

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Berdasarkan dengan judul penelitian oleh penulis mengenai “*Pengembangan Web api Pada Sistem Assesmen Dan Berbasis Tag Sebagai Pembantu Penyusunan Strategi Pembelajaran*” maka perlu penjelasan sedikit tentang mengenai *Web api* atau sering disebut dengan *Web Service*.

API merupakan singkatan dari *Application Programming Interface* yang artinya adalah kumpulan fungsi perintah, atau protokol untuk menggantikan bahasa yang digunakan dalam system call dengan bahasa yang lebih mudah dimengerti. Programmer sering menggunakan *API* untuk menggunakan fungsi standar demi berinteraksi dengan sistem lain. *Web api* lebih populer dengan sebutan *web service*, karena *API* tersebut memberikan layanan *service* yang diminta oleh klien. Pada umumnya, *web api* dengan klien dan server tidak terjadi secara langsung, karena klien dan server dijembatani oleh file *web api* dalam format tertentu. *Web api* terdiri atas kumpulan fungsi-fungsi *Javascript*, *JSON*, *REST*, *SOAP* dan *XML*.

Pada penjelasan penulis kali ini ialah dimana penulis dapat mengembangkan sebuah aplikasi *website* yang akan memberikan layanan *service* file *web api* dalam bentuk format *JSON* serta dapat memberikan pemahaman dan tujuan cara penggunaan *web api* pada aplikasi tersebut.

Asep Herman Suyanto (2007) dalam penelitiannya “*Web Service*” adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* di atas jaringan. *Web service* mempunyai alat penghubung yang diuraikan di dalam format *machine-processable* (secara spesifik WSDL). Sistem lain saling berhubungan dengan *Web service* di dalam cara yang ditentukan oleh deskripsinya yang secara khas disampaikan menggunakan *HTTP* dengan *XML serialization*, bersama dengan standar lain yang terkait dengan *web* (Booth et al., 2004). Untuk menjalankan fungsinya, *web service* memerlukan agen. Agen adalah bagian perangkat lunak atau perangkat keras yang mengirimkan dan menerima pesan. Agen dapat ditulis dengan berbagai bahasa pemrograman. Dan dapat berganti-ganti bahasa pemrograman dengan fungsi yang sama. Tujuan *web service* adalah untuk menyediakan beberapa fungsi atas nama pemiliknya seseorang atau organisasi seperti bisnis atau perorangan.

Nurasanah (2014) dalam Penelitiannya “Perkembangan Antara *Web Service* Yang Menggunakan Protocol *Soap* Dan *Rest*” Sistem *Web services* ini tidak memerlukan browser atau HTML. Sehingga berikutnya *web service* dapat ditambahkan ke sebuah GUI (seperti laman *web* atau aplikasi desktop). *Web service* adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, ia akan menyediakan method-method yang dapat diakses oleh network. Ia juga akan menggunakan *XML* untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda.

Iqbal Firdaus, Dhanang Sukmana Adi, Fredi Aji Noorhadi (2014) dalam penelitiannya “Rancang Bangun E- *Resto* Menggunakan *Web Service* Untuk *Restoran Family* Di Kotamadya Surakarta” Sistem ini bertujuan untuk

mengembangkan aplikasi *E-resto* dengan aplikasi *web-service* dimulai dengan kasus yang hilang catatan, data yang tidak sama masuk dan keluar dan sering salah arah. Merancang dan membangun sebuah sistem untuk Menangani hidangan pesanan *e-restaurant* dari para pelayan ke dapur dan sebaliknya dan caranya menerapkan layanan *web* sebagai penanganan layanan realtime untuk menggantikan komunikasi antara dapur dan para pelayan menggunakan metode pengamatan dan pendekatan untuk desain dalam bentuk UML. Pengembangan pesanan pengiriman *e-restaurant* diharapkan bisa membangun sebuah aplikasi yang membantu default dalam proses bisnis *restoran* sebagai jalur input sudah terkena standar penggunaan pesan secara efektif antara proses bisnis dalam pesan layanan di *restoran* bisa menangani pesanan atau salah terselip di antara.

2.2 Landasan Teori

Untuk mendukung hasil penelitian diperlukan suatu konsep dalam merumuskan definisi-definisi yang menunjang kegiatan penelitian baik teori dasar maupun teori umum.

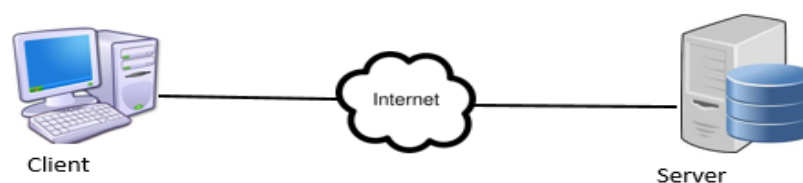
2.2.1 Sistem Web Service

Web Service adalah layanan yang tersedia di Internet. *Web Service* menggunakan format standar *JSON* untuk pengiriman pesannya. *Web Service* juga tidak terikat kepada bahasa pemrograman atau sistem operasi tertentu. *Web Service* adalah antar muka yang mendeskripsikan koleksi yang dapat diakses dalam jaringan menggunakan format standar *JSON* untuk pertukaran pesan. *Web Service* mengerjakan tugas yang spesifik. *Web Service* dideskripsikan menggunakan format standar notasi *JSON* yang disebut *services description*.

Tujuan *web service* adalah untuk menyediakan beberapa fungsi atas nama pemilik nya seseorang atau organisasi seperti bisnis atau perorangan. *Provider entity* adalah organisasi atau orang yang menyediakan agen yang sesuai untuk menerapkan *service* tertentu.

2.2.2 Arsitektur Aplikasi

Aplikasi ini akan digunakan pada setiap sekolah dengan *web server* dan basis data yang berbeda untuk setiap sekolah, sehingga peneliti mendapatkan sebuah analisis rancangan arsitektur *web* secara umum seperti pada gambar berikut :



Gambar 2. 1 Arsitektur Aplikasi

Database yang digunakan pada database server adalah Microsoft SQL Server 2014, kemudian akan terhubung melalui *web server*. User mengakses *web* dengan menggunakan komputer client, saat user mengakses *web*, sistem akan memuat antarmuka *web* dan melakukan pengambilan data dari database server. Melalui antarmuka *web*, user dapat menambahkan data dan disimpan kedalam database.

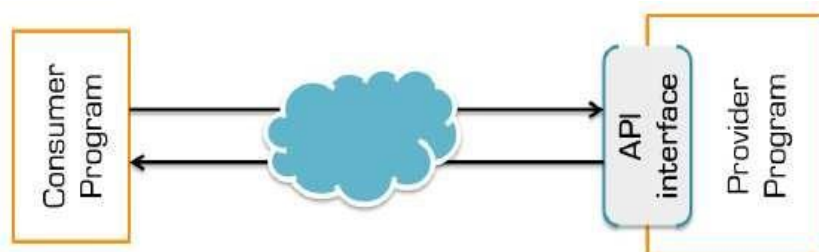
Dalam pembuatan aplikasi ini, pola yang akan digunakan adalah *MVC*. Pola *MVC* memisahkan sebuah aplikasi, khususnya aplikasi *website* menjadi tiga bagian, yaitu:

- *Model* : *Model* berperan untuk mengelola data, menyimpan dan mengambil *entity* yang digunakan aplikasi dari sebuah database server.
- *View* : *View* memiliki fungsi untuk menampilkan data yang sebelumnya diolah oleh *model*. Berhubungan kepada interface akhir aplikasi yang akan dilihat oleh user.
- *Controller* : *Controller* berguna sebagai penghubung antara *model* dan *view*. *Controller* menerima permintaan dari *client*, lalu memanggil *model* untuk melakukan operasi, kemudian mengirimkan data pada *view*. Akhirnya *view* akan menampilkan data tersebut dalam bentuk sebuah halaman cshtml.

2.2.3 *Web api*

Web api adalah antar muka program dari sistem yang dapat diakses lewat method dan header pada protokol *HTTP* yang standar. *Web api* dapat diakses dari berbagai macam *HTTP* client seperti browser dan perangkat mobile. *Web api* juga memiliki keuntungan karena menggunakan infrastruktur yang juga digunakan oleh *web* terutama untuk penggunaan caching dan concurrency (Block, 2014). *Web api* juga sebagai antarmuka pemrograman aplikasi (Application Programming Interface/*API*) adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk system operasi tertentu. *API* memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan system operasi. *Web api* juga menyediakan fungsi dan perintah dengan bahasa yang lebih terstruktur dan lebih mudah untuk dipahami oleh programer bila dibandingkan dengan System Calls, hal ini penting untuk aspek

editing dan pengembangan, sehingga programmer dapat mengembangkan sistem dengan mudah. *API* juga dapat digunakan pada Sistem Operasi mana saja asalkan sudah ada paket-paket *API* nya.



Gambar 2. 2 Skema Konektivitas *API*

Secara struktural, *API* merupakan spesifikasi dari suatu *data structure, objects, functions*, beserta parameter-parameter yang diperlukan untuk mengakses *resource* dari aplikasi tersebut. Seluruh spesifikasi tersebut membentuk suatu *interface* yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain, dan *API* dapat digunakan dengan berbagai bahasa *programming*, ataupun hanya dengan menggunakan URL (*Uniform Resource Locator*) yang telah disediakan oleh suatu *website*.

API dapat diklasifikasikan menjadi beberapa kategori, hal ini dilihat dari abstraksi apa yang dideskripsikan di dalam sistem. Kategori-kategori ini diantaranya :

Tabel 2 1 Kategori *API*

Kategori <i>API</i>	Deskripsi	Contoh
<i>Operating System</i>	<i>API</i> yang digunakan untuk fungsi dasar yang dapat dilakukan oleh komputer. Seperti proses I/O, eksekusi program.	<i>API</i> for MS Windows

<i>Programming Languages</i>	<i>API</i> yang digunakan untuk memperluas kapabilitas dalam melakukan eksekusi terhadap suatu bahasa pemrograman.	Java <i>API</i>
<i>Application Services</i>	<i>API</i> yang digunakan untuk memperluas kapabilitas dalam melakukan eksekusi terhadap suatu bahasa pemrograman.	<i>API</i> for mySAP
<i>Infrastructure Services</i>	Digunakan untuk mengakses infrastruktur dari suatu komputer. Infrastruktur disini adalah komputer beserta <i>peripheral</i> seperti <i>storage</i> , aplikasi, dan lain-lain.	Amazon EC2 (<i>Elastic Compute Cloud</i>) untuk akses untuk <i>virtual computing</i> dan Amazon S3 (<i>Simple Storage Service</i>) untuk menyimpan data dalam jumlah besar.
<i>Web Services</i>	<i>API</i> yang digunakan untuk mengakses <i>content</i> dan layanan yang disediakan oleh suatu <i>web application</i> .	<i>Web api TagBasedExam API</i> yang digunakan untuk mengakses informasi yang dapat dibagikan.

2.2.4 XML (Extensible Markup Language)

Extensible Markup Language (XML) adalah bagian dari SGML. Tujuannya untuk memungkinkan SGML umum untuk dilayani, diterima, dan diproses pada *Web* yang sekarang dengan HTML. *XML* telah dirancang untuk meringankan implementasi dan untuk interoperabilas dengan SGML dan HTML.

2.2.5 JSON (Javascript Object Notation)

Javascript Object Notation (JSON) 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.

2.2.6 WSDL (*Web Service Description Language*)

WSDL adalah sebagai protokol komunikasi dan format pesan distandardisasi di komunitas *web*, WSDL mungkin terus meningkat dan penting untuk bisa menguraikan komunikasi dalam beberapa cara terstruktur. WSDL menunjukkan kebutuhan ini dengan penjelasan *XML grammar* untuk menguraikan *service* jaringan seperti koleksi komunikasi *endpoints* untuk mampu bertukar pesan. Definisi *service* WSDL menyediakan dokumentasi untuk sistem terdistribusi dan bertindak sebagai petunjuk untuk mengotomatiskan detil yang dilibatkan di dalam komunikasi aplikasi (Christensen et.al., 2001).

WSDL adalah tatabahasa *XML* untuk penetapan *public interface* untuk *web service* (Cerami, 2002). *public interface* ini dapat meliputi:

1. Informasi pada semua fungsi yang tersedia secara publik.
2. Tipe Data informasi untuk semua pesan *JSON*.
3. Membungkus informasi tentang *specific transport protocol* untuk digunakan.
4. Menunjuk informasi untuk menempatkan *service* yang ditetapkan.

2.2.7 Visual Studio

Menurut situs resminya, Visual Studio adalah satu set lengkap alat pengembangan yang digunakan untuk membangun aplikasi *Web ASP.NET*, *XML Web Services*, aplikasi desktop, dan aplikasi mobile. Visual Basic, Visual C #, dan Visual C ++ menggunakan *Integrated Development Environment (IDE)* yang sama,

yang memungkinkan alat untuk berbagi dan memudahkan terciptanya solusi dalam menggunakan bahasa campuran. Selain itu, bahasa-bahasa ini menggunakan fungsionalitas *.NET Framework*, yang menyediakan akses ke teknologi kunci yang mempermudah pengembangan aplikasi *Web ASP* dan *XML Web Services*.

2.2.8 Metode MVC dan ASP.NET

Aplikasi *web* yang dijadikan penelitian adalah aplikasi *web* yang dibangun menggunakan metode *MVC (Model-View-Controller)*. *Model-View-Controller* atau *MVC* adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan bagaimana cara memprosesnya (*Controller*). Dalam implementasinya, kebanyakan *framework* dalam aplikasi *web* adalah berbasis arsitektur *MVC (Model-View-Controller)* (Rosmala, Ichwan, & Muhammad, 2011).

Aplikasi *web* akan dibangun menggunakan *ASP.NET (Active Server Pages.NET)* yang mendukung metode *MVC*. *ASP.NET* merupakan platform pembuatan aplikasi *web* yang menyatu dengan *.NET Framework* serta menyediakan fasilitas-fasilitas bagi developer untuk membangun aplikasi *web* untuk level perusahaan.

2.2.9 MS SQL Server

Menurut (Agus, 2013:11), *MS SQL Server* adalah salah satu produk *Relational Database Management System (RDBMS)* populer yang berfungsi sebagai relasi database dalam sebuah program aplikasi. *Microsoft SQL Server* mendukung *SQL* sebagai bahasa pemrograman query. *SQL* yang sebagaimana sudah diketahui secara luas merupakan bahasa standar internasional untuk proses query database.