

```
<?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();

2. Lat4_2a dan Lat4_2B

- A. Menambahkan construct pada Lat4_2a

The image shows a PHP IDE on the left and a web browser on the right. The IDE displays the following code:

```

1 <?php
2 class mahasiswa
3 {
4     public $nama;
5     public $nim;
6     function __construct($a, $b,)
7     {
8         $this->nama = $a;
9         $this->nim = $b;
10    }
11    echo "Kelas telah dibuat<br/><br/>";
12 }
13 public $prodi;
14 public $semester;
15 function __construct2($c, $d)
16 {
17     $this->prodi = $c;
18     $this->semester = $d;
19 }
20 echo "Keterangan Mahasiswa sudah jelas <br/><br/>";
21 function cetak()
22 {
23     echo $this->nama . "<br/>" . $this->nim . "<br/>";
24 }
25 function cetak2()
26 {
27     echo $this->prodi . "<br/>" . $this->semester . "<br/><br/>";
28 }
29 function __destruct()
30 {
31     echo "Kelas telah dihancurkan";
32 }
33 }
34 }
35

```

The browser shows the output of the class methods:

```

Kelas telah dibuat

Nurul Khusain
G.211.21.0107
FTIK
Semester 3

Kelas telah dihancurkan

```

- B. Tampilan

- C. Kesimpulan : Constructor dalam OOP PHP tidak dapat di Override

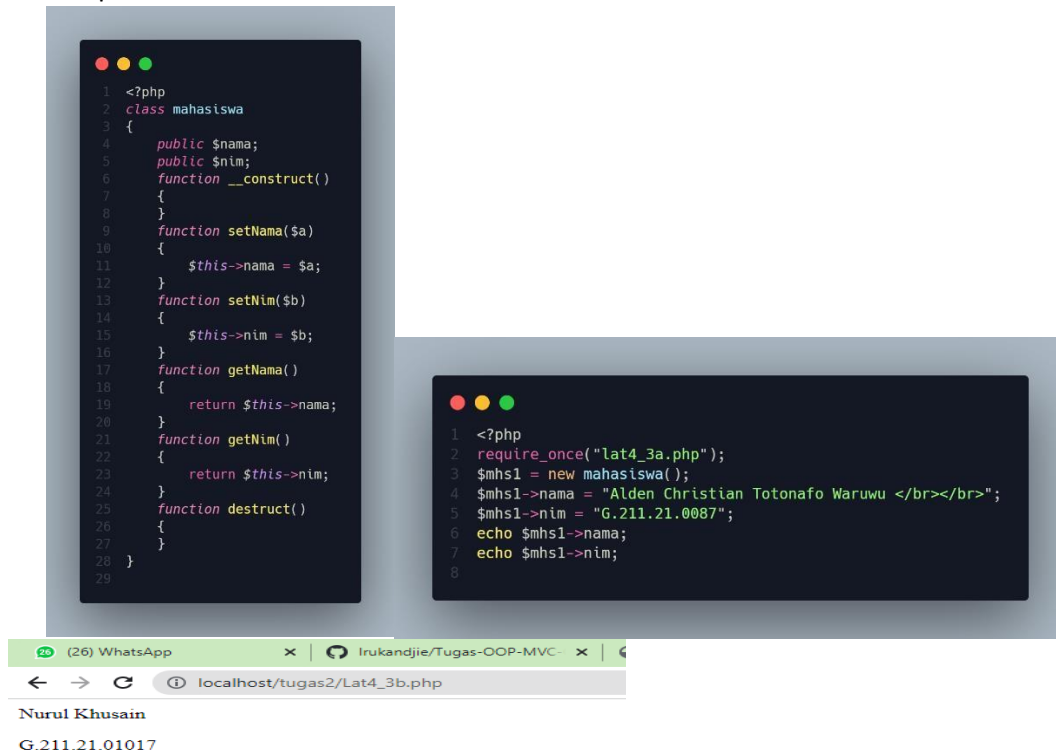
3. Lat4_3a dan Lat4_3b

- A. Program tersebut mengalami beberapa error, Hal itu terjadi dikarenakan Property yang bermodifikasi private hanya bisa digunakan pada class mahasiswa sendiri.
- B. Modifier protected dan Public

b.1 Protected

b.2 Public

Tampilan:



The image shows a code editor with two files. The left file, 'lat4_3a.php', defines a 'mahasiswa' class with public properties '\$nama' and '\$nim', and methods for setting and getting these values. The right file, 'lat4_3b.php', uses the 'mahasiswa' class to create an object and set its values. Below the code editor, a web browser window shows the output of the script, displaying the name 'Nurul Khusain' and the ID 'G.211.21.01017'.

```
1 <?php
2 class mahasiswa
3 {
4     public $nama;
5     public $nim;
6     function __construct()
7     {
8     }
9     function setNama($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 }
29
```

```
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new mahasiswa();
4 $mhs1->nama = "Alden Christian Totonafa Waruwu </br></br>";
5 $mhs1->nim = "G.211.21.0087";
6 echo $mhs1->nama;
7 echo $mhs1->nim;
8
```

(26) WhatsApp x | Irukandjie/Tugas-OOP-MVC x |

localhost/tugas2/Lat4_3b.php

Nurul Khusain
G.211.21.01017

C. Kesimpulan:

I. Penggunaan Modifier

Modifier keterangan public

Untuk mendefinisikan data atau metode yang akan terlihat dari luar oleh siapapun dan dimanapun.

A. Private

Untuk Mendefinisikan data atau metode yang akan terlihat pada class/object itu sendiri.

B. Protected

Untuk mendefinisikan data atau metode untuk tidak terlihat dari luar (seperti private), tetapi akan dapat diakses oleh "anak" dari class tersebut.

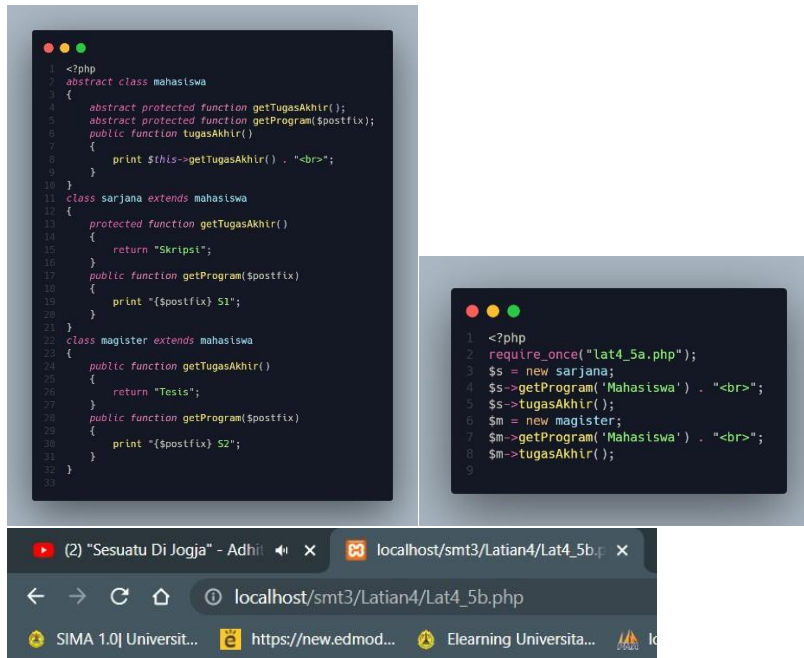
- II. Modifier Protected dan Private, Properties bisa dipanggil dengan mengimplementasikan setter-getter.

4. Lat4_4a dan Lat4_4b

- A. Tampilan
B. Kesimpulan: Pada PHPOOP, Class asisten (child) bisa memanggil method dari mahasiswa (Parent).

5. Lat4_5a dan Lat_5b

- A. Tampilan Program



```
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)
29     {
30         print "{$postfix} S2";
31     }
32 }
33
```

```
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
```

(2) "Sesuatu Di Jogja" - Adh... X localhost/smt3/Lat4/Lat4_5b.php X
localhost/smt3/Lat4/Lat4_5b.php
SIMA 1.0| Universit... https://new.edmod... Elearning Universita...

Mahasiswa S1Skripsi
Mahasiswa S2Tesis

- B. Kesimpulan
class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.

6. Lat4_6

- A. Maksud dari program diatas adalah penggunaan object interface
B. Tampilan

```
Lat4_6.php > d > bar
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 }
23
```

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\TUGAS 2 FRAMEWORK OOP MVC\Lat4_6.php on line 14

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. Setelah menambahkan class baru yaitu class e

```
Lat4_6.php > e > foo
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 }
26 class e
27 {
28     public function foo()
29     {
30     }
31     public function bar()
32     {
33     }
34     public function baz()
35     {
36     }
37 }
```

Error, method foo dan bar terdapat pada interface a dan b, pada penerapannya sebuah class baru tidak dapat mengimplementasikan method yang sama pada dua interface.

D. Kesimpulan

- 1) Interface didefinisikan dengan “Interface” keyword, mirip dengan deklarasi class biasa, hanya saja definisi atau detail method tidak dituliskan.
- 2) Seluruh method yang dideklarasikan pada interface harus memiliki modifier “ public ”.
- 3) Untuk mengimplementasikan sebuah interface, kita dapat menggunakan “implement” keyword.

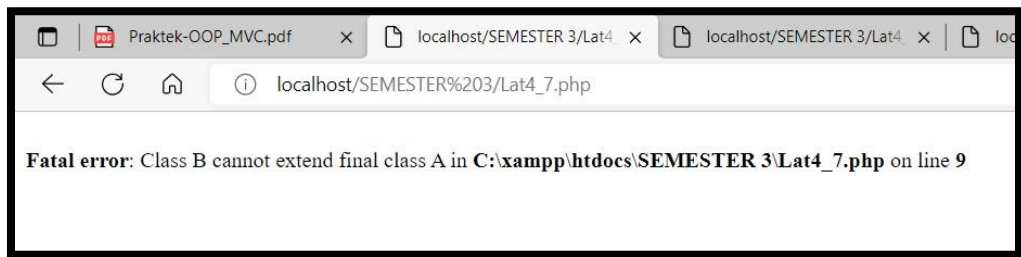
- 4) Seluruh method yang ada pada interface harus diimplementasikan seluruhnya. Sebuah class bisa mengimplementasikan lebih dari satu interface.
- 5) Class tidak bisa mengimplementasikan dua interface yang mempunyai nama method yang sama.
- 6) Interface bisa diwariskan seperti class menggunakan “extends”.
- 7) Class yang mengimplementasikan interface harus menggunakan method-method yang ada pada interface tersebut dengan nama dan spesifikasi yang sama persis.

7. Lat4_7

- a. Tampilan program (program tersebut mengalami error)



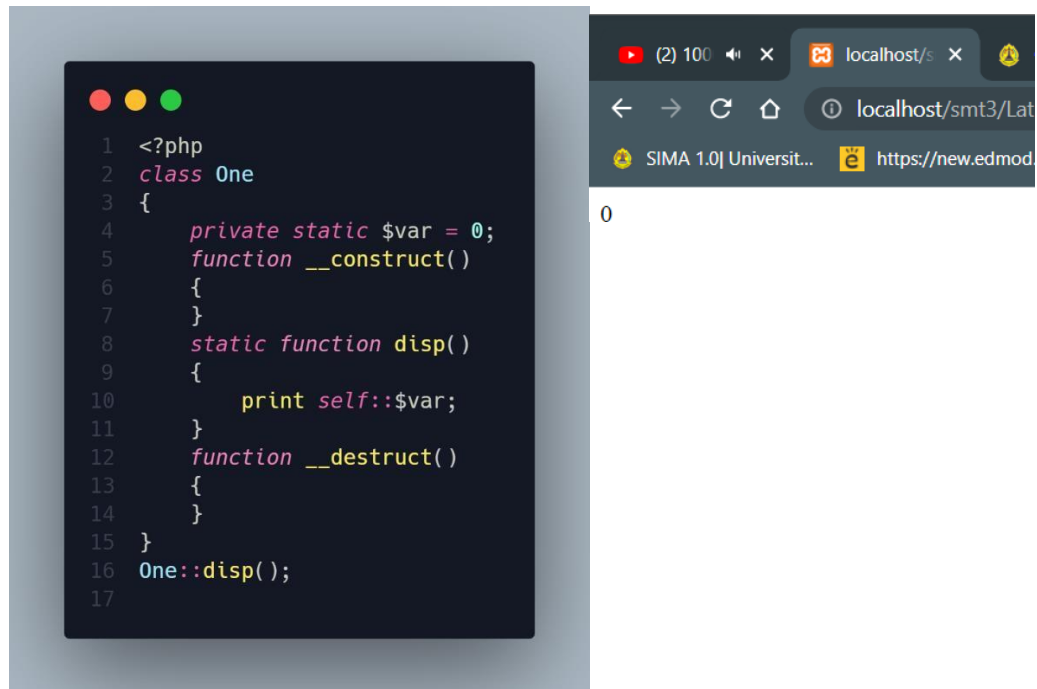
- b. Setelah memindahkan kata final dari baris 5 ke baris 2 (program tersebut tetap mengalami error), error pada program tersebut dikarenakan...



- c. Kesimpulan
1. final, akan mencegah proses overriding method pada class anak (sub-class)
 2. 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).

8. Lat4_8

- A. Tampilan



B. 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 "->"

9. Latian_9

a. Database

perpustakaan	id_buku	judul	pengarang	penerbit	tahun_terbit	isbn	jumlah_buku	tgl_input	file
anggota	13	Laut Bercerita	Leila S. Chudori	Gramedia	1997	30	2017-12-10	03102022193447	
buku	14	Antares	Rweinda	Loveable	2020	30	2021-01-01	03102022193646	
transaksi	15	Dikta dan Hukum	Dhian Farah	Loveable	2021	30	2022-05-01	03102022193840	
user	16	Dragon Ball	Akira Toriyama	Elex media komputindo	1995	30	2004-02-10	13122019052904	
phpmyadmin	17	Sword Art Online	Kawahara Reki	A-1 Pictures	2009	30	2019-12-01	13122019073332	
test									

b. Model

```

1 <?php
2
3 class Koneksi {
4     var $host = 'localhost';
5     var $user = 'root';
6     var $pass = '';
7     var $db_name = 'perpustakaan'; // nama database
8
9     public function __construct() {
10         $this->konek = mysqli_connect($this->host, $this->user, $this->pass, $this->db_name);
11     }
12     //
13 }
14 ?>

```

c. Controller

```

1 <?php
2
3 require_once '../..../config/perpustakaan.php';
4
5 class Buku extends perpustakaan {
6     public function getDataPetugas($id) {
7         $stmt = mysqli_query($this->konek, "SELECT * FROM tb_petugas WHERE id_petugas = '" . $id . "'");
8
9         return $stmt;
10     }
11 }

```

d. View

e. Tampilan

The screenshot displays the Miracle Library web application interface. The browser's address bar shows the URL `localhost/perustakaan/index_admin.php?page=buku`. The application has a dark sidebar on the left with the 'Miracle Library' logo and three menu items: 'Data Buku', 'Data Peminjam', and 'Transaksi Peminjaman'. The main content area features a 'Tambah Buku' button and a table titled 'Table Buku'. The table lists 5 books with columns for No, Judul, Pengarang, Penerbit, ISBN, Jumlah Buku, and Aksi. Each book entry has 'Ubah' (Edit) and 'Hapus' (Delete) buttons. Below the table, it indicates 'Showing 1 to 5 of 5 entries' and includes pagination controls for 'Previous', '1', and 'Next'.

No	Judul	Pengarang	Penerbit	ISBN	Jumlah Buku	Aksi
1	Laut Ber cerita	Leila S. Chudori	Gramedia		30	Ubah Hapus
2	Antares	Rweinda	Loveable		30	Ubah Hapus
3	Dikta dan Hukum	Dhian Farah	Loveable		30	Ubah Hapus
4	Dragon Ball	Akira Toriyama	Elex media komputindo		30	Ubah Hapus
5	Sword Art Online	Kawahara Reki	A-1 Pictures		30	Ubah Hapus