

## Modul 1: Pemrograman Berorientasi Objek pada PHP

Sesuai dengan judulnya, pada bagian ini akan dibahas tentang metode pemrograman berorientasi objek pada PHP. Pada pemrograman website berbasis MCV(model, controller, view), pemanfaatan elemen PBO seperti class, object dan method akan sering ditemukan. Setiap controller dan model pada codeigniter merupakan suatu class yang selalu berhubungan. Class controller selalu menggunakan object dari class model dan method pada controller dapat langsung diakses melalui browser.

Pembahasan tentang pemrograman berorientasi objek akan membicarakan tentang abstraksi, enkapsulasi, pewarisan sifat dan polimorfisme. Pada matakuliah sebelumnya telah dibahas mengenai 4 sifat pemrograman berorientasi objek yang disebutkan sebelumnya. Pada modul ini, hanya akan dibahas tentang bagian deklarasi class, atribut, konstruktor dan method serta contoh penggunaan object yang selalu digunakan pada framework Codeigniter.

### deklarasi class

Class merupakan rancangan suatu objek yang memuat *attribut* dan *method*. *Attribut* merupakan ciri dari suatu objek. Contoh untuk objek mahasiswa, *attribut* yang melekat dan menjadi ciri adalah ***nim*** dan ***nama***. *Method* adalah hal-hal yang dapat dikerjakan pada suatu objek. Contoh untuk objek mahasiswa, *method* yang dapat dikerjakan adalah menambahkan data mahasiswa dan melakukan transaksi KRS.

Contoh class mahasiswa beserta *attribut* dan *method*:

```
class mahasiswa{
    public $nim; //attribut
    public $nama;

    public function set_mahasiswa($nim,$nama){ //method
        $this->nim=$nim;
        $this->nama=$nama;
    }

    public function get_mahasiswa(){
        $mhs['nim']=$this->nim;
        $mhs['nama']=$this->nama;
        return $mhs;
    }
}
```

Contoh 1 : deklarasi class;

Selain *attribut* dan *method*, elemen *class* lain yang penting adalah konstruktor/*constructor*. *Constructor* merupakan method yang secara otomatis dijalankan ketika *object* dideklarasikan. Pada pemrograman web menggunakan codeigniter, konstruktor dimanfaatkan untuk deklarasi attribute yang berhubungan dengan class lainnya.

```
class mahasiswa{
    public $nim; //attribut
    public $nama;

    Public function __construct($nim= "", $nama = ""){
        $this->nim = $nim;
        $this->nama = $nama;
    }
    public function get_mahasiswa(){
        $mhs['nim']=$this->nim;
        $mhs['nama']=$this->nama;
        return $mhs;
    }
}
```

Contoh 2 : deklarasi class dengan constructor

## Object

*Object* merupakan representasi dari data atau dapat dikatakan sebagai komponen dari program yang memiliki beberapa hal yang dapat dikerjakan dan dapat berhubungan dengan objek-objek lainnya. Seperti dikatakan sebelumnya bahwa *class* merupakan rancangan dari *object*. Semisal jika kita akan mengolah data mahasiswa, kita perlu membuat rancangan dari *objek* mahasiswa yang mempunyai ciri memiliki nim dan nama. dengan demikian, *class* yang akan dirancang memiliki *attribut* nim dan nama.

Terdapat perbedaan bentuk objek untuk class yang menggunakan konstruktor dan tidak menggunakan konstruktor. Contoh berikut membantu memahami kedua bentuk dari class tersebut.

```
$objek = new mahasiswa;
$objek->set_mahasiswa("57201", "siapa");
Echo $objek->nim;
Echo $objek->nama;
```

Contoh 3 : deklarasi objek tanpa konstruktor

Contoh 3 merupakan contoh untuk pembentukan objek dari class yang ditunjukkan pada contoh 1, dimana class pada contoh 1 tidak menggunakan konstruktor.

Perhatikan tanda "*->*" yang digunakan untuk mengakses method dan attribut. Penggunaan tanda ini merupakan cara untuk mengakses properti dari class baik method maupun attribut.

```
$objek = new mahasiswa("57201", "siapa");  
Echo $objek->nim;  
Echo $objek->nama;
```

#### Contoh 4 : deklarasi objek dengan konstruktor

contoh 4 merupakan contoh untuk membentuk objek dari class pada contoh 2 yang menggunakan konstruktor. Perbedaan antara contoh 4 dengan contoh 3 akan mudah ditemukan dan dipahami jika dipraktikkan.

## Latihan

1. Buatlah file html menggunakan code seperti berikut dengan file “index.html”

```
<form action = “hitung.php” method = “post”>
Paramater 1: <input type = “number” name = “var1”></br>
Paramater 2: <input type = “number” name = “var2”></br>
<input type = “submit” value= “+” name = “tambah”>
<input type = “submit” value= “-” name = “kurang”>
<input type = “submit” value= “x” name = “kali”>
<input type = “submit” value= “:” name = “bagi”>
```

2. Buatlah file php pada direktori yang sama menggunakan code berikut dengan nama file “hitung.php”

```
<?php
class hitung{
    public function __construct($var1, $var2){
        $this->var1 = $var1;
        $this->var2 = $var2;
    }
    public function tambah(){
        return ($this->var1 + $this->var2);
    }
    public function kurang(){
        return ($this->var1 - $this->var2);
    }
    public function kali(){
        return ($this->var1 * $this->var2);
    }
    public function bagi(){
        return ($this->var1 / $this->var2);
    }
}
$hitung = new hitung($_POST['var1'], $_POST['var2']);
if($_POST['tambah']){
    echo $hitung->tambah();
}else if($_POST['kurang']){
    echo $hitung->kurang();
}else if($_POST['kali']){
    echo $hitung->kali();
}else if($_POST['bagi']){
    echo $hitung->bagi();
}
?>
```

**Hasil**

Screenshot code:

Screenshot hasil:

Deskripsi kendala yang ditemukan: