

Analisis dan Perancangan Sistem *Mediation* dengan Protokol Soap pada *Web Service* untuk Mengintegrasikan Antar Sistem Informasi yang Berbeda *Platform*

Muhamad Femy Mulya¹, Nofita Rismawati²

¹ Faculty of Information System, Tanri Abeng University, Jakarta, Indonesia

² Program Studi Teknik Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia

femy.mulya@tau.ac.id

novi_9001@yahoo.co.id

Diterima 30 Oktober 2017

Disetujui 20 Desember 2017

Abstract— Web service is actually a collection of functions and methods contained on a server that can be called by the client remotely, then to call those methods we are free to use applications that will be created with any programming language that run on any platform. The existence of web service technology can bridge the technological differences of each source. The purpose of this system design is to build a mediation system that can integrate different information systems platforms, databases and functions, so that can communicate between systems. Software development methods used include three main parts of the method of analysis, design and literature study. Methods of analysis include the collection of data using samples, learning on existing systems, and analysis of the results of interviews aimed at the parties concerned. Design method using Unified Modeling Language (UML) approach. The result is a system-based mediation system design using the SOAP (Simple Object Access Protocol) protocol.

Index Terms— *Web Service, Mediation*

I. PENDAHULUAN

Web service merupakan sebuah sistem perangkat lunak yang dirancang untuk mendukung interoperasi dalam interaksi mesin ke mesin melalui sebuah jaringan. Interaksi dilakukan melalui sebuah mekanisme atau protokol tertentu [1]. *Web service* juga memiliki sebuah antarmuka yang mendeskripsikan seluruh layanan yang tersedia dalam format yang dapat diproses oleh mesin, yaitu *Web Service Description Language* (WSDL). Sistem-sistem lain yang nantinya akan berinteraksi dengan *web service* dengan mengacu pada antarmuka tersebut akan melalui suatu protokol, yaitu *Simple Object Access Protocol* (SOAP).

SOAP (*Simple Object Access Protocol*) adalah standar untuk bertukar pesan-pesan berbasis XML melalui jaringan komputer atau sebuah jalan untuk program yang berjalan pada suatu sistem operasi (OS)

untuk berkomunikasi dengan program pada OS yang sama maupun berbeda dengan menggunakan HTTP dan XML sebagai mekanisme untuk pertukaran data, maka SOAP dapat berkomunikasi dengan berbagai aplikasi meskipun terdapat perbedaan sistem operasi, teknologi, dan bahasa pemrogramannya [1].

Dengan adanya teknologi *web service* dapat menjembatani perbedaan-perbedaan teknologi dari masing-masing sumber. Dengan demikian, *web service* merupakan kumpulan layanan yang disediakan melalui jaringan berbasis web dengan standar yang telah ditetapkan dan mampu menunjang interoperabilitas, serta dapat berjalan diberbagai *platform* dan *framework*.

Dari uraian yang telah diberikan, melalui penelitian ini, akan dikembangkan suatu sistem *mediation* berbasis web dengan menggunakan protokol SOAP (*Simple Object Access Protocol*) yang dapat mengintegrasikan antar sistem informasi yang berbeda platform dan database serta dapat di kustomisasi untuk *interface inbound* (interface yang masuk ke sistem), *interface outbound* (interface yang keluar dari sistem) dan *rules* (kondisi) untuk setiap *interface*-nya.

Penelitian ini penting dilakukan, karena pada penelitian-penelitian yang sudah dilakukan hanya sebatas integrasi antar sistem informasi secara langsung tanpa adanya sistem *mediation*, kemudian adanya keterbatasan untuk kustomisasi setiap *interface*, serta tidak adanya parameter uji coba pengiriman data antar sistem informasi dalam bentuk persentase.

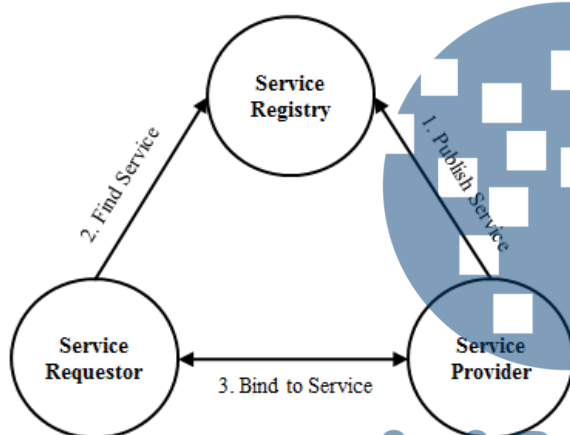
Dengan adanya sistem *mediation*, diharapkan dapat dimanfaatkan untuk mengintegrasikan antar sistem informasi yang berbeda platform, database dan fungsi, sehingga menjadi satu kesatuan sistem yang saling terhubung dan terdistribusi, serta dapat

membantu developer dalam mengembangkan sistem yang lebih *advanced* di masa depan. Perangkat lunak akan dikembangkan menggunakan bahasa pemrograman Java berbasis *web based* dan Apache Tomcat sebagai *web server*.

II. TINJAUAN PUSTAKA

A. Definisi dan Arsitektur Web Service

Web service sebenarnya adalah kumpulan dari fungsi dan metode yang terdapat pada sebuah server yang dapat dipanggil oleh klien dari jarak jauh, kemudian untuk memanggil metode-metode tersebut kita bebas menggunakan aplikasi yang akan dibuat dengan bahasa pemrograman apa saja yang dijalankan pada *platform* apa saja. Adanya teknologi *web service* dapat menjembatani perbedaan-perbedaan teknologi dari masing-masing sumber. Berikut adalah gambaran arsitektur *web service* [1].



Gambar 1. Arsitektur *web service* [1]

Web Service memiliki tiga entitas dalam arsitekturnya, yaitu [2]:

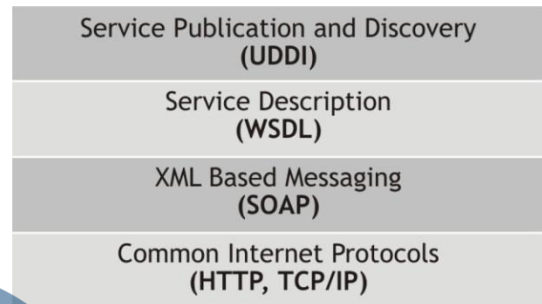
1. *Service Requestor* (peminta layanan), berfungsi untuk menyediakan layanan / *service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
2. *Service Provider* (penyedia layanan), berfungsi sebagai lokasi terpusat yang mendeskripsikan semua layanan / *service* yang telah di-*register*.
3. *Service Registry* (daftar layanan), berfungsi sebagai peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Secara umum, *web service* memiliki tiga operasi yang terlibat di dalamnya, yaitu [3]:

1. *Publish / Unpublish*, berfungsi untuk menerbitkan / menghapus layanan ke dalam atau dari *registry*.

2. *Find*, berfungsi sebagai *service requestory* yang mencari dan menemukan layanan yang dibutuhkan.
3. *Bind*, berfungsi sebagai *service requestor* yang setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *service provider* untuk melakukan interaksi dan mengakses layanan / *service* yang disediakan oleh *service provider*.

B. Komponen-Komponen Web Service



Gambar 2. Komponen-Komponen *Web Service* [4]

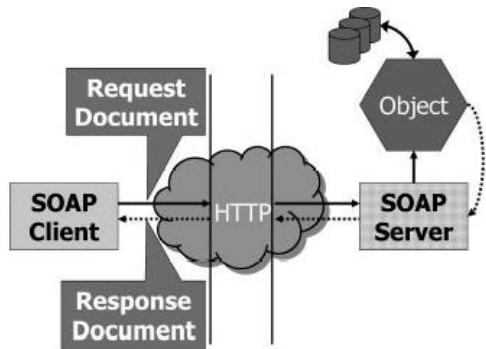
1. Layer 1 : Protokol internet standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.
2. Layer 2 : *Simple Object Access Protocol* (SOAP) adalah protokol ringan untuk XML sehingga dapat digunakan untuk menganalisa informasi dari *request* dan *response* pada *web service* sebelum dikirimkan melalui jaringan.
3. Layer 3 : *Web Service Definition Language* (WSDL) adalah sebuah bahasa berbentuk XML yang digunakan untuk mendeskripsikan kemampuan dari *web service* sebagai kumpulan dari komunikasi-komunikasi yang saling bertransaksi pesan.
4. Layer 4 : *Universal Description Discovery and Integration* (UDDI) adalah sebuah direktori yang didistribusikan secara *web based* sehingga dapat mendaftarkan diri ke internet sehingga dapat dijelajahi.

C. SOAP (*Simple Object Access Protocol*)

Sebenarnya peran SOAP di dalam teknologi *web service* sebagai protokol pemaketan untuk pesan-pesan (*messages*) yang digunakan secara bersama oleh aplikasi-aplikasi penggunaannya serta menspesifikan secara jelas bagaimana cara untuk meng-*encode header* HTTP dan *file XML* sehingga program pada suatu komputer dapat memanggil program pada komputer lain dan mengirimkan informasi, dan bagaimana program yang dipanggil memberikan tanggapan [5].

Protokol HTTP merupakan protokol yang paling sering digunakan dalam mengirimkan SOAP *message*. SOAP mendeskripsikan model pertukaran *message*

khusus untuk di HTTP. Dengan begitu cara ini cocok dengan model SOAP RPC (*request-response*) dikarenakan HTTP merupakan protokol bertipe *request-response* [6].



Gambar 3. Ilustrasi SOAP HTTP Request-Response[6]

Pesan SOAP *request* di *post* ke server dengan HTTP *request*, kemudian server menjawab dengan HTTP *response* yang berisi pesan SOAP *response*. Contoh HTTP *Request* yang mengandung SOAP *message* [7]:

```
POST /StockQuote HTTP/1.1
Content-Type: text/xml
Content-Length: nnnn
SOAPAction: "urn:StockQuote#GetQuote"

<s:Envelope xmlns:s="http://www.w3.org/2001/06/soap-envelope">
...
</s:Envelope>
```

Gambar 4. HTTP Request

Berikut adalah contoh HTTP *Response* yang mengandung SOAP *message* [7]:

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn
<s:Envelope xmlns:s="http://www.w3.org/2001/06/soap-envelope">
...
</s:Envelope>
```

Gambar 5. HTTP Response

III. METODE PENELITIAN

Untuk merancang sistem *mediation* dengan protokol SOAP pada *Web Service* dan mengintegrasikan antar Sistem Informasi yang berbeda *platform*, *database* dan fungsi dengan benar dan sesuai dengan konsep yang diperoleh, maka pada penelitian ini akan mengikuti tahapan siklus hidup pengembangan sistem yang terdiri dari :

- Perencanaan Sistem (*Systems Planning*).
- Analisis Sistem (*System Analysis*).
- Perancangan Sistem (*Systems Design*) Secara Umum.
- Seleksi Sistem (*System Selection*).

- Perancangan Sistem (*Systems Design*) Secara Terinci.
- Implementasi dan Pemeliharaan Sistem (*System Implementation and Maintenance*).

IV. HASIL DAN PEMBAHASAN

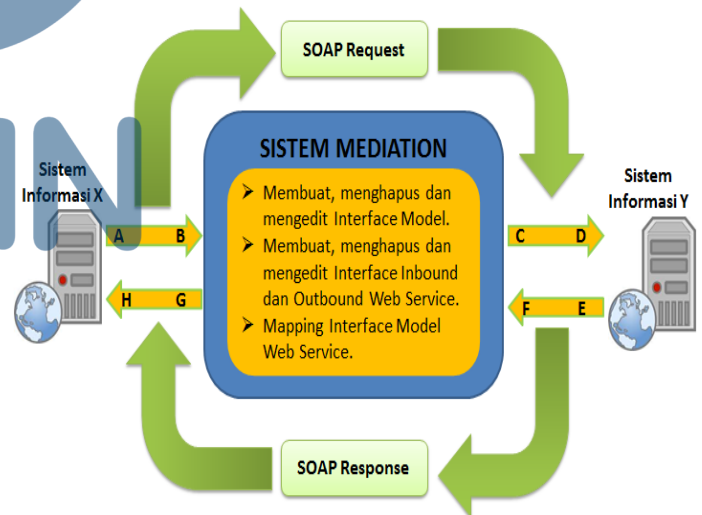
A. Analisa Permasalahan

Berikut adalah beberapa permasalahan yang diidentifikasi dan dianalisis, serta beberapa diantaranya direalisasikan dalam penelitian ini:

1. Bagaimana cara berkomunikasi antar Sistem Informasi yang berbeda platform, database dan fungsi.
2. Bagaimana proses mapping XML pada Sistem Informasi yang berbeda platform, database dan fungsi.
3. Bagaimana cara mengenali antara SOAP request dan SOAP response Pada Sistem Informasi yang berbeda platform, database dan fungsi.
4. Berapa persentase akurasi pengiriman data untuk sistem *mediation* ini.

B. Usulan Solusi

Dari permasalahan yang didapat, maka dirancang solusi seperti gambar berikut:



Gambar 6. Rancangan Sistem Mediation dengan protokol SOAP pada Web Service

Pada Gambar 6 di atas merupakan sebuah rancangan Sistem *Mediation* antara Sistem Informasi X dan Sistem Informasi Y dengan protokol komunikasi menggunakan SOAP (*Simple Object Access Protocol*). Adapun penjelasan lebih rincinya sebagai berikut.

1. Pada sistem *mediation* terdapat *interface model* yang berfungsi untuk membuat model dari

elemen/data SOAP pada setiap *interface inbound* (interface yang masuk ke sistem) dan *interface outbound* (interface yang keluar dari sistem). Adapun Penjelasan *interface inbound* dan *outbound* pada gambar diatas, sebagai berikut:

Tabel 1. *Interface Inbound dan Outbound*

Interface	Type of Interface		Interface Owner
	Inbound	Outbound	
A		√	Sistem Informasi X
B	√		Sistem Mediation
C		√	Sistem Mediation
D	√		Sistem Informasi Y
E		√	Sistem Informasi Y
F	√		Sistem Mediation
G		√	Sistem Mediation
H	√		Sistem Informasi X

2. Pada sistem *mediation* juga terdapat SOAP *Request* (A,B,C,D) berfungsi sebagai XML yang di *post* ke *server* dengan HTTP *Request*. Dan SOAP *Response* (E,F,G,H) berfungsi sebagai pesan SOAP *Response / Return Value Message*.

3. Pada proses *Mapping Interface Model* di definisikan sebagai berikut:

- XML *Interface Model A Outbound* akan di *mapping* dengan *Interface Model B Inbound*.
- XML *Interface Model B Inbound* akan di *mapping* dengan *Interface Model C Outbound*.
- XML *Interface Model C Outbound* akan di *mapping* dengan *Interface Model D Inbound*.
- XML *Interface Model E Outbound* akan di *mapping* dengan *Interface Model F Inbound*.
- XML *Interface Model F Inbound* akan di *mapping* dengan *Interface Model G Outbound*.
- XML *Interface Model G Outbound* akan di *mapping* dengan *Interface Model H Inbound*.
- Proses *Mapping Interface Model* bisa dilakukan di Sistem *Mediation* dengan bantuan *Web Interface*.
- Untuk *Filtering Rule* dan *Condition*, dapat dilakukan pada *Interface Model* untuk *Inbound* maupun *Outbound*.

4. Pada sistem *mediation* ini juga *support* dengan protokol http biasa maupun https, hanya saja pada https diperlukan *keystore* dengan ekstensi *.jks sebagai proses *authentication* antar *interface model*.

C. Uji Coba Akurasi Pengiriman Data

Pada penelitian ini dikembangkan suatu *prototype* untuk sistem *mediation*, sehingga dapat diketahui persentase tingkat akurasi pengiriman data setiap *interface inbound* maupun *interface outbound* yang melalui sistem *mediation*. Untuk menghitung persentase akurasi pengiriman data, dilakukan uji coba *sampling* pengiriman data sebanyak 100 kali *sampling* pengiriman data untuk setiap *interface*-nya, maka dari hasil uji coba didapat hasil persentase tingkat akurasi pengiriman data sebesar 99,75% untuk pengiriman data yang sukses dan 0,25% untuk pengiriman data yang gagal.

Tabel 2. *Uji Coba Akurasi Pengiriman Data*

Interface Prototype	Type of Interface				Interface Owner
	Inbound (Uji Coba 100 Kali Pengiriman Data)		Outbound (Uji Coba 100 Kali Pengiriman Data)		
	Data Sukses	Data Gagal	Data Sukses	Data Gagal	
A			99	1	Sistem Informasi X
B	100	0			Sistem Mediation
C			100	0	Sistem Mediation
D	99	1			Sistem Informasi Y
E			100	0	Sistem Informasi Y
F	100	0			Sistem Mediation
G			100	0	Sistem Mediation
H	100	0			Sistem Informasi X
Total (%)	99,75%	0,25%	99,75%	0,25%	

V. PENUTUP

A. Kesimpulan

Beberapa kesimpulan yang dapat ditarik antara lain:

1. Sistem *mediation* ini dapat membantu mengintegrasikan dua atau lebih Sistem Informasi yang berbeda *platform*, *database* dan fungsi, sehingga menjadi satu kesatuan sistem yang saling terhubung dan terdistribusi dengan tingkat akurasi pengiriman data 99,75% dari hasil uji coba akurasi pengiriman data yang sukses terkirim antar sistem informasi.
2. Dengan adanya desain ini, dapat sekaligus memperkenalkan teknologi *mediation* pada *developer software*.

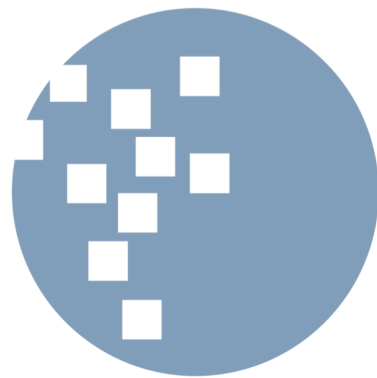
B. Saran

Mengingat teknologi ini tergolong baru maka desain ini tidak terlepas dari beberapa kekurangan, beberapa saran yang patut dipertimbangkan untuk proses pengembangan sistem ini antara lain :

1. Menggunakan skema enkripsi dan dekripsi pada setiap *interface inbound* dan *outbound*, sehingga data yang dikirim memiliki tingkat keamanan yang tinggi dan dapat terhindar dari pencurian data.
2. Pengembangan sistem *mediation* dengan memanfaatkan *cloud system* seperti: *google cloud*, *amazon cloud*, dan lainnya.

DAFTAR PUSTAKA

- [1] FX. Rusiawan, "Penerapan Web Services dalam Konteks Enterprise Application Integrateion (EAI) pada Industri Proses", Tesis, ITB Central Library, Bandung, 2004.
- [2] A.S. Girsang, "Pemanfaatan web untuk Integrasi Data pada Sistem Informasi Eksekutif (Studi Kasus : Universitas Janabadra Yogyakarta)", Tesis, S2 Ilmu Komuter FMIPA UGM, Yogyakarta, 2006.
- [3] A. Ahmadi, "Implementasi Web Services pada Content Management Sistem", Tesis. ITB Central Library, Bandung, 2007.
- [4] D. Huang, "Integrating Web Services with competitive Strategies: The Ballanced Scorecard Approach", 2004.
- [5] W. Siswoutomo, "Membangun Web Service Open Source Menggunakan PHP", PT Gramedia, Jakarta, 2004.
- [6] G. Booch, R. Maksimuchuk, M. Engle, B. Young, J. Conallen, K. Houston, "Object Oriented Analysis and Design with Aplication", Pearson Education Corporate, Boston, 2007.
- [7] S. Boy, "Sistem Informasi Manajemen Rumah sakit", Editor Berkat, Tim Konsorsium, Cetakan 1, Yogyakarta, 2003.
- [8] M.C. Daconta, L.J. Obrst, K.T. Smith, "The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management", John Willey, 2005.
- [9] B. Santosa, "Solusi XML Web Services untuk Sistem Informasi Universitas, UPN Veteran Yogyakarta", Yogyakarta, 2007.
- [10] B.S. Stendy, "Aplikasi Web dengan XML menggunakan Dream Weaver 8", Andi Offset, Yogyakarta, 2007.
- [11] M. Fowler, "UML Distilled, A Brief Guide To The Standart Object Modelling Language", Pearson education Corporate, Boston, 2003.



UMN