

PEMANFAATAN WEB SERVICES PADA INTEGRASI DATA FARMASI DI RSU BANYUMAS

Riyanto¹⁾, Ema Utami²⁾, Armadiyah Amborowaty³⁾

^{1,2,3)}Magister Teknik Informatika STMIK AMIKOM Yogyakarta

e-mail : riyanto_0215@yahoo.co.id, ema.u@amikom.ac.id, armadiyah.a@amikom.ac.id

Abstrak

Rumah Sakit Umum Banyumas merupakan Rumah Sakit milik Pemerintah Kabupaten Banyumas. Salah satu pelayanan yang terpenting adalah pelayanan di Farmasi, untuk mempermudah dan memperlancar kinerjanya maka diperlukan adanya sistem informasi pengolahan data. Kendala yang terdapat di Farmasi, yakni untuk pelayanan resep pasien dengan debitur petugas Farmasi harus melakukan entri data pada Sistem Informasi Apotek dan Sistem Informasi Rumah Sakit. Oleh sebab itu, perlu dilakukan perbaikan dalam pengelolaan sebuah sistem informasi dengan menggunakan web services sebagai integrasi data. Pemanfaatan teknologi internet telah mewarnai perkembangan sistem distributed computing dengan munculnya konsep mengenai web services. web services merupakan paradigma baru dalam mengimplementasikan sistem terdistribusi melalui web yang menggunakan basis teknologi XML, dengan standard protokol HTTP dan SOAP. Mengingat standard terbuka yang digunakan dalam mendukung teknologi web services, maka implementasi web services menjanjikan banyak kemudahan dan perbaikan dalam mendukung integrasi berbagai platform sistem dan aplikasi, baik melalui infrastruktur intranet maupun internet atau ekstranet. Tahapan yang dilakukan mulai dengan planning, analysis, design, implementation, dan testing and evaluation, sehingga menghasilkan rancangan model analisis services berbasis obyek dan aplikasi web services sebagai integrasi data Farmasi di Rumah Sakit Umum Banyumas sehingga tidak diperlukan penggantian sistem yang lama.

Kata Kunci : web services, XML, integrasi data Farmasi, SOAP, HTTP

1. Pendahuluan

Sistem terintegrasi merupakan tantangan yang menarik dalam *software development* hal tersebut dikarenakan pengembangannya harus terus mengacu pada konsistensi sistem, agar sub-sub sistem yang sudah ada dan tetap dimanfaatkan secara operasional sehingga masih tetap berfungsi sebagaimana mestinya, baik ketika proses mengintegrasikan sistem maupun setelah terintegrasi. *Web services* menggunakan konsep *remote procedure call* (RPC) dimana format data yang dipertukarkan antara web service dengan web atau aplikasi pengaksesnya adalah *simple object access protocol* (SOAP). SOAP digunakan untuk mengakses objek terdistribusi dengan mengirimkan pesan XML. SOAP bekerja pada *protocol* komunikasi tingkat rendah yaitu *protocol* HTTP.

Sistem informasi yang ada di antara unit-unit organisasi atau di Rumah Sakit Umum Banyumas harus dapat berhubungan dan berkomunikasi dengan baik. Sasaran dari sistem yang terintegrasi adalah untuk menyediakan informasi yang akurat tepat waktu serta relevan terhadap semua komponen. Integrasi sistem informasi dapat dicapai dengan aplikasi yang terintegrasi pada unit Farmasi dan Rumah Sakit, semua data akan terkumpul menjadi satu pada database di suatu server yang telah disiapkan dan siap diakses oleh pengguna saat dibutuhkan. Dengan sistem terintegrasi ini maka input data yang telah dimasukkan oleh bagian atau unit lain tidak dimasukkan lagi oleh unit lain yang berbeda. Komunikasi antar unit bisa saling terhubung serta dapat mengakses data secara terpusat (*Server*) untuk dapat memenuhi kebutuhannya. Bertitik tolak dari hal tersebut, pada penelitian ini akan merancang model aplikasi integrasi data Farmasi menggunakan web services dengan metode *simple obyek access protocol*.

2. Tinjauan Pustaka

Ada beberapa komponen yang harus diperhatikan dalam pemanfaatan web services sebagai integrasi data Farmasi, komponen tersebut berfungsi dapat menghasilkan *software* yang berkualitas. Adapun komponen yang perlu diperhatikan tersebut diantaranya:

- *Web Services*
- XML (*eXtensible Markup Language*)
- SOAP (*Simple Object Application Protocol*)
- WSDL (*Web service Description Language*)
- UDDI (*Universal Description Discovery, and Integration*)
- Penelitian Terkait

2.1 Web services

Menurut Michael, S. I. dan Purba, J., (2002), *web services* merupakan komputasi yang dapat diakses melalui jaringan internet maupun *intranet* dengan standar *protocol* tertentu dalam *platform* dan antarmuka bahasa pemrograman yang independen. Tujuan *web services* pengembangan adalah untuk "menjembatani komunikasi antar program", sehingga aplikasi yang satu dengan aplikasi yang lain yang terdapat pada jaringan yang sama atau pada jaringan yang berbeda dapat saling berkomunikasi dengan menggunakan standar *protocol* yang ditetapkan oleh *web services*.

2.2 XML

Menurut Suprianto, A. (2007), XML (*eXtensible Markup Language*) merupakan suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa markup adalah mekanisme untuk mengenal suatu struktur dokumen. Tujuan dari XML secara umum untuk memungkinkan SGML untuk membantu, menerima dan memproses pada program web, dimana cara ini dapat dilakukan dengan HTML. XML mempunyai tujuan memungkinkan XML berfungsi sebagai suatu format pertukaran data (*data exchange*) antar dua entitas, XML juga dapat berperan dalam penyimpanan dokumen agar dapat digunakan oleh program lain, karena formatnya standar dan mampu mendeskripsikan data.

2.3 SOAP

Menurut Short S. (2003), *Simple Object Access Protocol* (SOAP) merupakan *protocol* pertama dari jenisnya yang akan diterima secara praktis oleh setiap perusahaan perangkat lunak besar dunia. SOAP menyediakan cara standar pemaketan sebuah pesan. Pesan SOAP tersusun dari sebuah amplop yang berisi tubuh pesan dan informasi header yang akan dipakai untuk menerapkan pesan tersebut. Elemen root dari dokumen tersebut adalah elemen *envelope*. Dalam amplop tersebut, pesan SOAP yang sah dapat berisi elemen – elemen anak lainnya. Anatomi pesan SOAP mempunyai beberapa bagian diantaranya: *SOAP Actors*, *elemen header*, *atribut actor*, *elemen body*.

2.4 WSDL

Menurut Yasin, V. (2012), WSDL merupakan bahasa *standard* yang menyediakan mekanisme untuk mendeskripsikan *service* yang disediakan oleh sistem (*web service*), lokasi keberadaan *service* tersebut dan bagaimana cara memperolehnya, secara terstruktur dalam format XML. WSDL dapat dianalogikan sebagai IDL (*Interface definition language*) dalam CORBA dan COM. *Service* dideskripsi kan sebagai koleksi dari *entry-point* atau *port* komunikasi. WSDL mendeskripsikan *service* dengan menggunakan elemen sebagai berikut:

1. *Type* – tipe data yang digunakan sebagai *argument* dan *return type*
2. *Message* – merepresentasikan definisi data yang ditransmisikan
3. *Port type* – sekumpulan operasi yang didukung oleh satu atau lebih *end-point*
4. *Binding* – mendefinisikan protocol dan format pertukaran data untuk operasi yang didefinisikan oleh *port type*
5. *Port* – menspesifikasikan *end-point* yang digunakan untuk *binding*

2.5 UDDI

Menurut Short S. (2003), DDI (*Universal Description, Discovery, and Integration*) merupakan suatu layanan direktori hirarkis dan terpusat. UDDI menyediakan layanan direktori pusat untuk mempublikasikan informasi teknikal layanan web.

2.6 Penelitian Terkait

Implementasi model web services telah banyak dilakukan berikut contoh-contoh model web services yang pernah dikembangkan dalam berbagai kasus. Dalam penelitian Deviana, H. (2011) membahas penerapan XML *web services* pada sistem distribusi barang. Penelitian tersebut membahas sistem yang dikembangkan dalam penelitian adalah sistem berbasis web yang dirancang untuk berfungsi dalam menangani sistem informasi distribusi barang menggunakan *arsitektur web services*. Aplikasi pada sistem terbagi menjadi dua jenis aplikasi, yaitu : Aplikasi pada server apotek pusat dan Aplikasi pada server apotek cabang atau *outlet*. Metode yang digunakan untuk pengembangan sistem dalam penelitian ini adalah menggunakan metode *prototipe*.

Penelitian Theophilus Wellem (2009) membahas perancangan *prototype* aplikasi *mobile* untuk pengaksesan *web services*. membahas perancangan *prototype* aplikasi *mobile* menggunakan *Java Micro Edition* (Java ME) untuk mengakses web service. Untuk memungkinkan pengaksesa *web service* oleh aplikasi Java ME, digunakan *web*

services application programming interface (WSA) untuk Java ME, yang menyediakan fungsi untuk melakukan *parsing XML* dan *remote procedure call*. Sebagai contoh, *web services* yang diakses oleh aplikasi yang dirancang adalah *web service* pada suatu sistem informasi akademik. *Web service* ini mengembalikan indeks prestasi semester, indeks prestasi kumulatif, dan nilai suatu matakuliah berdasarkan parameter yang diberikan kepada *web services* tersebut oleh aplikasi. Hasil pengujian terhadap *prototype* aplikasi menunjukkan bahwa aplikasi bekerja sesuai dengan yang diharapkan. Adanya WSA membuat pemrograman *web services client* pada *mobile device* berbasis Java ME tidak perlu lagi melakukan hal-hal yang bersifat *low-level*, seperti manipulasi SOAP, HTTP, dan pemetaan tipe data antara tipe data WSDL dan tipe data pada java, karena semuanya telah ditangani oleh WSA.

Penelitian Afriyudi (2011) membahas *mobile database query* menggunakan teknologi *web services*. Dalam penelitian tersebut *web services* digunakan sebagai media penghubung antara *mobile service client* dengan *web services server* yang menyediakan layanan query dari J2ME dengan platform CLDC terhadap server yang menggunakan *web services* dengan teknologi SOAP.

Jazi Eko Istiyanto dan Edhy Sutanta (2012), membahas model interoperabilitas antar aplikasi E-Government. rancangan model *interoperabilitas* antar aplikasi e-Gov yang menggunakan model arsitektur *web services* menggunakan metode REST, terdiri dari tiga rancangan yaitu Rancangan *provider*, Rancangan *agent* atau broker dan Rancangan *requester*.

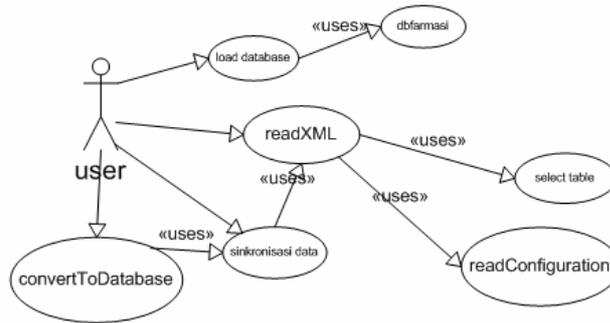
Edhy Sutanta dan Khabib Mustofa (2012), membahas kebutuhan *web services* untuk sinkronisasi data antar sistem informasi dalam E-Gov di Pemkab Bantul Yogyakarta. Pemkab Bantul yang merupakan bagian wilayah pemerintahan Propinsi DIY telah mengembangkan sebanyak 28 aplikasi sistem informasi untuk mendukung kegiatan administrasi, pengolahan data internal, dan layanan informasi public. Perancangan *provider* Penentuan nama sumberdaya informasi publik pada *web services*, yaitu struktur format parameter URI acuan sebagai deskripsi *web application description language* (WADL), dengan parameter: resource, format input (XML atau TXT), format output (XML, TXT, atau HTML), dan data input. Perancangan *agent* atau *broker*; yaitu menyediakan aplikasi *web services* untuk proses registrasi (*registry*) dan penemuan kembali (*discovery*) untuk memudahkan pengelolaan dan pencarian layanan dengan cara melakukan pencatatan dalam database. Perancangan *requester*; *requester* dapat menggunakan fungsi layanan setelah memperoleh perintah URL pada *browser* di *client*. Jika hasil dari fungsi layanan akan digunakan sebagai sumber data hasil pengolahan, maka perintah tersebut perlu dimasukkan ke dalam kode program *requester* melalui fungsi komunikasi dalam bahasa pemrograman (*socket*) atau pengolahan *file* jarak jauh (PHP, ASP, JSP).

3. Metode Penelitian

Makalah ini merupakan hasil dari kajian pustaka yang meliputi review konseptual mengenai *web services*, dilanjutkan dengan perencanaan (*planning*) yang dilakukan dalam proses pembangunan sistem, tahapan selanjutnya dengan melakukan kegiatan *analysis* terhadap sistem yang sedang berjalan untuk mendapatkan aktifitas fungsionalitas dari masing-masing *user*, *design* digunakan untuk menggambarkan bagaimana bisnis beroperasi, serta mengilustrasikan bagaimana data berpindah diantara aktifitas tersebut, *implementation* dilakukan dengan membuat perancangan *web services* kemudian membangun aplikasinya., dan *testing and evaluation* untuk mengetahui apakah pemanfaatan *web services* pada Integrasi data Farmasi di RSUD Banyumas sesuai dengan kebutuhan bisnis atau belum. Selanjutnya akan dilakukan evaluasi terhadap model yang sudah dibuat.

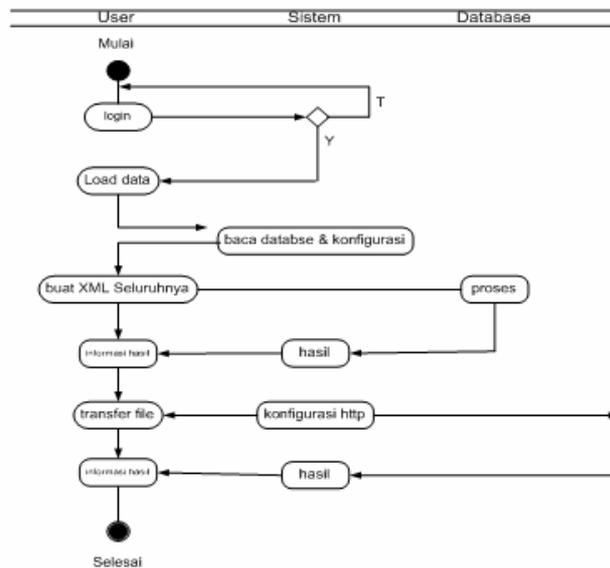
3.1 Rancangan Arsitektur Sistem

Diagram *use case* menunjukkan interaksi antara *use case* dan aktor untuk sistem *web service* sebagai integrasi data Farmasi. Aktor user menggunakan beberapa *use case*, diantaranya: load database Farmasi, konversi file, dan transfer file. Gambar *use case* dapat dilihat pada gambar 1



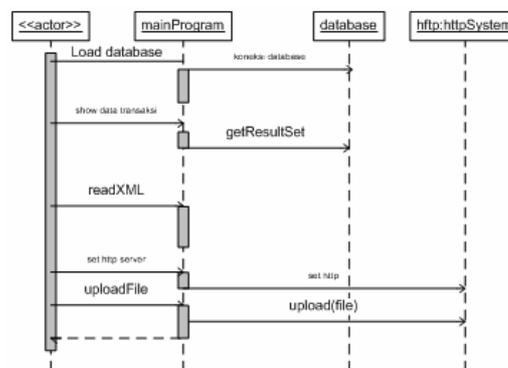
Gambar 1. Use Case

Activity diagram dimulai dengan load data dengan membaca ke database dan mengkonfigurasinya. Selanjutnya akan membuat file XML yang dilanjutkan dengan proses menampilkan hasil dari proses tersebut. Setelah itu akan dilakukan proses transfer file dengan konfigurasi dari http, jika berhasil sistem akan memberikan informasi. Activity diagram dapat dilihat pada gambar 2



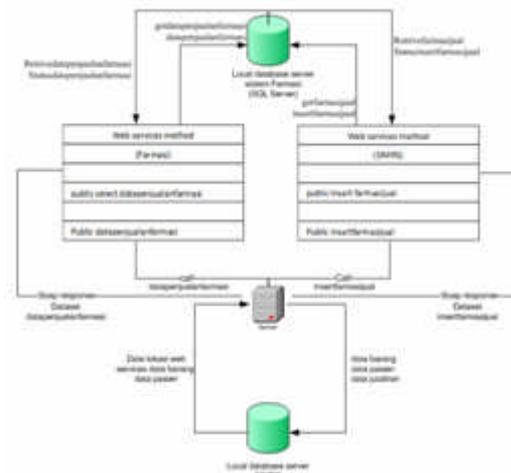
Gambar 2. Activity Diagram

Sequential Diagram memperlihatkan himpunan Use Case dan Aktor (jenis khusus dari kelas). Diagram ini penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. Sequential Diagram Load Database lebih rinci seperti terlihat pada Gambar 3



Gambar 3. Sequential Diagram

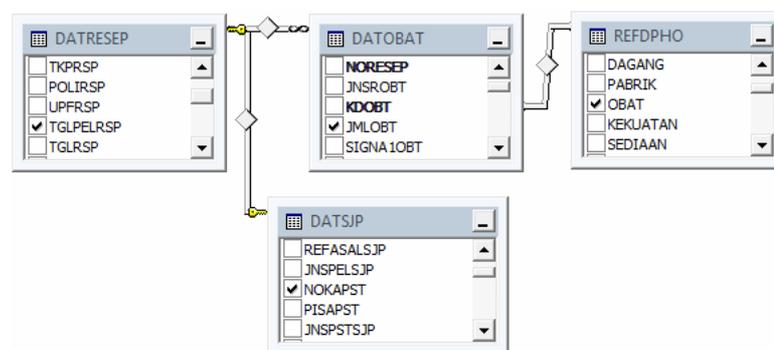
Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi. Berikut class diagram aplikasi load data, seperti terlihat pada Gambar 4



Gambar 7. Data Processing

3.2 Rancangan Basis Data

Database Farmasi yang dipergunakan sebagai data masukan pada database SIMRS mengambil dari tabel datresep, datobat, refdpho, datsjp (TGLPELRSP, JMLOBT, NOKAPST, OBAT, NOSJP). Data tersebut seperti terlihat pada Gambar 7



Gambar 8. Database Farmasi

Database SIMRS digunakan untuk proses penyesuaian data yang diperlukan, sebelum dimasukkan kedalam tabel jualtran. Data tersebut diambil dari tabel barang dan tabel pasien untuk mengambil kd_kel, kd_brg, harga, no_pasien. Data yang digunakan pada proses tersebut seperti pada Gambar 9



Gambar 9. Database SIMRS

4. Hasil dan Pembahasan

a. SOAP

Sebuah client web service mengirim request berbentuk XML kepada *provider web service*. Provider mem-*parsing request* tersebut, menjalankan *service*, dan mengirim *response* kembali ke client juga dalam bentuk XML. Baik *request* dan *response* tersebut keduanya menggunakan protokol SOAP. Service data penjualan farmasi akan mengembalikan data tanggal yang dimasukkan atau invoke. Gambar service tersebut seperti terlihat pada Gambar 11

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DataPenjualanFarmasi xmlns="http://tempuri.org/">
      <TANGGAL>string</TANGGAL>
    </DataPenjualanFarmasi>
  </soap:Body>
</soap:Envelope>
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DataPenjualanFarmasiResponse xmlns="http://tempuri.org/">
      <DataPenjualanFarmasiResult>
        <xsd:schema>schema</xsd:schema>xml</DataPenjualanFarmasiResult>
      </DataPenjualanFarmasiResponse>
    </soap:Body>
  </soap:Envelope>
    
```

Gambar 11. SOAP

b. Web Services

Dalam mengambil data penjualan obat dapat dilakukan dengan menekan tombol Lihat Data, pada tanggal pelayanan resep secara otomatis akan terlihat tanggal sesuai transaksi yang terjadi. Data tanggal tersebut tidak bisa dirubah yang bertujuan untuk menghindari terjadinya kesalahan pengambilan data. Proses pengambilan data dilakukan pada jam kerja siang setelah proses pelayanan selesai dilakukan, dengan harapan tidak mengganggu aktifitas di Farmasi sendiri. Setelah operator menekan tombol Lihat Data maka akan terlihat data yang disajikan melalui datagrid seperti terlihat pada Gambar 12

	NOSJP	JMLOBT	NOKAPST	TGLPELRSP	OBAT
▶	1111A0110909Y...	2.00	0000028655504	2/15/2013	Otsu-D40 25 ml Otsu
	1111A0110909Y...	15.00	0000028609727	2/15/2013	Antalgin 500 mg Cors
	1111A0110909Y...	30.00	0000028609727	2/15/2013	Zypraz 0.5 mg. Kalb
	1111A0110909Y...	15.00	0000028609727	2/15/2013	Vastigo 6 mg Dexa
	1111A0110909Y...	30.00	0000027344788	2/15/2013	Simarc 2 mg Prat
	1111A0110909Y...	30.00	0000027344788	2/15/2013	Letonal 25 mg Otto
	1111A0110909Y...	30.00	0000027344788	2/15/2013	Gralxa 40 mg Graf
	1111A0110909Y...	30.00	0000027344788	2/15/2013	Noperten 5 mg Dexa
	1111A0110909Y...	30.00	0000027344788	2/15/2013	Digoksin 0,25 mg Yari
	1111A0110909Y...	30.00	0000026357477	2/15/2013	Letonal 25 mg Otto
	1111A0110909Y...	30.00	0000026357477	2/15/2013	Gralxa 40 mg Graf
	1111A0110909Y...	90.00	0000026357477	2/15/2013	Farsorbid 5 mg Prat
	1111A0110909Y...	30.00	0000026357477	2/15/2013	Tensiphar 10 mg Acta
	1111A0110909Y...	15.00	0000026357477	2/15/2013	Biscor 5 mg Dexa
	1111A0110909Y...	15.00	0000026357477	2/15/2013	Valsartan NI 160 mg Nova
	1111A0110909Y...	30.00	0000026357477	2/15/2013	Miniasoi 80 mg. Mers

Gambar 12. Form Aplikasi Client

c. Pengujian

Tabel 1. Pengujian Fungsionalitas

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
1	User melakukan load database farmasi	Load database	Sistem menampilkan data transaksi sesuai tanggal transaksi	Sistem menampilkan data transaksi sesuai tanggal transaksi
2	User melakukan konversi file	Konversi file	Sistem melakukan konversi file dari database ke file XML	Sistem melakukan konversi file dari database ke file XML
3	User melakukan transfer file	Transfer file	Sistem dapat melakukan transfer file menggunakan konfigurasi http	Sistem dapat melakukan transfer file menggunakan konfigurasi http
4	User dapat membaca file XML	Read XML	Sistem menampilkan file xml dalam bentuk tabel	Sistem menampilkan file xml dalam bentuk tabel
5	User dapat melakukan sinkronisasi data dari dua database yang berbeda	Sinkronisasi data	Sistem dapat melakukan sinkronisasi database dari dua server yang berbeda	Sistem dapat melakukan sinkronisasi database dari dua server yang berbeda
6	User dapat melakukan convert database	Convert database	Sistem dapat melakukan convert database simrs	Sistem dapat melakukan convert database simrs

5. Kesimpulan

Berdasarkan implementasi dan pengujian didapatkan beberapa kesimpulan yakni Pembangunan web services sebagai integrasi data Farmasi di Rumah Sakit Umum Banyumas telah dapat diimplementasikan sesuai dengan perancangan. Penerapan metode *Simple Object Access Protocol* pada *web services* di *server* Farmasi dan *server* SIMR telah berhasil dibangun sebagai integrasi data Farmasi di RSUD Banyumas. Pemodelan *web services* dapat digunakan sebagai integrasi data dari sistem yang berbeda sehingga tidak diperlukan adanya sistem yang baru.

6. Daftar Pustaka

- Afryudi, 2011, *Mobile Database Query Menggunakan Teknologi Web service*, Universitas Bina Darma Palembang, Palembang
- Defiana, H., 2011, *Penerapan XML Web service Pada Sistem Distribusi Barang*, Universitas Sriwijaya, Palembang
- Istiyanto, E.J.; Edhy, S., 2012, *Model Interoperabilitas Antar Aplikasi E-Government*, Universitas Gadjah Mada Yogyakarta, Yogyakarta
- Michael, S.; Purba, J., 2007, *Membongkar Teknologi Pemrograman Web service*, Gava Media, Yogyakarta
- Short, S., 2003, *Building XML Web services For The Microsoft .Net Platform*, Elex Media Komputindo, Jakarta
- Sutanta, E.; Khabib, M., 2012, *Kebutuhan Web service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov Di Pemkab Bantul Yogyakarta*, STMIK BANDUNG, Bandung
- Wellem, T., 2009, *Perancangan Prototype Aplikasi Mobile Untuk Pengaksesan Web service*, UPN "Veteran" Yogyakarta, Yogyakarta
- Yasin, V., 2012, *Rekayasa Perangkat Lunak Berorientasi Objek Pemodelan, Arsitektu dan Perancangan (Modeling, Architecture and Design)*, Mitra Wacana Media, Jakarta