



---

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian *Framework*

Salah satu alasan mengapa orang menggunakan framework terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah framework biasanya sudah tersedia struktur aplikasi yang baik, *standard coding*, *best practice*, *design pattern*, dan *common function*. Dengan menggunakan *framework* kita dapat langsung fokus kepada *business process* yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain.

Menurut Hakim (2010:3) menjelaskan bahwa, *Framework* adalah koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal.

Sedangkan menurut Raharjo (2015:2), *Framework* adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi *web*.

Jadi, *Framework* adalah kumpulan-kumpulan potongan program yang dipadukan menjadi satu kerja kerja yang digunakan untuk membantu dalam pembuatan sebuah aplikasi.

##### 2.1.2 Pengertian *CodeIgniter*

Proses pengembangan *web* dapat dilakukan dengan beragam bahasa pemrograman seperti *PHP*, *Python*, *Ruby*, *Perl*, *C++*, *JAVA* dan sebagainya. Saat ini, banyak bermuculan *framework web* yang dirancang untuk bahasa-bahasa pemrograman tersebut. Salah satunya adalah *Code Igniter*.



Menurut Hakim (2010:3) *CodeIgniter* adalah sebuah *framework* PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal.

Sedangkan, menurut Raharjo (2015:3) *CodeIgniter* adalah *framework web* untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab.

Jadi *CodeIgniter* adalah sebuah *framework* buatan Rick Ellis yang digunakan untuk mempermudah pada *developer* dalam mengembangkan suatu aplikasi *web*.

### 2.1.3 Keuntungan Menggunakan *CodeIgniter*

Menurut Raharjo (2015:4) *CodeIgniter* merupakan sebuah *toolkit* yang ditujukan untuk orang yang ingin membangun aplikasi *web* dalam bahasa pemrograman *PHP*. Beberapa keunggulan yang ditawarkan oleh *CodeIgniter* adalah sebagai berikut :

1. *CodeIgniter* adalah *framework* yang bersifat *free* dan *opensource*.
2. *CodeIgniter* memiliki ukuran yang kecil dibandingkan dengan *framework* lain. Setelah proses instalasi, *framework CodeIgniter* hanya berukuran kurang lebih 2 MB. Dokumentasi *CodeIgniter* memiliki ukuran sekitar 6 MB.
3. Aplikasi yang dibuat menggunakan *CodeIgniter* bisa berjalan cepat.
4. *CodeIgniter* menggunakan pola desain *Model-View-Controller* (MVC) sehingga satu *file* tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara dikemudian hari.
5. *CodeIgniter* dapat diperluas sesuai dengan kebutuhan.
6. *CodeIgniter* terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh *CodeIgniter* dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.



## 2.2 Teori Khusus

### 2.2.1 Pengertian Pemrograman Berorientasi Objek

Menurut Sukamto dan Shalahuddin (2014:100) Berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.

### 2.2.2 Metodologi RUP (*Rational Unified Process*)

Menurut Sukamto dan Shalahuddin (2014:124) menjelaskan bahwa *RUP* (*Rational Unified Process*) adalah tahapan pengembangan sistem secara iteratif khusus untuk pemrograman berorientasi objek. *RUP* menyediakan pendefinisian yang baik untuk alur hidup proyek perangkat lunak.

### 2.2.3 Penerapan Tahap Metodologi Pengembangan Perangkat dengan RUP

Menurut Sukamto dan Shalahuddin (2014:128) dalam *Rational Unified Process* terdapat empat tahap pengembangan perangkat lunak yaitu :

1. *Inception* (Permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (Perluasan/Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem dan juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Pada tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada punawarna sistem (*prototype*).

3. *Construction* (Konstruksi)

Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini



menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasi awal.

#### 4. *Transition* (Transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari batas tonggak kemampuan operasi awal. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

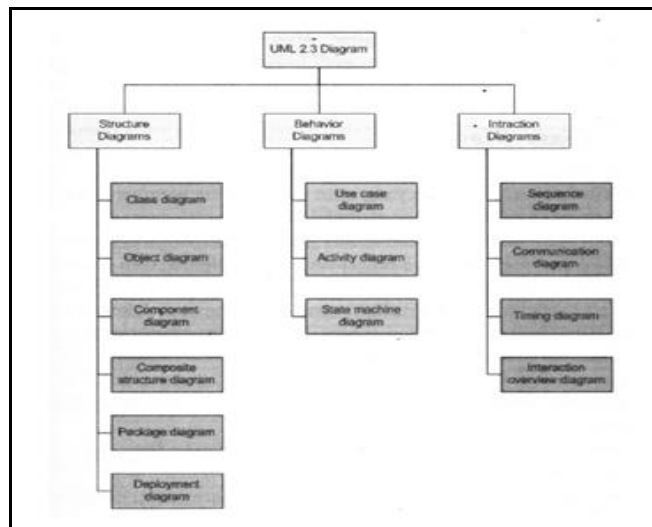
### 2.2.4 Pengertian UML (*Unified Modeling Language*)

Sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu *Unified Modelling Language (UML)*. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

Menurut Sukamto dan Shalahuddin (2014:133) menjelaskan *UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

### 2.2.5 Macam-Macam Diagram UML

Menurut Sukamto dan Shalahuddin (2014:140), Pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



**Gambar 2.1** Diagram *UML*

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. *Behavior Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

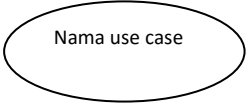
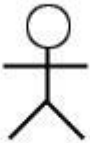

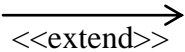
c. *Interaction Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.


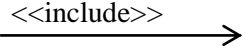
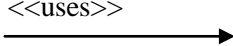
## 2.2.6 Pengertian Diagram *Use Case*

Sukamto dan Shalahuddin (2014:155) menjelaskan bahwa *Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2.1** Simbol-simbol Diagram *Use Case*

No	Simbol	Keterangan
1.	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anatar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama use case
2.	<p>Aktor/<i>actor</i></p> 	Orang , proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
3.	<p>Asosiasi/ <i>association</i></p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	<p>Ekstensi/ <i>extend</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

Lanjutan **Tabel 2.1** Simbol-simbol Diagram *Use Case*

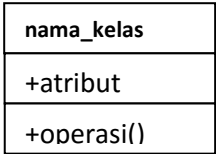

No	Simbol	Keterangan
5.	Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.	Menggunakan/ <i>include/use</i>   	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber : Sukamto dan Shalahuddin (2014:156-158)


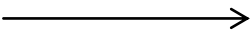
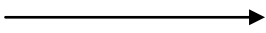
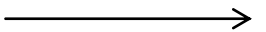

### 2.2.7 Pengertian Diagram Kelas

Sukamto dan Shalahuddin (2014:75) menjelaskan bahwa Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2.2** Simbol-simbol Diagram Kelas

No	Simbol	Keterangan
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>Interface</i>  <b>nama_interface</b>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Lanjutan **Tabel 2.2** Simbol-simbol Diagram Kelas


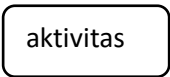
No	Simbol	Keterangan
3.	Asosiasi/ <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	Kebergantungan/ <i>dependency</i> 	Kebergantungan antarkelas
7.	Agregasi/ <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

Sumber : Sukamto dan Shalahuddin (2014:146-147)

### 2.2.8 Pengertian Diagram Aktivitas

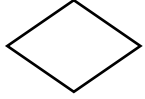



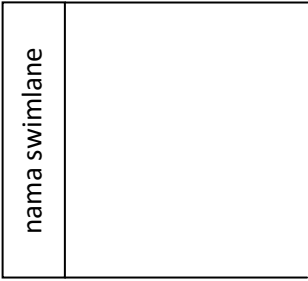
Sukamto dan Shalahuddin (2014:161) menjelaskan bahwa, Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

**Tabel 2.3** Simbol-simbol Diagram Aktivitas

No	Simbol	Keterangan
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja



Lanjutan **Tabel 2.3** Simbol-simbol Diagram Aktivitas

No	Simbol	Keterangan
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	<i>Swimlane</i>  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.


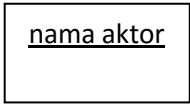

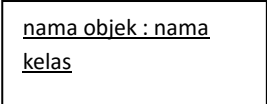

Sumber : Shalahuddin (2014 : 162-163)



### 2.2.9 Pengertian Diagram Sekuensial

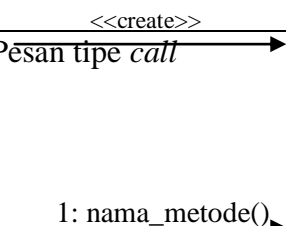
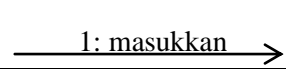
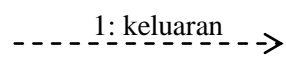
Sukanto dan Shalahuddin (2014:165) menjelaskan bahwa, Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada sekuen diagram:

**Tabel 2.4** Simbol-simbol dalam Diagram Sekuensial

N	Simbol	Keterangan
1.	<p>Aktor</p>  <p>atau</p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.	<p>Garis hidup/ <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya



Lanjutan Tabel 2.4 Simbol-simbol dalam Diagram Sekuensial

No	Simbol	Keterangan
5.	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.		Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.		Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek yang lainnya, arah panah mengarah pada objek yang dikirim
8.		Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber: Shalahuddin (2014:165-167)



## **2.3 Pengertian Judul**

### **2.3.1 Pengertian Aplikasi**

Menurut Budiharto (2013:5) menjelaskan bahwa, Aplikasi merupakan program yang dapat berjalan di komputer tersendiri (*stand alone computer*), dari mulai program yang simpel sampai dengan program besar dan rumit.

Menurut Maseleno (2003:10) menjelaskan bahwa, Aplikasi adalah *Software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu.

Jadi, aplikasi adalah program atau *software* yang dapat berjalan disebuah komputer untuk mengerjakan tugas-tugas tertentu.

### **2.3.2 Pengertian Web**

Salah satu elemen yang paling menarik dalam internet ialah adanya *World Wide Web* yang merupakan teknologi canggih saat ini dalam dunia internet. *WWW* atau sering disingkat *web*, berisi halaman-halaman yang dapat menampilkan teks, gambar, grafik, suara, animasi, serta elemen-elemen yang ditampilkan bersifat interaktif.

Sibero (2014:11) menyatakan, *World Wide Web (W3)* atau yang dikenal juga dengan istilah *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet.

Suryana (2014:15), *World Wide Web* merupakan jaringan yang menghubungkan jaringan-jaringan lokal ke dalam suatu jaringan global, di mana satu komputer di negara tertentu, dapat langsung berkomunikasi dengan komputer lain di negara lain.”

Jadi, *World Wide Web* merupakan jaringan yang menghubungkan secara global yang digunakan sebagai media untuk menampilkan berbagai teks, maupun gambar.



### **2.3.3 Pengertian Pemesanan**

Dalam Kamus Besar Bahasa Indonesia, Pemesanan proses, perbuatan, cara memesan atau memesankan.

### **2.3.4 Pengertian Gedung**

Menurut Sugono (2008:446), Gedung adalah bangunan (rumah) untuk maksud tertentu

Menurut Menteri Pekerjaan Umum Republik Indonesia (2007:1) menjelaskan bahwa, Gedung adalah wujud fisik hasil pekerjaan konstruksi yang menyatu dengan tempat dan kedudukannya, sebagian atau seluruhnya berada di atas dan/atau di dalam tanah dan/atau air, yang berfungsi sebagai tempat manusia melakukan kegiatan, baik untuk hunian atau tempat tinggal, kegiatan keagamaan, kegiatan usaha, kegiatan sosial, budaya, maupun kegiatan khusus.

Jadi, Gedung adalah bangunan yang berfungsi sebagai tempat tinggal, maupun kegiatan khusus.

### **2.3.5 Pengertian Aula**

Menurut Sugono (2008:104) menjelaskan bahwa, Aula adalah ruangan besar untuk berapat atau mengadakan upacara dan sebagainya.”

### **2.3.6 Pengertian Aplikasi Web Pemesanan Gedung Aula pada SMK Negeri 3 Palembang**

Aplikasi Web Pemesanan Gedung Aula pada SMK Negeri 3 Palembang adalah sebuah aplikasi yang dibangun untuk mempermudah konsumen dalam melakukan proses pemesanan gedung aula dimana pun dan kapan pun.



## 2.4 Teori Program

### 2.4.1 Pengertian *PhpMyAdmin*

Menurut Nugroho (2004:65) menjelaskan bahwa, *PhpMyAdmin* adalah suatu program *open source* berbasis web yang dibuat menggunakan aplikasi *PHP*.

### 2.4.2 Pengertian HTML (*HyperText Markup Language*)

Menurut Sibero (2014:19) menjelaskan bahwa, *HyperText Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*.

### 2.4.3 Pengertian XAMPP

Menurut Wahana (2014:72) menjelaskan bahwa, XAMP adalah singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket.

### 2.4.4 Pengertian *Cascading Style Sheet* (CSS)

Menurut Sibero (2014:112) menjelaskan bahwa, *Cascading Style Sheet* dikembangkan untuk menata gaya pengaturan halaman *web*. *Cascading Style Sheet* memiliki arti gaya menata halaman bertingkat, yang berarti setiap satu elemen yang telah diformat dan memiliki anak dan telah diformat, maka anak dari elemen tersebut secara otomatis mengikuti format elemen induknya.

### 2.4.5 Pengertian *MySQL*

Menurut Kadir (2008:2) menjelaskan bahwa, *MySQL* (baca: mai-se-kyu-el) merupakan *software* yang tergolong *DBMS* (*Database Management System*) yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *men-download* (mengunduh) di internet secara gratis.



#### **2.4.6 Adobe Dreamweaver**

Menurut Wahana (2009:2) menjelaskan bahwa, *Dreamweaver* adalah suatu aplikasi yang dapat dimanfaatkan untuk merancang desain *web* secara visual atau aplikasi editor *web*.

##### **2.4.6.1 Adobe Dreamweaver CS4**

Aplikasi *adobe dreamweaver* menyediakan banyak fasilitas yang akan memudahkan dalam mendesain dan menghasilkan sebuah situs *web* yang interaktif dan profesional. Salah satu versi terbaru dari *Adobe Dreamweaver* adalah *adobe dreamweaver CS4*.

Menurut Wahana (2009:2) menjelaskan bahwa, *Adobe dreamweaver CS4* adalah versi terbaru dari *Adobe dreamweaver* yang sebelumnya, *Adobe dreamweaver CS3*.



**Gambar 2.2** Logo *Dreamweaver CS4*



## 2.4.7 PHP

### 2.4.7.1 Pengertian PHP

Menurut Sibero (2014:49) menjelaskan bahwa, *PHP* adalah proses penerjemah baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan.



**Gambar 2.3** Tampilan Logo *PHP*

### 2.4.7.2. Script *PHP*

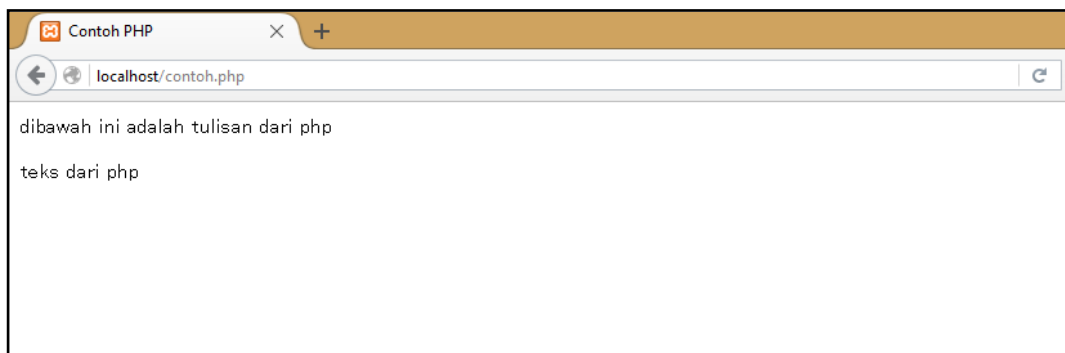
*PHP* adalah program yang fleksibel, artinya *script-script PHP* dapat Anda tuliskan pada sela-sela tag *HTML*. Karena *PHP* memiliki sifat yang dapat berkonteraksi dengan program lain, maka kita tidak usah bingung dalam menuliskan skripnya. Jadi, kita dapat membuat sebuah program *PHP* yang di dalamnya berupa tag-tag *HTML* murni yang kemudian disisipi *script PHP*. Akan tetapi, *PHP* juga dapat berdiri sendiri sehingga tidak memerlukan *HTML* murni untuk meletakkan *script-scriptnya*.



**Contoh script:**

```
<HTML>
<HEAD>
<TITLE> Contoh PHP </TITLE>
</HEAD>
<BODY>
<p> dibawah ini adalah tulisan dari php</p>
<?php
    echo 'teks dari php';
?>
</BODY>
</HTML>
```

Bila dijalankan melalui browser, maka kode tersebut akan membentuk tampilan seperti berikut:



**Gambar 2.4** Contoh Script PHP