

Chapter 2

PHP & Object Oriented Programming

Syarat utama untuk menguasai CodeIgniter adalah memahami PHP dan konsep Object Oriented Programming. Pada bab ini kita akan membahas sedikit mengenai PHP dan OOP.

Apasih PHP?

PHP (PHP: Hypertext Preprocessor) adalah sebuah bahasa pemrograman di sisi server. Ketika Anda mengakses sebuah URL, maka web browser akan melakukan request ke sebuah web server. Misalnya kita me-request sebuah file PHP <http://www.koder.web/index.php> , maka webserver akan melakukan *parsing* terhadap file PHP tersebut. PHP parser yang menjalankan kode-kode PHP yang terdapat pada file index.php lalu mengirimkan hasilnya ke web browser.

Sejarah PHP

Pada tahun 1994, Rasmus Lerdorf mengembangkan sebuah perkakas yang digunakan sebagai *engine parsing* sebagai penerjemah/interpreter beberapa *macro*. Pada saat itu engine digunakan untuk pembuatan buku tamu, counter dan beberapa homepage. Ia menamai engine parser tersebut dengan nama PHP/FI.

Dengan semangat *opensource*, para pengembang di dunia mencoba mengembangkan PHP/FI. Sampai pada tahun 1997, lebih dari 500.000 website di dunia menggunakan PHP/FI untuk menyelesaikan masalah seperti koneksi ke database, menampilkan content yang dinamis dan lain-lain.

Pada juni 1998, PHP 3.0 dirilis. Pada saat itu PHP sudah mendukung *multiflatform* (bukan hanya linux), webserver, sejumlah database, SNMP (Simple Network Management Protocol) and IMAP (Internet Message Access Protocol). Menurut survei yang dilakukan oleh <http://netcraft.org>, saat ini pengguna PHP sudah mencapai 9,5 juta domain.

Menggunakan PHP

Ketika kita ingin mempelajari PHP, hal pertama yang harus disiapkan adalah sebuah web server, seperti yang sudah dibahas sebelumnya bahwa PHP merupakan bahasa pemrograman di sisi server. Banyak sekali jenis web server yang dapat digunakan, antara lain Apache, IIS, iPlanet, Omni, Xintami, dan lain-lain. Setelah web server terinstal barulah kita install PHP sebagai modul.

PHP dapat diperoleh secara gratis dengan cara *men-download* dari situs resmi PHP (<http://www.php.net/downloads.php>) atau website lain yang mempunyai salinan program PHP untuk di-*download*.

Sebelum menginstall PHP, terlebih dahulu harus meng-install web server. Setelah PHP terinstall. Anda dapat meletakkan *source code* PHP ke dalam folder yang akan di publish oleh web server dengan ekstensi file **.php**.

Jika tidak ingin direpotkan oleh proses konfigurasi dan implementasi webserver, Anda dapat menggunakan software yang sudah terpaket menjadi satu antara PHP, APACHE, dan MySQL. Contoh Software-nya adalah XAMPP, PHPTRiad, FOX Server, dan lain-lain.

Apa sih Object Oriented Programming (OOP)?

Object Oriented Programming (OOP) merupakan paradigma pemrograman yang berorientasikan kepada obyek. Semua data dan fungsi pada paradigma ini dibungkus dalam kelas-kelas atau obyek-obyek. Bandingkan dengan logika pemrograman terstruktur, setiap obyek dapat menerima pesan, memproses data, dan mengirim pesan ke obyek lainnya. OOP diciptakan untuk mengatasi keterbatasan pada bahasa pemrograman tradisional. Konsep dari OOP sendiri adalah semua pemecahan masalah dibagi ke dalam obyek. Dalam konsep OOP data dan fungsi-fungsi yang akan mengoperasikannya digabungkan menjadi satu kesatuan yang dapat disebut sebagai obyek.

Apa itu Object

Sederhananya, sebuah obyek adalah kumpulan dari variabel dan fungsi yang dibungkus menjadi satu entitas. Entitas tersebut dapat berupa variabel biasa. Sebuah obyek diciptakan melalui sebuah kelas atau dengan istilah *instance of class*. Obyek memiliki 2 elemen utama:

1. **Attributes** atau **Properties**: Yaitu nilai-nilai yang tersimpan dalam objek tersebut dan secara langsung maupun tidak langsung menentukan karakteristik dari obyek tersebut.
2. **Method**: Yaitu suatu aksi yang akan dijalankan atau dikerjakan oleh obyek tersebut.

Apa itu Class?

Class dapat didefinisikan sebagai struktur data atau cetak biru dari suatu obyek. Lebih jelasnya adalah sebuah bentuk dasar atau *blueprint* yang mendefinisikan variabel, method umum pada semua obyek. Obyek sendiri adalah kumpulan variabel dan fungsi yang dihasilkan dari template khusus atau disebut class. Obyek adalah elemen pada saat run-time yang akan diciptakan, dimanipulasi, dan dibuang/di-*destroy* ketika eksekusi. Adapun class merupakan definisi statik dari himpunan obyek yang mungkin diciptakan sebagai instantiasi dari class.

Perhatikan contoh class di berikut ini.

```
<?php
/** Contoh kelas **/
class Kendaraan{}
/** end of class **/
?>
```

Contoh diatas memperlihatkan bagaimana mendefinisikan sebuah class dan meng-create sebuah instance dari class. Pada contoh di atas kita membuat sebuah kelas bernama “Kendaraan”. Dalam pembuatan kelas, pertama kita menggunakan kata kunci **class** yang diikuti oleh **nama kelas**, kemudian diakhiri dengan kurung kurawal. Di dalam kurung kurawal kita menuliskan kode-kode (berisi property dan method) supaya kelas tersebut bekerja seperti yang diinginkan.

Kode-kode di dalam sebuah kelas terbagi menjadi dua kelompok, yaitu **property** dan **method**. Property adalah suatu wadah penyimpanan di dalam kelas yang bisa menampung informasi. Sederhananya property itu bisa disebut sebagai variabel di dalam kelas. Sedangkan method adalah fungsi yang ada di dalam kelas. Perhatikan contoh berikut.

```
<?php
/** contoh kelas **/
class kendaraan{

    /** property class **/
    private $warna;
    private $jumlah_pintu;
    private $jumlah_roda;
    public $harga;
    public $merk;

    /** method class **/
    public function __construct(){
        echo 'ini adalah object kendaraan. <br />';
    }

    public function set_harga($harga){
        $this->harga = $harga;
    }

    public function show_harga(){
        echo 'harga kendaraan : rp.'. $this->harga.'. <br />';
    }

    public function jalan(){
        echo 'brrroooooom!!!';
    }

}
/** end of class **/
/** contoh object **/
#mengcreate object $saya_adalah_object dari class kendaraan
$saya_adalah_object= new kendaraan;
$saya_adalah_object->set_harga(100000);
$saya_adalah_object->show_harga();
```

```
$saya_adalah_object->jalan();
```

Setiap property dan method memiliki *identifier*. Identifier-lah yang mengatur bagaimana property dan method digunakan. Identifier tersebut adalah **public**, **private** dan **protected**. Private berarti method atau property yang ada di dalam suatu kelas hanya bisa diakses di dalam kelasnya. Sedangkan pada method atau property yang bersifat public berarti method atau property tersebut bisa diakses di dalam dan di luar kelas.

Inheritance

Inheritance atau dalam bahasa Indonesianya disebut sebagai pewarisan adalah suatu cara untuk membuat sebuah kelas yang baru dengan menggunakan kelas lain yang sebelumnya sudah dibuat. Pada hubungan inheritance, sebuah class turunan mewarisi kelas leluhur (parent class). Oleh karena mewarisi, maka semua atribut dan method class dari induk akan dibawa (kecuali yang bersifat private), secara intrinsik menjadi bagian dari class anak. Adapun keuntungan yang didapat dari inheritance menambah fitur baru pada kelas anak dan mengubah atau mengganti fitur yang diwarisi dari kelas parent

Adapun contoh kelas yang menggunakan konsep inheritance adalah

```
<?php
/* inheritance.php */
class Bapak {
    private $nama = "Bapak";
    function Bapak($n) {
        $this->nama = $n;
    }
    function Hallo() {
        echo "Halo, saya $this->nama <br>";
    }
}

class Anak extends Bapak {
    function Hai(){
        Echo "hai dari kelas anak";
    }
}

$test = new Anak("Anak dari Bapak");
$test->Hallo();
```

Jika kita perhatikan di kelas anak sama sekali tidak memiliki fungsi hello, tetapi karena parentnya memiliki fungsi tersebut maka si anak dapat menggunakan fungsi tersebut. Selain menggunakan fungsi bapak, anak juga dapat menambah fungsi baru yaitu fungsi Hai.