

# **PROJECT OBJECT ORIENTED PROGRAMMING**

## **(PEMROGRAMAN BERORIENTASI OBJEK)**

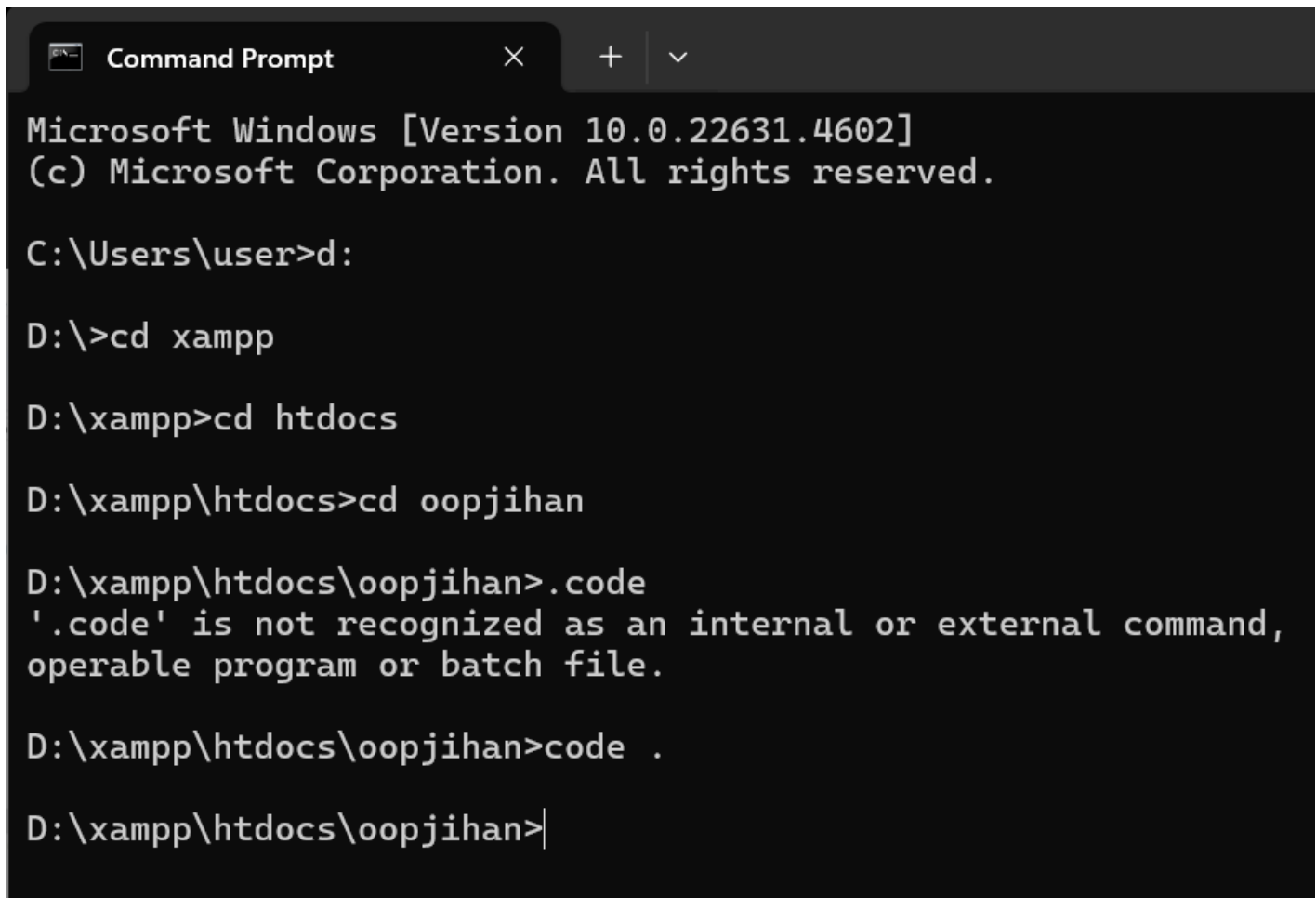


**OLEH:**  
**JIHAN SYAFITRI**  
**NISN. 0083055957**

**REKAYASA PERANGKAT LUNAK**  
**SMK NEGERI 1 KARANG BARU**  
**PEMERINTAH PROVINSI ACEH**  
**2025**

### a. Apa itu OOP?

OOP merupakan suatu metode pemrograman yang berorientasi kepada objek. Dalam bahasa Indonesia OOP dikenal dengan PBO (Pemrograman Berorientasi Objek).



```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>d:

D:\>cd xampp

D:\xampp>cd htdocs

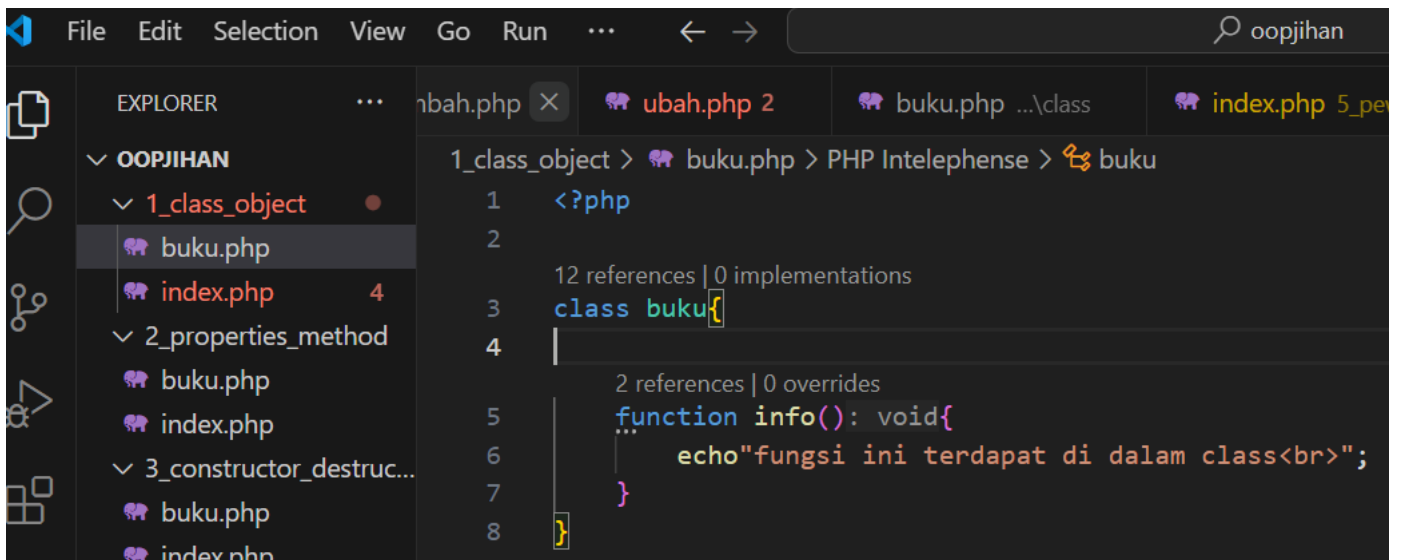
D:\xampp\htdocs>cd oopjihan

D:\xampp\htdocs\oopjihan>.code
'.code' is not recognized as an internal or external command,
operable program or batch file.

D:\xampp\htdocs\oopjihan>code .

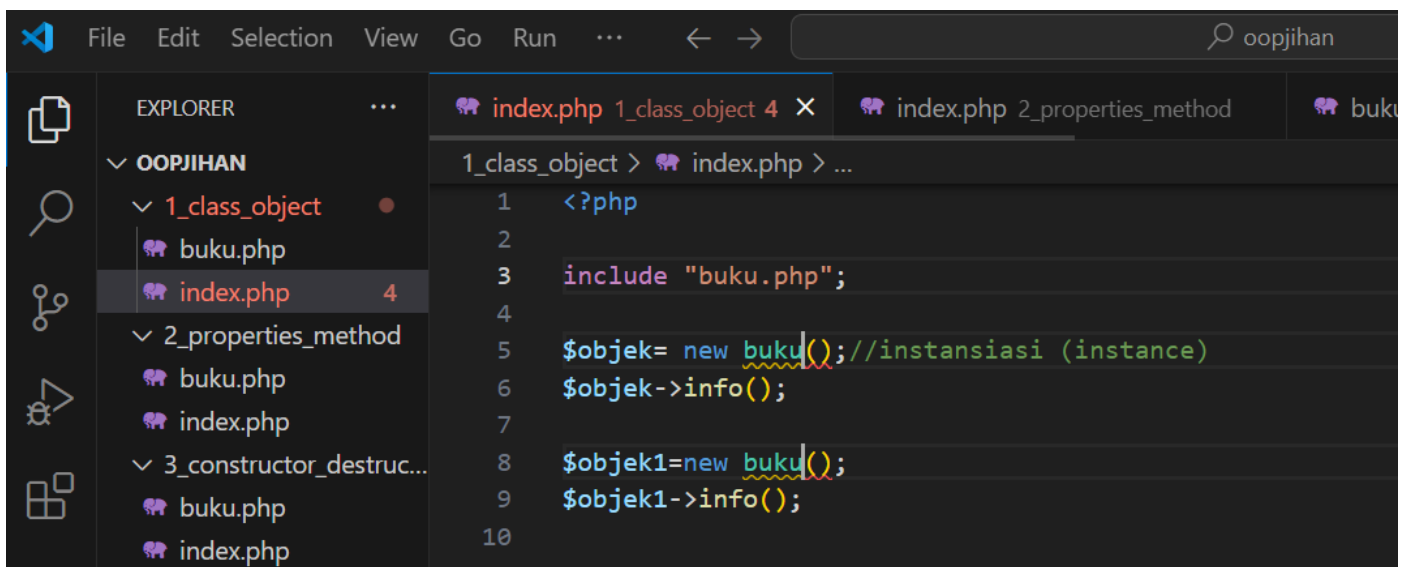
D:\xampp\htdocs\oopjihan>
```

## 1. Class dan object



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The 'OOPJIHAN' folder is expanded, showing '1\_class\_object' and 'buku.php'. The 'buku.php' file is selected, and its content is displayed in the editor. The code defines a class named 'buku' with a method 'info()' that echoes a message.

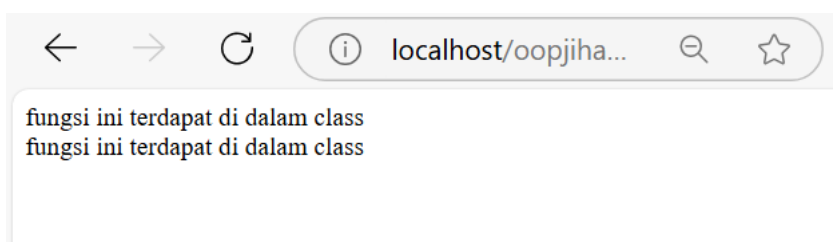
```
1 <?php
2
3 class buku{
4
5     2 references | 0 overrides
6     function info(): void{
7         ...
8         echo "fungsi ini terdapat di dalam class<br>";
9     }
10 }
```



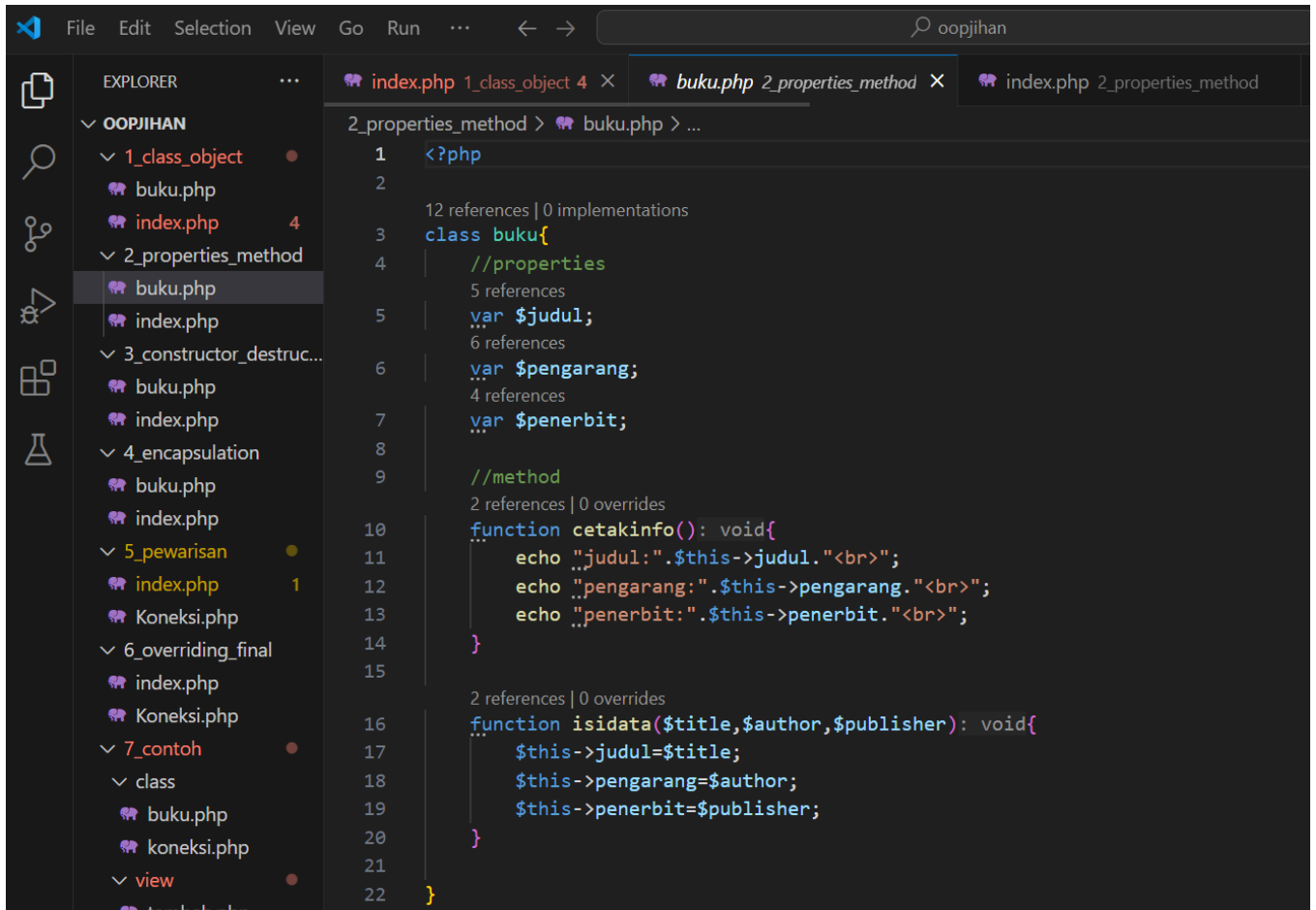
The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The 'OOPJIHAN' folder is expanded, showing '1\_class\_object' and 'index.php'. The 'index.php' file is selected, and its content is displayed in the editor. The code includes 'buku.php' and creates two instances of the 'buku' class, '\$objek' and '\$objek1', and calls their 'info()' method.

```
1 <?php
2
3 include "buku.php";
4
5 $objek= new buku(); //instansiasi (instance)
6 $objek->info();
7
8 $objek1=new buku();
9 $objek1->info();
10
```

Output:

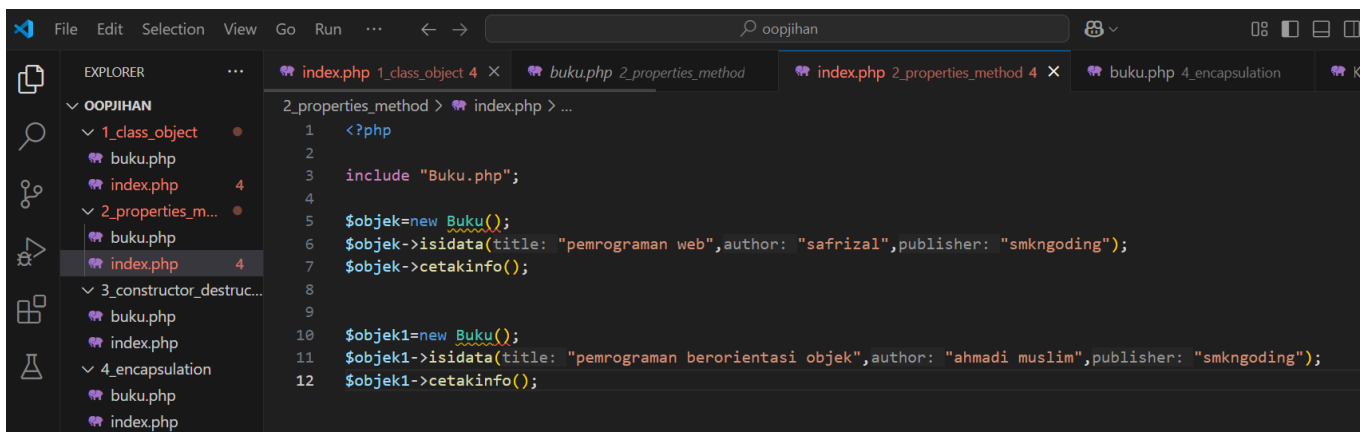


## 2. Properties dan method



The screenshot shows the Visual Studio Code editor with a PHP file named `buku.php` open. The Explorer sidebar on the left shows a project named `OOPJIHAN` with a file structure including `1_class_object`, `2_properties_method`, `3_constructor_destruc...`, `4_encapsulation`, `5_pewarisan`, `6_overriding_final`, `7_contoh`, `class`, `koneksi.php`, and `view`. The `2_properties_method` folder is expanded, showing `buku.php` and `index.php`. The main editor displays the code for `buku.php`, which defines a `Buku` class with properties `$judul`, `$pengarang`, and `$penerbit`, and methods `cetakinfo()` and `isidata()`. The `cetakinfo()` method echoes the values of the properties, and the `isidata()` method sets the values of the properties. The code is as follows:

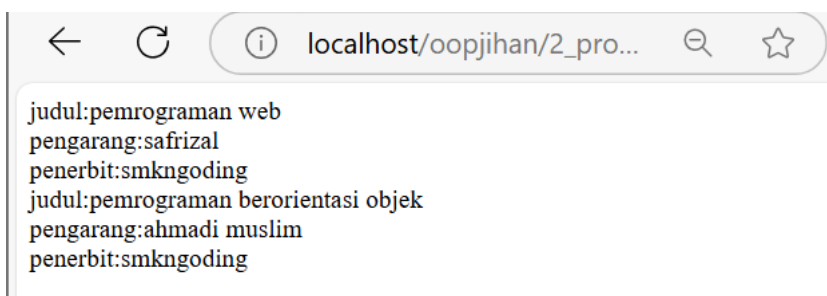
```
1 <?php
2
3 12 references | 0 implementations
4 class buku{
5     //properties
6     5 references
7     var $judul;
8     6 references
9     var $pengarang;
10    4 references
11    var $penerbit;
12
13    //method
14    2 references | 0 overrides
15    function cetakinfo(): void{
16        echo "judul:". $this->judul."<br>";
17        echo "pengarang:". $this->pengarang."<br>";
18        echo "penerbit:". $this->penerbit."<br>";
19    }
20
21    2 references | 0 overrides
22    function isidata($title,$author,$publisher): void{
23        $this->judul=$title;
24        $this->pengarang=$author;
25        $this->penerbit=$publisher;
26    }
27 }
```



The screenshot shows the Visual Studio Code editor with a PHP file named `index.php` open. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The main editor displays the code for `index.php`, which includes the `buku.php` file and uses the `Buku` class to create two objects, `$objek` and `$objek1`, and call their `isidata()` and `cetakinfo()` methods. The code is as follows:

```
1 <?php
2
3 include "Buku.php";
4
5 $objek=new Buku();
6 $objek->isidata(title: "pemrograman web",author: "safrizal",publisher: "smkngoding");
7 $objek->cetakinfo();
8
9
10 $objek1=new Buku();
11 $objek1->isidata(title: "pemrograman berorientasi objek",author: "ahmadi muslim",publisher: "smkngoding");
12 $objek1->cetakinfo();
```

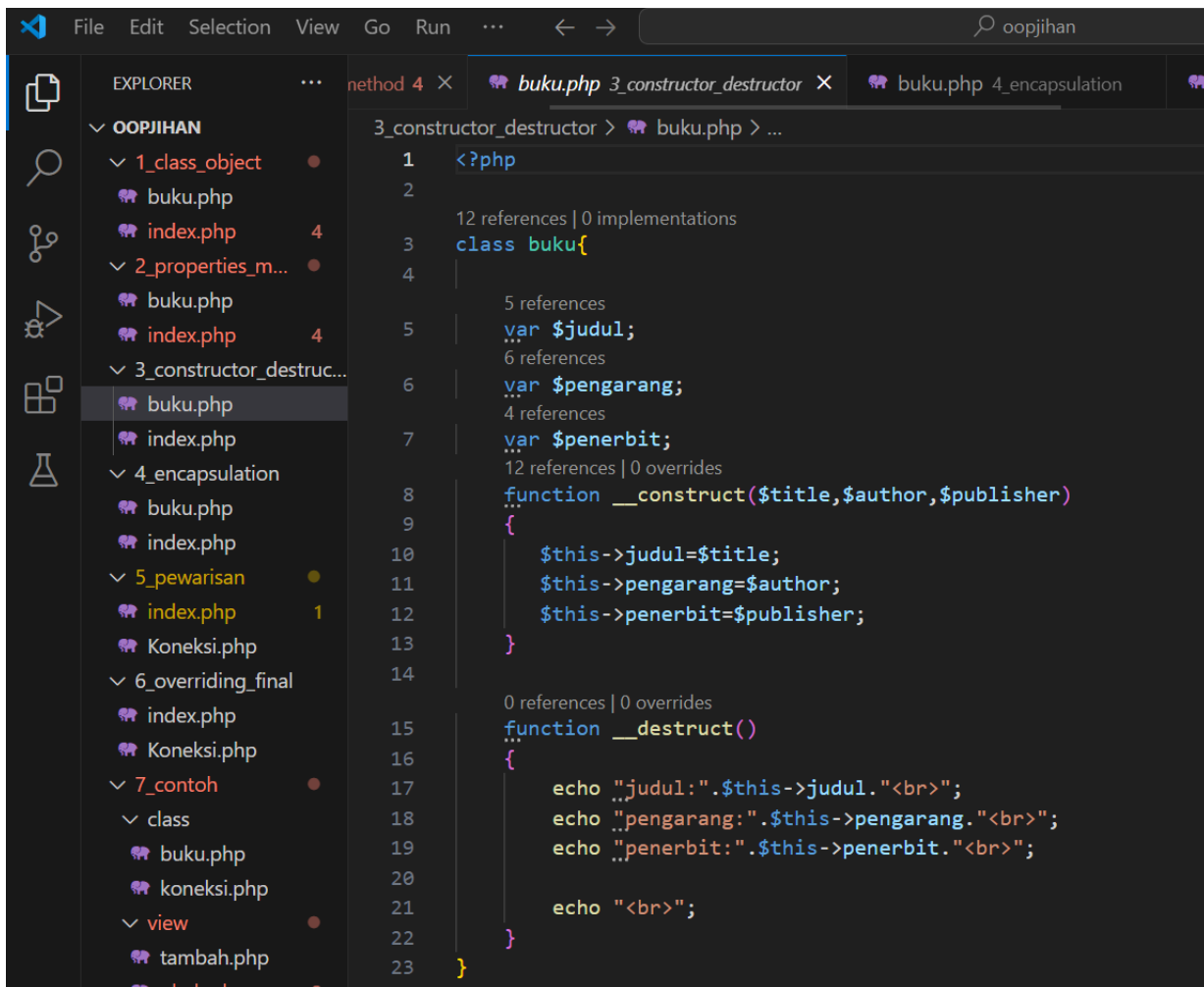
Output:



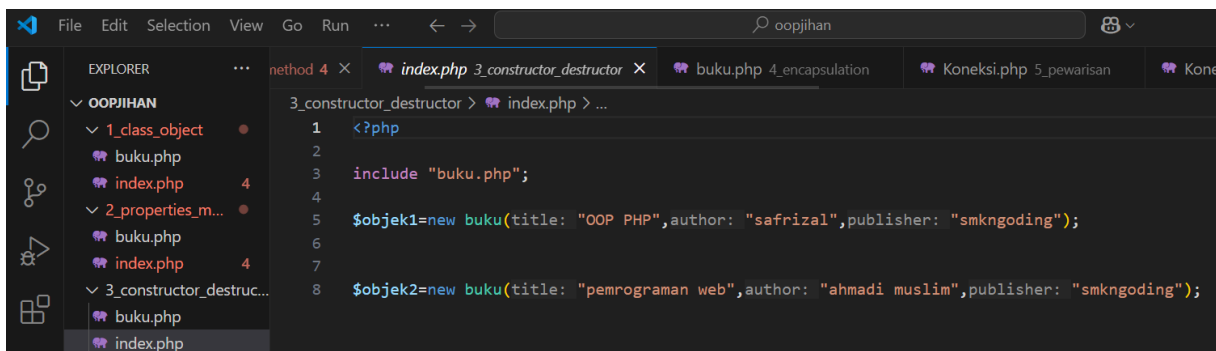
The screenshot shows a web browser window with the address bar displaying `localhost/oopjihan/2_pro...`. The browser shows the output of the PHP script, which is the result of the `cetakinfo()` method calls. The output is as follows:

```
judul:pemrograman web
pengarang:safrizal
penerbit:smkngoding
judul:pemrograman berorientasi objek
pengarang:ahmadi muslim
penerbit:smkngoding
```

### 3. Constructor dan destructor

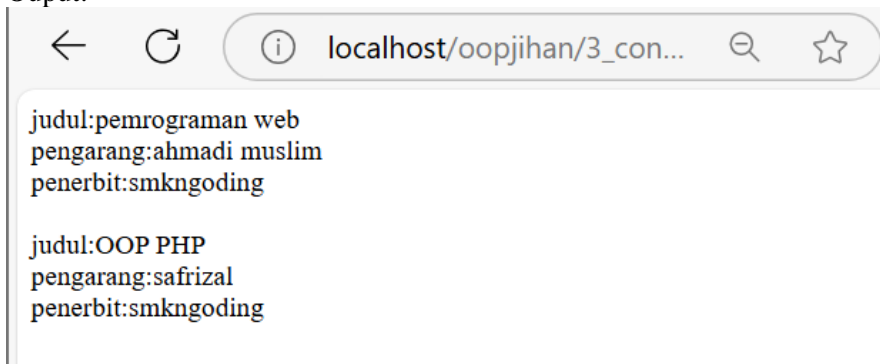


```
1 <?php
2
3 12 references | 0 implementations
4 class buku{
5     5 references
6     var $judul;
7     6 references
8     var $pengarang;
9     4 references
10    var $penerbit;
11    12 references | 0 overrides
12    function __construct($title,$author,$publisher)
13    {
14        $this->judul=$title;
15        $this->pengarang=$author;
16        $this->penerbit=$publisher;
17    }
18
19    0 references | 0 overrides
20    function __destruct()
21    {
22        echo "judul:". $this->judul. "<br>";
23        echo "pengarang:". $this->pengarang. "<br>";
24        echo "penerbit:". $this->penerbit. "<br>";
25
26        echo "<br>";
27    }
28 }
```

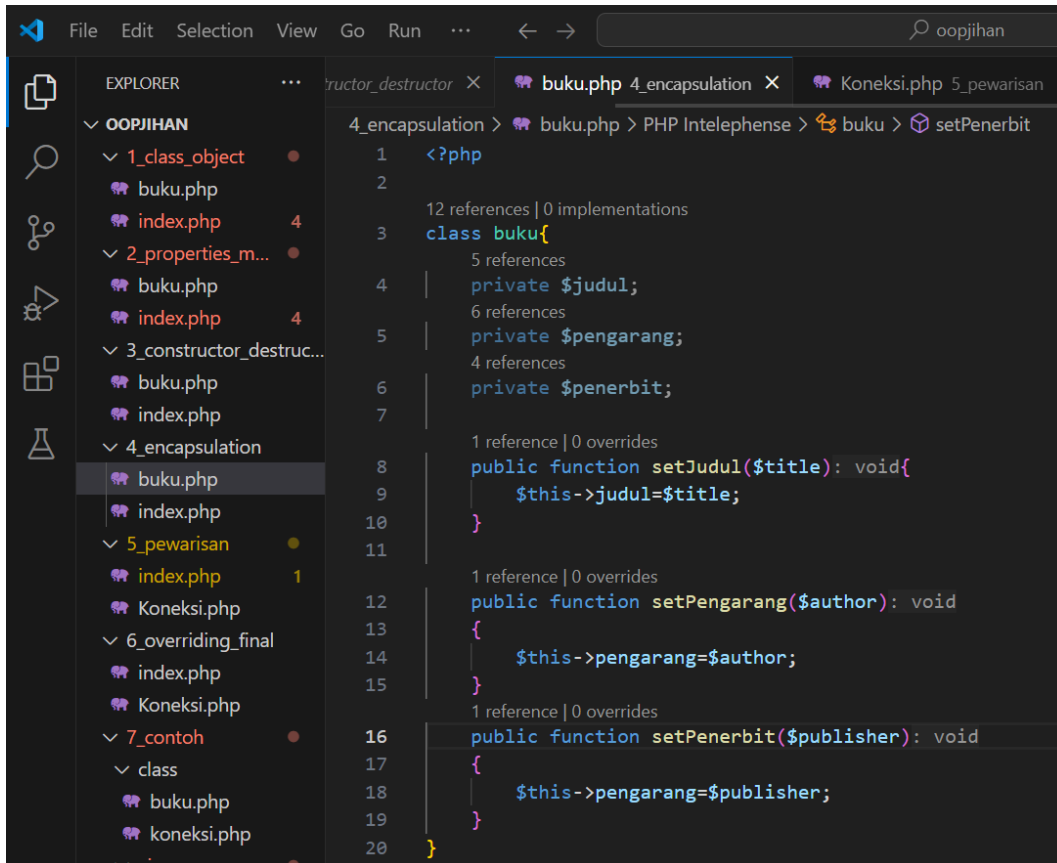


```
1 <?php
2
3 include "buku.php";
4
5 $objek1=new buku(title: "OOP PHP",author: "safrizal",publisher: "smkngoding");
6
7
8 $objek2=new buku(title: "pemrograman web",author: "ahmadi muslim",publisher: "smkngoding");
```

Ouput:

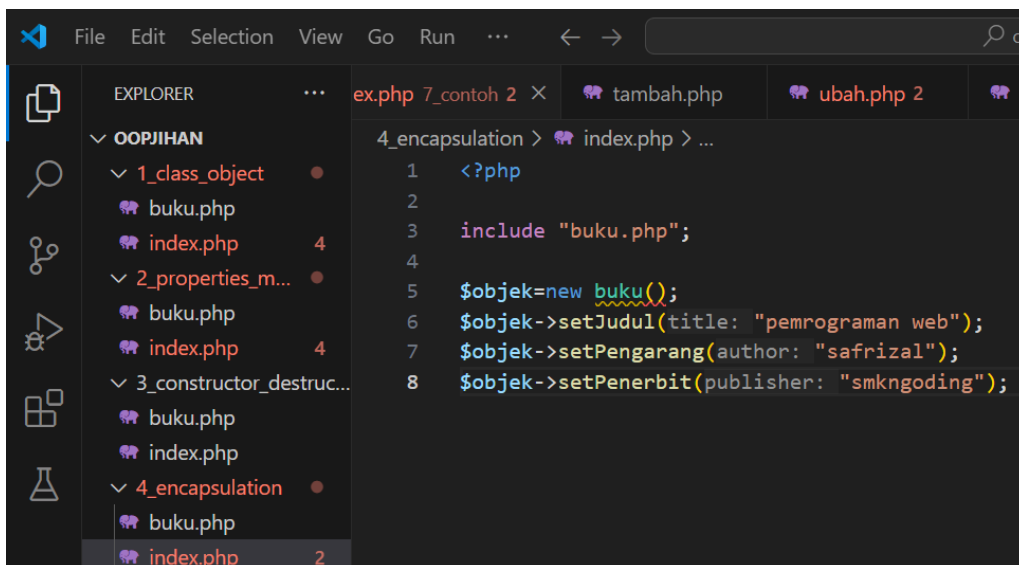


## 4. Encapsulation



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer is expanded to show the '4\_encapsulation' folder, which contains 'buku.php' and 'index.php'. The 'buku.php' file is selected, and its contents are displayed in the main editor. The code defines a class 'buku' with three private properties: '\$judul', '\$pengarang', and '\$penerbit'. It also has three public methods: 'setJudul(\$title)', 'setPengarang(\$author)', and 'setPenerbit(\$publisher)'. Each method sets the corresponding property using '\$this->property = value;'. The Explorer sidebar also shows other folders like '1\_class\_object', '2\_properties\_m...', '3\_constructor\_destruc...', '5\_pewarisan', '6\_overriding\_final', and '7\_contoh'.

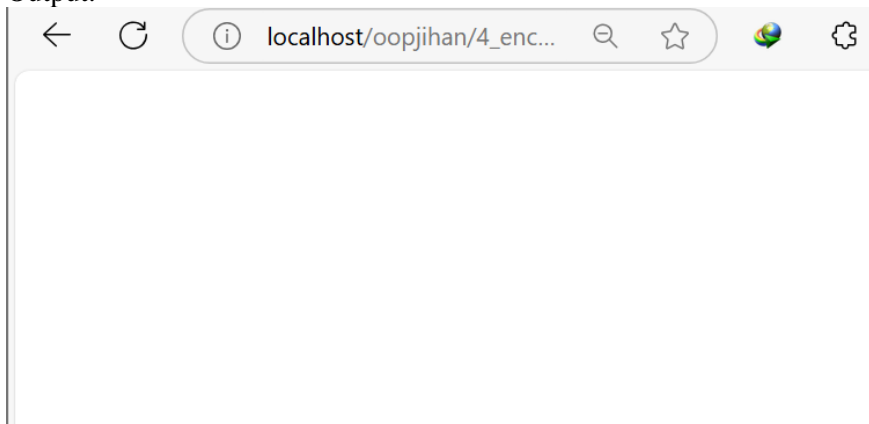
```
1 <?php
2
3 class buku{
4     private $judul;
5     private $pengarang;
6     private $penerbit;
7
8     public function setJudul($title): void{
9         $this->judul=$title;
10    }
11
12    public function setPengarang($author): void
13    {
14        $this->pengarang=$author;
15    }
16    public function setPenerbit($publisher): void
17    {
18        $this->pengarang=$publisher;
19    }
20 }
```



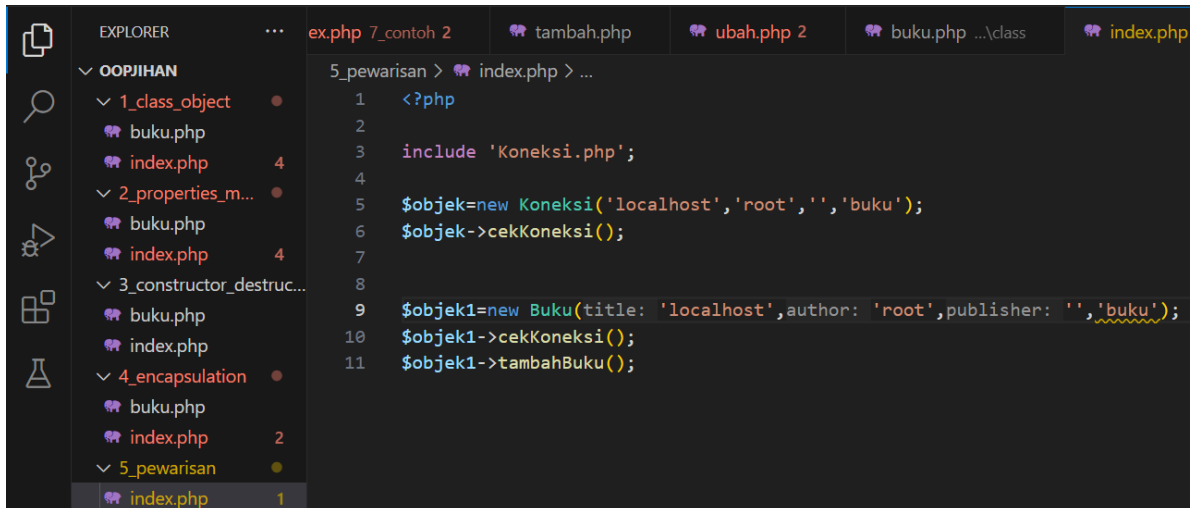
The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer is expanded to show the '4\_encapsulation' folder, which contains 'buku.php' and 'index.php'. The 'index.php' file is selected, and its contents are displayed in the main editor. The code includes 'buku.php' and creates a new object '\$objek' of the 'buku' class. It then calls the 'setJudul', 'setPengarang', and 'setPenerbit' methods on the '\$objek' object with specific values. The Explorer sidebar also shows other folders like '1\_class\_object', '2\_properties\_m...', '3\_constructor\_destruc...', and '7\_contoh'.

```
1 <?php
2
3 include "buku.php";
4
5 $objek=new buku();
6 $objek->setJudul(title: "pemrograman web");
7 $objek->setPengarang(author: "safrizal");
8 $objek->setPenerbit(publisher: "smkngoding");
```

Output:

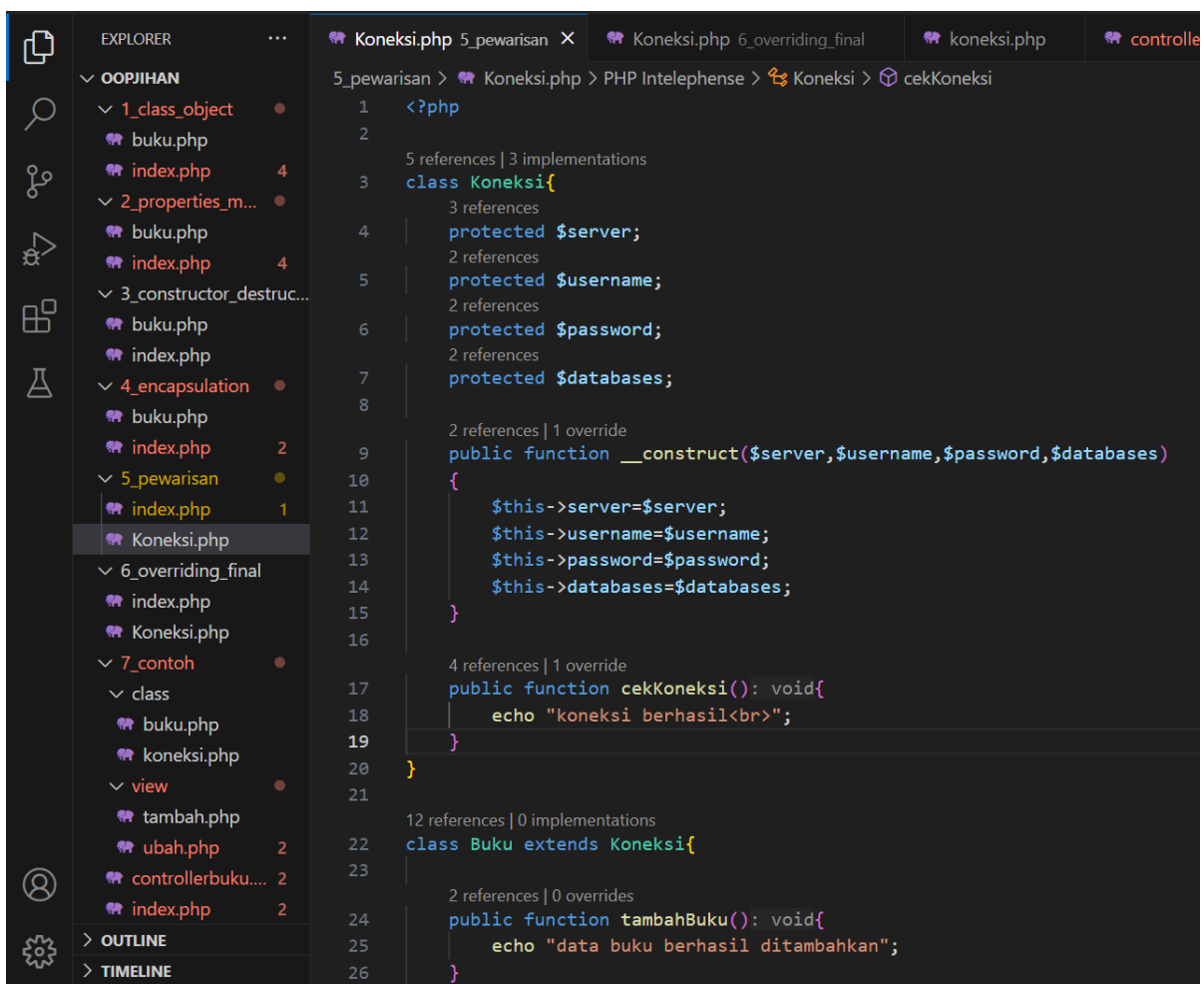


## 5. Pewarisan



The screenshot shows the VS Code interface. The Explorer sidebar on the left displays the file structure for the '5\_pewarisan' directory, including '1\_class\_object', '2\_properties\_m...', '3\_constructor\_destruc...', '4\_encapsulation', and '5\_pewarisan'. The '5\_pewarisan' directory contains 'buku.php' and 'index.php'. The Editor window shows the 'index.php' file with the following PHP code:

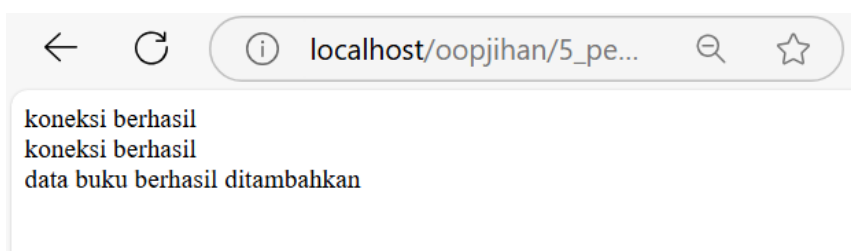
```
1 <?php
2
3 include 'Koneksi.php';
4
5 $objek=new Koneksi('localhost','root','','buku');
6 $objek->cekKoneksi();
7
8
9 $objek1=new Buku(title: 'localhost',author: 'root',publisher: '', 'buku');
10 $objek1->cekKoneksi();
11 $objek1->tambahBuku();
```



The screenshot shows the VS Code interface with the 'Koneksi.php' file selected in the Explorer. The Editor window displays the code for 'Koneksi.php' with PHPDoc annotations indicating references and implementations:

```
1 <?php
2
3 class Koneksi{
4     protected $server;
5     protected $username;
6     protected $password;
7     protected $databases;
8
9     public function __construct($server,$username,$password,$databases)
10     {
11         $this->server=$server;
12         $this->username=$username;
13         $this->password=$password;
14         $this->databases=$databases;
15     }
16
17     public function cekKoneksi(): void{
18         echo "koneksi berhasil<br>";
19     }
20 }
21
22 class Buku extends Koneksi{
23
24     public function tambahBuku(): void{
25         echo "data buku berhasil ditambahkan";
26     }
27 }
```

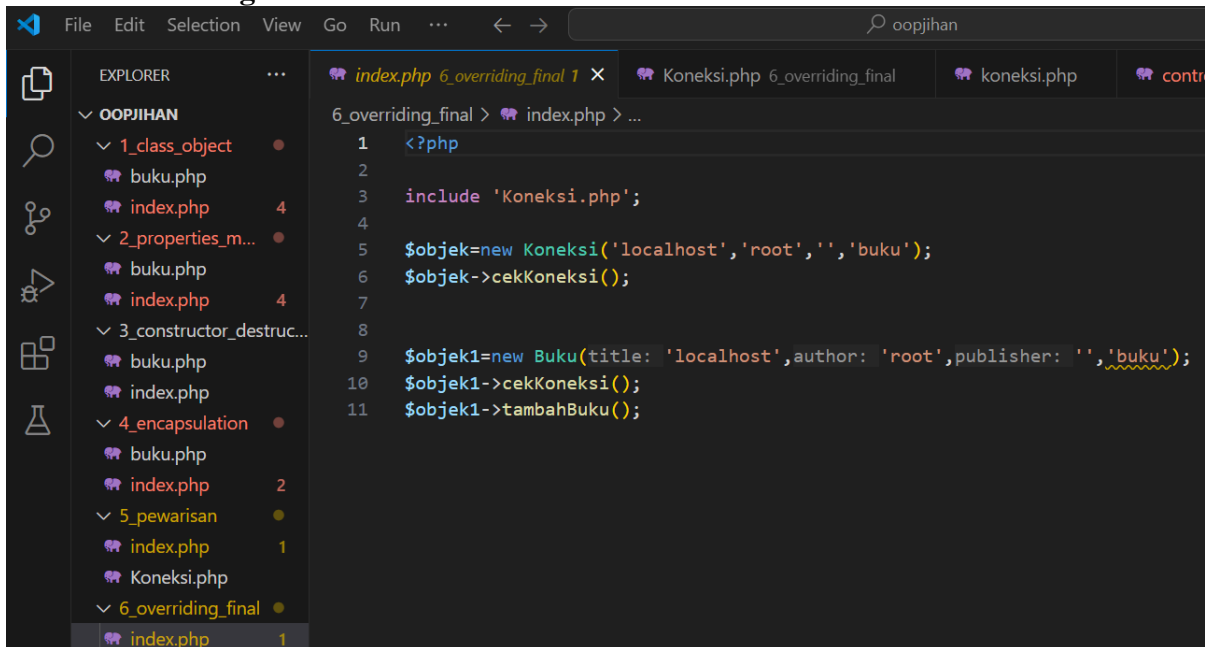
Output:



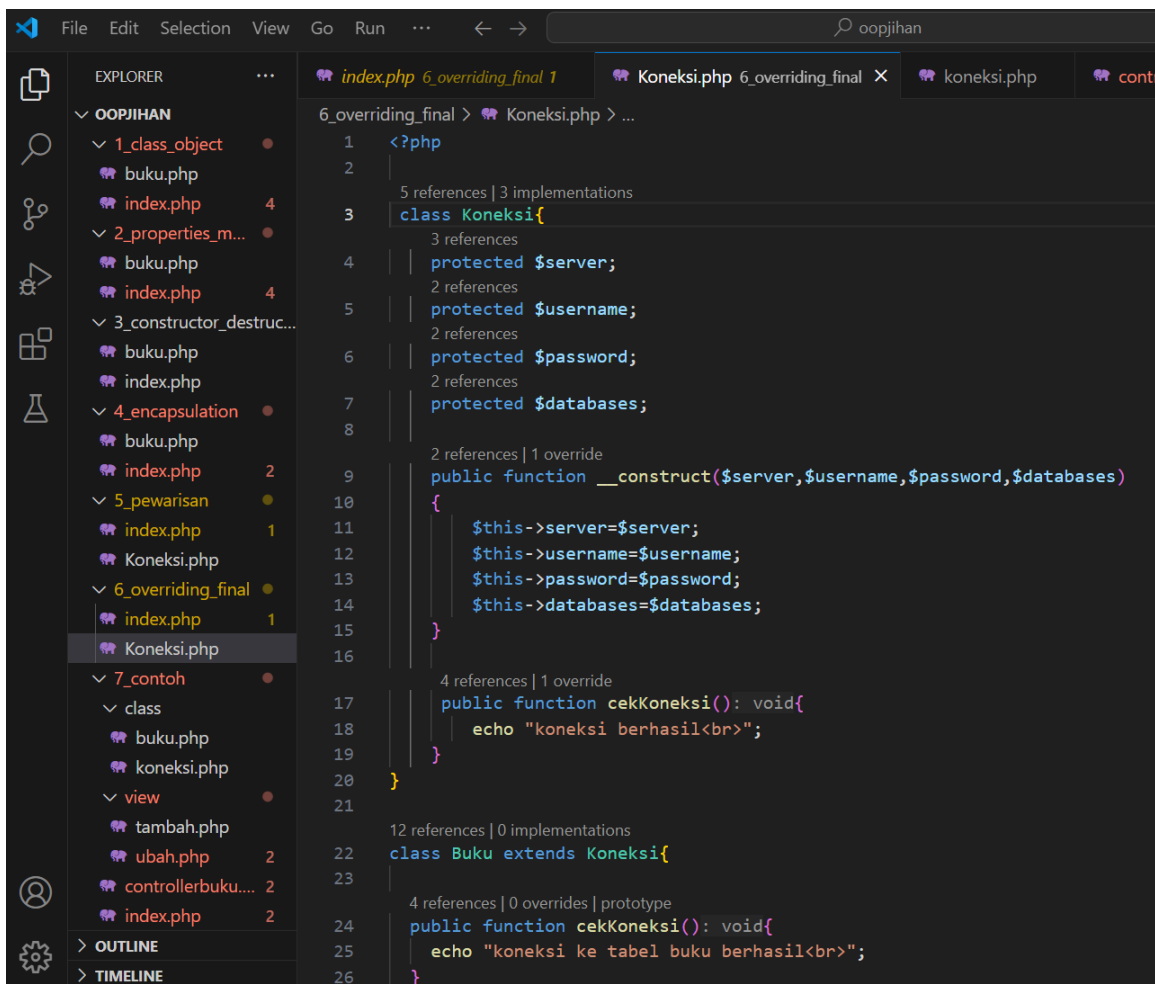
The screenshot shows a web browser window with the address bar displaying 'localhost/oopjihan/5\_pe...'. The browser content area shows the output of the PHP script:

```
koneksi berhasil
koneksi berhasil
data buku berhasil ditambahkan
```

## 6. Overriding final

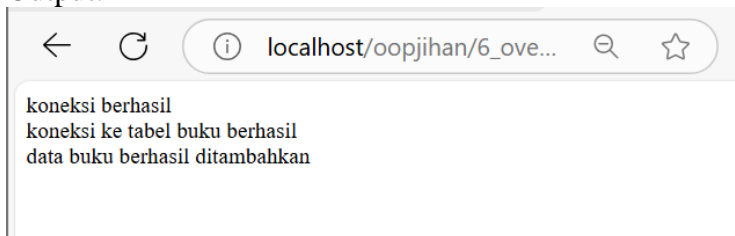


```
1 <?php
2
3 include 'Koneksi.php';
4
5 $objek=new Koneksi('localhost','root','','buku');
6 $objek->cekKoneksi();
7
8
9 $objek1=new Buku(title: 'localhost',author: 'root',publisher: '','buku');
10 $objek1->cekKoneksi();
11 $objek1->tambahBuku();
```



```
1 <?php
2
3 class Koneksi{
4     protected $server;
5     protected $username;
6     protected $password;
7     protected $databases;
8
9     public function __construct($server,$username,$password,$databases)
10     {
11         $this->server=$server;
12         $this->username=$username;
13         $this->password=$password;
14         $this->databases=$databases;
15     }
16
17     public function cekKoneksi(): void{
18         echo "koneksi berhasil<br>";
19     }
20 }
21
22 class Buku extends Koneksi{
23
24     public function cekKoneksi(): void{
25         echo "koneksi ke tabel buku berhasil<br>";
26     }
27 }
```

Output:

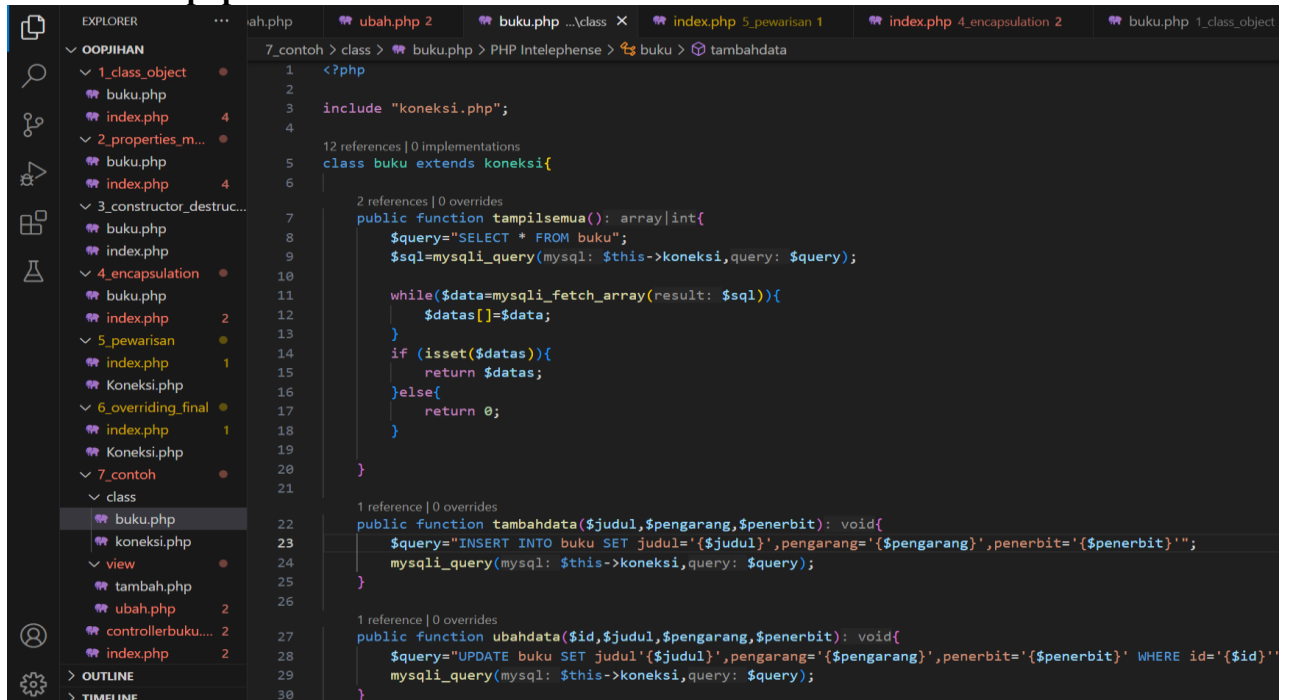




## 7. Implementasi/contoh

Folder class:

- **Buku.php**



```
<?php
include "koneksi.php";

class buku extends koneksi{

    2 references | 0 overrides
    public function tampilsemua(): array|int{
        $query="SELECT * FROM buku";
        $sql=mysqli_query(mysql: $this->koneksi,query: $query);

        while($data=mysqli_fetch_array(result: $sql)){
            $datas[]=$data;
        }
        if (isset($datas)){
            return $datas;
        }else{
            return 0;
        }
    }

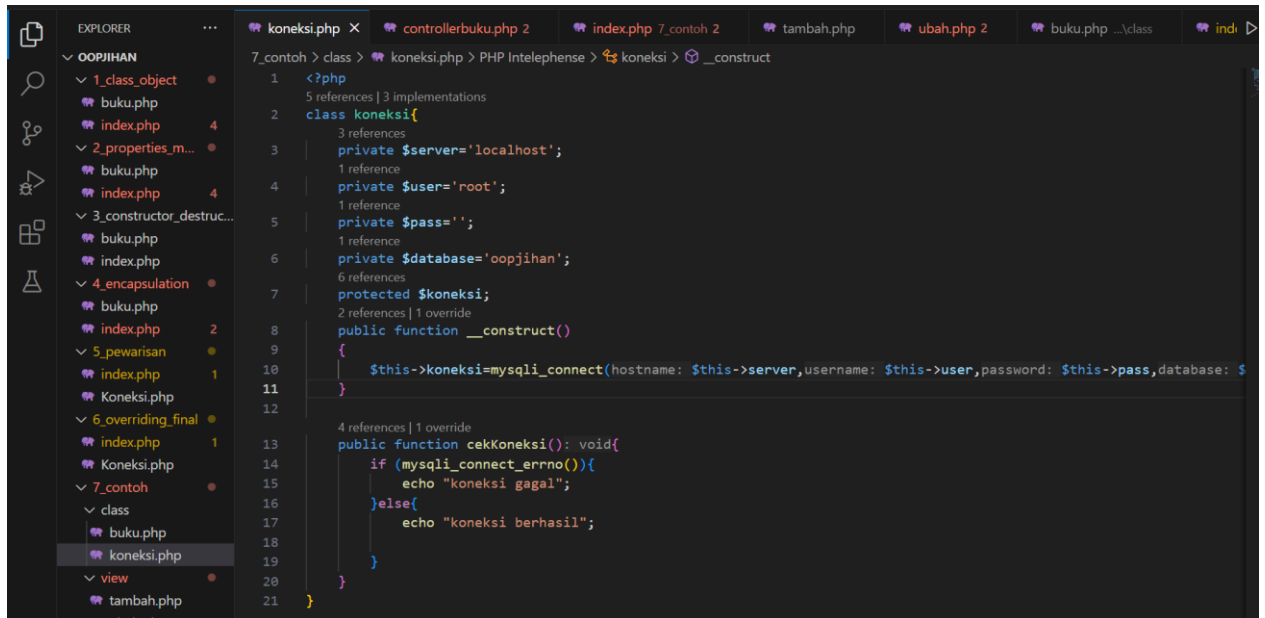
    1 reference | 0 overrides
    public function tambahdata($judul,$pengarang,$penerbit): void{
        $query="INSERT INTO buku SET judul='{$judul}',pengarang='{$pengarang}',penerbit='{$penerbit}'";
        mysqli_query(mysql: $this->koneksi,query: $query);
    }

    1 reference | 0 overrides
    public function ubahdata($id,$judul,$pengarang,$penerbit): void{
        $query="UPDATE buku SET judul='{$judul}',pengarang='{$pengarang}',penerbit='{$penerbit}' WHERE id='{$id}'";
        mysqli_query(mysql: $this->koneksi,query: $query);
    }
}
```

```
1 reference | 0 overrides
public function hapusdata($id): void{
    $query="DELETE FROM buku WHERE id='{$id}'";
    mysqli_query(mysql: $this->koneksi,query: $query);
}

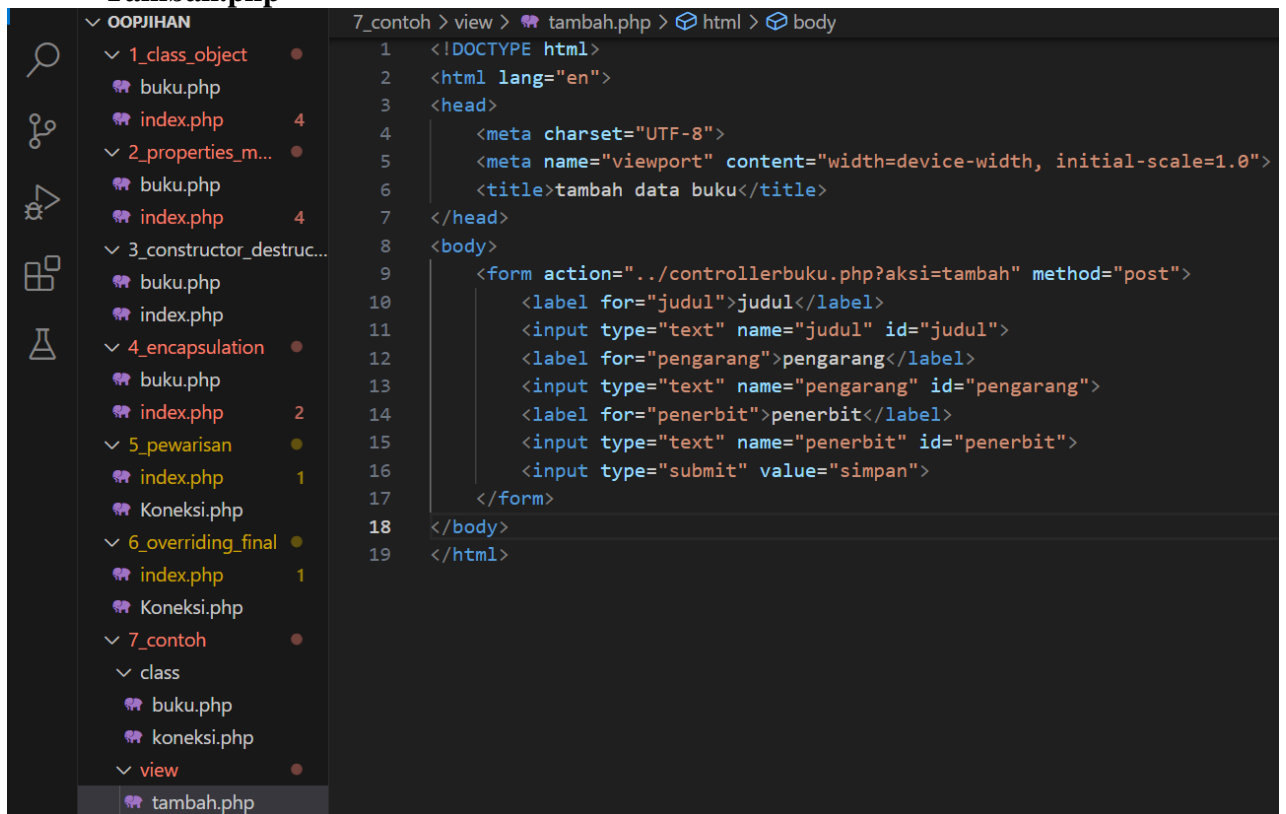
1 reference | 0 overrides
public function caridatabyid($id): array|bool|null{
    $query="SELECT * FROM buku WHERE id='{$id}'";
    $sql=mysqli_query(mysql: $this->koneksi,query: $query);
    $data=mysqli_fetch_assoc(result: $sql);
    return $data;
}
}
```

- **Koneksi.php**



## Folder view:

- **Tambah.php**



- **Ubah.php**

EXPLORER

ambah.php buku.php ...class ubah.php 1 x index.php 5\_pewarisan index.php 4\_encapsulation

7\_contoh > view > ubah.php > html > body

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>tambah data buku</title>
7 </head>
8 <body>
9   <?php
10     include "../class/buku.php";
11     $objek=new buku();
12     $data=$objek->caridatabyid($_GET['id'])
13   ?>
14   <form action="../controllerbuku.php?aksi=ubah"
15     method="post">
16     <input type="hidden" name="id" value="<?=$data['id']?>">
17     <label for="judul">judul</label>
18     <input type="text" name="judul" id="judul" value="<?=$data['judul']?>">
19     <label for="pengarang">pengarang</label>
20     <input type="text" name="pengarang" id="pengarang" value="<?=$data['pengarang']?>">
21     <label for="penerbit">penerbit</label>
22     <input type="text" name="penerbit" id="penerbit" value="<?=$data['penerbit']?>">
23     <input type="submit" value="simpan">
24   </form>
25 </body>
26 </html>
```

1\_class\_object index.php

2\_properties\_method buku.php index.php

3\_constructor\_destruc... buku.php index.php

4\_encapsulation buku.php index.php

5\_pewarisan index.php Koneksi.php

6\_overriding\_final index.php Koneksi.php

7\_contoh

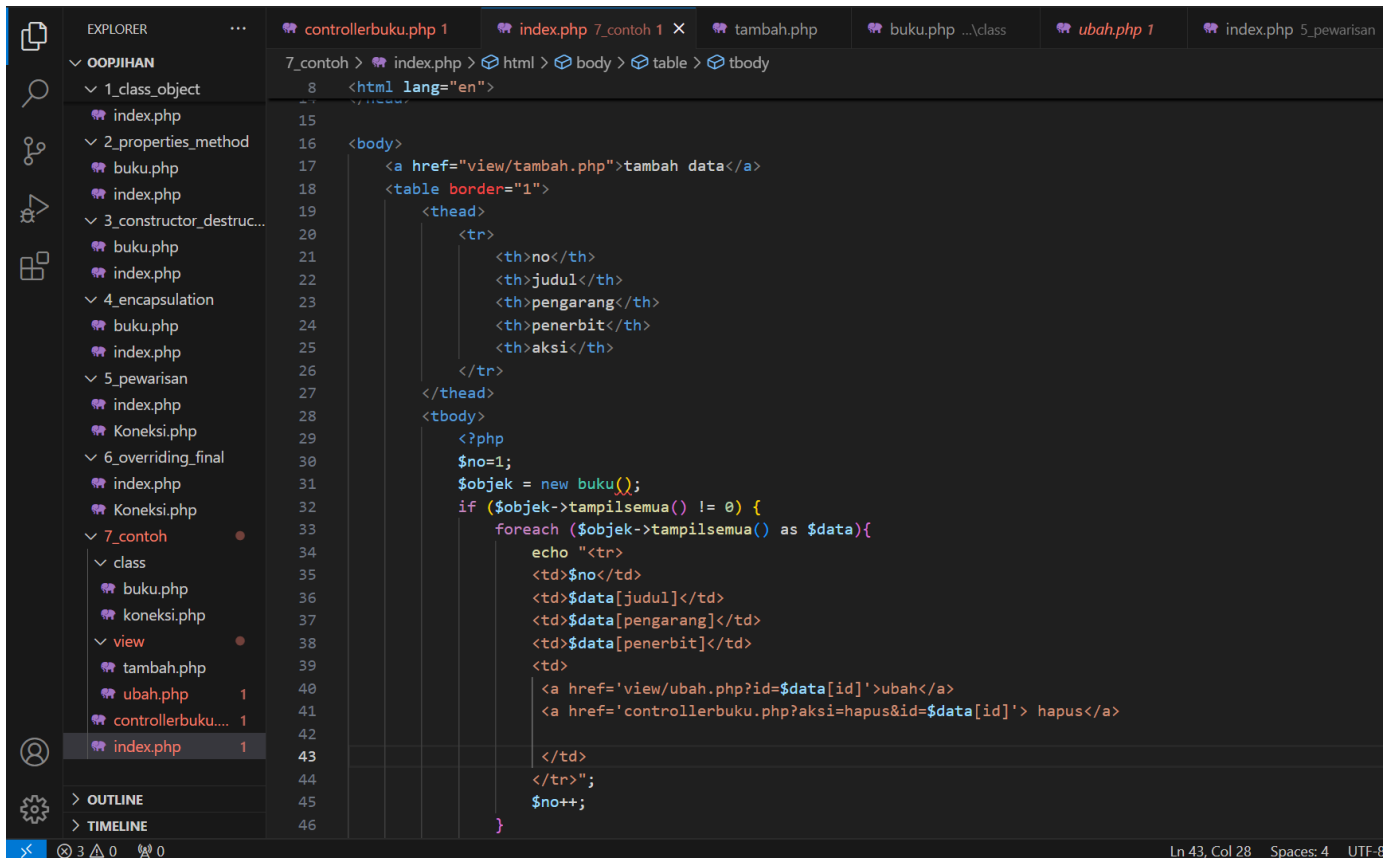
class buku.php koneksi.php

view tambah.php ubah.php 1

## Folder controllerbuku.php

```
1  <?php
2
3  include "class/buku.php";
4  $buku=new buku();
5  $aksi=$_GET['aksi'];
6
7  if($aksi=='tambah'){
8      $buku->tambahdata(
9          $_POST['judul'],
10         $_POST['pengarang'],
11         $_POST['penerbit']
12     );
13
14 }elseif($aksi=='hapus'){
15     $buku->hapusdata($_GET['id']);
16
17 }elseif($aksi=='ubah'){
18     $buku->ubahdata(
19         $_POST['id'],
20         $_POST['judul'],
21         $_POST['pengarang'],
22         $_POST['penerbit']
23     );
24 }
25 header("location:index.php");
```

## Folder index.php



```
7_contoh > index.php > html > body > table > tbody
8 <html lang="en">
15
16 <body>
17 <a href="view/tambah.php">tambah data</a>
18 <table border="1">
19 <thead>
20 <tr>
21 <th>no</th>
22 <th>judul</th>
23 <th>pengarang</th>
24 <th>penerbit</th>
25 <th>aksi</th>
26 </tr>
27 </thead>
28 <tbody>
29 <?php
30 $no=1;
31 $objek = new buku();
32 if ($objek->tampilsemua() != 0) {
33     foreach ($objek->tampilsemua() as $data){
34         echo "<tr>
35             <td>$no</td>
36             <td>$data[judul]</td>
37             <td>$data[pengarang]</td>
38             <td>$data[penerbit]</td>
39             <td>
40                 <a href='view/ubah.php?id=$data[id]''>ubah</a>
41                 <a href='controllerbuku.php?aksi=hapus&id=$data[id]''> hapus</a>
42             </td>
43         </tr>";
44         $no++;
45     }
46 }
```

Output:

