

PRAKTIKUM IX

OBJECT-ORIENTED PHP

9.1. CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menjelaskan konsep Object Oriented pada PHP, membuat aplikasi PHP dengan teknik Object Oriented dan membuat class PHP untuk melakukan manipulasi basis data.

9.2. PERANGKAT KERAS YANG DIGUNAKAN

Komputer/laptop, proyektor.

9.3. PERANGKAT LUNAK YANG DIGUNAKAN

Text Editor dan peramban Internet.

9.4. MATERI PRAKTIKUM

9.4.1. Pengantar Pemrograman Berorientasi Objek dengan PHP

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

9.4.2. 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 bagianbagian 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.

9.4.3. 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 (_).

9.4.4. Mendefinisikan Class

Bentuk umum mendefinisikan sebuah class adalah sebagai berikut:

```
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):

```
class Mobil
{
    var $warna;
    var $merk;
    var $harga;
    //Tambahkan pendefinisian method di sini
}
```

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 :

```
class Mobil
{
    var $warna = "Biru";
    var $merk = "BMW";
    var $harga = "10000000";
    //Tambahkan pendefinisian method di sini
}
```

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 terserah (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.

Contoh sebagai berikut:

```
class Mobil
{
    var $warna = "Biru";
    var $merk = "BMW";
    var $harga = "10000000";

    function gantiWarna ($warnaBaru)
    {
        $this->warna = $warnaBaru;
    }

    function tampilWarna ()
    {
        echo "Warna mobilnya : " . $this->warna;
    }
}
```

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**.

Menambah 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 :

```
class Mobil {
    var $warna;
    var $merk;
    var $harga;
    function __construct(){
        $this->warna = "Biru";
        $this->merk = "BMW";
        $this->harga = "10000000";
    }
    function gantiWarna ($warnaBaru){
        $this->warna = $warnaBaru;
    }
    function tampilWarna (){
        echo "Warna mobilnya : " . $this->warna;
    } }
}
```

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();
```

9.5. PRAKTIKUM

Nama file : pbo01.php

Deskripsi : Program sederhana pendefinisian class dan pemanggilan class.

```
<?  
class Mobil  
{  
    var $warna;  
    var $merk;  
    var $harga;  
    function __construct(){  
        $this->warna = "Biru";  
        $this->merk = "BMW";  
        $this->harga = "10000000";  
    }  
    function gantiWarna ($warnaBaru){  
        $this->warna = $warnaBaru;  
    }  
    function tampilWarna (){  
        echo "Warna mobilnya : " . $this->warna;  
    }  
}  
$a = new Mobil();  
$b = new Mobil();  
echo "<b>Mobil pertama</b><br>";  
$a->tampilWarna();  
echo "<br>Mobil pertama ganti warna<br>";  
$a->gantiWarna("Merah");  
$a->tampilWarna();  
//  
echo "<br><b>Mobil kedua</b><br>";  
$b->gantiWarna("Hijau");  
$b->tampilWarna(); ?>
```

Deskripsi : Program class untuk membuat sebuah form inputan sederhana.

```
<?php
class Form
{
    var $fields = array();
    var $action;
    var $submit = "Submit Form";
    var $jumField = 0;
    function __construct($action, $submit){
        $this->action = $action;
        $this->submit = $submit;
    }
    function displayForm(){
        echo "<form action='". $this->action.'" method='POST'>";
        echo "<table width='100%'>";
        for ($j=0; $j<count($this->fields); $j++) {
            echo "<tr><td align='right'>". $this->fields[$j]['label']. "</td>";
            echo "<td><input type='text' name='". $this->fields[$j]['name']. "'></td></tr>";
        }
        echo "<tr><td colspan='2'>";
        echo "<input type='submit' value='". $this->submit. "'></td></tr>";
        echo "</table>";
    }
    function addField($name, $label){
        $this->fields [$this->jumField]['name'] = $name;
        $this->fields [$this->jumField]['label'] = $label;
        $this->jumField ++;
    }
}
?>
```

Nama file : pbo3.php

Deskripsi : Program memanfaatkan pbo02.php untuk membuat form inputan sederhana.

```
<?php
    include "pbo02.inc.php";
    echo "<html><head><title>Mahasiswa</title></head><body>";

    $form = new Form ("","Input Form");
    $form->addField ("txtnim", "Nim");
    $form->addField ("txtnama", "Nama");
    $form->addField ("txtalamat", "Alamat");

    echo "<h3>Silahkan isi form berikut ini :</h3>";
    $form->displayForm();
    echo "</body></html>";
?>
```

9.6. LATIHAN

Buatkan CRUD dengan PDO seperti pada Praktikum VII dengan konsep PBO.