

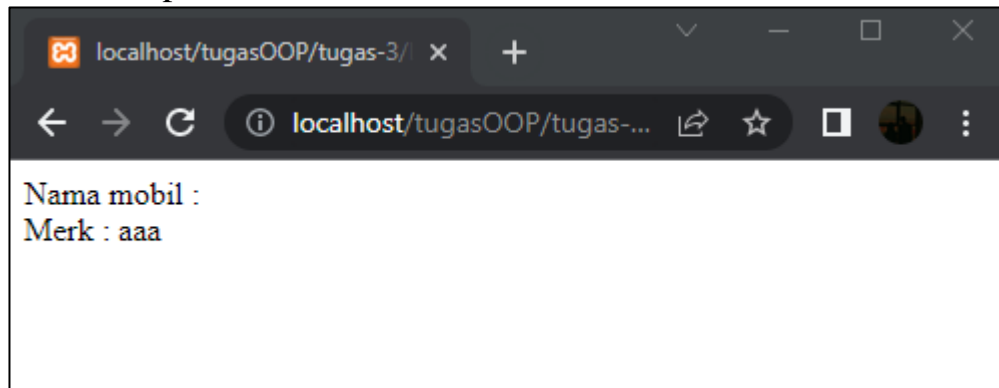
## TUGAS OOP & MVC

Nama : Ahmad Miftahul Ulum

NIM : G.211.21.0095

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

- a. Hasil Tampilan.



- b. Buatlah sebuah method overload getInfo dengan parameter \$a. Lalu jalankan dan amati perubahan yang terjadi.

```
<?php
class mobil
{
    public $nama, $merk, $a;

    function getInfo() Duplicate symbol declaration 'getInfo'.
    {
        echo "Nama mobil : " . $this->nama . "<br/>";
        echo "Merk : " . $this->merk . "<br/>";
    }

    function getInfo($a){ Duplicate symbol declaration 'getInfo'.
    $this->nama = $a;
    }
}

$ferari = new Mobil();
$ferari->a = "xxx";
$ferari->merk = "aaa";
$ferari->getInfo();
```

Program error, karena PHP tidak mendukung overload method.

- c. Simpulan.

Pada latihan 1 kita dapat pemahaman tentang cara membuat class, property, method, inisiasi objek, pemanggilan method, dan mendefinisikan objek.

- **Latihan 2**

- a. Menambahkan 1 konstruktor lagi

```

<?php
class mahasiswa{
    public $nama, $nim;

    function __construct($a, $b){ Duplicate symbol declaration '__construct'.
        $this->nama = $a;
        $this->nim = $b;
        echo " Kelas telah dibuat<br/><br/>";
    }

    function __construct($c, $d){ Duplicate symbol declaration '__construct'.
        $this->a = $c;
        $this->b = $d;
    }

    function cetak(){
        echo $this->nama . "<br/>" . $this->nim . "<br/><br/>";
    }

    function __destruct(){
        echo "Kelas telah dihancurkan";
    }
}

```

Program error, karena PHP tidak mendukung overload method construct.

b. Simpulan

Pada latihan 2 kita dapat pemahaman tentang penggunaan construct dan destruct.

### • Latihan 3

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

```

<?php

class mahasiswa{
    private $nama, $nim;
    function __construct(){}
    function setName($a){
        $this->nama = $a;
    }

    function setNim($b){
        $this->nim = $b;
    }

    function getName(){
        return $this->nama;
    }

    function getNim(){
        return $this->nim;
    }

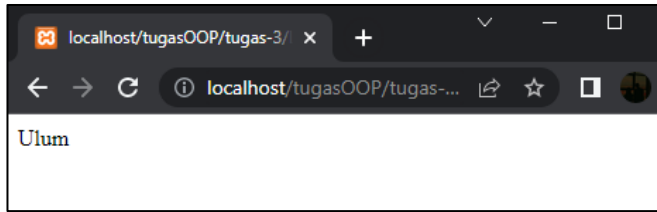
    function __destruct(){}
}

```

Program error karena visibility private tidak bisa dipanggil diluar suatu class.

b. Modifier dari variable nama dan nim menjadi protected dan public.

Jika visibility protector, program tetap error, sedangkan jika visibility public bakal tampil



c. Simpulan.

Pada tugas 3 kita dapat memahami penggunaan visibility public, private, dan protected

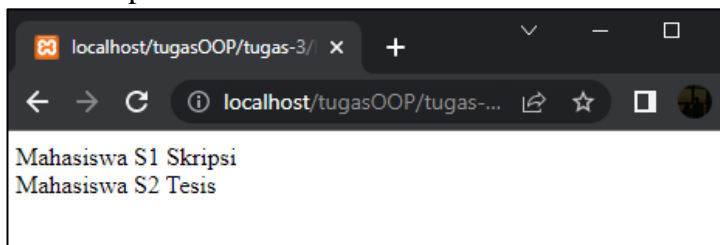
- **Latihan 4**

a. Simpulan

Setiap class child dapat memanggil method dari kelas parentnya.

- **Latihan 5: Abstract Class**

a. Hasil tampilan.



b. 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?

```
class magister extends mahasiswa{
    // public function getTugasAkhir(){
    //     return "Tesis";
    // }
}
```

'magister' does not implement method 'getTugasAkhir'

Program error karena setiap child class harus menuliskan semua method abstrak yang ada pada parent classnya.

c. Simpulan.

Setiap child class harus menuliskan semua method yang ada pada parent classnya.

- **Latihan 6**

a. Maksud program.

Penggunaan object interface.

b. Hapuslah kode baris 27 – 29, lalu jalankan lat4\_6.php. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut dapat terjadi?

```
class d implements c{
    public function foo(){
    public function bar(){
    // public function baz(){
    }
}
```

'd' does not implement method 'baz'

Program error karena class d implementasi dari class c maka semua method class c harus di tuliskan semuanya.

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

Program Erro karena sebuah class tidak bisa mengimplementasikan dua interface pada kelas berbeda.

d. Simpulan.

Pada tugas ini kita mengetahui cara penggunaan interface.

- **Latihan 7**

a. tampilan program

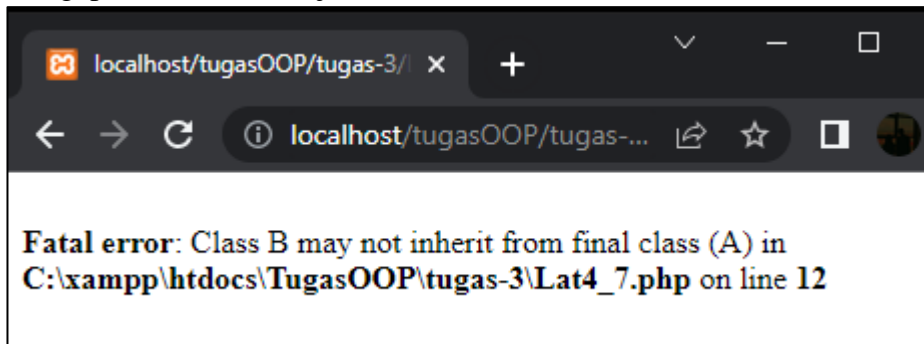
```
<?php
class A{
    final public function disp(){
        echo "Inside the final function";
    }
}

class B extends A{
    function disp(){
        Method 'B::disp()' cannot override final method 'A::disp()'.
        echo "Inside the final function";
    }
}

$obj = new B();
$obj->disp();
```

Proram error karena methodnya teroverload.

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?



Program Error karena keyword final mencegah adanya overload method pada child class.

c. Simpulan.

karena keyword final mencegah adanya overload method pada child class.

- **Latihan 8**

a. Simpulan.

Pada latihan ini kita mengimplementasikan penggunaan property static, Sifat static dapat diakses tanpa adanya pemanggilan objek dari kelas.

- **Latihan 9**

a. book.php

```

<?php
class book
{
    public $judul;
    public $pengarang;
    public $penerbit;
    public $tahun;

    public function __construct($title, $author, $publisher, $year)
    {
        $this->judul = $title;
        $this->pengarang = $author;
        $this->penerbit = $publisher;
        $this->tahun = $year;
    }
}

```

b. controller.php

```

<?php
include_once 'model.php';
class controller
{
    function oke()
    {
        $model_data = new model();
        $row_data = $model_data->getData();
        $data = "<table border=1>
        <tr>
            <th>JUDUL</th>
            <th>PENGARANG</th>
            <th>PENERBIT</th>
            <th>TAHUN</th>
        </tr>";

        foreach ($row_data as $key => $value) {
            $data .=
            "<tr>
                <td>" . $value->judul . "</td>
                <td>" . $value->pengarang . "</td>
                <td>" . $value->penerbit . "</td>
                <td>" . $value->tahun . "</td>
            </tr>";
        }
        $data .= "</table>";
        include('view.php');
    }
}

```

c. index.php

```
<?php
include_once 'controller.php';

$main_ctrl = new controller();
$main_ctrl->oke();
```

d. model.php

```
<?php
include_once 'book.php';
class model
{
    public function getData()
    {
        return array(
            new book('Pemrograman PHP & MySql', 'Bayu Priyambadha', 'UB Press', '2011'),
            new book('Pemrograman Framework MVC dengan PHP', 'Widhy', 'UB Press', '2011'),
            new book('Membangun Aplikasi Web dengan PHP dan AJAX', 'Achmad Arwan', 'UB Press', '2012'),
            new book('Kolaborasi PHP, MVC dan AJAX', 'Bayu Priyambadha', 'UB Press', '2012')
        );
    }
}
```

e. view.php

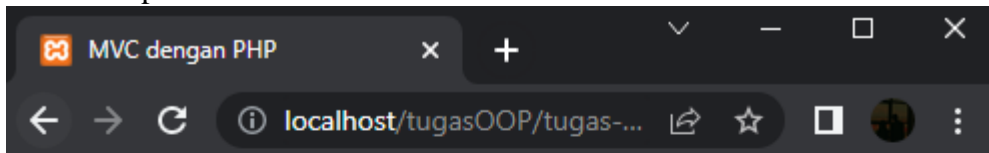
```
<html>
<head>
<title>MVC dengan PHP</title>
</head>

<body>
<center>

<?= $data ?>

</center>
</body>
</html>
```

f. Hasil Tampilan



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