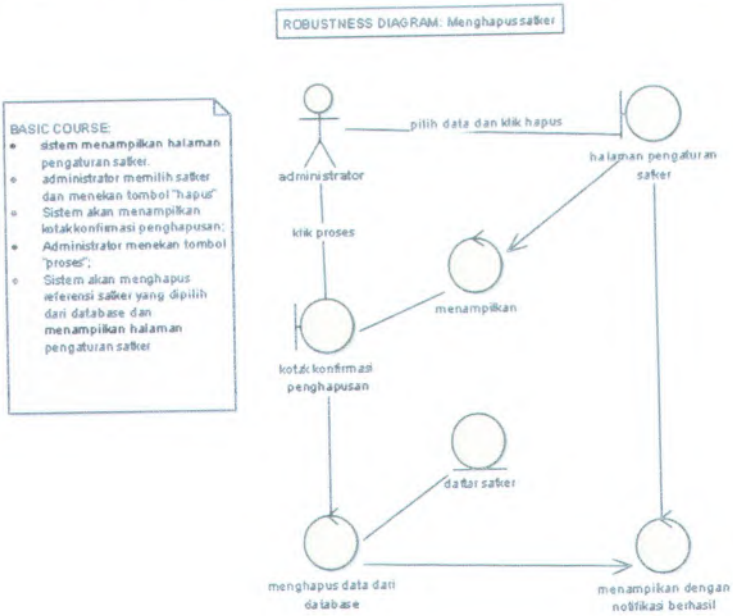


Diagram Robustness Merekam Anggaran (RD25)

BASIC COURSE:

- sistem menampilkan halaman pengaturan pengaturan anggaran
- administrator menekan tombol "rekam"
- Sistem menampilkan halaman perekaman anggaran
- Administrator mengisi field kode dan nama departemen, unit organisasi dan satker, jenis, nomor dan tanggal dokumen dan jumlah anggaran per jenis belanja dan menekan tombol "simpan"
- Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi perekaman
- Administrator menekan tombol "proses";
- Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan anggaran

ALTERNATE COURSE:
jika administrator tidak mengisi satu atau lebih field yang ada dan tidak mengisi tabel sama sekali maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi field dan tabel harus diisi.
jika terdapat nomor dokumen yang sama di database maka sistem akan menampilkan halaman perekaman referensi satker dengan notifikasi data yang dimasukkan sudah ada di database.

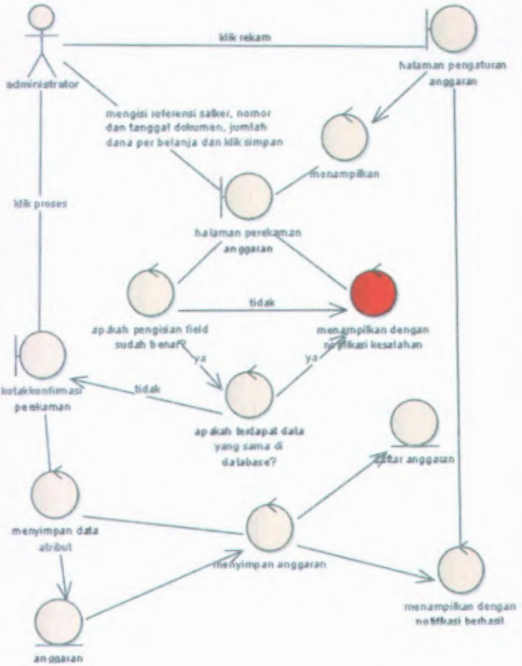


Diagram Robustness Mengubah Anggaran (RD26)

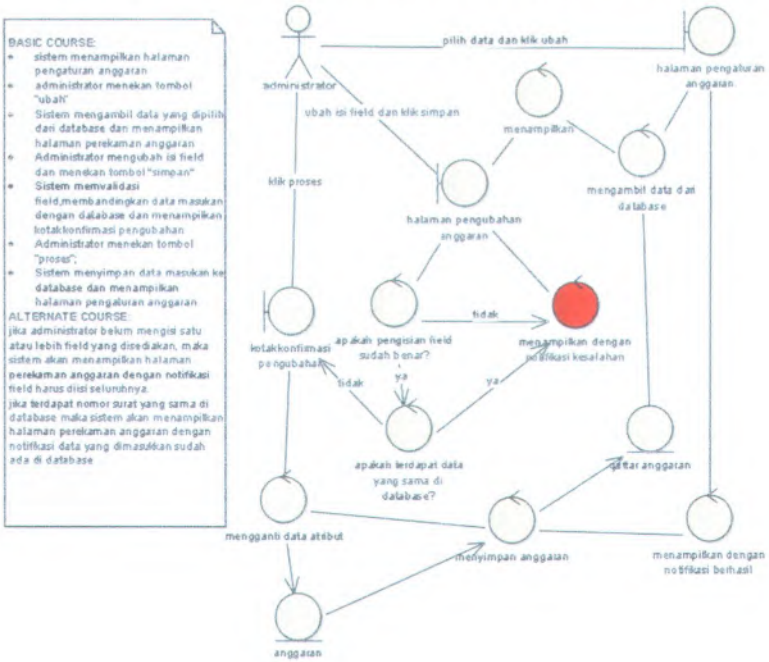


Diagram Robustness Menghapus Anggaran (RD27)

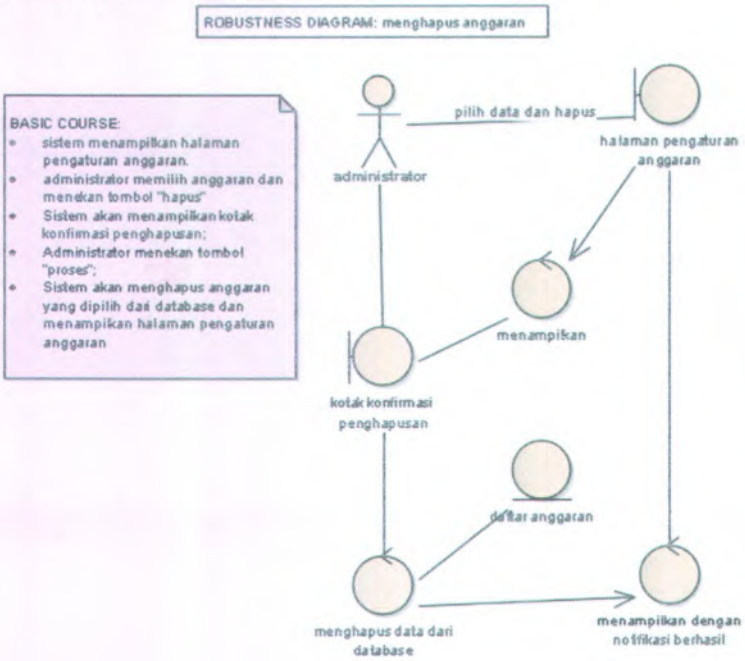
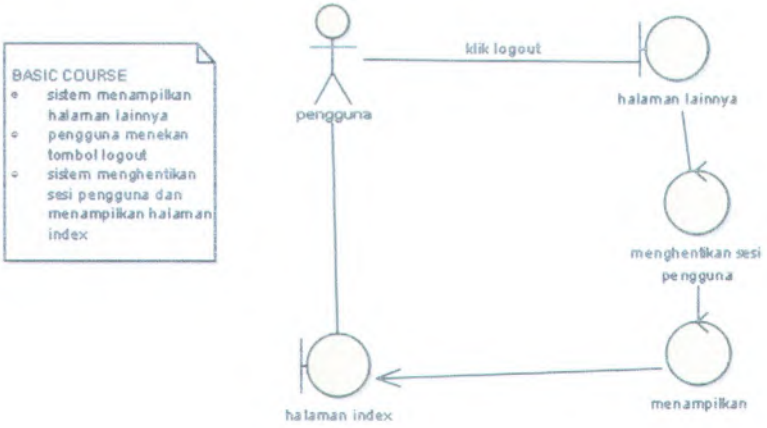


Diagram Robustness Melakukan Logout (RD28)



5. DIAGRAM SEQUENCE

Diagram Sequence Melakukan Login (SD01)

BASIC COURSE:

- Sistem menampilkan halaman index
- Pengguna mengisi field NIP dan kata sandi pada form login dan menekan tombol "login";
- Sistem melakukan validasi atas data yang dimasukkan. Sistem membandingkan NIP dan kata sandi yang dimasukan dengan NIP dan kata sandi yang tersedia pada database. Sistem memulai sesi pengguna.

ALTERNATE COURSE:

Jika terdapat field yang kosong, sistem akan menampilkan kembali halaman utama disertai dengan notifikasi bahwa field dimaksud tidak boleh kosong. Jika NIP dan kata sandi yang dimasukkan tidak sama dengan NIP dan kata sandi yang terdapat pada database, sistem akan menampilkan kembali halaman utama disertai dengan notifikasi "NIP atau kata sandi anda belum benar".

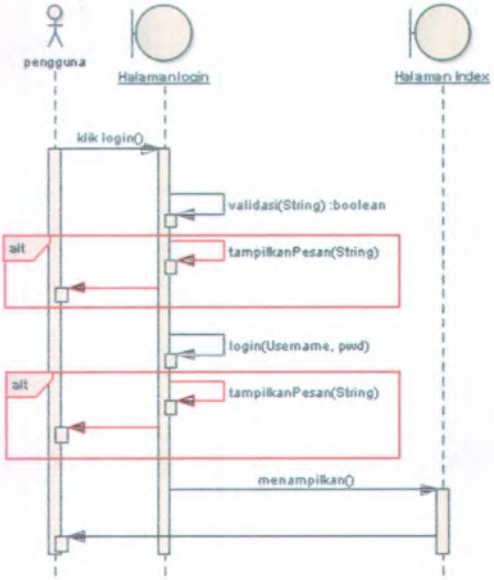


Diagram Sequence Merekam Data Pejabat (SD02)

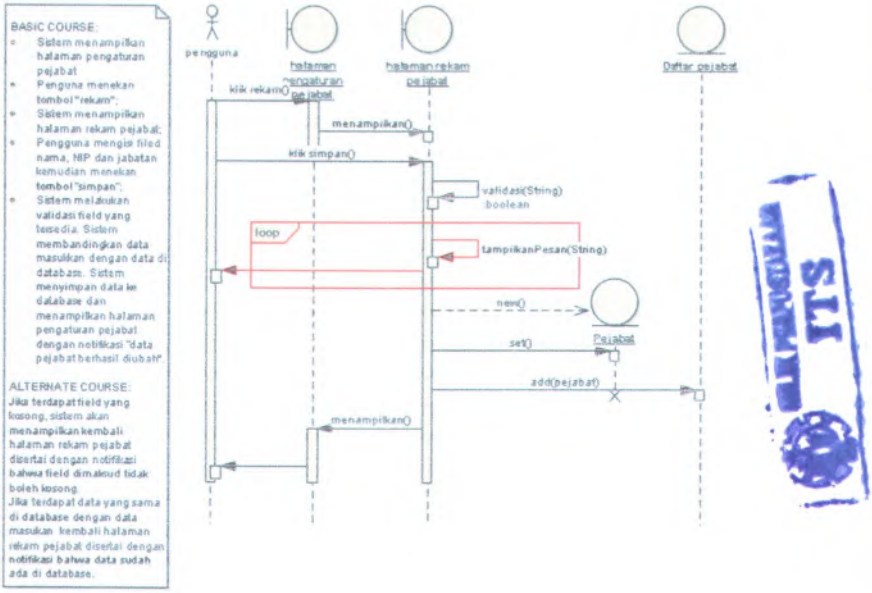


Diagram Sequence Mengubah Data Pejabat (SD03)

BASIC COURSE:

- Sistem menampilkan halaman pengaturan pejabat.
- Pengguna memilih pejabat dan menekan tombol "ubah".
- Sistem menampilkan halaman rekam pejabat dengan field nama, NIP dan jabatan terisi sesuai dengan data pejabat yang dipilih.
- Pengguna mengubah filed nama, NIP atau jabatan kemudian menekan tombol "simpan".
- Sistem melakukan validasi atas data yang dimasukkan. Sistem membandingkan data masukkan dengan data di database. Sistem mengganti data lama pejabat di database dengan data yang baru.

ALTERNATE COURSE:
Jika terdapat field yang kosong, sistem akan menampilkan kembali halaman utama disertai **dengan notifikasi bahwa field** dimaksud tidak boleh kosong. **jika terdapat data di database** yang sama dengan data masukan, sistem akan menampilkan kembali halaman utama disertai dengan notifikasi data sudah ada.

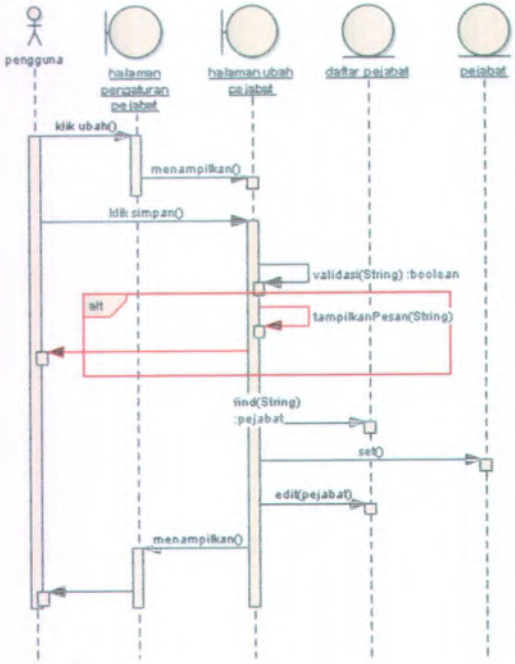


Diagram Sequence Menghapus Data Pejabat (SD04)

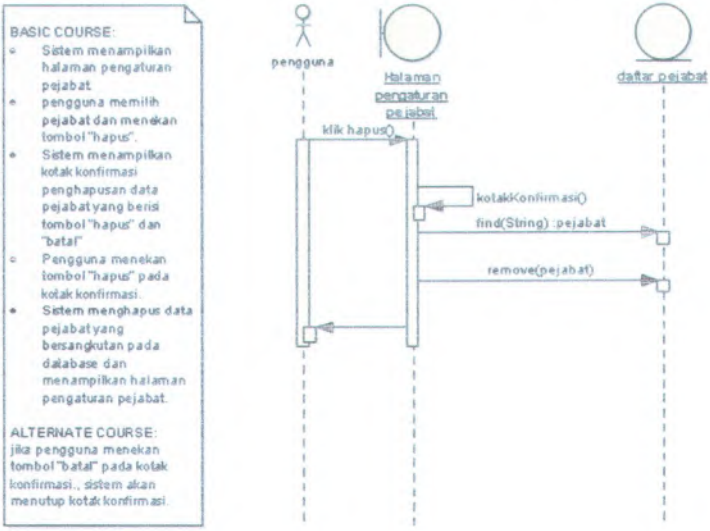


Diagram Sequence Memilih Default Pejabat (SD05)

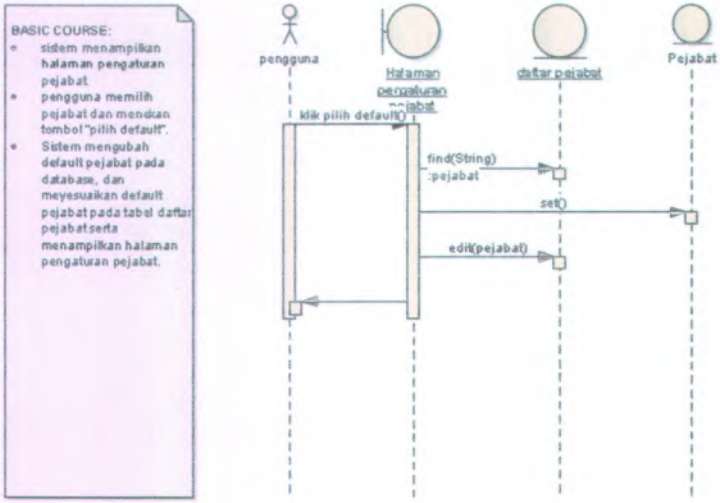


Diagram Sequence Mengubah Rincian Pengguna (SD06)

BASIC COURSE:

- Sistem menampilkan menampilkan halaman pengguna
- Pengguna menekan tombol "pengaturan pengguna"
- Sistem menampilkan halaman pengaturan pengguna yang terdapat field nama, jabatan dan kota pengguna.
- Pengguna mengisi field yang disediakan kemudian menekan tombol "simpan"
- Sistem menyimpan data yang dimasukkan ke database, kemudian menampilkan halaman pengguna

ALTERNATE COURSE
jika pengguna pada halaman pengaturan pengguna menekan tombol "batal" maka sistem akan menampilkan kembali halaman pengaturan

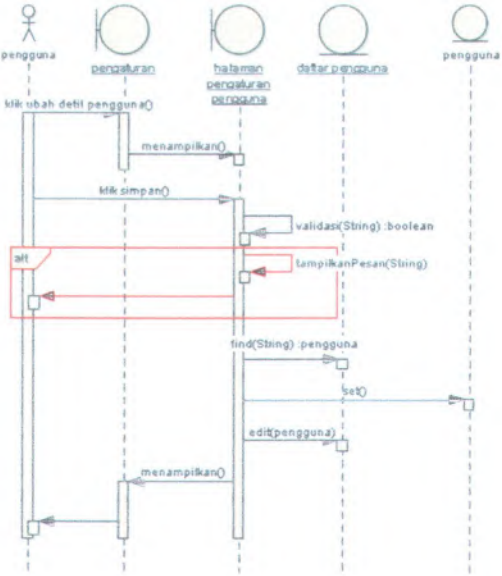


Diagram Sequence Mengubah Perencanaan Pencairan Dana Bulanan (SD07)

BASIC COURSE:

- Sistem menampilkan halaman perencanaan pencairan dana bulanan
- Pengguna menekan ikon "ubah" pada salah satu jenis belanja di tabel perencanaan pencairan dana bulanan
- Sistem ngambil data dari database dan menampilkan halaman ubah perencanaan pencairan dana bulanan, dengan field setiap bulan perencanaan terisi dengan data dipilih.
- Pengguna mengubah nilai yang terdapat pada satu atau lebih field pada bulan setelah bulan berjalan, kemudian menekan tombol "simpan";
- Sistem memvalidasi field yang tersedia, memastikan perencanaan tidak melebihi anggaran, mengganti data perencanaan pencairan dana bulanan di database dengan data yang baru, kemudian sistem menampilkan halaman perencanaan pencairan dana.

ALTERNATE COURSE:

- jika terdapat field yang kosong maka sistem akan menampilkan halaman ubah perencanaan pencairan dana bulanan dengan notifikasi field tidak boleh kosong
- jika perencanaan melebihi anggaran maka sistem akan menampilkan halaman ubah perencanaan pencairan dana bulanan dengan notifikasi perencanaan tidak boleh melebihi anggaran

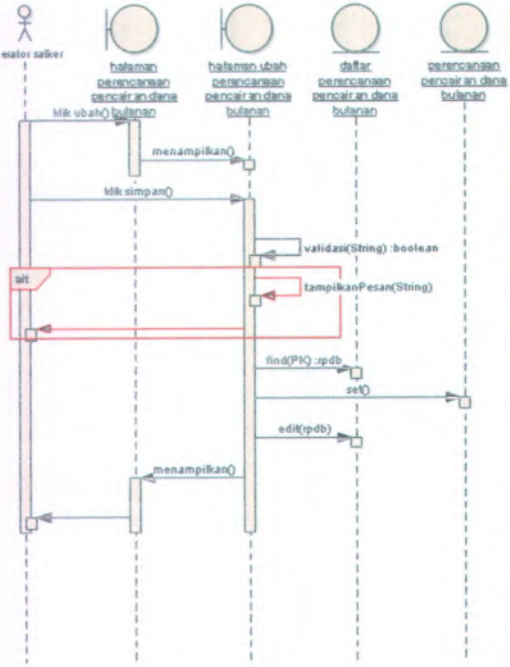


Diagram Sequence Mengubah Dengan Dispensasi (SD08)

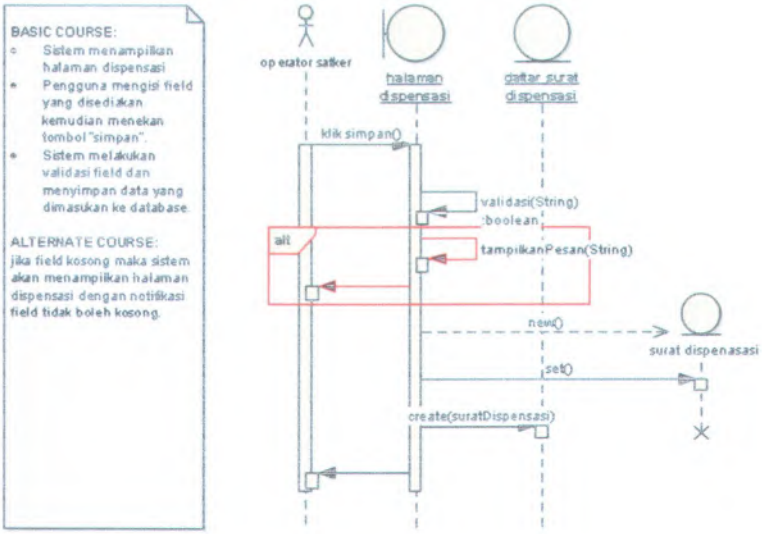


Diagram Sequence Mencetak Rencana Pencairan Dana Bulanan (SD09)

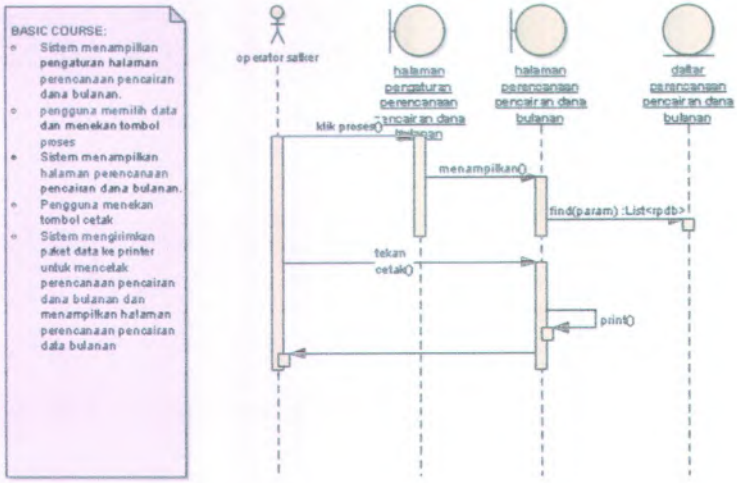


Diagram Sequence Merekam Perencanaan Pecairan Dana Harian (SD10)

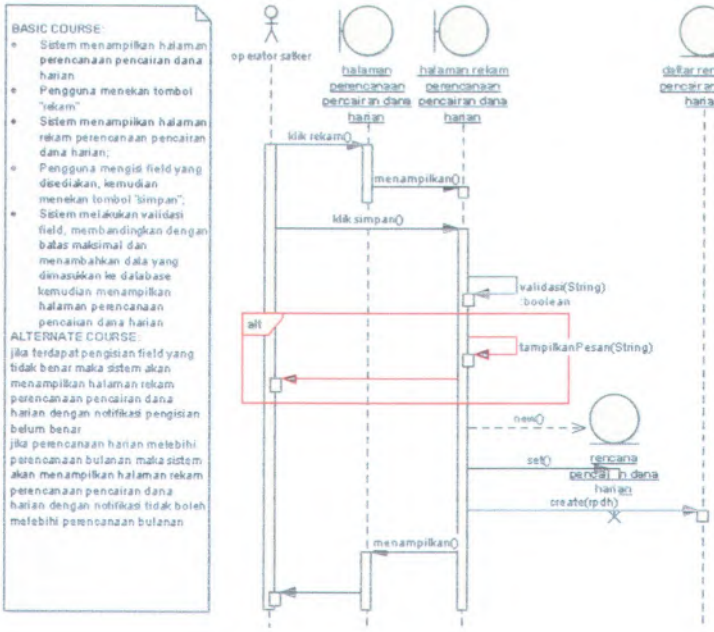


Diagram Sequence Mengubah Perencanaan Pencairan Dana Harian (SD11)

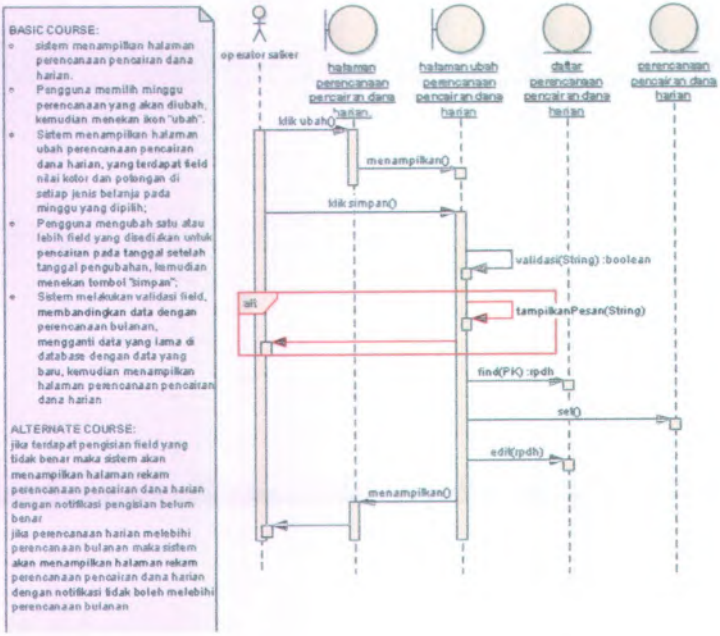


Diagram Sequence Menghapus Perencanaan Pencairan Dana Harian (SD12)

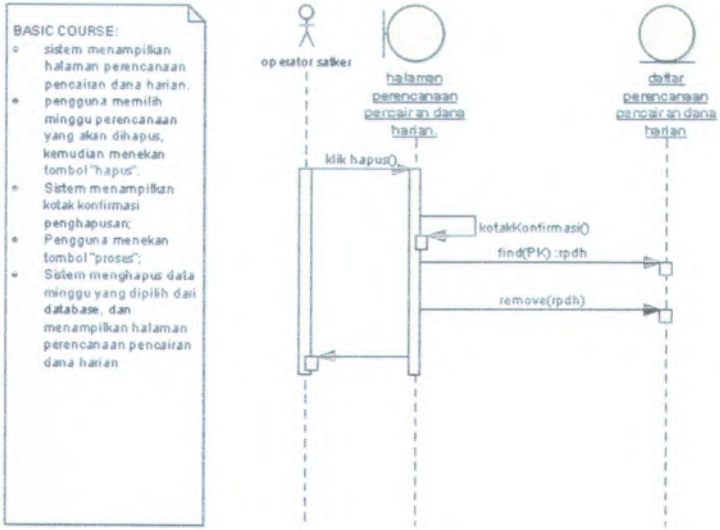


Diagram Sequence Merekam Realisasi Pencairan Dana (SD13)

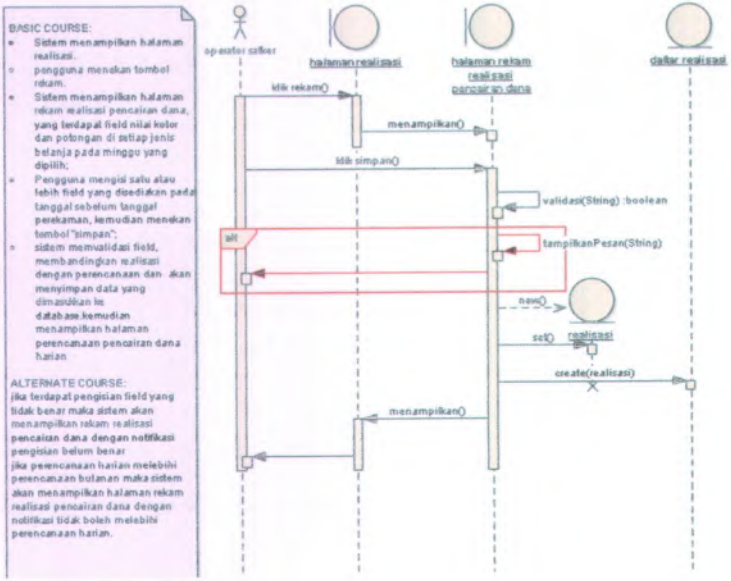


Diagram Sequence Mencetak Rencana Pencairan Dana Harian (SD14)

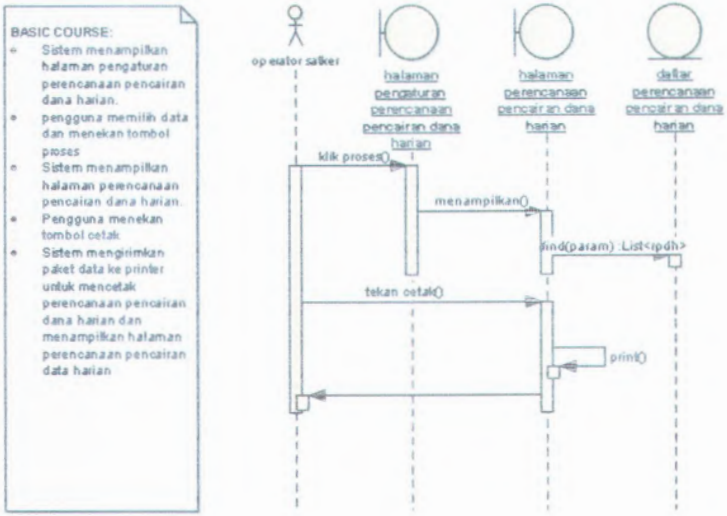


Diagram Sequence Mencetak Rencana Pencairan Dana Mingguan (SD15)

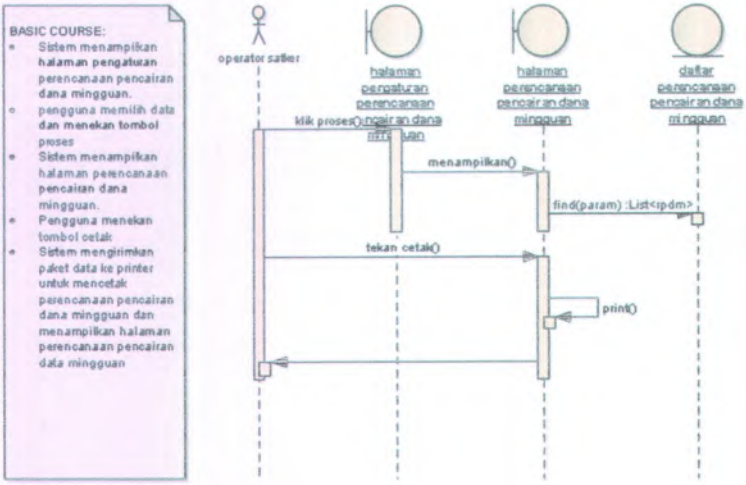


Diagram Sequence Mencetak Rekapitulasi Rencana Pencairan Dana Bulanan (SD16)

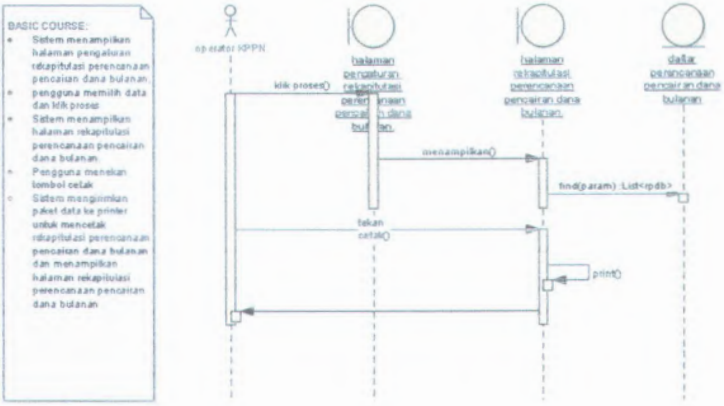


Diagram Sequence Mencetak Rekapitulasi Rencana Pencairan Dana Mingguan (SD17)

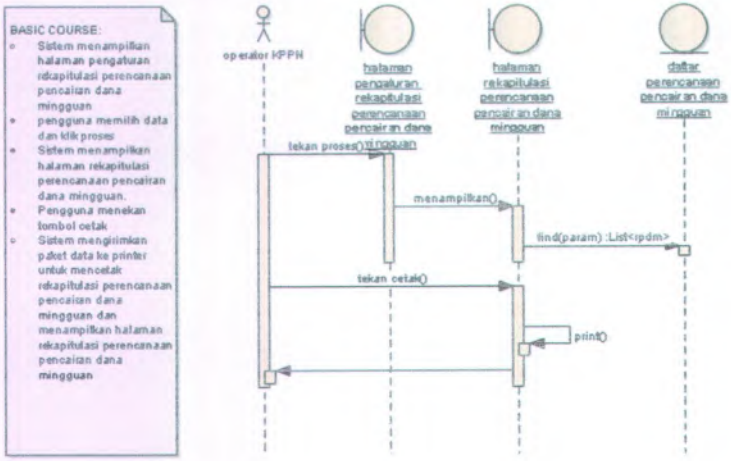


Diagram Sequence Mencetak Rekapitulasi Rencana Pencairan Dana Harian (SD18)

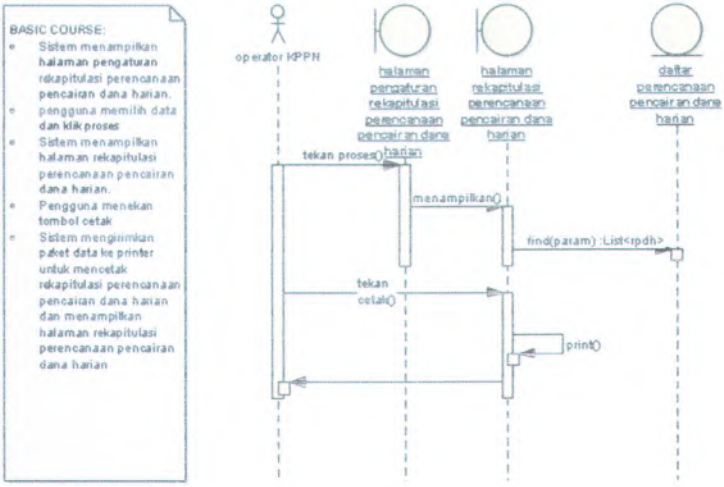


Diagram Sequence Menampilkan Akurasi Perencanaan (SD19)

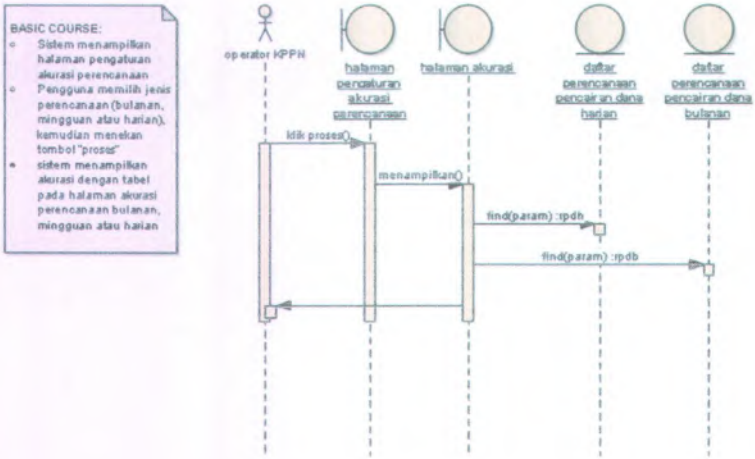


Diagram Sequence Merekam Pengguna (SD20)

Diagram Sequence Menghapus Pengguna (SD21)

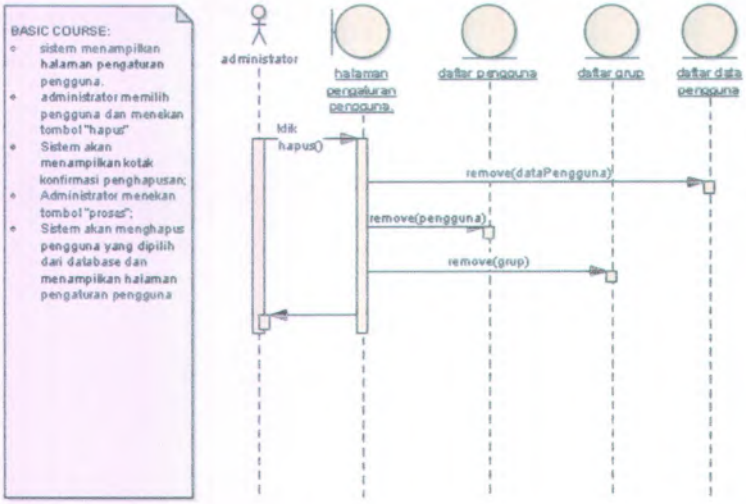


Diagram Sequence Merekam Satker (SD22)

BASIC COURSE:

- sistem menampilkan halaman pengaturan satker
- administrator menekan tombol "rekam"
- Sistem menampilkan halaman perekaman satker
- Administrator mengisi field kode dan nama departemen, unit organisasi dan satker dan menekan tombol "simpan"
- Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi perekaman
- Administrator menekan tombol "proses";
- Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan satker

ALTERNATE COURSE:
jika administrator belum mengisi satu atau lebih field yang disediakan, maka sistem akan menampilkan halaman perekaman satker dengan notifikasi field harus diisi seluruhnya.
jika terdapat kombinasi kode departemen, unit organisasi dan satker yang sama di database maka sistem akan menampilkan halaman perekaman satker dengan notifikasi data yang dimasukkan sudah ada di database

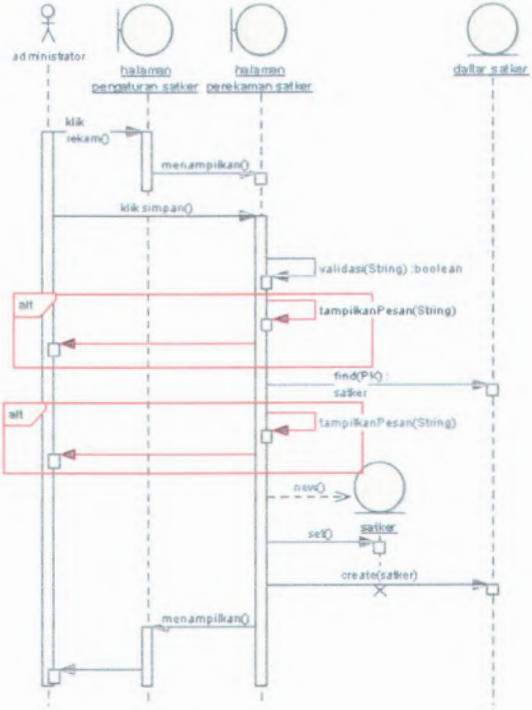


Diagram Sequence Mengubah Satker (SD23)

BASIC COURSE:

- sistem menampilkan halaman pengaturan satker
- administrator menekan tombol "ubah"
- Sistem mengambil data yang dipilih dari database dan menampilkan halaman perekaman satker
- Administrator mengubah isi field dan menekan tombol "simpan"
- Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi pengubahan
- Administrator menekan tombol "proses";
- Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan satker

ALTERNATE COURSE:
jika administrator belum mengisi satu atau lebih field yang disediakan, maka sistem akan menampilkan halaman perekaman satker dengan notifikasi field harus diisi seluruhnya.
jika terdapat kombinasi kode program, fungsi, sub fungsi, kegiatan, sub kegiatan dan jenis belanja yang sama di database maka sistem akan menampilkan halaman perekaman satker dengan notifikasi data yang dimasukkan sudah ada di database

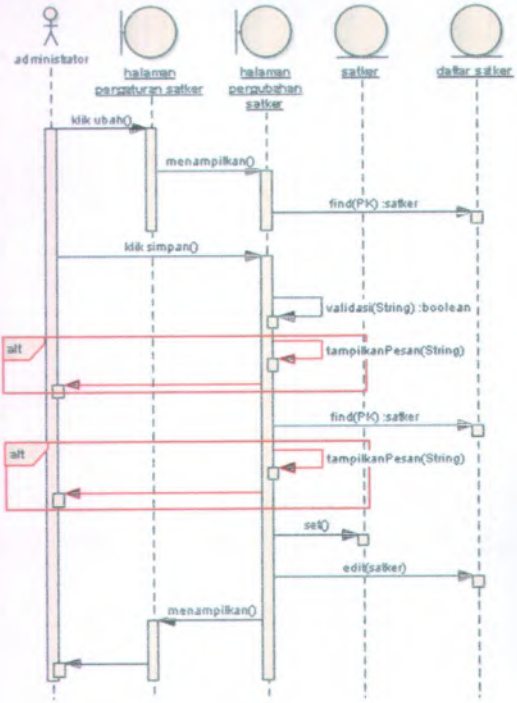


Diagram Sequence Menghapus Satker (SD24)

BASIC COURSE:

- sistem menampilkan halaman pengaturan satker.
- administrator memilih satker dan menekan tombol "hapus"
- Sistem akan menampilkan kotak konfirmasi penghapusan;
- Administrator menekan tombol "proses";
- Sistem akan menghapus referensi satker yang dipilih dari database dan menampilkan halaman pengaturan satker

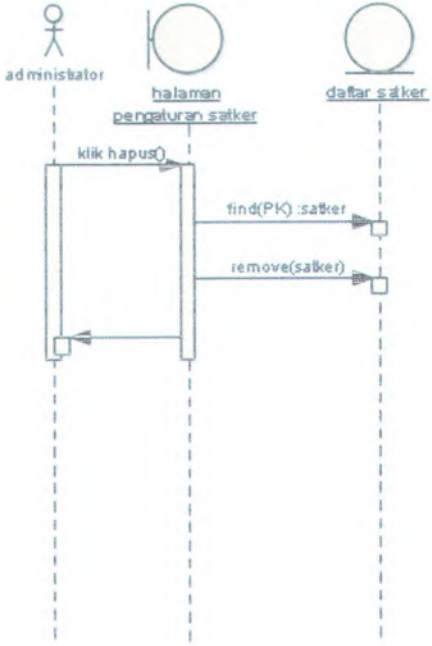


Diagram Sequence Merekam Anggaran (SD25)

BASIC COURSE:

- sistem menampilkan halaman pengaturan pengaluran anggaran
- administrator menekan tombol "rekam"
- Sistem menampilkan halaman perekaman anggaran
- Administrator mengisi field kode dan nama departemen, unit organisasi dan satker, jenis, nomor dan tanggal dokumen dan jumlah anggaran per jenis belanja dan menekan tombol "simpan"
- Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi perekaman
- Administrator menekan tombol "proses";
- Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan anggaran

ALTERNATE COURSE:

jika administrator tidak mengisi satu atau lebih field yang ada dan tidak mengisi tabel sama sekali maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi field dan tabel harus diisi. jika terdapat nomor dokumen yang sama di database maka sistem akan menampilkan halaman perekaman referensi satker dengan notifikasi data yang dimasukkan sudah ada di database.

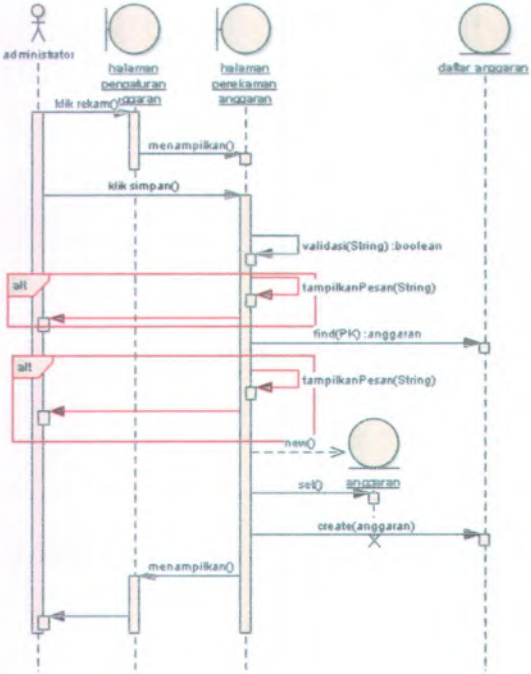


Diagram Sequence Mengubah Anggaran (SD26)

BASIC COURSE:
- sistem menampilkan halaman pengaturan anggaran
- administrator menekan tombol "ubah"
- Sistem mengambil data yang dipilih dari database dan menampilkan halaman perekaman anggaran
- Administrator mengubah isi field dan menekan tombol "simpan"
- Sistem memvalidasi field,membandingkan data masukan dengan database dan menampilkan kotak konfirmasi pengubahan
- Administrator menekan tombol "proses"
- Sistem menyimpan data masukan ke database dan menampilkan halaman pengaturan anggaran

ALTERNATE COURSE:
jika administrator belum mengisi satu atau lebih field yang diseditkan, maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi field harus diisi seluruhnya.
jika terdapat nomor surat yang sama di database maka sistem akan menampilkan halaman perekaman anggaran dengan notifikasi data yang dimasukkan sudah ada di database

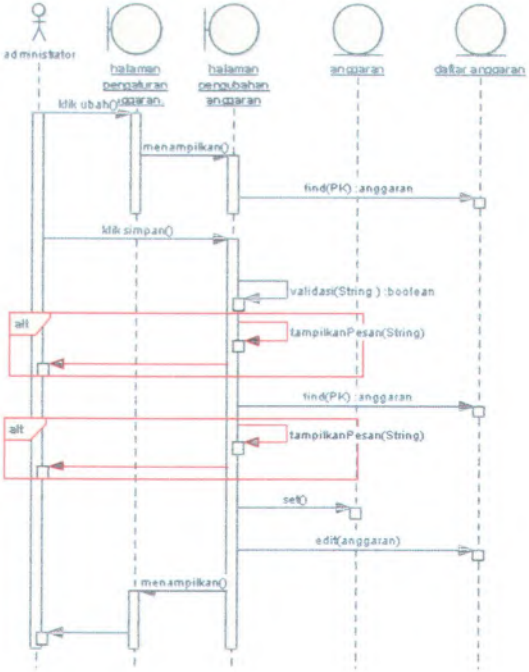


Diagram Sequence Menghapus Anggaran (SD27)

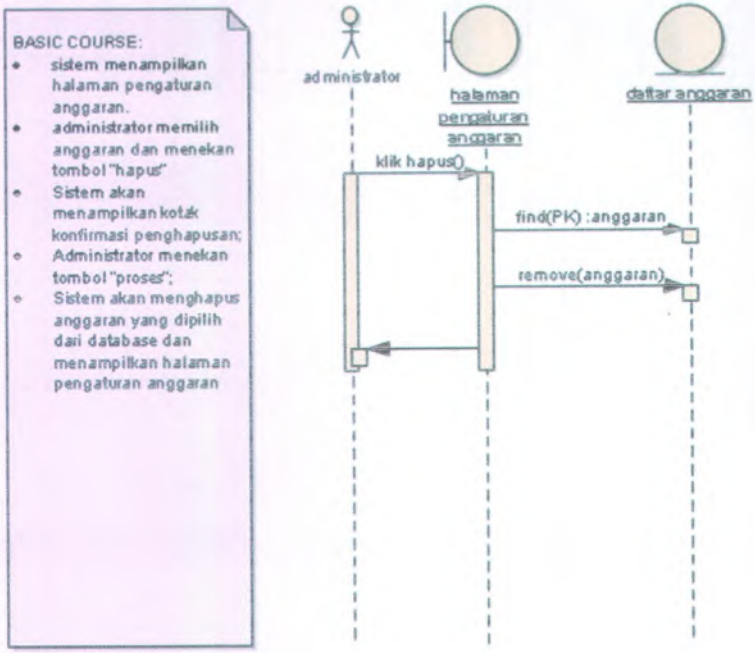
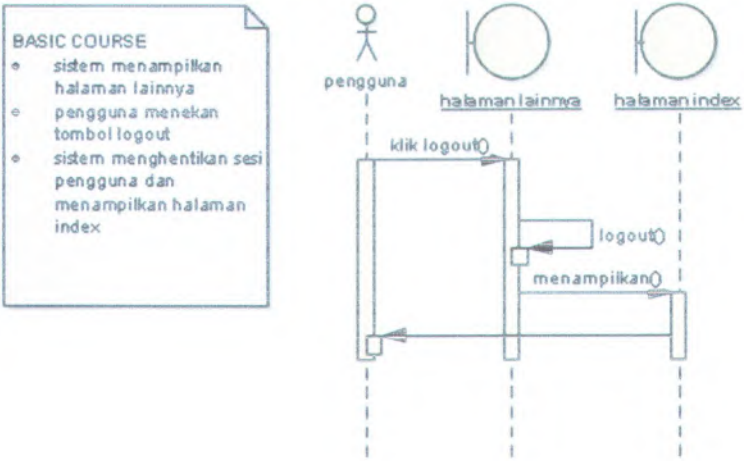


Diagram Sequence Melakukan Logout (SD28)



RIWAYAT PENULIS

Agastya Vitadhani lahir di Kabupaten Semarang, Jawa Tengah, pada tanggal 14 Agustus 1985. Mengenyam pendidikan di SDN Karangjati 2, SMPN 1 Ungaran, SMUN 1 Salatiga dan STAN Jakarta Setelah menjalani kedinasan selama empat tahun di Kementerian Keuangan, penulis melanjutkan pendidikan di ITS Surabaya.

Tema tugas akhir yang diangkat merupakan permasalahan di kedinasan yang dengan ini diharapkan dapat ditemukan solusinya.