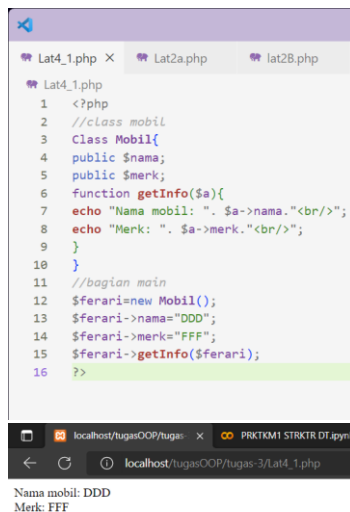


# Tugas Framework Project OOP pada PHP

Husain Hikmal Ali A

## ● Latihan 1

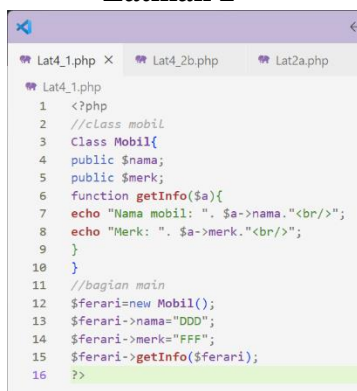


```
1 <?php
2 //class mobil
3 class Mobil{
4     public $nama;
5     public $merk;
6     function getInfo($a){
7         echo "Nama mobil: ". $a->nama."<br/>";
8         echo "Merk: ". $a->merk."<br/>";
9     }
10 }
11 //bagian main
12 $ferari=new Mobil();
13 $ferari->nama="DDD";
14 $ferari->merk="FFF";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


Nama mobil: DDD  
Merk: FFF

1. Membuat sebuah 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



```
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 ?>
```



```
1 <?php
2
3 require_once ("Lat4_2a.php");
4 $mhs2 = new mahasiswa("Husain", "G.211.21.0065");
5 $mhs2->cetak();
6
```

## After

```
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 __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 ?>
```

Fatal error: Cannot redeclare mahasiswa::\_\_construct() in C:\xampp\htdocs\TugasOOP\tugas-3\Lat4\_2a.php on line 10

Terjadi error karena memiliki tidak dapat memanggil function sama yang terdefinisi dalam satu file yang sama

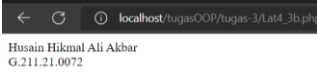
### • Latihan 3

```
Lat4_3a.php
1 <?php
2 class mahasiswa{
3     private $nama;
4     private $nim;
5     public function tampilkan_nama(){
6         return $this->nama;
7     }
8     function __construct(){
9     }
10    function setNama($a){
11        $this->nama = $a;
12    }
13    function setNim($b){
14        $this->nim = $b;
15    }
16
17    function getNama(){
18        return $this->nama;
19    }
20
21    function getNim(){
22        return $this->nim;
23    }
24
25    function __destruct(){
26    }
27 }
```

```
Lat4_3b.php
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new mahasiswa();
4 $mhs1->nama = "Husain";
5 $mhs1->nim = "G.211.21.0065";
6 echo $mhs1->nama . "<br/>";
7 echo $mhs1->nim;
8
```

## After

```
Lat4_3a.php
1 <?php
2 class mahasiswa
3 {
4     private $nama;
5     private $nim;
6     function __construct()
7     {
8         $a = "Hussain Hikmal Ali Akbar <br>",
9         $b = "G.211.21.0072"
10    }
11    {
12        $this->nama = $a;
13        $this->nim = $b;
14    }
15    function setNama($a)
16    {
17        $this->nama = $a;
18    }
19    function setNim($b)
20    {
21        $this->nim = $b;
22    }
23    function getNama()
24    {
25        return $this->nama;
26    }
27    function getNim()
28    {
29        return $this->nim;
30    }
31 }
```



## ● Latihan 4

```
Lat4_4a.php
1 <?php
2 require_once("lat4_3a.php");
3 class asisten extends mahasiswa
4 {
5     function __construct()
6     {
7     }
8 }
9
10 // Pada php oop, class asisten (child) bisa memanggil method dari mahasiswa (parent)
11
12 |
```

```
Lat4_4b.php
1 <?php
2 require_once("lat4_4a.php");
3 $as = new asisten();
4 $as->setNama("tes");
5 echo $as->getNama();
6
7 // Pada php oop, class asisten (child) bisa memanggil method dari mahasiswa (parent)
8
```

## ● Latihan 5

```
Lat4_5a.php
1  <?php
2
3  abstract class mahasiswa
4  {
5      abstract protected function getTugasAkhir();
6      abstract protected function getProgram($postfix);
7
8      public function tugasAkhir()
9      {
10         print $this->getTugasAkhir() . "<br>";
11     }
12 }
13
14 class sarjana extends mahasiswa
15 {
16     protected function getTugasAkhir()
17     {
18         return " Skripsi";
19     }
20
21     public function getProgram($postfix)
22     {
23         print "{$postfix} S1";
24     }
25 }
26
27 class magister extends mahasiswa
28 {
29
30
31     public function getProgram($postfix)
32     {
33         print "{$postfix} S2";
34     }
35 }
```

```
Lat4_5b.php
1  <?php
2  require_once("lat4_5a.php");
3  $s = new sarjana;
4  $s->getProgram('Mahasiswa') . "<br>";
5  $s->tugasAkhir();
6
7  $m = new magister;
8  $m->getProgram('Mahasiswa') . "<br>";
9  $m->tugasAkhir();
10
11 // Error, pada class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.
12 // Kesimpulan:
13 // class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.
14 // Sek bingung
15
```

## ● Latihan 6

```
Lat4_6.php
1  <?php
2  interface a
3  {
4      public function foo();
5  }
6
7  interface b
8  {
9      public function bar();
10 }
11
12 interface c extends a, b
13 {
14     public function baz();
15 }
16
17 class d implements c
18 {
19     public function foo()
20     {
21     }
22
23     public function bar()
24     {
25     }
26 }
27
28
29
30 class e implements d
31 {
32     public function foo()
33     {
34     }
35 }
```

```
Lat4_6.php
28
29
30 class e implements d
31 {
32     public function foo()
33     {
34     }
35     public function bar()
36     {
37     }
38 }
39
40 // Error, karena pada object interface, ketika kita mengimplementasikan object tersebut, seluruh method pada
// interface harus diimplementasikan seluruhnya. Karena class d mengimplementinterface c, maka method-method pada
// interface c harus diimplementasikan seluruhnya. Error, method foo dan bar terdapat pada interace a dan b, pada
// penerapannya sebuah classbaru tidak dapat mengimplementasikan method yang sama pada dua interface.
41
```

## ● Latihan 7

```
Lat4_7.php
1  <?php
2  class A
3  {
4      public function disp()
5      {
6          echo "Inside the final function";
7      }
8  }
9
10 final class B extends A
11 {
12     function disp()
13     {
14         echo "Inside the final function";
15     }
16 }
17
18 $obj = new B();
19
20 $obj->disp();
21 ?>
22
```

PRKTKM1 STRKTR DT.pyrb - Co X | Praktikum 4 PHP OOP da  
localhost/tugasOOP/tugas-3/Lat4\_7.php

Inside the final function

## • Latihan 8

```

Lat4_7.php Lat4_8.php •
Lat4_8.php
1 <?php
2 class One
3 {
4
5     private static $var = 0;
6
7     function __construct()
8     {
9     }
10    static function disp()
11    {
12        print self::$var;
13    }
14
15    function __destruct()
16    {
17    }
18 }
19
20 One::disp();
21
22 // Static property dan static method adalah property dan method yang langsung dapat
23 // diakses dari class tanpa instansiasi class (pembuatan object) terlebih dahulu. Cara
24 // mendeklarasinya yaitu property atau method diawali menggunakan keyword static

```

## • Latihan 9

```

index.php X
tugas_9 > index.php
1 <?
2 include_once 'controller.php';
3
4 $main_ctrl = new controller();
5 $main_ctrl->__invoke();
6 ?>

```

```

PRKTKM1 STRKTR DT.ipynb - Co X | Praktikum 4 PHP OOP dan MVC X
localhost/tugasOOP/tugas-3/tugas_9/
__invoke(); ?>

```

## After

```

index.php X
tugas_9 > index.php
1 <?php
2 include_once 'controller.php';
3
4 $main_ctrl = new controller();
5 $main_ctrl->__invoke();
6 ?>

```

JUDUL	PENGARANG	PENERBIT	TAHUN
Penrograman PHP & MySql	Bayu Priyambadha	UB Press	2011
Penrograman Framework MVC dengan PHP	Widhy	UB Press	2011
Memangun Aplikasi Web dengan PHP dan AJAX	Achmad Arwan	UB Press	2012
Kolaborasi PHP, MVC dan AJAX	Bayu Priyambadha	UB Press	2012