

```

10         $a->merk = "Ferrari";
11         $a->getInfo($a);
12     }
13 }
14
15 </p>
16 </div>
17 <div class="btn-group btn-group-sm" style="width: 40px;">
18     <a href="Lat4_2a.php" class="btn btn-danger">
19         < </a>
20     </div>
21 <p style="font-size: 15px;" class="text-center mt-2">
22     Wisnu Cahya Ramadhan || 6.211.21.0069
23 </p>
24 </div>
25 </body>
26 </html>

```

C. Kesimpulan : Jadi, pada saat menggunakan method overload getInfo dengan parameter \$a ketika dijalankan akan terjadi error, hal ini terjadi karena parameter \$a pada fungsi getInfo tidak tahu apa yang akan dia jalankan, akan tetapi jika kita mengganti \$this pada nama maupun merk dengan parameter \$a di dalam fungsi getInfo dan memasukkan variable \$mobil ke dalam fungsi \$mobil->getInfo(); maka parameter \$a akan mendeteksi bahwa \$a akan mengacu ke nama dan juga merk. dan ketika dijalankan hasilnya akan sama seperti program awalnya.

2. Lat4_2a dan Lat4_2b

a. menambahkan construct pada Lat4_2a

```

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

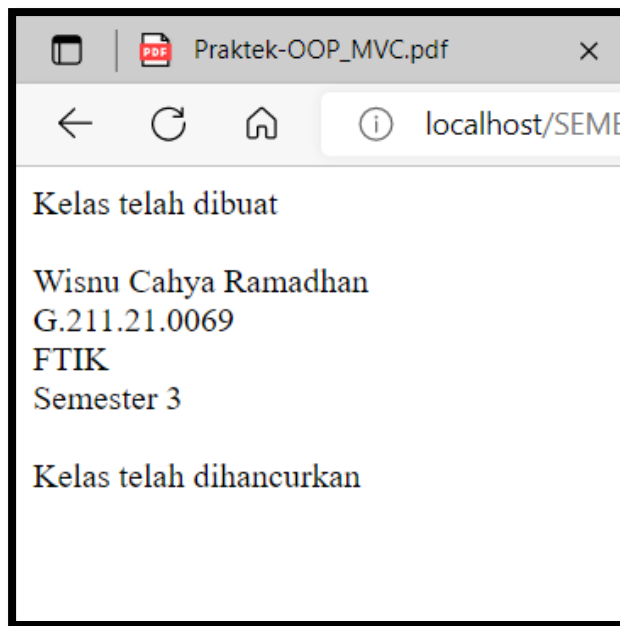
```

```

1 <?php
2 require_once("Lat4_2a.php");
3 $mhs2 = new mahasiswa("Wisnu Cahya Ramadhan", "G.211.21.0069", "FTIK",
4     "Semester 3");
5 $mhs2->cetak();

```

b. Tampilan



- c. **Kesimpulan** : Contructor dalam php oop tidak bisa di override dan jika ingin menambahkan konstruktor paling tidak kita harus membuat satu class lain yang didalamnya terdapat 1 buah konstruktor. Begitu juga dengan desduktor.

3. Lat4_3a dan Lat4_3b

- a. Ya program tersebut mengalami error, hal itu dikarenakan property yang bermodifier private hanya bisa digunakan pada class mahasiswa sendiri.

b. Modifier protected dan Public

tampilan :

```

terminal Help Lat4_3a.php
Lat4_3a.php > mahasiswa > setNama
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

```

```

terminal Help Lat4_3b.php - SEMESTER 3 - Visual Studio Code
Lat4_3b.php > ...
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new mahasiswa();
4 $mhs1->nama = "WISNU CAHYA RAMADHAN </br></br>";
5 $mhs1->nim = "G.211.21.0069";
6 echo $mhs1->nama;
7 echo $mhs1->nim;
8

```



c. Tampilan private :

```

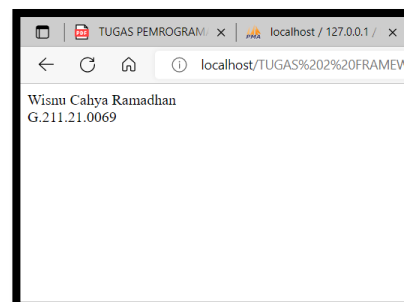
terminal Help Lat4_3a.php - TUGAS 2
Lat4_3a.php > mahasiswa > __construct > $a
1 <?php
2 class mahasiswa
3 {
4     private $nama;
5     private $nim;
6     function __construct(
7     {
8         $a = "Wisnu Cahya Ramadhan <br>";
9         $b = "G.211.21.0069";
10    }
11    {
12        $this->nama = $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    function destruct()
32    {
33    }
34 }
35

```

```

terminal Help Lat4_3b.php - TUGAS 2 FRAMEWORK OOP
Lat4_3b.php > ...
1 <?php
2 require_once("lat4_3a.php");
3 $mhs1 = new mahasiswa();
4 echo $mhs1->getnama() . $mhs1->getnim();
5

```



d. Kesimpulan :

1. Penggunaan modifier, Modifier Keterangan :

a. Public

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

b. Private

Untuk mendefinisikan data atau metode agar hanya terlihat pada class/object itu sendiri.

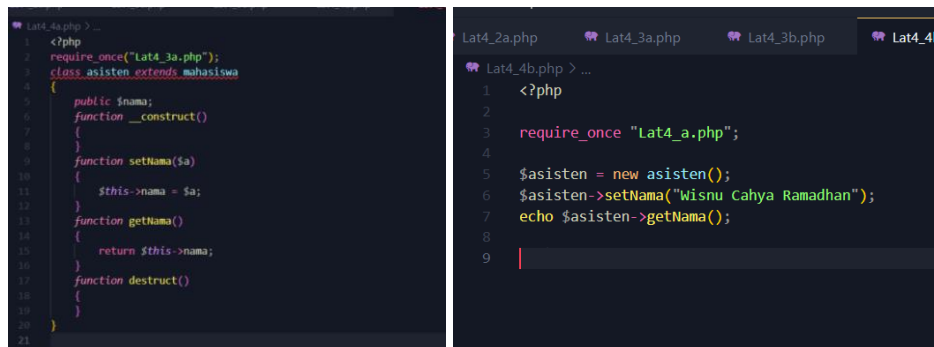
c. Protected

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

2. Modifier protected dan private, properties bisa dipanggil dengan mengimplementasikan setter-getter.

4. Lat4_4a dan Lat4_4b

a. Tampilan



The image shows two side-by-side screenshots of a code editor. The left screenshot displays the code for a class named 'asisten' which extends 'mahasiswa'. It includes a public property '\$nama', a constructor '___construct()', a 'setNama(\$a)' method, a 'getNama()' method, and a 'destruct()' method. The right screenshot shows a script 'Lat4_4b.php' that includes 'Lat4_a.php', creates a new 'asisten' object, sets its name to 'Wisnu Cahya Ramadhan', and echoes the result of the 'getNama()' method.

```
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_2a.php Lat4_3a.php Lat4_3b.php Lat4_4b
Lat4_4b.php > ...
1 <?php
2
3 require_once "Lat4_a.php";
4
5 $asisten = new asisten();
6 $asisten->setNama("Wisnu Cahya Ramadhan");
7 echo $asisten->getNama();
8
9
```

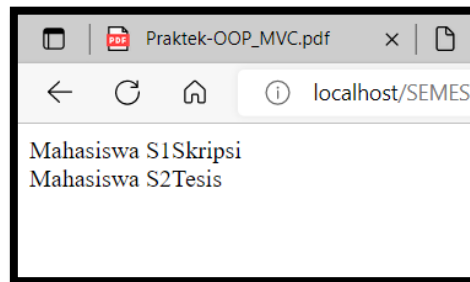
- b. **Kesimpulan :** Pada php oop, class asisten (child) bisa memanggil method dari mahasiswa (parent). Tidak terjadi error karena getNama masih berada di dalam class Mahasiswa Kalau getNama dipindahkan ke class asisten yang terjadi adalah error, oleh karena itu agar hal tersebut tidak error maka modifier private bisa diganti dengan protected agar kita bisa akses getNama di subclass-nya/class child.

5. Lat4_5a dan Lat4_5b

a. Tampilan program

```
Terminal Help Lat4_5a.php - SEMESTER 3 - Visual Studio Code
Lat4_3a.php blbaba.php Lat4_3b.php Lat4_5a.php
Lat4_5a.php > magister
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 }
```

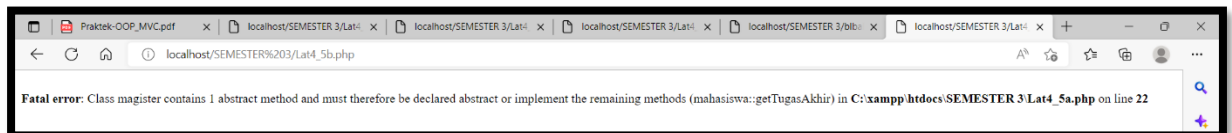
```
Terminal Help Lat4_5b.php - SEMESTER 3 - Visual Studio Code
Lat4_3a.php blbaba.php Lat4_3b.php Lat4_5a.php Lat4_5b.php
Lat4_5b.php > ...
1 <?php
2 require_once("lat4_5a.php");
3 $s = new sarjana;
4 $s->getProgram('Mahasiswa') . "<br>";
5 $s->tugasAkhir();
6 $m = new magister;
7 $m->getProgram('Mahasiswa') . "<br>";
8 $m->tugasAkhir();
9 }
```



b. Setelah menghapus baris ke 29-32

```
Terminal Help Lat4_5a.php - SEMESTER 3 - Visual Studio Code
Lat4_3a.php blbaba.php Lat4_3b.php Lat4_5a.php
Lat4_5a.php > magister
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 }
29
30 }
```

```
Terminal Help Lat4_5b.php - SEMESTER 3 - Visual Studio Code
Lat4_3a.php blbaba.php Lat4_3b.php Lat4_5a.php Lat4_5b.php
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 }
```



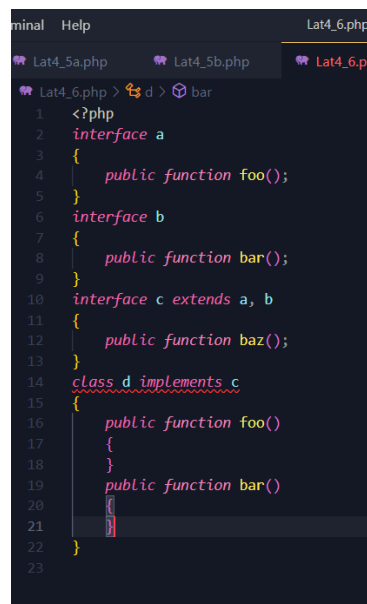
c. Kesimpulan :

class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya. Dapat disimpulkan bahwa class dan method dengan keyword abstract hanya bisa diakses melalui turunannya saja

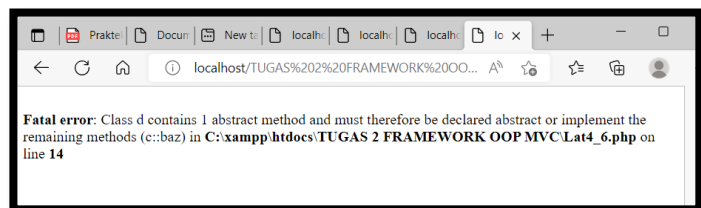
6. Lat4_6

- a. **Maksud dari program diatas adalah** Interface a b c merupakan sebuah class yang didalamnya terdapat sebuah method berupa method foo(), bar(), dan baz() tetapi diboleh memiliki properti didalamnya dan method tersebut harus dalam bentuk abstract, sedangkan class d merupakan implements dari class c dan dimana class c juga merupakan turunan dari interface a dan b. Nah, di class d ini lah yang boleh menaruh properti di dalamnya seperti modifier private maupun protected dan mengisi semua method yang digunakan diparentnya.

b. Tampilan



```
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
```



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**

```
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 }
```

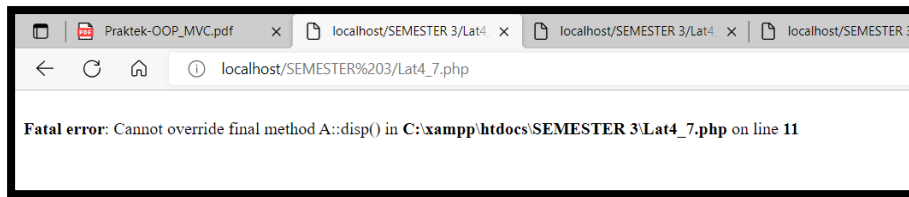
Error, method foo dan bar terdapat pada interace a dan b, pada penerapannya sebuah classbaru tidak dapat mengimplementasikan method yang sama pada dua interface.

d. Kesimpulan

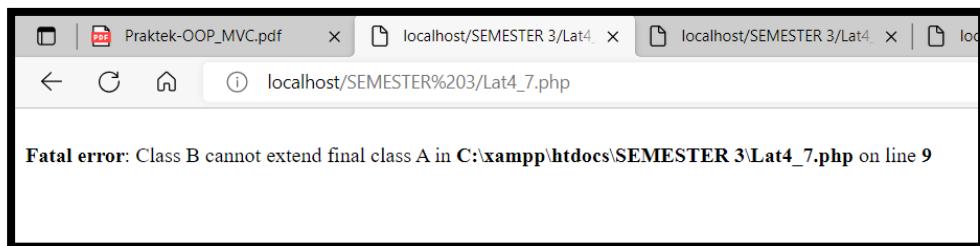
1. Interface didefinisikan dengan “Interface” keyword, mirip dengan deklarasi class biasa, hanya saja definisi atau detail method tidak dituliskan.
2. Seluruh method yang dideklarasikan pada interface harus memiliki modifier “public”.
3. Untuk mengimplementasikan sebuah interface, kita dapat menggunakan “implement” keyword.
4. Seluruh method yang ada pada interface harus diimplementasikan seluruhnya. Sebuahclass bisa mengimplementasikan lebih dari satu interface.
5. Class tidak bisa mengimplementasikan dua interface yang mempunyai nama methodyang sama.
6. Interface bisa diwariskan seperti class menggunakan “extends”.
7. Class yang mengimplementasikan interface harus menggunakan method-method yangada 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...

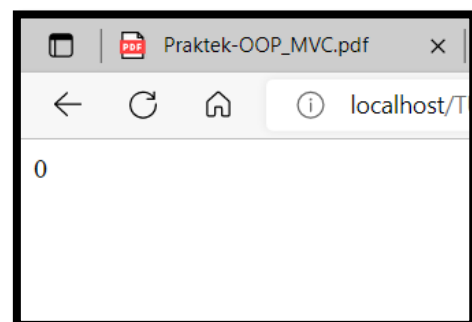
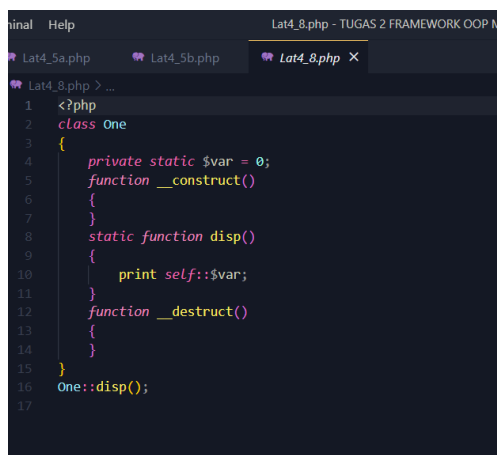


c. Kesimpulan

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;

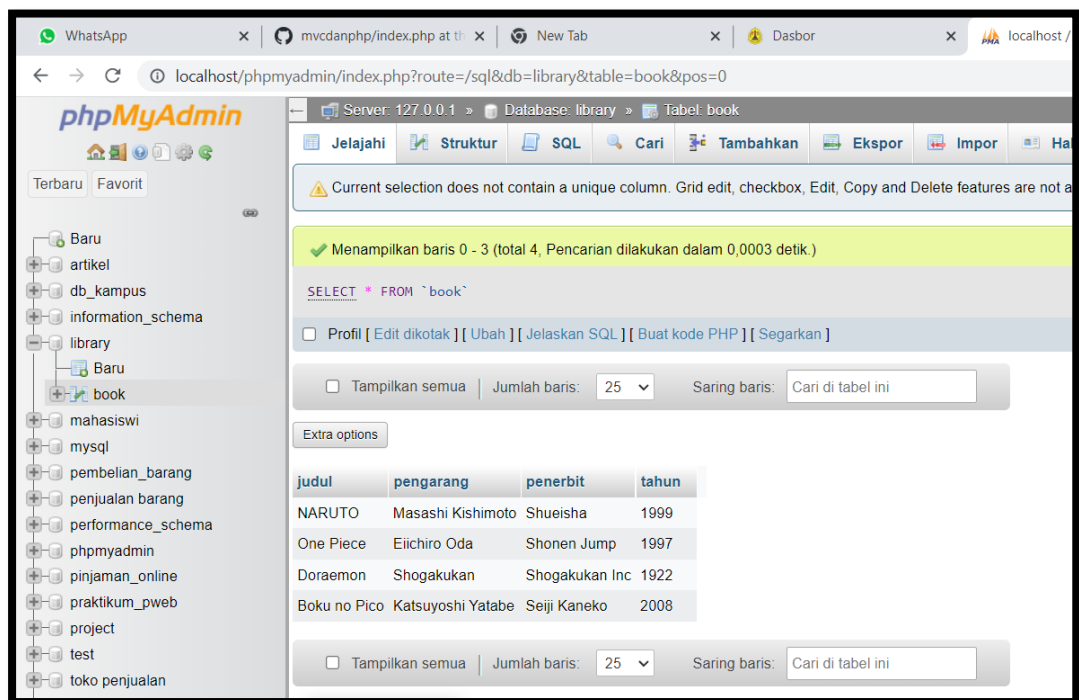
2.Sifat statis dapat diakses tanpa perlu sebuah contoh objek dari kelas, menggunakan nama kelas bersama dengan ::- Sintaks:

ClassName :: \$nama_property method_name();

Perhatikan bahwa properti statis menggunakan tanda dollar (\$).Properti statis tidak dapat diakses melalui obyek menggunakan operator panah "->".

9. Lat4_9

a. Database



The screenshot shows the phpMyAdmin interface. The left sidebar lists databases and tables, with 'library' selected. The main panel shows the 'book' table structure and data. The table has columns: judul, pengarang, penerbit, and tahun. The data is as follows:

judul	pengarang	penerbit	tahun
NARUTO	Masashi Kishimoto	Shueisha	1999
One Piece	Eiichiro Oda	Shonen Jump	1997
Doraemon	Shogakukan	Shogakukan Inc	1922
Boku no Pico	Katsuyoshi Yatabe	Seiji Kaneko	2008

b. Controllers

```
Get Started library.php X libraryModel.php daftar-buku.php
controllers > library.php > Library > render
1 <?php
2 class Library
3 {
4     private $model;
5
6     function __construct()
7     {
8         $this->model = new LibraryModel();
9     }
10
11    public function index()
12    {
13        $data['listBuku'] = $this->model->getLibrary();
14        return $this->render('daftar-buku', $data);
15    }
16
17    private function render($view, $data = [])
18    {
19        if ($data) {
20            foreach ($data as $key => $value) {
21                $$key = $value;
22            }
23        }
24
25        require_once __DIR__ . '/../views/' . $view . '.php';
26    }
27 }
28
```

c. Models

```
Help libraryModel.php - Lat4_9 - Visual Studio Code
library.php X libraryModel.php X daftar-buku.php
libraryModel.php > ...
function __construct()
{
    $this->connection = new mysqli(
        'localhost',
        'root',
        '',
        'library'
    );
}

public function getLibrary($id = 1)
{
    if ($id) {
        $query = "SELECT * FROM book WHERE id = $id";
        $result = $this->connection->query($query);
        if ($result->num_rows === 0) {
            return false;
        } else {
            return $result->fetch_assoc();
        }
    } else {
        $query = "SELECT * FROM book";
        $result = $this->connection->query($query);
        if ($result->num_rows === 0) {
            return false;
        } else {
            while ($row = $result->fetch_assoc()) {
                $result[] = $row;
            }
            return $result;
        }
    }
}
}
```

d. Views

```

terminal  Help  daftar-buku.php - TUGAS 2 FRAMEWORK OOP MVC - Visual Studio Code
Lat4_9 > views > daftar-buku.php > html > body > table.table > thead
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Daftar Buku || 6.211.21.0069</title>
9  </head>
10
11 <body>
12     <table class="table">
13         <thead>
14             <tr>
15                 <th>Judul</th>
16                 <th>Penulis</th>
17                 <th>Penerbit</th>
18                 <th>Tahun</th>
19             </tr>
20         </thead>
21         <tbody>
22             <?php
23                 $no = 1;
24                 if ($listBuku) : foreach ($listBuku as $value) : ?>
25
26                     <tr>
27                         <td><?= $value["Judul"]; ?></td>
28                         <td><?= $value["Pengarang"]; ?></td>
29                         <td><?= $value["Penerbit"]; ?></td>
30                         <td><?= $value["Tahun"]; ?></td>
31                     </tr>
32                 <?php
33                     endforeach;
34                 else : ?>
35                     <tr>
36                         <td colspan="5">
37                             <b>Data belum ada</b>

```

```

37                             <b>Data belum ada</b>
38                         </td>
39                     </tr>
40                 <?php
41                     endif;
42                 ?>
43             </tbody>
44         </table>
45     </body>
46 </html>

```

e. Index.php

```

Lat4_9 > index.php > ...
1  <?php
2
3  require_once __DIR__ . "/controllers/Library.php";
4  require_once __DIR__ . "/models/libraryModel.php";
5
6  $url = isset($_SERVER["PATH_INFO"]) ? explode("/", ltrim($_SERVER
7  ["PATH_INFO"], "/")) : "/";
8  $index = new Library();
9
10 if ($url[0] = "/") {
11     return $index->index();
12 }

```

f. Tampilan

JUDUL	PENGARANG	PENERBIT	TAHUN
NARUTO	Masashi Kishimoto	Shueisha	1999
One Piece	Eiichiro Oda	Shonen Jump	1997
Doraemon	Shogakukan	Shogakukan Inc	1922
Boku no Pico	Katsuyoshi Yatabe	Seiji Kaneko	2008