

NAMA : Muhammad Annand Yunaz

NIM : G.211.21.0075

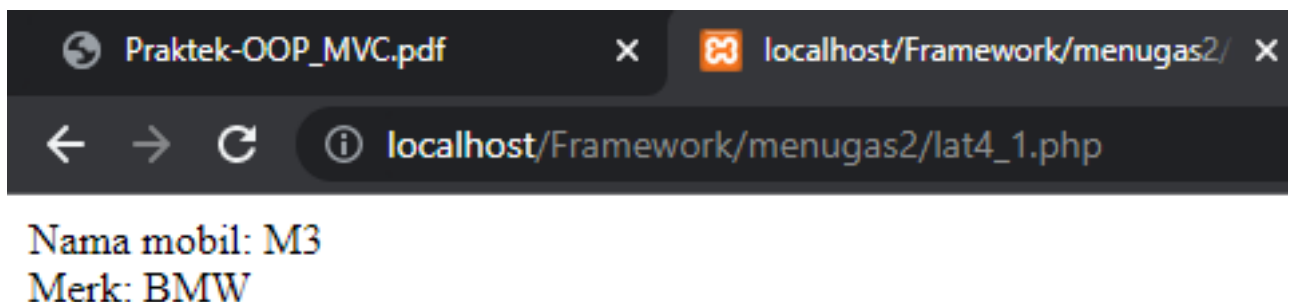
PEMROGRAMAN FRAMEWORK

LATIHAN 1 : Class dan Object + Modifier

```
Framework > menugas2 > lat4_1.php > ...
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="M3";
14 $ferari->merk="BMW";
15
16 $ferari->getInfo();
17
```

a) Bagaimana hasil tampilan diatas?

Jawaban :



Praktek-OOP_MVC.pdf x localhost/Framework/menugas2/ x

localhost/Framework/menugas2/lat4_1.php

Nama mobil: M3
Merk: BMW

b) Buatlah sebuah method overload get info dengan parameter \$a. Lalu jalankan dan amati perubahan yang terjadi.

Jawaban : Error, karena di php tidak bisa mengoverload method

c) Lalu simpulkan apa yang anda peroleh dari Latihan 1

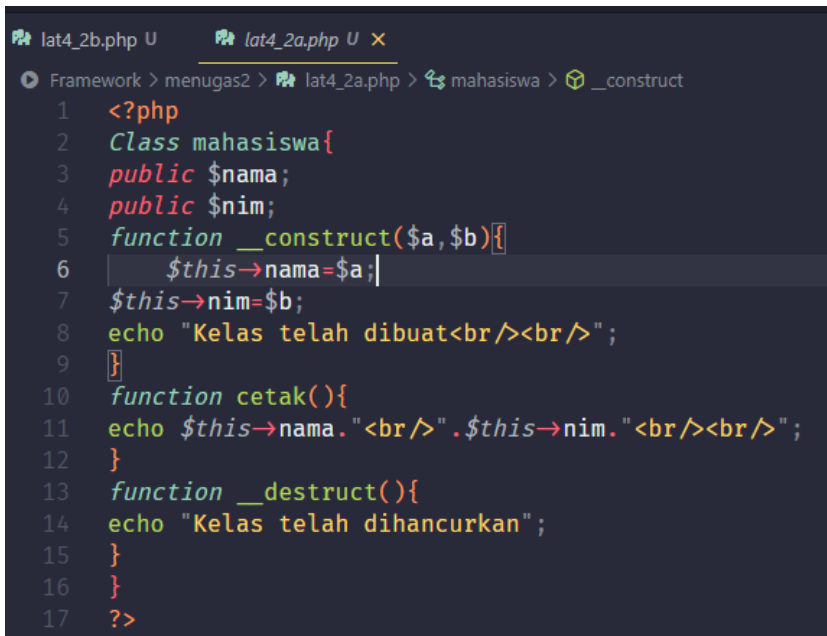
Kesimpulan :

1. Cara membuat class pada php
<?php
Class nama_class{
}
?>

2. Cara penulisan property
Modifier \$nama_properti;
3. Penulisan method
Modifer 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 :

Lat4_2a.php :

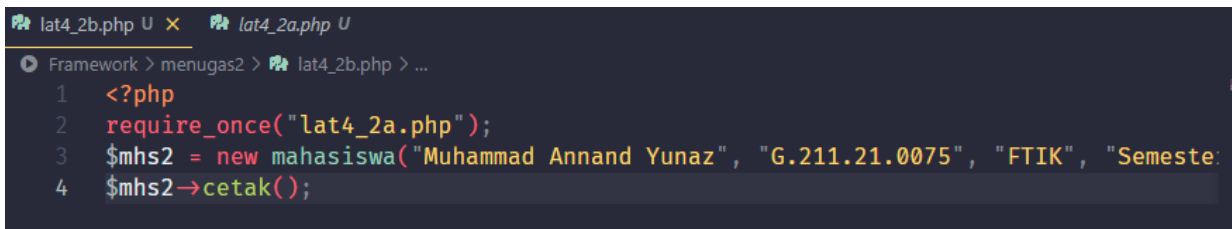


```

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 ?>

```

Lat4_2b.php



```

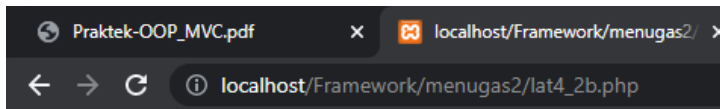
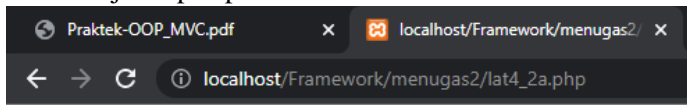
1  <?php
2  require_once("lat4_2a.php");
3  $mhs2 = new mahasiswa("Muhammad Annand Yunaz", "G.211.21.0075", "FTIK", "Semeste:
4  $mhs2->cetak();

```

Modifikasilah Latihan 4_2a dengan menambahkan 1 konstruktor lagi di lat4_2a baris 11. Lalu jalankan. Bagaimana hasil tampilan diatas sebelum dan sesudah dimodifikasi? Lalu simpulkan apa yang anda peroleh dari Latihan 2!

Jawaban :

- Tidak terjadi apa apa



Kelas telah dibuat

Muhammad Annand Yunaz
G.211.21.0075

Kelas telah dihancurkan

- Constructor dalam php oop tidak bisa di override

LATIHAN 3 :

- Lat4_3a

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

- Lat4_3b

```
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new mahasiswa();
4 $mhs1->nama = "Muhammad Annand Yunaz<br><br>";
5 $mhs1->nim = "G.211.21.0075";
6 echo $mhs1->nama;
7 echo $mhs1->nim;
8 ?>
```

Fatal error: Uncaught Error: Cannot access private property mahasiswa::\$nama in C:\xampp\htdocs\Framework\menugas2\lat4_3b.php:4 Stack trace: #0 {main} thrown in C:\xampp\htdocs\Framework\menugas2\lat4_3b.php on line 4

a) Apakah program eror? Jika eror mengapa hal itu dapat terjadi?

Jawaban : Error, karena property yang bermodifer private hanya bisa digunakan pada class mahasiswa sendiri.

- b) Rubahlah modifier dari variable nama dan nim menjadi protected dan public, lalu amati perubahan yang terjadi

Jawaban : pada saat modifier menjadi public lalu dijalankan tidak akan terjadi error

Tetapi jika modifier protected dijalankan akan terjadi error

- c) Modifikasilah Lat4_3b sehingga dapat memberi dan mencetak isi dari nim dan nama dengan modifier private

Jawaban :

```
lat4_3a.php U x lat4_3b.php U
Framework > menugas2 > lat4_3a.php > mahasiswa > getNim
1 <?php
2 class mahasiswa
3 {
4     private $nama;
5     private $nim;
6     function __construct(
7         $a = "Muhammad Annand Yunaz <br>",
8         $b = "G.211.21.0075"
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 }
```

```
lat4_3b.php - Untitled (Workspace) - Visual Studio Code
lat4_3a.php U lat4_3b.php U x
Framework > menugas2 > lat4_3b.php > ...
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new Mahasiswa();
4 echo $mhs1->getNama();
5 echo $mhs1->getNim();
6
```

Praktek-OOP_MVC.pdf
localhost/Praktek-OOP_MVC.pdf

Muhammad Annand Yunaz
G.211.21.0075

- d) Kesimpulan :
- modifier protected dan private, properties bisa dipanggil dengan mengimplementasikan setter-getter

LATIHAN 4 :

- Lat4_4a.php

```
lat4_4b.php U  lat4_4a.php U X
Framework > menugas2 > lat4_4a.php > ...
1  <?php
2  require_once("Lat4_3a.php");
3  class asisten extends mahasiswa
4  {
5      public $nama;
6      function __construct()
7      {
8      }
9      function setNama($a)
10     {
11         $this->nama = $a;
12     }
13     function getNama()
14     {
15         return $this->nama;
16     }
17     function destruct()
18     {
19     }
20 }
21
```

- Lat4_4b.php

```
lat4_4b.php U X  lat4_4a.php U
Framework > menugas2 > lat4_4b.php > ...
1  <?php
2  require_once("lat4_4a.php");
3  $as = new asisten();
4  $as->setNama("hai");
5  echo $as->getNama();
6
```

Simpulkan apa yang anda peroleh dari Latihan 4 :

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

LATIHAN 5 : Abstract Class

- Lat4_5a.php

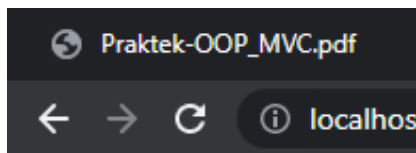
```
lat4_5a.php U lat4_5b.php U
Framework > menugas2 > lat4_5a.php > ...
1 <?php
2 abstract class mahasiswa
3 {
4     abstract protected function getTugasAkhir();
5     abstract protected function getProgram($postfix);
6     public function tugasAkhir()
7     {
8         print $this->getTugasAkhir() . "<br>";
9     }
10 }
11 class sarjana extends mahasiswa
12 {
13     protected function getTugasAkhir()
14     {
15         return "Skripsi";
16     }
17     public function getProgram($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)
```

- Lat4_5b.php

```
lat4_5a.php U lat4_5b.php U
Framework > menugas2 > lat4_5b.php > ...
1 <?php
2 require_once("lat4_5a.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();
9
```

a) Bagaimana hasil tampilan dari program di atas?

Jawaban :



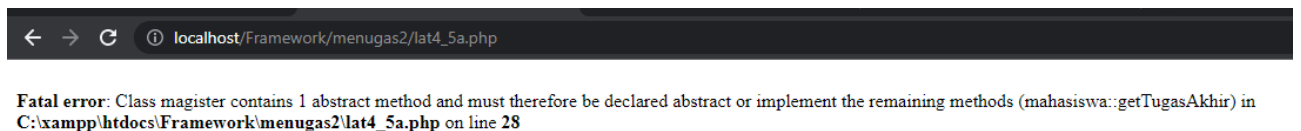
Mahasiswa S1Skripsi
Mahasiswa S2Tesis

- b) Hapuslah kode baris 29-32 pada lat4_5a.php. bagaimana hasil tampilan program diatas?
Jelaskan mengapa hal tersebut terjadi?

Jawaban :



```
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 "{$postfix} S1";
20     }
21 }
22 class magister extends mahasiswa
23 {
24     public function getProgram($postfix)
25     {
26         print "{$postfix} S2";
27     }
28 }
```



← → ↻ ⓘ localhost/Framework/menugas2/lat4_5a.php

Fatal error: Class magister contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (mahasiswa::getTugasAkhir) in C:\xampp\htdocs\Framework\menugas2\lat4_5a.php on line 28

Error, karena method `getTugasAkhir()` tidak ada di turunannya/subclass

- c) Simpulkan apa yang anda peroleh pada Latihan 5

Jawaban : bahwa class dan method dengan keyword abstract hanya bisa diakses melalui turunannya saja.

LATIHAN 6 :

```
lat4_5a.php U lat4_5b.php U lat4_6.php U x
Framework > menugas2 > lat4_6.php > ...
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 }
```

a) Jelaskan maksud dari program diatas?

Jawaban : maksud dari program tersebut adalah penggunaan object interface

b) Hapuslah kode baris 27-29, lalu jalankan. Bagaimana tampilan program diatas?
Jelaskan mengapa hal tersebut dapat terjadi?

Jawban : eror karena class d mengimplement interface c, maka method method pada interface c harus di implementasikan seluruhnya

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

Jawaban :

```
26 class e implements a, b
27 {
28     public function foo()
29     {
30     }
31     public function bar()
32     {
33     }
34 }
```

d) Simpulkan apa yang anda peroleh di Latihan 6

Jawaban : jadi sebuah interface dengan method abstract harus mempunyai class dengan implements yang sama dengan parentnya. Sebuah interface hanya bisa diisi dengan method abstract saja dan memiliki visibility public.

LATIHAN 7 :

Lat4_7.php

```
Framework > menugas2 > lat4_7.php > ...
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 diatas? Jelaskan mengapa hal tersebut bisa terjadi?

Jawaban :

Fatal error: Cannot override final method A::disp() in C:\xampp\htdocs\Framework\menugas2\lat4_7.php on line 9

Error karena method pada class A tidak dapat di override

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

Jawaban :

Fatal error: Class B may not inherit from final class (A) in C:\xampp\htdocs\Framework\menugas2\lat4_7.php on line 12

Karena tujuan digunakannya keyword final adalah agar class tidak akan di override.

- c) Simpulkan apa yang anda peroleh dari Latihan 7

Jawaban : 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 :

```
1 <?php
2 class One
3 {
4     private static $var = 23;
5     function __construct()
6     {
7     }
8     static function disp()
9     {
10         print self::$var;
11     }
12     function __destruct()
13     {
14     }
15 }
16 One::disp();
17
```



23

Simpulkan apa yang diperoleh dari Latihan 8 :

Jawaban : Jadi keyword static digunakan untuk mencetak method tanpa melakukan instansiasi, yaitu dengan memanggil class nya kemudian titik dua, dua kali lalu method nya atau properti yang ingin dipanggil.