

# ANALISA *SERVICE ORIENTED ARCHITECTURE*(SOA) MENGGUNAKAN WEB *SERVICE* UNTUK PENJADWALAN MUBALLIGH PADA IKMI DAN DDI

Iwan Iskandar<sup>1</sup>, Nur Jannah<sup>2</sup>

Teknik Informatika, Fakultas Sains dan Teknologi  
Universitas Islam Negeri Sultan Syarif Kasim Riau

Jl. H. R Subrantas No. 155 KM. 18 Simpang Baru Panam – Pekanbaru. PO.Box 1004

[iwaniskandar01@gmail.com](mailto:iwaniskandar01@gmail.com); [nurjannah@gmail.com](mailto:nurjannah@gmail.com)

## Abstrak

DDI (Dewan Dakwah Indonesia) merupakan organisasi Islam yang bergerak dibidang dakwah. IKMI (Ikatan Mesjid Indonesia) merupakan bagian dari DDI yang bertugas dilapangan dalam mengatur khutbah jumat, instruktur pengajian, penceramah dan lain-lain. Pada saat ini terdapat 370 masjid dan 1400 muballigh dibawah naungan IKMI. Selama ini pengaturan jadwal penceramah untuk beberapa mesjid dilakukan secara manual. Pengurus mengalami beberapa kendala terutama bagi penceramah yang berhalangan dan kembali harus mengatur penjadwalan ulang, minimnya tingkat ketelitian dalam penyusunan jadwal, mengingat banyaknya muballigh dan mesjid, kurang efektifnya buku panduan karena faktor umur dan daya ingat serta pemantauan dan evaluasi yang kurang berjalan dengan baik. Untuk mengatasi hal tersebut maka dilakukan analisa terhadap faktor-faktor yang menjadi penyebab kendala-kendala tersebut. Dalam hal ini diterapkan arsitektur SOA dimana setiap objek informasi dapat dijadikan sebagai *Service-Service* yang dapat diakses serta dapat dikembangkan untuk digunakan kembali pada aplikasi lain (*reusable*). Pada penerapannya maka digunakan teknologi *web Service* karena mampu menyediakan standard dari perbedaan aplikasi dan berjalan dibanyak *platform* sehingga dapat digunakan untuk integrasi antar pengawas, pengurus dan petugas dilapangan. Pada analisa ini dihasilkan beberapa *Service* penting untuk mengatur dan melakukan integrasi dari beberapa sistem yang ada. Arsitektur sistem dihasilkan sebagai proyeksi sistem dalam menjalankan *Service-Service* yang telah dibuat dan dapat dikembangkan pada aplikasi yang lain. Pengiriman informasi pada aplikasi yang berbeda dengan memanfaatkan *web Service* menjadi solusi yang baik dari kendala yang dihadapi oleh pengurus IKMI selama ini.

**Kata Kunci:** DDI, IKMI, *Reusable*, *Service*, *Web Service*

## Abstract

DDI (Dewan Dakwah Indonesia) is an Islamic organization engaged in proselytizing. IKMI (Ikatan Mesjid Indonesia) is part of the DDI in charge of the field in regulating the Friday sermon, recitation instructors, lecturers and others. There currently are 370 mosques and preachers in 1400 under the auspices of IKMI. During this schedule setting in some mosque preacher to be done manually. Administrators having some problems especially for preachers who can not and should re-arrange the rescheduling, the low level of accuracy in the preparation of the schedule, considering the number of preachers and mosques, the lack of effective guide book because of age and memory as well as the lack of monitoring and evaluation work well. To overcome this then do an analysis of the factors that cause these constraints. In this case applied SOA architecture where each object can be used as an information *Services* that are accessible and can be developed for reuse in other applications (*reusable*). In practice the use of web technology is able to provide standard *Service* because of differences in the application and run in many platforms that can be used for integration between supervisors, administrators and officers. On this analysis produced some important *Service* to manage and perform the integration of several existing systems. The system architecture is generated as a projection system in running the *Services* that have been created and can be developed in other applications. Shipping information for different applications by utilizing *web Service* be a good solution of the constraints faced by the board IKMI.

**Key Word:** DDI, IKMI, *Reusable*, *Service*, *Web Service*

## PENDAHULUAN

DDI (Dewan Dakwah Indonesia) merupakan organisasi Islam yang bergerak dibidang dakwah. IKMI (Ikatan Mesjid Indonesia) merupakan bagian dari DDI yang bertugas

dilapangan. Ada 370 masjid dan 1400 muballigh dibawah naungan IKMI yang bertugas sebagai khutbah jumat, instruktur pengajian, penceramah dan lain-lain. Tugas-tugas ini disusun oleh pengurus dan

dibagikan kepada petugas dilapangan. pada kenyataannya pengurus dan petugas menemukan banyak kendala, mulai dari tingkat ketelitian dalam penyusunan jadwal, mengingat banyaknya muballigh dan mesjid, kurang efektifnya buku panduan karena faktor umur dan daya ingat, pemantauan dan evaluasi yang kurang berjalan dengan baik karena pemantauan terjadi tidak secara langsung dan dalam waktu yang telah ditentukan, sementara dilapangan petugas bisa mendapat masalah setiap waktu dan membutuhkan penanganan yang cepat.

Berbagai terobosan dilakukan untuk meminimalisir dampak yang terjadi dari kendala yang dihadapi oleh pengurus dan petugas dilapangan. Untuk konsep integrasi dan kerjasama yang dimanis dan terbuka maka harus difokuskan pada sistem dari organisasi tersebut, Professional IT secara tidak langsung dapat menjawab tantangan tersebut untuk mampu membuat dan mengontrol teknologi dalam batasan fungsi layanan organisasi serta heterogenitas yaitu suatu sistem yang berbasis *Service-oriented*. Arsitektur system yang berbasis *Service* dikenal dengan istilah *Service Oriented Architecture* (SOA).

SOA dapat diimplementasikan dengan beberapa teknologi salah satunya *web Service*. Dimana *web Service* mampu menyediakan standard dari perbedaan aplikasi dan berjalan dibanyak platform, Sehingga dapat digunakan untuk integrasi antar pengawas, pengurus dan petugas dilapangan. Selain itu aplikasi dapat dimodelkan dari sekumpulan *Service* yang disediakan oleh komponen, dimana lokasi keberadaan komponen dapat ditemukan clien secara dinamis. *Web Service* merupakan konsep yang sangat mendukung untuk kemajuan dan keberhasilan kerja organisasi melayani masyarakat.

Beberapa penelitian yang menerapkan *Service oriented architecture* (SOA) untuk beberapa aplikasi telah dilakukan. berdasarkan penelitian sebelumnya oleh Toninetti (2009) Jurusan Database, BINUS University Jakarta yaitu Analisis Dan Perancangan Modul AplikasiPenerimaan

Mahasiswa Baru Berbasiskan *Service-Oriented Architecture*(SOA) berkesimpulan modul-modul yang telah dikembangkan sebelumnya bisa digunakan kembali untuk sistem yang baru dengan berbasis pengembangan *Services* sehingga dapat mempercepat proses pengembangan. Selain itu penelitian juga dilakukan oleh Luthfi Ramadani,(2012) program sarjana sistem dan teknologi informasi-STIE ITB dengan judul Perancangan Sistem Apotek Rumah Sakit Berbasis SOA dan *Cloud Computing*, yang mendapat beberapa keunggulan arsitektur SOA yang tercakup dalam aspek *agility* dan *reusability* yang bisa dikontrol tanpa mengganggu sistem lain. Dan penelitian oleh Marini 2012 mengatakan SOA menyediakan suatu bentuk abstraksi suatu sistem dan *web Service* merupakan teknologi yang mengatur bagaimana suatu sistem dapat berinteraksi dengan yang lain. Konsep yang dimiliki SOA berdampak pada kemudahan dalam membangun sistem.

Penelitian mengenai membangun sistem berbasis SOA diorganisasi IKMI dan DDI sangat diperlukan, untuk mengidentifikasi pada strategi untuk bertugas dan memberi layanan kepada masyarakat dengan baik, memperkecil anggaran belanja organisasi, dan mempunyai arti penting bagi masyarakat, membuka wawasan akan pentingnya keberadaan muballigh ditengah kehidupan masyarakat.

### ***SERVICE ORIENTED ARCHITECTURE(SOA)***

Mabrouk (2008) mengutip definisi layanan dari “*Web Services and Service-Oriented Architecture: The Savvy Manager’s Guide*” sebagai fungsi-fungsi yang telah terdefinisi, bersifat independen, serta tidak bergantung pada keadaan layanan yang lainnya. Kemudian definisi lain dikutip dari “*SearchSystemChannel.com*” sebagai satu unit pekerjaan yang dilakukan oleh entitas komputasi tertentu, seperti manusia ataupun program komputer.

Sejumlah layanan yang terpisah dan dikumpulkan menjadi satu (agregasi) untuk

mengotomatisasi tugas ataupun proses bisnis tertentu disebut *Service composition*. Pada *Service composition*, setidaknya harus ada minimal dua buah layanan dan sebuah penggerak yang memulai komposisi (*initiator*). Jika tidak, interaksi hanya merupakan pertukaran data biasa antara satu layanan ke layanan lain.

Berorientasi layanan dapat diartikan sebagai suatu cara berpikir yang terdiri dari seperangkat prinsip tertentu. Penerapan prinsip ini dalam mendesain solusi akan menghasilkan solusi yang berorientasi layanan. Unit paling mendasar pada solusi berorientasi layanan adalah layanan (*Service*) (Erl, 2009).

Erl (2005) mendefinisikan SOA sebagai bentuk arsitektur teknologi yang menganut prinsip-prinsip berorientasi layanan. Ketika diwujudkan dalam teknologi *web Service*, SOA memberikan potensi untuk mendukung prinsip-prinsip ini di seluruh proses bisnis dan domain otomatisasi perusahaan. Idealnya, SOA distandarisasi di seluruh interprice, namun untuk mencapai target ini dibutuhkan transisi yang direncanakan dengan matang serta dapat mendukung teknologi ataupun sistem yang saat ini masih ada dan digunakan (*legacy system*).

SOA membangun model arsitektur yang fokus pada peningkatan efisiensi, kelincahan, serta produktivitas dengan menempatkan layanan sebagai hal utama. Sebagai bentuk arsitektur teknologi, implementasi SOA dapat berupa kombinasi berbagai teknologi, produk, *application programming interface* (API), tambahan infrastruktur pendukung, serta bagian lain (Erl, 2009).

## PRINSIP BERORIENTASI LAYANAN

Erl (2005) memaparkan beberapa prinsip umum pada berorientasi layanan sebagai berikut:

- a. *Service reusability*. Layanan dapat digunakan ulang, sehingga layanan harus didesain agar dapat mendukung penggunaan kembali.
- b. *Service contract*. Agar *Service* dapat berinteraksi, mereka harus memberikan

kontrak yang berisi bentuk layanan dan informasi yang dipertukarkan.

- c. *Service loose coupling*. Layanan tidak terhubung erat. Maksudnya layanan didesain agar dapat bekerja tanpa bergantung pada layanan tertentu.
- d. *Service abstraction*. Layanan membungkus logika. Bagian yang tampak dari luar hanyalah kontrak layanan. Logika serta implementasi di balik itu tidak kelihatan dan tidak penting bagi peminta layanan. *Service composability*. Beberapa layanan dapat disusun untuk membentuk layanan baru.
- e. *Service autonomy*. Layanan bersifat mandiri. Pengembangan layanan dapat dilakukan terpisah. Layanan memiliki kendali penuh terhadap logika yang dimilikinya.
- f. *Service statelessness*. Layanan tidak didesain untuk menyimpan informasi keadaan tertentu (*state*).
- g. *Service discoverability*. Layanan harus dapat ditemukan. Deskripsi layanan harus dapat dicari dan dimengerti oleh peminta layanan.

## WEB SERVICE

*Web Service* adalah bentuk implementasi layanan yang menyediakan kontrak terpisah.

*Web Service* merupakan antarmuka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet, dalam bentuk pesan XML (Kreger dalam Priyambodo, 2005).

*Web Service* mempublikasi kemampuannya dalam bentuk operasi (*operation*), membentuk antarmuka teknis yang tidak tergantung framework maupun *platform* tertentu. *Web Service* terdiri dari *Servicecontract*, unit layanan dan unit pemrosesan pesan. Berikut adalah bagian yang sama yang terdapat pada *web Service*:

- a. Pesan SOAP (*SOAP messages*)
- b. Operasi layanan web (*web Service operations*)
- c. Layanan web (*web Services*)
- d. Aktivitas (*activity*)

Setiap layanan terdiri dari satu atau lebih operasi. Setiap operasi mengatur pengolahan fungsi tertentu yang dapat ditangani oleh layanan. Pengolahan ini terdiri dari pengiriman dan penerimaan pesan. Dengan menyusun bagian-bagian ini, layanan membentuk suatu aktivitas atau proses yang dapat mengotomatisasi suatu tugas tertentu.

## **WEB SERVICE DESCRIPTION LANGUAGE (WSDL)**

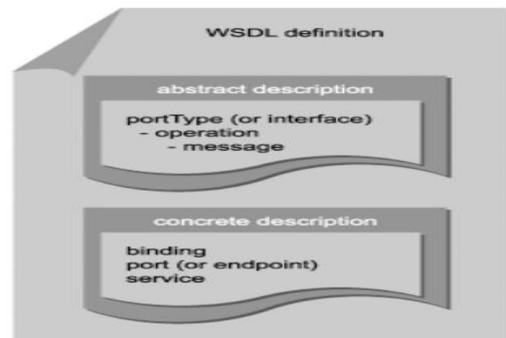
Mabrouk (2008) memaparkan bahwa WSDL merupakan dokumen yang ditulis menggunakan XML untuk menggambarkan *web service*. WSDL menunjukkan lokasi dan operasi apa yang diberikan oleh layanan sehingga dapat digunakan oleh peminta layanan (*serviceconsumer*). Suatu WSDL menggambarkan empat hal berikut:

1. Layanan yang disediakan oleh *web service*, seperti daftar operasi yang dapat digunakan serta atribut pesan.
2. Tipe data pesan, baik pesan masukan (*input*) maupun pesan keluaran (*output*).
3. Informasi *binding* untuk protokol transportasi, seperti HTTP dan JMS.
4. Alamat layanan yang digunakan untuk memanggilnya.

WSDL merupakan sebuah bahasa berbasis XML yang digunakan untuk mendefinisikan *web service* dan menggambarkan cara untuk mengakses *web service* tersebut (Shohoud dalam Priyambodo, 2005). Deskripsi WSDL mendefinisikan sebuah *service* sebagai kumpulan dari *port*, di mana tiap-tiap *port* didefinisikan secara abstrak sebagai *portType* yang mendukung sekumpulan operasi. Tiap-tiap operasi memproses sekumpulan pesan tertentu (Priyambodo, 2005). Berdasarkan tulisan yang berjudul "Introduction to WSDL", WSDL telah menjadi rekomendasi W3C sejak 26 Juni 2007.

Erl (2005) menjelaskan bahwa WSDL menggambarkan titik hubungan sebuah

penyedia layanan (*service provider*), yang juga dikenal dengan sebutan *service endpoint* atau *endpoint* saja. WSDL menyediakan definisi formal dari antarmuka *endpoint*, sehingga apapun yang akan berkomunikasi dengan layanan mengetahui dengan tepat bagaimana ia harus menyusun pesannya. WSDL juga memberikan lokasi fisik (alamat) sebuah layanan. Suatu WSDL dapat dibagi kedalam dua kategori yaitu deskripsi abstrak dan deskripsi konkrit, seperti diperlihatkan pada Gambar 1.



**Gambar 1 Deskripsi Abstrak dan Deskripsi Konkrit pada WSDL (Erl, 2005)**

Deskripsi abstrak memberikan informasi tentang antarmuka *web service* tanpa tergantung teknologi apapun yang digunakan. Dengan memisahkan informasi ini, integritas deskripsi layanan (*service description*) dapat dijaga terhadap semua perubahan yang mungkin terjadi pada *platform* di mana layanan dijalankan.

Ada tiga bagian utama dalam deskripsi abstrak, yaitu *portType*, *operation* dan *message*. Bagian *portType* menjelaskan pesan (*message*) apa saja yang dapat diproses oleh layanan ke dalam sekumpulan fungsi yang disebut sebagai operasi (*operation*). Setiap operasi menunjukkan aksi tertentu yang dilakukan oleh layanan. Operasi terdiri dari sekumpulan pesan masukan (*inputmessage*) dan pesan keluaran (*outputmessage*). Pada WSDL versi 2.0, *portType* berganti nama menjadi *interface*.

Agar suatu *web service* dapat menjalankan prosesnya, definisi antarmuka abstrak perlu dihubungkan dengan suatu implementasi teknologi

yang nyata, yakni protokol *transportasi*. Hubungan ini didefinisikan dalam deskripsi konkrit suatu WSDL, yang terdiri dari *binding*, *port* dan *service*. *Binding* menggambarkan teknologi *transportasi* yang akan digunakan layanan untuk berkomunikasi.

SOAP adalah bentuk yang paling umum, namun teknologi lain juga dapat digunakan. *Binding* dapat diterapkan pada keseluruhan antarmuka atau pada operasi tertentu. *Port* menggambarkan lokasi fisik di mana layanan dapat diakses dengan protokol tertentu. *Service* digunakan untuk menunjukkan sekumpulan *endpoint* yang saling berhubungan

### **SIMPLE OBJECT ACCESS PROTOCOL (SOAP)**

Mabrouk (2008) menerangkan bahwa SOAP merupakan protokol yang berbasis XML. Protokol ini memberikan kemudahan kepada aplikasi untuk bertukar informasi menggunakan protokol *transportasi* seperti HTTP. Beberapa keuntungan yang membuat SOAP sangat direkomendasikan dan telah menjadi protokol komunikasi yang paling banyak digunakan untuk *web Service* adalah sebagai berikut:

SOAP tidak tergantung pada *platform* manapun (*platform independent*). SOAP dapat digunakan dalam jaringan internet. Pesan SOAP dapat dibaca manusia, terstruktur, serta berbasis teks karena berupa XML. SOAP merupakan protokol yang digunakan untuk mempertukarkan data atau informasi dalam format XML (Scheinbum dalam Priyambodo, 2005). SOAP dapat dikatakan sebagai gabungan antara HTTP dengan XML karena SOAP umumnya menggunakan protokol HTTP sebagai sarana *transportasi* datanya dan data yang akan dipertukarkan ditulis dengan format XML. Karena SOAP menggunakan HTTP dan XML, maka SOAP memungkinkan pihak-pihak yang berbeda dapat saling

mempertukarkan datanya (Priyambodo, 2005).

Identifikasi kandidat operasi business *Service* Langkah ini bertujuan untuk melakukan identifikasi dari langkah– langkah yang terdapat pada proses bisnis yang fungsi *logiknya* dapat dienkapsulasi ke dalam kandidat operasi dari business *Service*. Kandidat operasi *Service* pada proses bisnis yang dapat dikategorikan sebagai proses *logic* yang potensial untuk dienkapsulasi oleh kandidat *Service* ditentukan oleh dua kriteria:

1. Langkah tersebut tidak termasuk langkah yang dilakukan secara manual dan tidak dapat atau tidak boleh diautomatisasi.
2. Langkah tersebut sudah dilakukan oleh sistem yang sudah ada (*legacysystem*).

Berdasarkan hasil identifikasi kandidat operasi business *Service* dari workflow didapatkan *list* dari kandidat operasi *business Service*, Berdasarkan proses bisnis maka di dapat Pemetaan kandidat *Service* di IKMI Pekan baru seperti yang disajikan pada Tabel 1 berikut :

**Tabel 1. Pemetaan kandidat *Service* IKMI Pekan baru**

N O	PROSES BISNIS	KANDIDAT <i>SERVICE</i>
1.	Registrasi masjid	<i>Service</i> pengolahan data masjid
2.	Penusun antemada kwah	<i>Service</i> pengolahan antemada kwah
3.	Jadwal pembinaan masjid	<i>Service</i> pengolahan data masjid
		<i>Service</i> pengolahan data muballigh
		<i>Service</i> pengolahan jadwal pembinaan
4.	Registrasi muballigh	<i>Service</i> pengolahan data muballigh
6.	Jadwal pengajian rutin	<i>Service</i> pengolahan jadwal pengajian
		<i>Service</i> pengolahan data muballigh
		<i>Service</i> pengolahan data masjid
7.	Jadwal pengajian	<i>Service</i> pengolahan jadwal

	majlistalim	walpengajianmajelistalim
		Servicepengolahan data muballigh
		Servicepengolahan data masjid
8.	Jadwalpembrantas anbutaaksara	Servicepengolahanjadwal
		Servicepengolahan data muballigh
		Servicepengolahan data masjid
9.	Penyusunanjadwalkhutbahjumat	Servicepengolahanjadwalkhutbahjumat
		Servicepengolahan data masjid
		Servicepengolahan data muballigh
10	Laporan	

Berdasarkan proses bisnis maka di dapat Pemetaan kandidat *Service* di DDIPekanbaru seperti yang disajikan pada Tabel 2 berikut :

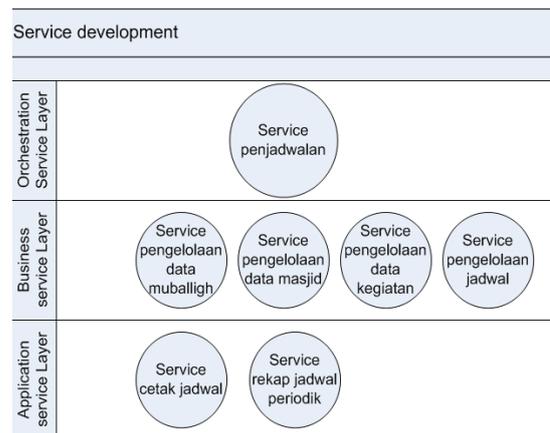
**Tabel 2. Pemetaan kandidat *Service*DDI Pekanbaru**

N O	PROSES BISNIS	KANDIDAT <i>SERVICE</i>
1	Pembinaanmuallaf	Servicepegolahan data survey Sevicepengolahan data muballigh Servicepengolahanjadwalpembinaanmuallaf
2	Pembinaandesatertingal	Servicepengolahan data survey Servicepengolahandana Sevicepengolahan data muballigh Servicepengolahanjadwalpembinaandesatertingal
3	Jadwalpembinaanbutaaksara	Servicepengolahanjadwalmengajaralqurandanarabmelayu Sevicepengolahan data muballigh SevicepengolahanjadwalMasjidAn-Nur
4	Jadwalpembinaanlapas	Servicepengolahan data survey Servicepengolahan data muballigh Servicepengolahanjadwalkegiatan di lapas

5	Jadwalpembinaaanmuallafdesatertinggal	Servicepengolahan data survey Servicepengolahan data muballigh Servicepengolahanjadwalkegiatan di desa
6	Laporan	

## PEMETAAN KANDIDAT *SERVICE*

Setelah beberapa kandidat *Service* diperoleh maka selanjutnya dibuat pemetaan terhadap seluruh kandidat *Service* tersebut untuk dibagi kedalam kelompok layer masing-masing. Dalam hal ini dibuat 3 layer yakni *Application layer*, *Bussines Layer* dan *Orchestration Layer*. Berikut Gambar 2 tentang pemetaan masing-masing kandidat *Service*.



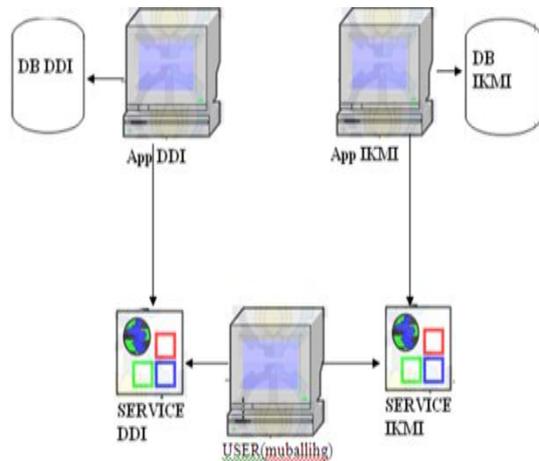
**Gambar2.PemetaanKandidat*Service***

Untuk kandidat *Service* *Service* Cetak Jadwal dan *Service* Rekap Jadwal Periodik dimasukkan kedalam *Application Layer* karena *Service* yang berhubungan langsung dengan pengguna. Sedangkan untuk kandidat *Service* *Service* Pengelolaan data Muballiq, *Service* Pengelolaan data Mesjid, *Service* Pengelolaan data Kegiatan dan *Service* Pengelolaan Jadwal dimasukkan kedalam *BussinesLayer* karena *Service-Service* tersebut merupakan *Service* yang bekerja pada sistem secara langsung. Pada layer paling atas yakni *Orchestration Layer* merupakan kumpulan dari beberapa *Service* menjadi satu kesatuan sistem web *Service*.

Dalam hal ini yang menjadi komponen dari layer tersebut adalah *Service* Penjadwalan.

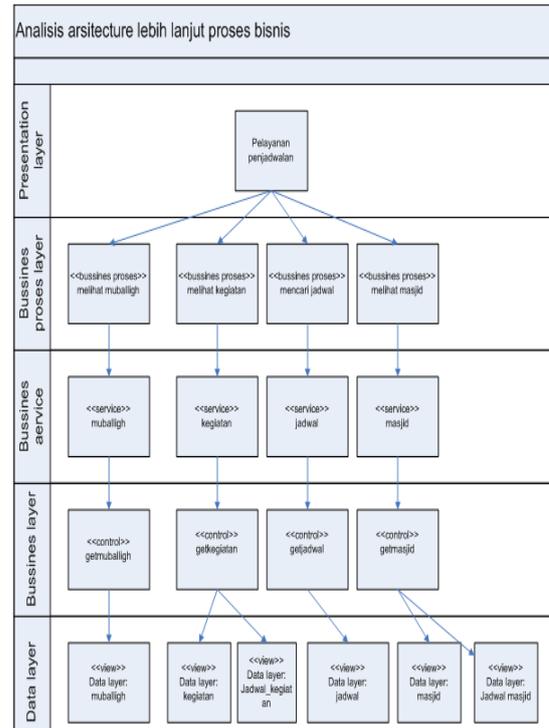
### IDENTIFIKASI ARSITEKTUR SISTEM

Arsitektur sistem yang digunakan berbasis *Client-Server*. Dalam hal ini server merupakan aplikasi yang memiliki *webService* untuk melayani semua permintaan (*request*) data yang dibutuhkan oleh *Cient*. Berikut Gambar 3 yang menerangkan tentang arsitektur sistem yang dibuat.



**Gambar3.Arsitektur Sistem**

Setiap adanya permintaan dari *Client* maka akan dilayani oleh *web Service* yang ada pada server. *Web Service* tersebut bekerja secara efektif untuk memberikan data permintaan *client* yang dimiliki oleh server tanpa mengganggu kinerja server. Dengan adanya *web Service* tersebut dapat menjalankan sistem yang berbeda aplikasi, baik antara aplikasi yang ada di server maupun di *client*. Berikut Gambar 4 yang menjelaskan detail tentang arsitektur sistem yang dibuat.



**Gambar4. Identifikasi Arsitektur Sistem**

Pada identifikasi Arsitektur Sistem tersebut dibagi kedalam 5 layer, yakni Presentation Layer yakni inti dari *Service* yang dibuat, *Bussines Process Layer* yakni fungsi dari *Service*, *BussinesService* yakni *Service-Service* yang akan dibuat untuk menangani setiap permintaan dari *client*, *Bussines Layer* merupakan fungsi dari *Service* untuk pemanggilan data, *Data Layer* yakni data yang akan diambil oleh *Service* untuk diteruskan ke *client* yang memintanya.

### HASIL DAN PEMBAHASAN

Adanya beberapa *Service* yang dibuat untuk aplikasi di server dalam melayani setiap permintaan *client* akan mempermudah dan mempercepat proses tersebut tanpa harus memperhatikan jenis aplikasi yang digunakan pada masing-masing server atau *client*. *Web Service* dibuat untuk menjembatani dari beberapa teknologi yang berbeda. Selain itu juga *Service* yang dibuat terdiri dari modul-modul yang terdapat pada aplikasi lain dan dapat dikembangkan untuk digunakan kembali menjadi sistem baru. Hal ini dikarenakan *Service* yang dibangun tersebut berbasis SOA dengan tingkat *reusability* yang tinggi.

## KESIMPULAN

Penerapan aplikasi berbasis SOA dengan web *Service* mempermudah setiap penanganan permintaan dari *client*. Hal ini dikarenakan setiap permintaan ditangani oleh *Service-Service* yang telah dibuat sesuai dengan kebutuhan yang ada. Perbedaan teknologi yang selama ini menjadi penghambat pertukaran data kini dapat diselesaikan dengan adanya web *Service* tersebut. Teknologi berbasis SOA ini akan berkembang seiring dengan kebutuhan data serta perkembangan teknologi yang beragam. Arsitektur sistem dihasilkan sebagai proyeksi sistem dalam menjalankan *Service-Service* yang telah dibuat dan dapat dikembangkan pada aplikasi yang lain. Pengiriman informasi pada aplikasi yang berbeda dengan memanfaatkan web *Service* menjadi solusi yang baik dari kendala yang dihadapi oleh pengurus IKMI selama ini.

## DAFTAR PUSTAKA

**Erl, Thomas**, (2009), "*SOA : Principles of Service Design*", Prentice Hall

**Krafzig, D., Karl Banke, Dirk Slama**. (2004). *Enterprise SOA: Service-Oriented Architecture Best Practices*. USA: Prentice Hall.

**Mabrouk, Mohamed I.**(2008). "*SOA fundamentals in a nutshell*". IBM Developer Works [available online <http://www.ibm.com/developerworks/webServices/edu/ws-dw-ws-soa-ibmcertified.html>], diakses tanggal 3 Maret 2013.

**Priyambodo, T. K.**(2005). "*Implementasi web-Service untuk pengembangan layanan pariwisata terpadu*". Jurnal Teknologi

**D. Austin, A. Barbir, C. Ferris, S. Garg**, *Web Services Architecture Requirements*, W3C Web Services Architecture Group, [available online <http://www.w3.org/TR/2004/NOTE-wsa-reqs-20040211>], diakses maret 2013

**Bieberstein, et al.** (2008). *Executing SOA: A Practical Guide for The Service Oriented Architecture*. IBM Press, Indiana.

**Brown, Paul C.** (2008). *Implementating SOA : Total Architecture in Practice*. Addison Wesley Profesional, Massachusetts.

**Hasan, Jeffrey.** *Expert Service-Oriented Architecture in C# Using the Web Services Enhancements 2.0*. Appress, New York.

**Turban E., Rainer, R. K. Jr., Potter, R. E.** (2006). *Pengantar Teknologi Informasi*. Edisi 3. Penerbit Salemba Infotek, Jakarta.

**Whitten, J. L., Bentley, L. D., Dittman, K. C.** (2004). *Systems Analysis and Design Methods*. 6<sup>th</sup> Edition. McGraw-Hill, New York.

**Durvasula, S.** (2006). *SOA Practitioner's Guide Part 3, Introduction to Services Lifecycle*.