

Pertemuan 10

Pemrograman Berorientasi Objek dalam PHP

- ❖ Pengantar Pemrograman Berorientasi Objek
- ❖ Object dan Class
- ❖ Properties dan Method
- ❖ Mendefinisikan Class

Pengantar Pemrograman Berorientasi Objek

PHP pada awalnya hanyalah kumpulan script sederhana. Dalam perkembangannya, selanjutnya ditambahkan berbagai fitur pemrograman berorientasi objek. Hal ini dimulai sejak PHP 4. Dengan lahirnya PHP 5, fitur-fitur pemrograman berorientasi objek semakin mantap dan semakin cepat. Dengan PHP 5, script yang menggunakan konsep *object-oriented* akan lebih cepat dan lebih efisien.

Pemrograman berorientasi objek atau *object-oriented programming* (OOP) merupakan suatu pendekatan pemrograman yang menggunakan object dan class. Saat ini konsep OOP sudah semakin berkembang. Hampir setiap perguruan tinggi di dunia mengajarkan konsep OOP ini pada mahasiswanya. Pemrograman yang banyak dipakai dalam penerapan konsep OOP adalah Java dan C++.

OOP bukanlah sekedar cara penulisan sintaks program yang berbeda, namun lebih dari itu, OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah *object*. Sebuah *object* mewakili suatu bagian program yang akan diselesaikan.

Beberapa konsep OOP dasar, antara lain :

1. *Encapsulation* (Class dan Object)
2. *Inheritance* (Penurunan sifat), dan
3. *Polymorphisme*

PHP khususnya PHP 5 sudah mendukung beberapa konsep OOP. Akan tetapi PHP 5 tidak mendukung konsep *Multiple-inheritance* dan *polymorphisme*.

Object dan Class

Bagian dasar dari sebuah program yang berorientasi objek adalah **objects**. Secara mudah kita dapat memahami mengenai *object* ini. Sebagai contoh, sebuah **mobil** adalah objek. Sebuah mobil mempunyai *properties* atau bagian-bagian di dalamnya, seperti warna, mesin, roda, pintu dsb. Sebuah mobil juga dapat melakukan sesuatu (ada sesuatu yang bisa dilakukan dengan mobil), seperti mengisi bensin, menyalakan mesin, berjalan, mengerem dsb.

Biasanya *object* adalah sebuah kata benda. **Orang** adalah *object*. Demikian juga mobil, pohon, bunga, komputer, TV, buku dsb. Namun, *object* tidak selamanya sebuah objek fisik. Bisa saja sebuah benda abstrak, seperti *account* bank, sebuah file di komputer, database, pesan email, acara TV, dsb.

Class merupakan penjelasan atau deskripsi dari object. Di dalam class, terdapat penjelasan tentang suatu *object* termasuk properties yang dimilikinya serta kelakuan atau method yang bisa dilakukan oleh object. Sebagai contoh, class **Orang**. Class Orang tentu setidaknya memiliki beberapa bagian seperti tangan, kaki, mata, telinga dsb. Class Orang juga setidaknya harus bisa jalan, bisa loncat, bisa lari, bisa melihat, bisa bicara dsb.

Salah satu keuntungan program didefinisikan dengan konsep OOP adalah adanya pengkapsulan (*encapsulation*) program dalam *class* dan *object*, dimana programmer yang menggunakan *class* tidak perlu mengetahui isi dan jalannya *class* secara detail, hanya perlu tahu bagaimana cara menggunakannya. Sama halnya dengan sebuah **mobil** misalnya. Seorang pemilik mobil tentunya tidak

perlu mengetahui bagian-bagian mobil secara menyeluruh. Dia tidak perlu mengetahui bagaimana mesin mobil melakukan pembakaran dan bagaimana mesin mobil bisa menggerakkan roda, dsb. Dia hanya perlu tahu bagaimana cara menjalankan mobil, bagaimana menghentikan mobil, dan fungsi mobil lainnya.

Properties dan Method

Setiap class memiliki *properties* yang kadang disebut juga *attributes*. *Properties* dari sebuah mobil misalnya warna, ukuran, harga dsb. Di dalam class, *properties* dinyatakan dengan sebuah variabel. Misalnya \$warna, \$harga, dsb.

Method merupakan sesuatu yang bisa dilakukan oleh *object*. Method dalam PHP sama artinya dengan sebuah fungsi. Method yang mungkin dipunyai dari sebuah mobil misalnya, method untuk menghidupkan mobil, menjalankan mobil, menghentikan mobil, dsb.

Penamaan *properties* dan *method* memiliki aturan yang sama dengan penamaan sebuah variabel atau fungsi. Akan tetapi berdasarkan kesepakatan (*convention*), penamaan *properties* dan *method* harus menggunakan *camel Caps*, dimana tiap kata diawali dengan huruf besar kecuali kata pertama, setiap kata digabung tanpa spasi atau *under-score* (_).

Mendefinisikan Class

Bentuk umum mendefinisikan sebuah *class* adalah sbb :

```
class namaClass
{
    Deklarasikan dan definisikan properties di sini
    Definisikan semua method di sini
}
```

Penamaan **namaClass** pada dasarnya sama dengan penamaan variabel. Penamaan bebas, boleh apa saja, kecuali **stdClass**. PHP sudah menggunakan nama **stdClass** sebagai nama *class built-in*. Isi tubuh class terletak di antara tanda kurung kurawal buka ({) dan kurawal tutup (}). Di tubuh class terdapat pendefinisian properties (variabel) dan method-method class.

Menambahkan Properties (Variabel)

Perhatikan contoh pendefinisian class beserta propertiesnya, sbb :

```
1 class Mobil
2 {
3     var $warna;
4     var $merk;
5     var $harga;
6
7     //Tambahkan pendefinisian method di sini
8 }
```

Pendeklarasian variabel dalam class seperti pada contoh di atas, bukanlah suatu keharusan, karena dalam PHP variabel tidak perlu dideklarasikan, cukup digunakan saja. Namun demikian, pendeklarasian variabel yang digunakan sangat dianjurkan untuk kemudahan pembacaan dan pemahaman program.

Variabel class juga dapat langsung diinisialisasi dengan sebuah nilai. Namun inisialisasi variabel tidak boleh mengandung operasi aritmatika maupun operasi yang lainnya. Perhatikan contoh sebagai berikut :

```
1 class Mobil
2 {
3     var $warna = "Biru";
4     var $merk = "BMW";
5     var $harga = "10000000";
6 }
```

```

7 //Tambahkan pendefinisian method di sini
8 }

```

Menambahkan Method

Untuk menambahkan method, tinggal mendefinisikan method seperti halnya fungsi biasa. Misalnya kita akan menambahkan fungsi atau method untuk mengganti warna mobil dan untuk menampilkan warna mobil. Nama method pada dasarnya terseher (mengikuti aturan nama variabel). Namun demikian, jangan menggunakan nama method yang diawali dengan dua buah *under-score* (__) yaitu **__construct()**, **__destruct()** dan **__clone()** karena ketiga fungsi tersebut mempunyai arti tersendiri di PHP.

Lihat contoh menambahkan method sbb :

```

1 class Mobil
2 {
3     var $warna = "Biru";
4     var $merk = "BMW";
5     var $harga = "10000000";
6
7     function gantiWarna ($warnaBaru)
8     {
9         $this->warna = $warnaBaru;
10    }
11
12    function tampilWarna ()
13    {
14        echo "Warna mobilnya : " . $this->warna;
15    }
16 }

```

Pada listing program di atas sudah ditambahkan fungsi atau method **gantiWarna()** dan **tampilWarna()**. Untuk mengakses *properties* (variabel) bisa menggunakan keyword **\$this**. Keyword ini me-refer pada *class* tempatnya berada. Method gantiWarna() mempunyai satu parameter, yaitu \$warnaBaru.

Menambahkan Konstruktur

Konstruktur merupakan sebuah *method* khusus yang akan secara otomatis dijalankan saat *object* terbentuk. Konstruktur tidak harus ada, namun dalam satu *class* hanya boleh ada satu konstruktur. Method konstruktur biasanya berisi pemberian nilai *default* dari masing-masing *properties* (variabel).

Untuk membuat konstruktur, cukup dengan mendefinisikan suatu fungsi dengan nama **__construct()**. Perhatikan contoh sebagai berikut :

```

1 class Mobil
2 {
3     var $warna;
4     var $merk;
5     var $harga;
6
7     function __construct()
8     {
9         $this->warna = "Biru";
10        $this->merk = "BMW";
11        $this->harga = "10000000";
12    }
13
14    function gantiWarna ($warnaBaru)
15    {
16        $this->warna = $warnaBaru;
17    }
18
19    function tampilWarna ()
20    {
21        echo "Warna mobilnya : " . $this->warna;

```

```

22     }
23 }

```

Membentuk Objek *Class*

Untuk menggunakan sebuah objek, harus dibentuk objek dari classnya. Dari sebuah class bisa dibentuk beberapa objek sekaligus. Bentuk umum pembentukan objek adalah sbb :

```
$namaObjek = new namaClass();
```

Sedangkan untuk memanggil anggota (member) dari class dapat dengan format sbb :

```

$namaObjek->variabel;
$namaObjek->namaMethod();

```

Program 10.1

Nama File : pbo01.php

Deskripsi : Program sederhana pendefinisian class dan pemanggilan class.

```

1  <?
2  class Mobil
3  {
4      var $warna;
5      var $merk;
6      var $harga;
7
8      function __construct()
9      {
10         $this->warna = "Biru";
11         $this->merk = "BMW";
12         $this->harga = "10000000";
13     }
14     function gantiWarna ($warnaBaru)
15     {
16         $this->warna = $warnaBaru;
17     }
18
19     function tampilWarna ()
20     {
21         echo "Warna mobilnya : " . $this->warna;
22     }
23 }
24
25 $a = new Mobil();
26 $b = new Mobil();
27 echo "<b>Mobil pertama</b><br>";
28 $a->tampilWarna();
29 echo "<br>Mobil pertama ganti warna<br>";
30 $a->gantiWarna("Merah");
31 $a->tampilWarna();
32 //
33 echo "<br><b>Mobil kedua</b><br>";
34 $b->gantiWarna("Hijau");
35 $b->tampilWarna();
36 ?>

```

Program 10.2

Nama File : pbo02.inc.php

Deskripsi : Program class untuk membuat sebuah form inputan sederhana.

```

1  <?php
2  /*
3  Nama Class : Form

```

```

4 Deskripsi : Class untuk membuat form inputan text sederhana
5 */
6
7 class Form
8 {
9     var $fields = array();
10    var $action;
11    var $submit = "Submit Form";
12    var $jumField = 0;
13
14    function __construct($action, $submit)
15    {
16        $this->action = $action;
17        $this->submit = $submit;
18    }
19    function displayForm()
20    {
21        echo "<form action='\".$this->action.\"' method='POST'>";
22        echo "<table width='100%'>";
23        for ($j=0; $j<count($this->fields); $j++) {
24            echo "<tr><td align='right'>\".$this->
fields[$j]['label'].\"</td>";
25            echo "<td><input type='text' name='\".$this->
fields[$j]['name'].\"></td></tr>";
26        }
27        echo "<tr><td colspan='2'>";
28        echo "<input type='submit' value='\".$this->submit.
\"></td></tr>";
29        echo "</table>";
30    }
31    function addField($name, $label)
32    {
33        $this->fields [$this->jumField]['name'] = $name;
34        $this->fields [$this->jumField]['label'] = $label;
35        $this->jumField ++;
36    }
37 }
38 ?>

```

Program 10.3

Nama File : pbo03.php

Deskripsi : Program memanfaatkan Program 10.2 untuk membuat form inputan sederhana.

```

1 <?php
2 include "pbo02.inc.php";
3 echo "<html><head><title>Mahasiswa</title></head><body>";
4 $form = new Form ("","Input Form");
5 $form->addField ("txtnim", "Nim");
6 $form->addField ("txtnama", "Nama");
7 $form->addField ("txtalamat", "Alamat");
8 echo "<h3>Silahkan isi form berikut ini :</h3>";
9 $form->displayForm();
10 echo "</body></html>";
11 ?>

```

Penjelasan Program 10.3

Program 10.3 di atas merupakan contoh pemanggilan class yang didefinisikan pada Program 10.2 (pbo02.inc.php). Untuk itu pertama kali program **pbo02.inc.php** harus di-include-kan dengan menggunakan fungsi include(). Perhatikan program baris ke-2. Untuk menambahkan form inputan text box, hanya perlu dengan memanggil fungsi **addField()**. Hasil tampilan dari program di atas dapat dilihat pada gambar 10.1



Gambar 10.1. Tampilan Program 10.3

Memanfaatkan Class yang Tersedia

Dengan adanya konsep pemrograman berorientasi objek, pada programmer PHP di seluruh dunia dapat lebih mudah berbagi teknik programming. Kita bisa membuat suatu class dan programmer lain dapat dengan mudah menggunakannya tanpa perlu mengetahui proses jalannya class tersebut. Just use it! Di internet saat ini bertebaran begitu banyak class dengan fungsinya masing-masing yang tersedia secara cuma-cuma dan dapat dimodifikasi secara bebas. Situs yang berisi kumpulan-kumpulan class PHP yang cukup terkenal antara lain <http://www.phpclasses.org>. Silahkan berkunjung dan ambil class yang Anda inginkan

Berikut ini beberapa aplikasi PHP yang dibuat dengan konsep OOP yang banyak digunakan dalam aplikasi berbasis web :

1. PEAR

PEAR merupakan singkatan dari PHP Extension and Application Repository. PEAR berisi kumpulan class PHP yang dapat digunakan sewaktu-waktu. Class yang cukup banyak digunakan dari PEAR adalah **PEAR::DB** yang merupakan *abstraction layer* antara aplikasi PHP dengan database. Dengan PEAR::DB program menjadi lebih fleksibel dan secara mudah dapat berganti jenis database apapun (MySQL, PostgreSQL, Oracle dsb). PEAR dapat di-download dan dilihat di situs <http://pear.php.net>

2. Smarty

3. fpdf

4. PHPGraph

5. dll (belum selesai)