

Pengembangan Sistem Informasi Akuntansi dan Pengelolaan Keuangan Modul Penganggaran Berbasis Web Menggunakan Teknologi Java Dan Postgresql

Hari Rakhmanto Z. dan Febriliyan Samopa

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111

E-mail: iyan@is.its.ac.id

Abstrak— Banyaknya fungsi yang dijalankan di satuan kerja pemerintah telah didukung oleh teknologi informasi yaitu berupa aplikasi-aplikasi yang banyak pula. Namun banyaknya aplikasi tersebut menyebabkan munculnya permasalahan baru yaitu sistem database yang masih terpisah. Saat ini akan dikembangkan suatu aplikasi yang bernama Sistem Informasi Akuntansi dan Pengelolaan Keuangan yang akan mengintegrasikan semua aplikasi operasional di satuan kerja. Aplikasi ini berbasis web dan menggunakan teknologi Java dan PostgreSQL. Salah satu fungsi yang wajib ikut diintegrasikan aplikasinya adalah fungsi penganggaran dengan aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran. Aplikasi ini sangat penting karena merupakan awal dari proses pengelolaan keuangan. Jika aplikasi ini tidak ikut diintegrasikan maka akan berpotensi tidak sinkronnya data Daftar Isian Pelaksanaan Anggaran dari aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran dengan aplikasi Surat Perintah Membayar. Terpisahnya database dari aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran dengan aplikasi Surat Perintah Membayar juga dapat menimbulkan potensi tindakan kriminal manipulasi data Daftar Isian Pelaksanaan Anggaran. Dengan diintegrasikannya aplikasi Rencana Kerja Anggaran-Kementerian Negara/Lembaga Daftar Isian Pelaksanaan Anggaran ke dalam aplikasi Sistem Informasi Akuntansi dan Pengelolaan Keuangan maka dapat meningkatkan kinerja satuan kerja sekaligus meningkatkan kualitas laporan keuangannya.

Kata Kunci— Aplikasi, Penganggaran, Integrasi, , Satuan Kerja.

I. PENDAHULUAN

Satuan kerja (satker) adalah instansi pemerintah yang merupakan bagian dari struktur kementerian di pemerintahan. Setiap satu satker mempunyai kepala satker (kepala dinas, kepala kantor dsb.), bendahara satker, pejabat pembuat komitmen, pejabat penanda tangan Surat Perintah Membayar (SPM), pembuat daftar gaji (PDG), dan pegawai yang bekerja di instansi tersebut [1]. Beberapa contoh dari satker adalah Dinas Perhubungan Kota Surabaya, Kantor Kejaksaan Tinggi Surabaya, Kantor Departemen Agama Kota Surabaya dan lain-lain.

Satker mendapatkan dana dari Anggaran Pendapatan dan Belanja Negara (APBN) untuk melaksanakan kegiatannya.

Satker sebagai organisasi pemerintah menjalankan fungsi akuntansi dan pengelolaan keuangan atas kegiatan yang dilakukannya [2]. Dalam menjalankan fungsinya, satker telah menerapkan teknologi informasi untuk meningkatkan kinerja dan kualitas laporan keuangannya. Fungsi tersebut antara lain pembukuan pengeluaran dana dari APBN yang ditunjang dengan aplikasi SPM, pembukuan bendahara memakai aplikasi Sistem Informasi Keuangan (SISKA), Penganggaran satker yang dibantu oleh aplikasi Rencana Kerja Anggaran Kementerian/Lembaga dan Daftar Isian Pelaksanaan Anggaran (RKA-K/L DIPA), akuntansi kantor satker yang menggunakan Sistem Akuntansi Kuasa Pengguna Anggaran (SAKPA) dan penatausahaan barang-barang inventarisasi satker dengan memakai aplikasi Persediaan dan Sistem Informasi Manajemen Akuntansi Barang Milik Negara (SIMAK BMN). Semua aplikasi tersebut dibangun secara terpisah-pisah berdasarkan urgensi kebutuhan satker.

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

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

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

Untuk mengatasi masalah di atas, saat ini akan dibangun satu aplikasi yang bernama Sistem Aplikasi Akuntansi dan

Pengelolaan Keuangan (SIAPKAN) yang akan mengintegrasikan semua fungsi satker. Aplikasi ini akan menggunakan sistem *single database* sehingga data yang disajikan akan selalu relevan jika terjadi perubahan data pada satu sisi fungsi. Aplikasi SIAPKAN terdiri dari 7 modul:

1. Modul Penganggaran
2. Modul Pembayaran
3. Modul Bendahara
4. Modul Persediaan
5. Modul Aset Tetap
6. Modul Administrasi
7. Modul Pelaporan

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

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

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

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

Servlets adalah suatu kode aplikasi berbasis Java yang berjalan di web server. Servlets merupakan aplikasi yang berfungsi menjembatani antara client dengan server, dalam

artian servlets meneruskan permintaan (request) dari web browser atau dari HTTP client menuju aplikasi atau database yang ada di server [3].

Servlets mempunyai banyak kelebihan dibandingkan dengan teknologi aplikasi web sebelumnya CGI. Keunggulan ini antara lain [3]:

1. Efisien
2. Tepat dan lengkap
3. Handal
4. Portable
5. Aman
6. Murah

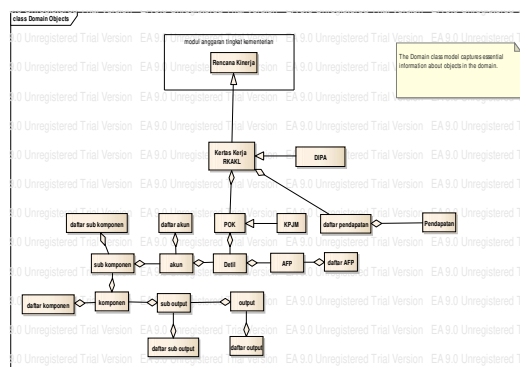
Teknologi JavaServer Pages (JSP) adalah suatu halaman yang mampu menggabungkan halaman web yang regular, statik dan dinamis yang dihasilkan dari Servlets. Dalam sisi penggunaan JavaBeans, penulisan dalam format JSP akan lebih memudahkan programmer. Karena JSP mampu menggabungkan kode java dan konten seperti teks statik, kode HTML, DHTML, XHTML, XML dan kode lainnya dalam bentuk skrip atau tag [4].

Sistem database yang akan digunakan dalam aplikasi ini adalah sistem database relasional PostgreSQL. PostgreSQL adalah sistem basis data relasional yang dikembangkan dari sistem basis data Postgres. PostgreSQL memiliki lisensi BSD klasik yaitu bebas diambil, diubah, dipakai dan didistribusikan dengan tanpa biaya. PostgreSQL dikembangkan pertama kali oleh Andrew Yu dan Jolly Chen [5].

II. DESAIN APLIKASI

A. Domain Model

Sesuai dengan proses pada ICONIX, *domain model* digunakan untuk menggambarkan obyek-obyek utama yang akan digunakan serta menginisialisasi cakupan dari sistem dan hal-hal yang terlibat di dalamnya. *Domain model* dapat berubah seiring dengan pengembangan desain dan aplikasi, sehingga obyek-obyek yang digambarkan pada *domain model* akan semakin lengkap dan akurat sesuai dengan alur sistem. Pada aplikasi SIAPKAN yang dikembangkan pada Tugas Akhir ini, penulis menginisialisasi *domain model* yang diperlihatkan Gambar 4.2 sebagai berikut.



Gambar 1 Domain Model

Berdasarkan *domain model* pada Gambar 4.2, terdapat beberapa *class* yang diinisialisasi, yaitu :

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

B. Use Case

Berikut ini merupakan daftar *use case* yang ada pada Modul Penganggaran :

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

11. Use Case Cetak AFP
12. Use Case Cetak Kertas Kerja
13. Use Case Cetak Laporan

C. Testing

Untuk mengetahui keandalan aplikasi, diperlukan proses percobaan sebelum aplikasi diimplementasikan. Tes yang akan digunakan *scenario test* yang menguji setiap kemungkinan input dari user di setiap use case yang ada. Tes ini berisi antara lain :

1. Test case 1 sampai dengan 7 berupa rekam dan ubah output.sampai dengan AFP
2. Test case 8 sampai dengan 13 berupa hapus dari output sampai dengan detil
3. Test case 14 approval AFP
4. Test case 15 berupa tayang DIPA

Untuk tes mengenai rekam dan ubah, ada beberapa skenario yang diterapkan, antara lain :

1. Tes untuk input field yang diisi kosong
2. Input angka diisi bukan angka
3. Input karakter khusus, misal “”, “?”, “()” dan lain-lain
4. Data yang sama yang diinputkan lebih dari satu kali
5. Mematikan fitur *javascript* kemudian melakukan input data seperti yang dilakukan di atas.
6. Mematikan *javascript* untuk melakukan proses hapus
7. Mematikan *javascript* untuk melakukan proses tayang dan monitoring data AFP,DIPA dan rincian belanja.

Setelah melakukan uji coba, didapatkan hasil sebagai berikut :

Table 1 hasil uji coba

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

