

2018

[OOP DALAM PHP]

PEMROGRAMAN WEB LANJUT S1 INFORMATIKA

AINUL YAQIN, M.KOM

1. Pengertian OOP
2. Class dan Object
3. Method, Property, Constanta
4. Keyword This dan Self

1. Object Oriented Programming

Pemrograman berbasis object (OOP)¹ adalah sebuah paradigma pemrograman yang berorientasikan kepada object. Semua data dan fungsi dalam paradigma ini dibungkus dengan class-class atau object-object.

Dalam pemrograman berbasis objek, kita diminta untuk memahami sebuah masalah dan memodelkan masalah tersebut menjadi sebuah class dan kemudian class akan diinstansiasi menjadi sebuah object pada saat runtime. Setiap class/object dalam pemrograman berbasis object dapat saling berinteraksi dan berkomunikasi satu sama lain untuk mendukung sebuah solusi dari suatu masalah.

➤ Kelebihan OOP dibandingkan dengan procedural antara lain

1. Lebih terstruktur dan mudah dibaca.
2. Class-Class dapat digunakan kembali pada project yang lain (reuse).
3. Pemetaan masalah jadi lebih mudah sehingga mudah untuk membuat solusinya.
4. Pembatasan akses terhadap suatu fungsi dapat meningkatkan keamanan code.
5. Interaksi antara code lebih terasa.

2. Class dan Object

Secara gampang, class adalah sebuah model/cetakan sedangkan object adalah realisasinya. Dalam OOP, Class memiliki property dan method. Property adalah sesuatu yang dimiliki oleh class, sedangkan method adalah apa-apa saja yang bisa dilakukan oleh class.

Bila diibaratkan dengan Mobil, maka property adalah roda, kursi, pintu, dan lain sebagainya. Sedangkan method adalah maju, mundur, berbelok, mengerem dan lain sebagainya.

```
<?php

//Class
class Mobil {
//Property
    private $jumlahRoda;

//Property
    private $jumlahKursi;

    public function setJumlahRoda($jumlahRoda){
        $this->jumlahRoda = $jumlahRoda;
    }
}
```

```
    }

    //Method
    public function setJumlahKursi($jumlahKursi)
    {
        $this->jumlahKursi = $jumlahKursi;
    }

    //Method
    public function cetak()
    {
        echo 'Mobil punya '.$this->jumlahRoda.' roda dan
        '.$this->jumlahKursi.' kursi.';
    }
}

$sedan = new Mobil(); //Object
$sedan->setJumlahRoda(4);
$sedan->setJumlahKursi(4);
$sedan->cetak();
echo PHP_EOL;
?>
```

Pada contoh diatas, class Mobil adalah sebuah prototype/model sedangkan \$sedan adalah realisasinya. Sementara \$jumlahRoda dan \$jumlahKursi adalah property. Sedangkan setJumlahRoda(\$jumlahRoda) sampai pada cetak() dinamakan method (akan dijelaskan secara khusus pada bab tersendiri).

3. Keyword this dan Self

Dalam pemrograman berbasis object, keyword \$this pasti ada, walaupun cara penulisan dan mungkin namanya berbeda. Keyword \$this dalam OOP adalah sebuah variabel yang merujuk pada object yang diinstansiasi. Maksudnya keyword \$this ini nantinya akan diganti dengan variabel apapun tergantung dari variabel object yang diinstansiasi. Perhatikan contoh dibawah ini.

```

<?php
class Mobil{
private $jumlahRoda;

public function setJumlahRoda($jumlahRoda)
{
$this->jumlahRoda = $jumlahRoda;
}
}

$mobil = new Mobil();
$mobil->setJumlahRoda(4);

?>

```

Pada contoh diatas, kita membuat object class Mobil dengan nama \$mobil. Maka saat itu \$this akan digantikan dengan variabel \$mobil. Dan bila kita membuat object lainnya misal \$mobil2 maka \$this akan digantikan dengan \$mobil2. Dapat disimpulkan bahwa keyword \$this, digunakan untuk merujuk pada object yang belum diketahui dan digunakan untuk mempermudah kita dalam menuliskan code. Perlu Anda ketahui bahwa antara \$mobil dan \$mobil2 itu adalah dua object yang berbeda sehingga ketika memanggil

\$mobil->setJumlahRoda(4) dan \$mobil2->setJumlahRoda(7)

maka nilai \$jumlahRoda pada \$mobil tidak akan ditimpa oleh nilai \$jumlahRoda pada \$mobil2. Untuk lebih jelas, perhatikan contoh dibawah ini:

```

<?php
class Mobil{
private $jumlahRoda;

public function setJumlahRoda($jumlahRoda)
{
$this->jumlahRoda = $jumlahRoda;
}

public function getJumlahRoda()
{
return $this->jumlahRoda;
}
}

```

```
}  
}  
$mobil = new Mobil();  
$mobil->setJumlahRoda(4);  
$mobil2 = new Mobil();  
$mobil2->setJumlahRoda(7);  
echo $mobil->getJumlahRoda();//Output: 4  
echo PHP_EOL;  
echo $mobil2->getJumlahRoda();//Output: 7  
  
?>
```

Keyword \$this hanya dapat digunakan pada internal class dan tidak dapat dipanggil dari luar class. Selain itu, keyword \$this juga tidak bisa ditimpa nilainya (read-only variable).

Tidak jauh berbeda dengan keyword \$this, keyword self pun memiliki karakteristik yang sama. Yang membedakan dengan keyword \$this adalah bahwa keyword self digunakan hanya untuk memanggil property atau method yang bersifat static. Contoh yang property yang bersifat static adalah constanta. Sehingga ketika kita memanggil constanta didalam class maka kita memanggilnya dengan self::NAMA_CONSTANTA. Pemahaman lebih dalam tentang sifat static pada class, akan dibahas pada bab tersendiri.

4. Instansiasi Object

Method adalah suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object. untuk membuat method bisa dengan dimulai dengan keyword function lalu di ikuti oleh nama function nya. berikut ini adalah contoh method Setter dan Getter untuk memberikan value kepada properti dan mengakses properti tersebut :

Class_lib.php

```
<?php
    class person {
        var $name;
        function set_name($new_name) {
            $this->name = $new_name;
        }

        function get_name() {
            return $this->name;
        }
    }
?>
```

```
<?php include("class_lib.php"); ?>
<?php
    $stefan = new person();
    $jimmy = new person;

    $stefan->set_name("Stefan Mischook");
    $jimmy->set_name("Nick Waddles");

    echo "Stefan's full name: " . $stefan->get_name();
    echo "Nick's full name: " . $jimmy->get_name();
?>
```