

PENGEMBANGAN APLIKASI LAPORAN SURAT PERTANGGUNGJAWABAN ANGGARAN PADA DINAS PENDIDIKAN KAB. GARUT MENGGUNAKAN METODOLOGI BERORIENTASI OBJEK RIPPLE

Angga Riswana¹, Eri Satria², Kiki Aisyah³

Jurnal Algoritma
Sekolah Tinggi Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia
Email : jurnal@sttgarut.ac.id

¹0906022@sttgarut.ac.id

²erisatria@sttgarut.ac.id

Abstrak – Dinas Pendidikan Kab. Garut merupakan salah satu dari 17 dinas daerah dan 33 SKPD (Satuan Kerja Pemerintah Daerah) yang mempunyai tugas untuk melaksanakan urusan pemerintahan daerah berdasarkan asas otonomi dan tugas pembantuan dalam bidang pendidikan. Dengan visi Penyelenggaraan layanan pendidikan yang baik, merata, partisipatif dan berkualitas untuk mewujudkan insan yang cerdas, kompetitif, mandiri dan berkearifan lokal berlandaskan pada ketakwaan kepada Tuhan YME dan salah satu misinya yaitu membina dan mengendalikan layanan pendidikan yang diselenggarakan secara transparan, responsif, partisipatif, akuntabel dan berkelanjutan. Salah satu bentuk perwujudan dari misi tersebut adalah pembuatan Surat Pertanggungjawaban (SPJ). SPJ yang selama ini dibuat memiliki banyak perbedaan antara satu bagian dengan bagian lainnya sehingga tidak ada keseragaman format SPJ. Selain itu, dalam pembuatan SPJ yang ada selama ini banyak melakukan proses *input*, padahal banyak data yang sama telah di-*input* di aplikasi RKA. Untuk itu, penelitian ini diarahkan pada pembuatan aplikasi yang bisa memanfaatkan sumber daya data dari aplikasi yang RKA telah ada. Penelitian ini akan menggunakan metode berorientasi objek dalam pengembangan sistemnya yaitu metodologi *Ripple*. *Ripple* adalah metodologi yang menggabungkan tahapan-tahapan dalam SDLC dengan notasi UML dari OOP sebagai diagram pemodelan sistem. Metode pengujian yang digunakan adalah metode pengujian Blackbox.

Kata Kunci – Aplikasi, Laporan Surat Pertanggungjawaban, Dinas Pendidikan, Ripple.

I. PENDAHULUAN

Dinas Pendidikan Kab. Garut merupakan salah satu dari 17 dinas daerah dan 33 SKPD (Satuan Kerja Pemerintah Daerah) yang mempunyai tugas untuk melaksanakan urusan pemerintahan daerah berdasarkan asas otonomi dan tugas pembantuan dalam bidang pendidikan. Dinas Pendidikan Kab. Garut ini terdiri dari 1 Kepala Dinas, 1 Sekretariat dengan 3 Subbagian 6 Bidang, 18 Seksi, 42 UPTD dan Kelompok Jabatan. Dalam melaksanakan program kerjanya, Dinas Pendidikan Kab. Garut berpegang pada visi: Penyelenggaraan layanan pendidikan yang baik, merata, partisipatif dan berkualitas untuk mewujudkan insan yang cerdas, kompetitif, mandiri dan berkearifan lokal berlandaskan pada ketakwaan kepada Tuhan YME. Dan untuk menciptakan visi tersebut, salah satu misi yang berkaitan dengan penelitian ini adalah membina dan mengendalikan layanan pendidikan yang diselenggarakan secara transparan, responsif, partisipatif, akuntabel dan berkelanjutan.

Salah satu bentuk perwujudan dari misi tersebut adalah pembuatan Surat Pertanggungjawaban (SPJ) dari penggunaan anggaran kegiatan yang dilakukan. Anggaran kegiatan yang telah dibuat dalam aplikasi Rencana Kerja dan Anggaran (RKA) yang dimiliki Dinas

Pendidikan Kab. Garut tersebut menjadi acuan dalam pelaksanaan kegiatan serta pembuatan pelaporan dalam hal ini SPJ. Namun di Dinas Pendidikan Kab. Garut ini belum tersedia aplikasi pembuat SPJ yang memanfaatkan sumber daya yang telah tersedia tersebut. Selain itu, dalam pembuatan SPJ yang selama ini dilakukan banyak melakukan proses *input*, padahal banyak data yang sama telah di-*input* di aplikasi RKA.

Selain mengacu pada aplikasi RKA tadi penelitian ini juga membandingkan pada penelitian-penelitian lain yang telah dilakukan. Pertama adalah penelitian dengan judul Analisis dan Perancangan Aplikasi Laporan Keuangan Daerah Pada Dinas Pendidikan, Pemuda dan Olahraga Kabupaten Maluku Tenggara [8]. Dimana penelitian ini berfokus pada pembuatan laporan keuangan secara menyeluruh dengan menggunakan teknologi Java untuk *desktop* serta MySQL sebagai alat pengolah basis data dan metodologi yang digunakan dalam pengembangan sistem adalah metodologi SDLC (*Software Development Life Cycle*). Penelitian kedua adalah penelitian dengan judul Pengembangan Sistem Informasi Akuntansi dan Pengelolaan Keuangan Modul Bendahara Menggunakan Teknologi Java dan PostgreSQL [3]. Dimana penelitian membuat modul bendahara dengan penggunaan PostgreSQL berbasis web serta memakai metodologi berorientasi objek dalam pengembangan sistemnya. Dan yang terakhir adalah penelitian yang dilakukan di Kelurahan Kebayoran Lama Selatan, Jakarta Selatan mengenai Aplikasi Pengelolaan Dana dan Belanja Pemerintah Daerah Berbasis Web [2]. Dimana PHP digunakan sebagai bahasa pemrograman dengan memakai MySQL sebagai alat pengolah basis data serta menggunakan metodologi SDLC untuk pengembangan sistemnya.

Melihat kepada penelitian-penelitian yang telah dilakukan tersebut maka penelitian ini akan menggunakan metode berorientasi objek dalam pengembangan sistemnya yaitu metodologi *Ripple* yang dikembangkan oleh Mike O'Docherty. *Ripple* adalah metodologi yang menggabungkan tahapan-tahapan dalam SDLC dengan notasi UML dari OOP sebagai diagram pemodelan sistem. Dengan melihat kepada uraian-uraian di atas, maka tugas akhir ini diarahkan kepada judul **“PENGEMBANGAN APLIKASI LAPORAN SURAT PERTANGGUNGJAWABAN ANGGARAN PADA DINAS PENDIDIKAN KAB. GARUT MENGGUNAKAN METODOLOGI BERORIENTASI OBJEK *RIPPLE*”**.

II. TINJAUAN PUSTAKA

A. Aplikasi

Ada beberapa definisi aplikasi berbeda menurut para ahli. Menurut Dhanta [1], aplikasi (*application*) merupakan software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, *Microsoft Excel*. Menurut Jogiyanto [4], aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*. Sedangkan menurut Menurut Kamus Besar Bahasa Indonesia [5] aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

B. Laporan Surat Pertanggungjawaban

Surat pertanggungjawaban (SPJ) adalah suatu laporan yang sistematis tentang rincian pemakaian anggaran dalam suatu kegiatan. Laporan ini disusun dengan tujuan memberikan laporan tentang pelaksanaan kegiatan dari suatu unit organisasi kepada unit organisasi yang lebih tinggi atau sederajat. Surat pertanggungjawaban berguna sebagai bahan evaluasi terhadap seluruh proses pelaksanaan kegiatan dan hasil-hasil yang dapat dicapai dari kegiatan tersebut, yang selanjutnya dapat dijadikan sebagai bahan pertimbangan bagi perbaikan-perbaikan dan peningkatan kualitas pelaksanaan kegiatan pada masa yang akan datang. SPJ dibuat sebelum, pada saat dan setelah pelaksanaan suatu kegiatan. SPJ juga merupakan suatu bentuk upaya transparansi keuangan suatu

badan atau lembaga terhadap masyarakat atau pihak yang berkepentingan. SPJ secara otomatis dibuat di bulan yang sama dengan pelaksanaan kegiatan. Dan jika melampaui bulan berjalan, maka SPJ harus dibuat kembali untuk bulan pada saat itu.

C. Metodologi Ripple

Metodologi *Ripple* adalah versi berorientasi objek semua fase klasik pengembangan perangkat lunak. Karena orientasi objek ketika diakses, pengembang dapat terlibat dalam semua fase, pelanggan dapat terlibat dalam tahap awal, yang membantu pengembang untuk melakukan pekerjaan mereka, dan manajer pun ikut terlibat, sehingga komunikasi dapat ditingkatkan. Tahapan yang akan dilakukan hanya mencakup tahapan *requirements, analysis, design, implementation, dan testing*. Tahapan ini memiliki kesamaan tahapan seperti yang ada pada metodologi pengembangan sistem klasik. Namun artefak yang dihasilkan berbeda dari metodologi klasik. Artefak yang dihasilkan merupakan diagram-diagram dari UML

Tahap-tahap dalam metodologi *Ripple* adalah sebagai berikut [7]:

Requirements: Mengumpulkan persyaratan/ *requirement* adalah tentang menemukan apa yang akan dicapai dengan perangkat lunak baru kita dan memiliki dua aspek yaitu persyaratan bisnis dan sistem. Pemodelan persyaratan sistem (atau spesifikasi fungsional) berarti memutuskan kemampuan apa yang akan dimiliki perangkat lunak baru dan menuliskan kemampuan tersebut. Kita harus jelas tentang apa yang akan perangkat lunak lakukan dan apa yang tidak akan dilakukan, sehingga pengembangan tidak keluar dari daerah-daerah yang tidak relevan dan kita tahu baik apakah aplikasi sudah selesai dan sukses

Analysis: Analisis berarti memahami apa yang sedang kita hadapi. Sebelum dapat merancang solusi, harus dijelaskan tentang entitas apa yang relevan, sifat mereka dan hubungan di antara mereka. Kita juga perlu untuk memverifikasi pemahaman kita. Hal ini dapat melibatkan pelanggan dan pengguna akhir, karena mereka cenderung menjadi masalah subjek ahli

Design: Dalam tahap ini akan dibuat keputusan, berdasarkan pengalaman, estimasi dan intuisi, tentang *software* yang akan kita tulis dan bagaimana kita akan menyebarkannya. Desain sistem memecah sistem ke dalam subsistem logis (proses) dan subsistem fisik (komputer dan jaringan), memutuskan bagaimana mesin akan berkomunikasi, memilih teknologi yang tepat untuk pekerjaan, dan sebagainya

Implementation: Tahap ini adalah tahap dimana kita melakukan pekerjaan kodifikasi, menulis potongan kode yang bekerja sama untuk membentuk subsistem, yang pada gilirannya berkolaborasi untuk membentuk seluruh sistem.

Testing: Ketika perangkat lunak telah selesai dibangun, maka perangkat lunak tersebut harus diuji terhadap persyaratan sistem untuk melihat apakah perangkat lunak sesuai dengan tujuan awal atau tidak. Pada dasarnya, pengujian akan dilakukan dengan 2 cara, yaitu dengan teknik *Black-Box* untuk menguji *usage ability* dan teknik pengujian langsung oleh pengguna. Pengujian *Black-Box* muncul dari filosofi "kita tidak peduli bagaimana kode mencapai ujungnya, asalkan mereka mencapai tujuan". Hal ini cocok dengan gagasan bahwa persyaratan sistem yang ditulis sebelum perangkat lunak diproduksi adalah aspek yang paling penting dari pengembangan perangkat lunak

D. Bahasa Pemodelan dan Pengembangan Sistem

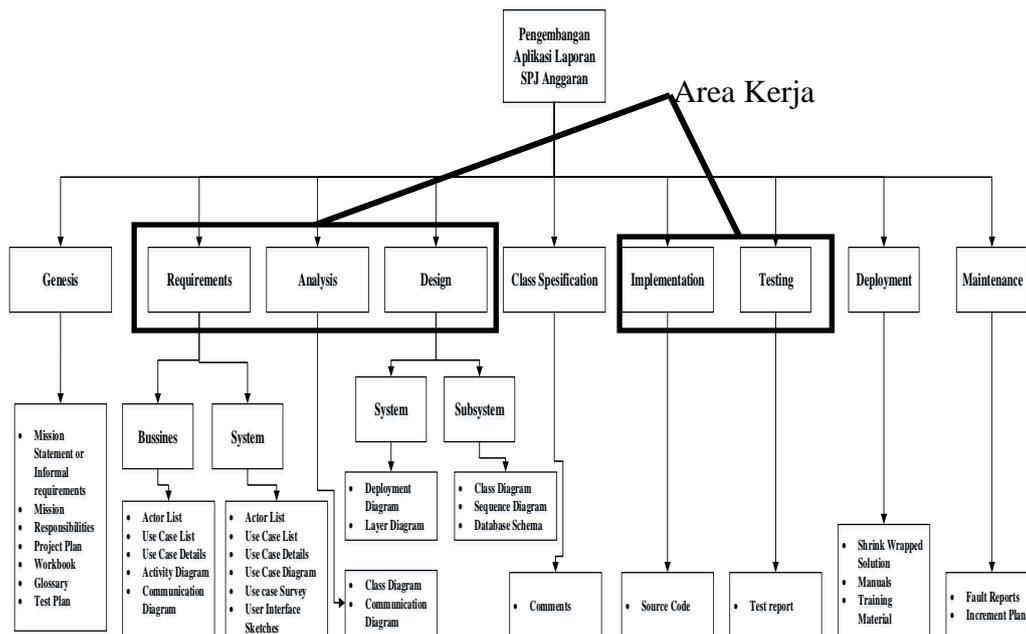
Bahasa pemodelan yang umum digunakan dalam orientasi objek adalah UML (*Unified Modeling Language*). UML adalah notasi yang akan digunakan untuk dokumentasi tingkat tinggi, beberapa juga menyatakan bahwa UML menjadi bahasa pemrograman bergambar, menghasilkan kode atau mensintesis dari kode yang ada. UML memiliki 13 jenis diagram. Spesifikasi UML tidak menyebutkan di mana diagram ini harus digunakan dalam suatu metodologi tertentu, tetapi bebas digunakan dimana saja yang dianggap tepat pada setiap tahap oleh pengembang sistem [7].

Dalam pengembangan sistem yang dilakukan, bahasa pemrograman yang akan dipakai adalah Java. Java merupakan bahasa pemrograman berorientasi obyek yang menggunakan abstraksi, enkapsulasi, inheritance, dan polimorfisme untuk memberikan fleksibilitas yang besar, modularitas, dan usabilitas dalam pengembangan *software* [7].

III. KERANGKA KERJA KONSEPTUAL

Pekerjaan-pekerjaan yang akan dilakukan dalam penelitian ini didasarkan kepada tahap-tahap yang ada pada metodologi *Ripple*. Secara singkat langkah-langkah kerja dalam metodologi *Ripple* ini akan digambarkan pada gambar 1.1 di bawah. Gambar tersebut memuat tahap-tahap yang harus dilakukan beserta artifak-artifak apa saja yang harus dihasilkan dalam setiap tahapnya.

Tahap-tahap yang ada dalam metodologi *Ripple* tidak semua akan digunakan dalam penelitian ini karena ada beberapa tahap yang memang tidak diperlukan. Seperti *Genesis*, *Deployment* dan *Maintenance* tidak akan digunakan dalam penelitian ini karena tahap-tahap ini biasanya digunakan dalam proyek besar yang memerlukan tim dalam pengerjaannya. Jadi hanya tahap *Requirements*, *Analysis*, *Design*, *Class Spesification*, *Implementation* dan *Testing* yang akan digunakan. Tahap-tahap yang ini merupakan tahap seperti yang terdapat dalam metodologi klasik



Gambar 1 Struktur Rincian Kerja

Tahap *requirements* terbagi menjadi dua, yaitu *bussines* dan *system requirement*. *Bussines requirements* melakukan kegiatan seperti membuat daftar aktor yang terlibat dalam proses bisnis pembuatan SPJ, membuat daftar *use case* dari proses bisnis pembuatan SPJ, membuat *communication diagram* dari salah satu *use case*, membuat *activity diagram* dari salah satu *use case* dan membuat rincian dari *use case* yang telah dibuatkan daftar sebelumnya. *System requirement* membuat daftar aktor yang terlibat langsung dengan sistem pembuatan SPJ, membuat daftar *use case* dari interaksi dengan sistem pembuatan SPJ, membuat *use case diagram* berdasarkan pada *use case list* yang telah dibuat, membuat *use case survey* sistem pembuatan SPJ, membuat rincian dari *use case* yang telah dibuatkan daftar sebelumnya dan membuat ilustrasi interaksi sistem dalam bentuk sebuah rancangan antar muka sistem. Semua kegiatan menggunakan data hasil wawancara dengan narasumber, data hasil observasi langsung ke lapangan, dokumen-dokumen yang dilaporkan dalam SPJ dan data keluaran dari proses sebelumnya sebagai sumber masukan.

Tahap *analysis* diantaranya membuat *class diagram* dari objek-objek atau entitas yang terlibat dalam sistem pembuatan SPJ disertai atribut dari setiap kelas dan *operation list* serta membuat *use case realization* yang digambarkan oleh *communication diagram*. Data masukan kegiatan tersebut berasal dari data hasil wawancara dengan narasumber, data hasil observasi langsung ke lapangan dan dokumen-dokumen yang dilaporkan dalam SPJ.

Tahap *design* terbagi menjadi dua kegiatan, yaitu membuat *system* dan *subsystem design*. *System design* diantaranya membuat *layer diagram* yang menggambarkan lapisan sistem yang bekerja serta membuat *deployment diagram* yang menampilkan keterkaitan antara mesin, proses dan artefak bekerja. Sedangkan kegiatan *subsystem design* membuat *class diagram* yang menunjukkan keterkaitan antar kelas dalam sistem, membuat skema basis data berdasarkan *class diagram* yang dibuat sebelumnya, dan membuat *sequence diagram* untuk memperlihatkan urutan proses dari setiap *use case* yang merupakan turunan dari *communication diagram* yang dibuat pada proses analisis sebelumnya. Kegiatan-kegiatan tersebut menggunakan data hasil wawancara dengan narasumber, data hasil observasi langsung ke lapangan dan data keluaran dari proses sebelumnya sebagai sumber masukan.

Tahap *implementation* yaitu tahap untuk menuliskan kode program berdasarkan pada model-model yang dihasilkan pada proses-proses yang telah dibuat sebelumnya untuk menghasilkan kode program serta program jadi.

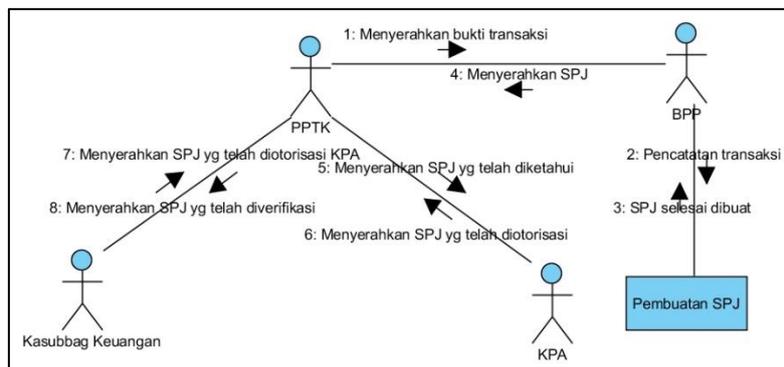
Tahap *testing* yaitu menguji program yang telah dihasilkan pada proses sebelumnya apakah bisa digunakan atau tidak (*usage ability*) melalui teknik pengujian, yaitu teknik Black-Box untuk kemudian dituliskan pada *testing report* sebagai hasil keluaran.

IV. HASIL DAN PEMBAHASAN

Tahap pertama yang dilakukan adalah *requirement*, dalam tahap ini ada dua jenis *requirement* yang dibuat, yaitu *requirement* untuk proses bisnis dan *requirement* untuk sistem. Ada beberapa langkah dalam menentukan *requirement* untuk proses bisnis di antaranya *actor list*, *communication diagram* dan *activity diagram* sedangkan untuk sistem yaitu *actor list*, *use* dan *use case diagram* serta *user interface sketches*.

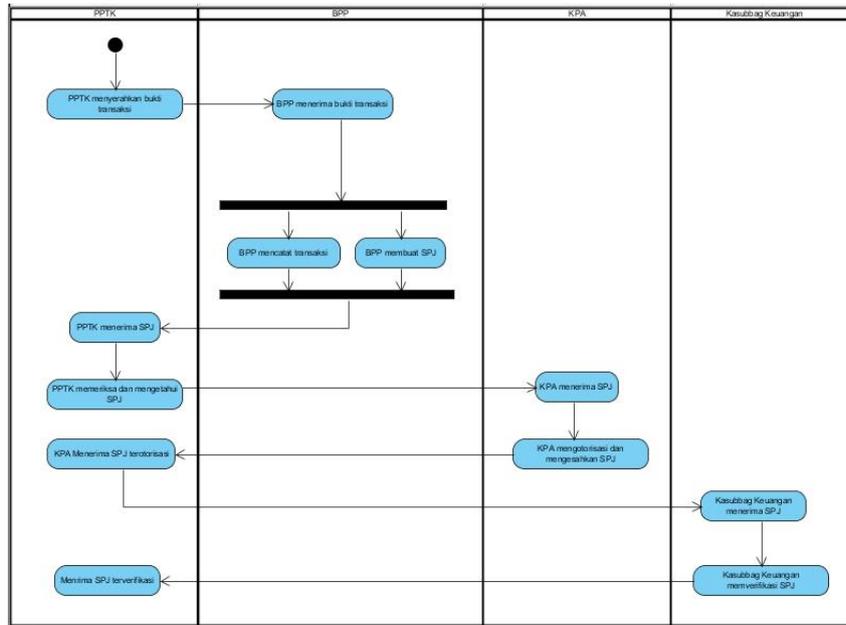
Bussiness Actor List yang terlibat adalah Kuasa Pengguna Anggaran (KPA), Pejabat Pelaksana Teknis Kegiatan (PPTK), Bendahara Pengeluaran Pembantu (BPP), dan Kasubbag Keuangan. Sedangkan untuk sistem aktor yang terlibat hanya BPP dan PPTK.

Untuk menggambarkan kegiatan yang dilakukan oleh masing-masing aktor dalam kegiatan bisnis maupun sistem akan digambarkan dalam sebuah *communication* dan *activity diagram* untuk proses bisnis dan sebuah *use case diagram* untuk sistem. Berikut adalah gambaran diagram-diagram tersebut:



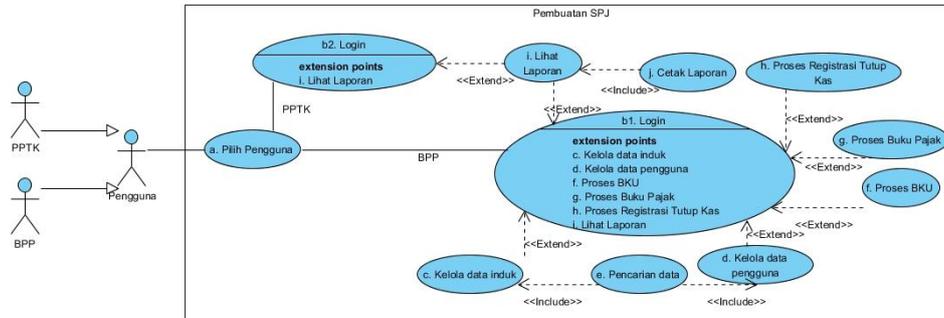
Gambar 2 *Communication Diagram* untuk Proses Bisnis SPJ

Berikut adalah *activity diagram* yang menggambarkan bagaimana proses bisnis pembuatan SPJ.



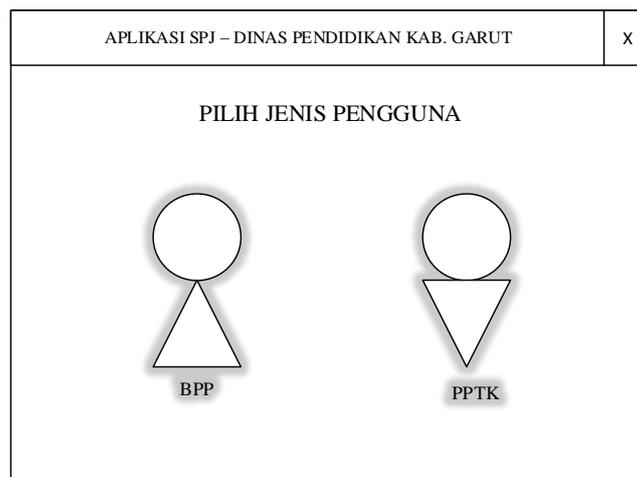
Gambar 3 Activity Diagram untuk Proses Bisnis SPJ

Di bawah ini adalah use case diagram dari sistem pembuatan SPJ.



Gambar IV Use case Diagram Pembuatan SPJ

Untuk menerapkan proses-proses tadi maka diperlukan sebuah antarmuka program yang akan menjalankan dan mengeksekusi setiap perintah yang diberikan, berikut adalah sketsa dari antarmuka tersebut:



Gambar 5 Form Pilih Jenis Pengguna

APLIKASI SPJ – DINAS PENDIDIKAN KAB. GARUT		X
SILAHKAN LAKUKAN LOGIN		
USERNAME :	<input type="text"/>	
PASSWORD :	<input type="text"/>	
	<input type="button" value="MASUK"/>	<input type="button" value="BATAL"/>

Gambar 6 Form Login

APLIKASI SPJ – DINAS PENDIDIKAN KAB. GARUT				X
KELOLA DATA	TRANSAKSI	LAPORAN		
DATA PENGGUNA	BUKU KAS UMUM	BUKU KAS UMUM		
DATA INDUK	BUKU PAJAK	BUKU PAJAK		
KODE KEGIATAN	REGISTER TUTUP KAS	RINCIAN PER AKUN		
KODE REKENING		KEADAAN KAS		
		PENGIRIMAN SPJ		
		REGISTER TUTUP KAS		

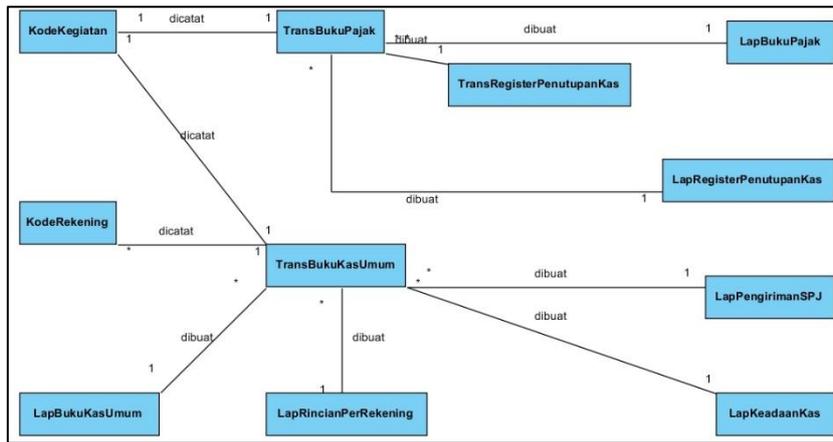
Gambar 7 Tampilan Menu Utama untuk BPP

APLIKASI SPJ – DINAS PENDIDIKAN KAB. GARUT				X
KELOLA DATA	TRANSAKSI	LAPORAN		
			BUKU KAS UMUM	
			BUKU PAJAK	
			RINCIAN PER AKUN	
			KEADAAN KAS	
			PENGIRIMAN SPJ	
			REGISTER TUTUP KAS	

Gambar 8 Tampilan Menu Utama untuk PPTK

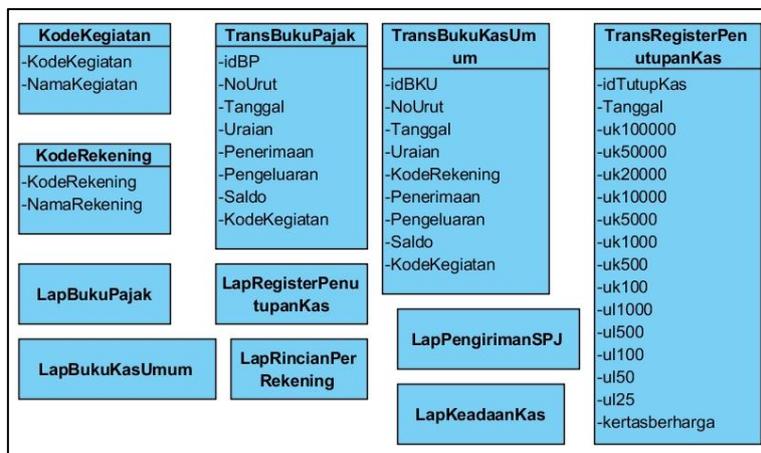
APLIKASI SPJ – DINAS PENDIDIKAN KAB. GARUT				X
		KOP DINAS		
DATA				
..... :	<input type="text"/>	Cari :	<input type="text"/>	
..... :	<input type="text"/>			
..... :	<input type="text"/>			
<input type="button" value="TAMBAH"/> <input type="button" value="UBAH"/> <input type="button" value="HAPUS"/>				
<input type="button" value="KOSONGKAN"/> <input type="button" value="KEMBALU"/>				

Gambar 9 Sketsa Form Kelola Data

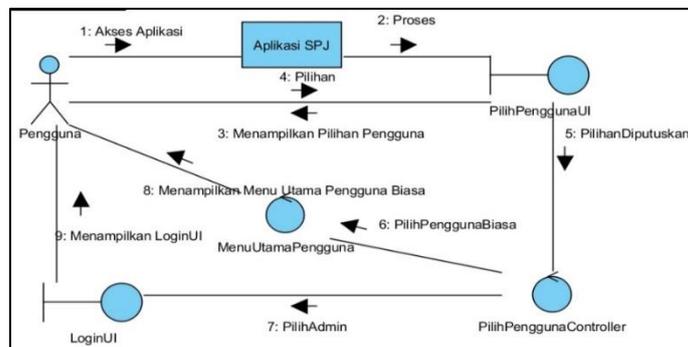


Gambar 13 Analysis Class Diagram

Atribut dari setiap class akan diperlihatkan pada gambar di bawah ini:

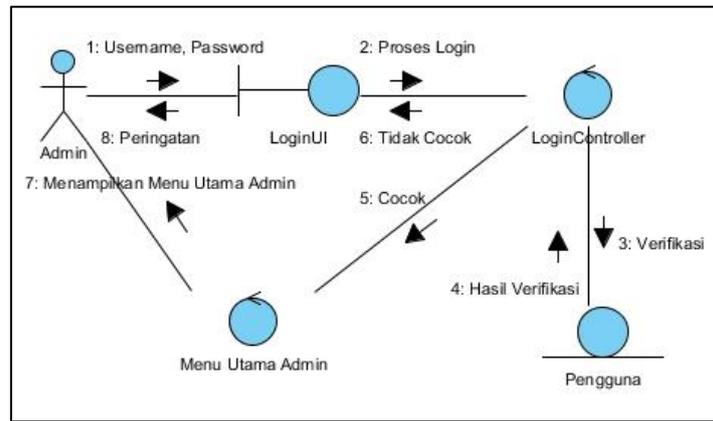


Gambar 14 Analysis Attribute

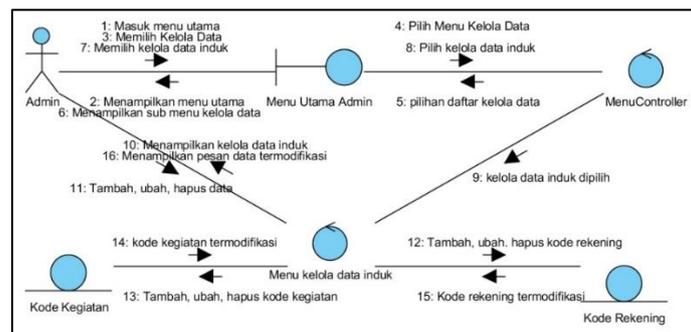


Gambar 15 Communication Diagram untuk Pilih Pengguna

Pada saat pengguna mengakses aplikasi maka pengguna akan di beri suguhan tampilan pilihan pengguna. Pengguna diharuskan memilih untuk menggunakan aplikasi sebagai PPTK atau sebagai BPP. Setelah pilihan diputuskan maka sistem akan memproses pilihan dan menampilkan menu utama bagi PPTK dan tampilan login bagi pengguna yang memilih sebagai BPP.

Gambar 16 *Communication Diagram* untuk Login

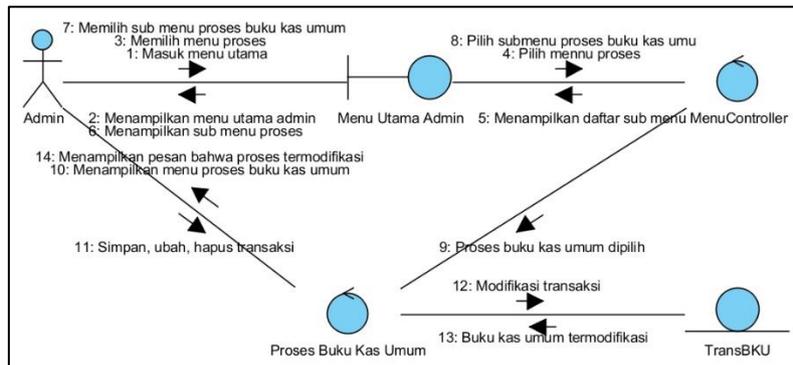
Untuk *use case Login*, setelah pengguna memilih BPP maka pengguna disuguhkan tampilan *Login*. Kemudian pengguna akan memasukkan *username* dan *password* yang sudah dibuatkan sebelumnya. Sistem akan mencocokkan *username* dan *password* yang dimasukkan, jika cocok maka pengguna akan dibawa langsung ke tampilan menu utama untuk BPP dan jika tidak cocok maka pengguna akan diberikan keterangan bahwa kombinasi *username* dan *password* yang dimasukkan ada yang salah.

Gambar 17 *Communication Diagram* untuk Kelola Data Induk

Selanjutnya adalah *use case Kelola Data*, di mana kelola data ini meliputi kelola data induk dan kelola data pengguna. Pada dasarnya *use case* keduanya sama, perbedaan hanya pada tampilan yang muncul serta basis data yang diakses.

Pengguna atau dalam hal ini sudah berubah status menjadi BPP masuk ke menu utama. Menu utama tertampil dan BPP memilih menu kelola data. kemudian sistem menampilkan daftar submenu yang ada di bawah menu kelola data, yaitu pengguna dan induk. Data induk sendiri memiliki anak menu yaitu Kode Kegiatan dan Kode Rekening. Setelah memilih salah satu submenu, sistem akan menampilkan antarmuka kelola data sesuai dengan pilihan yang dibuat oleh BPP.

Dalam menu kelola data ini, ada beberapa fungsi yang dapat diterapkan pada isi data yaitu tambah, ubah dan hapus data. Selain itu disediakan tombol Kosongkan untuk mengosongkan *field* serta tombol kembali untuk kembali ke menu utama. Setiap aksi yang dibuat oleh BPP, sistem akan memberikan keterangan di layar bawah aksi tersebut berhasil atau tidak.

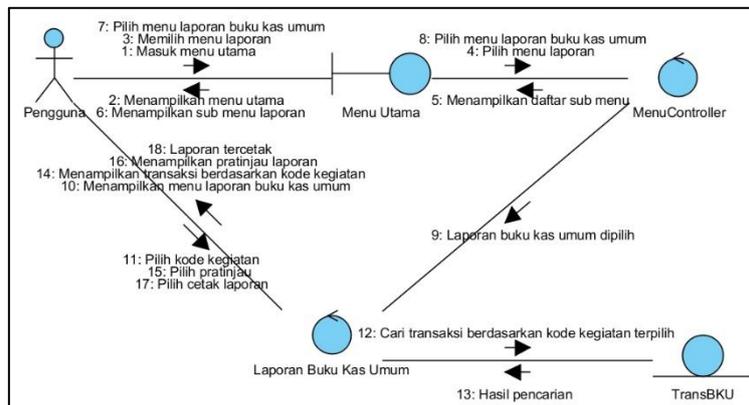


Gambar 18 *Communication Diagram* untuk Proses Buku Kas Umum

Sama halnya dengan use case kelola data, use case proses juga memiliki kesamaan antara proses yang satu dengan yang lainnya. Perbedaan masih terletak pada tampilan antar muka serta basis data yang diakses. Ada tiga proses yang memiliki kesamaan tersebut, yaitu proses buku kas umum, proses buku pajak, dan proses register penutupan kas.

Masih sebagai BPP, pengguna masuk menu utama dan akan disuguhkan antarmuka menu utama. Kemudian BPP memilih submenu yang ada di bawah menu Proses, yaitu proses buku kas umum, proses buku pajak, dan proses register penutupan kas. Ketika salah satu submenu dipilih, maka tampilan dari proses yang dipilih akan muncul.

Dalam tampilan proses, pengguna dapat menambah, mengubah dan menghapus data. Selain itu tombol kosongkan dan kembali masih disediakan untuk mengosongkan *field* dan kembali ke menu utama. Keterangan aksi pun masih akan tertampil jika pengguna melakukan aksi simpan, ubah dan hapus tadi.



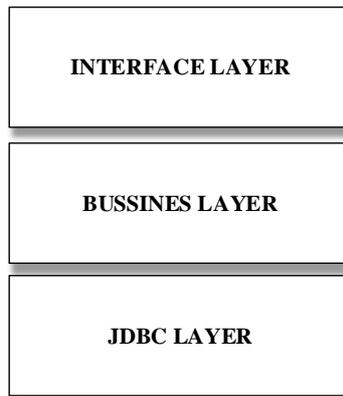
Gambar 19 *Communication Diagram* untuk Laporan Buku Kas Umum

Begitu pun pada Laporan, setiap laporan baik laporan buku kas umum, laporan buku pajak, laporan rincian per rekening, laporan keadaan kas, laporan pengiriman SPJ serta laporan register penutupan kas memiliki kesamaan dalam prosesnya. Perbedaan terletak pada tampilan yang disuguhkan serta basis data yang digunakan.

Ketika pengguna masuk ke menu utama, tampilan menu utama akan menghadirkan beberapa menu dan submenu di bawahnya. Ketika menu laporan dipilih, submenu dari laporan tadi akan muncul. Setelah dipilih salah satu laporan, maka tampilan laporan tersebut akan langsung muncul.

Pada laporan, hal yang bisa pengguna lakukan adalah melihat laporan sebelum dicetak. Serta menyaring laporan yang akan ditampilkan melalui pilihan yang disediakan. Ketika tombol pratinjau ditekan, maka pengguna dapat melihat laporan sebelum dicetak. Di tampilan pratinjau ini akan tertampil menu untuk mencetak laporan.

Tahap selanjutnya adalah *design*. Ada dua perancangan yang akan dibuat, yaitu rancangan sistem dan rancangan subsistem. Dalam perancangan sistem akan dibuat sebuah *layer diagram* dan *deployment diagram*. Kemudian dalam perancangan sub sistem akan dibuat sebuah *business class diagram*, *database schema*, serta *use case realization*.

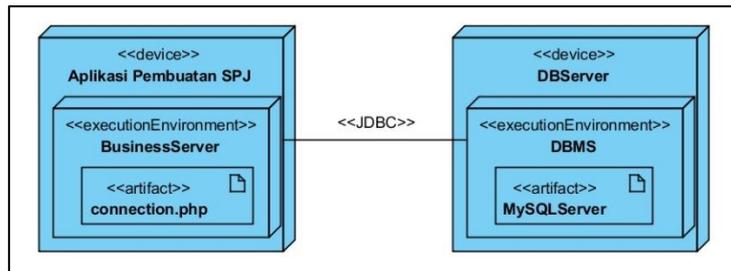


Gambar 20 Layer Diagram Aplikasi SPJ

JDBC (Java Database Connectivity) layer memuat pustaka-pustaka dari JDBC untuk mengakses basis data yang tersimpan dan digunakan oleh sistem. Layer ini memuat driver yang menghubungkan antara aplikasi dengan basis data. Koneksi dibuat dengan satu buah kelas yang berisi kode sumber untuk menggunakan JDBC di dalam aplikasi berbasis Java.

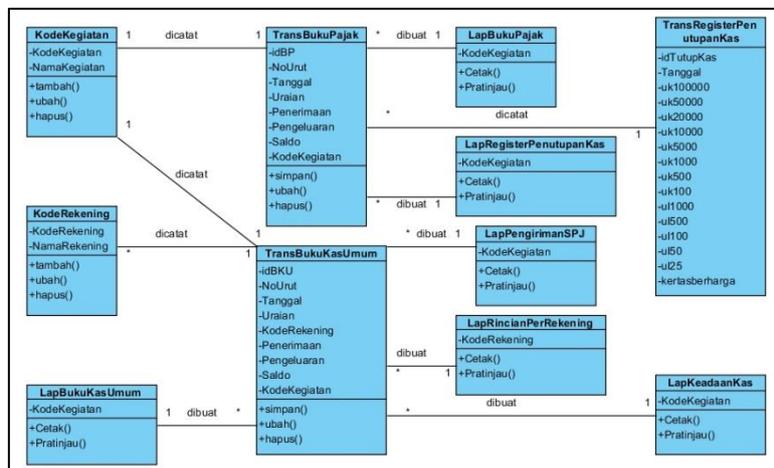
Layer bisnis adalah layer yang memuat semua operasi yang tersedia dalam sistem. Layer ini merupakan implementasi dari setiap entitas objek dari class diagram pada tahap analysis beserta berbagai macam objek pendukung. Objek-objek ini memuat kode JDBC untuk mengirimkan data dari dan ke dalam basis data.

Layer antarmuka merupakan layer yang menjadi media komunikasi antara pengguna dengan sistem. Layer ini menyediakan tampilan untuk pengguna agar dapat mengoperasikan sistem dengan lebih mudah.



Gambar 21 Deployment Diagram Aplikasi SPJ

Dalam sistem terdapat dua device yang menjadi pembangun aplikasi SPJ yang dibuat. Di mana masing-masing device memiliki fungsi sebagai tempat pengolahan data serta tempat penyimpanan data. Tempat pengolahan data merupakan aplikasi utama yang menggunakan akses JDBC sebagai jembatan untuk mengakses tempat penyimpanan data dengan menggunakan artifak connection.php sebagai jalan untuk mengaktifkan JDBC tersebut.



Gambar 22 Business Class Diagram

Diagram kelas pada bagian ini sama dengan diagram kelas yang terdapat pada tahap analisis. Hanya saja pada tahap ini, kelas-kelas telah dilengkapi dengan atribut serta metode masing-masing

Selanjutnya dalam *database scheme* yang dapat digambarkan ke dalam tabel berikut:

Tabel 1 Tabel Penjelasan Basis Data

Nama Tabel	Atribut	Tipe Data	Panjang Data	Keterangan	
kodkeg	idKodeKegiatan	integer	11	Primary Key	
	KodeKegiatan	varchar	15		
	NamaKegiatan	longtext			
kodrek	idKodeRekening	integer	11	Primary Key	
	KodeRekening	varchar	15		
	NamaRekening	varchar	110		
pengguna	idUser	integer	11	Primary Key	
	username	varchar	15		
	password	varchar	15		
	tipeuser	enum()			
transBKU	idBKU	integer	11	Primary Key	
	NoUrut	varchar	3		
	Tanggal	dDate			
	Uraian	text			
	KodeRekening	integer	11		Foreign Key
	Penerimaan	integer	12		
	Pengeluaran	integer	12		
	Saldo	integer	12		
transBP	KodeKegiatan	varchar	15	Foreign Key	
	idBP	varchar	8		Primary Key
	NoUrut	varchar	3		
	Tanggal	dDate			
	Uraian	text			
	Penerimaan	integer	12		
	Pengeluaran	integer	12		
	Saldo	integer	12		
tranTK	KodeKegiatan	varchar	15	Foreign Key	
	TipePajak	varchar	50		
	idTutupKas	integer	11		Primary Key
	Tanggal	date			
	Uk100000	integer	11		
	Uk50000	integer	11		
	Uk20000	integer	11		
	Uk10000	integer	11		
	Uk5000	integer	11		
	Uk1000	integer	11		
	Uk500	integer	11		
	Uk100	integer	11		
	U11000	integer	11		
	U1500	integer	11		
	U1100	integer	11		
	U150	integer	11		
U125	integer	11			
KertasBerharga	integer	11			

Untuk *use case realization* akan digambarkan pada sebuah *sequence diagram*. *Sequence diagram* akan menjadi penggambaran dari realisasi ini. Di mana diagram ini merupakan pengembangan dari *communication diagram* yang telah dibuat pada tahap sebelumnya. Satu *business service* digambarkan oleh satu *sequence diagram*.

Pada dasarnya *sequence diagram* ini sama dengan *communication diagram* yang telah dibuat sebelumnya. Hanya saja urutan proses terlihat lebih jelas. Proses serta perbedaan dari masing-masing kategori *use case* pun sama dengan *communication diagram*.

Tahap selanjutnya adalah implementasi yaitu tahap kodifikasi dimana rancangan yang telah dibuat diterjemahkan ke dalam bahasa pemrograman. Setelah itu dilakukan pengujian terhadap fungsionalitas program yang telah dibangun. Berikut adalah cpengujian terhadap fungsi-fungsi program yang ada.

Berikut adalah tampilan dari implementasi kode sumber yang diterapkan. Ada beberapa tampilan yang akan dibahas, mulai dari tampilan pilih pengguna, login menu utama, kelola data,

transaksi dan laporan. Di mana pada bagian kelola data, transaksi dan laporan hanya akan ditampilkan satu untuk setiap kategori dan sisanya akan disimpan pada lampiran.



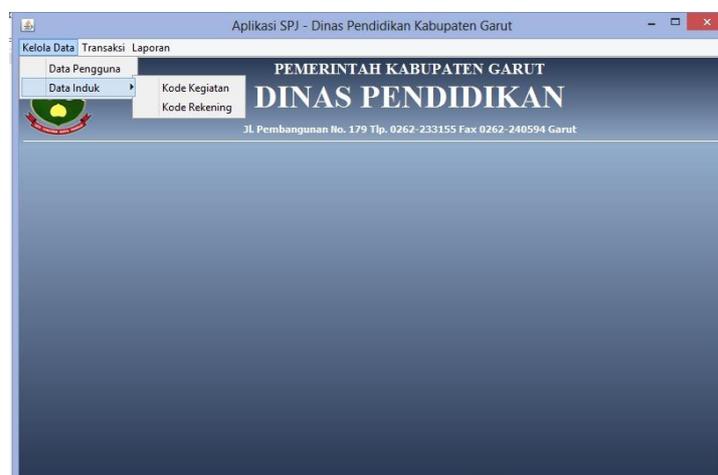
Gambar 23 Tampilan Pilih Pengguna

Tampilan ini adalah tampilan program yang pertama kali dijumpai oleh pengguna. Pada tampilan ini pengguna akan dihadapkan pada dua pilihan pengguna, yaitu BPP yang bertindak sebagai admin dan PPTK yang bertindak hanya sebagai pengguna biasa. Setelah memilih masing-masing jenis, pengguna akan dihadapkan pada *form login*. Berikut adalah tampilan *login* tersebut.



Gambar 24 Tampilan Login

Pengguna baik sebagai BPP atau sebagai PPTK akan diminta untuk memasukkan *username* dan *password* pada *form Login* ini untuk bisa melanjutkan ke menu utama dan mengoperasikan aplikasi. Dengan asumsi bahwa *username* dan *password* telah didapatkan sebelumnya. Jika kombinasi *username* dan *password* cocok maka akan dibawa ke halaman utama aplikasi dan jika kombinasi *username* dan *password* salah maka akan ada peringatan bahwa kombinasi *username* dan *password* yang dimasukkan tidak cocok. Berikut adalah tampilan menu utama untuk kedua pengguna.



Gambar 25 Tampilan Antarmuka Menu Utama

Tampilan menu utama untuk kedua pengguna pada dasarnya tidak ada perbedaan. Pada aplikasi ini terdapat 3 menu utama yaitu Kelola Data terdapat sub menu Data Pengguna dan Data Induk (Kode Kegiatan dan Kode Rekening). Pada menu Transaksi terdapat sub menu Buku Kas Umum, Buku Pajak dan Register Tutup Kas. Sedangkan pada menu Laporan terdapat 6 sub menu, yaitu Buku Kas Umum, Buku Pajak, Rincian per Rekening, Keadaan Kas, Pengantar SPJ dan Register Tutup Kas. Namun tidak semua fungsi menu utama pada tampilan PPTK akan diaktifkan, menu tersebut adalah Kelola Data dan Transaksi. Sehingga PPTK hanya dapat mengakses menu Laporan saja. Lain halnya dengan BPP di mana semua menu akan terlihat dan bisa diakses penuh baik itu Kelola Data, Transaksi maupun Laporan. Semua menu ini bisa diakses secara alternatif dengan menggunakan tombol *Tab* pada *keyboard* yang berfungsi untuk memindahkan kursor pada setiap menu ketika pengguna sedang tidak menggunakan *mouse*. Berikut adalah tampilan dari ketiga menu tersebut, yang akan diwakilkan oleh satu sub menu pada masing-masing menu.



Gambar 26 Tampilan Kelola Data Pengguna

Kelola Data Pengguna adalah salah satu contoh sub menu dari menu Kelola Data. Tampilan pada sub menu ini sama untuk semua sub menu pada Data Induk dan Transaksi yang menghadirkan beberapa tombol, *view area*, dan kolom isian data. Tombol-tombol yang terdiri dari Tambah, Ubah, Hapus, Simpan, Kosongkan dan Kembali.



Gambar 27 Tampilan Transaksi Buku Kas Umum

Seperti telah disebutkan sebelumnya bahwa keberadaan komponen dalam tampilan ini sama dengan tampilan pada menu kelola data. dengan beberapa tombol serta kolom isian data serta *view area* yang menampilkan isi basis data. Untuk mengisi kolom tanggal disediakan fasilitas kalender

untuk memilih tanggal transaksi. Selain itu pada keadaan *default* tanggal akan terisi sesuai dengan tanggal yang terpasang pada komputer. Kemudian pada kolom Kode Rekening telah disediakan daftar kode rekening yang telah diperoleh dari data aplikasi RKA, sehingga kita hanya cukup memilih saja tanpa harus mengetikkannya. Jika proses pencatatan transaksi telah selesai, kita dapat melihat dan mencetak hasilnya pada menu laporan.

Pada menu laporan ini setiap laporan memiliki tombol dan pilihan filter cetak yang sama, yakni filter laporan berdasarkan bulan dan tahun. Namun ada satu laporan yang memiliki filter berdasarkan kode akun, yaitu pada laporan Rincian per Rekening. Selebihnya sama-sama memiliki tombol Tampilkan, Pratinjau dan Keluar. Di mana tombol Tampilkan akan menampilkan data yang telah di filter di view area sebagai data yang akan dicetak nantinya. Tombol Pratinjau adalah tombol untuk melihat laporan sebelum dicetak. Pada tampilan Pratinjau inilah fungsi cetak dokumen terdapat. Selain itu laporan juga dapat disimpan dalam berbagai format dokumen. Berikut adalah salah satu contoh tampilan Pratinjau Laporan.

PEMERINTAH KABUPATEN GARUT
DINAS PENDIDIKAN
Jl. Pembangunan No. 179 Tlp. 0262-233155 Fax 0262-240594 Garut

LAPORAN BUKU KAS UMUM

Preview Laporan
Bulan: October Tahun: 2013 Tampilkan

PEMERINTAH KABUPATEN GARUT
BUKU KAS UMUM

SKPD :
Penggunaan Anggaran :
Bendahara Pengeluaran Pembantu :

No Urut	Tanggal	Uraian	Kode Rekening	Penerimaan Rp.	Pengeluaran Rp.	Saldo Rp.
1	2	3	4	5	6	7
1	27/10/2013	Program peningkatan kapasitas sumber	1010105	125.000.000,00	0,00	125.000.000,00
2	27/10/2013	Honorarium tenaga ahli program	5210103	0,00	41.600.000,00	83.400.000,00
3	27/10/2013	Dibayarkan kepada Nove Copy Center	5220101	0,00	19.700.000,00	63.700.000,00
4	27/10/2013	Dibayarkan kepada Top Photo belanja	5220102	0,00	250.000,00	63.450.000,00
5	27/10/2013	Dibayarkan kepada PT. POS belanja	5220104	0,00	120.000,00	63.330.000,00
6	27/10/2013	Dibayarkan kepada Mega Printing	5220104	0,00	2.800.000,00	60.530.000,00
7	27/10/2013	Dibayarkan kepada Malabar Photo...	5220101	0,00	50.000,00	60.480.000,00

Page 1 of 1
Pratinjau Keluar

Gambar 28 Tampilan Laporan Buku Kas Umum

V. KESIMPULAN/RINGKASAN

Kesimpulan yang didapat dari hasil penelitian yang sudah dilakukan diantaranya:

1. Aplikasi menggunakan sumber daya basis data dari aplikasi RKA, sehingga dokumen hasil keluaran RKA yang menjadi acuan dari pembuatan Surat Pertanggungjawaban dapat digunakan secara maksimal;
2. Dengan penggunaan basis data dari aplikasi RKA yang telah ada, proses *input* data dapat dikurangi sehingga tidak ada redundansi pekerjaan *input* data;
3. Dengan berkurangnya proses *input* data, kesalahan data akibat *input* data pun bisa dikurangi;
4. Dengan berkurangnya proses *input* data serta sedikitnya kesalahan yang mungkin timbul, maka waktu pengerjaan pembuatan surat pertanggungjawaban pun pada akhirnya bisa dipersingkat.

UCAPAN TERIMA KASIH

Penulis A. R. mengucapkan terima kasih kepada kedua orang tua yang selalu membantu secara moril maupun materil. Penulis juga perkenankan untuk menyampaikan ucapan terima kasih kepada Bapak Eri Satria, M.Si. selaku pembimbing I dan ibu Kiki Aisyah, MT. selaku pembimbing II yang telah memberikan arahan serta bimbingan selama penyelesaian penelitian ini.

DAFTAR PUSTAKA

- [1] Dhanta, R. (2009). *Kamus Istilah Komputer Grafis & Internet*. Surabaya: Indah.
- [2] Fajriah, S. (2008). Aplikasi Pengelolaan Dana dan Belanja Pemerintah Daerah Berbasis Web. diakses pada 12 Juni 2013 dari Google.com di World Wide Web: <http://openjurnal.politekniktelkom.ac.id/index.php/Jurnal%20PA/KA/2008>.
- [3] Hidayat, A.S., & Samopa, F. (2012, September). Pengembangan Sistem Informasi Akuntansi dan Pengelolaan Keuangan Modul Bendahara Menggunakan Teknologi Java dan PostgreSQL. diakses pada 12 Juni 2013 dari Google.com di World Wide Web: <http://ejurnal.its.ac.id/index.php/teknik/article/view/1248>.
- [4] Jogiyanto, H. M., (1999). Analisis dan Disain Sistem Informasi. Yogyakarta: Andi.
- [5] Kamus Besar Bahasa Indonesia. (1998). Jakarta : Pustaka Amani.
- [6] Liang, Y. D. (2011). *Introduction to Java Programming, 8th Edition*. New Jersey: Prentice Hall.
- [7] O'Docherty, M. (2005). Object Oriented Analysis and Design Understanding System Development with UML. England: John Wiley & Sons Ltd.
- [8] Pataha, M. Faisal. (2010, Juli 08). Analisis dan Perancangan Aplikasi Laporan Keuangan Daerah Pada Dinas Pendidikan, Pemuda dan Olahraga Kabupaten Maluku Tenggara. diakses pada 12 Juni 2013 dari Google.com di World Wide Web: http://repository.amikom.ac.id/index.php/add_downloader/PUBLIKASI_06.11.1222.pdf/2219.