Nama :Resha Raditya Danaris

NIM :G.211.21.0067

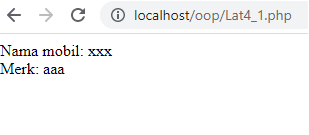
Matkul :Framework

**Latihan 1: Class dan Object + Modifier**

Tuliskan code dibawah ini, simpan dalam file Lat4\_1.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <?php  //class mobil Class Mobil{  public $nama; public $merk;  function getInfo(){  echo "Nama mobil: ".$this->nama."<br/>"; echo "Merk: ".$this->merk."<br/>";  }  }  //bagian main  $ferari=new Mobil();  $ferari->nama="xxx";  $ferari->merk="aaa";  $ferari->getInfo();  ?> |

* 1. Bagaimanahasiltampilan di atas?
  2. Buatlahsebuah method overload getInfodengan parameter $a. Lalujalankandanamatiperubahan yang terjadi.
  3. Lalu simpulkan apa yang Anda peroleh dari Latihan 1!

1. 
2. Error, karena di php tidak bisa mengoverload method
3. Kesimpulan:

1.Cara membuat class pada php

<?php

Class nama\_class{

}

?>

2.Cara penulisan property

Modifier $nama\_properti;

3.Penulisan method

Modifier function nama\_method(){

Isi\_method;

}

4.Cara inisiasi object

$nama\_object = new nama\_class();

5.Cara mengisi property atau mendefinisikan property

$nama\_object->properties=”aaa”;

6.Cara memanggil/menjalankan method pada suatu class

$nama\_object->nama\_methode();

**Latihan 2:**

Tuliskan code dibawah ini, simpan dalam file Lat4\_2a.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <?php  Class mahasiswa{  public $nama; public $nim;  function construct($a,$b){ |

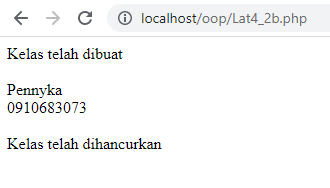
|  |  |
| --- | --- |
| 7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | $this->nama=$a;  $this->nim=$b;  echo "Kelas telah dibuat<br/><br/>";  }  function cetak(){  echo $this->nama."<br/>".$this->nim."<br/><br/>";  }  function destruct(){  echo "Kelas telah dihancurkan";  }  }  ?> |

Tuliskan code dibawah ini, simpan dalam file Lat4\_2b.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <?php require\_once("lat4\_2a.php");  $mhs2=new mahasiswa("Pennyka","0910683073");  $mhs2->cetak();  ?> |

Modifikasilah latihan 4\_2a dengan menambahkan 1 konstruktor lagi di lat 4\_2a baris 11. Lalu jalankan. Bagaimana hasil tampilan di atas sebelum dan sesudah dimodifikasi? Lalu simpulkan apa yang Anda peroleh dari Latihan 2!

Jawab:



**Latihan 3:**

Tuliskan code dibawah ini, simpan dalam file Lat4\_3a.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | <?php  class mahasiswa{  private $nama; private $nim;  function construct(){} function setNama($a){  $this->nama=$a;  }  function setNim($b){  $this->nim=$b;  }  function getNama(){  return $this->nama;  }  function getNim(){  return $this->nim;  }  function destruct(){}  }  ?> |

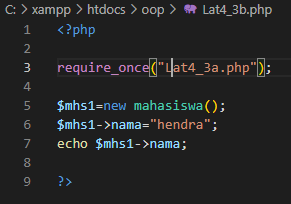
Tuliskan code dibawah ini, simpan dalam file Lat4\_3b.php!

|  |  |
| --- | --- |
| 1  2 | <?php |

|  |  |
| --- | --- |
| 3  4  5  6  7  8 | require\_once("lat4\_3a.php");  $mhs1=new mahasiswa();  $mhs1->nama="hendra"; echo $mhs1->nama;  ?> |

1. Apakah program error? Jika error mengapahalitudapatterjadi?
2. Rubahlah modifier dari variable namadannimmenjadi protected dan public. Laluamatiperubahan yang terjadi.
3. Modifikasilah Lat4\_3b sehingga dapat member dan mencetak isi dari nim dan nama dengan modifier private
4. Simpulkanapa yang andaperolehdarilatihan 3!

Jawab:

1. Error, karena property yang bermodifier private hanya bisa digunakan pada class mahasiswa sendiri
2. Protected tetep error, public hasilnya hendra
3. 
4. Kesimpulan:

1.Penggunaan modifier

Modifier Keterangan

-Public :Untuk mendefinisikan data atau metode yang akan terlihat dariluar oleh siapapun dan dimanapun.

-Private :Untuk mendefinisikan data atau metode agar hanya terlihatpada class/object itu sendiri.

-Protected:Untuk mendefinisikan data atau metode untuk tidak terlihat dariluar (seperti private), tetapi akan dapat diakses oleh “anak” dari class tersebut.

2.Modifier protected dan private, properties bisa dipanggil dengan mengimplementasikan setter-getter.

**Latihan 4:**

Tuliskan code dibawah ini, simpan dalam file Lat4\_4a.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <?php require\_once("lat4\_3a.php");  class asisten extends mahasiswa  {  function construct(){}  } |

Tuliskan code dibawah ini, simpan dalam file Lat4\_4b.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <?php require\_once("lat4\_4a.php");  $as = new asisten();  $as->setNama("tes"); echo $as->getNama(); |

Simpulkan apa yang Anda peroleh dari Latihan 4!

Jawab:

Pada php oop, class asisten (child) bisa memanggil method dari mahasiswa (parent)

**Latihan 5: Abstract Class**

Tuliskan code dibawah ini, simpan dalam file Lat4\_5a.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | <?php  abstract class mahasiswa  {  abstract protected function getTugasAkhir(); abstract protected function getProgram($postfix);  public function tugasAkhir()  {  print $this->getTugasAkhir() . "<br>";  }  }  class sarjana extends mahasiswa  {  protected function getTugasAkhir()  {  return "Skripsi";  }  public function getProgram($postfix) |

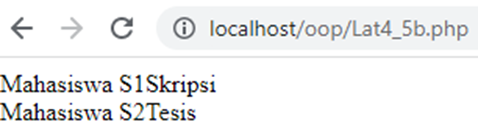
|  |  |
| --- | --- |
| 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | {  print "{$postfix} S1";  }  }  class magister extends mahasiswa  {  public function getTugasAkhir()  {  return "Tesis";  }  public function getProgram($postfix)  {  print "{$postfix} S2";  }  } |

Tuliskan code dibawah ini, simpan dalam file Lat4\_5b.php!

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <?php require\_once("lat4\_5a.php");  $s = new sarjana;  $s->getProgram('Mahasiswa') ."<br>";  $s->tugasAkhir();  $m = new magister;  $m->getProgram('Mahasiswa') ."<br>";  $m->tugasAkhir(); |

1. Bagaimana hasil tampilan dari program di atas?
2. Hapuslah kode baris 29 – 32 pada lat4\_5a.php, lalu jalankan lat4\_5b.php. Bagaimana hasil tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?
3. Simpulkan apa yang anda peroleh pada latihan 4?

Jawab:



1. Error, pada class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.
2. Kesimpulan:

class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya

**Latihan 6:**

Tuliskankodeberikutpada lat4\_6.php

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <?php interface a  {  public function foo();  }  interface b  {  public function bar();  }  interface c extends a, b  {  public function baz();  }  class d implements c  {  public function foo()  {  }  public function bar()  { |

|  |  |
| --- | --- |
| 25  26  27  28  29  30  31 | }  public function baz()  {  }  }  ?> |

1. Jelaskan maksud dari program di atas?
2. Hapuslah kode baris 27 – 29, lalu jalankan lat4\_6.php. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut dapat terjadi?
3. Dari contoh kode diatas, buatlah class baru dengan nama “e” yang mempunyai method **foo**

dan **bar**.

1. Simpulkan apa yang anda peroleh darilatihan 6!

Jawab

1. Maksud dari program tersebut adalah penggunaan object interfaceb.
2. Error, karena pada object interface, ketika kita mengimplementasikan object tersebut, seluruhmethod pada interface harus diimplementasikan seluruhnya. Karena class d mengimplementinterface c, maka method-method pada interface c harus diimplementasikan seluruhnya.c.
3. Error, method foo dan bar terdapat pada interace a dan b, pada penerapannya sebuah classbaru tidak dapat mengimplementasikan method yang sama pada dua interface
4. kesimpulan

-Interface didefinisikan dengan “Interface” keyword, mirip dengan deklarasi class biasa,

hanya saja definisi atau detail method tidak dituliskan.

-Seluruh method yang dideklarasikan pada interface harus memiliki modifier “public”

-Untuk mengimplementasikan sebuah interface, kita dapat menggunakan “implement” keyword.

-Seluruh method yang ada pada interface harus diimplementasikan seluruhnya. Sebuahclass bisa mengimplementasikan lebih dari satu interface.

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

-Interface bisa diwariskan seperti class menggunakan “extends”.

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

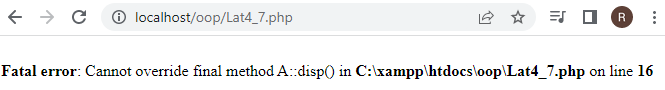
**Latihan 7:**

Tuliskan kode berikut pada lat4\_7.php

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | <?php class A  {  final public function disp(){ echo "Inside the final function";  }  }  class B extends A{ function disp(){  echo "Inside the final function";  }  }  $obj=new B();  $obj->disp();  ?> |

1. Bagaimana tampilan program di atas?Jelaskan mengapa hal tersebut terjadi?
2. 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?
3. Simpulkan apa yang anda peroleh dari latihan 7!

Jawab:

1. Error karena method pada class A tidak dapat di override
2. Tampilan karena tujuandigunakannya keyword final adalah agar class tidak akan di override.
3. Kesimpulan:

-final, akan mencegah proses overriding method pada class anak (sub-class)

-Apabila metode kita berikan status final, maka metode tersebut tidak akan bisadioverr

ide, begitu juga pada class, apabila kita berikan status “final” pada deklarasi class

maka class tersebut tidak bisa diperpanjang (diwariskan)

**Latihan 8:**

Tuliskan kode berikut pada lat4\_8.php

|  |  |
| --- | --- |
| 1  2 | <?php |

|  |  |
| --- | --- |
| 3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | class One{  private static $var=0;  function construct(){} static function disp(){  print self::$var;  }  function destruct(){}  }  One::disp();  ?> |

Simpulkan apa yang anda peroleh dari latihan 8!

Jawab:

a.Lat4\_8.php mengimplementasikan penggunaan property static dengan modifier private.Properti statis dideklarasikan dengan menggunakan kata kunci statis sebelum modifier

- Sintaks:

modifier static $nama\_property = nilai;

b.Sifat statis dapat diakses tanpa perlu sebuah contoh objek dari kelas, menggunakannama kelas bersama dengan ::- Sintaks:

ClassName :: $nama\_property method\_name();

Perhatikan bahwa properti statis menggunakan tanda dollar ($).Properti statis tidak dapat diakses melalui obyek menggunakan operator panah "->"

**Latihan 9:**

Dari contoh kode MVC pada penjelasan diatas buatlah :

1. Model berhubungan langsung dengan database mysql dengan sepesifikasi sebagai berikut :
   1. Database : Library
   2. Table : book Username dan password menyesuaikan.
2. Buatlah view menjadi sebuah object, susun class view dimana fungsinya adalah mengambil file-file template yang sudah disediakan!
3. Tuliskan masing-masing kodenya, buatlah simulasinya!