

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Teknologi *web service* sudah diterapkan di banyak bidang seperti pendidikan, pemerintahan, kesehatan dan lain-lain. Sehingga objek penelitian dan sistem yang dihasilkan berbeda - beda. Adapun penggunaan teknologi *web service* yang terkait sebagai berikut:

Fiftin Noviyanto dan Joko Purwanto (2012), melakukan penelitian yang berjudul Rancang Bangun *Digital Library Multiserver* dengan Teknologi *Web Service*. Dalam penelitian tersebut menggunakan teknologi *Rest Web Service* untuk menghasilkan Terintegrasinya informasi pada *digital library* dengan menggunakan teknologi *web service* dan memanfaatkan pertukaran data antar *server digital library* setiap universitas.

Didiek S. Wiyono dan Ardhi Wijayanto (2012), melakukan penelitian yang berjudul Implementasi *Rest Web Service* Dengan Menggunakan *Json* Pada Aplikasi *Mobile Enterprise Resource Planning*. Dalam penelitian tersebut menggunakan teknologi *Rest Web Service* untuk menghasilkan Aplikasi *Mobile Enterprise Resource Planning* untuk *platform android*.

Ni Kadek Sukerti dan Ni Wayan Cahya Ayu Pratami (2014), melakukan penelitian yang berjudul Implementasi Sistem Informasi Reservasi *Speedboat* Berbasis *Web Service* Dan *Sms Reply*. Dalam penelitian tersebut menggunakan teknologi *Rest Web Service* untuk menghasilkan Sistem Informasi Reservasi *Speedboat* yang diakses melalui perangkat android.

Penidas Fiodinggo Tanaem, dkk (2016), melakukan penelitian yang sejenis berjudul *RESTFul Web Service* Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ. Dalam penelitian tersebut menggunakan teknologi *Rest Web Service* untuk menghasilkan sebuah arsitektur *RESTFul Web Service* yang aman bagi PT. XYZ. Adapun *RESTFul WS* yang dibangun, menggunakan *JSON Web Token (JWT)* dalam mengamankan komunikasi yang terjadi.

Moh. Erwin Indrawan dan Ahmat Adil (2016), melakukan penelitian yang berjudul Implementasi *Restful Web Service One Chip Multi-Client* untuk Mengoptimalkan Penjualan Pulsa All Operator. Dalam penelitian tersebut menggunakan teknologi *Rest Web Service* untuk menghasilkan Aplikasi *Restful Web Service* yang dapat mengoptimalkan penjualan pulsa yaitu mampu memperbanyak area penjualan dan mempercepat penjualan pulsa.

Penelitian di atas digunakan sebagai rujukan dalam melakukan penelitian tentang pembuatan sistem pencarian buku berbasis web dari beberapa

perpustakaan. Adapun perbedaan penelitian – penelitian sebelumnya dengan penelitian yang dilakukan sekarang bisa dilihat pada table 2.1.

Tabel 2.1 Tabel Perbandingan Penelitian

Penulis	Judul Penelitian	Objek	Teknologi	Hasil Penelitian
Fiftin Noviyanto dan Joko Purwanto, 2012	Rancang Bangun <i>Digital Library Multiserver</i> dengan Teknologi <i>Web Service</i>	<i>Digital Library</i>	<i>Rest Web Service</i>	Terintegrasinya informasi pada <i>digital library</i> dengan menggunakan teknologi <i>web service</i> dan memanfaatkan pertukaran data antar <i>server digital library</i> setiap universitas.
Didiek S. Wiyono dan Ardhi Wijayanto, 2012	Implementasi <i>Rest Web Service</i> Dengan Menggunakan <i>Json</i> Pada Aplikasi <i>Mobile Enterprise Resource Planning</i>	Aplikasi <i>Mobile Enterprise Resource Planning</i>	<i>Rest Web Service</i>	Aplikasi mobile <i>ERP</i> dengan teknologi <i>REST web service</i> yang menggunakan format <i>JSON</i> dalam pertukaran data.
Ni Kadek Sukerti dan Ni Wayan Cahya Ayu Pratami, 2014	Implementasi Sistem Informasi Reservasi <i>Speedboat</i> Berbasis <i>Web Service</i> Dan <i>Sms Reply</i>	Sistem Informasi Reservasi <i>Speedboat</i>	<i>Rest Web Service</i>	Sistem Informasi Reservasi <i>Speedboat</i> Berbasis <i>Web Service</i> dan <i>Sms Reply</i> yang dibangun mampu menampilkan informasi terkait jadwal <i>Speedboat</i> serta informasi lainnya yang terkait.
Penidas Fiodinggo Tanaem, dkk. 2016	<i>RESTful Web Service</i> Untuk Sistem Pencatatan Transaksi Studi Kasus PT. XYZ	Sistem Pencatatan Transaksi	<i>Rest Web Service</i>	Arsitektur <i>RESTful WS</i> yang aman bagi PT. XYZ, Menggunakan <i>JSON Web Token (JWT)</i> dalam mengamankan komunikasi yang terjadi.
Moh.Erwin Indrawan dan Ahmat Adil, 2016	Implementasi <i>Restful Web Service One Chip Multi-Client</i> untuk Mengoptimalkan Penjualan Pulsa <i>All Operator</i>	Penjualan Pulsa <i>All Operator</i>	<i>Rest Web Service</i>	Aplikasi <i>Restful Web Service</i> yang dapat mengoptimalkan penjualan pulsa yaitu mampu memperbanyak area penjualan dan mempercepat penjualan pulsa.

Tabel 2.1 Tabel Lanjutan

Penulis	Judul Penelitian	Objek	Teknologi	Hasil Penelitian
Dedi Irawan, 2017	<i>Penerapan Rest Web Service untuk Sistem Pencarian Buku dari Beberapa Perpustakaan</i>	Perpustakaan	<i>Rest Web Service</i>	Sistem Pencarian buku berbasis web yang dapat mengambil data buku dari beberapa <i>server</i> perpustakaan yang memiliki <i>database</i> dan struktur tabel berbeda.

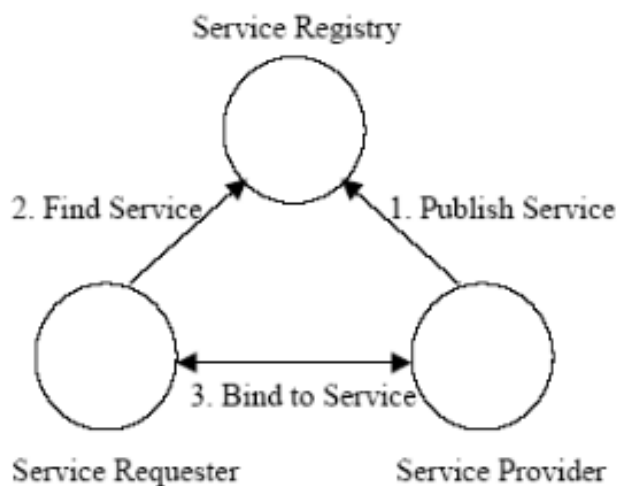
2.2 Dasar Teori

2.2.1 *Web Service*

Menurut Ivan Michael Siregar (2012). *Web service* adalah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam *platform* dan antarmuka bahasa pemrograman yang independen. Tujuan pengembangannya adalah untuk menjembatani komunikasi antar program, sehingga aplikasi yang satu dan aplikasi yang lain yang terdapat pada suatu jaringan yang sama atau pada jaringan berbeda dapat saling berkomunikasi asalkan menggunakan standar protokol yang ditetapkan oleh *web service*. Hal ini bisa terjadi, karena standar protokol itu tidaklah terikat pada suatu platform atau bahasa pemrograman.

2.2.2 *Arsitektur Web Service*

Menurut Gottschalk et al (2002) layanan *web service* memiliki tiga *service* dan tiga operasi yang diinginkan pada gambar 2.1.



Gambar 2.1 Arsitektur dan Operasi *Web Service*

Ada tiga *services* sebagaimana yang diperlihatkan pada gambar 2.1 yaitu :

1. *Service Provider* : Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
2. *Service Registry* : Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-register.
3. *Service Requestor* : Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Tiga operasi *web service* tersebut antara lain :

1. *Publish/Unpublish* : Menerbitkan/menghapus layanan ke dalam atau dari *registry*.

2. *Find* : *Service requestor* mencari dan menemukan layanan yang dibutuhkan.

3. *Bind* : *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *service provider* untuk melakukan interaksi dan mengakses layanan/*service* yang disediakan oleh *service provider*. (Cerami, 2002) .

2.2.3 *Rest Web Service*

REST merupakan singkatan dari *REpresentative State Transfer*. *REST* adalah sebuah metode dalam menyampaikan *resource* melalui media *web*. Sedangkan *resource* sendiri didefinisikan sebagai segala sesuatu yang dapat disimpan didalam sebuah komputer dan ditampilkan sebagai urutan bit, misalnya sebuah dokumen, tabel dalam sistem basis data, atau hasil dari sebuah perhitungan (Sandoval, 2008). *REST* (*REpresentational State Transfer*) adalah model arsitektur yang pada dasarnya memanfaatkan teknologi dan protokol yang sudah ada seperti *HTTP* (*Hypertext Transfer Protocol*) dan *XML*. Berikut merupakan penggunaan *methods*(metode) *HTTP* dalam *REST Web services* (Riyadi, 2013).

Tabel 2.2 Tabel Metode *HTTP* dan Penggunaannya dalam *REST*

Metode	Deskripsi
<i>GET</i>	Mendapatkan (<i>read</i>) sebuah sumber daya (<i>resource</i>) yang diidentifikasi dengan <i>URI (Uniform Resource Identifier)</i> .
<i>POST</i>	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk membuat (<i>create</i>) sumber daya baru.
<i>PUT</i>	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk memasukkan (<i>insert</i>) atau memperbarui (<i>update</i>) sumber daya yang tersimpan.
<i>DELETE</i>	Menghapus (<i>delete</i>) sumber daya (<i>resource</i>) yang diidentifikasi dengan <i>URI</i> .
<i>HEAD</i>	Mendapatkan <i>metadata (response header)</i> dari sumber daya (<i>resource</i>) yang diidentifikasi dengan <i>URI</i> .

2.2.4 *JSON*

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*, Standar ECMA-262 Edisi ke-3 Desember 1999. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga *C* termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dll. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran-data. *JSON* terbuat dari dua struktur:

- a) Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- b) Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. (JSON Team, www.json.org).

2.2.5 Google Maps API

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. Google Maps adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, Google Maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu browser. Kita dapat menambahkan fitur Google Maps dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun

dengan Google Maps API. Google Maps API adalah suatu *library* yang berbentuk *JavaScript*. (Kindarto, 2008).