BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Menurut **Martinus Raditia Sigit Surendra** yang berjudul "**Implementasi PHP Web Service Sebagai Penyedia Data Aplikasi Mobile**". Aplikasi *smart phone* dengan fitur informasi promosi dan informasi tempat membutuhkan data yang dinamis seiring dengan berubahnya data terkait tempat-tempat bisnis dan promosi yang ada di suatu wilayah. Solusi yang diajukan adalah menyimpan data di satu *database server*. Selanjutnya setiap aplikasi melakukan *request* untuk mendapatkan data dari *server*. Begitu juga untuk pengiriman data, setiap aplikasi akan mengirimkan data ke *server*. Untuk mendukung *scalability* maka digunakan teknologi *web service* sebagai solusi pertukaran data. Dengan memanfaatkan *web service* maka setiap *platform* baik aplikasi *smart phone*, web, maupun aplikasi *desktop* bisa memanggil *method*/fungsi yang diinginkan.*Web service* adalah layanan yang diidentifikasi dengan URI (*Uniform Resource Identifier*) yang mengekspos fiturnya melalui internet menggunakan protokol dan bahasa standar internet serta dapat diimplementasikan menggunakan standar internet seperti XML (*Extensible Markup Language*). Sampai dengan saat ini teknologi *web service* terus berkembang. Salah satu teknologi yang populer saat ini adalah REST (*Representational State Transfer*) atau terkadang disebut dengan RESTful. Beberapa contoh *RESTful web service* adalah Amazon's Simple Storage Service (S3), Atom Publishing Protocol, dan Google Maps. Pada prinsipnya *request* ke suatu RESTful *web service* sebenarnya adalah suatu *HTTP Request*. Berdasarkan dari proses yang sudah dilakukan maka dapat disimpulkan bahwa *PHP Web*

Service bisa diimplementasikan dalam aplikasi mobile yang membutuhkan data dinamis. Pengujian atas *web service* bisa dilakukan dengan membuat file PHP secara manual ataupun menggunakan SOAP *web service*. Untuk memudahkan pemanggilan data bisa dilakukan modifikasi dengan memberikan *layer* tambahan berupa PHP File yang memanggil pada SOAP *web service*.

Menurut **R. Hidayat dan A. Ashari** dalam jurnal yang berjudul " **Penerapan Teknologi Web Service Untuk Integrasi Layanan Puskesmas dan Rumah Sakit**". Aplikasi *web service* layanan informasi kesehatan merupakan suatu aplikasi *web service* yang berfungsi melakukan pengambilan dan pengiriman data yang diperoleh bisa diinput oleh pengunjung yang merupakan operator atau praktisi kesehatan yang berguna untuk memecahkan masalah atau layanan. Dimana rumah sakit rujukan memberikan layanan-layanan bagi pasien secara online menggunakan aplikasi berbasis web untuk beberapa rumah sakit. Ketika seorang pasien akan melakukan pendaftaran menjadi pasien rumah sakit rujukan tersebut sebelumnya melakukan pengecekan layanan terhadap suatu penyakit , baik dokter yang menangani, ruang rawat inap, harga dan lain-lainnya, maka pasien akan melakukan pencarian di beberapa rumah sakit rujukan dengan menggunakan aplikasi berbasis web yang ada. Prototipe sistem berhasil dibangun berupa integrasi system layanan informasi kesehatan diabetes melitus berbasis *web service* dapat menyatukan beberapa system yang berbeda dari segi bahasa pemograman (ASP.NET, PHP) dan databasenya (SQL Server, MYSQL). Hal ini menunjukkan bahwa telah terjadi komunikasi antara *service provider* dan *service requester* dapat berjalan dengan baik melalui sebuah jaringan.Dengan fitur-fitur dokter, fasilitas, pasien dan registrasi. Prototipe sistem yang dikembangkan dapat

memberikan informasi yang terintegrasi melalui sebuah aplikasi berbasis web yang dikembangkan dengan menggunakan PHP sebagai *client* dari dua *Web Service* yang ada. Implementasi teknologi web service yang berfungsi sebagai middleware mampu melakukan pertukaran pesan (*message*) dengan memanfaatkan protocol HTTP melalui sebuah jaringan antara aplikasi dan *database* antara rumah skit pemberi rujukan (puskesmas) ke rumah sakit rujukan (AMC/RSU).

2.2 Konsep Dasar Sistem Informasi

2.2.1 Sistem

Sistem dapat didefinisikan dengan dua pendekatan yaitu :

1) Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.

2) Pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Suatu sistem sebenarnya terdiri atas dua bagian, yaitu struktur dan proses. Struktur adalah komponen dari sistem tersebut dan proses adalah prosedurnya (Jogiyanto, 2005).

2.2.2 Informasi

Menurut Jogiyanto, HM. (2005) kualitas dari suatu informasi (*quality of information*) dikatakan berkualitas apabila mempunyai karakteristik sebagai berikut :

1. Akurat (*accurate*)

Suatu informasi mempunyai titik ketelitian tinggi, harus bebas dari kesalahan dan dapat dipertanggung jawabkan.

2. Relevan (*relevance*)

Informasi tersebut mempunyai nilai dan manfaat untuk pemakainya, yaitu benar-benar relevan dengan masalah yang dihadapi.

3. Tepat Waktu (*timeliness*)

Informasi harus tersedia tepat pada waktu yang dibutuhkan. Informasi yang datang pada penerima tidak boleh terlambat karena informasi yang kurang tidak akan mempunyai nilai lagi.

2.2.3 Sistem Informasi

Sistem informasi adalah kumpulan antara sub-sub sistem yang saling berhubungan yang membentuk suatu komponen yang didalamnya mencakup input-proses-output yang berhubungan dengan pengolahan data menjadi informasi sehingga lebih berguna bagi pengguna (Kadir, 2003). Sistem informasi mencakup sejumlah komponen (manusia, komputer, dan teknologi informasi), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

2.2.4 *Internet*

Internet adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia. Setiap komputer dan jaringan terhubung secara langsung maupun tidak langsung ke beberapa jalur utama yang disebut *internet backbone*. Masing-masing *backbone* dibedakan atau dengan yang lainnya menggunakan *unique name* yang biasa disebut IP *address* (contoh: 192.168.11.26). Untuk dapat menghubungkan beberapa platform yang berbeda, maka dibutuhkan sebuh *protocol* yang standar, yaitu : TCP/IP (*Transmission Control Protocol/Internet Protocol)* (Tutang dan Ismulyana Djan, Kitas Sukses Bisnis di Internet, 2010)

2.2.5 *World Wide Web* (WWW)

Awalnya informasi dapat dicari pada internet dengan menggunakan fasilitas *information service* berbasis *archive, gopher* dan WAIS (*Wide Area Information System).* Pencarian informasi berdasarkan menu-menu pada sistem-sistem tersebut dan output yang duhasilkan berbasis teks. Saat ini dengan teknologi *World Wide Web,* dimungkinkan untuk mengakses informasi secara interaktif, dan bentuk informasinya berupa tampilan grafis maupun teks.Hal ini dimungkinkan dengan adanya *Hypertext Transfer Protocol* (HTTP) yang digunakan untuk mengakses suatu informasi yang disimpan pada suatu situs web (*website*). Untuk dapat menggunakan sarana ini, dibutuhkan aplikasi *Web Browser.*

2.2.6 *Hypertext Markup Language* (HTML)

HTML dikenal sebagai bahasa standar untuk membuat dokumen web. Sesungguhnya Hypertext Markup Language (HTML) justru tidak dibuat untuk mempublikasikan informasi di web, namun oleh karena kesederhanaan serta kemudahan penggunaannya, HTML kemudian dipilh orang untuk mendistribusikan informasi di web.

Perintah-perintah HTML diletakkkan dalam file berekstensi *.html dan ditandai dengan mempergunakan tag (tanda) berupa karakter “<” dan “>”. Tidak seperti bahasa pemrograman berstruktur seperti pascal atau C, HTML tidak mengenal jumping atau looping. Kode-kode HTML dibaca oleh web browser dari atas kebawah tanpa adanya lompatan-lompatan.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu header dan body.Masing-masing ditandai oleh pasangan container tag<head> dan <body>. Bagian head berisikan judul dokumen dan informasi-informasi dasar lainnya sedangkan bagian body adalah data dokumennya. Pengaturan format teks dan pembentukn link dilakukan terhadap objeknya langsung dengan ditandai oleh tag-tag HTML.

2.3 Point Of Sale

Pengertian *Point Of Sale* atau yang biasa yang disingkat POS yaitu, merupakan kegiatan yang berorientasi pada penjualan serta sistem yang membantu proses transaksi. Setiap POS terdiri dari *hardware* dan *software* dimana kedua komponen tersebut digunakan untuk setiap proses transaksi.

POS *software* merupakan komponen utama dari sistem pos yang pada akhirnya menentukan jalannya proses, seperti apa yang harus dilakukan dan bagaimana harus melakukan. Sedangkan *hardware* POS dibutuhkan untuk menjalankan fungsinya, membantu proses pembayaran dan membuat tanda terima untuk pelanggan. Dalam hal pemilihan *hardware* ini, sebaiknya mencocokan dengan lingkungan kerja, seperti yang akan digunakan oleh penulis pada laporan skripsi ini adalah *barcode scanner*, yang merupakan bagian terpenting untuk mempercepat proses pemasukkan barang dan proses pelayanan penjualan.

2.4 *Web Services*

2.4.1 Definisi *Web Services*

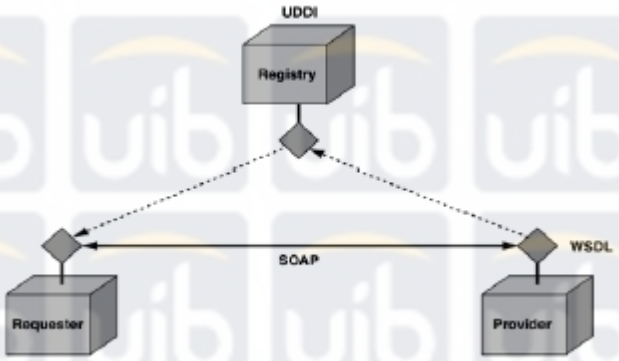
Web services adalah suatu *system* perangkat lunak yang didisain untuk mendukung interaksi mesin ke mesin pada suatu jaringan. Ia mempunyai suatu interface yang diuraikan dalam suatu format *machine-processible* seperti *WSDL* (*Web Service Description Language*). Sistem lain yang berinteraksi dengan *Web service* dilakukan melalui *interface*/antar muka menggunakan pesan seperti pada *SOAP*. Pada umumnya pesan ini melalui *HTTP* dan *XML* yang merupakan salah satu *standard web*. (W3C, 2004).

Konsep arsitektur yang mendasari teknologi *Web service* adalah *Service Oriented Architecure* (*SOA*), *SOA* mendefinisikan 3 peran berbeda yang menunjukkan peran dari masing-masing komponen dalam system, yaitu (W3C, 2004) :

a. *Service provider*, yaitu suatu entitas yang menyediakan *interface* terhadap sistem yang menjalankan suatu sekumpulan tugas tertentu.

b. Service *requestor* , yaitu suatu entitas yang meminta/memperoleh (dan menemukan) *software service* dalam rangka meyelesai kan suatu tugas tertentu atau menyediakan solusi bisnis tertentu.

c. Service *registry* , yaitu entitas yang bertindak sebagai penyimpan (repository) suatu software service yang dipublikasikan oleh Service provider .



Gambar 2.1 Implementasi *Web Service*

Menurut Kreger (2001) *web-service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya *internet*, dalam bentuk pesan *XML*. Sedangkan menurut Manes (2001), *web-service* diartikan sebagai sepotong atau sebagian informasi atau proses yang dapat diakses oleh siapa saja, kapan saja dengan menggunakan piranti apa saja, tidak terikat dengan sistem operasi atau bahasa pemrograman yang digunakan.

Web service dapat dirancang untuk mendukung interopabilitas mesin-ke-mesin yang dapat berinteraksi melalui jaringan. *Web service* memiliki antarmuka

yang dijelaskan dalam format mesin-*processable* (khusus *WSDL*). Sistem lain berinteraksi dengan *Web service* dalam cara ditentukan oleh deskripsi dengan menggunakan pesan *SOAP*, biasanya disampaikan menggunakan *HTTP* dengan serialisasi *XML* dalam hubungannya dengan *Web* lainnya yang terkait standar. *Web service* juga memungkinkan untuk dipanggil dengan menggunakan *protocol* lain seperti *SMTP* (*Simple Mail Transfer Protocol*), namun yang paling umum digunakan *HTTP*. *Web Services* dapat di definisikan sebagai aplikasi yang diakses oleh aplikasi yang lain. (Wijaya, 2012)

Gambar 2.2 Implementasi *Web Service*

Menurut Meiyanto (2001) pada sistem *multi-tier*, aplikasi maupun dokumen *XML* dapat dilewatkan ke pihak lain dan diolah oleh pihak tersebut. Dalam sistem ini dimungkinkan suatu aplikasi dapat mengambil data dari satu sumber tanpa harus tahu bahwa sebenarnya data tersebut dihasilkan melalui proses pengolahan oleh sistem lain sehingga dapat terjadi integrasi data maupun aplikasi yang sering disebut dengan *A2A* (*application to application*).

Dalam Kreger (2001) dikatakan bahwa model dari sebuah *web-service* didasarkan pada interaksi antara 3 komponen yang berperan dalam *web-service*, yaitu: service *provider*, service *registry* dan service *requestor*/*consumer*. Interaksi yang terjadi antara ketiga komponen tersebut juga melibatkan operasi publish, *find* dan *bind*. *Service provider* menyediakan *service* yang dapat diakses melalui jaringan komputer, misalnya *internet*. Kemudian, *service provider* mendeskripsikan *service* yang dibangun dan mem-publish-kan *service description* tersebut ke *service registry* atau secara langsung ke *service consumer*. Service *requestor*/*consumer* menggunakan operasi *find* untuk mendapatkan *service description* secara local maupun melalui *service registry*. *Service description* yang diperoleh itu kemudian digunakan untuk men-*bind service provider* dan berinteraksi dengan implementasi *web-service* yang akan digunakan tersebut.

2.4.2 Konsep *Web Service*

Web service adalah sebuah *software* yang dirancang untuk mendukung interoperabilitas interaksi mesin-ke-mesin melalui sebuah jaringan. *Web service* secara teknis memiliki mekanisme interaksi antar sistem sebagai penunjang interoperabilitas, baik berupa agregasi (pengumpulan) maupun sindikasi (penyatuan). *Web service* memiliki layanan terbuka untuk kepentingan integrasi data dan kolaborasi informasi yang bisa diakses melalui internet oleh berbagai pihak menggunakan teknologi yang dimiliki oleh masingmasing pengguna. Sekalipun mirip dengan *Application Programming Interface* (API) berbasis *web*, *web service* lebih unggul karena dapat dipanggil dari jarak jauh melalui internet.

Pemanggilan *web service* bisa menggunakan bahasa pemrograman apa saja dan dalam *platform* apa saja, sementara API hanya bisa digunakan dalam *platform* tertentu. *Web service* dapat dipahami sebagai *Remote Procedure Call* (RPC) yang mampu memproses fungsi-fungsi yang didefinisikan pada sebuah aplikasi *web* dan mengekspos sebuah API atau *User Interface* (UI) melalui *web*.

Kelebihan *web service,* yaitu :

1) lintas *platform*

2) *language independent*

3) jembatan penghubung dengan *database* tanpa perlu *driver database* dan tidak harus mengetahui jenis DBMS

4) mempermudah proses pertukaran data

5) penggunaan kembali komponen aplikasi.

Berdasarkan konsep hubungan dan penyampaian informasi, *web service* dikembangkan melalui empat model arsitektur, masing-masing berorientasi pada *message*, *action*, *resource*, dan *policy*. Pengembangan model yang diturunkan berdasarkan orientasi pada *action* (*Service Oriented Model*/SOM)) menghasilkan *Services Oriented Architecture* (SOA), yaitu model arsitektur berbasis layanan. Sementara pengembangan model yang diturunkan berdasarkan orientasi pada *resource* (*Resource Oriented Model*/ROM) menghasilkan *Resource Oriented Architecture* (ROA), yaitu model arsitektur berbasis sumberdaya informasi.

Dalam perkembangannya, model *web service* memiliki dua metode yang berorientasi pada layanan dan sumberdaya informasi, yaitu: SOAP (*Simple Object Access Protocol*) dan REST (*REpresentational State Transfer*). Impementasi

model SOA telah banyak dilakukan dan dikembangkan oleh banyak *vendor* (misal: Microsoft, Sun dan IBM, melalui dukungan *platform* infrastruktur .*Net* dan Java). Proses layanan dengan arsitektur SOAP memiliki tiga komponen utama, yaitu *service provider*, *service requester*, dan *service broker*, serta komponen pendukung yaitu: XML, SOAP-XML (terdiri atas *header* dan *body*), WSDL, dan UDDI. Metode REST telah dikembangkankan oleh Fielding yang didasari oleh empat prinsip utama teknologi, yaitu *Resource identifier through Uniform Resource Identifier* (URI), *uniform interface* (sumberdaya CRUD menggunakan operasi PUT, GET, POST, dan DELETE), *self-descriptive messages* (sumberdaya tidak terikat sehingga dapat mengakses konten HTML, XML, PDF, JPEG, plain text, meta data, dll), dan *stateful interactions through hyperlinks* (bersifat *stateless*). Metode REST lebih sederhana karena menggunakan format standar (HTTP, HTML, XML, URI, MIME), namun jika diperlukan proses pertukaran data, maka konten berupa teks dari hasil eksekusi *web service* dapat diolah dalam format teks (seperti XML atau HTML) dengan menggunakan utilitas komunikasi data berupa koneksi *socket* protokol HTTP. Utilitas ini umumnya tersedia dalam pustaka komunikasi pada bahasa pemrograman (seperti Java, Visual Basic, Delphi, PHP, ASP, dan JSP).

Model *web service* memberikan layanan untuk proses pertukaran data antar sistem informasi yang merupakan bentuk implementasi konsep interoperabilitas. Model layanan *web service* setidaknya melibatkan dua hal penting, yakni:

1. Problem utama pada format data. Selama ini problem ini diselesaikan menggunakan format netral yaitu XML, yaitu sebuah format dokumen yang mampu menjelaskan struktur dan semantic (makna) dari data yang dikandung oleh dokumen, lebih fokus pada substansi data, struktur data dan semantik data yang ditransfer tidak "hilang", dan telah menjadi standar *defacto* pertukaran data antar sistem.

2. Problem pada mekanisme pertukaran data. Solusi masalah ini dapat menggunakan *Service-Oriented Architecture* (SOA), yaitu sebuah skema yang memungkinkan komunikasi antar sistem dilakukan secara *loosely-coupled*, artinya masingmasing pihak tidak memiliki ketergantungan yang tinggi satu sama lain. Dalam SOA, komunikasi didasarkan pada konsep layanan menggunakan prinsip *client-server*, ada yang menyediakan layanan dan yang lain bisa meminta layanan. Permintaan layanan dilakukan dengan cara memanggil fungsi yang merepresentasikan layanan tersebut. Apabila fungsi dipanggil, maka aplikasi penyedia layanan wajib memberikan layanannya ke aplikasi pemanggil. Keunggulan SOA adalah detil internal yang terlibat dalam pemanggilan fungsi layanan sepenuhnya disembunyikan. Ada *interface* yang memisahkan secara tegas bagian publik (boleh diketahui oleh aplikasi lain) dan bagian privat (aplikasi lain tidak perlu tahu). Dengan *interface* tersebut, aplikasi *client* tidak perlu tahu tentang detil internal, namun cukup mengetahui sintaks fungsinya saja. SOA bisa mengakomodasi kepentingan *server* yang tidak perlu menunjukkan detil data yang sensitif/rahasia, sementara *client* tetap bisa meminta data yang diinginkannya kepada *server*.

2.4.2 *XML (Extensible Markup Language)*

XML merupakan bagian terpenting dari *programmer* yang ingin mengembangkan *Web Services*. Hal ini karena *XML* dibangun dengan kemampuan melakukan *transfer data* antar *platform*. *XML* juga memiliki kemampuan untuk integrasi data disamping pertukaran data antar *platform*.

Menurut Walsh (1998), *XML* merupakan sebuah *Markup Language* untuk dokumentasi terstruktur. Dokumen-dokumen terstruktur adalah dokumen-dokumen yang mempunyai isi/*content* (kata, gambar) serta indikasi yang menyatakan makna dari *content* tersebut. *XML* mempunyai kelebihan sebagai berikut (Tidwell, 1999):

a) XML tidak tergantung pada *platform* atau *system* operasi yang digunakan.

b) Hasil pencarian data lebih akurat.

c) Dokumen *XML* dapat diterjemahkan ke dalam beberapa format yang berbeda karena dalam *XML* data dan instruksi dipisahkan.

Ada 6 jenis *markup* yang bisa muncul dalam sebuah dokumen *XML* yaitu:

a) Elemen dan atribut. Elemen menyatakan sifat dari content yang dilingkupinya sedangkan atribut merupakan pasangan dari nama-nilai yang muncul dalam *tag* setelah nama elemen.

b) *Entity reference*, digunakan supaya tanda *markup* dapat dimasukkan ke dalam dokumen *XML* dan dianggap sebagai *content*.

c) *Comment* atau komentar.

d) *Processing Instruction* (PI), memungkinkan dokumen berisi suatu instruksi untuk suatu aplikasi.

e) *CDATA Section*. Dalam sebuah dokumen, *CDATA* Section menginstruksikan parser untuk mengabaikan karakter-karakter tertentu yang mungkin akan dikenali sebagai karakter *markup*.

f) *Document Type Declaration* (*DTD*). *DTD* berisi deklarasi *markup* yang memenuhi *grammar* untuk suatu kelas dokumen.

2.4.5 *Restful API*

RESTful API adalah sebuah *web service* yang diimplementasikan dengan menggunakan HTTP dan prinsip REST (*Representational State Transfer*), dan sumber dayanya tersimpan di penyimpanan data. *Web service* ini digunakan sebagai media pertukaran data antara sisi klien dengan sisi *server* yang menyimpan basis data. Klien dapat mengirimkan permintaan dan *server* akan memproses permintaan tersebut (seperti permintaan membuat, menerima, merubah, dan menghapus sumber daya). Setelah *server* selesai melakukan pemrosesan permintaan, *server* akan mengirimkan respon menuju klien sebagai hasil dari selesainya sebuah aksi. Format data yang dihasilkan dapat berupa *xml* atau *json*. Kedua format ini sangat umum digunakan dalam bidang pertukaran data dan didukung oleh banyak bahasa pemrograman. Format ini sangat ringan dan terbukti lebih hemat penggunaan memori saat melakukan pertukaran data. Aplikasi klien hanya perlu membaca format ini sehingga dapat diolah kembali untuk dipergunakan.

Dalam penelitian ini, pembangunan *RESTful API* mengembalikan data dengan *format json*. Format ini dipilih karena terbukti lebih cepat dan

membutuhkan sedikit sumber daya dibandingkan dengan penggunaan *xml* berdasarkan berbagai penelitian. *Format json* juga telah didukung oleh sistem operasi *Android* dengan menggunakan *library* yang sudah cukup banyak tersedia.

REST adalah *web service* yang menerapkan konsep perpindahan antar *state* dimana dalam bernavigasi *REST* melalui *link HTTP* untuk melakukan aktivitas tertentu. Dalam pengaplikasiannya *REST* banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource* sebagai layanannya dan bukan kumpulan dari aktifitas yang mengolah *resource* itu. Response dari *web service REST* dapat berupa *XML* atau *JSON*.

2.4.6 *Web Service Definition Language* (WSDL)

Sebelum mengakses sebuah *Web Services* pastinya perlu mengetahui *method-method* apa saja yang disediakan oleh *Web Services* tersebut, untuk 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.

2.5 *Database* dan *Software* Pendukung

2.5.1 Definisi *Database*

Menurut Frederick Constantianus (2012), Basis Data *(database)* adalah kumpulan data yang diorganisasikan agar informasi yang terkandung didalamnya dapat dengan mudah diakses, dikelola serta diperbaharui. Basis data digunakan untuk menyimpan, memanipulasi dan mengambil data hampir semua tipe perusahaan termasuk bisnis, pendidikan, rumah sakit, pemerintahan dan perpustakaan.

2.5.2 DBMS (*The Database Managements*)

Menurut Firdayanti, Meriza (2012), DBMS adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola dan mengendalikan pengaksesan basis data. Tujuan utama DBMS adalah menyediakan langkah yang nyaman dan efisien untuk penyimpanan dan pengambilan data dari basis data. DBMS berperan memberi abstraksi data tingkat tinggi ke pemakai.

2.5.3 *Mysql*

MySQL (bisa dibaca dengan mai-es-ki-el atau bisa juga mai-se-kuel) adalah suatu perangkat lunak *database* relasi (*Relational Database Management System* atau *DBMS*), seperti halnya *ORACLE*, *POSTGRESQL*, *MSSQL*, dan sebagainya. *SQL* merupakan singkatan dari *Structure Query Language*, idefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa program

yang digunakan untuk mengelola suatu *database*. Jadi *MySQL* adalah softwarenya dan *SQL* adalah bahasa perintahnya.

2.5.4 *Hypertext Preprocessor* (PHP)

HP: *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk memrogram situs web dinamis. *PHP* dapat digunakan untuk membangun sebuah *CMS*.

Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page* (Situs personal). *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya *PHP*/*FI*. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan *PHP*.

Program *Hello World* yang ditulis menggunakan *PHP* adalah sebagai berikut:

```
<?php
echo "Hello World";
?>
```

Koneksi *PHP* dengan *Database* :

1. Jika konek ke *database* berhasil, perlu memilih *database*.
2. Perintah untuk memilih *database*:

mysql_select*_db(*data_base*,pengenal_hubungan)** dimana : ***data_base = nama *database* **pengenal_hubungan** = nama pengenal yang digunakan dalam koneksi.

```
<HTML><BODY>
MEMILIH DATABASE<BR>
<?php
  $pemakai="root";
  $password="";
  $id_mysql=mysql_connect("localhost",$pemakai,$password);
  if(!$id_mysql)
    die("DATA BASE GAK BISA DIBUKA");
  if(! mysql_selectdb("coba",$id_mysql))
    die("DATA BASE TAK TERPILIH");
  mysql_close($id_mysql);
  print("SUKSES KONEKSI DAN SELEK DATABASE");
?>
</BODY></HTML>
```

Gambar 2.3 koneksi ke *database*

2.5.5 *Apache*

Apache merupakan *web server* yang ada pada dasarnya merupakan perangkat lunak khusus yang melayani permintaan-permintaan dari *browser web*akan dokumen-dokumen yang tersimpan didalamnya untuk menampilkan pada suatu *web browser*.

Apache merupakan *web server* yang paling banyak digunakan di *internet* saat ini. Ini disebabkan oleh beberapa factor seperti kecepatan, perfoma, kestabilan dan bersifat *free* atau gratis.

Keuntungan menggunakan *source code Apache* adalah kita dapat memodifikasi *source code* program *Apache* dengan leluasa sesuai dengan kebutuhan. Tetapi untuk dapat melakukannya dibutuhkan keterampilan

penguasaan bahasa C dan C++ serta pemrograman *network* (TCP/IP) (Bachtiar, 2008).

2.5.6 *Cascading Style Sheet* (CSS)

CSS digunakan dalam dokumen HTML untuk menciptakan suatu *style* (penyajian) yang dapat menbuat kemampuan HTML menjadi lebih luas. Yang menarik, *style* dapat didefinisikan pada *file* terpisah sehingga tidak menambah keruwetan dokumen. Selain itu, tentu saja *style* dapat digunakan pada sejumlah dokumen.

2.5.7 *JavaScript*

Pada awalnya *JavaScript* dikembangkan pada *web browser Netscape* oleh *Brendan Eich* dengan nama *Mocha*, kemudian berubah menjadi *LiveScript* dan yang akhirnya sampai sekarang ini menjadi *JavaScript*, *JavaScript* adalah bahasa skrip (*Scripting Language*) yaitu kumpulan instruksi perintah ayng digunakan untuk mengendalikan beberaqpa bagian dari sistem operasi. Bentuk bahasa skrip dari *JavaScript* mengambil model penulisan pada pemrograman Cdan JAVA, yang terdiri dari variable, fungsi dna lainnya (Sibero, 2011).

2.6 *Programming Diagram* dan *Data Dictionary*

2.6.1 *Flowcharts*

Menurut Adelia (2011), *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart*

menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara *sekuensial*. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

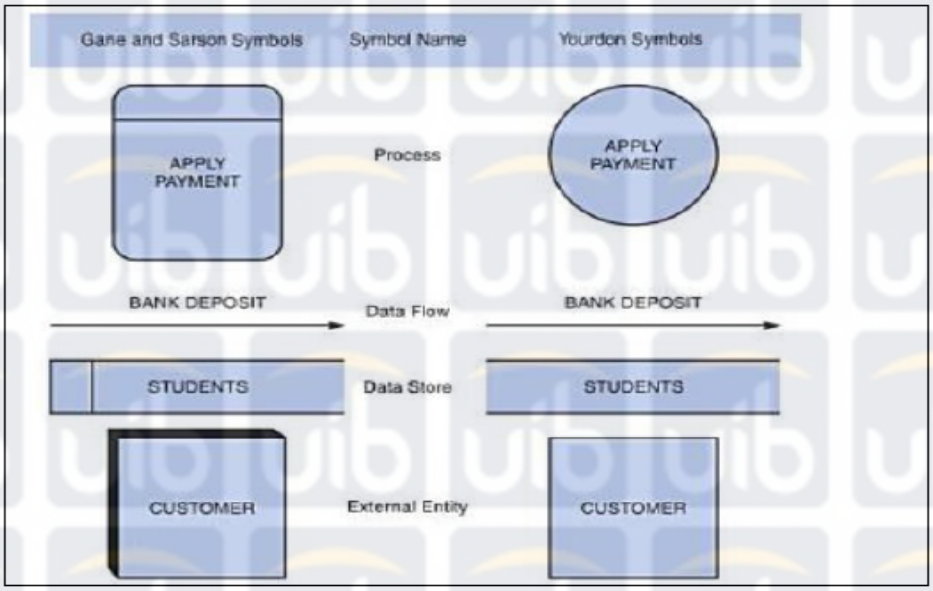
	*Terminal*	Simbol untuk menunjukkan awal, akhir, dan interupsi dalam sistem
	*Display*	Simbol yang menyatakan peralatan *output* yang digunakan yaitu *layer*, *plotter*, *printer*, dan sebagainya
	*Document*	Simbol yang menyatakan *input* berasal dari dokumen dalam bentuk kertas atau *output* dicetak ke kertas
	*Multiple Document*	Simbol yang menyatakan dokumen yang sama dicetak beberapa kali untuk kepentingan tertentu
	*Input/Output*	Simbol yang menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya
	*Magnetic Disk*	Simbol yang menyatakan data yang disimpan pada *magnetic disk*

Gambar 2.4 Simbol pada flowchart

2.6.2 *Data Flow Diagram* (DFD)

Data flow diagram adalah suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk dan simbol-simbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan. *Data flow diagram* ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem.

Dengan kata lain, *data flow diagram* adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. *Data flow diagram* ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. (Adelia, 2011)



Gambar 2.5 Simbol data flow diagram (DFD)