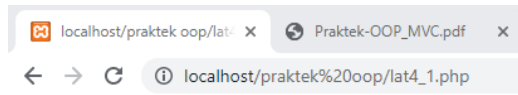


Nama : Salma maratuzzulfa

NIM : G.211.21.0100

Project Praktek OOP pada PHP File

Latihan 1: Class dan Object + Modifier



- a.
- b. Error, karena di php tidak bisa mengoverload method
- c. 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

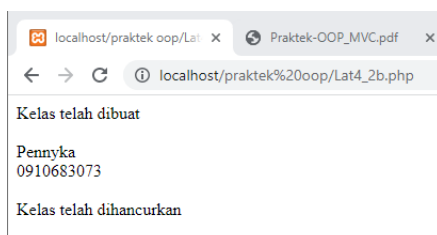
```
$nama_object->nama_methode();
```

Latihan 2

➔ Sebelum : Error

Fatal error: Cannot redeclare mahasiswa::__construct() in C:\xampp\htdocs\PT\Latihan_4\Lat4_2a.php on line 12

➔ Sesudah



Kesimpulan : Constructor dalam php oop tidak bisa di override

Latihan 3

- Error, karena property yang bermodifier private hanya bisa digunakan pada class mahasiswa sendiri.
- Protected tetap error, public hasilnya Hendra

```
Lat4_3b.php
1  <?php
2  require_once("lat4_3a.php");
3
4  $mhs1=new mahasiswa();
5  /*sebelum dimodifikasi
6  //$mhs1->nama="hendra";
7  //echo $mhs1->nama;
8  */
9  //setelah dimodifikasi
10 $mhs1->setNama ("Hendra");
11 echo $mhs->getNama ();
12
13 $mhs1->setNim ("nim mas hendra berapa?");
14 echo $mhs1->getNim();
15 ?>
```

-
-
-
- Kesimpulan :
 - Penggunaan modifier

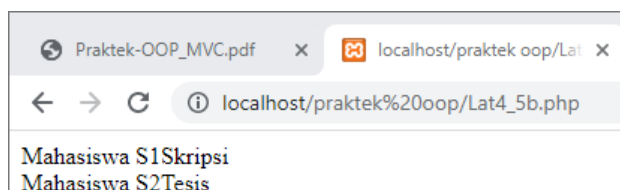
Modifier	Keterangan
Public	Untuk mendefinisikan data atau metode yang akan terlihat dari luar oleh siapapun dan dimanapun.
Private	Untuk mendefinisikan data atau metode agar hanya terlihat pada class/object itu sendiri.
Protected	Untuk mendefinisikan data atau metode untuk tidak terlihat dari luar (seperti private). tetapi akan dapat diakses oleh "anak" dari class tersebut.

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

Latihan 4

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

Latihan 5 : Abstract class



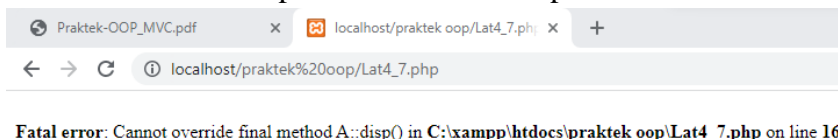
-
- Error, pada class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.
- Kesimpulan : class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.

Latihan 6

- a. Maksud dari program tersebut adalah penggunaan object interface
- b. Error, karena pada object interface, ketika kita mengimplementasikan object tersebut, seluruh method pada interface harus diimplementasikan seluruhnya. Karena class d mengimplement interface c, maka method-method pada interface c harus diimplementasikan seluruhnya.
- c. Error, method foo dan bar terdapat pada interace a dan b, pada penerapannya sebuah class baru tidak dapat mengimplementasikan method yang sama pada dua interface.
- d. 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. Sebuah class bisa mengimplementasikan lebih dari satu interface.
 - 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

- a. Error karena method pada class A tidak dapat di override



- b. Karena tujuan digunakannya keyword final adalah agar class tidak akan di override.
- c. Kesimpulan :
 - final, akan mencegah proses overriding method pada class anak (sub-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

Kesimpulan :

1. 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;
```

2. Sifat statis dapat diakses tanpa perlu sebuah contoh objek dari kelas, menggunakan nama 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 "->"

3. Untuk mengakses metode statis atau properti dari dalam kelas yang sama dapat menggunakan kata kunci self.

```
self :: $ properti;
```