

Nama:Satrio Raihan W

NIM:G.211.21.0072

Matkul:Pemrograman Framework

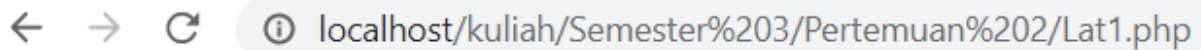
Latihan 1:Class dan Object+Modifier



```
1 <?php
2 //class mobil
3 Class Mobil{
4     public $nama;
5     public $merk;
6     function getInfo(){
7         echo "Nama mobil: ".$this->nama."<br/>";
8         echo "Merk: ".$this->merk."<br/>";
9     }
10 }
11 //bagian main
12 $ferari=new Mobil();
13 $ferari->nama="xxx";
14 $ferari->merk="aaa";
15 $ferari->getInfo();
16 ?>
```

a.Bagaimana hasil tampilan diatas

Jawab:



← → ↻ ⓘ localhost/kuliah/Semester%203/Pertemuan%202/Lat1.php

Nama mobil: xxx

Merk: aaa

b. Buatlah method overload getInfo dengan parameter \$a, lalu jalankan dan amati perubahan yang terjadi.

Jawab:



The screenshot shows a web browser window with the address bar displaying 'localhost/kuliah/Semester%203/Pertemuan%202/Lat1.php'. The main content area displays the output of a PHP script: 'Nama mobil: DDD' followed by 'Merk: FFF' on a new line. Above the browser window, a code editor window is visible, showing the following PHP code:

```
1 <?php
2 //class mobil
3 Class Mobil{
4 public $nama;
5 public $merk;
6 function getInfo($a){
7 echo "Nama mobil: ". $a->nama."<br/>";
8 echo "Merk: ". $a->merk."<br/>";
9 }
10 }
11 //bagian main
12 $ferari=new Mobil();
13 $ferari->nama="DDD";
14 $ferari->merk="FFF";
15 $ferari->getInfo($ferari);
16 ?>
```

c. Lalu simpulkan apa yang anda peroleh dari Latihan 1

Jawab:

1. Membuat sebuah class:

```
<?php
Class nama_class
{
} ?>
```

2. Membuat sebuah properti:

```
Modifier $nama_properti;
```

3.Membuat sebuah method:

```
Modifier function nama_method(){  
    Isi_method;  
}
```

4.Cara memanggil sebuah method:

```
$nama_object->nama_method();
```

Latihan 2:

Sebelum dimodifikasi:

```
1  <?php  
2  Class mahasiswa{  
3  public $nama;  
4  public $nim;  
5  function __construct($a,$b){  
6      $this->nama=$a;  
7      $this->nim=$b;  
8  echo "Kelas telah dibuat<br/><br/>";  
9  }  
10 function cetak(){  
11 echo $this->nama."<br/>".$this->nim."<br/><br/>";  
12 }  
13 function __destruct(){  
14 echo "Kelas telah dihancurkan";  
15 }  
16 }  
17 ?>
```

```
1  <?php  
2  require_once("Lat2a.php");  
3  $mhs2=new mahasiswa("Pennyka","0910683073");  
4  $mhs2->cetak();  
5  ?>
```

← → ↻ ⓘ localhost/kuliah/Semester%203/Pertemuan%202/lat2B.php

Kelas telah dibuat

Pennyka
0910683073

Kelas telah dihancurkan

Sesudah dimodifikasi:

```
1 <?php
2 Class mahasiswa{
3 public $nama;
4 public $nim;
5 function __construct($a,$b){
6     $this->nama=$a;
7 $this->nim=$b;
8 echo "Kelas telah dibuat<br/><br/>";
9 }
10 function __construct($a,$b){
11     $this->nama=$a;
12 $this->nim=$b;
13 echo "Kelas telah dibuat<br/><br/>";
14 }
15 function cetak(){
16 echo $this->nama."<br/>".$this->nim."<br/><br/>";
17 }
18 function __destruct(){
19 echo "Kelas telah dihancurkan";
20 }
21 }
22 ?>
```

```
1 <?php
2 require_once("Lat2a.php");
3 $mhs2=new mahasiswa("Pennyka","0910683073");
4 $mhs2->cetak();
5 ?>
```

← → ↻ ⓘ localhost/kuliah/Semester%203/Pertemuan%202/lat2B.php

Fatal error: Cannot redeclare mahasiswa::__construct() in E:\Program\htdocs\kuliah\Semester 3\Pertemuan 2\Lat2a.php on line 10

Jawab: *Terjadi error karena memiliki function sama yang terdefinisi dalam satu file yang sama*

Latihan 3:

```
1 <?php
2 class mahasiswa{
3 private $nama;
4 private $nim;
5 public function tampilkan_nama() {
6     return $this->nama;
7 }
8 function __construct(){
9 function setName($a){
10 $this->nama=$a;
11 }
12 function setNim($b){
13 $this->nim=$b;
14 }
15 function getName(){
16 return $this->nama;
17 }
18 function getNim(){
19 return $this->nim;
20 }
21 function destruct(){
22 }
23 ?>
```

```
1 <?php
2 require_once("Lat3a.php");
3 $mhs1=new mahasiswa();
4 $mhs1->nama="hendra";
5 echo $mhs1->nama;
6 ?>
```

Fatal error: Uncaught Error: Cannot access private property mahasiswa::\$nama in E:\Program\htdocs\kuliah\Semester 3\Pertemuan 2\Lat3b.php:4 Stack trace: #0 {main} thrown in E:\Program\htdocs\kuliah\Semester 3\Pertemuan 2\Lat3b.php on line 4

a. Apakah program error? Jika error mengapa hal itu dapat terjadi?

Jawab: terjadi error karena modifier private hanya dapat digunakan dalam class mahasiswa itu sendiri.

b. Rubahlah modifier dari variable nama dan nim menjadi protected dan public. Lalu amati perubahan yang terjadi.

Jawab:

Saat diubah ke modifier protected tetap terjadi error

Fatal error: Uncaught Error: Cannot access protected property mahasiswa::\$nama in E:\Program\htdocs\kuliah\Semester 3\Pertemuan 2\Lat3b.php:4 Stack trace: #0 {main} thrown in E:\Program\htdocs\kuliah\Semester 3\Pertemuan 2\Lat3b.php on line 4

Saat diubah ke modifier public, tercetak hasil sebagai berikut

hendra

c. Modifikasilah Lat4_3b sehingga dapat member dan mencetak isi dari nim dan nama dengan modifier private

```

1 <?php
2 class mahasiswa
3 {
4     private $nama;
5     private $nim;
6     function __construct(
7         $a = "Satrio Raihan Wijanarko <br>",
8         $b = "G.211.21.0072"
9     ){
10         $this->nama = $a;
11         $this->nim = $b;
12     }
13     function setNama($a)
14     {
15         $this->nama = $a;
16     }
17     function setNim($b)
18     {
19         $this->nim = $b;
20     }
21     function getNama()
22     {
23         return $this->nama;
24     }
25     function getNim()
26     {
27         return $this->nim;
28     }
29 }
30 ?>

```

```

1 <?php
2 require_once("Lat3a.php");
3 $mhs1=new mahasiswa();
4 echo $mhs1->getNama(). $mhs1->getNim();
5 ?>

```

Satrio Raihan Wijanarko
G.211.21.0072

d.Simpulkan apa yang anda peroleh dari latihan 3!

1.Public:Untuk mendefinisikan data atau metode yang akan terlihat dari luar oleh siapapun dan dimanapun.

2.Private:Untuk mendefinisikan data atau metode agar hanya terlihat pada class/object itu sendiri. 3.Protected:Untuk mendefinisikan data atau metode untuk tidak terlihat dari luar (seperti private), tetapi akan dapat diakses oleh “anak” dari class tersebut.

Latihan 4:

```
1 <?php
2 require_once("Lat3a.php");
3 class asisten extends mahasiswa
4 {
5     function __construct(){}
6 }
7
```

```
1 <?php
2 require_once("Lat4a.php");
3 $as = new asisten();
4 $as->setNama("tes");
5 echo $as->getNama();
```

tes

Simpulkan apa yang Anda peroleh dari Latihan 4!

Jawab:Dalam PHP OOP,inheritance(pewarisan) dalah cara untuk menggunakan kembali kode objek yang ada, atau untuk mendirikan subtype dari objek yang sudah ada, atau keduanya, tergantung pada dukungan bahasa pemrograman.

Latihan 5:

```
1 <?php
2 abstract class mahasiswa
3 {
4     abstract protected function getTugasAkhir();
5     abstract protected function getProgram($postfix);
6     public function tugasAkhir()
7     {
8         print $this->getTugasAkhir() . "<br>";
9     }
10 }
11 class sarjana extends mahasiswa
12 {
13     protected function getTugasAkhir()
14     {
15         return "Skripsi";
16     }
17     public function getProgram($postfix)
18     {
19         print "{$postfix} S1";
20     }
21 }
22 class magister extends mahasiswa
23 {
24     public function getTugasAkhir()
25     {
26         return "Tesis";
27     }
28     public function getProgram($postfix)
29     {
30         print "{$postfix} S2";
31     }
32 }
33
```

```
1 <?php
2 require_once("Lat5a.php");
3 $s = new sarjana;
4 $s->getProgram('Mahasiswa') . "<br>";
5 $s->tugasAkhir();
6 $m = new magister;
7 $m->getProgram('Mahasiswa') . "<br>";
8 $m->tugasAkhir();
```

a. Bagaimana hasil tampilan dari program di atas?

Jawab:

← → ↻ ⓘ localhost/kuliah/Semester%203/Pertemuan%202/Lat5b.php

Mahasiswa S1Skripsi

Mahasiswa S2Tesis

b. Hapuslah kode baris 29 – 32 pada Lat4a.php, lalu jalankan Lat5b.php. Bagaimana hasil tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?

Jawab: *Terjadi error dikarenakan subclass yang mewarisi superclass harus menuliskan semua method dari abstrak dari superclassnya*

c. Simpulkan apa yang anda peroleh pada latihan 5?

Jawab: *Class dan method abstract hanya bisa dipanggil melalui class turunannya saja, kemudian Setiap class yang mewarisi class abstract, wajib menuliskan seluruh method abstract yang dipunyai oleh parent class-nya (super class).*

Latihan 6:

a. Jelaskan maksud dari program di atas?

Jawab: *Program diatas merupakan contoh dari penggunaan object interfaces pada PHP OOP*

b. Hapuslah kode baris 27–29, lalu jalankan lat4_6.php. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut dapat terjadi?

Jawab: *Terjadi error dikarenakan dalam object interfaces, class yang mengimplementasikan suatu interface harus menggunakan method-method yang ada pada interface tersebut dengan nama dan spesifikasi yang sama persis. Maka dari itu jika method baz pada class D dihapus akan menyebabkan error karena class D mengimplementasi class C yang mana method baz merupakan bagian dari class C.*

c. Dari contoh kode diatas, buatlah class baru dengan nama “e” yang mempunyai method foo dan bar.



```
1  <?php
2  interface a
3  {
4      public function foo();
5  }
6  interface b
7  {
8      public function bar();
9  }
10 interface c extends a, b
11 {
12     public function baz();
13 }
14 class d implements c
15 {
16     public function foo()
17     {
18     }
19     public function bar()
20     {
21     }
22     public function baz()
23     {
24     }
25 }
26 class e implements a, b
27 {
28     public function foo()
29     {
30     }
31     public function bar()
32     {
33     }
34 }
35 ?>
36
```

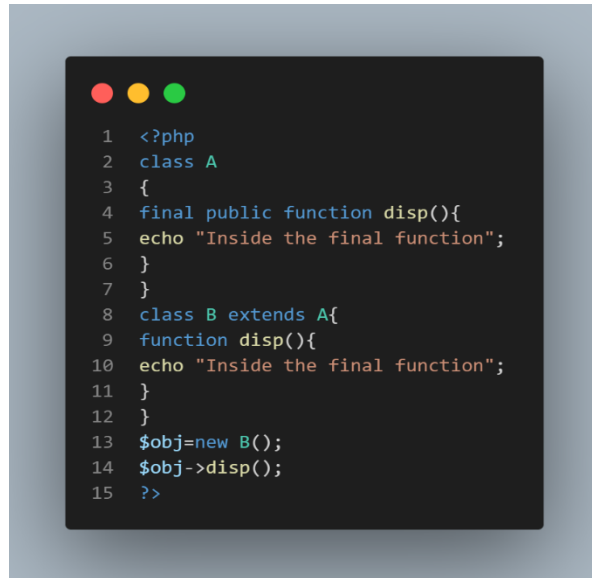
d. Simpulkan apa yang anda peroleh dari latihan 6!

-Class tidak bisa mengimplementasikan dua interface yang mempunyai nama method yang sama.

-Interface bisa diwariskan seperti class menggunakan “extends”.

- Class yang mengimplementasikan interface harus menggunakan method-method yang ada pada interface tersebut dengan nama dan spesifikasi yang sama persis.

Latihan 7:



```
1 <?php
2 class A
3 {
4     final public function disp(){
5     echo "Inside the final function";
6     }
7 }
8 class B extends A{
9     function disp(){
10    echo "Inside the final function";
11    }
12 }
13 $obj=new B();
14 $obj->disp();
15 ?>
```

a. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?

Jawab: *Terjadi error karena method A tidak dapat di override*

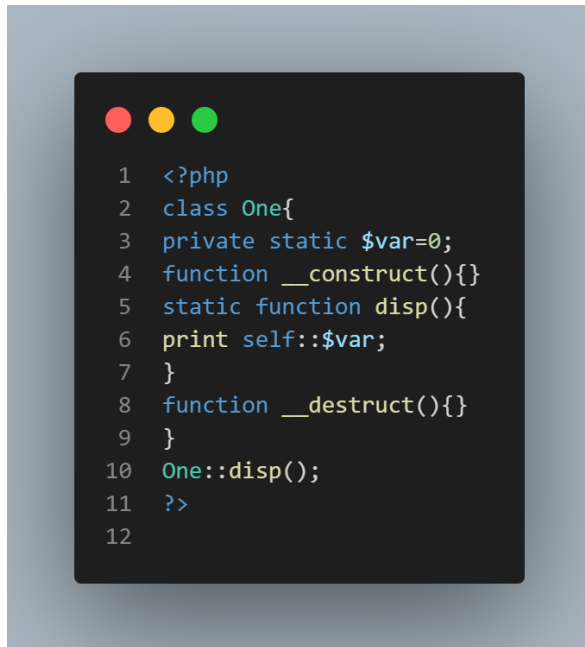
b. Modifikasi program di atas dengan menghapus kata final pada kode baris 5 dan menambahkan kata final pada baris 2. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?

Jawab: *karena tujuan digunakannya keyword final adalah agar class tidak akan di override.*

c. Simpulkan apa yang anda peroleh dari latihan 7!

Jawab: *Final keyword berfungsi untuk mencegah proses overriding method pada class anak (subclass). Hal ini bisa kita terapkan pada metode dan class. Apabila metode kita berikan status final, maka metode tersebut tidak akan bisa dioverride, begitu juga pada class, apabila kita berikan status "final" pada deklarasi class maka class tersebut tidak bisa diperpanjang (diwariskan).*

Latihan 8:



```
1  <?php
2  class One{
3  private static $var=0;
4  function __construct(){
5  static function disp(){
6  print self::$var;
7  }
8  function __destruct(){
9  }
10 One::disp();
11 ?>
12
```

localhost/kuliah/Semester%203/Pertemuan%202/Lat8.php

0

Simpulkan apa yang anda peroleh dari latihan 8!

Jawab: *Static property dan static method adalah property dan method yang langsung dapat diakses dari class tanpa instansiasi class (pembuatan object) terlebih dahulu. Cara mendeklarasikannya yaitu property atau method diawali menggunakan keyword static.*