



Praktikum

2

Object Oriented Programming (OOP) Dasar Menggunakan PHP

A TUJUAN

1. Mahasiswa dapat memahami konsep object oriented programming (OOP)
2. Mahasiswa dapat memahami OOP menggunakan PHP
3. Mahasiswa dapat membuat program sederhana menggunakan OOP PHP

B PRETEST

1. Apakah yang dimaksud dengan Object Oriented Programming?
2. Apa yang disebut dengan class, method dan object?
3. Apa manfaat/kelebihan OOP?

C DASAR TEORI

Dua istilah yang paling penting untuk OOP adalah *Class* dan *Object*. *Class* adalah generalisasi definisi sesuatu. *Class* dapat dipandang sebagai cetak biru *Object* yaitu implementasi yang spesifik dari sesuatu hal. Salah satu prinsip OOP adalah modularitas yaitu memecah aplikasi menjadi submodul tertentu. Website banyak melakukan hal tersebut misalnya: berinteraksi dengan database, menangani formulir, mengirim email, menghasilkan HTML, dll. Masing-masing hal tersebut dapat dijadikan modul, yaitu *Class*. Kode dapat dikembangkan secara mandiri dengan memisahkan yang tidak terkait (walaupun berinteraksi) elemen. Disamping itu juga mempermudah membuat perawatan dan update kode, dan menyederhanakan debugging

Class

Secara sintak, definisi *Class* dimulai dengan kata **class**, yang diikuti dengan nama class tersebut. Nama class tidak dapat menggunakan *reserved word*. Selanjutnya nama class diikuti dengan kurung kurawal buka dan tutup.

Method dalam class diawali dengan kata **function**, yang diikuti dengan nama method serta argumen. Atribut dalam class didefinisikan sebagai variabel yang terdiri 3 jenis yaitu **public**, **private** dan **protected**. Luaran dari sebuah method dituliskan menggunakan kata **return**.

Contoh:

```
class matematika{  
    function tambah($a,$b){  
        $c = $a + $b;  
        return $c;  
    }  
}
```

Object

OOP digunakan melalui 2 tahap. Pertama, mendefinisikan class. Kedua, menggunakan class tersebut dengan membuat sebuah *object* (*class instance*). Untuk membuat objek digunakan kata **new**. Contoh **\$object = new ClassName();**

Untuk memanggil methods menggunakan sintak object operator -> contoh **\$object -> methodName();**

Untuk mengakses properti object digunakan sintaks **\$object -> propertyName;**



Contoh:

```
class matematika{
    function tambah($a,$b){
        $c = $a + $b;
        return $c;
    }
}

$math = new matematika();
$bil1 = 7;
$bil2 = 5;
$hasil = $math->tambah($bil1,$bil2);
echo $hasil;
```

This

Sebuah method dimungkinkan untuk mengakses properti atau method dalam class yang sama menggunakan kata **this**.

Contoh:

```
class matematika{
    public $pi = 3.14;
    function tambah($a,$b){
        $c = $a + $b;
        return $c;
    }
    function kuadrat($x){
        return $x * $x;
    }
    function keliling_lingkaran($r){
        $kel = 2 * $this->$pi * $r;
        return $kel;
    }
    function luas_lingkaran($r){
        $luas = $this->pi * $this->kuadrat($r);
        return $luas;
    }
}
```

Constructor

Constructor merupakan method yang dieksekusi secara otomatis pertamakali saat sebuah class diturunkan menjadi sebuah objek

Contoh:

```
class matematika{
    public $pi;

    function __construction(){
        $this->pi = 3.14;
    }

    function keliling_lingkaran($r){
        $kel = 2 * $this->$pi * $r;
        return $kel;
    }
}
```



D PERCOBAAN

Percobaan 1

```
<?php
class matematika{
    public $pi = 3.14;
    function tambah($a,$b){
        $c = $a + $b;
        return $c;
    }
    function kuadrat($x){
        return $x * $x;
    }
    function keliling_lingkaran($r){
        $kel = 2 * $this->$pi * $r;
        return $kel;
    }
    function luas_lingkaran($r){
        $luas = $this->pi * $this->kuadrat($r);
        return $luas;
    }
}

$math = new matematika();

$jari = 10;
$kel_lingkaran = $math->keliling_lingkaran($jari);
$luas_lingkaran = $math->luas_lingkaran($jari);

echo "Menghitung Keliling dan Luas Lingkaran<br>";
echo "Jari-Jari : ".$jari."<br>";
echo "Keliling = ".$kel_lingkaran."<br>";
echo "Luas = ".$luas_lingkaran;
?>
```

Percobaan 2

Buatlah file php dengan nama class_matematika.php

```
<?php
class matematika{
    public $pi = 3.14;
    function tambah($a,$b){
        $c = $a + $b;
        return $c;
    }
    function kuadrat($x){
        return $x * $x;
    }
    function keliling_lingkaran($r){
        $kel = 2 * $this->$pi * $r;
        return $kel;
    }
    function luas_lingkaran($r){
        $luas = $this->pi * $this->kuadrat($r);
        return $luas;
    }
}
?>
```



Buatlah file php dengan nama lingkaran.php.
Simpan dalam satu folder yang sama dengan file class_matematika.php

```
<?php
    include 'class_matematika.php';
    $math = new matematika();

    $jari = 10;
    $kel_lingkaran = $math->keliling_lingkaran($jari);
    $luas_lingkaran = $math->luas_lingkaran($jari);

    echo "Menghitung Keliling dan Luas Lingkaran<br>";
    echo "Jari-Jari : ".$jari."<br>";
    echo "Keliling = ".$kel_lingkaran."<br>";
    echo "Luas = ".$luas_lingkaran;
?>
```

Buka/jalankan di browser file keliling.php

E LATIHAN

Buatlah program OOP PHP untuk menghitung

1. Luas dan volume balok menggunakan form HTML (input: panjang, lebar tinggi)
2. Luas dan volume bola menggunakan form HTML (input: jari-jari)
3. Konversi nilai angka mata kuliah ke nilai huruf menggunakan form HTML (input: nilai angka)