

PRAKTIKUM PEMROGRAMAN FRAMEWORK

PHP OOP & MVC

Nama : Muhammad Rafif Abidin

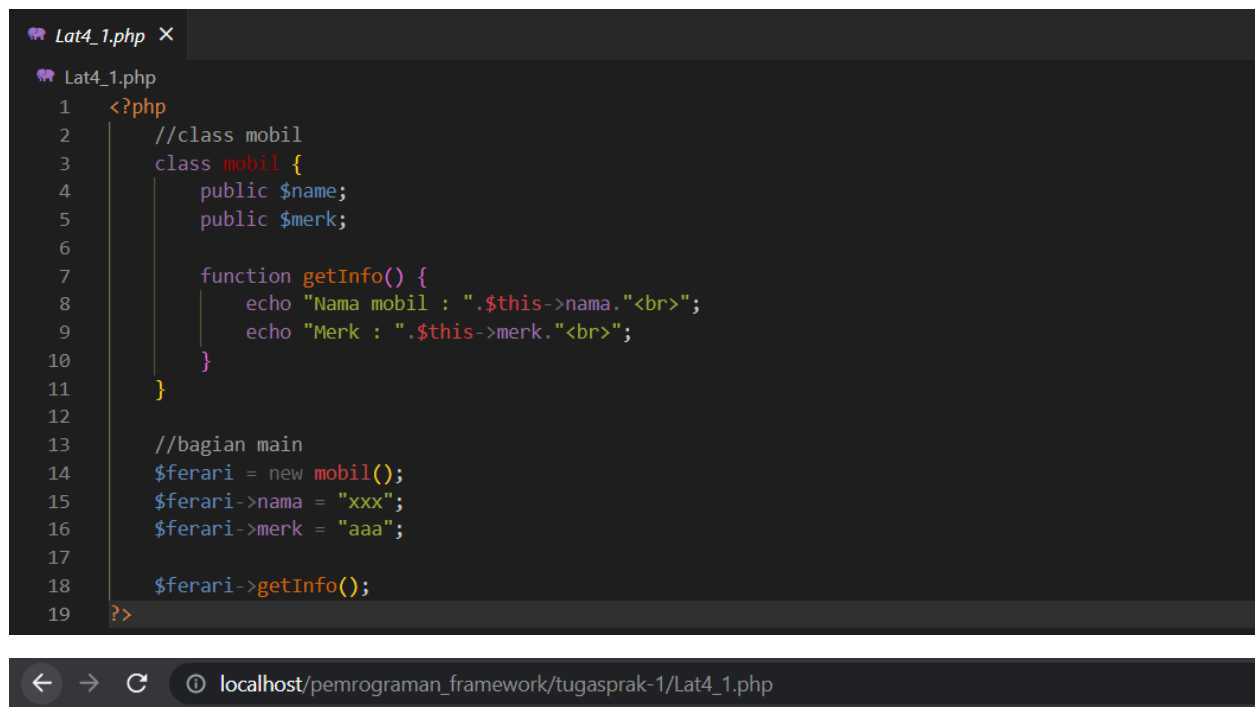
NIM : G.231.21.0176

Link : <https://github.com/ParkRaPep/Tugas-Praktikum-OOP-MVC>

Latihan 1 : Class dan Object + Modifier

- a. Bagaimana hasil tampilan di atas?

Jawaban :



The screenshot shows a code editor with a file named 'Lat4_1.php'. The code defines a class 'mobil' with two public properties, '\$name' and '\$merk', and a public method 'getInfo()'. The method 'getInfo()' uses 'echo' to display the values of '\$name' and '\$merk' separated by a line break. In the main section, an object '\$ferari' of the 'mobil' class is instantiated, its '\$name' property is set to 'xxx' and its '\$merk' property is set to 'aaa'. Finally, the 'getInfo()' method is called on the '\$ferari' object. Below the code editor, a browser window shows the output of the script: 'Nama mobil : xxx' followed by 'Merk : aaa' on a new line.

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

localhost/pemrograman_framework/tugasprak-1/Lat4_1.php

Nama mobil : xxx
Merk : aaa

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

Jawaban :

```
Lat4_2a.php Lat4_1_after.php X Lat4_2b.php
Lat4_1_after.php
1  <?php
2      //class mobil
3      class mobil {
4          public $name;
5          public $merk;
6
7          function getWarna() {
8              return $this->warna = $a;
9          }
10
11         function getInfo() {
12             echo "Nama mobil : ".$this->nama."<br>";
13             echo "Merk : ".$this->merk."<br>";
14             echo "Warna : ".$this->warna."<br>";
15         }
16     }
17
18     //bagian main
19     $ferari = new mobil();
20     $ferari->nama = "xxx";
21     $ferari->merk = "aaa";
22     $ferari->warna = "merah";
23
24     $ferari->getInfo();
25 }
```

← → ↻ localhost:8080/pemrograman_framework/tugasprak-1/Lat4_1_after.php

Nama mobil : xxx
Merk : aaa
Warna : merah

c. Lalu simpulkan apa yang anda peroleh dari Latihan 1!

Jawaban :

Dengan menambahkan method function `getWarna()` dengan parameter `$a`, operasi masih bisa dijalankan.

Latihan 2 :

```
Lat4_2a.php ×
Lat4_2a.php
1  <?php
2      class mahasiswa {
3          public $nama;
4          public $nim;
5
6          function __construct($a, $b) {
7              $this->nama = $a;
8              $this->nim = $b;
9              echo "Kelas telah dibuat<br/><br/>";
10         }
11
12         function cetak() {
13             echo $this->nama."<br/>".$this->nim."<br/><br/>";
14         }
15         function __destruct() {
16             echo "Kelas telah dihancurkan";
17         }
18     }
19 ?>
```

```
Lat4_2b.php ×
Lat4_2b.php
1  <?php
2      require_once("Lat4_2a.php");
3
4      $mhs2 = new mahasiswa ("Pennyka", "0910683073");
5
6      $mhs2->cetak();
7  ?>
```

Modifikasilah Latihan 4_2a dengan menambah 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!

Jawaban :

```
Lat4_2a.php 1 x Lat4_2b.php
Lat4_2a.php
1  <?php
2      class mahasiswa {
3          public $nama;
4          public $nim;
5
6          function __construct($a, $b) {
7              $this->nama = $a;
8              $this->nim = $b;
9              echo "Kelas telah dibuat<br/><br/>";
10         }
11         function construct() {
12
13         }
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  ?>
```

← → ↻ localhost:8080/pemrograman_framework/tugasprak-1/Lat4_2b.php

Fatal error: Cannot redeclare mahasiswa::__construct() in C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_2a.php on line 11

Saat dijalankan terjadi error, karena operasi tidak dapat dijalankan dua constructor dalam satu kelas.

Latihan 3

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

```
Lat4_3b.php x
Lat4_3b.php
1  <?php
2      require_once("Lat4_3a.php");
3
4      $mhs1 = new mahasiswa();
5      $mhs1->nama = "hendra";
6      echo $mhs1->nama;
7  ?>
```

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

Jawaban :

```
<  →  ↻  localhost:8080/pemrograman_framework/tugasprak-1/Lat4_3b.php

Fatal error: Uncaught Error: Cannot access private property mahasiswa::$nama in C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_3b.php:5 Stack trace: #0 {main} thrown in C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_3b.php on line 5
```

- b. Rubahlah modifier dari variable nama dan menjadi protected dan public. Lalu amati perubahan yang terjadi.

Jawaban :

```
<  →  ↻  localhost:8080/pemrograman_framework/tugasprak-1/Lat4_3b.php

hendra
```

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

Jawaban :

```
Lat4_3a.php Lat4_3b.php Lat4_3a_after.php X Lat4_3b_after.php
Lat4_3a_after.php
1  <?php
2      class mahasiswa {
3          private $nama;
4          private $nim;
5
6          function __construct($nama, $nim) {
7              $this->nama = $nama;
8              $this->nim = $nim;
9          }
10         function getData() {
11             return "$this->nama $this->nim";
12         }
13         function __destruct() {}
14     }
15  ?>
```

```
Lat4_3a.php Lat4_3b.php Lat4_3a_after.php Lat4_3b_after.php X
Lat4_3b_after.php
1  <?php
2      require_once("Lat4_3a_after.php");
3
4      $mhs1 = new mahasiswa('Hendra', 'G.231.21.0176');
5      echo $mhs1->getData();
6  ?>
```

```
← → ↻ ⓘ localhost:8080/pemrograman_framework/tugasprak-1/Lat4_3b_after.php
```

Hendra G.231.21.0176

- d. Simpulkan apa yang anda peroleh dari Latihan 3!

Apabila sebuah property atau method diset sebagai private, maka satu-satunya yang dapat mengakses adalah class itu sendiri, berbeda dengan public yang mana dapat diakses dari mana saja bahkan di luar kelas.

Latihan 4

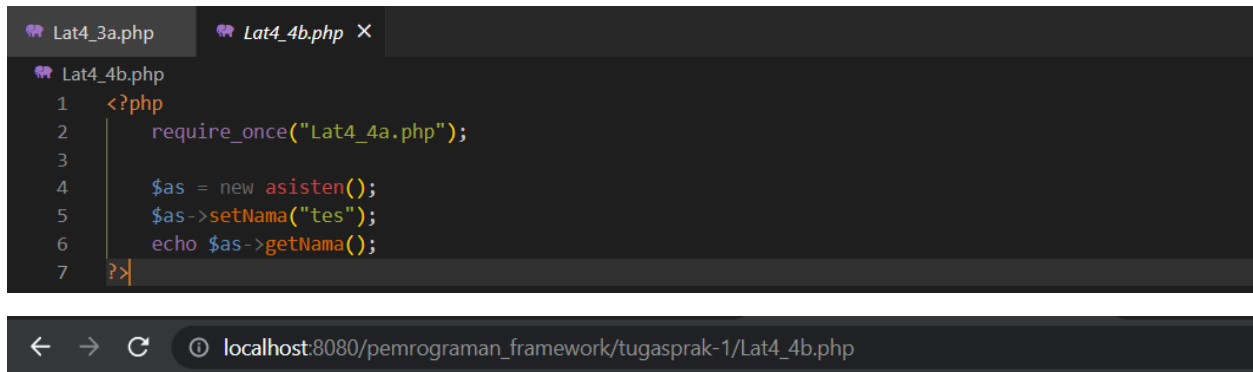
Simpulkan apa yang anda peroleh dari Latihan 4!

Jawaban :



```
Lat4_3a.php Lat4_4a.php X
Lat4_4a.php
1 <?php
2     require_once("Lat4_3a.php");
3
4     class asisten extends mahasiswa {
5     |     function __construct() {}
6     |     }
7 ?>
```

← → ↻ localhost:8080/pemrograman_framework/tugasprak-1/Lat4_4a.php



```
Lat4_3a.php Lat4_4b.php X
Lat4_4b.php
1 <?php
2     require_once("Lat4_4a.php");
3
4     $as = new asisten();
5     $as->setNama("tes");
6     echo $as->getNama();
7 ?>
```

← → ↻ localhost:8080/pemrograman_framework/tugasprak-1/Lat4_4b.php

tes

Pada modifier private bisa digunakan pada child class dengan metode setter dan getter.

Latihan 5

a. Bagaimana hasil tampilan dari program di atas?

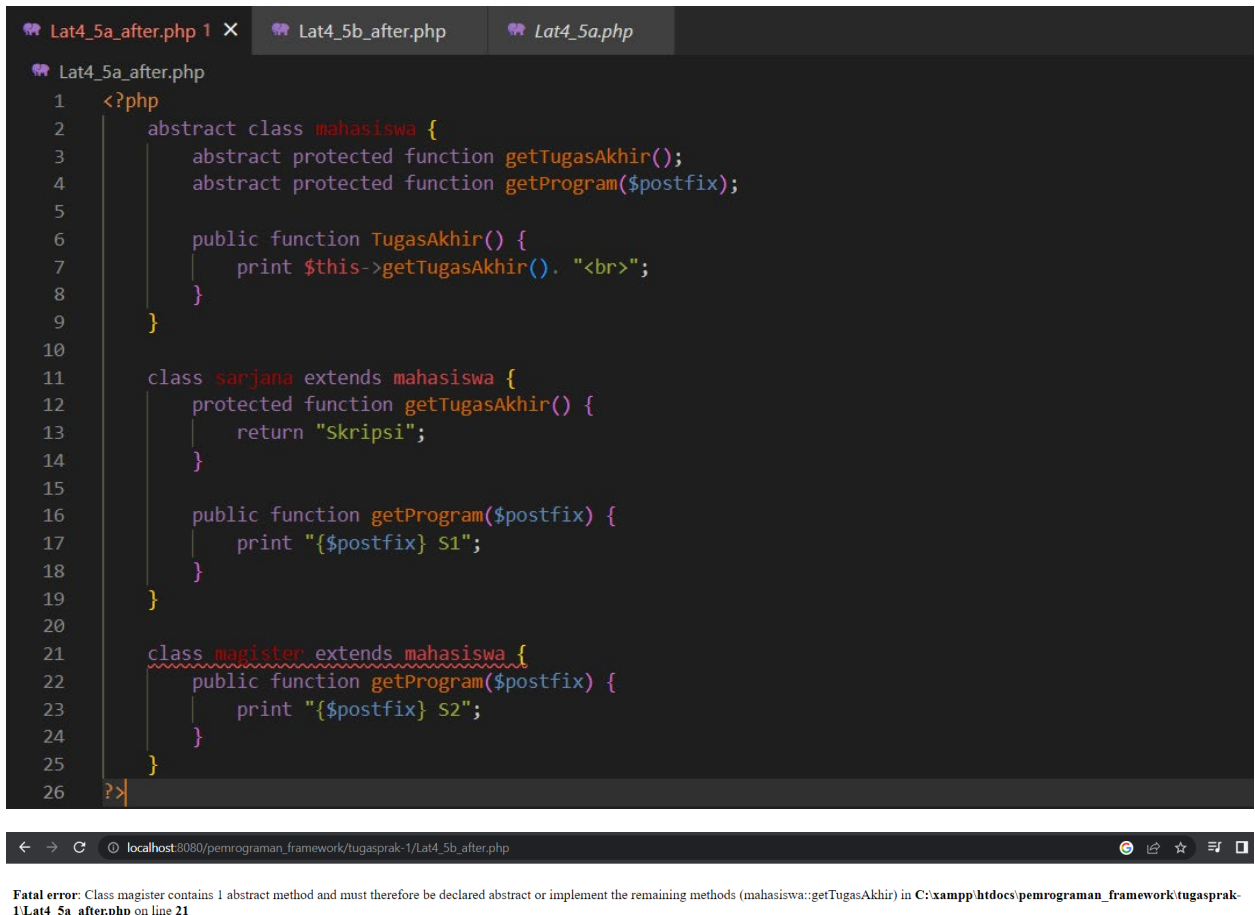
```
Lat4_5a.php X
Lat4_5a.php
1  <?php
2      abstract class mahasiswa {
3          abstract protected function getTugasAkhir();
4          abstract protected function getProgram($postfix);
5
6          public function TugasAkhir() {
7              print $this->getTugasAkhir(). "<br>";
8          }
9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

Mahasiswa S1Skripsi

Mahasiswa S2

Fatal error: Uncaught TypeError: Unsupported operand types: null / string in C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_5b.php:9 Stack trace: #0 {main} thrown in C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_5b.php on line 9

- 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?



```
Lat4_5a_after.php 1 X Lat4_5b_after.php Lat4_5a.php
Lat4_5a_after.php
1 <?php
2     abstract class mahasiswa {
3         abstract protected function getTugasAkhir();
4         abstract protected function getProgram($postfix);
5
6         public function TugasAkhir() {
7             print $this->getTugasAkhir(). "<br>";
8         }
9     }
10
11     class sarjana extends mahasiswa {
12         protected function getTugasAkhir() {
13             return "Skripsi";
14         }
15
16         public function getProgram($postfix) {
17             print "{$postfix} S1";
18         }
19     }
20
21     class magister extends mahasiswa {
22         public function getProgram($postfix) {
23             print "{$postfix} S2";
24         }
25     }
26 }>
```

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\pemrograman_framework\tugasprak-1\Lat4_5a_after.php on line 21

Hal ini terjadi karena class magister berisi satu method abstrak dan oleh karena itu harus dinyatakan atau diimplementasikan dengan method yang tersisa, yakni `getTugasAkhir()`.

- c. Simpulkan apa yang anda peroleh pada Latihan 5!

Apabila sebuah class memiliki sejumlah method abstrak, maka pada saat ingin meng-*extend* class/ membuat interface, method yang akan kita operasikan harus sesuai dengan abstract class.

Latihan 6


```
Lat4_6.php x
Lat4_6.php
1  <?php
2      interface a {
3          |   public function foo();
4      }
5
6      interface b {
7          |   public function bar();
8      }
9
10     interface c extends a, b {
11         |   public function baz();
12     }
13
14     class d implements c {
15         |   public function foo() {}
16         |   public function bar() {}
17         |   public function baz() {}
18     }
19  ?>
```

- a. Jelaskan maksud dari program di atas!

Program diatas menunjukkan penggunaan interface dan inheritance pad PHP. Terdapat tiga interface a, b, dan c. Interface a dan b memiliki satu method masing-masing yaitu foo() dan bar(), sementara interface c mewarisi kedua interface a dan b serta memiliki satu method sendiri baz(). Dalam contoh ini, ketiga method tersebut didefinisikan dengan body kosong (tanpa operasi yang sebenarnya dilakukan).

Dalam inheritance, class yang diturunkan atau subclass (dalam contoh ini, class d) mewarisi properti dan method dari class induk atau superclass. Dalam penggunaan interface, class yang mengimplementasikan interface diharuskan untuk mendefinisikan seluruh method yang ada di interface tersebut. Dengan demikian, penggunaan interface membantu memastikan bahwa class yang mengimplementasikan interface selalu memiliki method yang diperlukan, dan memungkinkan class tersebut untuk berinteraksi dengan class lain melalui interface yang sama.

- b. Hapuslah kode baris 27-29, lalu jalankan Lat4_6.php. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut dapat terjadi?



The screenshot shows a code editor with two tabs: 'Lat4_6.php' and 'Lat4_6_after.php'. The active tab is 'Lat4_6_after.php', which contains the following PHP code:

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

Below the code editor, a web browser window is open, displaying a fatal error message:

Fatal error: Class d contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (c::baz) in C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_6_after.php on line 14

Hal ini terjadi karena class yang diturunkan yakni class d mewarisi properti dan method dari class induk atau superclass. Dalam penggunaan interface, class yang mengimplementasikan interface diharuskan untuk mendefinisikan seluruh method yang ada di interface tersebut.

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



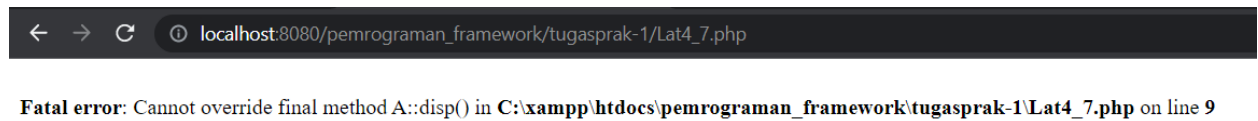
```
Lat4_6.php Lat4_6_after.php X
Lat4_6_after.php
1  <?php
2      interface a {
3          public function foo();
4      }
5
6      interface b {
7          public function bar();
8      }
9
10     interface c extends a, b {
11         public function baz();
12     }
13
14     class d implements c {
15         public function foo() {}
16         public function bar() {}
17         public function baz() {}
18     }
19
20     class e implements a, b {
21         public function foo() {}
22         public function bar() {}
23     }
24  ?>
```

- d. Simpulkan apa yang anda peroleh dari Latihan 6!

Dalam penggunaan interface, class yang mengimplementasikan interface diharuskan untuk mendefinisikan seluruh method yang ada di interface tersebut. Dengan demikian, penggunaan interface membantu memastikan bahwa class yang mengimplementasikan interface selalu memiliki method yang diperlukan, dan memungkinkan class tersebut untuk berinteraksi dengan class lain melalui interface yang sama.

Latihan 7

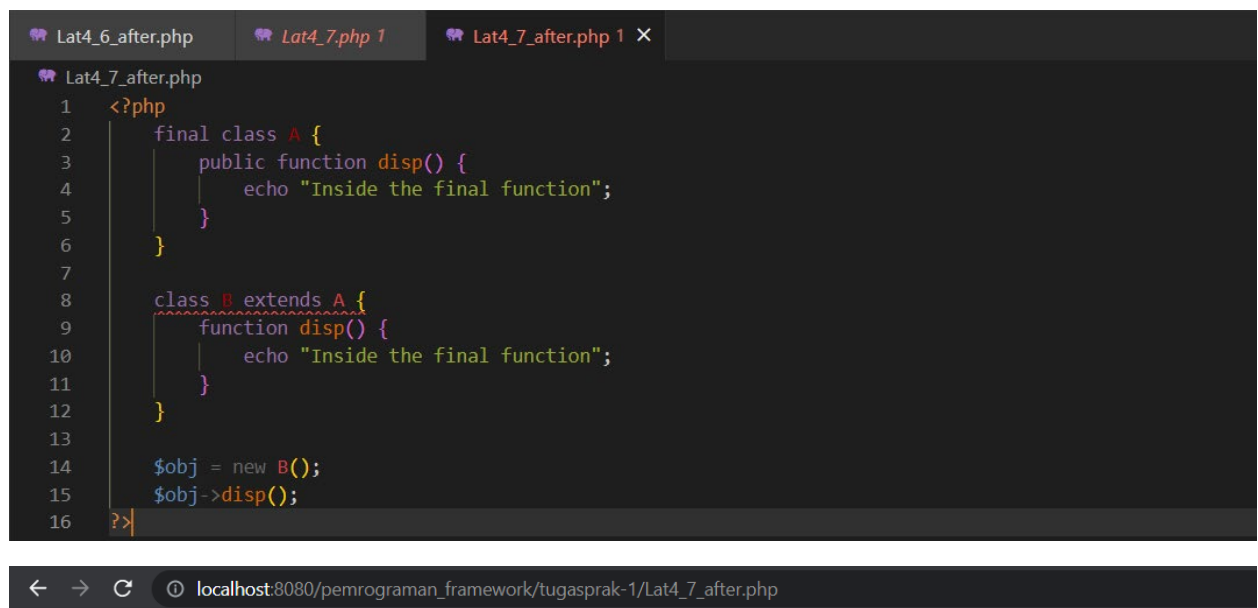
- a. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?



The screenshot shows a web browser at the URL `localhost:8080/pemrograman_framework/tugasprak-1/Lat4_7.php`. It displays a fatal error message: **Fatal error:** Cannot override final method A::disp() in `C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_7.php` on line 9.

Hal ini terjadi karena tidak dapat override final method.

- 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!



The screenshot shows a code editor with three tabs: `Lat4_6_after.php`, `Lat4_7.php 1`, and `Lat4_7_after.php 1`. The active tab is `Lat4_7_after.php`, which contains the following PHP code:

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

Below the code editor, a browser window shows the URL `localhost:8080/pemrograman_framework/tugasprak-1/Lat4_7_after.php` and a fatal error message: **Fatal error:** Class B cannot extend final class A in `C:\xampp\htdocs\pemrograman_framework\tugasprak-1\Lat4_7_after.php` on line 8.

Jika kita menghapus kata final pada baris 3 dan menambahkan kata final pada baris 2, maka program akan menghasilkan error. Karena ketika kita menambahkan kata final pada baris 2, itu berarti kelas A tidak dapat diturunkan lagi (subclassing) dan karena itu, kelas B tidak dapat mewarisi metode `disp()` dari kelas A.

- c. Simpulkan apa yang anda peroleh dari Latihan 7!

Kita dapat menandai sebuah method atau kelas dengan kata kunci final, yang berarti method atau kelas tersebut tidak dapat diubah lagi oleh subclass-nya. Jika kita mencoba untuk men-override method atau mewarisi kelas tersebut, maka akan terjadi error.

Latihan 8

Simpulkan apa yang anda peroleh dari Latihan 8!



```
1  <?php
2      class One {
3          private static $var = 0;
4
5          function __construct() {}
6          static function disp() {
7              print self::$var;
8          }
9
10         function __destruct() {}
11     }
12
13     One::disp();
14 ?>
```

localhost:8080/pemrograman_framework/tugasprak-1/Lat4_8.php

0

Hasil keluaran dari program tersebut adalah 0, karena nilai variabel static \$var diinisialisasi dengan 0 dan dicetak oleh method disp().

Latihan 9

Dari contoh kode MVC pada penjelasan di atas buatlah :

1. Model berhubungan langsung dengan database mysql dengan spesifikasi sebagai berikut :
 - a. Database : Library
 - b. 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!

book.php X

book.php

```
1  <?php
2      class book {
3          public $judul;
4          public $pengarang;
5          public $penerbit;
6          public $tahun;
7
8          public function __construct($title, $author, $publisher, $year) {
9              $this->judul = $title;
10             $this->pengarang = $author;
11             $this->penerbit = $publisher;
12             $this->tahun = $year;
13         }
14     }
15  ?>
```

controller.php X

controller.php

```
1  <?php
2      include_once 'model.php';
3
4      class controller {
5          function invoke() {
6              $model_data = new model();
7              $row_data = $model_data->getData();
8
9              $data = "<table border=1>
10             <tr>
11                 <th> Judul </th>
12                 <th> Pengarang </th>
13                 <th> Penerbit </th>
14                 <th> Tahun </th>
15             </tr>";
16
17             foreach($row_data as $key => $value) {
18                 $data .= "<tr><td>".$value->judul."</td>
19                 <td>".$value->pengarang."</td>
20                 <td>".$value->penerbit."</td>
21                 <td>".$value->tahun."</td></tr>";
22             }
23             $data .= "</table>";
24             include('view.php');
25         }
26     }
27  ?>
```

index.php X

index.php

```
1  <?php
2      include_once 'controller.php';
3
4      $main_ctrl = new controller();
5      $main_ctrl->invoke();
6  ?>
```

```
model.php X
model.php
1  <?php
2      include_once 'book.php';
3
4      class model {
5          public function getData() {
6              return array(
7                  new book('Pemrograman PHP & MySql','Bayu Priyambadha','UB Press','2011'),
8                  new book('Pemrograman Framework MVC dengan PHP','Widhy','UB Press','2011'),
9                  new book('Membangun Aplikasi Web dengan PHP dan AJAX','Achmad Arwan','UB Press','2012'),
10                 new book('Kolaborasi PHP, MVC, dan AJAX','Bayu Priyambadha','UB Press','2012'));
11          }
12      }
13  ?>
```

```
view.php X
view.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title> Prak-PemFrame-1 </title>
8  </head>
9  <body>
10     <center>
11         <?= $data ?>
12     </center>
13 </body>
14 </html>
```

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

~ Terima Kasih ~