BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang membahas mengenai rekam medik pernah dilakukan oleh Widya Teknika (2014). Aplikasi yang dibangun yaitu aplikasi rekam medik pasien dengan menerapkan *handwriting* yang berfungsi melakukan input data hasil rekam medik dengan cara menulis langsung pada aplikasi berbasis android. Pada aplikasi tersebut menggunakan metode *mobile* komputer dan memanfaatkan media *canvas* yang pada umumnya digunakan untuk menggambar pada media *android*.

Penelitian yang memanfaatkan *web services* pernah juga dilakukan oleh Hidayat dan Ashari (2014). Aplikasi ini terbagi atas dua yaitu aplikasi untuk rumah sakit umum dan aplikasi untuk rumah sakit Asri *Medical Center*. Tujuan pemberdayagunaan *web service* dalam sistem ini agar rumah sakit rujukan, baik itu rumah sakit umum maupun lainnya saling terintegrasi informasi dengan rumah sakit pemberi rujukan (puskesmas) sehingga dapat melakukan distribusi informasi baik informasi dokter, fasilitas, dan lain-lain dalam rangka memenuhi pelayanan terhadap masyarakat / pasien.

Pemanfaatan *web service* juga dilakukan oleh Adi dan Riyanto (2013) Hasil dari penelitian oleh Adi dan Riyanto (2013) adalah terbangunnya *web service* menggunakan *metode simple obyek acces protocol* sehingga memberikan solusi sistem informasi farmasi tanpa harus membangun aplikasi baru dengan fasilitas *web services* sebagai integrasi data.

Penelitian yang memanfaatkan *web service* pernah juga dilakukan oleh Widanarto (2013). Aplikasi yang dibangun berupa suatu aplikasi *web service* yang berbeda *platform* (*desktop* dan *web base*). *Web service* yang dibangun oleh Widanarto (2013) ini telah berhasil diimplementasikan dengan baik untuk menghubungkan aplikasi berbasis *desktop* dan *web*, sehingga dapat menjadi salah satu pilihan solusi untuk komunikasi data antar aplikasi yang berbeda *platform*.

Penelitian yang memanfaatkan *web service* pernah dilakukan oleh Saptono (2012). Aplikasi yang dibangun berupa suatu aplikasi *web service* yang berfungsi untuk verifikasi klaim data pasien. Selain itu, sistem dibangun untuk menyediakan layanan bagi *client*. Hasil penelitian yang dilakukan oleh Saptono (2012) ini menghasilkan sebuah *prototipe* integrasi data dari pihak Bapel Jamkesos dengan instansi rumah sakit umum daerah di wilayah DIY. Data yang diintegrasikan adalah data klaim pasien rumah sakit.

Pemanfaatan *web services* juga dilakukan oleh Sarwosri dan Naja (2011). Aplikasi yang dibangun berupa suatu aplikasi *web service* yang berfungsi untuk komunikasi pasien ke semua rumah sakit agar memberikan kemudahan kepada pasien dan meningkatkan kualitas pelayanan kesehatan. Dan membantu pasien untuk menelusuri rekam mediknya dari semua rumah sakit yang pernah dia kunjungi tanpa perlu mengunjungi satu per satu rumah sakit-rumah sakit tersebut. Pasien juga akan lebih mudah mendapatkan info dan fasilitas kesehatan yang dibutuhkan tanpa perlu mengunjungi satu per satu *web* rumah sakit yang ada.

Adapun perbandingan beberapa pustaka yang digunakan sebagai referensi penelitian ini adalah :

Tabel 2.1 Perbedaan Penelitian Sebelumnya

Penulis	Domain	Relevansi	Teknologi
Sarwosri dan Naja (2011)	Kesehatan	*Teknologi Web Service*	- Berbasis *Service- Oriented Architecture* - Menggunukan ASP.Net C# dan Oracle SE
Saptono (2012)	Data Klaim	*Teknologi Web Service*	- Menggunakan PHP - Menggunakan XML (*eXtensible Markup Language*) - Menggunakan SOAP *dan* WSDL - Menggunakan UDDI *(Universal Description Discovery, and Integration)*
Widanarto (2013)	Rumah Sakit dan Asuransi	*Teknologi Web Service*	- Menggunakan PHP - Menggunakan MySQL - Menggunakan *Visual Basic* - Mengguanakan *SQL Server* - Menggunakan bahasa *ASP.Net* dengan *.NETFramework* 3.5
Adi dan Riyanto (2013)	Farmasi	*Teknologi Web Service*	- Menggunakan XML (*eXtensible Markup Language*) - Menggunakan SOAP *dan* WSDL - Menggunakan UDDI *(Universal Description Discovery, and Integration)*
Hidayat dan Ashari (2014)	Puskesmas dan Rumah Sakit	*Teknologi Web Service*	- Menggunakan ASP.NET - Menggunakan PHP
Widya Teknika (2014)	Rekam medik	*Android Rekam medik*	- Menggunakan Android - Menggunakan Mobile Computing - Menggunakan web service
Usulan Penelitian (2016)	Rekam medik klinik panasea	*Web Service dan Android*	- Menggunakan XML (*eXtensible Markup Language*) - Menggunakan SOAP *(Simple Object Application Protocol)* - Menggunakan WSDL *dan* UDDI - Menggunakan web (PHP) - Menggunakan *Mobile Application* Android (HTML 5 dan Javascript)

2.2 Dasar Teori

2.2.1 Web service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung *interoperabilitas* (menjembatani komunikasi antar program berbeda) dan interaksi antar sistem pada suatu jaringan. Sehingga aplikasi yang satu dengan aplikasi yang lain pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standar protokol yang ditetapkan oleh *web service*. Umumnya w*eb service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*services*) yang disediakan oleh suatu sistem yang menyediakan *web service*.

Protokol tersebut dibangun oleh *Extensible Markup Language* (XML) yang didukung oleh banyak *platform*, bahasa pemrograman, dan oleh *developer*. *Web service* disusun oleh komponen standar yaitu :

1. *Simple Object Access Protocol* (*SOAP*), yaitu protokol yang berfungsi pada bagian pertukaran informasi pada lingkungan jaringan terdistribusi.
2. *Web Service Definition Language* (*WSDL*), dokumen standar yang ditulis dalam format XML, dan mendefinisikan kehadiran *web service* dalam suatu jaringan.
3. *Universal Description, Discovery, and Integration* (*UDDI*), yaitu suatu lokasi direktori yang berisikan *service* (layanan) dan bersifat bebas *platform* (*platform*

independent), dituliskan berbasis XML dan dapat diakses oleh entitas yang berbeda di dalam dan luar jaringan. (Ivan Michael Siregar dan Johanes Purba,)

Blok bangunan *Web Service*

Blok bangunan *web service* menyediakan fasilitas komunikasi jarak jauh antara 2 aplikasi yang merupakan layer arsitektur *web service* (Stephen, 2003). Dengan bagian layer yang terdiri dari :

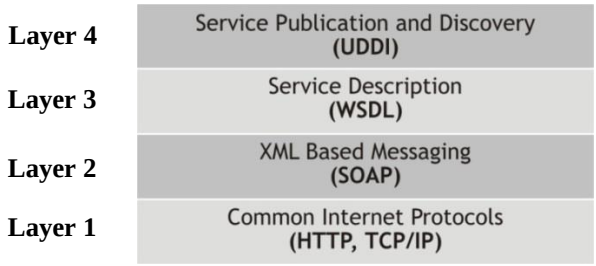
Layer 1 : Protokol internet standar yang digunakan sebagai saranan transportasi adalah HTTP dan TCP/IP.

Layer 2 : *Simple Object Access Protocol* (SOAP) berbasiskan XML dan digunakan untuk pertukaran informasi antar kelompok layanan.

Layer 3 : *Web Service Description Language* (WSDL) digunakan untuk mendeskripsikan atribut layanan.

Layer 4 : *Universal Description Discovery and Integration* (UDDI) yang mana merupakan direktori pusat untuk deskripsi layanan.

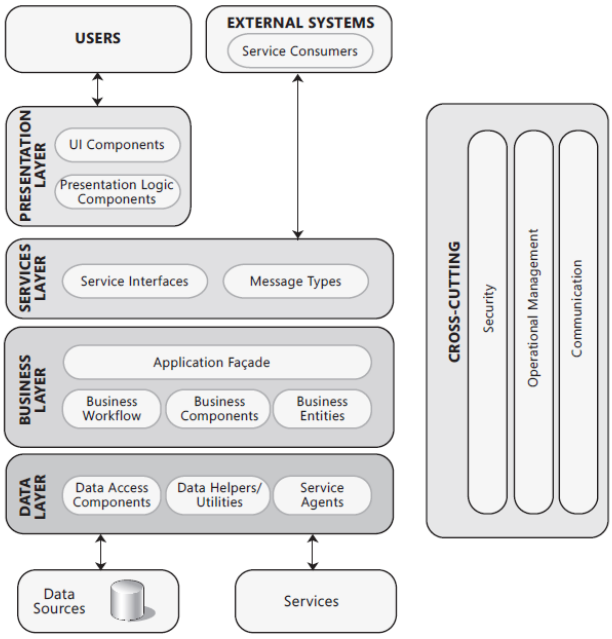
Arsitektur dari komponen *web service* tersebut disajikan pada Gambar 2.1



Gambar 2.1 Arsitektur komponen *web service*
(Stephens, 2003)

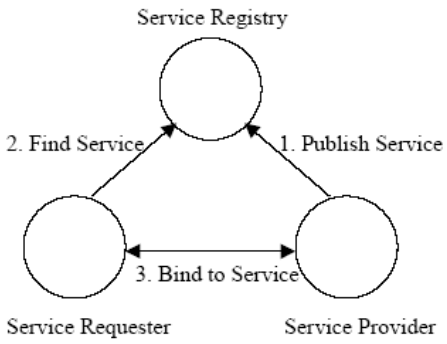
Arsitektur Aplikasi Web Service

Menurut S.Somasegar dan Scott Guthrie (2002) dalam buku Microsoft Application Architecture Guide, Arsitektur aplikasi *web service* terdiri dari layer penyusun aplikasi seperti pada Gambar 2.2 berikut ini :



Gambar 2.2 Arsitektur aplikasi *web service*
(S.Somasegar dan Scott Guthrie, 2002)

Sedangkan menurut Gottschalk *et al* (2002) layanan *web service* memiliki 3 *services* dan 3 operasi yang diinginkan pada Gambar 2.2.



Gambar 2.3 Operasi aplikasi *web service*
(Gottschalk *et al*, 2002)

Dari gambaran diatas terdapat tiga services *web service* sebagaimana yang diperlihatkan pada Gambar 2.3. yaitu :

1. ***Service Provider*** : berfungsi untuk menyediakan layanan dan mengelola register agar layanan tersebut dapat tersedia.
2. ***Service Registry*** : berfungsi sebagai lokasi sentral yang mendeskripsikan semua layanan yang telah diregister.
3. ***Service Requester*** : peminta layanan yang mencari dan menentukan layanan yang dibutuhkan serta menggunakan layanan web.

Operasi *web service* yang dikemukakan oleh Cerami pada tahun 2002 antara lain :

1. ***Publish/unpublish*** : menerbitkan/menghapus layanan di dalam atau dari *registry*. *Service provider* menghubungi *service broker* untuk di*publish* atau *unpublish* suatu layanan.
2. ***Finder*** *: requester* dan *provider* melaksanakan operasi *find* secara bersama. *Request* mencari dan menemukan *service broker* mencari layanan.
3. ***Bind*** *:* terjadi antara *service requester* dan *provider.* Setelah *service requester* menemukan layanan yang dicari, kemudian melakukan *binding* ke *service provider* untuk melakukan *binding* ke *service provider* untuk melakukan interaksi dan mengakses layanan yang disediakan oleh *service provider*.

2.2.2 Klinik

Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan/atau spesialistik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan

(perawat dan atau bidan) dan dipimpin oleh seorang tenaga medis (dokter, dokter spesialis, dokter gigi atau dokter gigi spesialis). Berdasarkan jenis pelayanannya, klinik dibagi menjadi Klinik Pratama dan Klinik Utama. Kedua macam klinik ini dapat diselenggarakan oleh pemerintah, pemerintah daerah atau masyarakat. Berikut adalah penjabaran dari jenis klinik tersebut :

1. Klinik Pratama adalah klinik yang menyelenggarakan pelayanan medik dasar.
2. Klinik Utama adalah klinik yang menyelenggarakan pelayanan medik spesialistik atau pelayanan medik dasar dan spesialistik. Sifat pelayanan kesehatan yang diselenggarakan bisa berupa rawat jalan, one day care, rawat inap dan/atau home care.

2.2.3 Codeigniter

Codeigniter adalah sebuah framework php yang bersifat open source dan menggunakan metode MVC(Model, View, Controller). Codeigniter bersifat free alias tidak berbayarjika menggunakannya. Framework codeigniter dibuat dengan tujuan sama seperti framework lainnya yaitu memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuat dari awal.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

1. Model

   Merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database seperti misalnya mengambil data dari database, menginput

dan mengolah database lainnya. Semua intruksi yang berhubungan dengan pengolahan database diletakkan di dalam model.

2. View

   Merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user. Tampilan dari user interface dikumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

3. Controller

   Merupakan kumpulan intruksi aksi yang menghubungkan model view, jadi user tidak akan berhubungan dengan model secara langsung, intinya dari view kemudian controller yang mengolah intruksi.

Kelebihan Codeigniter Yaitu :

a) Syntax yang terstruktur
b) Kemudahan dalam menggunakannya
c) Codeigniter menyediakan fasilitas helper dan library yang dapat membantu developer dalam membuat Pagination, Sesion, Manipulasi url dan lainnya.
d) Keamanan yang sudah lumayan karena user atau pengakses aplikasi tidak berhubungalangsung dengan database

   (dikutip dari Rick Ellis http://www.ellislab.com)

2.2.4 Library Volley

***Volley* adalah** *library* HTTP yang membuat permintaan jaringan untuk *Android apps* mudah dan cepat. *Volley* tersedia melalui repositori AOSP terbuka. Pada networking *library volley* tujuan utama dalam networking *calls* adalah agar *call* yang dijalankan menjadi mudah, singkat dan tidak perlu lagi menulis banyak code berulang. Defaultnya, *volley network calls* bekerja secara *asynchronously*, oleh karena itu tidak perlu repot lagi tentang *asynctask* selama pengembangan. Berikut fitur kelebihan *Android library volley* :

a. Automatic scheduling of network requests and Multiple concurrent network connections.
b. Transparent disk and memory response caching with standard HTTP cache coherence.
c. Support for request prioritization, debugging and tracing tools.
d. Cancellation request API. You can cancel a single request, or you can set blocks or scopes of requests to cancel.
e. Ease of customization, for example, for retry and backoff.
f. Strong ordering that makes it easy to correctly populate your UI with data fetched asynchronously from the network.
g. Penjadwalan permintaan jaringan otomatis dengan Multi koneksi jaringan bersamaan.
h. Transparan disk dan respon memori *cache* dengan standar koherensi *cache* HTTP.
i. Dukungan untuk prioritas permintaan dan Pembatalan permintaan API.