

PRAKTIKUM
PEMROGRAMAN WEB
SEMESTER GENAP TAHUN AKADEMIK 2021/2022

Tanggal

.....

Materi

PHP OOP

10

BAB I

DASAR TEORI

Object Oriented Programming (OOP)

Definisi yang pertama penting untuk OOP adalah class dan object. Class adalah generalisasi definisi sesatu, class dapat dijadikan sebagai cetak biru. Object yaitu implementasi yg spesifik dari suatu hal. Salah satu prinsip OOP adalah modulifitas yaitu memecah aplikasi menjadi submodul tertentu website beranak misalkan hal tersebut misalnya : berinteraksi dengan database menggunakan formulir pengisian email, menulis situs HTML idm masing-masing hal tersebut dpt disajikan menggunakan model yaitu class, kode dpt dikembangkan secara mandiri dengan memisalkan yg tidak terkait (maka dari berinteraksi) elemen. disamping itu juga mempermudah membuat perbaikan dan update kode dan memudahkan debugging.

* Class

Secara sintaks, definisi class dimulai dengan kata class yg dilanjuti dengan nama class tersebut, nama class tidak dpt menggunakan reserved word, selanjutnya dilanjuti dengan tanda kurung buka dan tutup method dalam class diawali dengan kata function yaitu dilanjuti dengan nama method serta argumen, atribut dalam class di definisikan sebagaimana variabel yang terdapat 3 jenis yaitu : Public, Private dan Protected. Lebih 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 menyelesaikan class tersebut dengan membuat sebuah objek (class instance). Untuk membuat object digunakan kata New contoh objek = New class name (); Untuk mengakses method menggunakan sintaks objek.method name (); Contoh : objek = New object () → method name (); Untuk mengakses seperti object digunakan sintaks \$objek → property name ;

Contoh :

```
class matematika {  
    function tambah ($a, $b) {  
        $c = $a + $b;  
        return $c;  
    }  
  
    $math = new matematika();  
    $bil1 = 2;  
    $bil2 = 5;  
    hasil = $math->tambah($bil2, $bil2);  
    echo $hasil;  
}
```

* this

Sebuah method dimulai dengan kata this 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 * $r * $r;  
        return $luas;  
    }  
}
```

* Konstruktor

Konstruktor merupakan method yang diinisiasi secara otomatis ketika saat sebuah class diturunkan menjadi sebuah objek.

Contoh :

```
class matematika () {  
    public $pi;  
  
    function __construct () {  
        $this->$pi = 3.14;  
    }  
  
    function keliling_lingkaran ($r) {  
        $kel = 2 * $this->$pi * $r;  
        return $kel;  
    }  
}
```


Pertobaan 3

< ?PhP

```
include 'class-Matematika.php';
$math = new Matematika();
$jati = 10;
$ke_lingkaran = $math -> keliling_lingkaran($jati);
$luas_lingkaran = $math -> luas_lingkaran($jati);
echo "menghitung keliling dan luas lingkaran <br>";
echo "$jati-jati = " . $jati . " <br>";
echo "keliling = ". kel_lingkaran . "<br>";
echo "luas = ". $luas_lingkaran;
```

?>

* Lecikan 1.

< ? PHP

Class batch {

Private \$P; Private \$I; Private \$t;

Function l - batch (\$P, \$I) {

\$this -> P = \$P; \$this -> I = \$I;

echo "mas = ". \$P. "x". \$I = \$this -> P * \$this -> I; CM < sup > 2 </

Sup >;

}

Function hitung - batch (\$P, \$l, \$t) {

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

echo "Volume = ". \$P. "x". \$l. "x". \$t; \$this -> P * \$this -> l * \$this -> t; CM

< sup > 3 < / sup >;

}

Function form - hitung batch () {

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

echo "< h1 > mas dan volume batch < / h1 >";

echo '< h3 > 1. Si data < / h3 >';

echo '< br > Panjang :';

echo '< br > < input type = "text" name = "P" Placeholder = "masukkan nilai" > cm < br >';

echo '< br > lebar :';

echo '< br > < input type = "text" name = "l" Placeholder = "masukkan nilai" > cm < br >';

echo '< br > tinggi :';

echo '< br > < input type = "text" name = "t" Placeholder = "masukkan nilai" > cm < br >';

echo '< br > < input type = "text" name = "t" Placeholder = "masukkan nilai" > cm < br >';

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

echo '< / form >';

}

\$ hitung - batch = new batch();

\$ l - batch = new batch();

\$ hitung - batch -> form - hitung batch ();

IF isset (\$ -> GET ['Submit']) {

\$P = \$ -> GET ['P']; \$l = \$ -> GET ['l']; \$t = \$ -> GET ['t'];

echo \$ l - batch -> l - batch (\$P, \$l);

echo "< br >";

echo \$ hitung - batch -> hitung batch (\$P, \$l, \$t);

}

?>

Lektion 2

< ? PHP

class bola {

private \$r;

public \$Phi = 3.14;

function lwas_bola (\$r) {

\$this->r = \$r; \$P2 = \$this->r * \$this->r;

\$lwas = 4 * \$this->Phi * \$P2;

echo 'lwas = 4 * ' . \$this->Phi . ' * ' . \$this->r . '

(\$lwas). ' cm < sup > 2 < sup > 2 < sup > = ' .

}

function Vbola (\$r) {

\$this->r = \$r; \$P3 = \$this->r * \$this->r * \$this->r;

\$volume = 4/3 * \$this->Phi * \$P3;

- echo 'volume = 4/3 * ' . \$this->Phi . ' * ' . \$this->r . '

' . (\$volume). ' cm < sup > 3 < sup > = ' .

}

function Vbola (\$r) {

\$this->r = \$r; \$P3 = \$this->r * \$this->r * \$this->r;

\$volume = 4/3

}

function Form_hilfebola () {

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

echo '< h1 > hilfe lwas der volume bola < /h1 >';

echo ' Jari' . 'Jari' = < input type = "text" name = "r" >';

Placeholder = "masukkan Jari" . 'Jari' > < br >';

echo '< br > < input type = "Submit" name = "Submit" value = "hilfe" >';

}

\$bola = new bola;

\$bola->Form_hilfebola();

if (!isset (\$ -> GET ['r'])) {

\$r = \$ -> GET ['r'];

echo " Jari" . 'Jari' = ". \$t. "cm < br >";

echo \$bola->lwas_bola (\$r);

echo \$bola->Vbola (\$r);

}

?>

Latihan 3

< ? PHP

```
class konversiNilai {
```

```
    Public $ nilai;
```

```
    Function nilai ($z) {
```

```
        $this->nilai = $z;
```

```
}
```

```
Function konversi () {
```

```
    If ($this->nilai <= 100.00 && $this->nilai >= 85) { $Grade = "A";
```

```
    } Else If ($this->nilai >= 85.00 && $this->nilai >= 77) { $Grade = "B";
```

```
    } Else If ($this->nilai <= 77.00 && $this->nilai >= 61) { $Grade = "C";
```

```
    } Else If ($this->nilai <= 61.00 && $this->nilai >= 0) { $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" Placeholder = "masukkan  
Nilai">;
```

```
Echo '<br><Input type = "Submit" name = "konversi">;
```

```
}
```

```
}
```

```
$Scote = new konversiNilai;
```

```
$Scote -> form_nilai();
```

```
If (!isset($_GET['konversi'])) {
```

```
Echo 'Nilai: : $_GET['Nilai']';
```

```
Echo '<br>Nilai dikonversi =';
```

```
$x = $Scote -> nilai($_GET['Nilai']);
```

```
Echo '$Scote -> konversi();';
```

```
}
```

? >

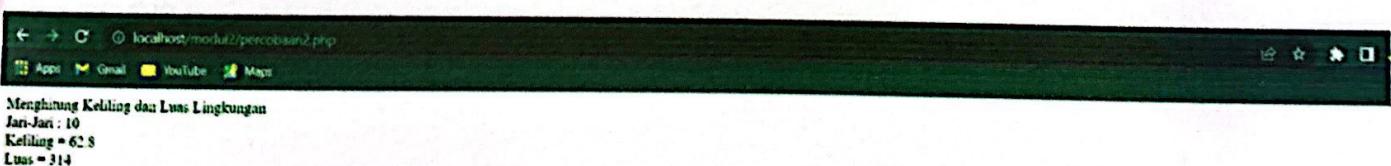
BAB III

TAMPILAN PROGRAM

Percobaan 1



Percobaan 2



localhost/modul1/balok.php?r=25&submit=hitung

App Gmail YouTube Maps

luas dan volume balok

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

localhost/modul2/bola.php?r=21&submit=hitung

App Gmail YouTube Maps

hitung luas dan volume bola

jari-jari : cm

hitung

jari-jari = 21cm
luas = $4 \times 3.14 \times 21^2 = 5535.96 \text{ Cm}^2$
volume = $4/3 \times 3.14 \times 21^3 = 38772.72 \text{ Cm}^3$

Latihan 3

Konversi Nilai

Nilai :

nilai : 90
nilai dikonversi : A

Konversi Nilai

Nilai :

nilai : 40
nilai dikonversi : E

BAB IV

KESIMPULAN

di kesimpulannya di OOP terdapat beberapa script yang ada:

class

object

contoh : \$ object = new class name();

this

this di sini untuk mengakses Properti atau method

• Constructor

Melakukan method yang di gunakan secara otomatis

Dari kesimpulan dapat dijelaskan bahwa :

- .) Mahasiswa dapat memahami Konsep Object Oriented Programming (OOP)
- .) Mahasiswa dapat memahami OOP menggunakan PHP
- .) mahasiswa dapat membuat program sederhana menggunakan OOP PHP