PHP : CLASSES AND OBJECTS

KL PEMROGRAMAN WEB LANJUT 2017

Dewi Anisa Istiqomah
dewianisaist.ugm@gmail.com

Apa itu Pemrograman Berbasis Objek?

- Salah satu cara membuat program (programming paradigm) dengan memecah alur program menjadi modul-modul sederhana yang disebut dengan objek.

- Setiap objek akan memiliki fungsi dan tugas tersendiri.

Fungsi Pemrograman Berbasis Objek dalam PHP

- Keuntungan pemrograman berbasis objek baru terasa ketika program tersebut telah ‘besar’ atau kita bekerja dengan tim untuk membagi tugas. Konsep ‘objek’ untuk memisahkan program menjadi bagian-bagian yang berdiri sendiri akan memudahkan dalam membuat program.
- Aplikasi framework PHP seperti ***Code Igniter***, ***Yii Framework***, ***Symfony*** dan ***Laravel***, semuanya menggunakan OOP.

Apa itu Class?

- Merupakan ‘blueprint’ dari object.
- Class digunakan hanya untuk membuat kerangka dasar.
- Yang akan kita pakai nantinya adalah hasil cetakan dari class, yakni object.

Contoh Penulisan Class

```php
<?php
class smartphone {
   // isi dari class smartphone...
}
?>
```

- Penulisan class diawali dengan keyword class
- Penulisan nama class diawali dengan huruf atau underscore.
- Isi dari class berada dalam tanda kurung kurawal.

Apa itu Property?

- Disebut juga atribut.
- Merupakan data yang terdapat dalam sebuah class.
- Aturan tata cara penamaan property sama dengan aturan penamaan variabel.
- Sebuah class tidak harus memiliki property

Contoh Penulisan Property

```php
<?php

class smartphone {

    var $pemilik;

    var $merk;

    var $ukuran_layar;

    // lanjutan isi dari class smartphone...

}

?>
```

Apa itu Method?

- Merupakan tindakan yang bisa dilakukan didalam class.
- Method pada dasarnya adalah function yang berada di dalam class.
- Aturan penulisan method di dalam PHP sama dengan cara penulisan function.
- Sebuah class tidak harus memiliki method.

Contoh Penulisan Method

```
<?php
class smartphone {
   function hidupkan_smartphone() {
   //... isi dari method hidupkan_smartphone
   }

   function matikan_smartphone() {
   //... isi dari method matikan_smartphone
   }

   //... isi dari class smartphone
}
?>
```

Apa itu Object?

- Merupakan hasil cetak dari class, atau hasil ‘konkrit’ dari class.
- Objek dari suatu class akan memiliki seluruh ciri-ciri class tersebut, yaitu property dan method-nya.
- Proses ‘mencetak’ objek dari class ini disebut dengan ‘instansiasi’.
- Proses instansiasi dilakukan dengan menggunakan keyword ‘new’.

Contoh Penulisan Object

```php
<?php

class smartphone {

    //... isi dari class laptop

}



$smartphone_ana = new smartphone();

?>
```

Cara Membuat Objek dalam PHP

- /* Code 1 */

```
<?php
// buat class smartphone
class smartphone {

   // buat property untuk class smartphone
   var $pemilik;
   var $merk;
   var $ukuran_layar;

   // buat method untuk class smartphone
   function hidupkan_smartphone() {
     return "Hidupkan smartphone";
   }
   function matikan_smartphone() {
     return "Matikan smartphone";
   }
}

// buat objek dari class smartphone (instansiasi)
$smartphone_ana = new smartphone();
?>
```

Cara Mengakses Objek dalam PHP

- Tambahkan kode di bawah ini ke code1.php

```
// set property
$smartphone_ana->pemilik="Ana";
$smartphone_ana->merk="Asus Zenfone 2";
$smartphone_ana->ukuran_layar="5 inchi";

// tampilkan property
echo $smartphone_ana->pemilik;
echo "<br />";
echo $smartphone_ana->merk;
echo "<br />";
echo $smartphone_ana->ukuran_layar;
echo "<br />";

// tampilkan method
echo $smartphone_ana->hidupkan_smartphone();
echo "<br />";
echo $smartphone_ana->matikan_smartphone();
```

Perhatikan!

- Menggunakan tanda panah (->) untuk mengakses property dan method dari objek.
- Penulisan nama property dan method dilakukan tanpa menggunakan tanda $.
- Tanda panah ini adalah operator khusus objek yang dikenal dengan istilah ‘Object Operator’.
- Sebuah class bisa digunakan untuk membuat berapapun banyak objek.

Enkapsulasi Objek

- Merupakan sebuah metode untuk mengatur struktur class dengan cara menyembunyikan alur kerja dari class tersebut.
- Struktur class yang dimaksud adalah property dan method.
- Dengan enkapsulasi, dapat dilakukan pembatasan akses kepada property dan method
- Proses enkapsulasi diterapkan dengan menggunakan 3 jenis hak akses: Public, Protected dan Private.

Hak Akses: Public

- Ketika sebuah property atau method dinyatakan sebagai public, maka seluruh kode program di luar class bisa mengaksesnya, termasuk class turunan.
- Jika hak akses property dan method tidak ditulis, maka PHP menganggapnya sebagai public.

- ***Tambahkan kata public sebelum nama property dan nama method di class smartphone! Amati!***

Hak Akses: Protected

- Jika sebuah property atau method dinyatakan sebagai protected, berarti property atau method tersebut tidak bisa diakses dari luar class, namun bisa diakses oleh class itu sendiri atau turunan class tersebut.

- ***Ubah kata public menjadi protected sebelum nama property dan nama method di class smartphone! Amati!***

Akses Protected, diakses dari dalam class itu sendiri

```php
<?php
// buat class smartphone
class smartphone {
   // buat protected property untuk class smartphone
   protected $pemilik = "Ana";
   public function akses_pemilik() {
      return $this->pemilik;
   }
   protected function hidupkan_smartphone() {
     return "Hidupkan smartphone";
   }
   public function paksa_hidup() {
      return $this->hidupkan_smartphone();
   }
}
// buat objek dari class smartphone (instansiasi)
$smartphone_ana = new smartphone();
// tampilkan method akses_pemilik()
echo $smartphone_ana->akses_pemilik();
echo "<br>";
// tampilkan method paksa_hidup()
echo $smartphone_ana->paksa_hidup();
?>
```

Akses Protected, diakses dari class turunan

```php
<?php
//buat class smartphone
class smartphone{
	//property dengan hak akses protected
	protected $pembuat = "China";
}
//buat class tablet
class tablet extends smartphone{
	public function tampilkan_pembuat(){
		return $this->pembuat;
	}
}
//buat objek dari class tablet
$tablet_baru = new tablet();
//jalankan method
echo $tablet_baru->tampilkan_pembuat();
?>
```

Hak Akses: Private

- Jika sebuah property atau method di-set sebagai private, maka satu-satunya yang bisa mengakses adalah class itu sendiri.

- Class lain tidak bisa mengaksesnya, termasuk class turunan.

Coba dan amati!

```php
<?php
//buat class smartphone
class smartphone{
	//property dengan hak akses protected
	private $pembuat = "China";
	public function tampilkan_pembuat(){
		return $this->pembuat;
	}
}
//buat class tablet
class tablet extends smartphone{
	public function tampilkan_pembuat(){
		return $this->pembuat;
	}
}
//buat objek
$smartphone_baru = new smartphone();
$tablet_baru = new tablet();
//jalankan method
echo $smartphone_baru->tampilkan_pembuat();
echo $tablet_baru->tampilkan_pembuat();
?>
```

Variabel $this dalam OOP PHP

- Merupakan sebuah variabel khusus dalam OOP PHP yang digunakan sebagai penunjuk kepada objek, ketika kita mengaksesnya dari dalam class.

Coba dan amati!

```php
<?php
// buat class
class smartphone {
   // buat property
   public $pemilik="Ana";
   // buat method
   public function hidupkan_smartphone() {
     return "Hidupkan smartphone $this->pemilik";
   }
}

// buat objek (instansiasi)
$smartphone_baru = new smartphone();
echo $smartphone_baru->hidupkan_smartphone();
echo "<br>";
//ubah isi property $pemilik pada objek $smartphone_baru
$smartphone_baru->pemilik="Bela";
echo $smartphone_baru->hidupkan_smartphone();
?>
```

Apa yang terjadi?

- Jika kode sebelumnya ditambahkan kode berikut:

```
// buat objek baru

$smartphone_lama = new smartphone();
echo $smartphone_lama->hidupkan_smartphone();
```

Cara Membuat Method dengan Argumen/Parameter

```
<?php
// buat class
class smartphone {
   // buat method
   public function hidupkan_smartphone($pemilik,$merk) {
     return "Hidupkan smartphone $merk punya $pemilik";
   }
}
// buat objek (instansiasi)
$smartphone_obj= new smartphone();
echo $smartphone_obj->hidupkan_smartphone("Ana", "Lenovo");
?>
```

Cara Membuat Method dengan Default Parameter

```php
<?php
// buat class
class smartphone {
   // buat method
   public function hidupkan_smartphone($pemilik = "Ciko",$merk = "Samsung") {
     return "Hidupkan smartphone $merk punya $pemilik";
   }
}
// buat objek (instansiasi)
$smartphone_obj= new smartphone();
echo $smartphone_obj->hidupkan_smartphone();
echo "<br>";
echo $smartphone_obj->hidupkan_smartphone("Ana", "Lenovo");
?>
```

Constructor

- Merupakan method khusus yang akan dijalankan secara otomatis pada saat sebuah objek dibuat (instansiasi), yakni ketika perintah “new” dijalankan.
- Constructor biasa digunakan untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property.
- Tidak wajib membuat constructor
- Constructor dibuat menggunakan method : __construct().

Destructor

- Merupakan method khusus yang dijalankan secara otomatis pada saat sebuah objek dihapus.
- Di dalam PHP, seluruh objek secara otomatis dihapus ketika halaman PHP dimana objek itu berada selesai diproses.
- Tidak wajib membuat method destructor.
- Buat method destructor jika ada script khusus yang ingin dieksekusi sebelum objek dimusnahkan.
- Destructor dibuat menggunakan method : __destruct().

```php
<?php
class smartphone {
   private $pemilik = "Ana";
   private $merk = "Lenovo";
   public function __construct(){
     echo "Ini berasal dari Constructor smartphone";
   }
   public function hidupkan_smartphone(){
     return "Hidupkan smartphone $this->merk punya $this->pemilik";
   }
   public function __destruct(){
     echo "Ini berasal dari Destructor smartphone";
   }
}
$smartphone_obj= new smartphone();



echo "<br />";
echo $smartphone_obj->hidupkan_smartphone();
echo "<br />";

// hapus objek $smartphone_obj
unset($smartphone_obj);

echo "<br />";
echo "Objek Telah Dihancurkan";
?>
```

Inheritance (Pewarisan)

- Merupakan konsep pemrograman dimana sebuah class dapat ‘menurunkan’ property dan method yang dimilikinya kepada class lain.

- Class yang akan ‘diturunkan’ bisa disebut sebagai class induk (parent class), super class, atau base class.

- Class yang ‘menerima penurunan’ bisa disebut sebagai class anak (child class), sub class, derived class atau heir class.

Cara Penggunaan Inheritance dalam PHP

- Dengan kata kunci: ‘extends’.

```
class induk {
   //...isi class induk
}

class anak extends induk
{
   //... class anak bisa mengakses
   //... property dan method class induk
}
```

```php
<?php
// buat class induk
class smartphone {
   protected function beli_smartphone() {
     return "Beli smartphone baru";
   }
}
// turunkan class smartphone
class tablet extends smartphone {
   protected function beli_tablet() {
     return "Beli tablet baru";
   }
}
// turunkan class tablet
class smartphone_dualsim extends tablet {
   protected function beli_smartphone_dualsim() {
     return "Beli smartphone dual sim baru";
   }
   public function beli_semua(){
     $a = $this->beli_smartphone();
     $b = $this->beli_tablet();
     $c = $this->beli_smartphone_dualsim();
     return "$a <br /> $b <br /> $c";
   }
}
// buat objek (instansiasi)
$gadget_baru = new smartphone_dualsim();
//panggil method objek
echo $gadget_baru->beli_semua();
//echo $gadget_baru->beli_smartphone();
?>
```

Scope Resolution Operator PHP

- Merupakan operator khusus di dalam PHP yang memungkinkan kita untuk mengakses ‘informasi khusus‘ dari dalam class.
- Informasi khusus ini terdiri dari: overridden property atau overridden method, static property atau static method, serta constanta class.
- Scope Resolution Operator ditulis dengan tanda dua kali titik dua (double colon), yakni “::”

Mengakses Property dan Method Parent Class

- Overridden property dan overridden method merupakan property atau method dari class anak memiliki nama yang sama dengan class induk.

- Untuk mengakses property dan method dari class induk, kita mengaksesnya dengan perintah:

```
parent::nama_property;

parent::nama_method();
```

Mengakses Constructor dan Destructor Parent Class

- Constructor dan destructor parent class akan dijalankan jika child class tidak mendefenisikan constructor dan destructor sendiri.

- Namun jika child class juga memiliki constructor dan desctructor, maka harus memanggil constructor dan destructor parent class secara manual.

Static Property dan Static Method

- Merupakan property (variabel) dan method (function) yang melekat kepada class, bukan kepada objek.

- Karena static property dan static method adalah milik class, maka kita tidak perlu membuat objek untuk mengaksesnya, tapi langsung menyebutkan nama class dan menggunakan operator ‘::’.

Mengakses Static Property dan Static Method Dari Class Itu Sendiri

- Jika kita menggunakan variabel $this untuk mengakses property dan method ‘normal’ dari dalam class, maka untuk mengakses static property dan static method, kita menggunakan keyword “self::”.

Mengakses Static Property dan Static Method Parent Class

- Untuk class dengan penurunan (inheritance), kita bisa menggunakan keyword ***parent::nama_property*** dan ***parent::nama_method*** untuk mengakses static property dan static method dari parent class.

Konstanta Class dalam Pemrograman Objek

- Merupakan konstanta yang berada di dalam class.
- Selain memiliki property dan method, PHP juga membolehkan menggunakan konstanta (constant) di dalam class.
- Contoh kode program pembuatan constanta:

```
class nama_class {
    const NAMA_KONSTANTA = nilai_konstanta;
}
```

Mengakses Konstanta Class

- Class constant seolah-olah berprilaku sebagai static property.
- Class constant juga terikat kepada class, bukan objek.
- Untuk mengakses nilai konstanta, menggunakan operator yang sama seperti static property, yakni menggunakan double colon ‘::’.

Mengakses Konstanta Class dari dalam Class itu Sendiri

- Untuk mengakses class constant dari dalam class itu sendiri, PHP menggunakan cara yang sama dengan static property, yaitu dengan perintah self::nama_konstanta.

Mengakses Konstanta Class milik Parent Class

- PHP menggunakan operator parent::nama_konstanta untuk mengakses konstanta milik parent class.

Final Method dan Final Class Pemrograman Objek

- Dengan menambahkan keyword final kepada sebuah method, maka method tersebut tidak dapat didefenisikan ulang di dalam child class.
- Dan jika sebuah class ditambahkan keyword final, maka class tersebut tidak bisa diturunkan sama sekali.
- Dalam PHP, tidak dikenal istilah final property, sehingga tidak ada mekanisme untuk membatasi class anak untuk menimpa nilai property dari class induk.

Cara Pembuatan Final Method dan Final Class

```
final public function nama_method(){

    //... isi method

}


final class nama_class {

    //... isi class

}
```

Abstract Class dan Abstract Method PHP

- Abstract Class merupakan sebuah class yang tidak bisa di-instansiasi (tidak bisa dibuat menjadi objek) dan berperan sebagai ‘kerangka dasar’ bagi class turunannya. Di dalam abstract class umumnya akan memiliki abstract method.
- Abstract Method adalah sebuah ‘method dasar’ yang harus diimplementasikan ulang di dalam class anak (child class). Abstract method ditulis tanpa isi dari method, melainkan hanya ‘signature’-nya saja. Signature dari sebuah method adalah bagian method yang terdiri dari nama method dan parameternya (jika ada).

Object Interface Dalam Pemrograman Berbasis Objek

- Object Interface adalah sebuah ‘kontrak’ atau perjanjian implementasi method.
- Interface lebih berperan untuk menyeragamkan method.

Polimorfisme dalam Pemrograman Objek PHP

- Merupakan konsep dimana terdapat banyak class yang memiliki signature method yang sama.
- Implementasi dari method-method tersebut diserahkan kepada tiap class, akan tetapi cara pemanggilan method harus sama.
- Agar dapat ‘memaksakan’ signature method yang sama pada banyak class, class tersebut harus diturunkan dari sebuah abstract class atau object interface.

Tujuan Implementasi Polimorfisme

- Yaitu untuk membuat struktur pola dari class dan turunannya.

- Polimorfisme menekankan alur kode program yang terorganisir untuk mengurangi adanya perulangan kode program

TERIMAKASIH