



BAB II

TINJAUAN PUSTAKA

Dalam bab ini akan diterapkan mengenai tinjauan pustaka yang bertujuan untuk membahas permasalahan yang diambil. Sesuai judul dalam pembuatan Tugas Akhir, yang akan dibuat penulis terdiri dari pengertian-pengertian sebagai berikut :

2.1. *Central Authentication Service (CAS)*

Central Authentication Service (CAS) Merupakan sebuah sistem otentikasi yang aslinya dibuat oleh Universitas Yale untuk menyediakan sebuah jalan yang aman untuk sebuah aplikasi untuk meng-otentikasi seorang pengguna. CAS kemudian diimplementasi sebagai sebuah *open source* komponen *server* Java dan mendukung *library* dari *client* untuk Java, PHP, Perl, *Apache*, *uPortal*, dan lainnya. Saat ini CAS di pegang oleh perusahaan JASIG dan dikembangkan untuk institusi tinggi pendidikan.

Central Authentication Service (CAS) Merupakan sebuah otentikasi yang membuat seluruh layanan yang ada dapat diotentikasikan menjadi satu, sehingga *user* hanya melakukan *login* satu kali untuk masuk ke semua aplikasi atau layanan yang ada.

2.2. *Pengertian XAMPP*

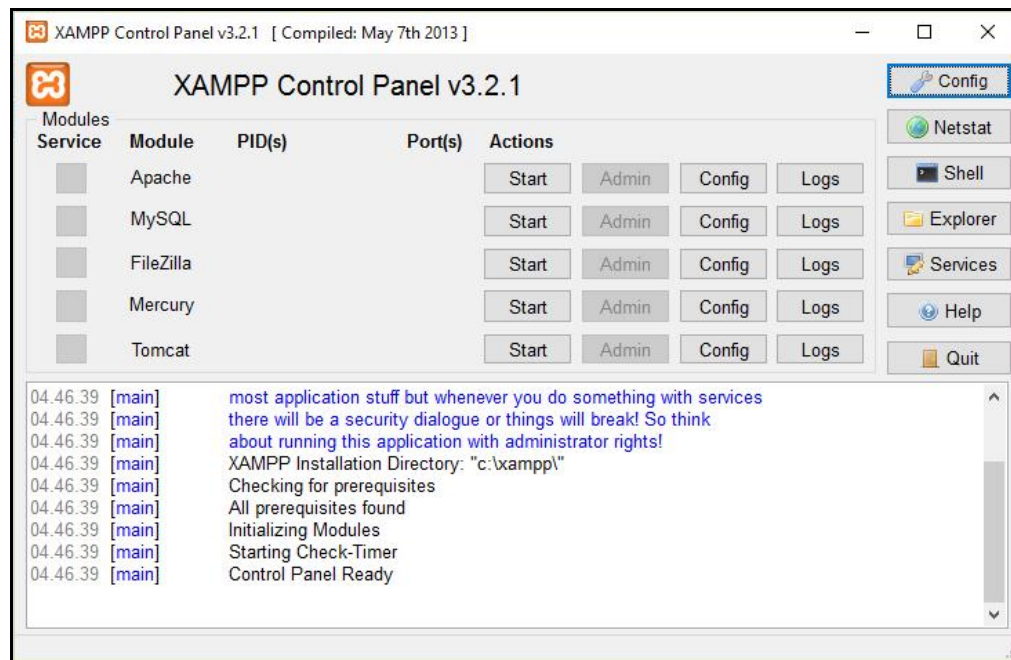
Wahana Komputer (2014:72), “XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket.”

Nugroho (2013), “XAMPP adalah paket program *web* lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya PHP dan MySQL”.

Buana (2014), “XAMPP adalah perangkat lunak *opensource* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows*, *linux*, *solaris*, dan *mac*”.



Jadi dari pengertian diatas dapat disimpulkan bahwa XAMPP adalah suatu perangkat lunak *opensource* untuk belajar pemrograman web.



Gambar 2.1. Tampilan XAMPP Control Panel

2.3. Single Sign On (SSO)

SSO merupakan teknologi yang memiliki kemampuan untuk memasukkan id dan password yang sama untuk login ke aplikasi dalam suatu perusahaan. SSO kini telah dikenal sebagai *reduced sign on* (RSO) sejak lebih dari satu jenis mekanisme otentikasi yang digunakan sesuai dengan model risiko perusahaan. Untuk jaringan yang sangat besar, pengguna diminta untuk mengisi informasi dirinya pada setiap aplikasi yang hendak di akses. Sistem SSO tidak memerlukan interaksi yang manual, untuk mengakses seluruh layanan aplikasi tanpa harus melakukan login dan mengetikkan *password*-nya berulang kali. SSO mengotentikasi pengguna pada semua aplikasi yang telah di-*authorized* untuk diakses. Ini menghilangkan permintaan authentication lagi ketika pengguna mengganti aplikasi selama *session* berlaku. SSO juga memperkenankan informasi autentikasi dan mengidentifikasi subjek secara ketat guna menghindari *login* ganda pada sistem atau kelompok sistem yang terpercaya.



2.4. Pengertian MySQL

Sibero (2013:97) menjelaskan bahwa MySQL atau dibaca “*My Sekuel*” adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. Berikut merupakan tabel beberapa pernyataan yang ada pada MySQL:

Tabel 2.1. Pernyataan MySQL

No.	Pernyataan	Kegunaan	Contoh
1	<i>INSERT</i>	Untuk memasukkan data ke dalam tabel.	<i>INSERT INTO</i> nama_tabel (kolom_1, kolom_2, ... kolom_n) <i>VALUES</i> ('nilai_kolom_1', 'nilai_kolom_2', ..., 'nilai_kolom_n');
2	<i>SELECT</i>	Untuk melihat isi tabel.	<i>SELECT</i> nama_kolom <i>FROM</i> nama_tabel;
3	<i>UPDATE</i>	Untuk mengubah data dalam tabel.	<i>UPDATE</i> nama_tabel <i>SET</i> kolom_1 = nilai_baru_1, kolom_2 = nilai_baru_2, ... kolom_n = nilai_baru_n <i>WHERE</i> kondisi;
4	<i>DELETE</i>	Untuk menghapus data dalam tabel.	<i>DELETE FROM</i> nama_tabel <i>WHERE</i> kondisi;

Sibero (2013:71) menjelaskan fungsi-fungsi PHP MySQL adalah sebagai berikut:

Tabel 2.2. Fungsi-fungsi PHP MySQL

No.	Fungsi MySQL	Penggunaan
1	<i>mysql_connect</i>	Membuka koneksi <i>database</i> .
2	<i>mysql_error</i>	Mengambil pesan kesalahan.
3	<i>mysql_fetch_array</i>	Mengambil hasil <i>record</i> dan mendefinisikan dalam bentuk array.
4	<i>mysql_num_rows</i>	Mengambil jumlah baris data hasil <i>query</i> .
5	<i>mysql_query</i>	Mengirimkan perintah <i>query</i> pada <i>server</i> .
6	<i>mysql_select_db</i>	Memilih dan mengaktifkan <i>database</i> .



2.5. Pengertian HTML

Sibero (2013:19), “*HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*.”

Suryana (2014:29), “*HyperText Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*.”

Jadi, dapat disimpulkan bahwa HTML (*HyperText Markup Language*) adalah bahasa yang digunakan untuk membuat halaman dokumen *web*.

Sibero (2013:19) menjelaskan bahwa struktur dokumen HTML sebagai berikut:

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

2.6. Web Service

Suryadi, dkk (2015) *Web service* adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web 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 (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*. *Web service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa *compiler*.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat



dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

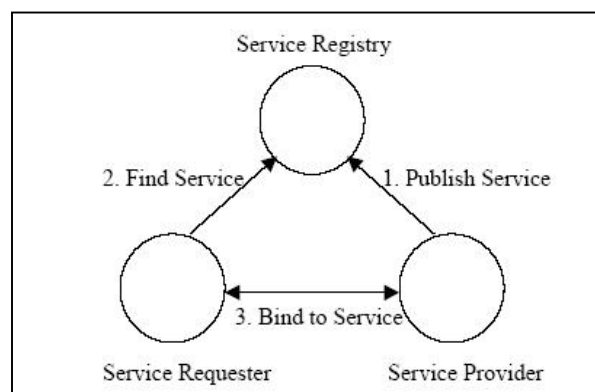
Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut:

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis *logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web service* berjalan di *port* 80 yang merupakan protokol standar HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus di sisi *firewall*.

2.6.1. Arsitektur Web Service

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

1. *Service Requester* (peminta layanan)
2. *Service Provider* (penyedia layanan)
4. *Service Registry* (daftar layanan)



Gambar 2.2 Arsitektur Web Service



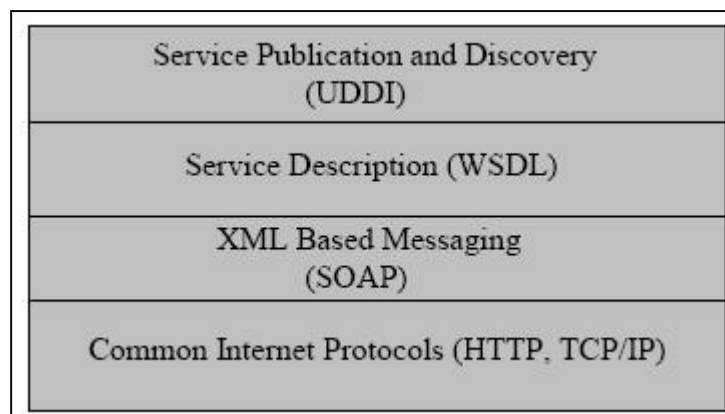
1. *Service Provider*: Berfungsi untuk menyediakan layanan/*service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.
2. *Service Registry*: Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*service* yang telah di-*register*.
3. *Service Requestor*: Peminta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

2.6.2. Operasi-Operasi Web Service

Secara umum, *web service* memiliki 3 (tiga) operasi yang terlibat di dalamnya, yaitu:

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*.

2.6.3. Komponen-Komponen Web Service



Gambar 2.3 Komponen Web Service



Web service secara keseluruhan memiliki 4 (empat) layer komponen seperti pada gambar di atas, yaitu:

1. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
2. Layer 2: *Simple Object Access Protocol* (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
3. Layer 3: *Web Service Definition Language* (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
4. Layer 4: *Service Publication and Discovery* (UDDI)

2.7. Pengertian CSS

Sibero (2013:), “*Cascading Style Sheet* (CSS) dikembangkan untuk menata gaya pengaturan halaman *web*.”

Suryana (2014:101), “CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis huruf, warna, dan semua yang berhubungan dengan tampilan.”

Wahana Komputer (2014:5) menjelaskan bahwa CSS merupakan bahasa pemrograman yang khusus menangani tampilan tiap elemen di dalam dokumen HTML. Dengan memanfaatkan CSS, struktur kode HTML kita akan terlihat rapi dan terstruktur.”

Jadi, dapat disimpulkan bahwa CSS adalah bahasa pemrograman yang digunakan untuk mengatur tampilan halaman HTML.

2.8. Pengertian PHP

Sibero (2013:49), “PHP adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut sebagai pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*.”



Wahana Komputer (2014:33), “PHP merupakan bahasa berbentuk *script* yang ditempatkan di dalam server baru kemudian diproses. Kemudian hasil pemrosesan dikirimkan kepada *web browser* klien. Bahasa pemrograman ini dirancang khusus untuk membentuk *web* dinamis.”

Jadi dapat disimpulkan bahwa PHP adalah bahasa pemrograman yang digunakan untuk membuat program berbasis *web*.

2.9. Metode Network Development Life Cycle (NDLC)

Network Development Life Cycle (NDLC) merupakan sebuah metode yang bergantung pada proses pembangunan sebelumnya seperti perencanaan strategi bisnis, daur hidup pengembangan aplikasi, dan analisis pendistribusian data. Tahapan pada Network Development Life Cycle (NDLC) :

1. Analisis
2. Design
3. Simulation Prototype
4. Implementation
5. Monitoring
6. Management

2.10. Referensi Jurnal

Penelitian tentang *Single Sign On* sudah banyak dilakukan. Dari beberapa penelitian yang dilakukan. Berikut ini beberapa referensi jurnal yang melakukan penelitian tentang *Single Sign On*.

Udayana dan Jasa (2016), Mengemukakan bahwa dengan menggunakan SSO, user hanya perlu melakukan satu kali login dalam mengakses semua aplikasi web yang terintegrasi. Untuk mengimplementasikan *system* SSO digunakan *Central Authentication Service* (CAS) sebagai pusat otentikasi dengan memanfaatkan struktur *Lightweight Directory Access Protocol* (LDAP) untuk manajemen user.

Kunang dan Yadi (2014), Mengemukakan bahwa banyaknya sistem dan aplikasi yang digunakan tersebut memberikan kendala tersendiri bagi pengguna



untuk mengingat *user* dan *password login* pada masing-masing sistem tersebut. Untuk itu dikembangkan *Single Sign On* yang aman dan bisa diimplementasikan di Universitas Bina Darma yang mengintegrasikan sistem *elearning*, blog dan *mail server*. Dengan sistem yang dikembangkan memungkinkan pengguna hanya perlu melakukan satu kali *login* pada ke 3 (tiga) sistem tersebut.

Sahare (2017), Mengemukakan bahwa Teknologi *Single Sign On* (SSO) telah diperkenalkan sebagai bentuk khusus mekanisme otentikasi. Teknologi ini dimaksudkan untuk memudahkan pekerjaan bagi pengguna dalam cara otentikasi, dengan satu kali otentikasi mereka bisa dapat mengakses ke beberapa aplikasi tanpa melakukan *login* berulang kali.

Garg dan Singh (2016), dalam jurnalnya menyatakan bahwa studi kasus pada proyek untuk memberikan solusi *Single Sign On* (SSO) untuk aplikasi berbasis web yang menggunakan mainframe sebagai penyimpanan data. Ini adalah proses yang memungkinkan pengguna terautentikasi untuk memasukkan *Username* dan *Password* mereka untuk mengakses berbagai aplikasi. Prosesnya dilakukan dengan menggunakan *Lightweight Directory Access Protocol* (LDAP) sebagai *server* penyimpanan.

Riswandha (2016), dalam jurnalnya menyatakan bahwa SSO adalah sebuah sistem dimana pengguna hanya menggunakan satu *username* dan satu *password* untuk menggunakan layanan integrasi tersebut dan sistem tersebut memberikan kemudahan bagi pengelola *user* atau *administrator* dan diharapkan memberikan kemudahan dan keamanan bagi pengguna layanan aplikasi *mobile office* online. Solusi ini dibangun pada jaringan *private cloud computing* yang terenkripsi dan terintegrasi dengan *Active Directory Domain Controller* untuk *Single Sign On* yang memberikan kemudahan pengguna dalam menggunakan aplikasi-aplikasi tersebut.