

**PRAKTIKUM**  
**PEMROGRAMAN WEB**  
**SEMESTER GENAP TAHUN AKADEMIK 2021/2022**

Tanggal

21 Mei 2022

Materi

OBJECT ORIENTED PROGRAMMING (OOP)

**PRODI SISTEM INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS NUSANTARA PGRI KEDIRI**

## BAB I

### DASAR TEORI

#### Object oriented programming (Oop)

Dua istilah yang paling penting untuk oop adalah class dan object. Class adalah generalisasi dari sesuatu. class dapat di pandang sebagai cetak biru. Object yaitu implementasi yang spesifik dari suatu 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, dan

Masing-masing hal tersebut dapat di jadikan modul, yaitu class. Kode dapat di kembangkan secara mandiri dengan memisahkan yang tidak terkait (walaupun berinteraksi) elemen. di samping itu juga mempermudah membuat perawatan dan update kode, dan menyederhanakan debugging.

#### \* Class

Secara Sintaks, definisi class di mulai dengan kata class, yang diikuti dengan nama method dalam class diawali dengan kurung kurawal buka dan tutup.

Method dalam class diawali dengan kata function, yang diikuti dengan nama method serta argumen. Atribut dalam class di definisikan sebagai Variabel. yang terdiri 3 jenis yaitu : Public, Private, dan Protected. Ivaran dari sebuah method di tuliskan 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

object digunakan kata new. Contoh = \$object = new class name(); Untuk memanggil methods menggunakan sintak object operator.

Contoh = Subject → method name(); 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 property atau method dalam class yang sama menggunakan kata this

Contoh:

```
Class matematika {
```

```
    Publik $ pi = 3.14;
```

```
    Function tambah ($a, $b) {
```

```
        $c = $a + $b;
```

```
        return $c;
```

```
    }
```

```
    Function keliling - lingkaran kuadrat ($r) {
```

```
        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;
```

```
    }
```

```
}
```

### \* Konstruktor

Konstruktor merupakan method yang dieksekusi secara otomatis pertama kali saat sebuah class diturunkan menjadi sebuah objek

Contoh:

```
Class matematika () {
```

```
    public $ pi;
```

```
    Function - Konstruktion () {
```

```
        $this -> pi = 3.14
```

```
    }
```

```
    Function keliling - lingkaran ($r) {
```

```
        $kel = 2 * $this -> $pi * $r;
```

```
        return $kel;
```

```
    }
```

```
}
```

## BAB II

### PERCOBAAN DAN LATIHAN

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 = $math -> math luas - lingkaran ($jari) ;  
echo "menghitung keliling dan luas lingkaran <br>" ;  
echo "Jari-jari : " . $jari . "<br>" ;  
echo "Keliling = " . $kel - lingkaran . "<br>" ;  
echo "Luas = " . $luas - lingkaran  
?>
```

## Percobaan 2

```
<?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;
```

```
    }
```

```
}
```

```
?>
```

\* Percobaan 3.

```
<?php
```

```
include class-matematika.php;
```

```
$mth = new matematika ();
```

```
$jari = 10;
```

```
$kel - lingkaran = $mth->keliling - lingkaran ($jari);
```

```
$luas - lingkaran = $mth->luas - lingkaran ($jari);
```

```
echo "menghitung keliling dan luas lingkaran <br>";
```

```
echo "Jari: $jari" <br>
```

```
echo "keliling = " . $kel - lingkaran . " <br>";
```

```
echo "luas = " . $luas - lingkaran
```

```
?>
```

## Latihan 1

PHP

Class balok {

private \$p; private \$l; private \$t;

function \_\_balok (\$p, \$l) {

\$this->p = \$p; \$this->l = \$l;

echo "luas = ". \$p. " x ". \$l. " = ". \$this->p \* \$this->l. " cm <sup>2</sup>";

}

function hitung\_balok (\$p, \$l, \$t) {

\$this->t = \$t; \$this->p = \$p; \$this->l = \$l;

echo "volume = ". \$p. " x ". \$l. " = ". \$this->p \* \$this->l \* \$this->t. " cm <sup>3</sup>";

}

function form\_hitung\_balok () {

echo "<form method='GET'>";

echo "<h1> luas dan volume balok </h1>";

echo "<h3> isi data </h3>";

echo "<br> Panjang;";

echo "<br><input type='text' name='l' place holder='masukan nilai'> cm <br>";

echo "<br> lebar;";

echo "<br><input type='text' name='l' place holder='masukan nilai'> cm <br>";

echo "<br> tinggi;";

echo "<br><input type='text' name='t' place holder='masukan nilai'> cm <br>";

echo "<p style='margin-left: 15px'> x <input type='Submit' name='Submit' value='hitung'>";

echo "</form>";

}

}



```

$hitung-balok = new balok;
$l-balok = new balok;
$hitung-balok -> from-hitung balok ();
if (isset ( $_GET ['Submit'] )) {
    $p = $_GET ['p']; $l = $_GET ['l']; $t = $_GET ['t'];
    echo $l-balok -> l-balok ($p, $l);
    echo "<br>";
    echo $hitung-balok -> hitung-balok ($p, $l, $t);
}
?>

```

Latihan 2.

<?php

```

class bola {

```

```

    private $r;

```

```

    public $phi = 3.14;

```

```

    function luas-bola ($r) {

```

```

        $this -> r = $r; $p2 = $this -> r * $this -> r;

```

```

        $luas = 4 * $this -> phi * $this -> r;

```

```

        ($luas). "cm <sup>2 </sup> <br>";
    }

```

```

    function vbola ($r) {

```

```

        $this -> r = $r; $p3 = $this -> r * $this -> r * $this -> r;

```

```

        $Volume = 1/3 * $this -> phi * $this -> r;

```

```

        echo "Volume = 1/3 * $this -> phi * $this -> r";

```

```

        <sup>3 </sup> = ". ($Volume). "cm <sup>3 </sup> <br>";
    }

```



```

}
function Form - hitung bola () {
    echo '<form method = "GET">';
    echo "<h1> hitung luas dan volume bola </h1>";
    echo 'Jari - Jari = <input type = "Submit" name = "Submit Volume"
hitung">'; echo '</form>';
}
}

```

```

$ bola = new bola;
$ bola => form - hitung bola ();
if (isset ( $_GET ['r'] )) {
    $r = $_GET ['r'];
    echo " jari - jari = " . $r . " cm <br>";
    echo $ bola -> luas - bola ($r);
    echo $ bola -> v bola ($r);
}
}

```

??

### Latihan 3.

<? php

```
class konversi nilai {  
    public $ nilai;  
    function nilai ($z) {  
        $ this → nilai = $ z;  
    }  
}
```

```
function konversi () {  
    if ($ this → nilai <= 100 && $ this → nilai >= 85) { $ grade = 'A';  
    } else if ($ this → nilai < 85 && $ this → nilai >= 77) { $ grade = 'B';  
    } else if ($ this → nilai < 77 && $ this → nilai >= 61) { $ grade = 'C';  
    } else if ($ this → nilai < 61 && $ this → nilai >= 50) { $ grade = 'D';  
    } else { $ grade = 'E';  
    } return $ grade;  
}
```

```
function form - nilai () {  
    echo "< form method = 'GET'>";  
    echo "< h1 > konversi nilai < / h1 >";  
    echo "< input type = 'text' name = 'nilai' place holder = 'masukan nilai'>";  
    echo "< br > < input type = 'Submit' name = 'konversi'>";  
    echo "< / form >";  
}
```

```
$ score = new konversi nilai;  
$ score → form nilai ();  
if (isset ($ - GET ['konversi'])) {  
    echo 'nilai : ' . $ - GET ['nilai'];  
    echo "< br > nilai dikonversi :";  
    $ x = $ score → nilai ($ - GET ['nilai']) /;  
    echo $ score → konversi ();  
}
```

?>

## BAB III

### TAMPILAN PROGRAM

#### Percobaan 1



#### Percobaan 2





## Latihan 1

luas dan volume balok

isi data

panjang :  cm

lebar :  cm

tinggi :  cm

hitung

luas =  $25 \times 10 = 250 \text{ cm}^2$

volume =  $25 \times 10 \times 15 = 3750 \text{ cm}^3$

## Latihan 2

hitung luas dan volume bola

jari-jari :  cm

hitung

jari-jari = 21cm

luas =  $4 \times 3.14 \times 21^2 = 5538.96 \text{ cm}^2$

volume =  $\frac{4}{3} \times 3.14 \times 21^3 = 38772.72 \text{ cm}^3$

### Latihan 3

localhost:3000/latihan3/latihan3-konversi-nilai

Konversi Nilai

masukkan nilai

Submit

nilai : 90

nilai dikonversi : A

localhost:3000/latihan3/latihan3-konversi-nilai

Konversi Nilai

masukkan nilai

Submit

nilai : 40

nilai dikonversi : E

## BAB 4

### KESIMPULAN

Jadi kesimpulan di oop terdapat beberapa script yaitu:

1. Class

2. Object

Contoh: `$object = new classhome();`

3. This

This disini untuk mengakses properti atau method

4. Constructor

merupakan method yang di eksekusi secara otomatis

Dari kesimpulan Tersebut. dapat di jelaskan bahwa:

- o> Mahasiswa dapat memahami konsep objek oriented programming (oop)
- o> Mahasiswa dapat memahami oop ~~melalui~~ menggunakan PHP
- o> Mahasiswa dapat membuat program Sederhana menggunakan oop PHP