

PEMODELAN SISTEM INFORMASI KEUANGAN DAERAH PADA DIREKTORAT EVALUASI PENDANAAN DAN INFORMASI KEUANGAN DAERAH

Demmy Dharma Bhkati¹

¹Pendidikan TIK STKIP Garut

¹cyifadetra@yahoo.co.id

Abstrak-Tata cara pengiriman data informasi keuangan daerah dari pemerintah daerah ke Direktorat EPIKD masih dirasa belum efektif, penyimpanan elemen data yang disatukan dalam sebuah tabel dan penggunaan berkas *csv* sebagai media berkas pun menjadi kendala tatkala harus di migrasi ke dalam basis data SIKD dan dibuat LKPD sebagai tujuan akhir dari SIKD. penelitian ini yang memiliki tujuan untuk menganalisis proses bisnis sistem informasi keuangan daerah, menganalisis kebutuhan sistem informasi keuangan daerah di kementerian keuangan, dan untuk merancang sistem informasi keuangan daerah berdasarkan proses bisnis SIKD. Proses analisis dan merancang SIKD pada penelitian ini menggunakan berbagai diagram pemodelan berbasis objek, yaitu UML dan SoaML. SoaML digunakan untuk kebutuhan pemodelan perancangan *web service* berteknologi SOAP. *Rational Unified Process* (RUP) yang terdiri *Inception* dan *Elaboration* dengan kegiatan *business modelling*, *requirements*, dan *analysis and design* pada setiap tahapan digunakan untuk memodelkan system. Penggunaan *web service* dengan teknologi SOAP mampu melayani komunikasi data secara netral tidak tergantung kepada *platform* yang digunakan sehingga semua aplikasi keuangan daerah yang saat ini dipakai di berbagai pemerintah daerah dapat mengirimkan data informasi keuangan daerah dengan berbasiskan teknologi *web service*.

Kata kunci- SIKD, LKPD, UML, SoaML.

I. PENDAHULUAN

Peraturan Menteri Keuangan Nomor 46/PMK.02/2006 tentang Tata Cara Penyampaian Informasi Keuangan Daerah, yang telah dirubah dengan Peraturan Menteri Keuangan Nomor 4/PMK.07/2011, maka pemerintah daerah wajib mengirimkan Informasi Keuangan Daerah (IKD) secara terkomputerisasi kepada pemerintah pusat dalam hal ini Kementerian Keuangan yang secara teknis dilaksanakan pada Direktorat Evaluasi Pendanaan dan Informasi Keuangan Daerah (EPIKD) yang berada di dalam Direktorat Jenderal Perimbangan Keuangan (DJPK) [1] [2].

Pengiriman IKD dari pemerintah daerah ke pemerintah pusat dimaksudkan untuk mendukung pembentukan statistik keuangan pemerintah yang akan memberikan gambaran luas tentang aktifitas pemerintah dalam berbagai sektor pembangunan, dalam hal ini khususnya pembangunan di daerah dengan dukungan dana anggaran pendapatan dan belanja daerah (APBD) dan dana lainnya. Jenis Informasi Keuangan Daerah yang

dikirimkan pemerintah daerah kepada pemerintah pusat untuk membentuk Laporan Keuangan Pemerintah Daerah (LKPD) terdiri dari APBD dan Laporan Realisasi Anggaran (LRA) [2]

Selama ini proses pengiriman IKD dari pemerintah daerah kepada kementerian keuangan dilakukan dengan dua cara, yaitu : dengan datang langsung ke Direktorat EPIKD DJPK Kementerian Keuangan untuk menyerahkan IKD dalam bentuk *hardcopy* dan dengan cara mengirimkan berkas dalam bentuk *csv* yang di padatkan melalui aplikasi Komandan SIKD. Kedua cara tersebut memiliki kekurangan masing-masing, terutama dalam segi masalah waktu pengiriman informasi yang tidak sesuai dengan waktu yang ditetapkan, dari segi biaya juga menjadi masalah dengan seringnya pemerintah daerah melakukan perjalanan dinas ke kementerian keuangan sehingga menjadi beban tambahan bagi pemerintah daerah. Kemudian masalah inkonsistensi data yang diterima juga terjadi dikarenakan data dalam bentuk *csv* akan diinput ulang ke dalam sistem yang berlaku saat ini di kementerian keuangan, sehingga terdapat kemungkinan IKD yang terdapat di kementerian keuangan akan berbeda dengan data yang dikirimkan oleh pemerintah daerah dikarenakan kesalahan input [2].

Masalah pengiriman IKD dari pemerintah daerah ke Kementerian Keuangan secara *online* juga terjadi dikarenakan heterogenitas aplikasi dan sistem yang dipergunakan oleh pemerintah daerah dalam mengolah data dan informasi keuangan daerah masing-masing. Oleh karena itu diperlukan adanya suatu sistem yang dapat menyediakan *interface* pelayanan transfer informasi dari pemerintah daerah ke pemerintah pusat yang mampu menerima informasi keuangan daerah secara *realtime* dan mampu menangani heterogenitas sistem dan aplikasi pengelola keuangan yang dipergunakan di berbagai pemerintah daerah.

II. MODEL PERANGKAT LUNAK

Model dari sebuah sistem perangkat lunak adalah representasi abstrak dari pengetahuan mengenai sistem. Sebuah model merupakan idealisasi dari aspek yang terpilih dari sebuah sistem seperti struktur, operasi, atau karakteristik sistem yang lain. Model adalah alat yang sangat berguna untuk menganalisis maupun merancang sistem. Sebagai alat komunikasi yang sangat efisien, model dapat menunjukkan bagaimana suatu operasi bekerja dan mampu merangsang untuk berpikir bagaimana meningkatkan atau memperbaikinya [3].

2.1 Unified Modeling Language (UML)

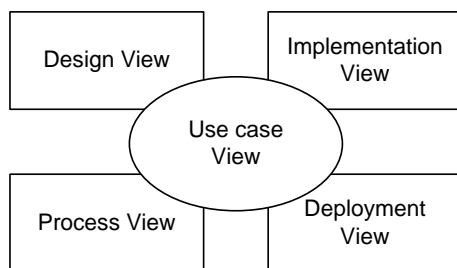
UML seperti namanya merupakan suatu penyatuandarijumlah bahasapemodelanberorientasi objek yang sudah ada sebelumnya, yaitu Grady Booch *Object-Oriented Design* (OOD), Jim Rumbaugh *Object Modeling Technique* (OMT), dan Ivar Jacobson *Object-Oriented Software Engineering* (OOSE). UML dimaksudkan untuk mengintegrasikan wawasan dari ketiga metode tersebut, dan untuk mendorong penyebaran teknik berorientasi objek menyediakan notasi tunggal yang diakui secara luas. UML dimaksudkan untuk menjadi bahasa yang terdefinisi dengan baik yang dapat dipergunakan secara produktif dalam berbagai proses yang berbeda [4].

UML adalah bahasa untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. UML menyediakan notasi yang tepat yang kita butuhkan untuk pemodelan sistem perangkat lunak. Hal ini penting untuk dicatat bahwa UML bukan hanya cara untuk mendokumentasikan ide-ide yang ada, UML juga membantu para pengembang menciptakan ide-ide, serta cara berkomunikasi antar mereka [5].

UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

2.2 View Model

UML digambarkan melalui visi dari struktur sistem perangkat lunak yang dikenal sebagai 4 + 1 *view model*, seperti dapat dilihat pada gambar 1 yang terdiri dalam lima tampilan, salah satunya adalah *use case view* yang memiliki peran khusus dalam mengintegrasikan isi dari empat *view* lainnya.



Gambar 1. 4+1 *View Model* [2]

Lima *view* yang ditunjukkan pada gambar 1 tidak berhubungan dengan diagram formal tertentu yang dapat dijelaskan dalam UML. Sebaliknya, setiap tampilan berhubungan dengan cara pandang tertentu dari sistem yang di analisis. Setiap *view* menyoroti aspek yang berbeda dari sistem yang menjadi kepentingan pemangku kepentingan tertentu. Penjelasan lengkap mengenai suatu sistem dapat dibentuk dengan menggabungkan informasi yang ditemukan dari semua lima *view*. Akan tetapi untuk tujuan tertentu ada kemungkinan cukup untuk

mempertimbangkan Informasi yang terkandung hanya dalam subset dari *view*.

Use case view mendefinisikan perilaku eksternal sistem dan merupakan kepentingan bagi pengguna akhir, analis dan penguji. *Use case view* mendefinisikan kebutuhan sistem, dan menjadi batasan pengembangan untuk *view* yang lain, yang menggambarkan aspek-aspek tertentu dari desain atau pembuatan sistem. Hal inilah yang menyebabkan *use case view* memiliki peran utama dan sering disebutkan sebagai pengatur proses pembangunan sistem.

Design view menggambarkan struktur logis untuk mendukung kebutuhan fungsional yang telah dinyatakan di dalam *use case view*. *Design view* terdiri dari definisi komponen program, terutama *class*, spesifikasi dari data yang mereka pegang, dan perilaku serta interaksi antar *class*. Informasi yang terkandung dalam *design view* adalah untuk kepentingan *programmer*, sebagai rincian dari fungsi bagaimana sistem akan diimplementasikan.

Implementation view menggambarkan komponen fisik dari sistem yang akan dibangun dan diimplementasikan, hal ini berbeda dari komponen logis yang telah dijelaskan sebelumnya di dalam *design view*. *Implementation view* mencakup hal-hal seperti *file executable*, *library* kode dan basis data. Informasi yang terkandung dalam *implementation view* relevan dengan kegiatan seperti manajemen konfigurasi dan integrasi sistem.

Process view berkaitan dengan isu-isu mengenai konkurensi di dalam sistem, dan *deployment view* menggambarkan bagaimana komponen fisik didistribusikan di seluruh lingkungan fisik seperti jaringan komputer dalam sistem berjalan. *Process view* dan *deployment view* mengatasi kebutuhan nonfungsional dari sistem seperti toleransi kesalahan dan masalah kinerja.

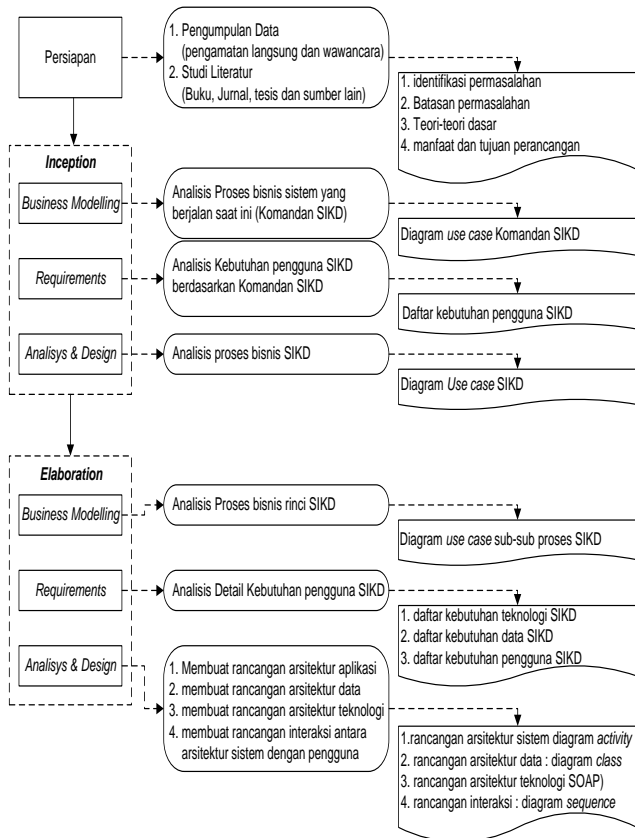
2.3 Service Oriented Architecture Modelling Language (SoaML)

SoaML merupakan standar bahasa pemodelan yang dibangun oleh *Open Management Group* (OMG) yang menyediakan domain bahasa pemodelan netral untuk arsitek dan model SOA. SoaML menggunakan pendekatan UML untuk melakukan desain sistem berbasis layanan. SoaML dibangun untuk menyediakan kemampuan pemodelan yang dibutuhkan dalam perancangan desain sistem berbasis layanan [6]. SoaML menggunakan kemampuan pemodelan dari UML untuk membantu memisahkan berbagai kepentingan layanan sehingga dapat dipergunakan kembali dalam arsitektur layanan. Soa memungkinkan kita untuk menggambarkan dan memahami bagaimana orang-orang, organisasi dan sistem komponen bekerja sama dengan menggunakan layanan untuk mencapai tujuan bisnis.

Tujuan dari SoaML adalah untuk mendukung kegiatan modeling layanan dan desain dan untuk masuk ke dalam pendekatan pengembangan model-driven keseluruhan [3]. Spesifikasi SoaML mengandung dua buah metamodel, yaitu metamodel SoaML dan SoaML UML *Profile*. SoaML UML *profile* mendukung pemodelan arsitektur berorientasi layanan, termasuk spesifikasi sistem pelayanan, spesifikasi antarmuka layanan individual, dan spesifikasi dari implementasi layanan.

III. KERANGKA PENELITIAN

Kerangka penelitian didasarkan pada urutan proses model pengembangan perangkat lunak RUP, yang terdiri dari beberapa tahapan, yaitu *inception*, *elaboration*, *construction*, dan *transition*. Penelitian dibatasi pada tahapan *inception* dan *elaboration* pada model RUP menurut Tsui yang terdiri dari tiga tahapan yaitu *Business Modelling*, *Requirements*, serta *analysis* dan *design*, yang diperinci sehingga menjadi kerangka penelitian seperti pada gambar 2.



Gambar 2. Kerangka Penelitian

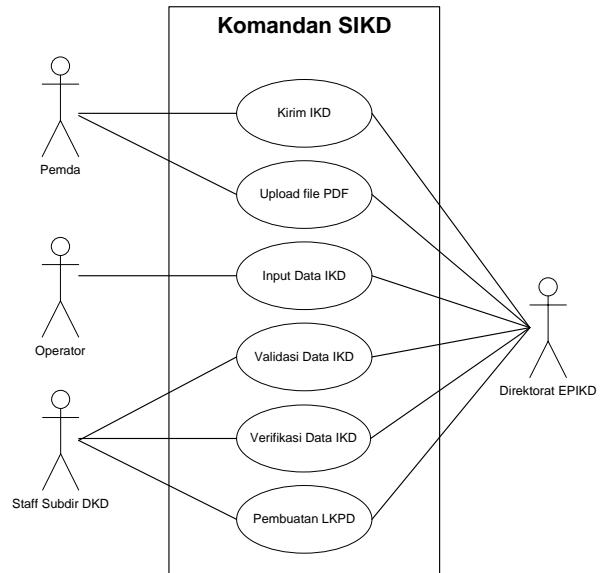
IV. ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Proses Bisnis Komandan SIKD

Proses bisnis Komandan SIKD secara garis besar terdiri dari enam proses yang saling berkaitan, yaitu:

1. Proses Kirim IKD dari Pemerintah Daerah ke Direktorat EPIKD
2. Proses Upload berkas PDF IKD dari Pemerintah daerah ke Direktorat EPIKD
3. Proses Input data IKD yang dilakukan oleh Operator
4. Proses Validasi Data IKD
5. Proses Verifikasi Data IKD
6. Proses Pembuatan LKPD

Pada gambar 3 terdapat diagram *use case* dari proses bisnis di dalam Komandan SIKD yang berjalan pada saat ini di Direktorat EPIKD.



Gambar 3. Diagram Use Case proses bisnis komandan SIKD

4.2 Analisis Proses Bisnis SIKD

Berdasarkan hasil analisis proses bisnis Komandan SIKD yang berjalan saat ini sebagai sistem pengelola keuangan daerah tingkat nasional dan analisis berbagai kebutuhan pengguna, maka dapat dilihat berbagai hal yang dapat lebih ditingkatkan untuk memaksimalkan pengelolaan keuangan daerah di Direktorat EPIKD, baik dalam hal penyampaian IKD, struktur data, atau dalam hal pemrosesan data pada komandan SIKD. Terdapat empat hal sangat penting yang harus diperhatikan pada Komandan SIKD, yaitu data, pemanfaatan data, prosedur, dan sistem aplikasi. Selanjutnya dilakukan perancangan proses bisnis dari SIKD sesuai dengan peraturan yang berlaku dan kebutuhan pengguna untuk meningkatkan Komandan SIKD sehingga menjadi sebuah sistem yang mampu melakukan pengolahan data keuangan daerah secara komprehensif.

Gambar 4 menampilkan diagram *use case* dari SIKD yang merupakan peningkatan dari Komandan SIKD berdasarkan hasil analisis kebutuhan pengguna SIKD di Direktorat EPIKD.

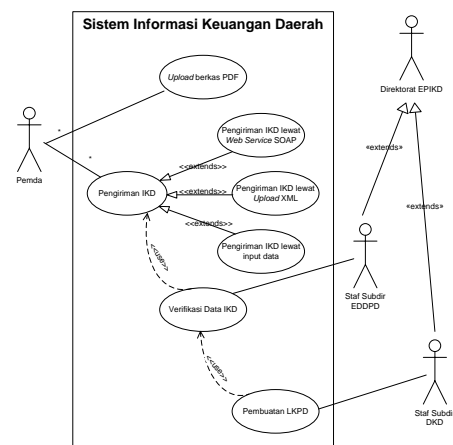


Diagram Use Case SIKD

Gambar 4.

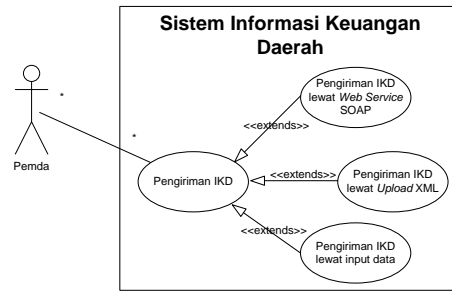
Diagram *use case* SIKD pada gambar 4 memiliki empat orang aktor yang terlibat di dalam sistem informasi keuangan daerah, yaitu pemda sebagai aktor yang melakukan pengiriman IKD dan upload berkas PDF ke Direktorat EPIKD, staf sub direktorat EDDPD yang melakukan proses verifikasi data IKD, staf sub direktorat DKD yang mengelola data IKD dan melakukan proses pembuatan LKPD, serta Direktorat EPIKD sendiri yang bertanggungjawab sebagai pengelola keuangan daerah di kementerian keuangan. Proses bisnis SIKD secara garis besar terdiri dari empat proses yang saling berkaitan, yaitu:

1. Proses *upload* berkas PDF IKD dari pemerintah daerah melalui SIKD
2. Proses pengiriman IKD dari pemerintah daerah melalui SIKD
3. Proses verifikasi Data IKD oleh staf subdir EDDPD.
4. Proses pembuatan LKPD oleh staf subdir DKD

4.3 Analisis Proses Bisnis Detail SIKD

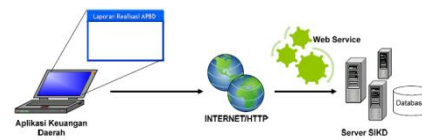
Proses bisnis pengiriman IKD dari pemerintah daerah kepada direktorat EPIKD melalui SIKD dapat dilakukan dengan menggunakan tiga cara seperti terlihat pada gambar 5 diagram *use case* dari tata cara pengiriman data IKD dari pemerintah daerah ke Direktorat EPIKD, yaitu:

1. Pemerintah daerah dapat mengirimkan IKD dengan memanfaatkan fasilitas *web service* (SOAP) pada SIKD. Fasilitas *web service* dipergunakan di SIKD agar pengiriman data IKD dapat dilakukan secara *online* dan *realtime* mengingat berbagai macam teknologi dan aplikasi yang dipergunakan pemerintah daerah dalam membantu pengelolaan keuangan daerahnya masing-masing.
2. Pemerintah daerah dapat mengirimkan IKD dengan memanfaatkan fasilitas *upload XML* pada SIKD. Fasilitas *upload XML* dipergunakan di SIKD untuk pemerintah daerah yang memiliki aplikasi dalam membantu pengelolaan keuangan daerahnya akan tetapi tidak memiliki jaringan internet yang bagus. Dengan adanya fasilitas *upload XML*, maka pemerintah daerah dapat mengirimkan datanya dengan cara membuat berkas XML melalui aplikasi pengelolaan keuangan daerahnya dan mengirimkannya di daerah lain yang memiliki jaringan internet yang lebih baik, sehingga data IDK tetap dapat tersampaikan dengan cepat.
3. Pemerintah daerah dapat mengirimkan Ikd dengan memanfaatkan fasilitas input data secara langsung pada SIKD. Fasilitas input data pada SIKD dipergunakan oleh pemerintah daerah yang sama sekali tidak memiliki aplikasi dalam membantu mengelola keuangan daerahnya. Pemerintah daerah dapat melakukan memasukkan data IKD secara langsung melalui SIKD, sehingga data tetap dapat tersampaikan ke Direktorat EPIKD.



Gambar 5. Diagram *Use Case* Pengiriman IKD

Proses Kirim IKD pada gambar 5 memiliki satu buah aktor yang terlibat di dalamnya yaitu pemda sebagai aktor pengirim data IKD dan tiga buah proses turunan yang dapat dipergunakan oleh pemerintah daerah untuk mengirimkan data IKD yang terdiri dari APBD dan LRA kepada Direktorat EPIKD. Ketiga proses tersebut memiliki tujuan dan kelebihan masing-masing. Proses pertama yang dapat dipergunakan oleh pemerintah daerah dalam mengirimkan data IKD adalah proses pengiriman IKD melalui fasilitas *web service* dengan teknologi SOAP seperti terlihat pada gambar 6 ilustrasi pengiriman IKD melalui *web service* teknologi SOAP.



Gambar 6. Ilustrasi Pengiriman IKD melalui *Web Service* Teknologi SOAP

4.4 Analisis Detail Kebutuhan Pengguna SIKD

Berdasarkan hasil analisis proses bisnis SIKD secara detail maka selanjutnya akan dilakukan analisis kebutuhan pengguna SIKD secara lebih mendetail. Analisis kebutuhan pengguna ini akan dipergunakan untuk melakukan tahapan perancangan dari SIKD yang akan dikembangkan, daftar kebutuhan pengguna SIKD dapat dilihat pada tabel 1.

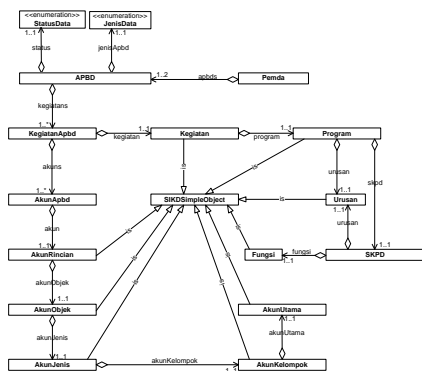
Tabel 1. Daftar Kebutuhan Pengguna SIKD

No	Kebutuhan
1.	Dibutuhkan rancangan arsitektur aplikasi yang menggambarkan SIKD secara jelas dan rinci.
2.	Dibutuhkan rancangan arsitektur data IKD yang terdiri dari APBD dan LRA yang terstruktur, jelas, dapat dikomunikasikan dan diolah secara mudah oleh pemerintah daerah ataupun oleh Direktorat EPIKD.
3.	Dibutuhkan rancangan arsitektur aplikasi pada SIKD yang mampu menerima IKD dari berbagai macam aplikasi dan sistem yang terdapat di pemerintah daerah.
4.	Dibutuhkan rancangan teknologi yang mampu menjembatani kesenjangan dan heterogenitas sistem yang terdapat di setiap pemerintah daerah.
5.	Dibutuhkan rancangan arsitektur aplikasi pada SIKD yang mampu menerima IKD dan membantu pemerintah daerah yang telah menggunakan aplikasi dalam pengolahan data keuangan daerahnya akan

	tetapi memiliki kekurangan dalam jaringan <i>internet</i> di daerahnya.
6.	Dibutuhkan rancangan arsitektur aplikasi pada SIKD yang mampu menerima IKD dan membantu pemerintah daerah yang belum memiliki aplikasi dalam pengolahan data keuangan daerahnya.
7.	Dibutuhkan rancangan arsitektur interaksi antara berbagai proses yang terdapat di SIKD dengan pengguna SIKD.

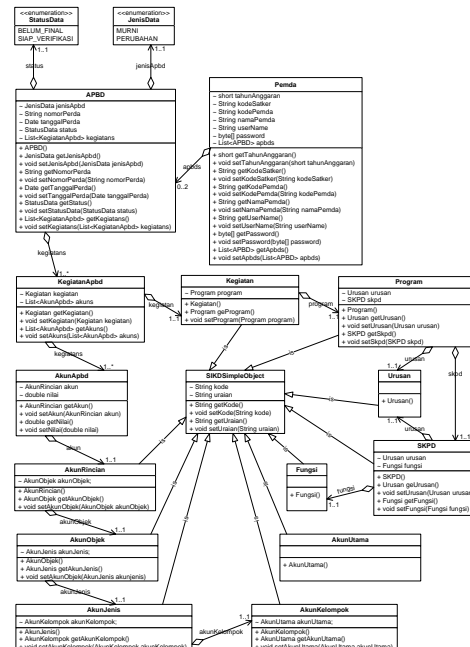
4.5 Perancangan Arsitektur Data APBD

Data APBD yang dikirimkan ke server SIKD merupakan data anggaran pendapatan dan belanja daerah secara rinci sampai kepada kode rekening rincian objek (level 5) beserta besaran nilainya. Berdasarkan elemen-elemen data pada Komandan SIKD yang harus diserahkan oleh pemerintah daerah dan berdasarkan kebutuhan pengguna SIKD maka dibuat diagram *class* APBD yang mencerminkan format data IKD APBD yang harus dikirimkan.



Gambar 7. Taksonomi Diagram *Class* APBD

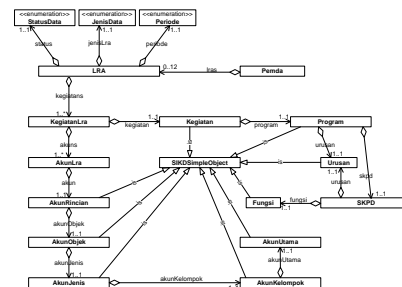
Untuk lebih memperjelas struktur *class-class* yang dipergunakan sebagai format data IKD APBD yang harus dikirimkan pemerintah daerah kepada Direktorat EPIKD, maka untuk selanjutnya akan digambarkan diagram *class* untuk struktur data IKD APBD secara lebih rinci dengan semua atribut dan *behaviour* yang terdapat di dalam setiap *class*.



Gambar 8. Diagram *Class* Data APBD

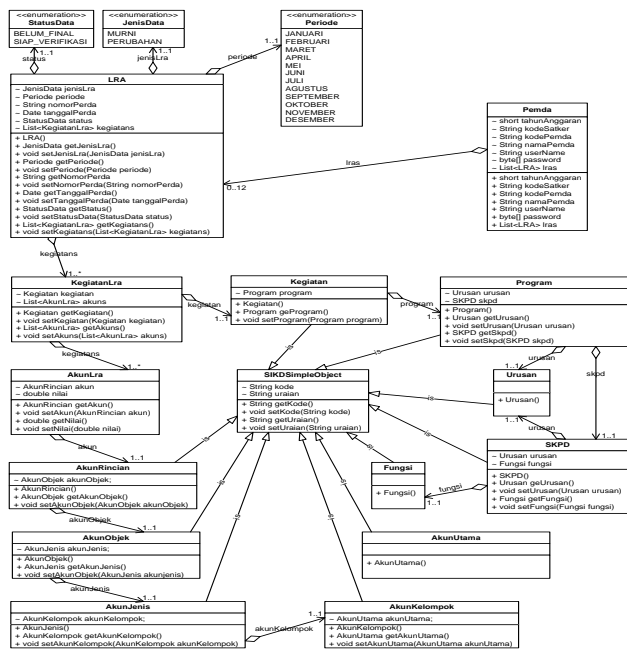
4.6 Arsitektur Data LRA

LRA merupakan data transaksi pelaksanaan penerimaan dan pengeluaran pada kode rekening APBD, yaitu meliputi seluruh realisasi penerimaan dan pengeluaran pada kode rekening rincian objek (level 5). Struktur data LRA harus dikirimkan ke server SIKD dengan periode pengiriman setiap bulan. Berdasarkan elemen-elemen data pada Komandan SIKD dan berdasarkan kebutuhan pengguna SIKD maka dibuat diagram *class* LRA yang mencerminkan format data IKD LRA yang harus dikirimkan.



Gambar 9. Taksonomi Diagram *Class* LRA

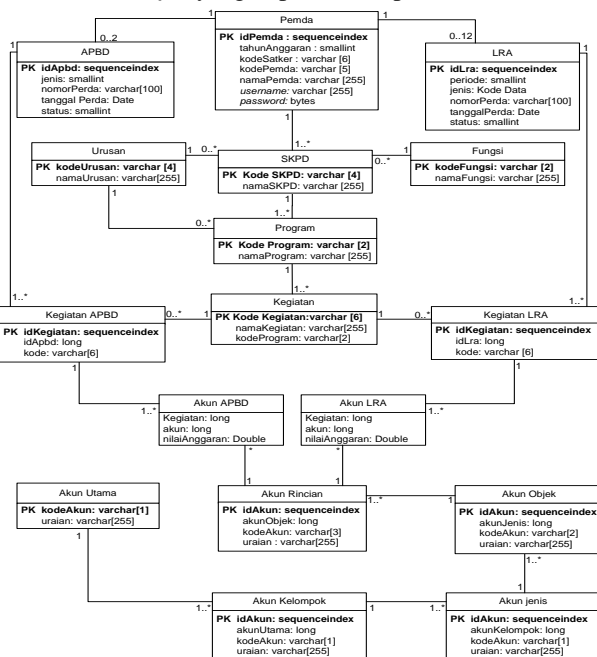
Untuk lebih memperjelas struktur *class-class* yang dipergunakan sebagai format data IKD LRA yang harus dikirimkan pemerintah daerah kepada Direktorat EPIKD, maka untuk selanjutnya akan digambarkan diagram *class* untuk struktur data IKD LRA secara lebih rinci dengan semua atribut dan *behaviour* yang terdapat di dalam setiap *class*.



Gambar 10. Diagram Class Data LRA

4.7 Arsitektur Data SIKD

Arsitektur data pada SIKD merupakan gabungan dari rancangan arsitektur data APBD dan LRA yang sudah dilakukan sebelumnya dengan menggunakan diagram class. Arsitektur data SIKD akan dipergunakan sebagai rancangan basis data pada SIKD. Rancangan arsitektur data SIKD akan digambarkan dengan menggunakan UML Data Model Profile yang dapat dilihat pada 11.

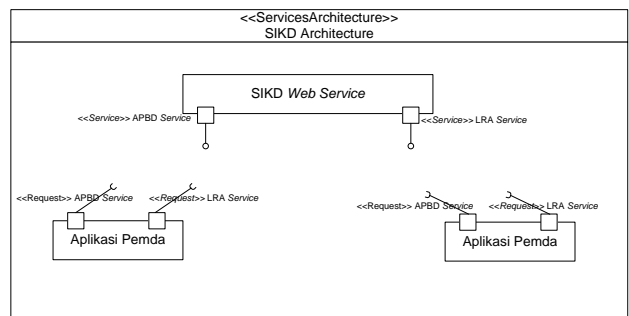


Gambar 11. UML Data Model Profile SIKD

4.8 Perancangan Arsitektur Teknologi

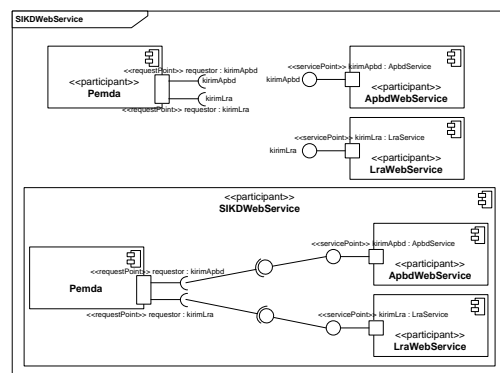
Dalam perancangan arsitektur teknologi yang akan dipergunakan pada SIKD, diperlukan sebuah teknologi yang mampu menjembatani komunikasi dan pertukaran data serta informasi antara pemerintah daerah dan

Direktorat EPIKD. Mengingat sangat luasnya wilayah Negara Kesatuan Republik Indonesia dan jumlah pemerintah daerah yang berjumlah 542 pemda dengan masing-masing memiliki data, aplikasi, dan teknologi yang bermacam-macam pula maka pemilihan teknologi ini adalah hal krusial yang harus secara hati-hati dipertimbangkan dan dipelajari. Salah satu teknologi yang memiliki kemampuan untuk menjembatani hal tersebut adalah web service. SOAP sebagai protokol yang dipergunakan dengan menggunakan XML sebagai dasar pertukaran informasi merupakan pilihan yang tepat untuk mengatasi heterogenitas aplikasi, luas wilayah, waktu penyampaian data dan besarnya data yang harus dikirimkan. Arsitektur web service SIKD dapat dilihat pada gambar 12.



Gambar 12. Arsitektur Web Service SIKD

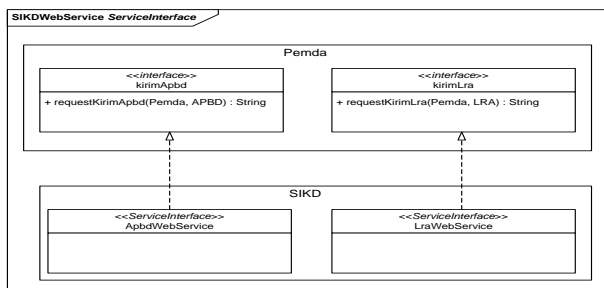
Gambar 12, menjelaskan arsitektur web service SIKD terdiri dari dua buah service yang disediakan di SIKD, yaitu APBD Service dan LRA Service. Setiap service yang ada pada SIKD selalu terbuka dan siap menerima permintaan (request) service dari pemerintah daerah. Setiap aplikasi atau sistem di pemerintah daerah harus memiliki aplikasi client yang mampu mengirimkan permintaan service kepada service-service yang ada pada SIKD. Aplikasi pada pemerintah daerah yang berfungsi untuk berkomunikasi dengan service server dinamakan sebagai serviceclient.



Gambar 13. Diagram Struktur Internal SIKDWebService

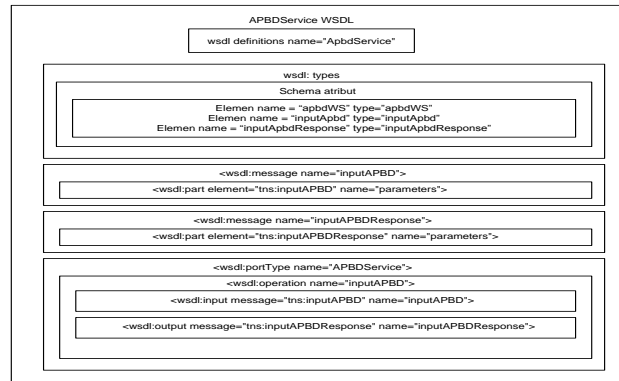
Gambar 13 menjelaskan struktur internal dari SIKDWebService yang digambarkan dengan menggunakan SoaML internal structure diagram. SIKDWebService memiliki tiga peserta di dalam prosesnya, yaitu Pemda sebagai pengirim data IKD, ApbdWebService yang menyediakan layanan untuk pengiriman APBD, dan LraWebService yang

menyediakan layanan untuk pengiriman LRA. Pada Pemda terdapat dua buah *requestPoint*, yaitu *requestPoint* kirimApbd dan *requestPoint* kirimLra. Dengan adanya *requestPoint* maka menandakan bahwa Pemda berlaku sebagai pihak yang melakukan permintaan layanan. *RequestPoint* kirimApbd berarti Pemda berlaku sebagai pihak yang melakukan permintaan layanan kirimApbd kepada sebuah penyedia layanan, dalam hal ini adalah ApbdWebService yang memiliki *servicePoint* kirimApbd sebagai penyedia layanan kirimApbd. *RequestPoint* kirimLra berarti Pemda berlaku sebagai pihak yang melakukan permintaan layanan kirimLra kepada sebuah penyedia layanan, dalam hal ini adalah LraWebService yang memiliki *servicePoint* kirimLra sebagai penyedia layanan kirimLra. Dengan adanya *requestPoint* dan *servicePoint* yang sama, maka antara Pemda dan ApbdWebService serta LraWebService dapat saling berkomunikasi untuk pengiriman data. Pemda dapat mengirim data IKD APBD dengan menggunakan layanan kirimApbd yang disediakan ApbdWebService, dan Pemda dapat mengirim data IKD LRA dengan menggunakan layanan kirimLra yang disediakan LraWebService. Untuk lebih jelasnya dapat dilihat pada gambar 14 mengenai *interface* yang disediakan oleh SIKD dalam melayani permintaan kirimApbd dan kirimLra dari pemda.



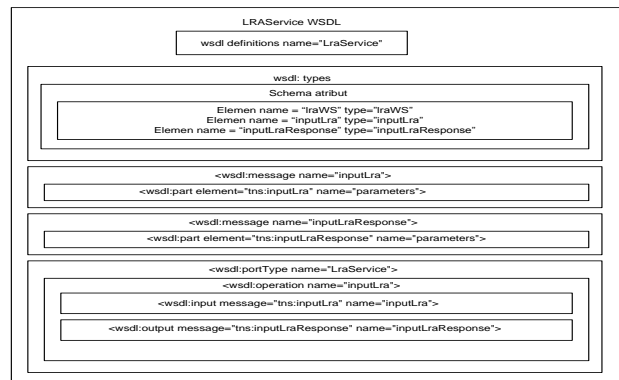
Gambar 14. Diagram *Service Interface* SIKDWebService

APBD *service* pada SIKD berfungsi untuk menerima kiriman data APBD dari aplikasi *client* di pemerintah daerah. Format data XML untuk APBD yang dikirimkan ke *service server* di SIKD memiliki format yang sudah ditentukan sesuai dengan perancangan struktur data pada diagram *class*. Format data untuk XML dinamakan sebagai *Web Service Definition Language* (WSDL). Gambar 15 berisi struktur WSDL untuk format APBD yang dapat dipergunakan sebagai acuan untuk melakukan pengiriman data APBD kepada SIKD.



Gambar 15. Struktur WSDL APbdService

Skema pembuatan pesan SOAP dapat menggunakan WSDL sebagai dasar pembuatan pesan pada pengiriman data dengan SOAP *Envelope* melalui *web service*. WSDL dibuat berdasarkan *class-class* dan *method* yang terdapat dan dipergunakan di dalam suatu *web service*. Struktur dari WSDL sendiri menggunakan format XML sehingga akan lebih mudah dipahami oleh berbagai bahasa pemrograman dan teknologi pada saat ini, baik *web base*, *desktop*, maupun *mobile*. Gambar 16 berisi struktur WSDL untuk skema pembuatan pesan LRA yang akan dikirimkan.

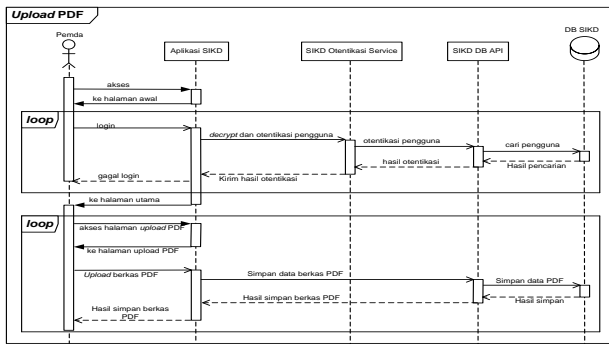


Gambar 16. Struktur WSDL LraService

4.9 Perancangan Arsitektur Interaksi Sistem dengan Pengguna

Salah satu diagram UML akan dipergunakan dalam penggambaran interaksi di dalam SIKD dengan pengguna SIKD. Untuk perancangan arsitektur interaksi ini hanya dibatasi pada penggambaran interaksi dalam pengiriman data dari pemerintah daerah ke Direktorat EPIKD melalui SIKD. Pengiriman data dari pemerintah daerah ke Direktorat EPIKD merupakan permasalahan yang selama ini masih menjadi kendala utama di dalam Komandan SIKD. Oleh karena itu sangatlah penting untuk mengutamakan rancangan arsitektur interaksi pemda dengan SIKD pada saat pengiriman data, baik berupa berkas PDF maupun data IKD.

Pada gambar 17 dijelaskan mengenai interaksi antara pemda dengan SIKD dalam proses upload berkas PDF, dimana pemda berlaku sebagai pengirim berkas PDF, dan SIKD berlaku sebagai penerima berkas PDF.

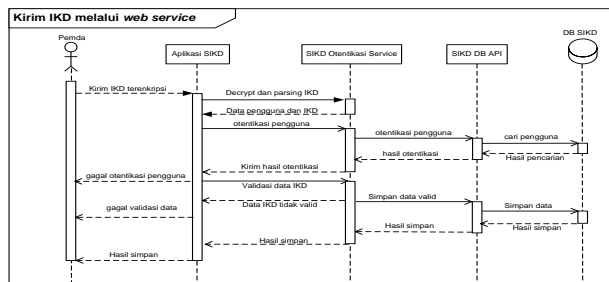


Gambar 17. Diagram Sequence Upload PDF

Seperti telah diuraikan pada sub bab sebelumnya bahwa pemerintah daerah dapat mengirimkan IKD ke Direktorat EPIKD melalui SIKD memiliki tiga buah cara yang disediakan oleh SIKD, yaitu:

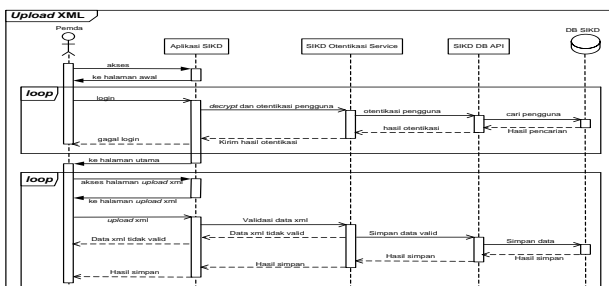
1. Melalui fasilitas *web service* (SOAP).
2. Melalui fasilitas *upload XML*.
3. Melalui fasilitas input data secara langsung ke SIKD.

Setiap fasilitas pengiriman data yang disediakan SIKD dapat dipergunakan pemerintah dengan kondisi yang berbeda-beda, sehingga memiliki interaksi yang berbeda-beda antara pemda sebagai pengirim dan di dalam SIKD sebagai penerima data. Pada gambar 18 terdapat gambar interaksi antara pemerintah daerah dengan SIKD pada saat pengiriman data IKD melalui *web service* yang akan digambarkan dengan diagram *sequence*.



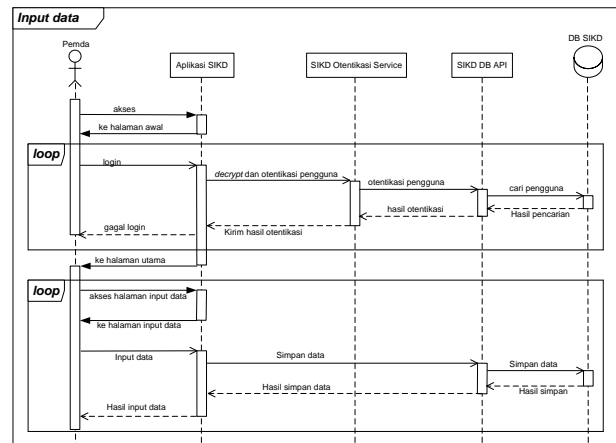
Gambar 18. Diagram Sequence Pengiriman Data Melalui Web Service

Pada gambar 19 terdapat gambar interaksi antara pemerintah daerah dengan SIKD pada saat pengiriman data IKD melalui fasilitas *upload XML*.



Gambar 19 diagram sequence pengiriman data melalui upload XML

Pada 20 terdapat gambar interaksi antara pemerintah daerah dengan SIKD pada saat pengiriman data IKD melalui fasilitas input data.



Gambar 20. Diagram Sequence Input Data

V. Kesimpulan

Analisis dan desain yang dilakukan bahwa, proses bisnis SIKD yang berjalan pada saat ini dengan menggunakan Komandan SIKD masih belum mampu melayani secara maksimal kebutuhan pengguna dalam menangani penerimaan data informasi keuangan daerah dari pemerintah daerah dengan berbagai aplikasi, sistem, platform sistem operasi, dan bahasa pemrograman yang dipergunakannya. Penggunaan *csv* yang tidak terstruktur sebagai format berkas pengiriman data kurang mampu menangani struktur data keuangan daerah yang rumit dan besar. Penggunaan teknologi *web service* dengan menggunakan SOAP mampu melayani komunikasi data secara netral tidak tergantung kepada platform sistem operasi maupun bahasa pemrograman apapun. Dengan demikian semua aplikasi keuangan daerah yang saat ini dipakai di berbagai pemerintah daerah seharusnya dapat mengirimkan data Informasi Keuangan Daerah dengan cara menambahkan modul SIKD Client pada aplikasi keuangan daerah dengan berbasis teknologi *web service*. Perancangan arsitektur SIKD secara detail sangat dibutuhkan agar pengembangan SIKD dapat tercapai secara baik dan sesuai dengan kebutuhan pemerintah daerah untuk mengirimkan data informasi keuangan daerah dan kebutuhan Direktorat EPIKD dalam menerima dan mengelola seluruh data informasi keuangan daerah untuk mencapai transparansi fiskal yang diharapkan.

VI. REFERENSI

- [1] Peraturan Menteri Keuangan Nomor 46/PMK. 02/2006 tentang tata cara penyampaian informasi keuangan daerah
- [2] Peraturan Menteri Keuangan Nomor 4/PMK. 07/2011 tentang perubahan Peraturan Menteri Keuangan Nomor 46/PMK. 02/2006
- [3] Pressman, R. (2001). Software Engineering A Practitioner's Approach Fifth Edition. New York: McGraw-Hill. ISBN 0-07-365578-3.
- [4] Priestley, M. (2003). Practical Object Oriented Design With UML: Second Edition. New York: McGraw-Hill Education. ISBN 0077103939.

- [5] Arlington, C, T. (2001). Enterprise Java™ with UML™. Canada: John Wiley & Sons Inc.
- [6] OMG. (2009). Service Oriented Architecture Modeling Language (SoaML) – *Specification For The UML Profile And Metamodel For Services (UPMS)*. FTF Beta 2.