

Model Integrasi Sistem dengan Pendekatan Metode *Service Oriented Architecture* dan *Model View Controller* pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia

Warkim^{#1}, Dana Indra Sensuse^{*2}

[#]Program Studi Magister Ilmu Komputer, Program Pascasarjana, Universitas Budi Luhur

Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

Telp. (021) 5853753, Fax. (021) 5869225

¹x1syah@icloud.com

^{*}Fakultas Ilmu Komputer, Universitas Indonesia
Kampus UI Depok, Kota Depok, Jawa Barat 16424

Telp. (021) 7863419, Fax. (021) 7863415

²dana@cs.ui.ac.id

Abstract -- The administration management as an activity and institute's main task which is also called as Back Office System, supported by several information systems. The data need and information in one information system cannot be fulfilled only by one information source, but it needs a composition from a two or more sources in one institute/organization. To solve this problem, we need an architecture model which can solve integration system between different information systems. This Integrated Information System Design Research Method uses Service Oriented Architecture (SOA) approach as the architecture base, Model View Controller (MVC) method as the model in the programming (coding). The applied development method SOA utilizes Service Oriented Modelling and Architecture (SOMA) development system, it is a system design method which classifies business process in to a service group. The SOA application is because of its loosely coupled, highly interoperable, reusable and interoperability characteristics cause SOA reliable in information development and integration. While the integrated information system which is built using MVC method, easier to be maintained and developed. This research produce employment service, asset service, supplies service, financial service and also system prototype as a dashboard for employment service with SOA approach and MVC method which uses Representational State Transfer (REST) technology.

Key Word: Integration Information System, Service Oriented Architecture, Model View Controller, Representational State Transfer, Service Oriented Modelling and Architecture

I. PENDAHULUAN

Perkembangan internet secara signifikan telah membawa pemanfaatan Teknologi Informasi dan Komunikasi (TIK) ke tingkatan yang lebih tinggi. Melalui jaringan internet, basis

data dan aplikasi bahkan sistem informasi terhubung satu sama lainnya membentuk jaringan yang jauh lebih kompleks. Namun demikian pemanfaatan yang optimal dari TIK belum sepenuhnya dapat tercapai. Salah satu penyebab pemanfaatan TIK belum maksimal adalah basis data dan aplikasi dibangun dengan menggunakan *platform* sistem informasi dan data yang berbeda-beda. Akibatnya satu basis data atau sistem informasi belum tentu dapat saling berkomunikasi untuk melayani suatu kegiatan yang sifatnya terpadu [1].

Service Oriented Architecture (SOA) dipilih dikarenakan SOA merupakan metode terbaik dalam konteks integrasi sistem [2]. Sedangkan Metode *Model View Controller* (MVC) dipilih dikarenakan perangkat lunak yang dibangun dengan menggunakan metode MVC membagi tanggung jawab menjadi tiga peran utama yang memungkinkan untuk kolaborasi agar lebih efisien sehingga akan lebih mudah dipelihara dan dikembangkan [3]. Penelitian ini menitikberatkan pada arsitektur SOA dan metode MVC dengan mempertimbangkan permasalahan pengintegrasian aplikasi terkait pengelolaan administrasi pada bagian tata usaha.

Berkaitan erat dengan permasalahan maka dilakukan penelitian membuat model integrasi sistem antar aplikasi dengan menggunakan pendekatan SOA, mengidentifikasi layanan-layanan yang diperlukan oleh bagian tata usaha agar sistem informasi yang digunakan dapat saling berkomunikasi serta membuat prototipe sistem terintegrasi dengan menggunakan metode MVC yang bertujuan untuk menjamin integrasi data dan menurunkan ambiguitas informasi. Penelitian ini diharapkan dapat mengurangi *inventory* dan sumberdaya manusia, meningkatkan produktifitas kerja, pengurangan biaya dan pemeliharaan teknologi informasi,

visibility dan transparansi informasi, serta dapat meningkatkan terciptanya proses baru dan kinerja pegawai pada bagian tata usaha Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia (PAPPIPTEK LIPI).

II. LANDASAN TEORI

Integrasi Sistem informasi yang baik harus melalui proses dengan perencanaan yang matang sesuai dengan kaidah-kaidah yang jelas sehingga dapat menghasilkan sistem yang sesuai dengan kebutuhan dan teknologi yang diterapkan oleh suatu organisasi. Penelitian integrasi sistem dengan pendekatan SOA dan metode MVC ini dilandasi dari beberapa penelitian yang sudah dilakukan sebelumnya.

Andika Agus Slameto (2015), dalam penelitiannya menerapkan integrasi sistem informasi inventaris dengan sistem informasi pelaporan kerusakan laboratorium dengan menggunakan SOA [4]. Mardiana dan Keijiro Araki (2012), dalam penelitiannya mengintegrasikan sistem informasi akademik dengan sistem informasi antar muka berbasis web dengan menggunakan metode *Model Driven Approach* (MDA) dan SOA, untuk kegiatan pembelajaran memakai aplikasi MOODLE (*Modular Object-Oriented Dynamic Learning*) [5]. Khalifa Mansouri, Bouchaib Riyami, Mohamed Youssfi dan Omar Bouattane (2016), dalam penelitiannya menerapkan *Model Driven Engineering* (MDE) dan *Model Driven Architecture* (MDA) dengan mengintegrasikan layanan-layanan yang tersedia pada perusahaan. Metodologi yang dikembangkan menggunakan metode SOA dengan mengusulkan *layer middleware* yang terdiri dari *Business Process Execution Language* (BPEL) dan *Atlas Transformation Language* (ATL) sebagai arsitektur untuk membangun interoperabilitas antar layanan di perusahaan yang merupakan salah satu bentuk transformasi, integrasi, homogenisasi dan adaptasi layanan [6]. Adi Nugroho dan Khabib Mustofa (2012), dalam penelitiannya membandingkan teknologi *web service* antara *java web service* berbasis SOAP dengan RESTful yang diimplementasikan dalam aplikasi Sistem Informasi Geografis terintegrasi dengan menggunakan format GML yang tersimpan pada basisdata dalam formal XML [7]. Chhikara, J. (2014), dalam penelitiannya menerapkan *framework* berbasis MVC sebagai pola desain dalam pengembangan aplikasi menggunakan HTML5 [8].

A. Integrasi Sistem

Integrasi sistem adalah suatu konsep sistem yang saling berhubungan antara sistem yang satu dengan sistem yang lainnya disesuaikan dengan keperluan. Hal ini sangat bermanfaat untuk keberlanjutan dari suatu sistem informasi yang diperlukan juga oleh sistem yang lainnya atau output suatu sistem menjadi input sistem yang lainnya. Dalam konteks sistem informasi, sistem terintegrasi (*integrated system*) merupakan sebuah rangkaian proses untuk menghubungkan beberapa sistem informasi maupun aplikasi baik secara fisik maupun secara fungsional [9]. Sistem terintegrasi akan menggabungkan komponen sub-sistem

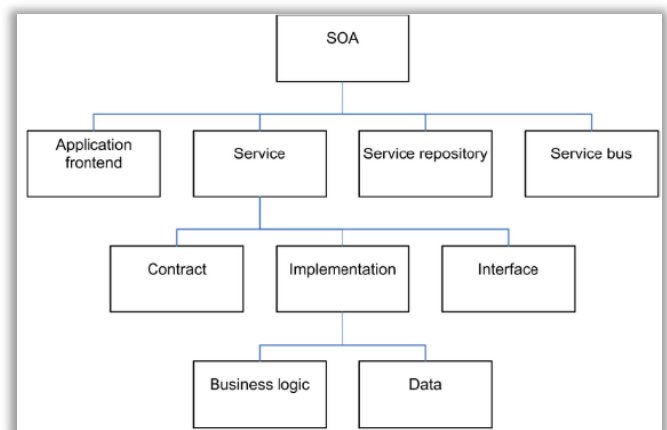
ke dalam satu sistem informasi yang menjamin fungsi-fungsi dari sub sistem tersebut sebagai satu kesatuan sistem.

B. Service Oriented Architecture

SOA adalah sebuah arsitektur yang bersifat *service oriented*, yaitu arsitektur yang membagi suatu masalah kedalam berbagai *services* kecil yang saling bekerja sama [10]. Dengan menggunakan SOA suatu aplikasi tidak lagi dipandang dari sisi teknologi, data, lingkungan implementasi melainkan dipandang dari *services* yang disediakan. Dengan cara ini suatu aplikasi dapat berkomunikasi dan bekerja sama dengan aplikasi yang lain tanpa memandang teknologi, data, lingkungan implementasi dari aplikasi tersebut. Menurut Pungus R.S mendefinisikan SOA adalah sebuah kerangka kerja untuk mengintegrasikan proses bisnis dan mendukung infrastruktur teknologi informasi dan menstandarisasi komponen-komponen layanan yang dapat digunakan kembali dan digabungkan sesuai dengan prioritas bisnis. SOA bersifat *loosely coupled* (tingkat kebergantungan antar komponen rendah), *highly interoperable* (mudah dioperasikan), *reusable* (dapat digunakan kembali) dan *interoperability* (dapat berkomunikasi antar platform) [11].

SOA menyediakan cara untuk merencanakan, mendesain dan menyampaikan fungsionalitas teknologi informasi sebagai layanan bisnis yang modular agar bisa memenuhi beberapa persyaratan bisnis yang spesifik [12]. Sebagai arsitektur perangkat lunak yang fungsionalitasnya dikelompokkan menjadi proses bisnis dan dikemas sebagai *interoperable service* atau dapat menggunakan lintas platform, SOA mampu mendeskripsikan bagaimana infrastruktur teknologi informasi dapat membantu pertukaran data/informasi dari aplikasi-aplikasi yang berbeda guna mendukung suatu proses bisnis yang diinginkan [13].

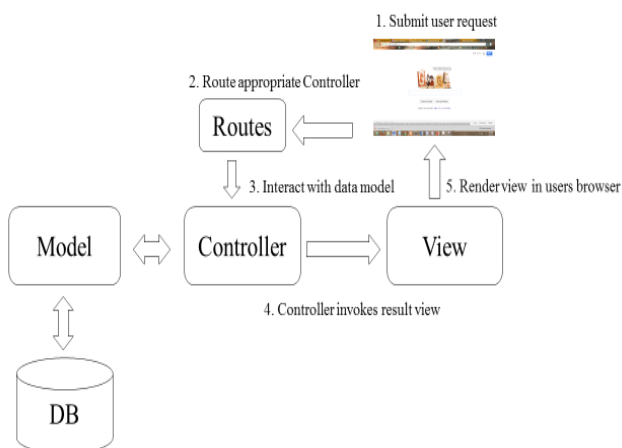
Gambar 1. menjelaskan struktur hirarki dari SOA, yaitu sebuah arsitektur perangkat lunak yang didasarkan pada konsep-konsep kunci dari sebuah aplikasi front-end, *service*, *service repository* dan *service bus*. Sebuah *service* terdiri dari sebuah *contract*, satu atau lebih antarmuka, dan sebuah implementasi yang mencakup data dan *business logic* [14].



Gambar 1: Struktur Hirarki SOA

C. Model View Controller

MVC didefinisikan sebagai arsitektur dalam pengembangan perangkat lunak yang memisahkan logika bisnis dari *input* dan presentasi logika yang terkait dengan tampilan antarmuka suatu aplikasi [15]. MVC mengikuti pendekatan yang paling umum dari *layering* yaitu sebuah logika yang membagi kode ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode.



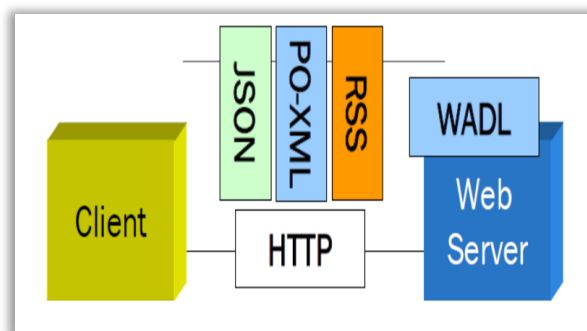
Gambar 2: Hubungan antara Model, View dan Controller

D. REST Web Service

Cara kerjanya, REST server menyediakan jalur untuk akses *resource* atau data, sedangkan REST client melakukan akses *resource* dan kemudian menampilkan atau menggunakannya. *Resource* yang dihasilkan sebenarnya berupa teks, namun formatnya bisa bermacam-macam tergantung keinginan *developer*, umumnya adalah JSON dan XML.

Dalam mengakses sebuah *resource*, REST juga menggunakan konsep URI dimana ada *method* yang digunakan adalah GET. Berikut ini *method-method* yang mendukung REST:

- 1) GET, cocok untuk *resource* yang hanya perlu dibaca saja (*read only*).
- 2) PUT, cocok digunakan untuk membuat/*create resource* baru.
- 3) DELETE, cocok digunakan untuk menghapus suatu *resource*.
- 4) POST, cocok digunakan untuk mengupdate suatu *resource*.
- 5) OPTIONS, cocok digunakan untuk mendapatkan operasi yang di *support* pada *resource*.



Gambar 3: Skema Client-Server REST Web Services

Gambar 3. merupakan ilustrasi REST Web Service jika dilihat dari sudut pandang *client-server*. Konsep yang penting dari REST adalah adanya *resource*, yang mana tiap *resource* yang ada diidentifikasi dengan suatu tanda pengenal. Untuk memanipulasi *resource* ini, representasi dari *resource* dikirimkan antara *client* dan *server*. Sehingga *resource* yang ada tidak secara langsung dimanipulasi oleh aplikasi yang ada, sebagai contoh: basis data di sisi *server* tidak dikirimkan ke *client*, yang dikirimkan adalah representasi dari data di basis data tersebut dalam format tertentu seperti XML [15].

III. METODOLOGI

Metode penelitian yang dilakukan menggunakan metode wawancara, dokumentasi dan *participatory observation*. Wawancara merupakan suatu cara dalam pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Dalam metode wawancara terdapat beberapa faktor yang akan mempengaruhi arus informasi dalam wawancara, yaitu: pewawancara, responden, pedoman wawancara, dan situasi wawancara. Sedangkan metode *participatory observation* adalah metode penelitian dimana penulis atau peneliti tersebut ikut serta dalam kegiatan yang berlangsung didalam suatu organisasi.

Metode pengembangan SOA yang digunakan dalam penelitian ini adalah metode pengembangan *Service Oriented Modelling and Architecture (SOMA)*. Metode SOA ini telah diadaptasi dari banyaknya pengalaman yang dipelajari dari kesulitan dan tantangan yang dihadapi oleh desain awal implementasi proyek SOA. Kunci utama dari tujuan penggunaan pengalaman ini menjadikannya sebagai suatu set peran tugas yang digunakan sebagai panduan dan praktis terbaik dari bidang yang bersangkutan menggunakan orientasi *service*. Metode ini mendefinisikan kunci dan menyediakan tugas perspektif dan mendeskripsikan panduan normatif untuk analisa, desain, implementasi, *testing*, dan pelaksanaan. Mulai dari *service*, komponen, alur, informasi, dan kebijakan dibutuhkan untuk merancang dan membuat solusi SOA yang dapat digunakan kembali (*reusability*) di dalam organisasi. Dengan tujuan mencapai tujuan akhir bisnis dalam menyediakan *service* yang konsisten ketika membuat *service* baru.

Instrumen penelitian yang digunakan untuk pengamatan obyek penelitian yaitu menggunakan metode wawancara. Wawancara dilakukan kepada manajemen terkait integrasi sistem pada bagian Tata Usaha. Sedangkan instrumen untuk melakukan pengujian sistem dilakukan dengan metode *Focus Discussion Group* dengan memberikan kuesioner kepada pengguna dalam hal ini pengembang aplikasi.

IV. PEMBAHASAN HASIL PENELITIAN

A. Analisis Masalah

Analisis masalah dalam pembahasan penelitian ini dibuat untuk memenuhi kebutuhan pertukaran data/informasi dan integrasi antar aplikasi yang berbeda *platform*, sehingga aplikasi atau sistem informasi yang ada di satuan kerja PAPPITPEK-LIPI dapat terintegrasi dan saling berkomunikasi. Lingkup analisis masalah yang akan dituangkan dalam bentuk objektif-objektif yang harus dicapai dalam pertukaran data dan integrasi antar aplikasi yang terdapat di satuan kerja PAPPITPEK-LIPI.

B. Analisis Kebutuhan Pengguna

Analisis kebutuhan yang memiliki keterkaitan langsung dengan sistem informasi yang akan dibangun. Analisis kebutuhan dalam membangun integrasi sistem ini diperoleh dari hasil wawancara dengan pemangku kepentingan yaitu pengelola sistem informasi, pegawai dan manajemen.

C. Analisis Service Oriented

Analisis *Service Oriented* merupakan tahapan untuk menganalisis permasalahan dan mengidentifikasi *service* apa saja yang akan dibangun dan logik apa saja yang akan dienkapsulasi. Tabel I. menggambarkan kegiatan yang dalam pengerjaannya menggunakan aplikasi/sistem informasi. Penggunaan sistem informasi tersebut selain terdapat di bagian tata usaha PAPPITPEK LIPI.

TABEL I.
PENJABARAN FUNGSI-FUNGSI DENGAN PROSES KERJA

Fungsi	Proses Kerja
Pengelolaan Kepegawaian	<ul style="list-style-type: none"> - Pelaksanaan analisis dan kebutuhan pegawai - Pelaksanaan urusan mutasi pegawai - Pelaksanaan urusan administrasi jabatan fungsional peneliti dan non peneliti - Penyusunan administrasi pengembangan sumber daya manusia - Pelaksanaan administrasi kehadiran pegawai (ijin, dinas, cuti dan jumlah jam kerja)
Pengelolaan Administrasi Umum	<ul style="list-style-type: none"> - Pelaksanaan urusan pengadaan barang milik negara - Pelaksanaan distribusi barang milik negara - Pelaksanaan inventarisasi barang milik negara - Pelaksanaan pencatatan barang persediaan

Fungsi	Proses Kerja
	<ul style="list-style-type: none"> - Pelaksanaan pendistribusian persediaan - Pelaksanaan pencatatan peminjaman barang milik negara
Pengelolaan Keuangan	<ul style="list-style-type: none"> - Penyusunan revisi anggaran - Penyusunan laporan pertanggungjawaban keuangan - Penyusunan realisasi anggaran - Penyusunan pembayaran gaji dan tunjangan jabatan - Penerbitan Surat Perintah Membayar (SPM) - Penerbitan Surat Perintah Pembayaran (SPP) - Penyusunan Surat Perintah Perjalanan Dinas (SPPD)

Berdasarkan fungsi-fungsi dan proses kerja yang terdapat dibagian Tata Usaha, perlu adanya dukungan suatu aplikasi yang dapat melancarkan tugas dan fungsi suatu instansi/lembaga. Hasil identifikasi sistem informasi yang berjalan terkait pengelolaan ketatausahaan terdiri dari beberapa aplikasi *desktop* dari Kementerian Keuangan dan aplikasi internal dengan berbasis web.

TABEL II.
SISTEM INFORMASI YANG BERJALAN DI BAGIAN TATA USAHA

No	Fungsi	Sistem Informasi
1	Pengelolaan Kehadiran Pegawai	SIAB (Sistem Informasi Absensi Pegawai)
2	Pengelolaan Kepegawaian	SIMPEG (Sistem Informasi Kepegawaian LIPI)
3	Pengelolaan Absensi	Attendance Management Program (AMP)
4	Pengelolaan Manajemen Aset	SIMA (Sistem Informasi Manajemen Aset)
5	Pengelolaan Barang Milik Negara	SIMAK-BMN (Sistem Informasi Manajemen dan Akuntansi Barang Milik Negara)
6	Pengelolaan Akuntansi Persediaan	ASAP (Aplikasi Sistem Akuntansi Persediaan)
7	Pengelolaan Persediaan	SIDIA (Sistem Informasi Persediaan)
8	Pengelolaan Anggaran	SIPA (Sistem Informasi Pengelolaan Anggaran)
9	Pengelolaan Anggaran	RKAKL (Rencana Kerja dan Anggaran Kementerian Negara/Lembaga)

Tabel II. merupakan sistem informasi/aplikasi yang digunakan dalam pengelolaan administrasi yang ada di bagian tata usaha. Pengelolaan administrasi tersebut terdiri dari sub bagian keuangan dan sub bagian kepegawaian & umum. Aplikasi yang digunakan pada sub bagian keuangan meliputi aplikasi SIPA dan RKAKL. Sedangkan aplikasi

yang digunakan pada sub bagian kepegawaian dan umum meliputi aplikasi SIAB, SIMPEG, AMP, SIMA, SIMAK BMN, ASAP dan SIDIA.

Hubungan keterkaitan sistem informasi yang berjalan dengan proses kerja menggambarkan bagaimana dukungan aplikasi/sistem informasi yang ada terhadap fungsi kerja pada bagian Tata Usaha PAPPITTEK-LIPI. Dari hasil pemetaan fungsi dan proses kerja terhadap aplikasi yang ada, didapatkan beberapa kandidat layanan-layanan yang terdapat di bagian Tata Usaha, kandidat layanan-layanan tersebut seperti yang terlihat pada Tabel III.

TABEL III.
MATRIK HUBUNGAN PROSES KERJA DENGAN APLIKASI BERJALAN YANG MEMBENTUK KANDIDAT LAYANAN

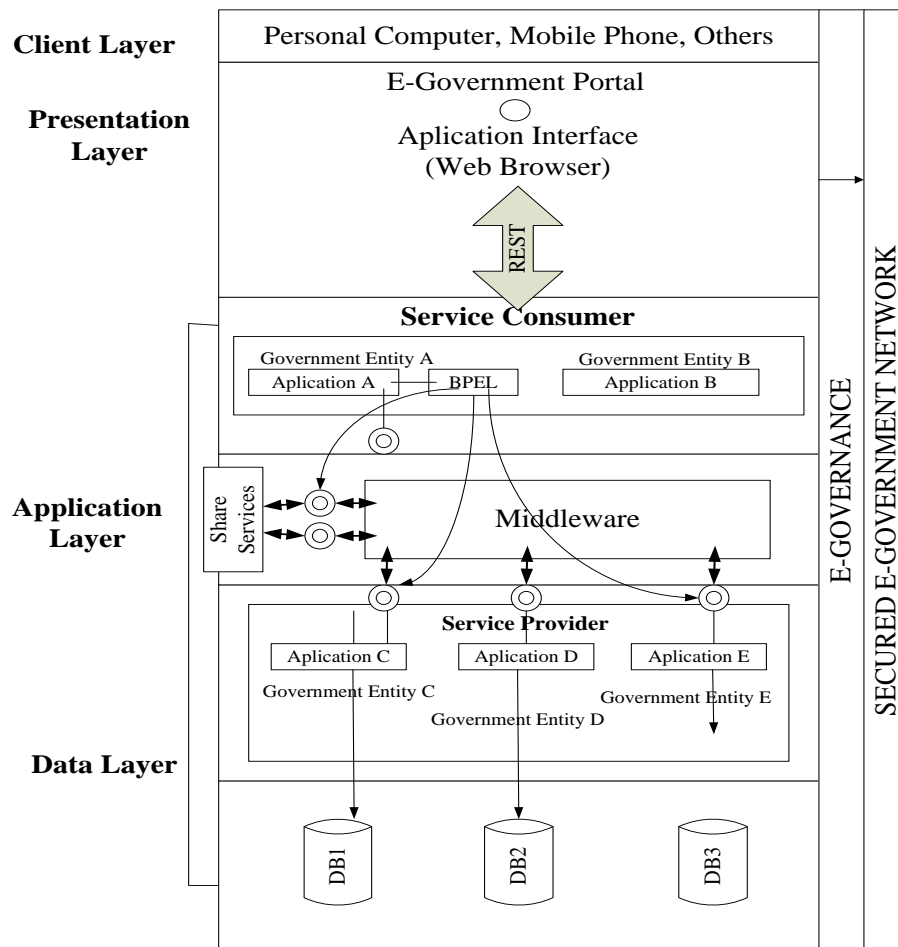
No	Proses Kerja	Aplikasi yang Ada	Kandidat Layanan
1	Pelaksanaan administrasi kehadiran pegawai	SIAB, SIMPEG, AMP, SIPA	Layanan Kepegawaian
2	Pelaksanaan urusan pencatatan barang milik negara	SIMAK BMN, SIMA	Layanan Aset
3	Pelaksanaan distribusi barang milik negara	SIMAK BMN, SIMA	
4	Pelaksanaan pencatatan peminjaman barang milik negara	SIMAK BMN, SIMA	
5	Pelaksanaan pencatatan barang persediaan	ASAP, SIDIA	Layanan Persediaan
6	Pelaksanaan pendistribusian persediaan	ASAP, SIDIA	

No	Proses Kerja	Aplikasi yang Ada	Kandidat Layanan
7	Penyusunan revisi anggaran	RKAKL	Layanan Keuangan
8	Penyusunan laporan pertanggungjawaban keuangan	RKAKL, SIPA	
9	Penyusunan realisasi anggaran	RKAKL, SIPA	

Tabel III. menjelaskan terdapat 9 (sembilan) proses kerja yang memiliki hubungan dengan aplikasi yang ada. Dari sembilan proses kerja tersebut dapat disimpulkan menjadi empat kandidat layanan-layanan yaitu layanan kepegawaian, layanan aset (barang milik negara), layanan persediaan, dan layanan keuangan.

D. Model Arsitektur Integrasi Sistem

Penyusunan model arsitektur sistem terintegrasi berdasarkan hasil dari analisis identifikasi kebutuhan layanan-layanan yang diprioritaskan oleh pengguna yang sudah dijelaskan sebelumnya. Model arsitektur sistem terintegrasi dengan pendekatan SOA yang di rancang meliputi beberapa aplikasi yang menjadi prioritas dari hasil analisis kebutuhan pengguna pada bagian tata usaha PAPPITTEK LIPI. Aplikasi yang menjadi prioritas tersebut meliputi aplikasi SIMPEG, AMP, SIMAK BMN, ASAP, RKAKL sebagai *service provider* sedangkan aplikasi SIAB, SIDIA, SIMA dan SIPA sebagai *service consumer* dalam arsitektur SOA disebut sebagai *application layer*. Model pengembangan arsitektur integrasi sistem dengan pendekatan SOA, seperti pada Gambar 4.

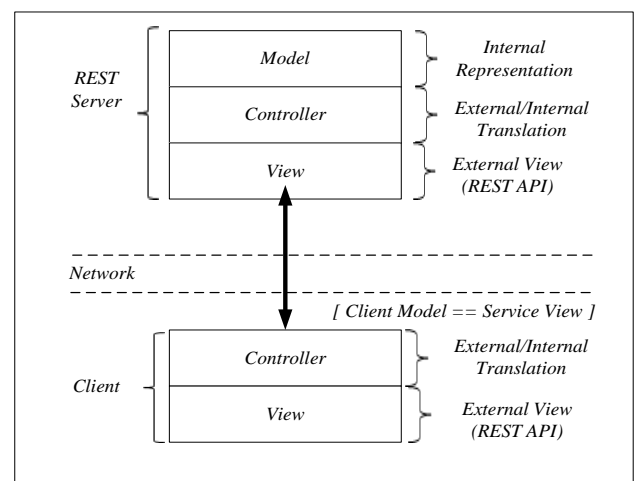


Gambar 4: Model Arsitektur Integrasi Sistem dengan Pendekatan SOA.

Model arsitektur sistem terintegrasi terhadap layanan-layanan yang terdapat pada masing-masing aplikasi akan dirancang dengan menggunakan *web service* dengan dukungan teknologi REST. Teknologi *web service* dengan menggunakan REST mampu menerapkan sebuah strategi yang mengintegrasikan dan mengkomunikasikan aplikasi-aplikasi yang ada didalam suatu instansi. Sisi sederhana yang membuat menarik dari teknologi REST adalah dapat dibangun dengan menggunakan sedikit *tools*. Selain itu untuk melakukan *testing* terhadap REST *services* dapat dilakukan secara sederhana dengan menggunakan *web browser* tanpa harus melakukan simulasi antara *client* dan *server*.

Arsitektur pengintegrasian berorientasi layanan membuat sebuah *level* dari antar muka layanan dapat direpresentasikan secara umum. Data, fungsi dan prosedur yang ada pada suatu aplikasi dapat dengan mudah untuk dibagikan atau digunakan oleh *user* yang membutuhkan dengan merepresentasikannya kedalam sebuah layanan (*web services*). Hal ini sangat mendukung terjadinya *interoperability* antar aplikasi yang terdapat pada suatu organisasi/instansi. Layanan-layanan yang telah dipilih kemudian dibangun *web service* dengan teknologi REST dan metode MVC. Metode MVC yang

digunakan untuk membangun *web service* yaitu dengan menggunakan *framework* laravel. Arsitektur REST dengan metode MVC seperti yang terlihat pada Gambar 5.



Gambar 5: Arsitektur REST dengan Metode MVC

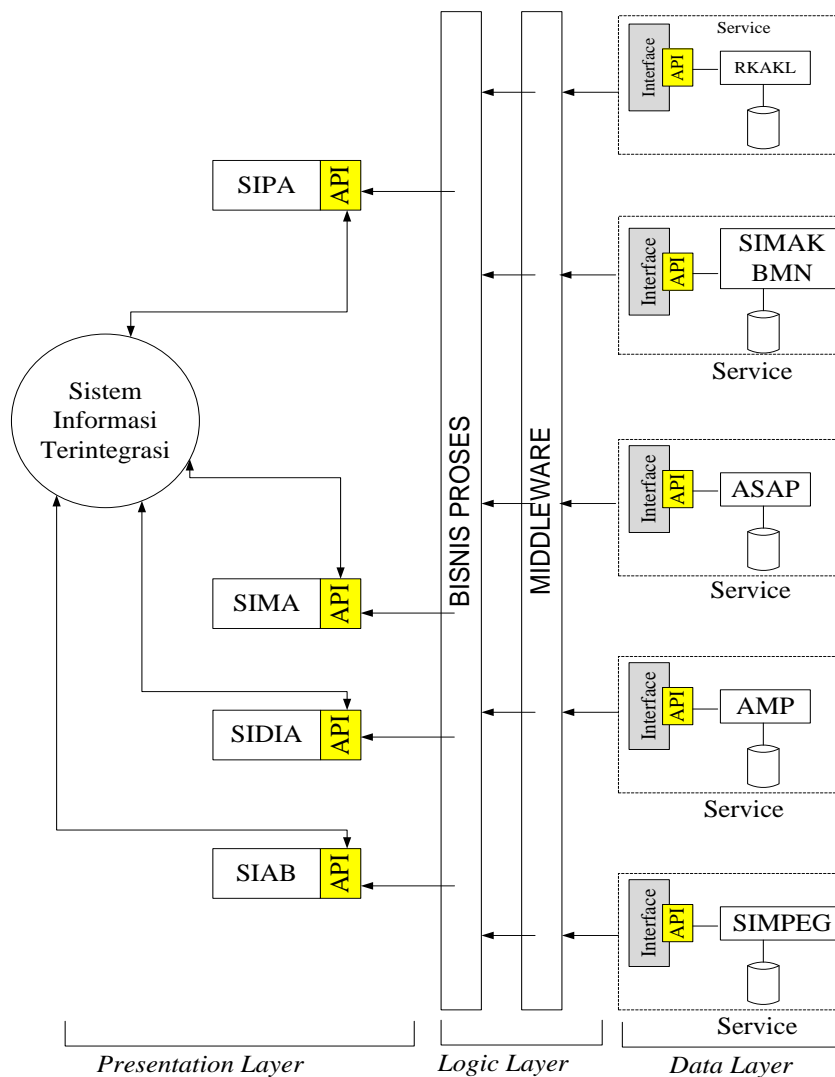
E. Perancangan Service

Teknologi *web service* menyediakan sebuah cara baru untuk mengizinkan suatu aplikasi yang telah ada dan aplikasi yang baru untuk melakukan transaksi (*interoperate*). Sehingga aplikasi yang baru dapat dikerjakan dengan cepat dan mudah menggabungkan antarmuka kedalam aplikasi-aplikasi yang akan dibangun tersebut. Perancangan *service* sistem terintegrasi yang dibangun berdasarkan hasil analisis model arsitektur SOA yang ditampilkan pada Gambar 6. yang menjelaskan terdapat beberapa aplikasi sebagai *data layer* yaitu RKAKL, SIMAK BMN, ASAP, AMP dan SIMPEG. Sedangkan aplikasi SIAB, SIDIA, SIMA dan SIPA merupakan aplikasi sebagai *presentation layer*.

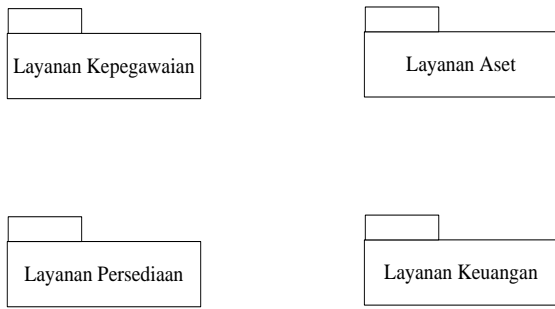
F. Perancangan Use Case Layanan

Perancangan *use case* akan memberikan gambaran secara umum aksi atau proses bisnis yang berlangsung serta pengguna yang terlibat dalam suatu layanan aplikasi. Penggunaan pemodelan *use case* memfasilitasi pengembang aplikasi dapat menemukan kebutuhan fungsional, membantu menggambarkan lingkup sistem menjadi bagian-bagian yang lebih mudah dimengerti dan dikelola, menyajikan spesifikasi fungsional untuk mendesain antarmuka pengguna dan aplikasi.

Berdasarkan dari matriks antara proses kerja dengan aplikasi yang ada tersebut dapat dikelompokkan kedalam empat layanan utama yang selanjutnya setiap layanan direpresentasikan kedalam *use case package*. Gambar 7. merepresentasikan layanan utama yang di representasikan kedalam *use case package* pada bagian tata usaha PAPPIPTEK LIPI.



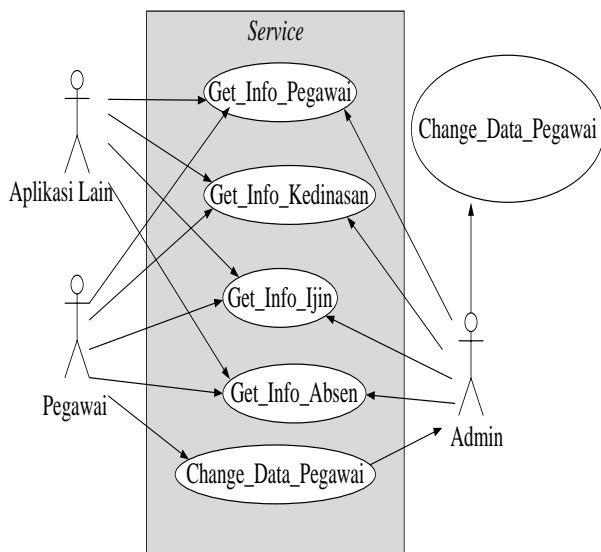
Gambar 6: Konsep Service Integrasi Sistem dengan Pendekatan SOA



Gambar 7: Use Case Package Layanan Utama Ketatausahaan

Identifikasi *service* dilakukan dengan pendekatan *entity-centric*, dikarenakan pendekatan ini akan menghasilkan perangkat lunak yang lebih modular. Oleh karena itu dilakukan analisis terhadap entitas-entitas apa saja yang terdapat didalam sistem beserta keterhubungannya. Hasil analisis pada layanan kepegawaian diperoleh entitas-entitas yang terkait untuk mengimplementasikan proses bisnis terkait kepegawaian, antara lain adalah:

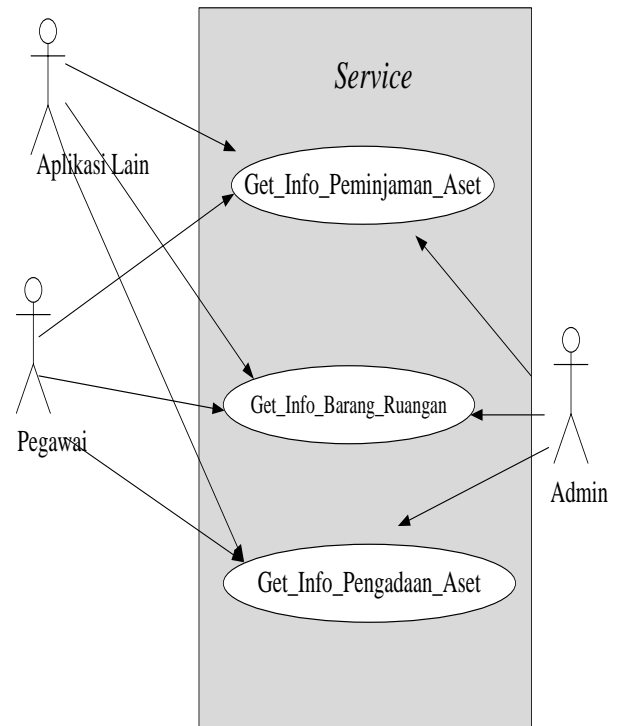
- 1) Entitas info pegawai, yang mewakili informasi pegawai dan segala propertinya.
- 2) Entitas info kedinasan, yang mewakili informasi kedinasan pegawai.
- 3) Entitas info ijin, yang mewakili informasi ijin pegawai seperti ijin tidak masuk kerja, terlambat masuk kerja, pulang sebelum waktunya dan sakit.
- 4) Entitas info absensi, yang mewakili informasi data absensi pegawai yang berasal dari *fingerprint*.



Gambar 8: Use Case Layanan Kepegawaian

Hasil analisis pada layanan aset merupakan entitas-entitas yang terkait untuk mengimplementasikan proses bisnis yang terkait dengan pengelolaan aset, antara lain adalah:

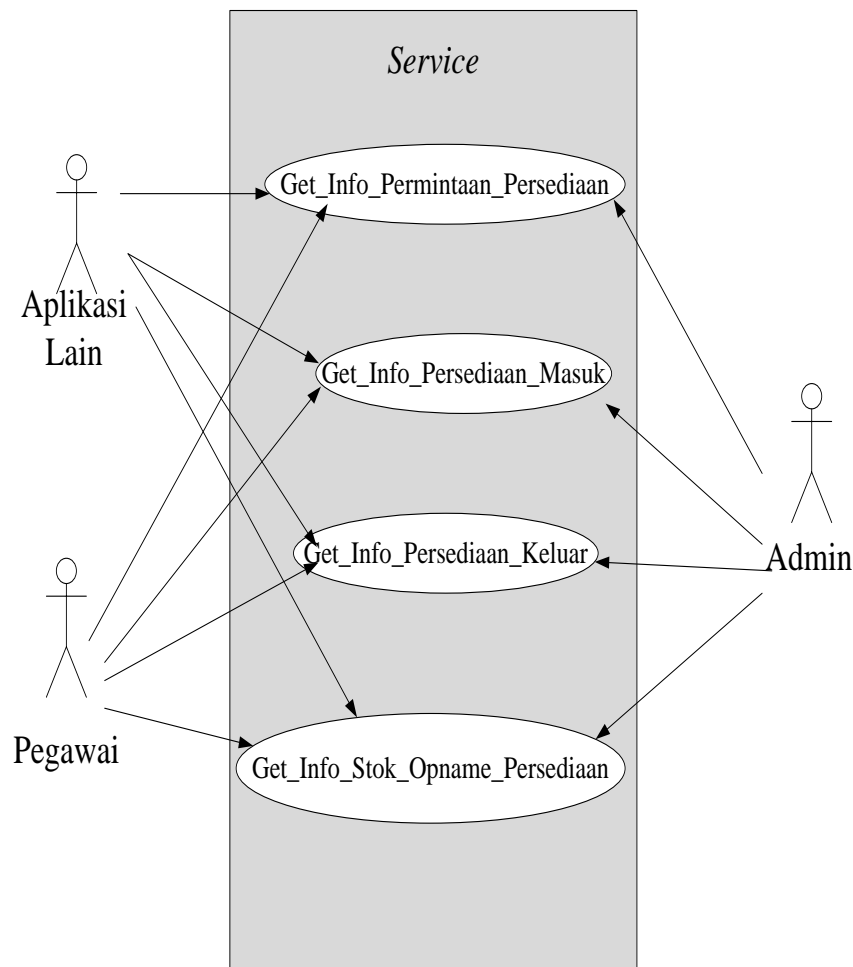
- 1) Entitas peminjaman aset, yang mewakili informasi peminjaman barang milik negara yang dipinjam/digunakan pegawai.
- 2) Entitas barang ruangan, yang mewakili informasi daftar barang per ruangan.
- 3) Entitas pengadaan aset, yang mewakili informasi pembelian barang milik negara.



Gambar 9: Use Case Layanan Aset

Hasil analisis pada layanan persediaan entitas-entitas yang terkait untuk mengimplementasikan proses bisnis yang terkait dengan pengelolaan persediaan, antara lain adalah:

- 1) Entitas permintaan persediaan, yang mewakili informasi permintaan persediaan dari pegawai.
- 2) Entitas persediaan masuk, yang mewakili informasi pembelian barang persediaan.
- 3) Entitas persediaan keluar, yang mewakili informasi pengeluaran barang persediaan.
- 4) Entitas stok opname persediaan, yang mewakili informasi stok barang persediaan.



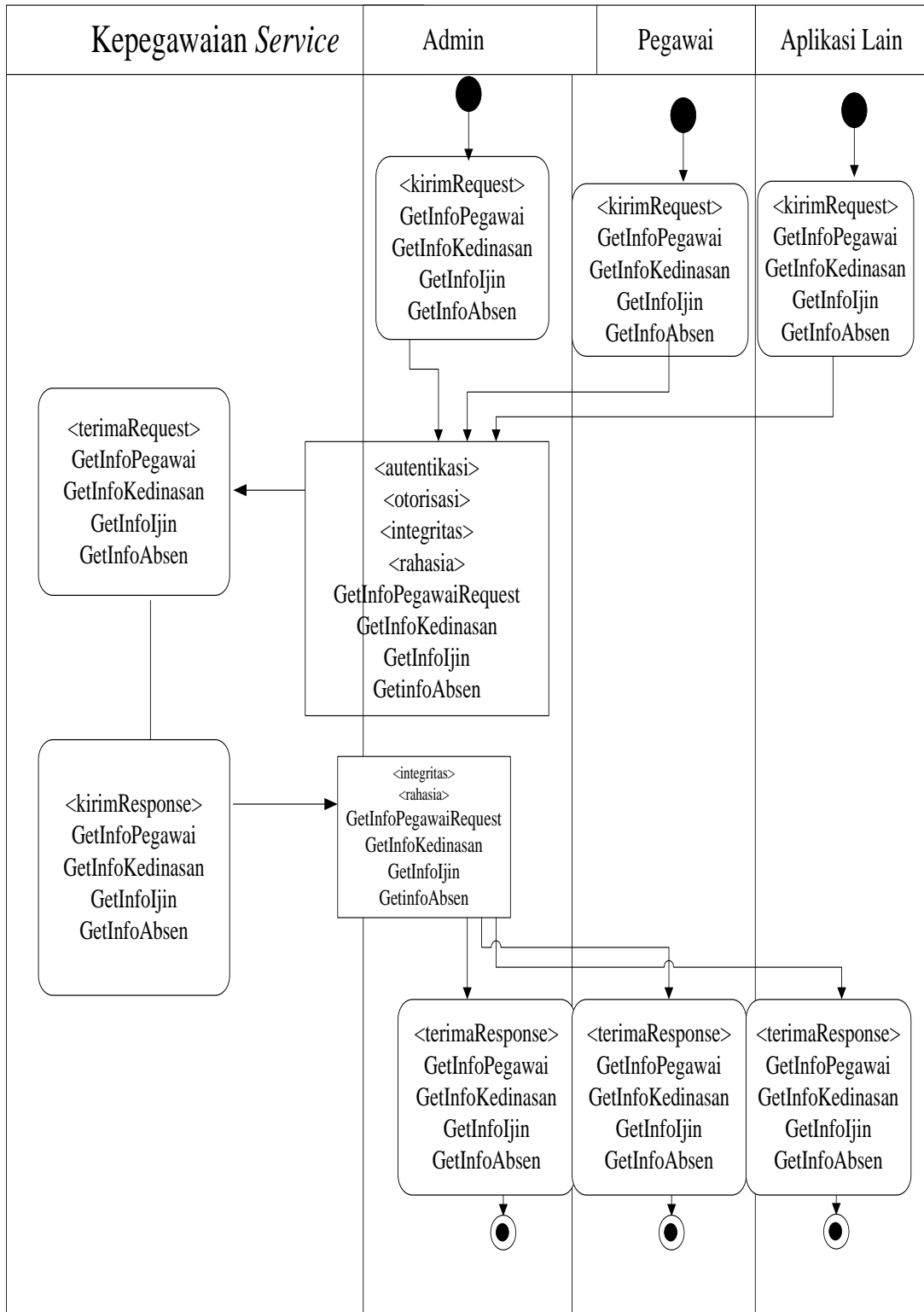
Gambar 10: Use Case Layanan Persediaan

Layanan keuangan yang menjadi prioritas dalam penelitian ini adalah penyusunan revisi anggaran, penyusunan realisasi anggaran penyusunan dan laporan pertanggungjawaban keuangan. Hasil analisis yang terdapat pada layanan keuangan diperoleh entitas-entitas yang terkait untuk mengimplementasikan proses bisnis dengan pengelolaan keuangan, antara lain:

- 1) Entitas revisi anggaran, yang mewakili informasi daftar anggaran kegiatan yang sudah di lakukan revisi.
- 2) Entitas realisasi anggaran, yang mewakili informasi realisasi anggaran dari kelompok kegiatan dari Rencana Anggaran Belanja (RAB) yang sudah dikeluarkan
- 3) Entitas pertanggungjawaban anggaran, yang mewakili informasi tentang laporan pertanggungjawaban anggaran yang sudah dikeluarkan.

G. Perancangan Activity Diagram Layanan

Rancangan *activity diagram* layanan kepegawaian lebih fokus menggambarkan urutan aktifitas sebuah proses dalam sistem terintegrasi yang dibangun. *Activity diagram* yang digambarkan berdasarkan dari rancangan *use case* yang sudah dijelaskan sebelumnya, terdiri dari empat buah *service* dan tiga buah aktor yaitu admin, pegawai, dan aplikasi lain. Aktor admin berperan sebagai pengelola untuk *maintenance* sistem terintegrasi yang berfungsi sebagai *dashboard*. Rancangan *activity diagram* layanan kepegawaian terdiri dari empat buah *service* yaitu *service* info pegawai, info kedinasan, info ijin dan info absensi yang dapat diakses oleh tiga buah aktor yaitu admin, pegawai, dan aplikasi lain. Rancangan *activity diagram* layanan kepegawaian seperti yang ditampilkan pada Gambar 11.



Gambar 11: Activity Diagram Layanan Kepegawaian

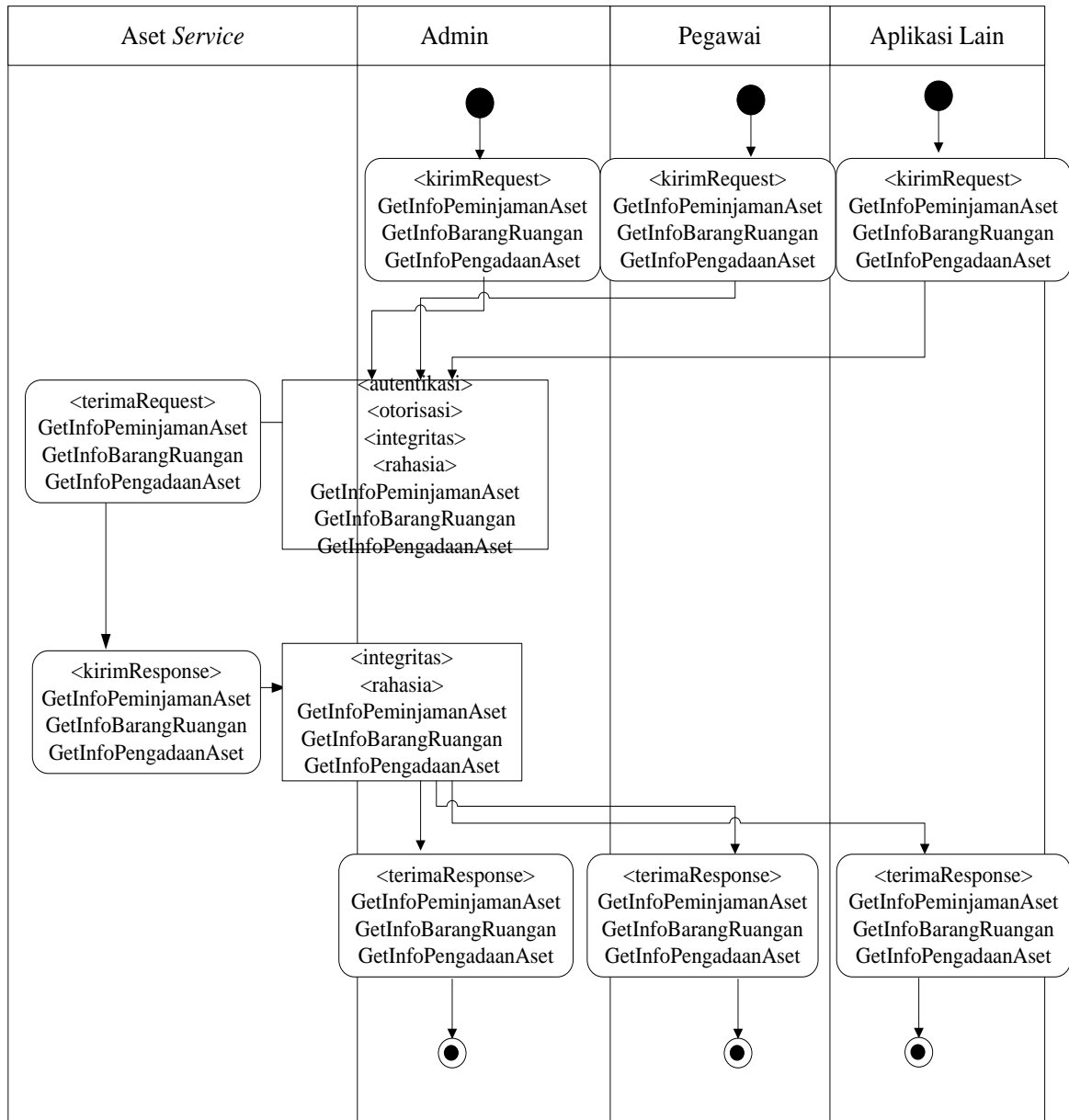
Rancangan *activity diagram* layanan aset terdiri dari tiga buah *service* yaitu *service* info peminjaman aset, info barang ruangan dan info pengadaan aset yang dapat diakses oleh tiga

buah aktor yaitu admin, pegawai, dan aplikasi lain. Layanan *service* yang diberikan hanya dapat mengakses saja,

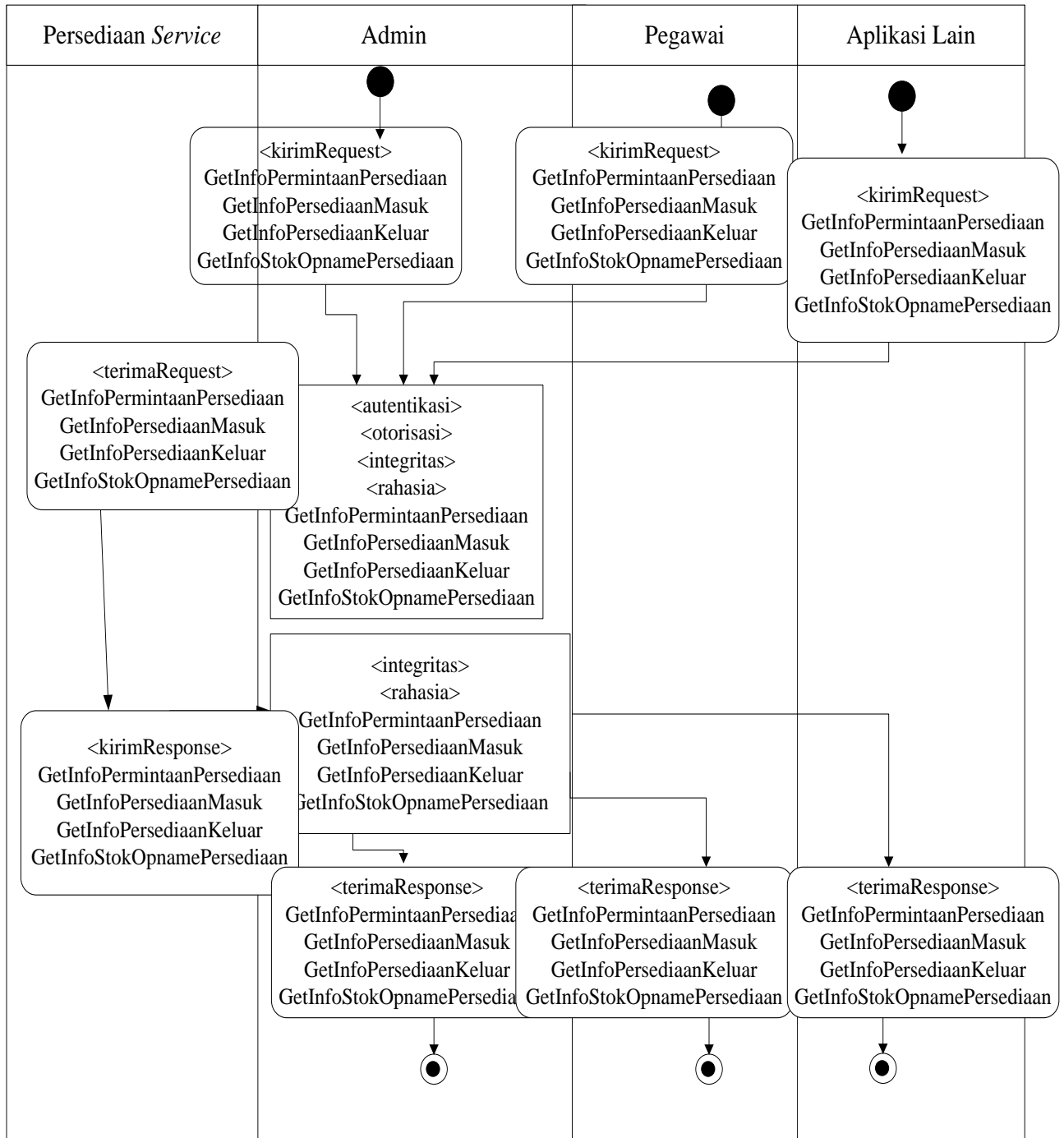
rancangan *activity diagram* layanan aset seperti yang ditampilkan pada Gambar 12.

Rancangan *activity diagram* layanan persediaan terdiri dari empat buah *service* yaitu *service* info permintaan persediaan, info persediaan masuk, info persediaan keluar

dan info stok opname persediaan yang dapat diakses oleh tiga buah aktor yaitu admin, pegawai, dan aplikasi lain. Layanan *service* yang diberikan hanya dapat mengakses saja, rancangan *activity diagram* layanan persediaan seperti yang ditampilkan pada Gambar 13.



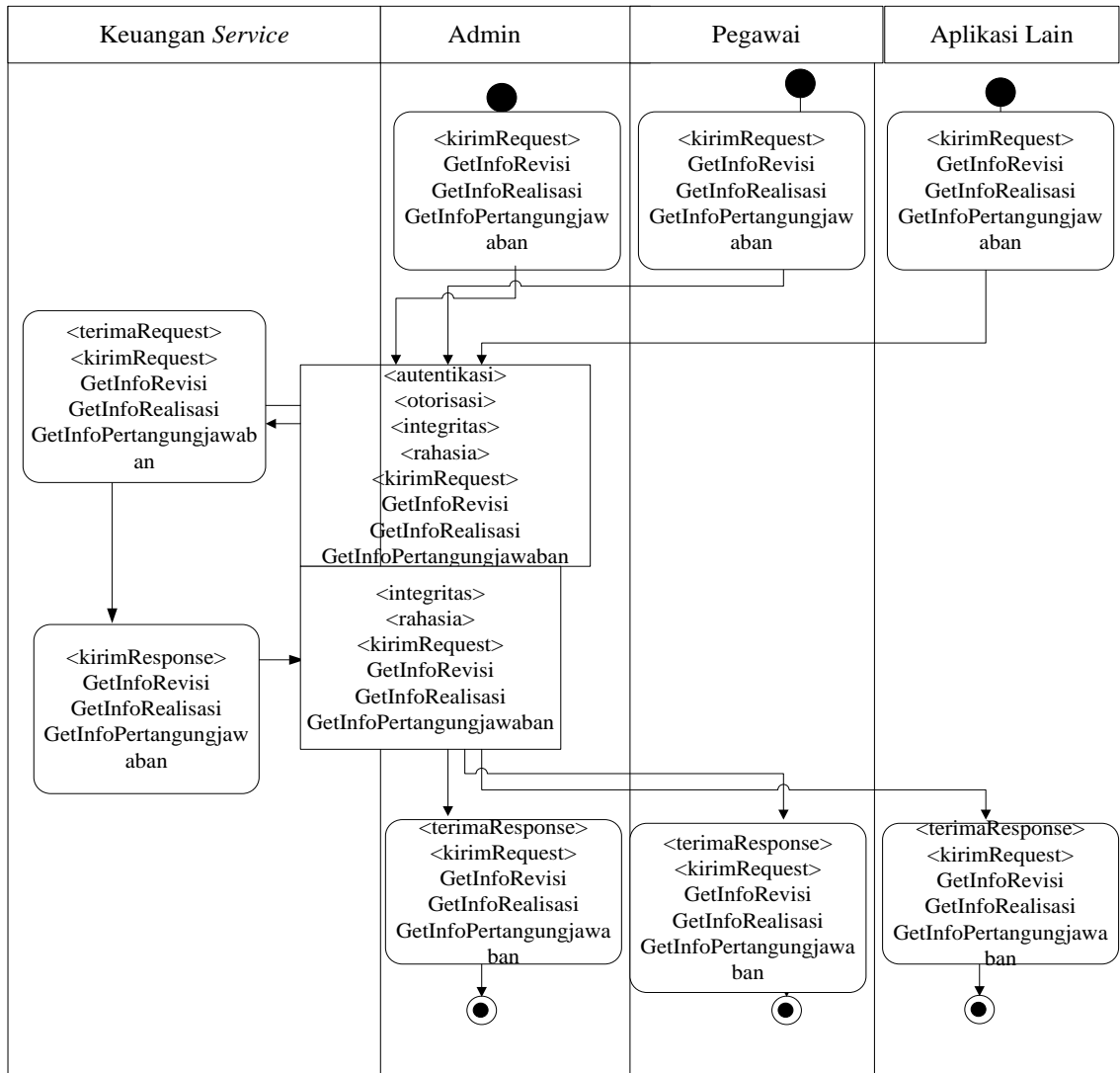
Gambar 12: Activity Diagram Layanan Aset



Gambar 13: Activity Diagram Layanan Persediaan

Rancangan *activity diagram* layanan keuangan terdiri dari tiga buah *service* yaitu *service* info revisi, info realisasi, dan info pertanggungjawaban yang dapat diakses oleh tiga buah aktor yaitu admin, pegawai, dan aplikasi lain. Layanan

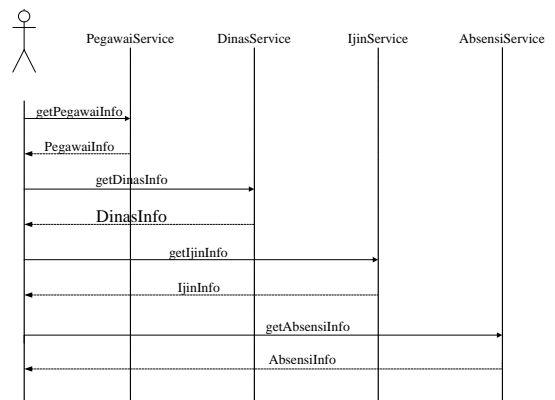
service yang diberikan hanya dapat mengakses saja, rancangan *activity diagram* layanan keuangan seperti yang ditampilkan pada Gambar 14.



Gambar 14: Activity Diagram Layanan Keuangan

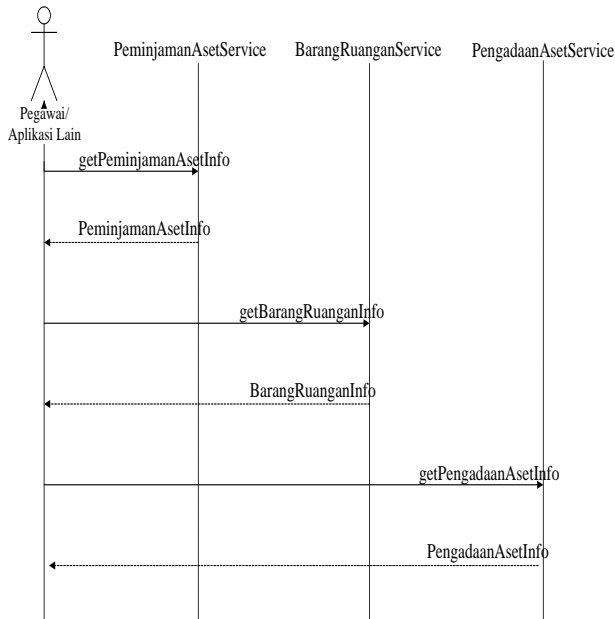
H. Perancangan Prototipe Interface

Perancangan prototipe *interface* berfungsi untuk memudahkan pengguna layanan dalam mengakses atau *invoke* sebuah *service* yang terdapat pada suatu layanan. Prototipe untuk layanan kepegawaian dapat direpresentasikan kedalam *sequence diagram* yang digambarkan sebagai berikut:



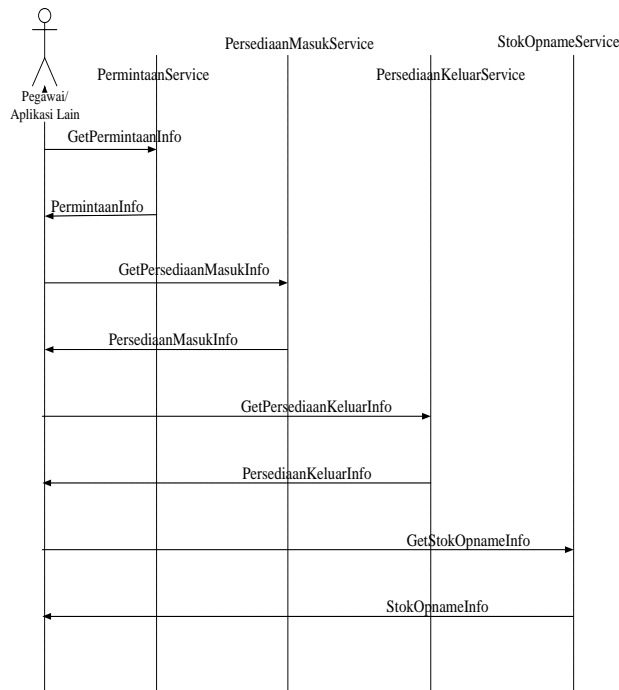
Gambar 15: Sequence Diagram Layanan Kepegawaian

Prototipe *interface* untuk layanan aset dapat direpresentasikan kedalam *sequence diagram* yang digambarkan sebagai berikut:



Gambar 16: Sequence Diagram Layanan Aset

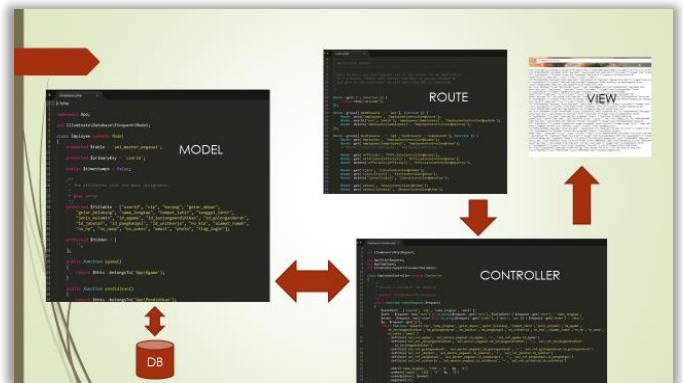
Prototipe *interface* untuk layanan persediaan dapat direpresentasikan kedalam *sequence diagram* yang digambarkan sebagai berikut:



Gambar 17: Sequence Diagram Layanan Persediaan

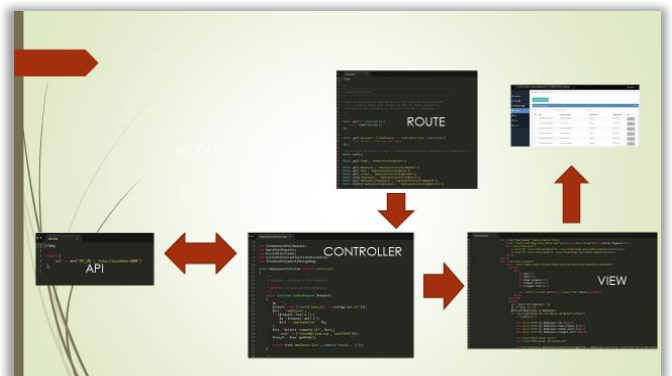
I. Prototipe

Tahapan pembuatan prototipe ini merupakan *fase development* yaitu menerapkan model arsitektur dengan pendekatan SOA dan metode MVC dengan *framework* Laravel yang menggunakan bahasa pemrograman PHP. Prototipe yang dibangun sesuai dengan kebutuhan dan hasil analisis layanan yaitu prototipe layanan kepegawaian, layanan aset, layanan persediaan dan layanan keuangan. Prototipe yang dibangun terdiri dari dua aplikasi yaitu aplikasi sisi *server* dan sisi *client*, konsep pemrograman pembuatan prototipe berdasarkan metode MVC seperti yang terlihat pada Gambar 18 dan Gambar 19.



Gambar 18: Konsep MVC Sisi Server

Konsep MVC dari sisi *server* menggunakan *framework* laravel menjelaskan database diakses dari *Model* kemudian direpresentasikan melalui *controller* untuk dapat ditampilkan ke halaman *browser* (*View*). *Route* dalam *framework* laravel pada prinsipnya hanya masalah *request-response* yaitu menyajikan URL yang diakses kemudian *server* akan memberikan respon. Fungsi *route* dalam *framework* laravel yaitu memetakan URL yang diminta ke bagian kode program tertentu.



Gambar 19: Konsep MVC Sisi Client

Konsep MVC pada sisi *client* data/informasi diakses melalui *web service* (API). Pada sisi *client* data/informasi diakses tidak menggunakan akses database langsung melainkan melalui alamat URL dari *web service*. *Controller* dapat berperan mengakses alamat URL yang diberikan oleh *client* untuk dapat mengakses data /informasi, sebelum ditampilkan ke *browser* melalui *View*.

J. Pembangunan Web Service

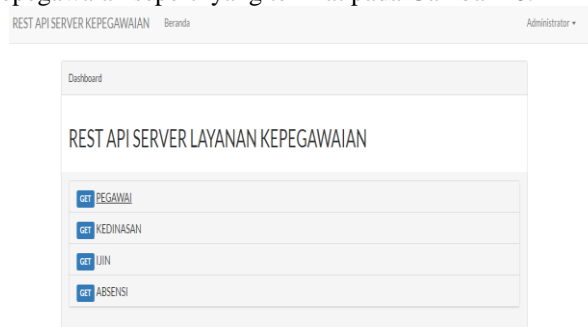
Proses perancangan *web service* sebagai prototipe yang akan dibangun semua menggunakan framework Laravel. Terdapat beberapa *web method* yang akan diciptakan sebagai prototipe layanan *web service*, yaitu pegawai, kedinasan, ijin, absensi, peminjaman aset, barang ruangan, pengadaan aset, permintaan persediaan, persediaan masuk, persediaan keluar, stok *opname*, revisi anggaran, realisasi anggaran dan pertanggungjawaban anggaran. Tabel IV. Merupakan deskripsi informasi layanan-layanan dalam penelitian ini yang akan dijadikan sebagai prototipe.

TABEL IV.
DESKRIPSI PROTOTYPE LAYANAN YANG DIBANGUN

No	Kelas Implementasi	Keterangan
1	Layanan Kepegawaian	
	employee.php	Implementasi prototipe <i>service</i> untuk melihat, merubah, mencari dan menghapus data pegawai.
	kedinasan.php	Implementasi prototipe <i>service</i> untuk melihat informasi kedinasan.
	ijin.php	Implementasi prototipe <i>service</i> untuk melihat informasi ijin.
	absensi.php	Implementasi prototipe <i>service</i> untuk melihat informasi absensi/kehadiran pegawai.
2	Layanan Aset	
	pinjam.php	Implementasi prototipe <i>service</i> untuk melihat, merubah, mencari dan menghapus data peminjaman barang milik negara.
	brgruang.php	Implementasi prototipe <i>service</i> untuk melihat informasi daftar barang ruangan.
	pengadaan.php	Implementasi prototipe <i>service</i> untuk melihat informasi pembelian barang milik negara.
3	Layanan Persediaan	
	permintaan.php	Implementasi prototipe <i>service</i> untuk melihat, merubah, mencari dan menghapus data permintaan persediaan.
	psmasuk.php	Implementasi prototipe <i>service</i> untuk melihat informasi pembelian barang persediaan.

No	Kelas Implementasi	Keterangan
	pskeluar.php	Implementasi prototipe <i>service</i> untuk melihat informasi pengeluaran barang persediaan.
	opname.php	Implementasi prototipe <i>service</i> untuk melihat informasi ketersediaan barang persediaan.
4	Layanan Keuangan	
	revisi.php	Implementasi prototipe <i>service</i> untuk melihat informasi revisi rencana anggaran belanja.
	realisasi.php	Implementasi prototipe <i>service</i> untuk melihat detail realisasi anggaran yang sudah dikeluarkan.
	tgjawaban.php	Implementasi prototipe <i>service</i> untuk melihat informasi pertanggungjawaban dari rencana anggaran belanja kegiatan yang sudah dikeluarkan.

Prototipe *web service* dalam penelitian ini hanya mencakup pada pembangunan *web service* untuk layanan kepegawaian yang terdiri dari sisi server dan sisi client. Layanan kepegawaian terdiri dari 4 (empat) buah layanan yaitu layanan pegawai, kedinasan, ijin dan absensi. Layanan pegawai yang diberikan dapat berfungsi untuk melihat (GET), menambah (POST), merubah (PUT), dan menghapus (DEL) informasi data pegawai. Sedangkan layanan kedinasan, ijin dan absensi yang diberikan hanya sebatas untuk melihat informasi saja. Tampilan layar REST API Server Kepegawaian seperti yang terlihat pada Gambar 18.



Gambar 20: Layanan Server Kepegawaian

Untuk mengecek layanan kepegawaian *service* dapat diakses dengan mengetik alamat URL yang disediakan. Contoh tampilan dengan mengakses menggunakan browser melalui alamat URL seperti yang terlihat pada Gambar 21 yang menjelaskan data hasil pengaksesan layanan pegawai yang ditampilkan dalam format JSON. Data yang ditampilkan tersebut merupakan data yang dapat diakses oleh client.



Gambar 21: Informasi Hasil Layanan Pegawai dalam Format JSON

Layanan pada sisi *client* yang digunakan merupakan prototipe dengan mengakses *web service* dari sisi *server* yang sudah bangun sebelumnya. Gambar 22 merupakan tampilan informasi layanan sisi *client* pada informasi pegawai yang

diakses dari *service* kepegawaian. Selain daftar pegawai (GET) yang ditampilkan, terdapat juga untuk mengurutkan data pegawai (SEARCH), menambah data pegawai (POST), merubah data pegawai (PUT) dan menghapus data pegawai (DEL).

NIP	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Pangkat/Golongan	Jabatan	Action
195207071977032002	Iin Surminah	Jakarta	1952-07-07	Pembina Tk. I	Peneliti Madya	[View] [Edit] [Delete]
195302171983031002	Azis Taba Pabeta	Makassar	1953-02-17	Pembina Utama Muda	Peneliti Madya	[View] [Edit] [Delete]
195309231982031001	Lukman Hakim	Curup	1953-08-23	Pembina Utama	Peneliti Utama	[View] [Edit] [Delete]
195510021977031007	Agus Santoso		1955-10-02	Pembina Tk. I	Peneliti Madya	[View] [Edit] [Delete]
195510081984031001	Erman Aminullah	Padang	1955-10-08	Pembina Utama	Peneliti Utama	[View] [Edit] [Delete]
195612251983032003	Hartiningsih	Sala	1956-12-25	Pembina Tk. I	Peneliti Madya	[View] [Edit] [Delete]
195710101983032003	Sri Mulatsih		1957-10-10	Pembina	Peneliti Madya	[View] [Edit] [Delete]
195710141988031002	Saut H. Siahaan	Bogor	1957-10-14	Pembina Tk. I	Peneliti Madya	[View] [Edit] [Delete]
196001011982031013	Sheffied	Jakarta	1960-01-01	Penata Muda Tk. I	Penata Usaha Informasi Ilmiah	[View] [Edit] [Delete]
196009031983032005	Ummi Aslamah	Ponorogo	1960-09-03	Penata	Pengadministrasi Keuangan	[View] [Edit] [Delete]

Gambar 22: Daftar Pegawai pada Sisi Client Layanan Kepegawaian

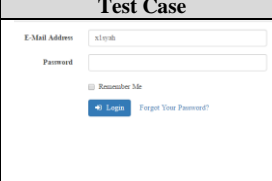
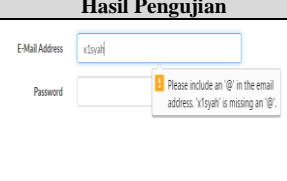
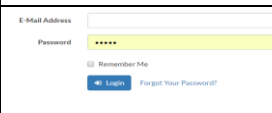
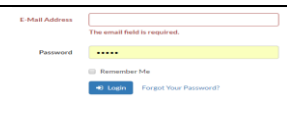
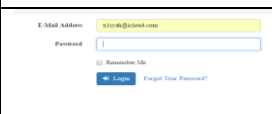
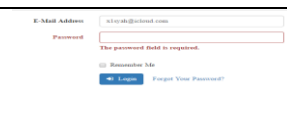
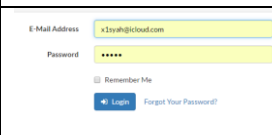
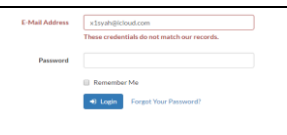
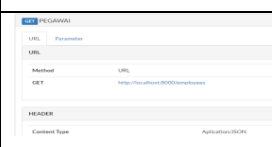
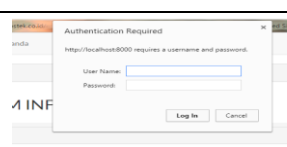
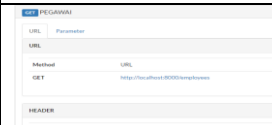

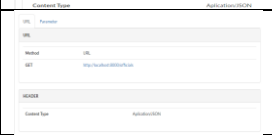

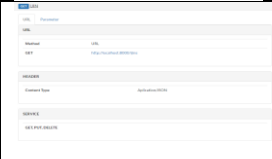

K. Pengujian

Pengujian merupakan bagian yang sangat penting dalam membangun suatu perangkat/prototipe, pengujian ditujukan untuk menemukan kesalahan-kesalahan pada suatu sistem dan memastikan sistem yang telah dibangun sudah sesuai dengan apa yang direncanakan. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari suatu perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal yaitu mampu mempresentasikan kajian pokok dan spesifikasi analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

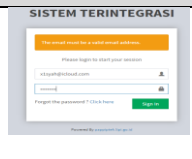
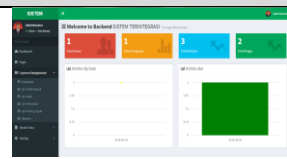
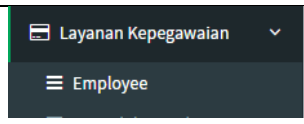
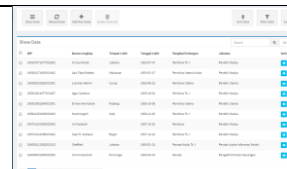
Pengujian dilakukan untuk mengetahui apakah penerapan SOA dan metode MVC dapat menunjang *web service* yang


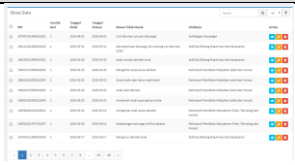

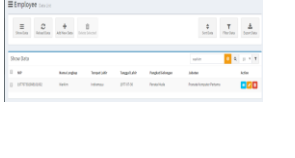
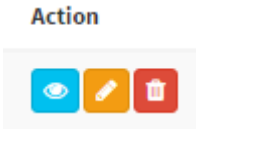

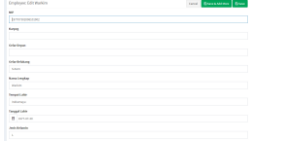
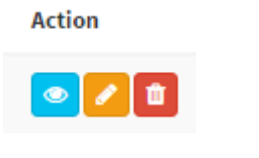

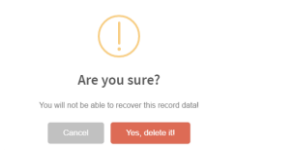
sesuai dengan analisis kebutuhan yang telah ditentukan. Pengujian ke responden dilakukan oleh beberapa staf dilingkungan bagian tata usaha yang terkait dengan penggunaan sistem informasi ketatausahaan antara lain pengelola persediaan, pengelola kepegawaian, pengelola aset, pengelola keuangan dan pranata komputer. Pelaksanaan pengujian ini berdasarkan skenario yang telah ditentukan dengan setiap proses dan kemungkinan kesalahan yang dapat terjadi. Masing-masing *service* dan modul yang ada pada aplikasi memiliki skenario *test*, dimana terdapat *test case* yang beragam, kemudian akan diamati hasilnya apakah sesuai dengan yang dikehendaki. Skenario pengujian dengan pendekatan *Black Box* dilakukan pada aplikasi dari sisi *server* dan sisi *client* seperti yang terlihat pada Tabel V. dan Tabel VI.

TABEL V.
PENGUJIAN DENGAN METODE *BLACK BOX* PADA SISI *SERVER*

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Input <i>username</i> selain email		Sistem akan menolak jika klik tombol login, dan menampilkan pesan “Please include an “@” in the email address “...” is missing an @.”		Valid
2	Isian email <i>address</i> tidak di isi		Sistem akan menolak jika klik tombol login dan menampilkan pesan “The Email field is required”.		Valid
3	Isian password tidak diisi		Sistem akan menolak jika klik tombol login dan menampilkan pesan “The password field is required”.		Valid
4	Mengisi isian <i>password</i> yang salah/tidak sesuai		Sistem akan menolak jika klik tombol login dan akan menampilkan pesan “These credentials do not match our record”.		Valid
5	Autentikasi untuk menampilkan data pegawai dalam format JSON		Sistem akan menampilkan autentikasi untuk memasukan <i>username</i> dan <i>password</i>		Valid
6	Menampilkan data pegawai dalam format JSON		Sistem akan menampilkan data profil pegawai dalam format JSON		Valid
7	Menampilkan/menerima <i>request</i> informasi data kedinasan		Sistem akan menampilkan data kedinasan pegawai dalam format JSON		Valid
8	Menampilkan/menerima <i>request</i> informasi data ijin (tidak masuk kerja, terlambat, pulang cepat dan sakit)		Sistem akan menampilkan data ijin pegawai dalam format JSON		Valid

TABEL VI.
SKENARIO PENGUJIAN DENGAN METODE *BLACK BOX* PADA SISI *CLIENT*

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	Dapat melakukan login ke aplikasi <i>client</i> layanan kepegawaian		Sistem akan menampilkan halaman <i>dashboard</i> sistem terintegrasi		Valid
2	Menampilkan/mengirim <i>request</i> informasi profil pegawai		Sistem akan menampilkan data/informasi daftar pegawai apabila melakukan klik <i>Employee</i> pada menu Layanan Kepegawaian		Valid

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
3	Menampilkan/mengirim request informasi data ijin pegawai		Sistem akan menampilkan data/informasi daftar pegawai apabila melakukan klik ijin tidak masuk, terlambat, pulang cepat dan sakit pada menu Layanan Kepegawaian		Valid
4	Dapat menampilkan pencarian data pegawai didalam daftar pegawai		Sistem akan menampilkan data/informasi profil pegawai apabila melakukan pencarian dengan memilih kriteria pencarian mengetik kata kunci yang dicari		Valid
5	Dapat merubah data pegawai yang ada dalam daftar pegawai		Sistem akan menampilkan form edit data pegawai apabila melakukan klik  pada kolom <i>action</i> .		Valid
6	Dapat menghapus data pegawai didalam daftar pegawai yang ada		Sistem akan menampilkan form edit data pegawai apabila melakukan klik  pada kolom <i>action</i> .		Valid

Berdasarkan hasil pengujian *Focus Group Discussion* dapat disimpulkan bahwa aplikasi sistem terintegrasi baik dari sisi *server* maupun sisi *client* sudah sesuai dengan spesifikasi kebutuhan pengguna. Berdasarkan hasil analisis, desain arsitektur menggunakan pendekatan SOA dan metode MVC pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia dapat berfungsi dalam menyediakan kebutuhan integrasi dan pertukaran data sehingga dapat memecahkan permasalahan integrasi sistem antar aplikasi yang ada dibagian tata usaha.

V. KESIMPULAN

Hasil penelitian ini diperoleh suatu desain arsitektur dan prototipe sistem terintegrasi dengan pendekatan SOA dan metode MVC yang diterapkan pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia. Hasil rancangan dalam penelitian ini adalah sebuah arsitektur dengan pendekatan SOA diimplementasikan dengan menggunakan teknologi REST yang menghasilkan 4 (empat) buah layanan yaitu layanan kepegawaian, layanan aset, layanan persediaan dan layanan keuangan. Selain hasil rancangan arsitektur SOA dalam penelitian ini dihasilkan pula sebuah prototipe sistem terintegrasi sebagai *dashboard* untuk layanan-layanan yang dibangun berbasis web. Implementasi sistem terintegrasi dibangun terdapat dua aplikasi yaitu aplikasi sisi *server* dan aplikasi sisi *client*, yaitu dengan menggunakan teknologi *web service* REST yang dibuat dengan menggunakan *framework* Laravel dan database MySQL.

Hasil dari penelitian ini diharapkan dapat diterapkan pada instansi pemerintah atau satuan kerja lainnya khususnya satuan kerja dilingkungan Lembaga Ilmu Pengetahuan Indonesia yang memiliki tugas dan fungsi kerja yang sama. Sistem terintegrasi hasil penelitian ini dapat diimplementasikan membutuhkan dukungan dan komitmen dari pimpinan. Dukungan berupa kebijakan untuk diterapkan serta dukungan sarana dan prasarana termasuk untuk membangun infrastruktur jaringan internet sesuai yang dibutuhkan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pimpinan serta rekan kerja di Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia. Dosen dan staf Pasca Sarjana Magister Ilmu Komputer Universitas Budiluhur. Terima kasih kami haturkan kepada bapak Dana Indra Sensuse selaku pembimbing dalam penelitian ini serta rekan-rekan mahasiswa Pasca Sarjana Angkatan 2014-2016 Magister Ilmu Komputer Universitas Budiluhur.

DAFTAR PUSTAKA

- [1] Mohammadi, M., Muchtar, M., 2013, *A Review of SOA Modeling Approaches for Enterprise Information Systems*, The 4th International Conference on Electrical Engineering and Informatics (ICEEI), Procedia Technology 11 (2013) 794 – 800.
- [2] Devi, C.P. ET AL., 2014, *A Model for Information Integration Using Service Oriented Architecture*. IJ. Information Engineering and Electronic Business, June, pp.34-43.
- [3] Paul Pop, D., 2013, *Design an MVC Model of Rapid Web Application Development*, 24th DAAAM International Symposium on Intelligent

- Manufacturing and Automation, Romanian-American University, Bucharest, Romania.
- [4] Slameto, A.A., 2015, *Penerapan SOA dalam Proses Integrasi Sistem Informasi Inventaris Laboratorium dan Sistem Informasi Laporan Kerusakan Komputer pada Laporan STMIK AMIKOM*, Jurnal Teknologi Informasi, Vol. X, No. 30.
- [5] Mardiana, Araki, K. 2012, *A Model Driven Approach for the Interoperability of Web Application*, Paper presented to the Korea-Japan Joint Workshop on ICT, Pohang, Korea 20-22 September.
- [6] Mansouri, K., Riyami, B., Youssfi, M., Bouattane, O., 2015, *Model of an Adaptation and Interfacing SOA Middleware for the Information Systems Interoperability*, Journal of Theoretical and Applied Information Technology, Vol. 84 No. 1 107-126.
- [7] Nugroho, A., Mustofa, K., 2012, *Perbandingan Antara BIG Web Service dengan Restful Web Service untuk Integrasi Data berformat GML*, Jurnal Informatika Vol. 11 No. 1, Yogyakarta, Indonesia.
- [8] Chhikara, J., 2014, *A Web Architectural Study of HTML5 with MVC Framework*, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 12, December 2013, Haryana, India.
- [9] Mohammadi, M., Muchtar, M., 2013, *A Review of SOA Modeling Approaches for Enterprise Information Systems*, The 4th International Conference on Electrical Engineering and Informatics (ICEEI), Procedia Technology 11 (2013) 794 – 800
- [10] Erl, Thomas. 2007, *Service Oriented Architecture: Principles of Service Design*. New Jersey, Pearson Education, Inc.
- [11] Stenly R. Pungus. (2008), *Penerapan Service Oriented Architecture Untuk Pengintegrasian Sistem Informasi Perguruan Tinggi Study Kasus di Klabat (UNKLAB) Manado*, Tesis Program Master, Institut Teknologi Bandung.
- [12] Pienwittayasakul, C. and Liu, Y., 2014, *Comparative Study on Service Oriented Architecture and Even Driven Architecture*, Prociding of the international Conference on Computing Technology and Information Management, Dubai, UAE..
- [13] Juniati, D.P., Nugroho, L.E., & Nugroho, E., 2014, *Prototipe Layanan Izin Pemanfaatan Ruang menggunakan Service Oriented Enterprise Architecture Framework*, Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI), Yogyakarta, Indonesia.
- [14] Dewi, R.K., 2010, *Desain Interoperabilitas Cross-Aplikasi dengan Service Oriented Architecture*, Tesis, MT, Institut Teknologi Bandung, Bandung.
- [15] Ghifary, M., Karya, G., 2011, *Pemodelan dan Implementasi Antarmuka Web Service Sistem Informasi UNPAR*, Laporan Penelitian, Universitas Parahiyangan, Bandung.