PROJECT OBJECT ORIENTED PROGRAMMING

(PEMROGRAMAN BERIORIENTASI OBJEK)

OLEH:

JIHAN SYAFITRI

NISN. 0083055957

REKAYASA PERANGKAT LUNAK

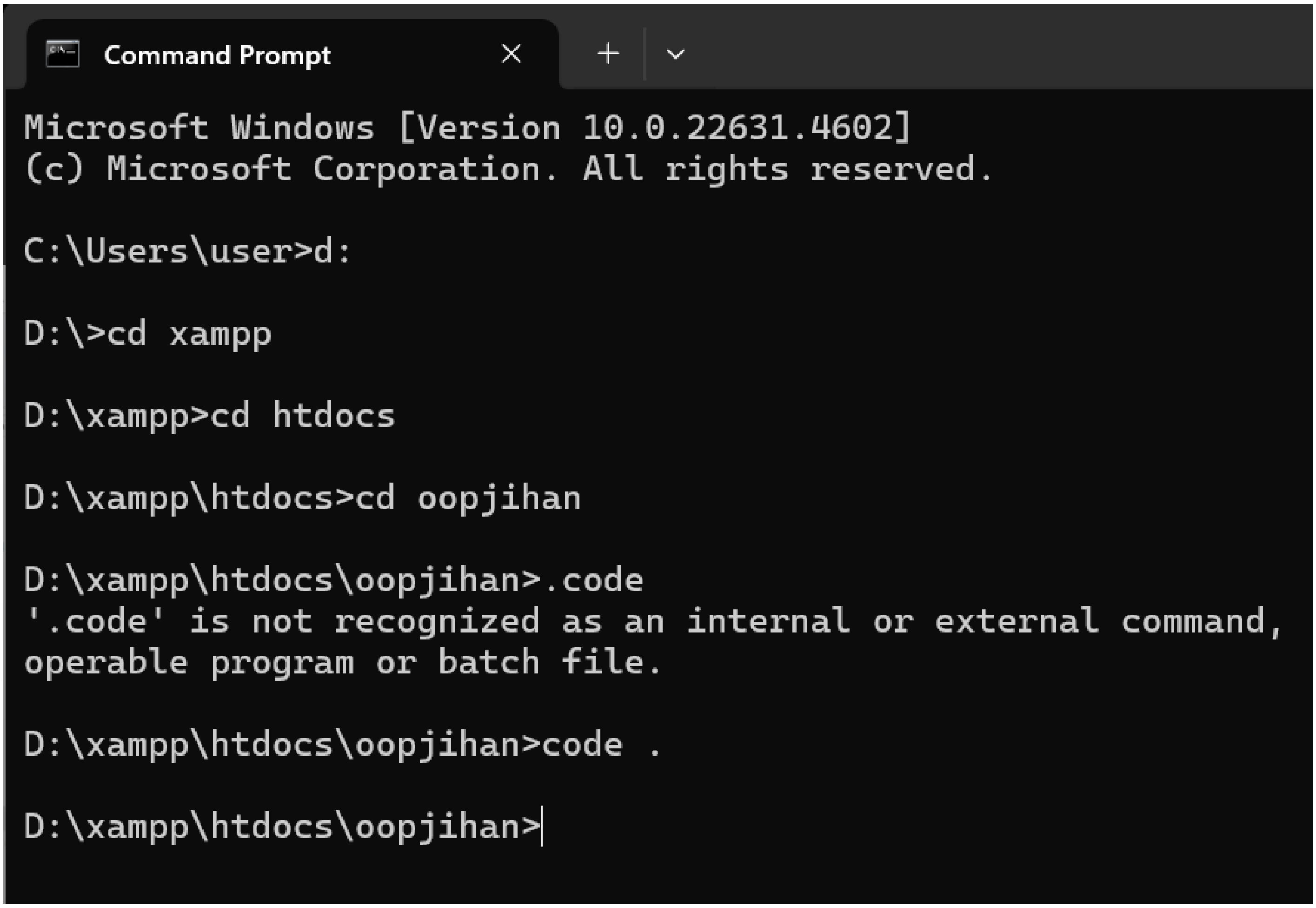
SMK NEGERI 1 KARANG BARU

PEMERINTAH PROVINSI ACEH

2025

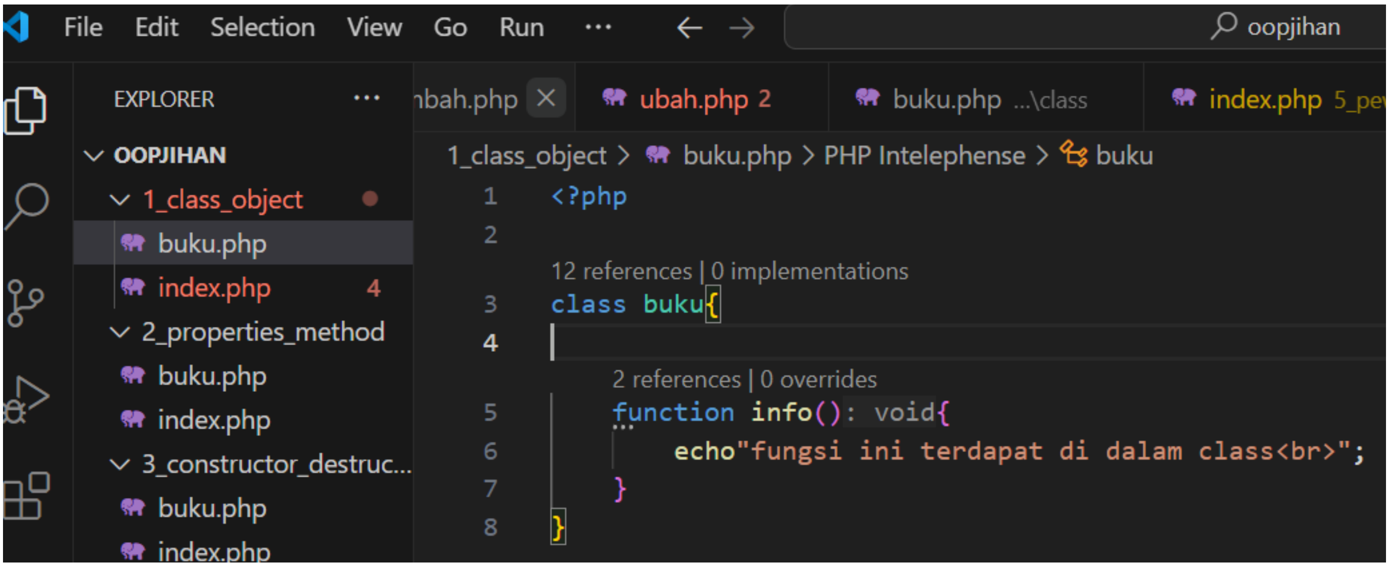
a. Apa itu OOP?

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

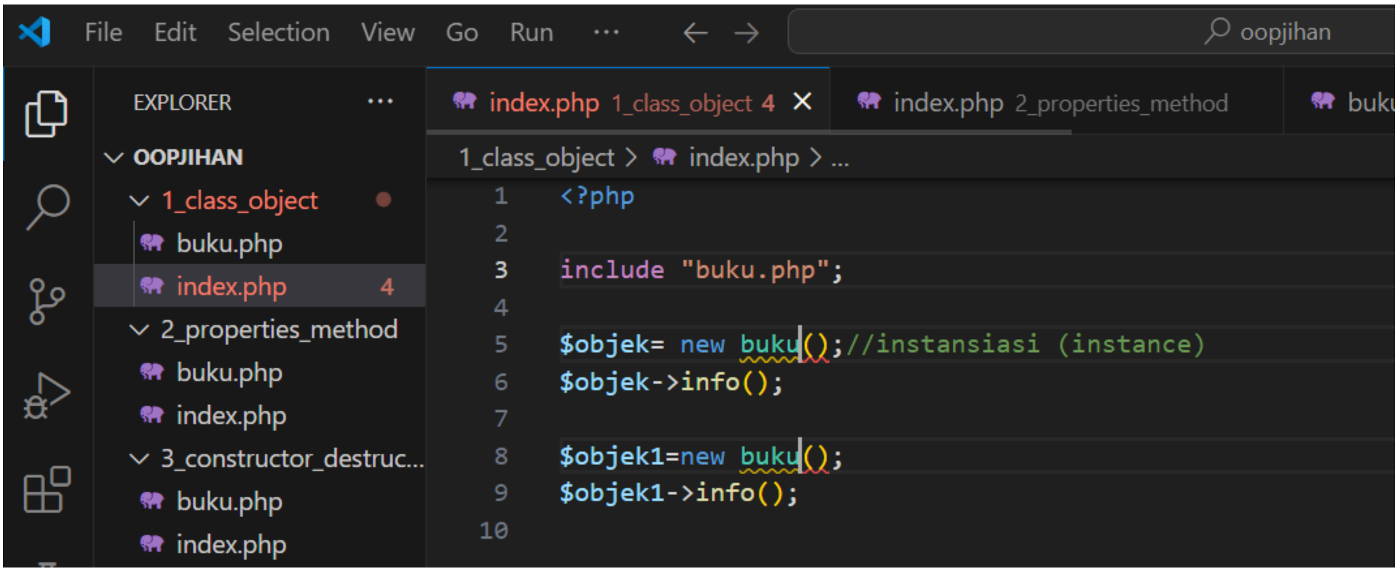


```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

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**

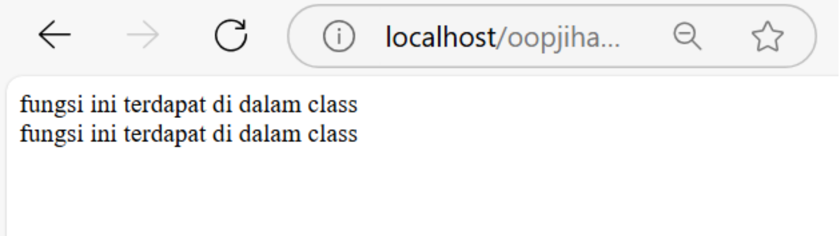


```php
<?php

class buku{

    function info(): void{
        echo"fungsi ini terdapat di dalam class<br>";
    }
}
```



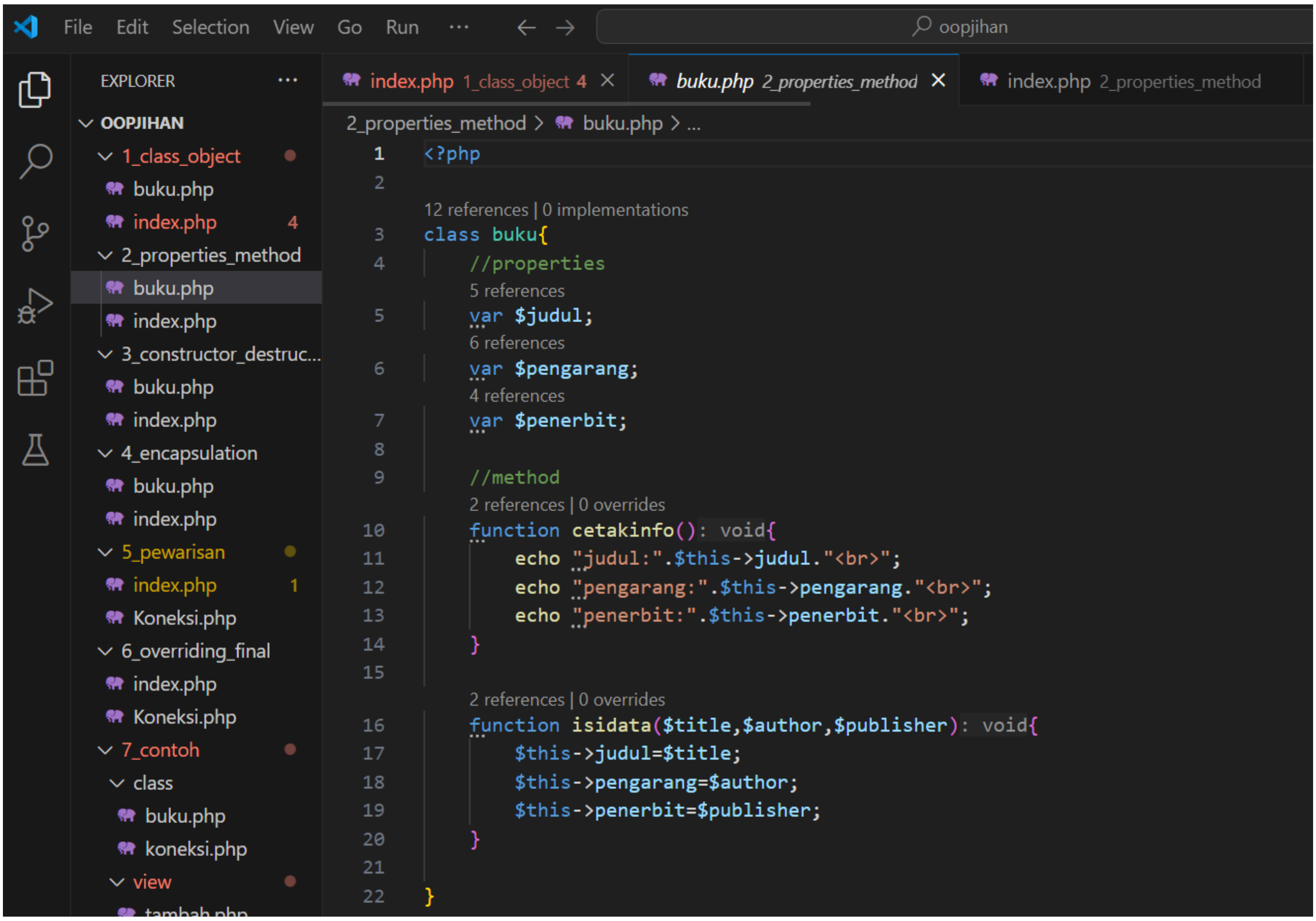
```php
<?php

include "buku.php";

$objek= new buku();//instansiasi (instance)
$objek->info();

$objek1=new buku();
$objek1->info();

```

Output:

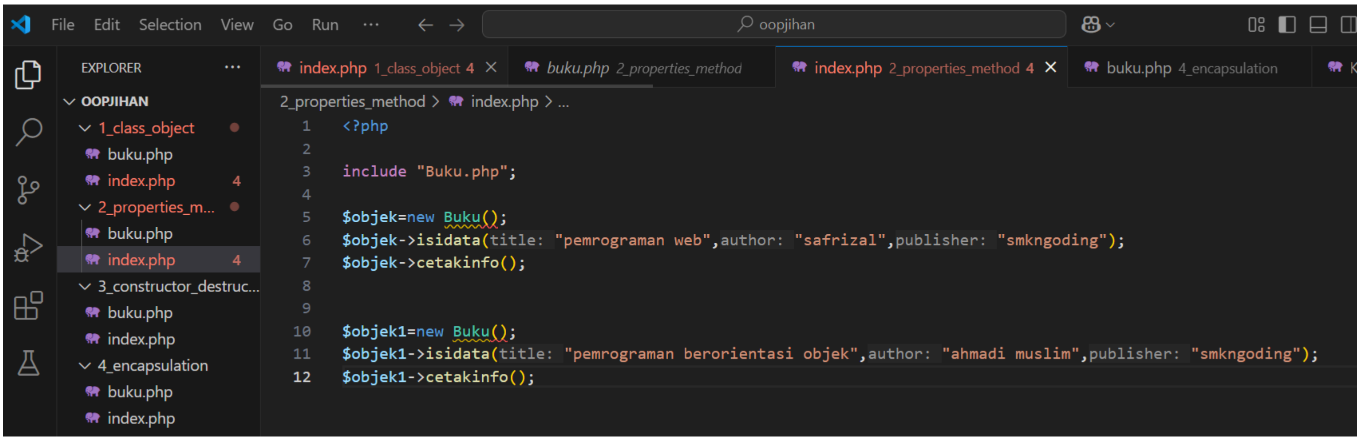


```
fungsi ini terdapat di dalam class
fungsi ini terdapat di dalam class
```

2. Properties dan method

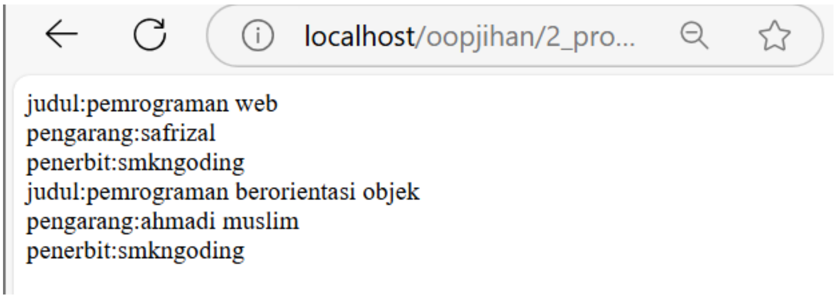


```php
<?php

class buku{
    //properties
    var $judul;
    var $pengarang;
    var $penerbit;

    //method
    function cetakinfo(): void{
        echo "judul:".$this->judul."<br>";
        echo "pengarang:".$this->pengarang."<br>";
        echo "penerbit:".$this->penerbit."<br>";
    }

    function isidata($title,$author,$publisher): void{
        $this->judul=$title;
        $this->pengarang=$author;
        $this->penerbit=$publisher;
    }
}
```



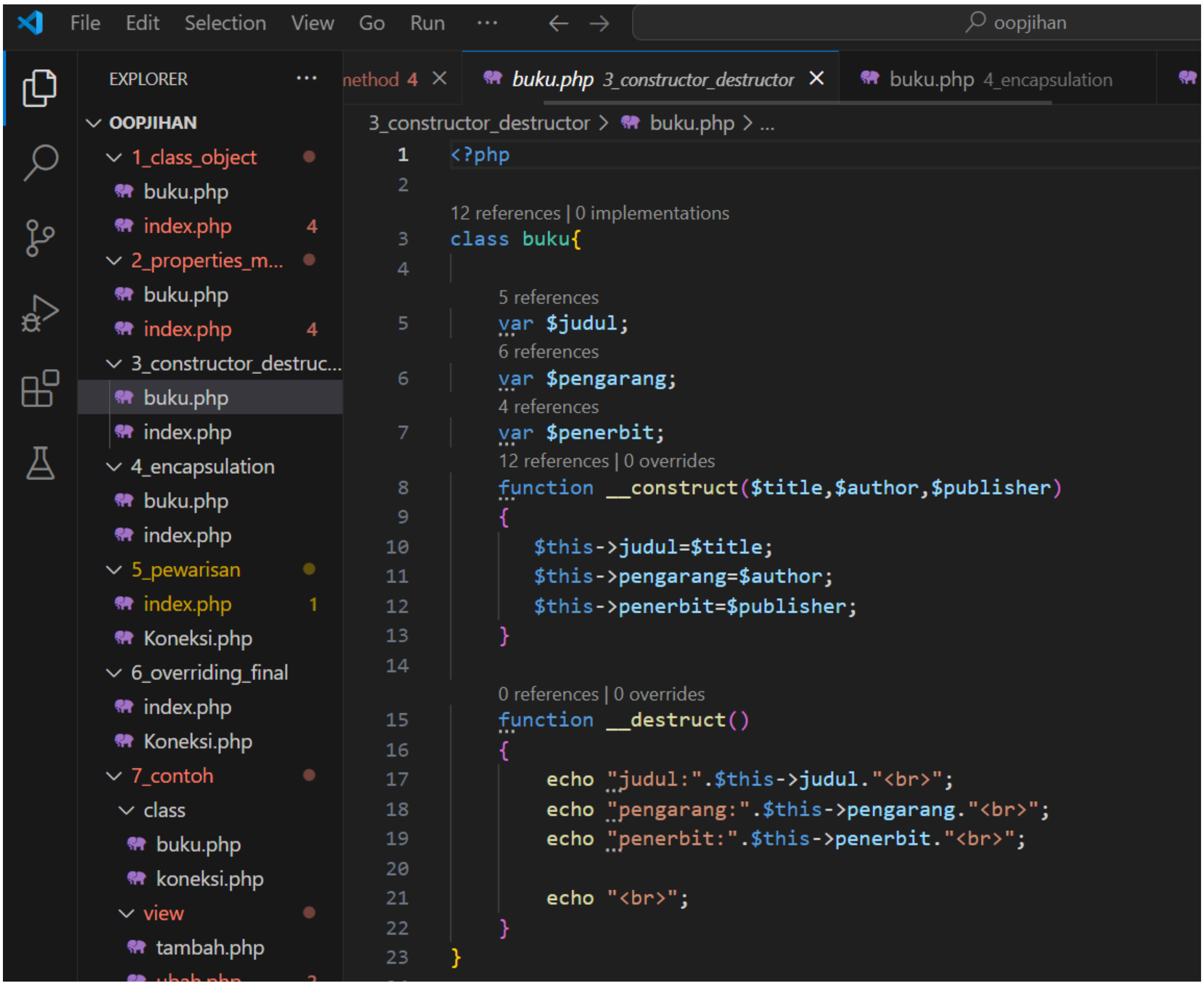
```php
<?php

include "Buku.php";

$objek=new Buku();
$objek->isidata(title: "pemrograman web",author: "safrizal",publisher: "smkngoding");
$objek->cetakinfo();


$objek1=new Buku();
$objek1->isidata(title: "pemrograman berorientasi objek",author: "ahmadi muslim",publisher: "smkngoding");
$objek1->cetakinfo();
```

Output:

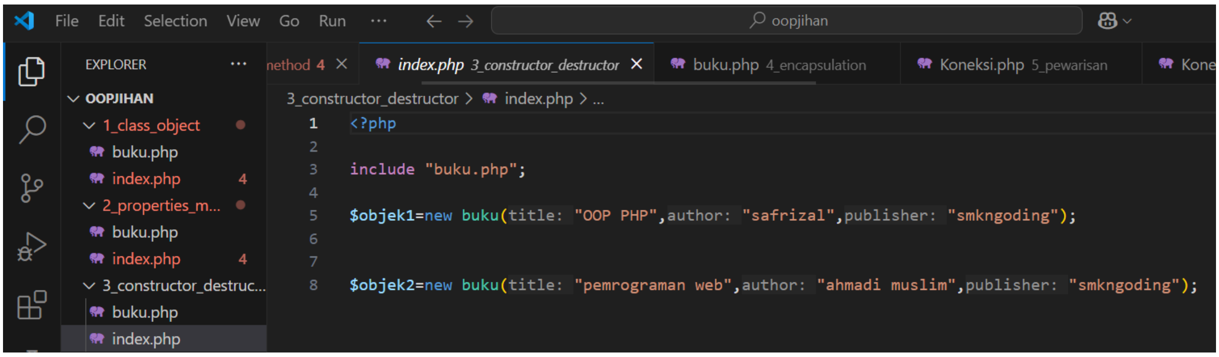


```
judul:pemrograman web
pengarang:safrizal
penerbit:smkngoding
judul:pemrograman berorientasi objek
pengarang:ahmadi muslim
penerbit:smkngoding
```

3. Constructor dan destructor

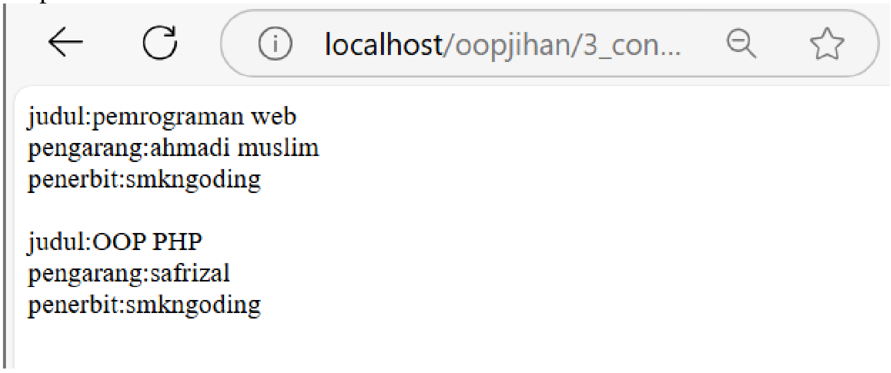


```php
<?php

12 references | 0 implementations
class buku{

    5 references
    var $judul;
    6 references
    var $pengarang;
    4 references
    var $penerbit;
    12 references | 0 overrides
    function __construct($title,$author,$publisher)
    {
       $this->judul=$title;
       $this->pengarang=$author;
       $this->penerbit=$publisher;
    }

    0 references | 0 overrides
    function __destruct()
    {
        echo "judul:".$this->judul."<br>";
        echo "pengarang:".$this->pengarang."<br>";
        echo "penerbit:".$this->penerbit."<br>";

        echo "<br>";
    }
}
```



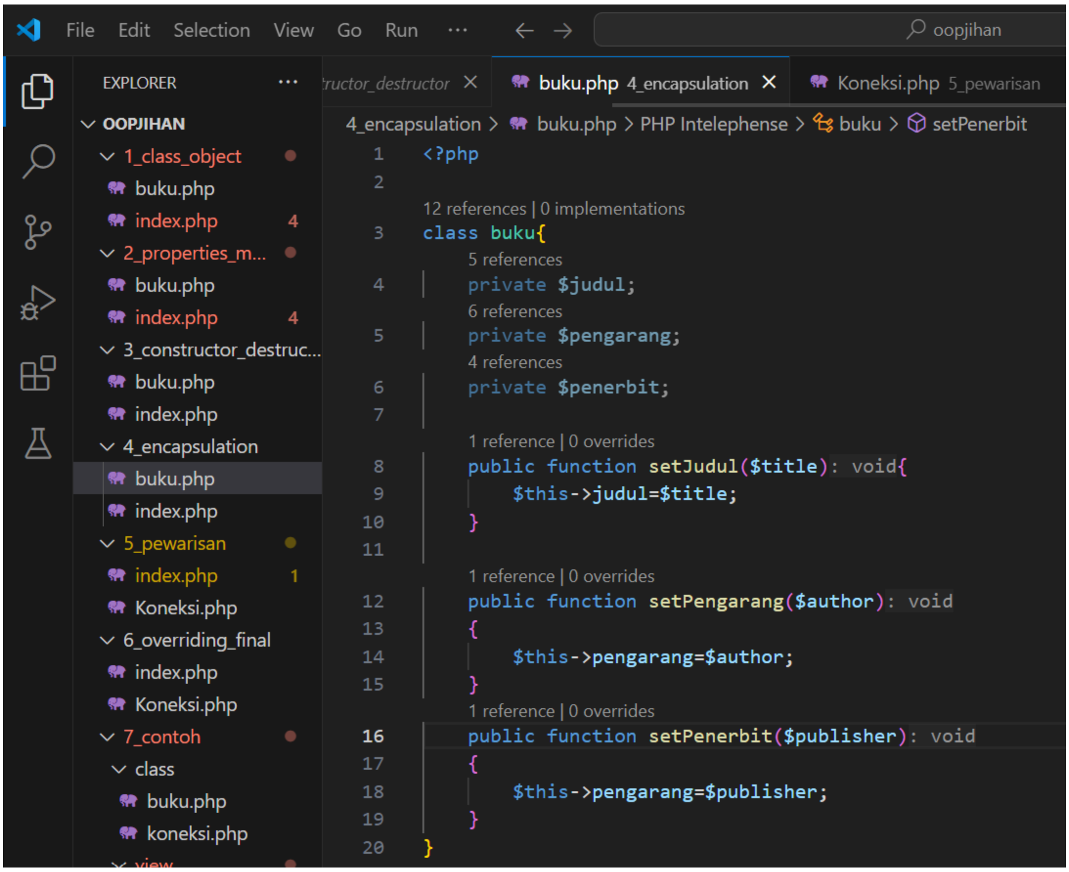
```php
<?php

include "buku.php";

$objek1=new buku(title: "OOP PHP",author: "safrizal",publisher: "smkngoding");


$objek2=new buku(title: "pemrograman web",author: "ahmadi muslim",publisher: "smkngoding");
```

Ouput:

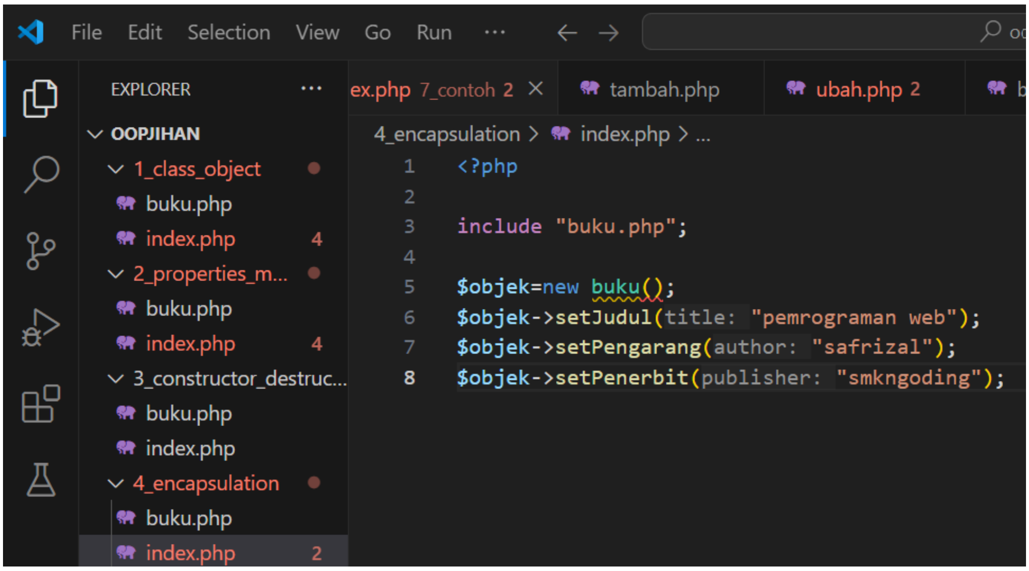


```
judul:pemrograman web
pengarang:ahmadi muslim
penerbit:smkngoding

judul:OOP PHP
pengarang:safrizal
penerbit:smkngoding
```

4. Encapsulation

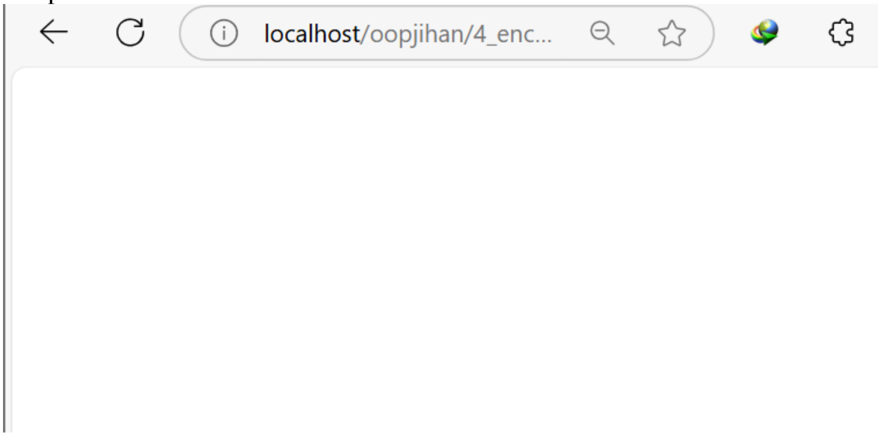


```php
<?php

12 references | 0 implementations
class buku{
    5 references
    private $judul;
    6 references
    private $pengarang;
    4 references
    private $penerbit;

    1 reference | 0 overrides
    public function setJudul($title): void{
        $this->judul=$title;
    }

    1 reference | 0 overrides
    public function setPengarang($author): void
    {
        $this->pengarang=$author;
    }
    1 reference | 0 overrides
    public function setPenerbit($publisher): void
    {
        $this->pengarang=$publisher;
    }
}
```

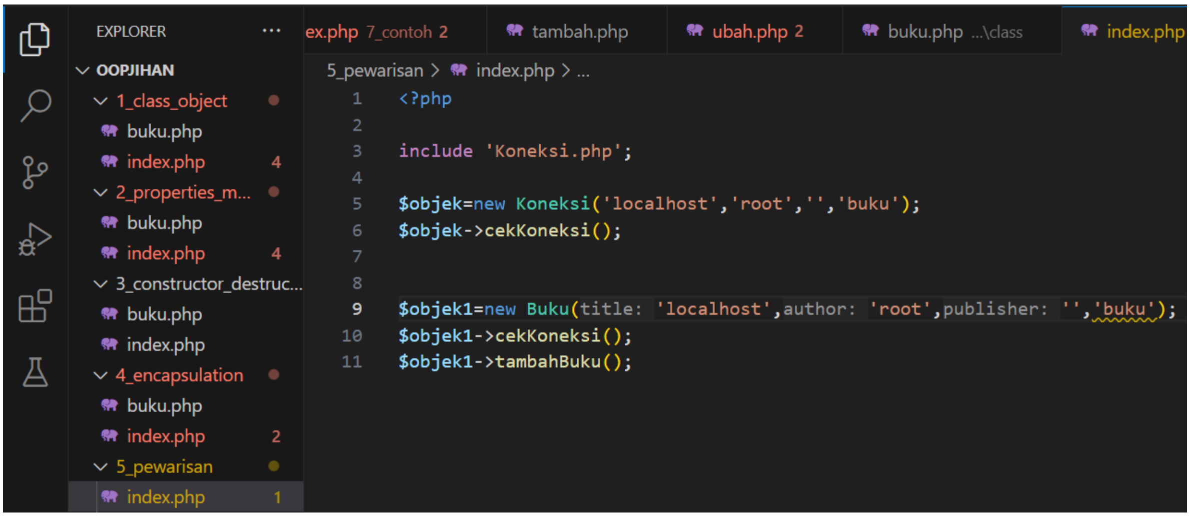


```php
<?php

include "buku.php";

$objek=new buku();
$objek->setJudul(title: "pemrograman web");
$objek->setPengarang(author: "safrizal");
$objek->setPenerbit(publisher: "smkngoding");
```

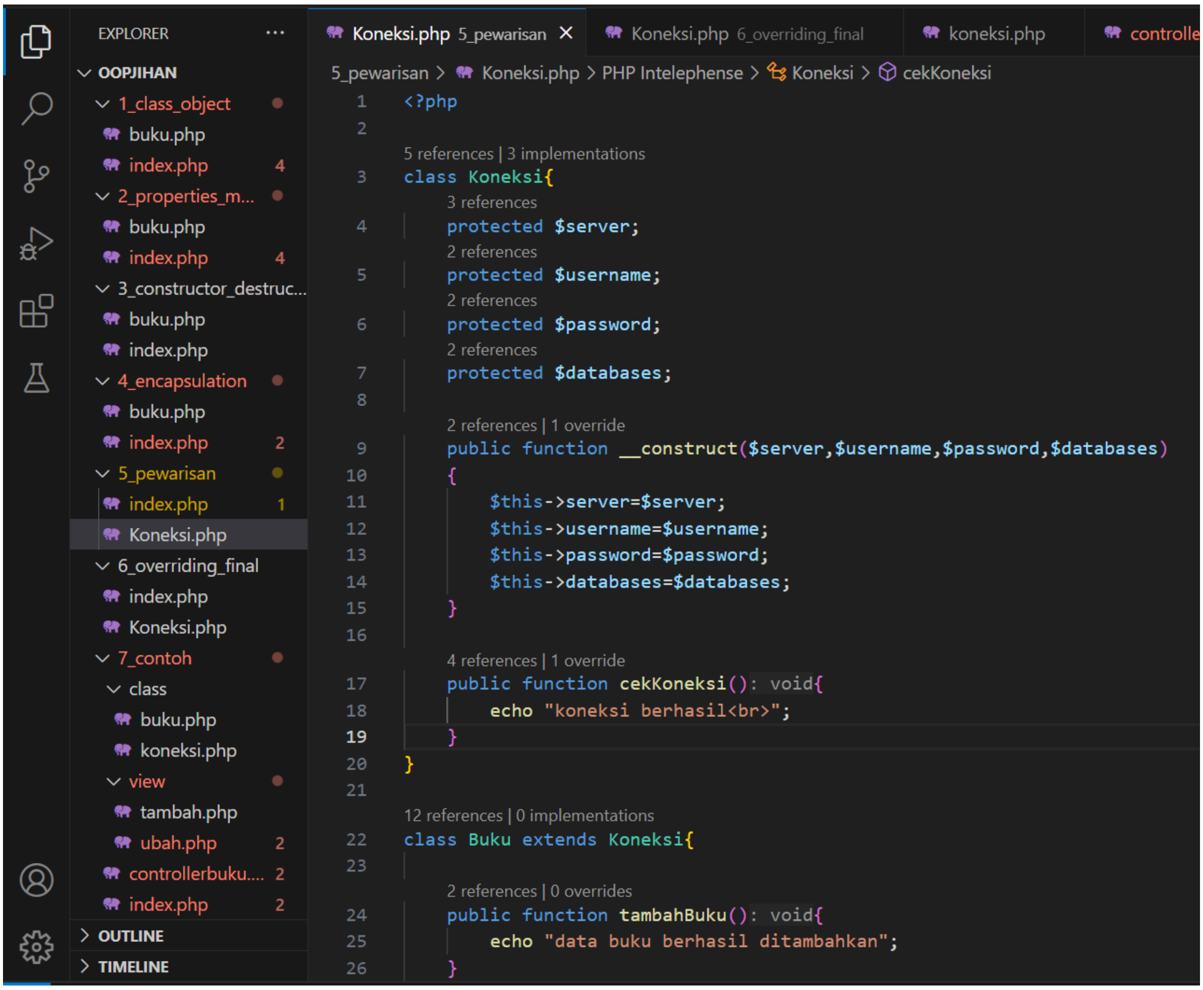
Output:



5. Pewarisan

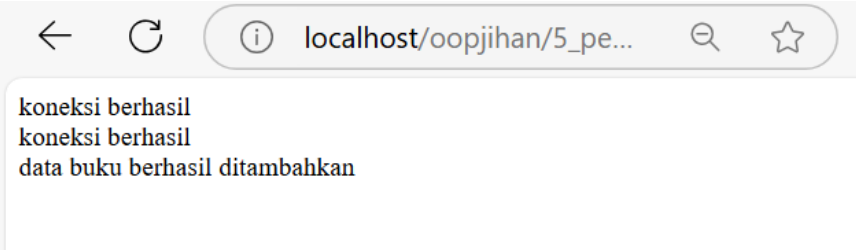


```php
<?php

include 'Koneksi.php';

$objek=new Koneksi('localhost','root','','buku');
$objek->cekKoneksi();


$objek1=new Buku(title: 'localhost',author: 'root',publisher: '','buku');
$objek1->cekKoneksi();
$objek1->tambahBuku();
```



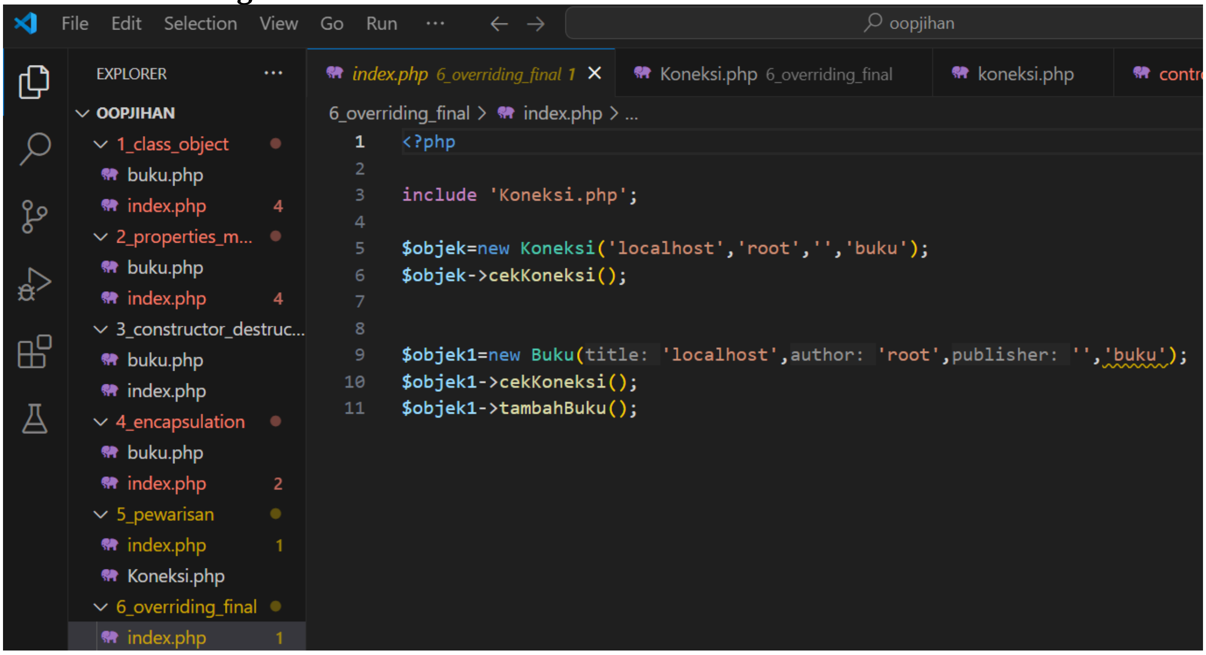
```php
<?php

5 references | 3 implementations
class Koneksi{
    3 references
    protected $server;
    2 references
    protected $username;
    2 references
    protected $password;
    2 references
    protected $databases;

    2 references | 1 override
    public function __construct($server,$username,$password,$databases)
    {
        $this->server=$server;
        $this->username=$username;
        $this->password=$password;
        $this->databases=$databases;
    }

    4 references | 1 override
    public function cekKoneksi(): void{
        echo "koneksi berhasil<br>";
    }
}

12 references | 0 implementations
class Buku extends Koneksi{

    2 references | 0 overrides
    public function tambahBuku(): void{
        echo "data buku berhasil ditambahkan";
    }
```

Output:

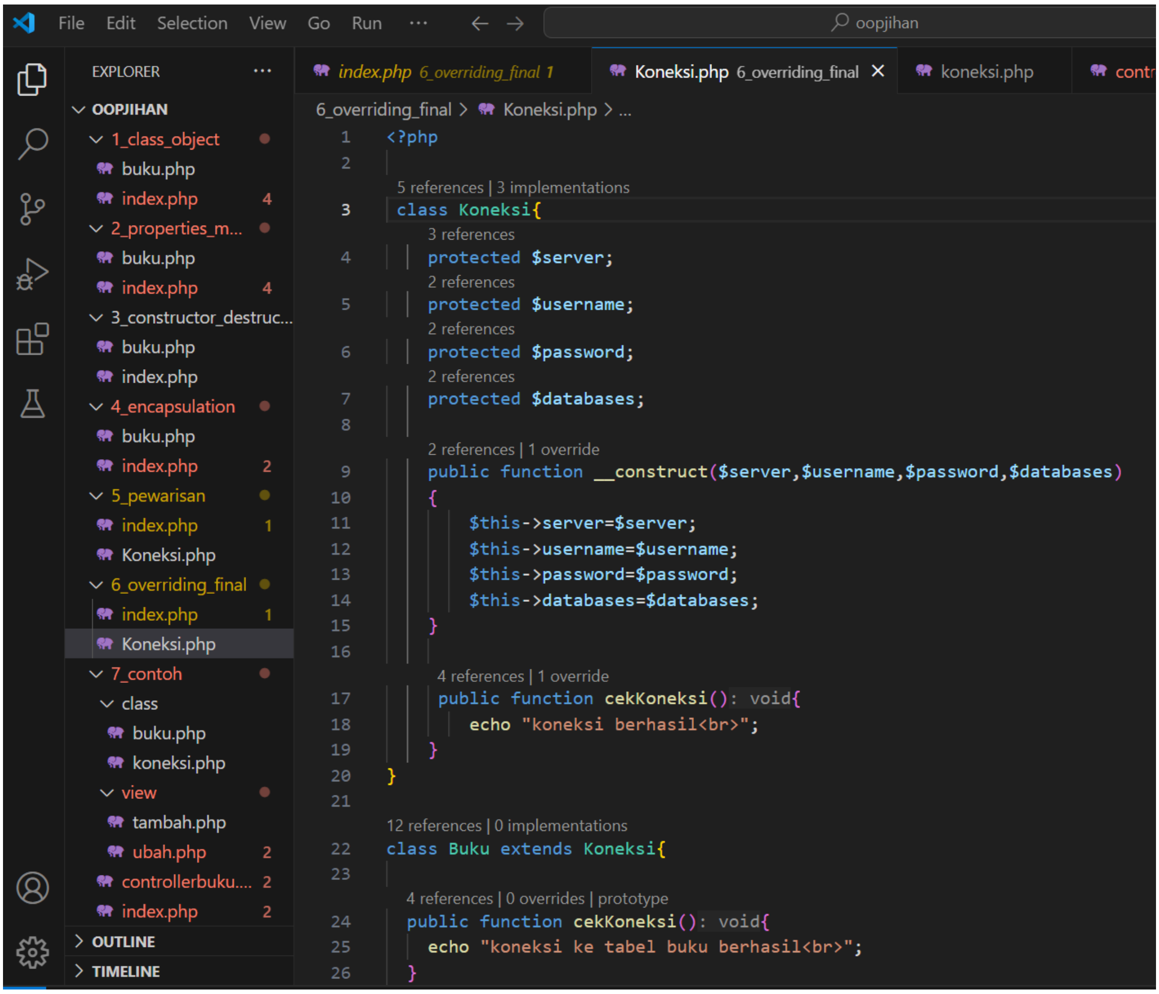


```
koneksi berhasil
koneksi berhasil
data buku berhasil ditambahkan
```

6. Overriding final

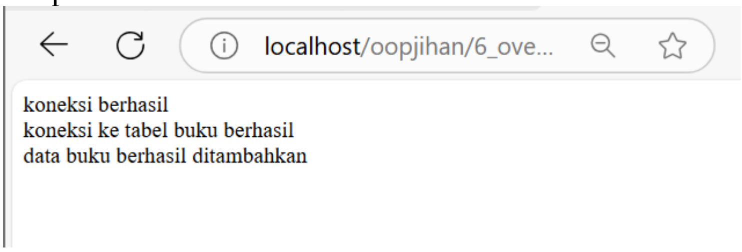


```php
<?php

include 'Koneksi.php';

$objek=new Koneksi('localhost','root','','buku');
$objek->cekKoneksi();


$objek1=new Buku(title: 'localhost',author: 'root',publisher: '','buku');
$objek1->cekKoneksi();
$objek1->tambahBuku();
```



```php
<?php

5 references | 3 implementations
class Koneksi{
    3 references
    protected $server;
    2 references
    protected $username;
    2 references
    protected $password;
    2 references
    protected $databases;

    2 references | 1 override
    public function __construct($server,$username,$password,$databases)
    {
        $this->server=$server;
        $this->username=$username;
        $this->password=$password;
        $this->databases=$databases;
    }

     4 references | 1 override
     public function cekKoneksi(): void{
        echo "koneksi berhasil<br>";
    }
}

12 references | 0 implementations
class Buku extends Koneksi{

  4 references | 0 overrides | prototype
  public function cekKoneksi(): void{
    echo "koneksi ke tabel buku berhasil<br>";
  }
```

Output:

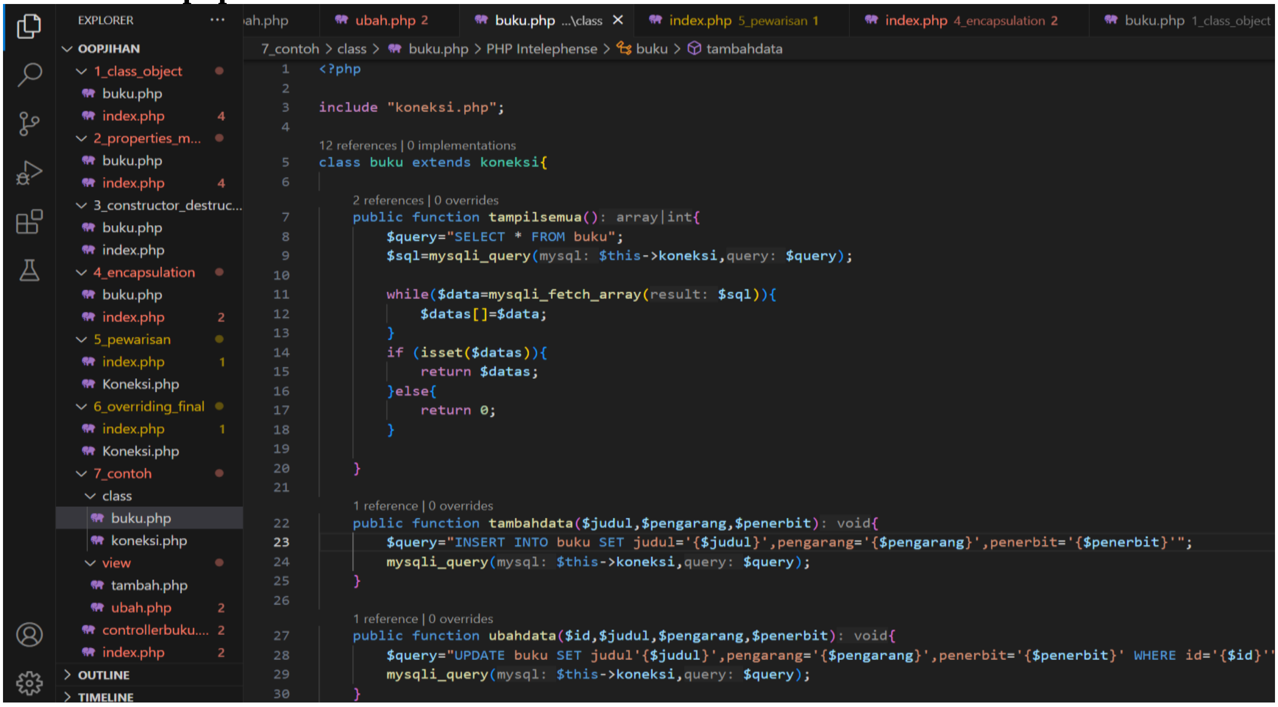


```
koneksi berhasil
koneksi ke tabel buku berhasil
data buku berhasil ditambahkan
```

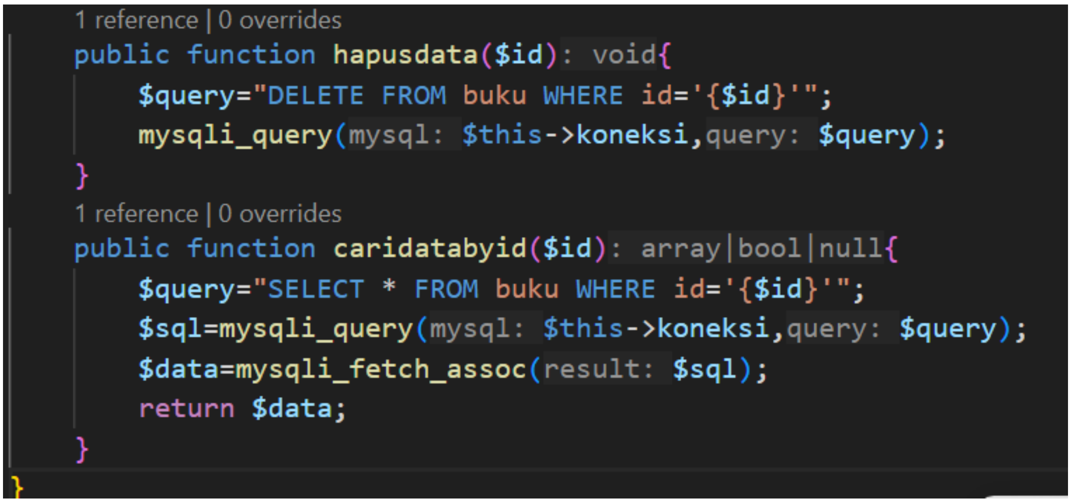
7. Implementasi/contoh

Folder class:

- **Buku.php**

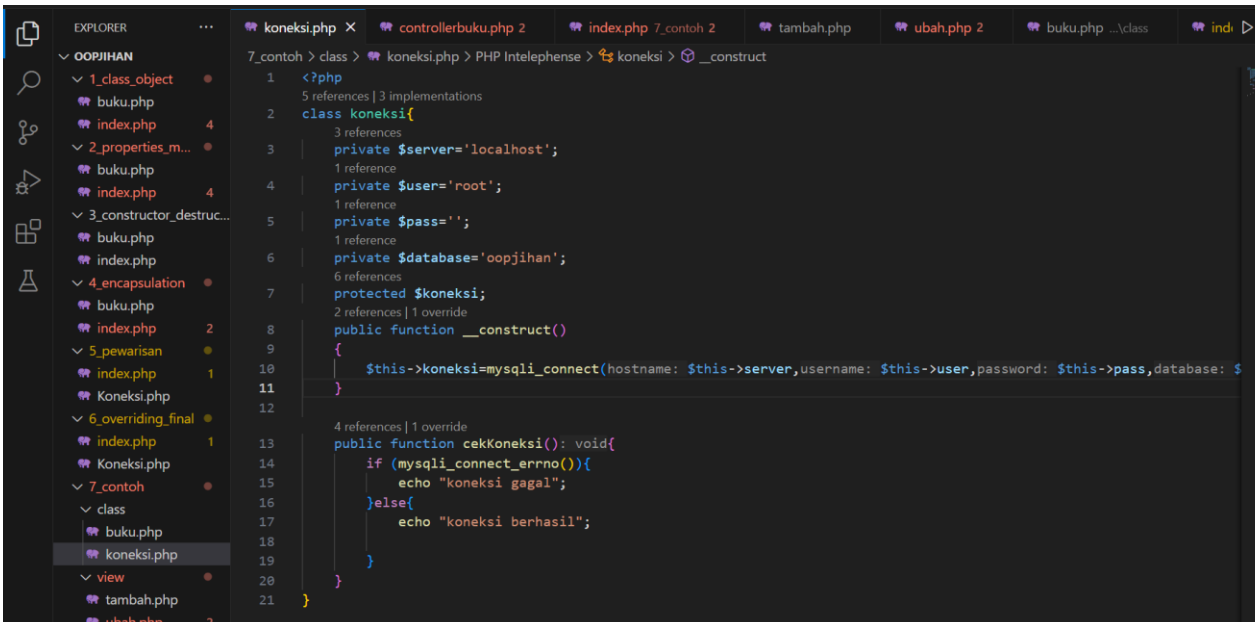


```php
<?php

include "koneksi.php";

12 references | 0 implementations
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);
    }
```



```php
    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;
    }
}
```

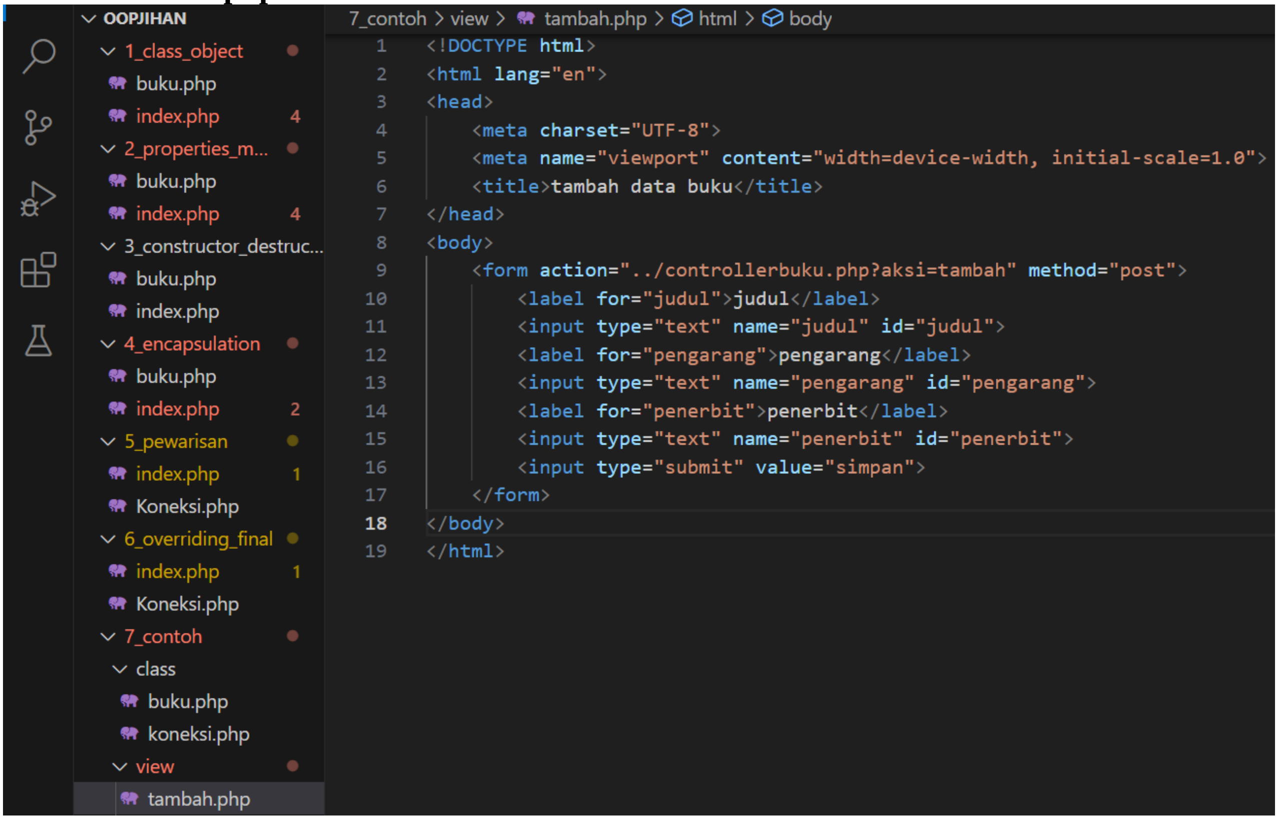
- **Koneksi.php**



```php
<?php
class koneksi{
    private $server='localhost';
    private $user='root';
    private $pass='';
    private $database='oopjihan';
    protected $koneksi;
    public function __construct()
    {
        $this->koneksi=mysqli_connect(hostname: $this->server,username: $this->user,password: $this->pass,database: $
    }

    public function cekKoneksi(): void{
        if (mysqli_connect_errno()){
            echo "koneksi gagal";
        }else{
            echo "koneksi berhasil";

        }
    }
}
```

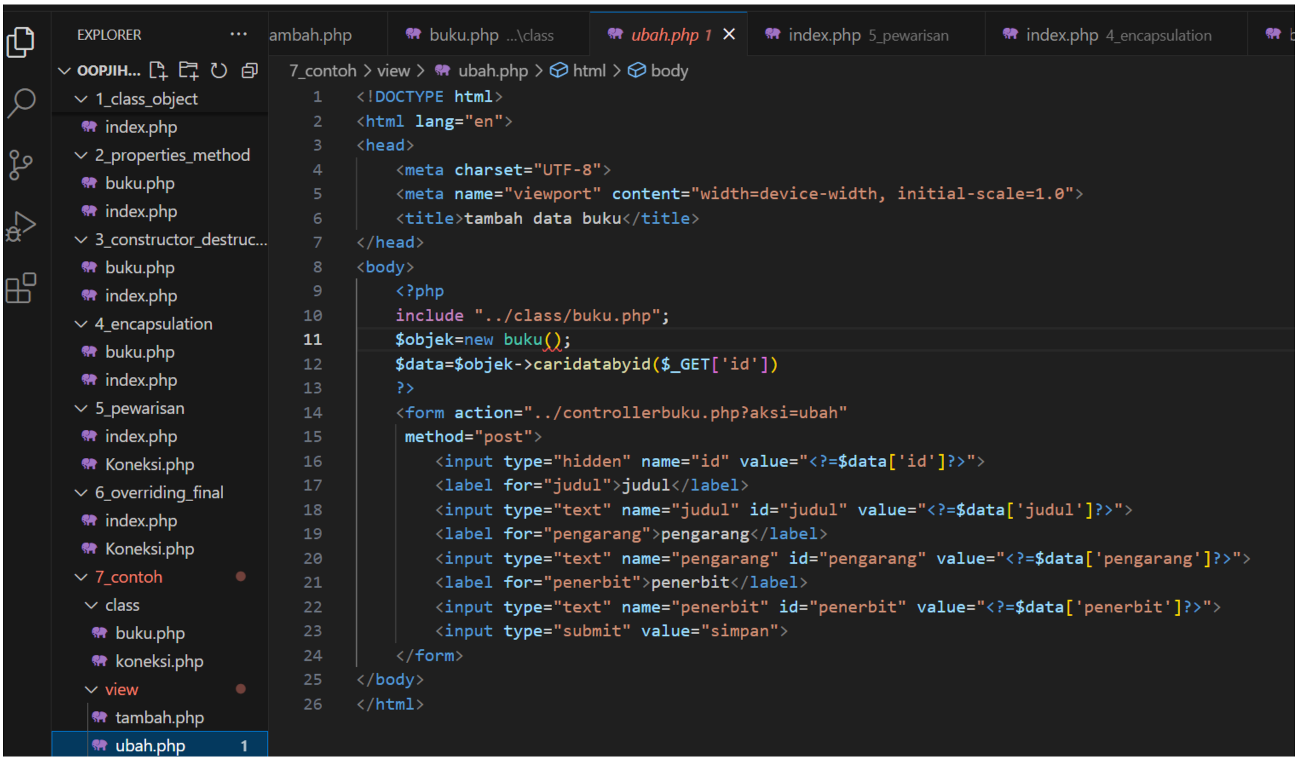
Folder view:

- **Tambah.php**



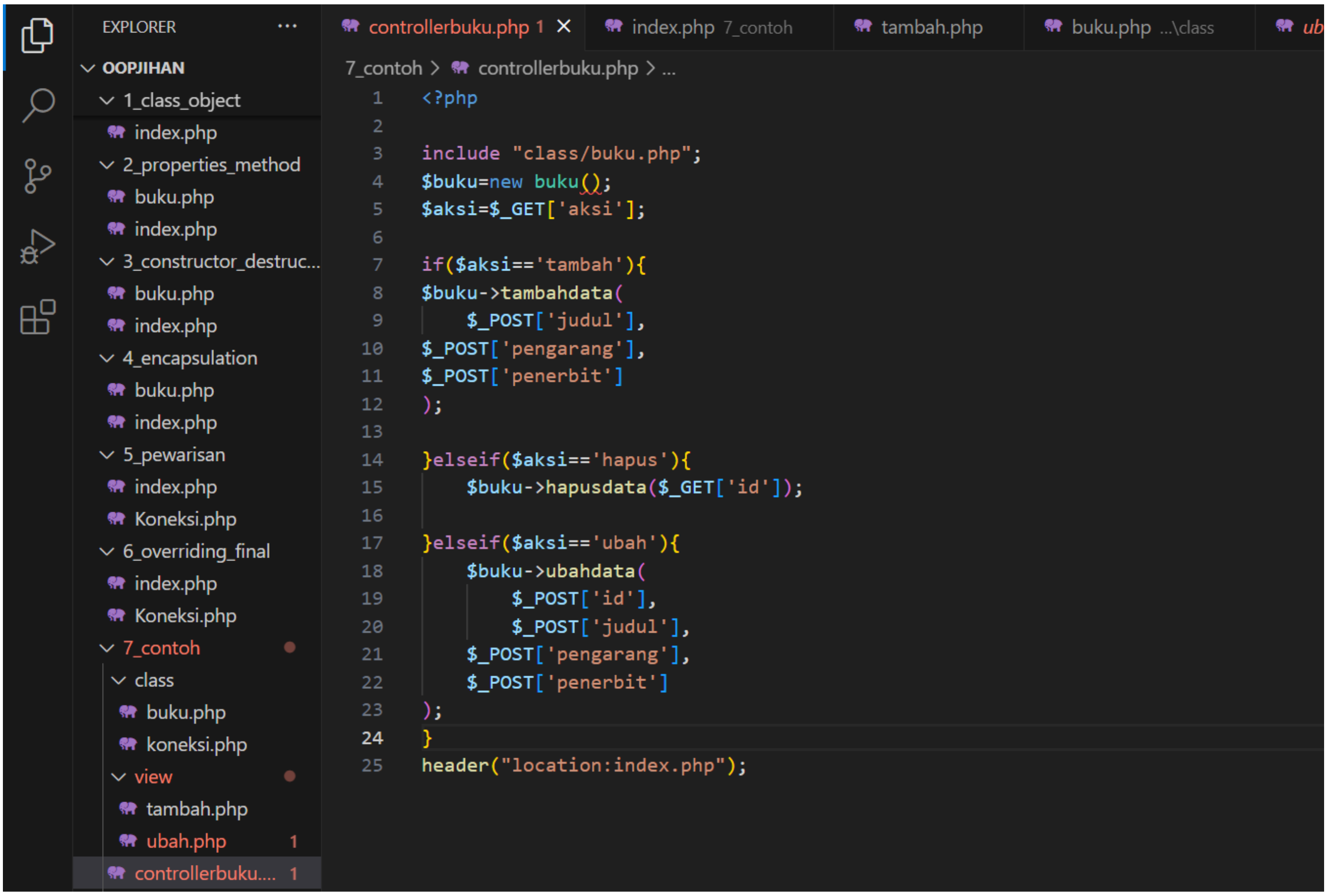
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>tambah data buku</title>
</head>
<body>
    <form action="../controllerbuku.php?aksi=tambah" method="post">
        <label for="judul">judul</label>
        <input type="text" name="judul" id="judul">
        <label for="pengarang">pengarang</label>
        <input type="text" name="pengarang" id="pengarang">
        <label for="penerbit">penerbit</label>
        <input type="text" name="penerbit" id="penerbit">
        <input type="submit" value="simpan">
    </form>
</body>
</html>
```

- **Ubah.php**



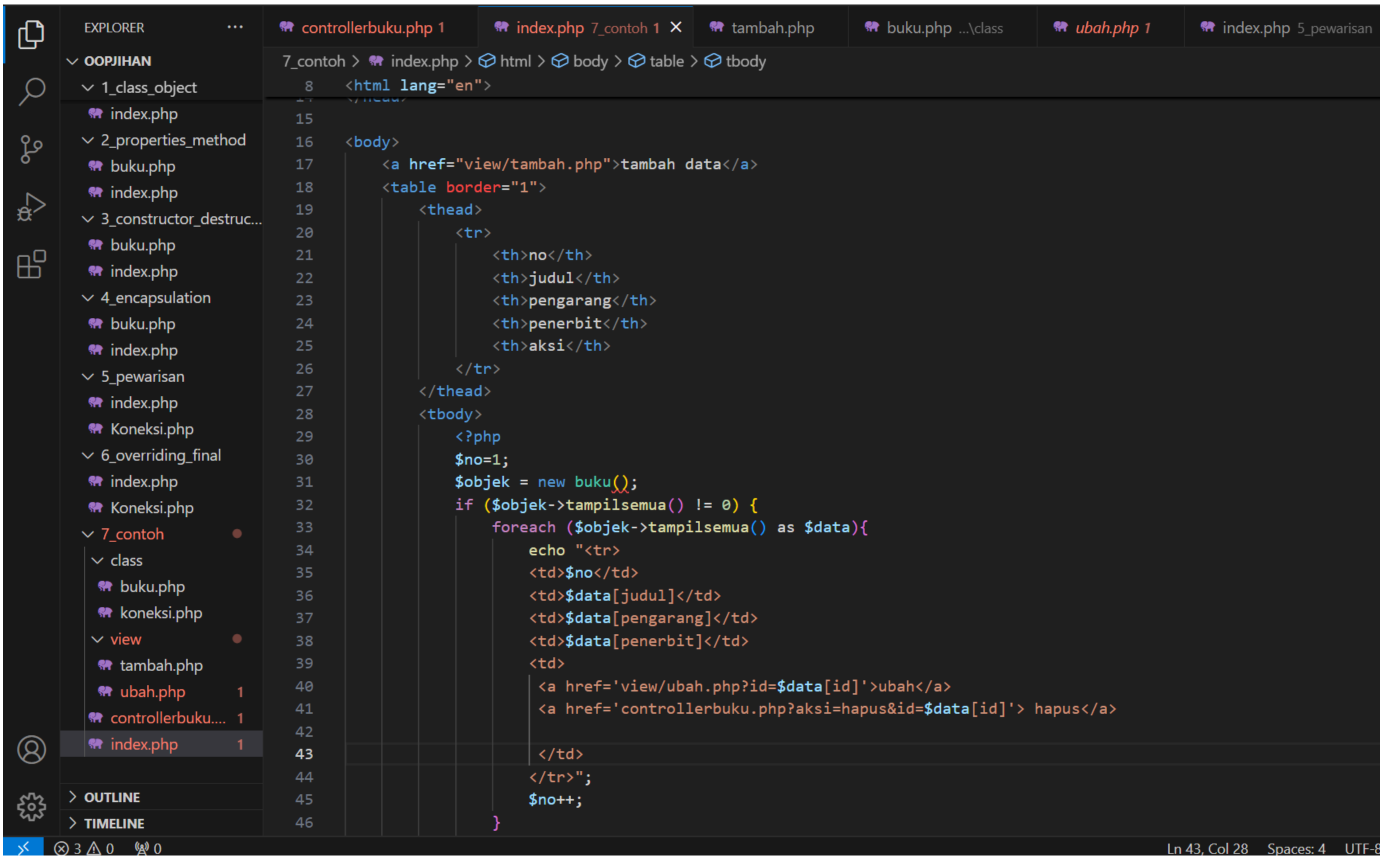
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>tambah data buku</title>
</head>
<body>
    <?php
    include "../class/buku.php";
    $objek=new buku();
    $data=$objek->caridatabyid($_GET['id'])
    ?>
    <form action="../controllerbuku.php?aksi=ubah"
     method="post">
        <input type="hidden" name="id" value="<?=$data['id']?>">
        <label for="judul">judul</label>
        <input type="text" name="judul" id="judul" value="<?=$data['judul']?>">
        <label for="pengarang">pengarang</label>
        <input type="text" name="pengarang" id="pengarang" value="<?=$data['pengarang']?>">
        <label for="penerbit">penerbit</label>
        <input type="text" name="penerbit" id="penerbit" value="<?=$data['penerbit']?>">
        <input type="submit" value="simpan">
    </form>
</body>
</html>
```

Folder controllerbuku.php



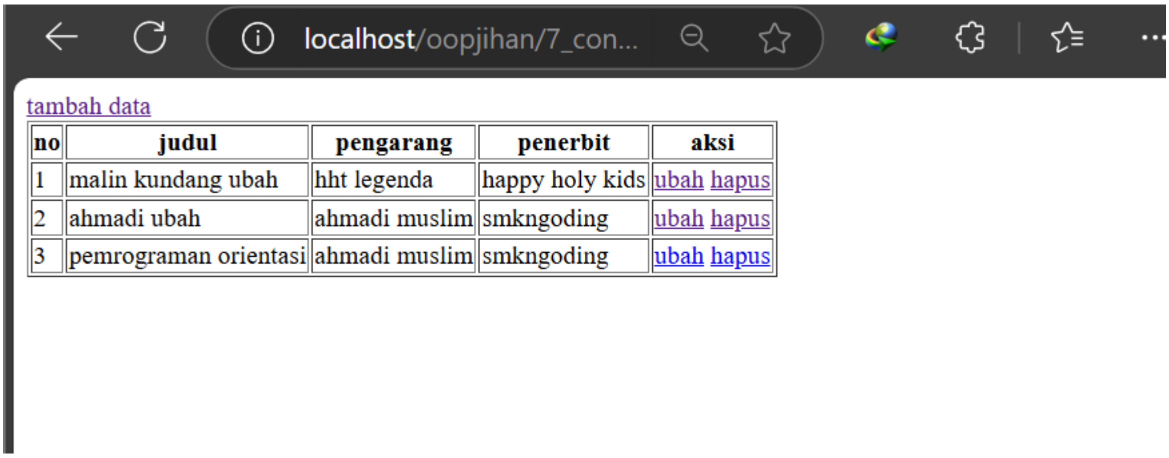
```php
<?php

include "class/buku.php";
$buku=new buku();
$aksi=$_GET['aksi'];

if($aksi=='tambah'){
$buku->tambahdata(
    $_POST['judul'],
$_POST['pengarang'],
$_POST['penerbit']
);

}elseif($aksi=='hapus'){
    $buku->hapusdata($_GET['id']);

}elseif($aksi=='ubah'){
    $buku->ubahdata(
        $_POST['id'],
        $_POST['judul'],
    $_POST['pengarang'],
    $_POST['penerbit']
);
}
header("location:index.php");
```

Folder index.php



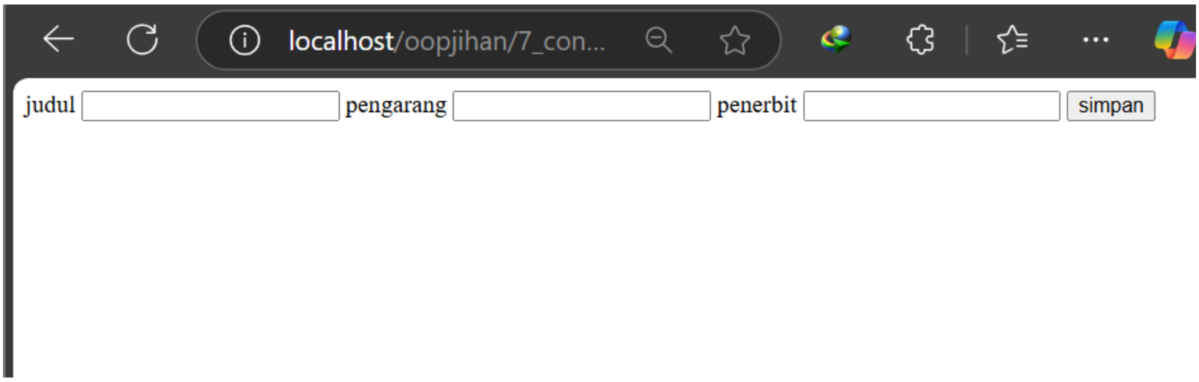
```
<html lang="en">

<body>
    <a href="view/tambah.php">tambah data</a>
    <table border="1">
        <thead>
            <tr>
                <th>no</th>
                <th>judul</th>
                <th>pengarang</th>
                <th>penerbit</th>
                <th>aksi</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $no=1;
            $objek = new buku();
            if ($objek->tampilsemua() != 0) {
                foreach ($objek->tampilsemua() as $data){
                    echo "<tr>
                    <td>$no</td>
                    <td>$data[judul]</td>
                    <td>$data[pengarang]</td>
                    <td>$data[penerbit]</td>
                    <td>
                     <a href='view/ubah.php?id=$data[id]'>ubah</a>
                     <a href='controllerbuku.php?aksi=hapus&id=$data[id]'> hapus</a>

                     </td>
                    </tr>";
                    $no++;
                }
```

Output:



tambah data

no	judul	pengarang	penerbit	aksi
1	malin kundang ubah	hht legenda	happy holy kids	ubah hapus
2	ahmadi ubah	ahmadi muslim	smkngoding	ubah hapus
3	pemrograman orientasi	ahmadi muslim	smkngoding	ubah hapus



judul ______ pengarang ______ penerbit ______ simpan