

OOP PHP

BAB 1 PEMROGRAMAN WEB LANJUT

Ide / Pola Pikir Pemrograman

- ▶ Pemrograman Prosedural
- ▶ Pemrograman Berorientasi Objek (PBO)

Kelebihan Prosedural

- ▶ Memiliki algoritma pemecahan masalah yang sederhana, standar dan efektif.
- ▶ Penulisan program memiliki struktur logika yang benar dan mudah dipahami
- ▶ Program hanya terdiri dari 3(tiga) struktur dasar, yaitu struktur berurutan, struktur seleksi dan struktur perulangan.

Kekurangan Prosedural

- ▶ Program cukup sulit untuk proses perawatan
- ▶ Fungsi yang tersedia, susah untuk diubah tanpa harus mempengaruhi fungsi sistem secara keseluruhan.
- ▶ Butuh usaha yang keras untuk menterjemahkan Business Models dalam programming models.
- ▶ Mungkin dapat bekerja dengan baik pada saat terisolasi tapi tidak pada saat terintegrasi dengan sistem lain.

Kelebihan OOP

- ▶ Dengan OOP, kode-kode yang kita buat menjadi lebih rapih dan terstruktur.
- ▶ Dengan OOP, proses reuse kode-kode yang kita buat untuk project yang hampir sama, mudah, karena kode yang kita buat rapih dan terstruktur. Sehingga untuk mengubah atau reuse kode jauh lebih mudah.

Kelebihan OOP lanjutan..

- ▶ Konsepnya per modul. Fungsi atau prosedur misal di java atau di PHP tinggal panggil saja nama fungsinya. Jika bermasalah nantinya akan mudah diketahui karena terbaca dari fungsi yang kita panggil. Itulah yang dimaksud konsep per modul. Sehingga dengan OOP, kita dimudahkan untuk membuat dan membaca kode kita (efisiensi kode).
- ▶ Konsep OOP juga memudahkan kita untuk menganalisa program yang kita akan buat. Ini akan sangat terasa kalau kita membuat program besar dan rumit.

Kekurangan OOP

- ▶ Kesulitan dalam ide programnya yang dapat digunakan dalam program
- ▶ Membangun awal perlu ketrampilan programming lebih

Pengertian PBO/OOP?

- ▶ OOP adalah cara pandang atau pola pikir dalam menganalisa suatu sistem atau permasalahan pemrograman, dalam OOP semua bagian program dianggap sebagai object

Konsep Dasar OOP

Beberapa konsep dasar OOP

1. Class dan Object
2. Encapsulation (method dan atribut dalam class)
3. Inheritance (Penurunan sifat)
4. Polimorfisme

Class dan Object

- ▶ Mobil, hewan, orang adalah Class dan nantinya bisa diciptakan objectnya, bukan hanya fisik saja bisa juga berupa abstrak seperti account bank, file di komputer, pesan email, acara tv dll
- ▶ Object mempunyai properties (atribut) dan melakukan sesuatu (behavior / method)

Class

Class adalah penjelasan dari object seperti untuk object mobil paling tidak mempunyai penjelasan properties tentang warna, tahun rakit dan penjelasan melakukan sesuatu seperti cara mengerem, cara parkir dll.

Contoh class : mobil

Contoh object : mobil pak udin, mobil di depan rumah dll

Penjelasan properties disebut atribut

Penjelasan melakukan sesuatu disebut method

Php dan OOP

- ▶ Awalnya php hanyalah kumpulan script sederhana
- ▶ Dalam perkembangannya ditambahkan fitur-fitur pemrograman berorientasi object pada PHP 4
- ▶ Pada PHP 5 fitur-fitur PBO semakin mantap dan cepat

Mendefinisikan class di php

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

Pendefinisian variabel dalam class

```
class Mobil
{
    var $warna;
    var $merk;
    var $harga;

    //Tambahkan pendefinisian method di sini
}
```

- Pendefinisian variabel bukanlah suatu keharusan karena variabel di php tidak perlu di deklarasi terlebih dahulu tapi sangat dianjurkan untuk kemudahan dalam pembacaan dan pemahaman program

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

- Dapat juga langsung di beri nilai dalam pendeklarasian tetapi tidak boleh menggunakan operator seperti aritmatika dsb

Pendefinisian method dalam class

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

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

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


enkapsulasi

Modifier	Keterangan
Public	Untuk mendefinisikan data atau metode yang akan terlihat dari luar oleh siapapun dan dimanapun.
Private	Untuk mendefinisikan data atau metode agar hanya terlihat pada class/object itu sendiri.
Protected	Untuk mendefinisikan data atau metode untuk tidak terlihat dari luar (seperti private), tetapi akan dapat diakses oleh "anak" dari class tersebut.

Contoh encapsulasi

```
<?php
class Person
{
    private $_name;
    private $_age;

    function __construct($name, $age = 0)
    {
        if (!is_int($age))
        {
            throw new Exception("Cannot assign non integer value to
integer field, 'Age'");
        }

        $this->_age = $age;
        $this->_name = $name;
    }
}
```

```
public function setAge($age)
{
    if (!is_int($age))
    {
        throw new Exception("Cannot assign non integer value to
integer field, 'Age'");
    }

    $this->_age = $age;
}

public function yearsToRetire()
{
    return 67 - $this->_age;
}

}

$person = new Person("Wes");
$person->setAge(31);

echo $person->yearsToRetire();

?>
```

```
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;
    }
}
```

- Konstruktor adalah method khusus yang dijalankan secara otomatis ketika object terbentuk. Tidak harus ada dan jika ada hanya satu saja kelas konstruktor

Membentuk object dan Memanggil anggota class

- Membentuk object

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

- Memanggil anggota class

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

Contoh penggunaan

```
$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();  
~
```

Pewarisan class

```
<?php
    //Class hewan
class hewan
{
    protected $jml_kaki;
    protected $warna_kulit;

    function __construct()
    {
    }

    function berpindah()
    {
        echo "Saya berpindah";
    }

    function makan()
    {
        echo "Saya makan";
    }
}
```

```
//Class kucing
class kucing extends hewan
{
    function berpindah()
    {
        echo "Saya merangkak dengan 4 kaki";
    }
}

//Class burung
class burung extends hewan
{
    protected $sayap;

    function berpindah()
    {
        echo "Saya terbang";
    }

    function makan()
    {
        echo "Saya makan dengan mematuk";
    }
}
```


Keyword final pada method

```
<?php
class BaseClass {
    public function test() {
        echo "BaseClass::test() called\n";
    }

    final public function moreTesting() {
        echo "BaseClass::moreTesting() called\n";
    }
}

class ChildClass extends BaseClass {
    public function moreTesting() {
        echo "ChildClass::moreTesting() called\n";
    }
}

// Results in Fatal error: Cannot override final method BaseClass::moreTesting()
?>
```

```
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 ++;
    }
}
```

Diakses oleh file lain...

```
<?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>";
?>
```



▶ next->coba oop php di praktikum...

Sumber

- ▶ Achmad Solichin, Diktat Kuliah pemrograman web 2