

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Implementasi model *web service* telah banyak dilakukan, berikut contoh-contoh model *web service* yang pernah dikembangkan dalam berbagai kasus termasuk dalam *e-commerce*.

Dalam artikel ilmiah yang ditulis oleh Hartono dkk pada tahun 2013 yang berjudul “Aplikasi Reservasi Tiket Bus pada Handphone Android menggunakan *Web service*” (Studi Kasus: PO. Rosalia Indah). Penyajian informasi ketersediaan tiket yang masih manual menjadi salah satu permasalahan yang dialami oleh jasa travel karena calon penumpang harus datang ke agen untuk mengetahui ketersediaan tiket. Berdasarkan permasalahan tersebut, maka dibutuhkan sebuah layanan yang mampu mendukung sebuah aplikasi *client*. *Web service* adalah layanan yang dirasa mampu mengatasi permasalahan tersebut. *Web service* yang berbasis XML sangat memungkinkan untuk diimplementasikan sehingga menjanjikan banyak kemudahan dan perbaikan dalam mendukung integrasi berbagai *platform system* dan aplikasi, baik melalui infrastruktur *Intranet* maupun *Internet/Exsternet*. Sistem reservasi tiket *online* yang dikembangkan dapat diakses melalui perangkat mobile memudahkan calon penumpang dalam mendapatkan informasi dan melakukan proses pemesanan tiket.

Dalam jurnal penelitian yang dilakukan oleh Marthasari dan kawan-kawan pada tahun 2010. Penelitian ini mengimplementasikan *web service* untuk mendukung interoperabilitas pada aplikasi *e-commerce*, dimana fleksibilitas sistem *e-commerce* yang masih dihalangi oleh persoalan interoperabilitas antar perangkat. Dengan memanfaatkan *Web service* memungkinkan perangkat yang berbeda *platform* dapat saling berkomunikasi yaitu *platform web, desktop*, dan HP. Dalam penelitian ini dihasilkan aplikasi *e-commerce* berbasis *web service* dengan studi kasus sebuah toko buku *online*. Aplikasi *e-commerce* ini dapat diakses melalui website, desktop, dan *mobile device*.

Jurnal penelitian lain tentang pembangunan aplikasi *e-commerce* yang mendistribusikan data inventori selain pada website dan juga melalui perangkat *mobile* sehingga user dapat mengakses data inventori melalui perangkat *mobile* (Suryadi dkk, 2011). Program ini bertujuan agar data pada basis data pada server dapat ditampilkan oleh perangkat *mobile* dan *website*. Pada aplikasi ini data yang ditampilkan terintegrasi dengan baik antara aplikasi dengan basis data. Serta data dapat terdistribusi dengan baik antara basis data dengan *website* dan perangkat *mobile*.

Penelitian yang dilakukan oleh Rozali pada tahun 2011 yang berjudul *Next Generation Mobile Application*. Paper ini membahas arsitektur-arsitektur untuk membangun aplikasi *Next Generation Mobile* menggunakan *Mobile Cloud Computing*, *Context Aware* dan *RESTful Web Service*. Teknologi *cloud computing* telah menyediakan layanan penyimpanan data, layanan komputasi, mekanisme antrian dan keamanan yang terjamin bagi lingkungan *mobile*. *Context awareness* dapat diterapkan pada *smart device*. Jika *context* diaktifkan, maka fitur ini memungkinkan kita untuk memastikan informasi tambahan dari perangkat komputasi itu sendiri tanpa membutuhkan input pengguna secara *eksplisit*. Teknologi *RESTful Web Service* digunakan sebagai layer konektivitas, REST membuat lapisan konektivitas yang bagus antara lingkungan *cloud* dan *platform mobile*, terutama bila digunakan bersama dengan HTTP. Dengan kombinasi beberapa teknologi yang disebutkan diatas, yaitu *smart mobile device*, *context enablement* menggunakan *sensor pada device*, dan *Cloud Computing* dengan *RESTful web-services*, kita memasuki era baru teknologi informasi dan komputasi. Smart Mobile Device dapat menggunakan sensors untuk membuat aplikasi *context aware*, yang akan mengurangi user input.

Penelitian lain, Mutakin (2011), berhasil membangun sebuah *web service* yang mampu mengirim data kendaraan dari *client (dealer)* ke *database server* di Samsat yang mempunyai aplikasi yang berbeda. Penyedia layanan administrasi utama diletakkan di Kantor Samsat dan klien para *dealer* cukup menyediakan sebuah *device* dan sebuah aplikasi klien yang dapat mengolah transaksi yang dikirimkan oleh *web*

*service* tersebut, sehingga pada akhirnya dihasilkan sebuah *Application Programming Interface (API)* dalam bentuk *web service* yang menyediakan layanan untuk mengelola data kendaraan dari *dealer*. *Web service* tersebut dibangun dengan menggunakan .NET.

Pada jurnal penelitian seminar nasional ilmu komputer Universitas Diponegoro Semarang pada tahun 2010. Saputra mengembangkan model *web service* untuk implementasi *e-commerce* pada katalog toko buku. Pengembangan aplikasi ini didasari pada masalah interoperabilitas atau pengintegrasian aplikasi yang berbeda *platform* aplikasi. Pada penelitian tersebut dibuat sebuah *prototype* pengintegrasian dua buah situs buku *online* dengan *database* yang berbeda. Pertukaran data dilakukan dengan format XML (*eXtensible Markup Language*) menggunakan teknologi SOAP (*Simple Object Access Protocol*).

Dalam kajiannya Muchallil dan Nazaruddin (2013), melakukan analisis waktu eksekusi verifikasi data wisuda baik yang menggunakan *web service* ataupun tidak dan kemudian membandingkannya. Permasalahan yang muncul karena adanya aplikasi *Online Exit-survey*, *Online Exit-survey* adalah proses yang wajib dilakukan oleh calon wisudawan sebelum proses wisuda. Penggunaan media online *Online Exit-survey* harus dapat menjamin bahwa responden adalah calon wisudawan dan responden yang mengisi kuesioner adalah responden itu sendiri tanpa mewakilkannya kepada orang lain. Hal ini untuk mencegah bias data yang didapatkan. Untuk melakukan proses verifikasi data ini maka perlu dibangun sebuah *gateway* untuk menjadi jembatan antara keduanya. Penelitian ini bertujuan untuk melihat secara jelas waktu eksekusi yang dihasilkan oleh proses verifikasi dengan menggunakan format data *JSON* pada *RESTful Web service*. Dari penelitian tersebut didapatkan hasil bahwa proses yang menggunakan *web service* akan memakan waktu yang lebih lama untuk dieksekusi. Untuk yang tidak menggunakan *web service* waktu eksekusi tergantung dari jumlah langkah yang dilaksanakan.

Hasil kajian dan riset di atas menunjukkan bahwa *web service* telah diimplementasikan secara luas, dalam beragam sistem, dan dapat menjadi solusi untuk integrasi dan *interoperabilitas* antar sistem informasi yang *heterogen*.



Tabel 2.1 Perbandingan penelitian yang dilakukan dengan penelitian-penelitian yang dibahas.

| <b>Fitur</b>                    | Saputra,<br>2010 | Marthasari<br>dkk, 2010 | Rozali,<br>2011 | Mutakin,20<br>11 | Muchallil<br>dan<br>Nazaruddin,<br>2013 | Hartono dkk,<br>2013 | Penulis<br>(2014)* |
|---------------------------------|------------------|-------------------------|-----------------|------------------|---|----------------------|--------------------|
| <i>Web-services</i>             | √                | √                       | √               | √                | √                                       | √                    | √                  |
| Interoperabilitas<br>data       | √                | √                       | √               | √                |   | √                    | √                  |
| <i>Implementasi<br/>RESTful</i> |                  |                         |                 |                  | √                                       |                      | √                  |
| Implementasi JSON               |                  |                         |                 |                  | √                                       |                      | √                  |

\*Dalam pengembangan.

## 2.2. Landasan Teori

### 2.2.1. Pengertian Reservasi atau Reservation

Berbicara mengenai reservasi (pemesanan kamar), terlebih dahulu kita perlu mengetahui defenisi dari reservasi itu sendiri. Pemesanan dalam bahasa Inggris adalah *Reservation* yang berasal dari kata “to reserve” yaitu menyediakan atau mempersiapkan tempat sebelumnya, sedangkan reservation yaitu pemesanan suatu tempat fasilitas (Christanto dkk, 2012).

Dalam perhotelan reservasi adalah suatu permintaan untuk memperoleh sejumlah kamar yang dilakukan beberapa waktu sebelumnya melalui berbagai sumber dengan menggunakan berbagai cara pemesanan untuk memastikan bahwa tamu akan memperoleh kamar tersebut pada waktu kedatangannya atau check-in.

*Booking* berasal dari kata *book* yang artinya adalah pemesanan atau pembukuan (Panuttary, 2013). *Booking* dilakukan dengan alasan agar kamar yang dikehendaki tidak diambil oleh orang lain. Tujuan dari reservasi atau pemesanan *online* adalah (Sulistiyanto, 2012):

1. Orang yang ingin memesan kamar lewat *internet* hanya membutuhkan akses *internet* dan interface-nya menggunakan *web browser*.
2. Menjadikan website reservasi tidak hanya untuk wadah pemesanan semata tetapi menjadi tempat berkumpulnya komunitas dengan membangun basis komunitas, membangun konsep reservasi kamar dalam berwisata sebagai pusat informasi (*release, product review, konsultasi, dll*).
3. Pengelolaan yang berorientasi pada pelayanan, kombinasi konsepsi pelayanan konvensional dan virtual : *Responsif* (respon yang cepat dan ramah), Dinamis, Informatif dan komunikatif.
4. Informasi yang *up to date*, komunikasi multi-arah yang dinamis.
5. Model pembayaran : kartu kredit atau transfer.

## 2.2.2. *Web service*

### 2.2.2.1. **Definisi**

*Web service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan (Perdede,dkk, 2013). *Web service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya *internet* dalam bentuk pesan *XML (eXtensible Markup Language)*(Kreger,2001).

*Web Service* adalah aplikasi perangkat lunak yang tersedia pada web yang melaksanakan fungsi yang spesifik (wulandari dan Wicaksana, 2006). Sedangkan menurut Michael C. Daconta (2005), *Web Service* adalah aplikasi perangkat lunak yang dapat ditemukan, diuraikan, dan diakses berdasarkan pada *XML* dan protokol *standard Web* pada *intranet, extranet, dan Internet*.

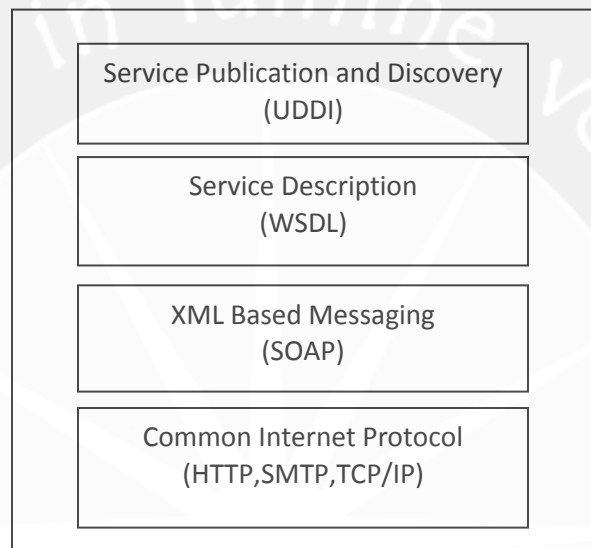
*Web service* menyediakan standar komunikasi di antara berbagai aplikasi *software* yang berbeda-beda, dan dapat berjalan di berbagai *platform* maupun *framework* (Hartono,dkk, 2012). *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service* (Predede,dkk, 2013).

Teknologi pada *web service* dapat mengubah kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi dengan pola *program-to-program (P2P)*. Fokus web selama ini didominasi oleh komunikasi program-to-user dengan interaksi business-to-consumer (B2C), sedangkan transactional web akan didominasi oleh program-to-program dengan interaksi business-to-business (Ghifari dan karya, 2011).

*Web service* sebenarnya adalah kumpulan dari fungsi dan method yang terdapat pada sebuah server yang dapat dipanggil oleh klien dari jarak jauh, kemudian untuk memanggil method-method tersebut kita bebas menggunakan aplikasi yang akan dibuat dengan bahasa pemrograman apa saja yang dijalankan pada *platform* apa saja (Marthasari, 2010).

Adanya teknologi *web service* dapat menjembatani perbedaan-perbedaan teknologi dari masing-masing sumber. Dapat ditarik kesimpulan bahwa *web service* merupakan kumpulan layanan yang disediakan melalui jaringan berbasis web dengan standar yang telah ditetapkan mampu menunjang interoperabilitas, dan dapat berjalan diberbagai *platform* dan *framework*.

#### 2.2.2.2. Arsitektur *Web service*



Gambar 2.1 Lapisan dasar *Web service*  
(Deviana, 2011;Ghifary dan Karya, 2011)

Gambar 2.1 merupakan blok bangunan *web service* yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur *web service*.

- a. Layer 1 : protokol *internet* standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.
- b. Layer 2 : *Simple Object Access Protocol* (SOAP) berbasiskan *XML* dan digunakan untuk pertukaran informasi antar sekelompok layanan.
- c. Layer 3 : *Web service* Definition Language (WSDL) digunakan untuk mendiskripsikan attribute layanan.
- d. Layer 4 : *Universal Description Discovery and Integration*, yang mana merupakan direktori pusat untuk deskripsi layanan.



Marthasari dkk (2010) dalam penelitiannya menjelaskan komponen *web service* yaitu:

a. *Extensible Markup Language (XML)*

*XML* merupakan dasar yang penting atas terbentuknya *Web services*. *Web services* dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilnya dengan menggunakan *XML*, karena *XML* berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protokol HTTP. Selain itu, *XML* juga bersifat *platform* independen sehingga informasi di dalamnya bisa dibaca oleh aplikasi apapun pada *platform* apapun selama aplikasi tersebut menerjemahkan tag-tag *XML*.

b. *Simple Object Access Protocol (SOAP)*

*XML* saja tidak cukup agar *Web services* dapat berkomunikasi dengan aplikasi yang lainya. *XML* yang digunakan untuk saling bertukar informasi antara *web services* dengan aplikasi yang lainya harus menggunakan sebuah format standard yang dapat dimengerti oleh keduanya. Format itulah yang dikenal dengan nama SOAP. SOAP (*Simple Object Access Protocol*) merupakan suatu format standard dokumen berbentuk *XML* yang digunakan untuk melakukan proses request dan responses antara *web services* dengan aplikasi yang memanggilnya. Dokumen SOAP digunakan untuk melakukan request disebut dengan SOAP *request* sedangkan dokumen SOAP yang diperoleh dari *Web services* disebut dengan SOAP *responses*.

c. *Web service Definition Language (WSDL)*

Sebelum mengakses sebuah *Web services* pastinya perlu mengetahui *method-method* apa saja yang disediakan oleh *Web services* tersebut, untuk mengetahuinya memerlukan sebuah dokumen yang bernama WSDL. WSDL (*Web services Description Language*) adalah sebuah dokumen dalam format *XML* yang isinya menjelaskan informasi detail sebuah *Web services*. Di dalam WSDL dijelaskan *method-method* apa saja yang tersedia dalam *Web services*, parameter apa saja yang diperlukan untuk memanggil sebuah

method, dan apa hasil atau tipe data yang dikembalikan oleh *method* yang dipanggil tersebut.

### 2.2.2.3. Teknologi REST

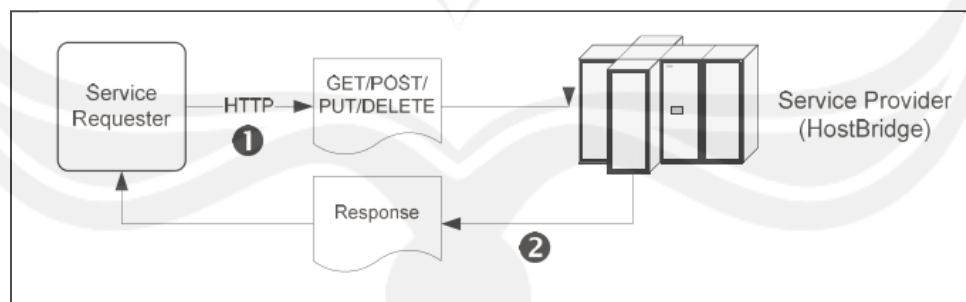
Ada beberapa pendekatan teknologi *web services* yang dapat digunakan, yaitu metode SOAP dan REST. Dalam penelitian ini akan digunakan pendekatan REST.

REST merupakan singkatan dari *REpresentative State Transfer*. Pertama kali dikemukakan dalam disertasi seorang program doktor bernama Roy Thomas Fielding pada tahun 2000 (Rozali, 2011). 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 (Riyadi,2013).

### 2.2.2.4. Arsitektur REST

Dibawah ini merupakan model dasar dari RESTful *Web services*.



Gambar 2.2 Model dasar RESTful *Web services* (HostBridge,2009).

Berikut merupakan langkah-langkah yang dilakukan dalam model dasar RESTful *Web services* (HostBridge, 2009):

1. Query Request Provider melalui HTTP dengan menggunakan URI (*Uniform Resource Identifier*). *Request* menggunakan *methods* (metode) HTTP untuk menentukan apakah request tersebut dimaksudkan untuk *Create*

(menciptakan), *Read* (membaca), *Update* (memperbarui), atau *Delete* (menghapus) data.

2. *HostBridge* mengembalikan sebuah dokumen dalam bentuk *XML* untuk *Requester* (pemohon) dengan CICS data *enclosed*.

Diagram di atas menunjukkan dua *web services* : sebuah Provider (*HostBridge*) yang menyediakan layanan *web services* dan sebuah Request yang digunakan pada *web services*. Dalam model RESTful layanan dideskripsikan secara sendiri (*self-describing*), sehingga tidak diperlukan WSDL dan Server UDDI yang bertindak sebagai Broker (*HostBridge*, 2009).

Sehingga pada development web modern aksi CRUD tersebut dapat dimappingkan dengan HTTP method sebagai berikut : *CREATE* sebagai *POST*, *RETRIVE* sebagai *GET*, *UPDATE* sebagai *PUT*, dan *DELETE* sebagai *DELETE*. Berikut merupakan penggunaan methods(metode) HTTP dalam REST *Web services* (Riyadi, 2013):

Tabel 2.2 Metode HTTP dan Penggunaannya dalam REST (Riyadi, 2013).

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

Arsitektur REST dibangun dengan sifat sebagai berikut (Gröhbiel, 2011):

1. *Addressability*

Dalam prinsip ini seluruh sumberdaya atau resource harus tersedia melalui sebuah alamat unik, pengalamatan ini dilakukan dengan menggunakan URI (*Unique Resource Identifiers*)

2. *Uniform Interface*

Semua interaksi sebaiknya dibangun dengan interface yang seragam. RESTful service menampilkan semua *resource* dan interaksinya dengan *interface* yang seragam, dalam metode REST antarmuka yang digunakan adalah dengan menggunakan HTTP. HTTP menawarkan semua operasi yang diperlukan, dikenal, dan tersebar luas. Semua interaksi antara klien dan sumber daya (*resource*) didasarkan pada metode dasar HTTP. Metode-metode interaksi yang digunakan dengan dasar HTTP ditunjukkan pada tabel 2.2

3. *Representation-oriented*

Representasi menjelaskan dalam bentuk apa data sedang dipertukarkan antara *client* dan *server*. Pada umumnya data dipertukarkan dalam bentuk XML, JSON, dan HTML.

4. *Statelessness*

Setiap interaksi antara *client* dan *server* harus memiliki state sendiri (atau dengan kata lain tidak dipengaruhi *session client*). Jadi *server* hanya akan memantau *resource* state bukan *client session*.

5. *Hypermedia As The Engine Of Application State* (HATEOAS)

*Hypermedia* sebagai state dari sebuah aplikasi (HATEOAS), menyatakan REST dapat menggunakan *link* untuk menghubungkan sumber daya atau *resource* ke sumber daya lain yang berkaitan. Hal ini mirip dengan web, dimana kami menggunakan *hyperlink* untuk menghubungkan antara situs web.

### 2.2.3. JSON (JavaScript Object Notation)

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 k-e3 Desember 1999 (Zainuddin dkk, 2013). Pertukaran data dengan menggunakan format JSON sangat ideal karena Format JSON berbasis teks dan terbaca oleh manusia, serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek)( Norwandi dkk, 2012).

JSON adalah sebuah format data yang tidak bergantung pada suatu bahasa pemrograman. Kode pengolahan dan pembuatan JSON telah tersedia untuk banyak bahasa pemrograman. Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi (Kusumawaty, 2012). Aplikasi utamanya adalah pada pemrograman aplikasi web AJAX dengan berperan sebagai alternative terhadap penggunaan tradisional format *XML* (Norwandi dkk, 2012).

Menurut Noertjahyana (2014), JSON dibangun di atas dua struktur, yaitu :

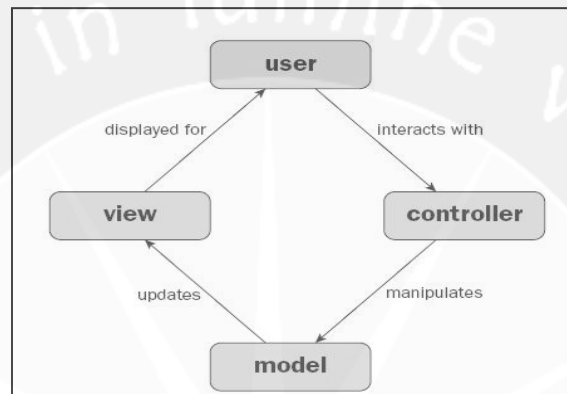
1. Sebuah koleksi name atau *value pairs*. Dalam berbagai bahasa, hal ini disebut sebagai record, *object*, struct, kamus, hash table, keyed list, atau associative array.
2. Sebuah ordered list dari nilai-nilai. Dalam kebanyakan bahasa, hal ini merupakan sebuah vector, array, daftar atau urutan.

Sebuah objek JSON merupakan *unordered set* dari *name* atau *value pairs*. Sebuah objek dimulai dengan tanda '{' (*left brace*) dan diakhiri dengan tanda '}' (*right brace*). Setiap name diikuti oleh tanda ':' (*colon*) dan *name* atau *value pairs* dipisahkan oleh tanda ',' (koma).

Kelebihan format data menggunakan JSON menurut nurseitov dkk (2009) adalah kecepatan proses jika dibandingkan dengan *XML* yang merupakan format data dalam *web service* yang telah ada selama ini. Kecepatan ini disebabkan karena untuk memarsing sebuah XML file maka dibutuhkan sebuah *library external* (Muchallil dan Nazaruddin, 2013). Sementara JSON sudah dapat diproses langsung oleh *javascript* tanpa memerlukan *library* lainnya. Masih meneurut Nurzeitov et al bahwa format data ini dapat terbaca dengan mudah oleh manusia dan dapat diparsing dengan cepat oleh computer.

#### 2.2.4. CodeIgniter

Menurut Myer (2008) CodeIgniter (CI) merupakan framework untuk aplikasi *web* dengan menggunakan bahasa pemrograman PHP. Dalam membuat suatu *web* aplikasi, CI menggunakan arsitektur MVC (*Model, View, Controller*), terlihat pada Gambar 2.3 Arsitektur ini memisahkan *business logic* dari *user interface* sehingga aplikasi menjadi lebih mudah untuk dimodifikasi.



Gambar 2.3 Arsitektur MVC pada CodeIgniter (Myer, 2008)

CI merupakan *PHP framework* yang awalnya ditulis oleh Rick Ellis, pendiri dan CEO EllisLab.com, perusahaan yang mengembangkan CodeIgniter. Tujuan dari pembuatan *framework* CI ini menurut *user manual*-nya adalah untuk menghasilkan *framework* yang akan dapat digunakan untuk pengembangan proyek pembuatan *website* secara lebih cepat dibandingkan dengan pembuatan *website* dengan cara *coding* secara manual, dengan menyediakan banyak sekali pustaka yang dibutuhkan dalam pembuatan *website*, dengan antarmuka yang sederhana dan struktur logika untuk mengakses pustaka yang dibutuhkan. Versi awal dari CI pertama kali di rilis pada bulan Maret 2006. Dan hingga sekarang CI masih terus dikembangkan oleh komunitas dan disebar ke seluruh dunia dengan lisensi bebas. Dimana CI pada penelitian ini digunakan untuk fungsionalitas dari sistem.

#### 2.2.5. Electronic Commerce (*E-commerce*)

*E-commerce* adalah suatu kegiatan bisnis yang dilakukan lewat dunia maya (*internet*), dengan memanfaatkan kemajuan teknologi para pengguna *e-commerce* dapat

melakukan penjualan, pembelian dan lain-lain. Menurut Turban (2010), *E-commerce* adalah suatu proses membeli, menjual, transfer atau pertukaran produk, pelayanan, dan informasi melalui jaringan computer termasuk *internet*.

*E-commerce* merupakan bagian dari sebuah e-business, dengan ruang lingkup yang lebih luas, tidak sebatas perniagaan tetapi mencakup juga pengkolaborasi mitra bisnis, pelayanan nasabah, lowongan pekerjaan dan lain-lain (Rosmala dkk,2012). *E-commerce* sangat menguntungkan untuk para penggunanya karena memudahkan untuk pembeli yang menginginkan suatu barang tanpa harus bepergian untuk membelinya.

Jadi dapat disimpulkan *E-commerce* adalah proses dari pengembangan, pemasaran, penjualan, pengiriman, pelayanan, dan pembayaran untuk berbagai produk dan jasa yang diperjualbelikan dalam pasar global berjejaring para pelanggan dengan dukungan dari jaringan para mitra bisnis di seluruh dunia (Denni,2011).

Menurut E.Turban dan Volonino (2010), E-Commerce memiliki beberapa jenis, yaitu:

1. *Business-to-business (B2B)*

Dalam transaksi B2B, Baik pembeli maupun penjual merupakan organisasi bisnis. Volume kegiatan EC terbanyak ada pada jenis ini.

*Business to Business eCommerce* memiliki karakteristik beberapa karakteristik (Irmawati, 2011):

1. *Trading partners* yang sudah diketahui dan umumnya memiliki hubungan (*relationship*) yang cukup lama. Informasi hanya dipertukarkan dengan partner tersebut. Dikarenakan sudah mengenal lawan komunikasi, maka jenis informasi yang dikirimkan dapat disusun sesuai dengan kebutuhan dan kepercayaan (*trust*).
2. Pertukaran data (*data exchange*) berlangsung berulang-ulang dan secara berkala, misalnya setiap hari, dengan format data yang sudah disepakati bersama. Dengan kata lain, layanan yang digunakan sudah tertentu. Hal ini memudahkan pertukaran data untuk dua entiti yang menggunakan standar yang sama.

3. Salah satu pelaku dapat melakukan inisiatif untuk mengirimkan data, tidak harus menunggu parternya.
  4. Model yang umum digunakan adalah peer-to-peer, dimana processing intelligence dapat didistribusikan di kedua pelaku bisnis.
2. *Collaborative commerce (c-commerce)*

Dalam *c-commerce*, sesama rekan bisnis berkolaborasi secara elektronik sepanjang mata rantai supply-nya. Atau dengan kata lain memungkinkan *partner* dagang mengakses data internal yang dapat menyatukan *buyer*, *seller* dan *supplier* dalam satu *website*.

3. *Business-to-consumers (B2C)*

Penjual sebagai organisasi dan pembelinya adalah para individu. B2C ini juga dikenal dengan nama *e-tailing*. *E-tailing (electronic retailing)* adalah menjual kembali barang dan jasa secara *online*.

4. *Consumer-to-consumer (C2C)*

Di sini seseorang menjual barang atau jasa kepada orang lain sama halnya dengan *customer to customer*.

5. *Business-to-business-to-consumers (B2B2C)*

Suatu bisnis menjual produk ke bisnis lain yang kemudian digunakan oleh konsumen individu.

6. *Consumers-to-businesses (C2B)*

Konsumen bertindak sebagai perantara untuk menyediakan atau mencarikan produk yang diperlukan oleh pengguna jasa atau barang.

Dalam C2B konsumen memeritahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen (Irmawati, 2011). Contohnya di *priceline.com*, dimana pelanggan menyebutkan produk dan harga yang diinginkan, dan *priceline* mencoba menemukan pemasok yang memenuhi kebutuhan tersebut.



7. *Intrabusiness (intraorganizational) commerce*

Di sini organisasi menggunakan *E-Commerce* secara internal dalam suatu perusahaan untuk kegiatan operasionalnya. Pada bagian ini juga terdapat aplikasi yang dikenal dengan nama *business-to-its-employees* (B2E) dimana perusahaan menyediakan barang dan jasa kepada karyawannya sendiri.

8. *Government to citizens (G2C) and to others*

Pemerintah menyediakan pelayanan teknologi EC kepada masyarakat. Pemerintah juga dapat menjalankan bisnisnya dengan unit pemerintahan lainnya (G2G) atau dengan pelaku bisnis (G2B).

Keuntungan yang didapat dengan menggunakan *E-Commerce* adalah (Turban dan volonino, 2010):

1. Bagi Organisasi

- a. Dapat memperluas penjualan dan pembelian perusahaan ke pasar nasional dan internasional. Dengan pengeluaran modal minimal, sebuah perusahaan dengan cepat dapat menemukan lebih banyak pelanggan, pemasok terbaik, dan mitra bisnis yang paling cocok di seluruh dunia.
- b. Memungkinkan perusahaan untuk melakukan pengadaan material dan jasa dari perusahaan lain dengan cepat dan biaya yang lebih hemat.
- c. Memotong atau bahkan menghilangkan saluran distribusi pemasaran, membuat harga produk yang lebih murah dan keuntungan bagi vendor yang lebih tinggi.
- d. Membantu usaha kecil bersaing dengan perusahaan besar.

2. Bagi Pelanggan

- a. Menyediakan produk dan jasa yang lebih murah dengan memungkinkan konsumen untuk melakukan perbandingan online dengan cepat.
- b. Memungkinkan pelanggan untuk berbelanja atau melakukan transaksi lainnya selama 24 jam sehari, di hampir setiap lokasi manapun.

- c. Memberikan informasi yang relevan dan rinci dalam hitungan detik.
  - d. Memungkinkan bagi orang untuk bekerja dan belajar di rumah.
  - e. Mungkin terjadinya lelang elektronik.
  - f. Memungkinkan konsumen untuk berinteraksi dalam komunitas elektronik, bertukar pikiran, dan membandingkan pengalaman.
3. Bagi Masyarakat
- a. Memungkinkan individu untuk bekerja di rumah dan melakukan sedikit perjalanan, sehingga lalu lintas jalan yang jarang dan polusi udara yang lebih rendah.
  - b. Memungkinkan beberapa barang dagangan yang akan dijual dengan harga yang lebih rendah, sehingga meningkatkan standar hidup rakyat.
  - c. Memungkinkan orang di negara berkembang dan daerah pedesaan untuk menikmati produk dan jasa yang mungkin tidak tersedia dinegara tersebut.
  - d. Memfasilitasi pelayanan publik, seperti hak pemerintah, mengurangi biaya distribusi dan kesempatan penipuan, dan meningkatkan kualitas pelayanan sosial.