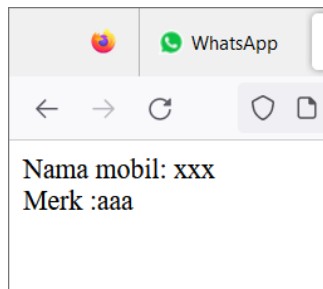


NAMA : DEDI PRASETYA

NIM : G.231.21.0215

LATIHAN 1. Class dan Object + Modifier

- a. Tampilan script



- b. Ditambahkan variable \$a

Script :

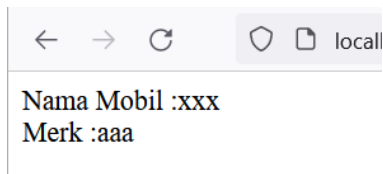
```
<?php
//class mobil
Class Mobil {
    public $nama;
    public $merk;

    function getInfoNama(){
        return $this->nama;
    }
    function getInfoMerk(){
        return $this->merk;
    }
    function setInfoNama($a){
        $this->nama=$a;
    }
    function setInfoMerk($a){
        $this->merk=$a;
    }
}

//main
$ferrari=new Mobil();
$ferrari->setInfoNama("xxx");
$ferrari->setInfoMerk("aaa");
echo "Nama Mobil :".$ferrari->getInfoNama()."<br>";
echo "Merk :".$ferrari->getInfoMerk();

?>
```

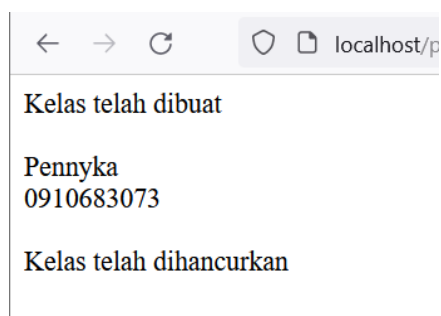
Screenshoot :



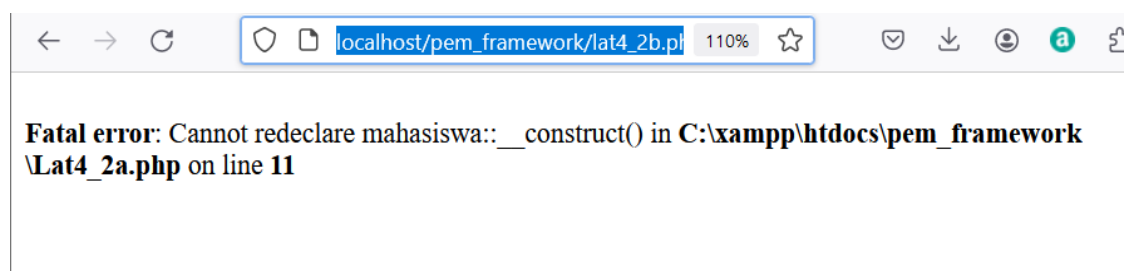
- c. Parameter yang sudah ditentukan diawal, masih bisa dirubah parameternya pada method yang kita jalankan.

LATIHAN 2

a. Tampilan sebelum modifikasi



b. Tampilan Setelah dimodifikasi

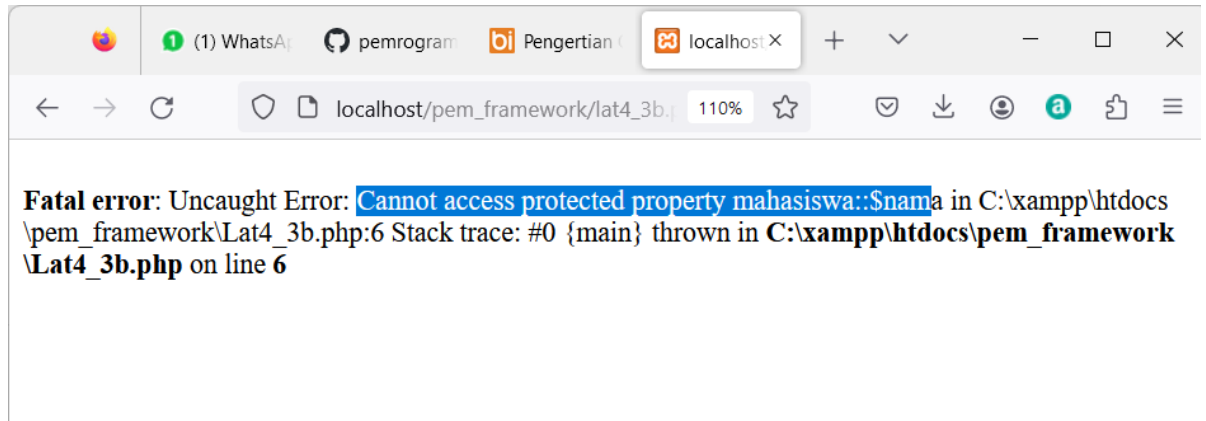


d. Kesimpulan

Konstruktor hanya dapat dideklarasikan atau dituliskan 1 kali pada suatu class.

LATIHAN 3

- Terjadi eror , Eror disebabkan property NIM dan Nama di dibuat Private, sehingga tidak bisa diakses oleh sub class nya.
- Perubahan yang terjadi

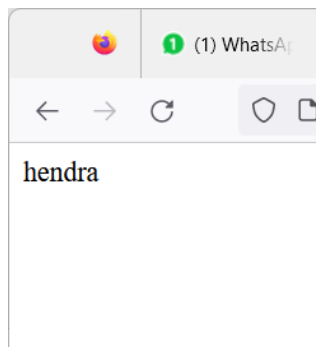


- Modifikasi Lat4_3b.php

```
<?php
require_once("lat4_3a.php");

$mhs1=new mahasiswa();
$mhs1->setNama("hendra");
echo $mhs1->getNama(); ===> nama dirubah dengan method getName().
?>
```

Hasilnya :



- Kesimpulan

Penentuan Visibility mengacu pada data yang di akses, ketika data tersebut sifatnya dapat di akses dari manapun maka dapat menggunakan visibility public, sedangkan data yang sifatnya rahasia pada kebutuhan menggunakan visibility private. Selain itu dapat menggunakan visibility protected ketika ada komunikasi antara class child dan parent.

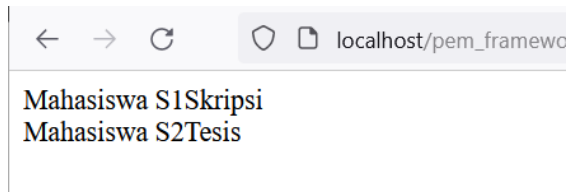
LATIHAN 4

Kesimpulan :

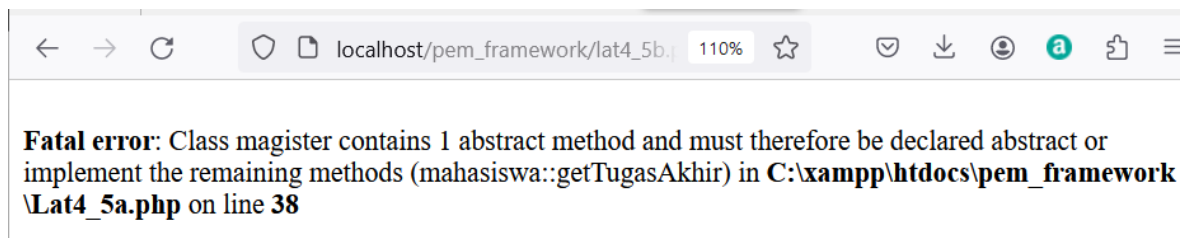
Sub class bisa ditulis pada file lain sepanjang file tersebut terhubung dengan file yang berisi class utama, untuk menghubungkan antar file menggunakan perintah **require_once()**

LATIHAN 5.

a. Hasil Tampilan



b. Tampilan Setelah dirubah



Hal ini terjadi karena setelah dihapus, sub class Magister hanya mempunyai 1 method , sedangkan parent classnya (mahasiswa) mempunyai 2 method.padahal di dalam aturan abstract class, jumlah method parent class dengan subclass harus sama.

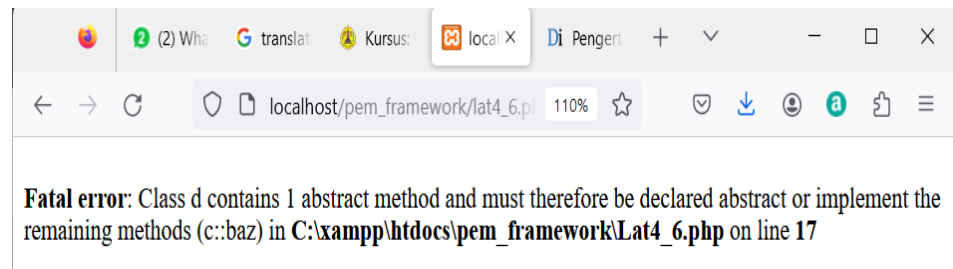
c. Kesimpulan

- Metode kelas anak harus didefinisikan dengan nama yang sama dan mendeklarasikan ulang metode abstrak induk
- Metode kelas anak harus ditentukan dengan pengubah akses yang sama atau kurang dibatasi
- Jumlah argumen yang diperlukan harus sama. Namun, kelas anak mungkin memiliki argumen opsional sebagai tambahan

LATIHAN 6

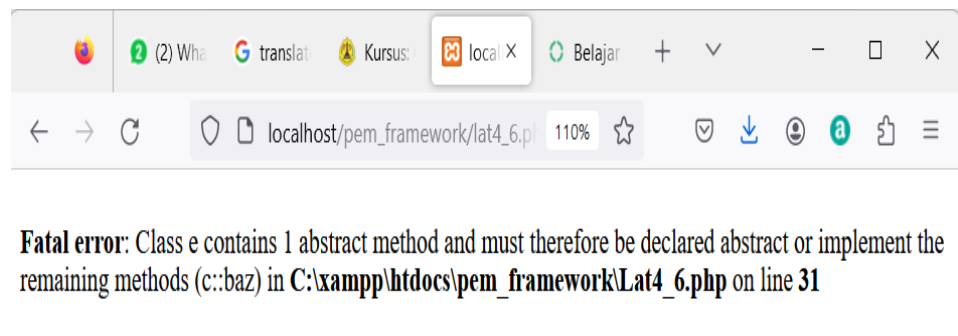
- a. Pembuatan Objek interface yang terdiri dari 2 Interface (A dan B), 1 sub interface C yang menjadi subinterface A maupun B. juga terdapat class d yang merupakan sub dari interface C . class d ini otomatis harus menampung semua method dari Interface A, B dan C.

- b. Tampilan setelah diubah



Hal ini terjadi karena ada method yang belum terbuat pada class d, yaitu method baz().

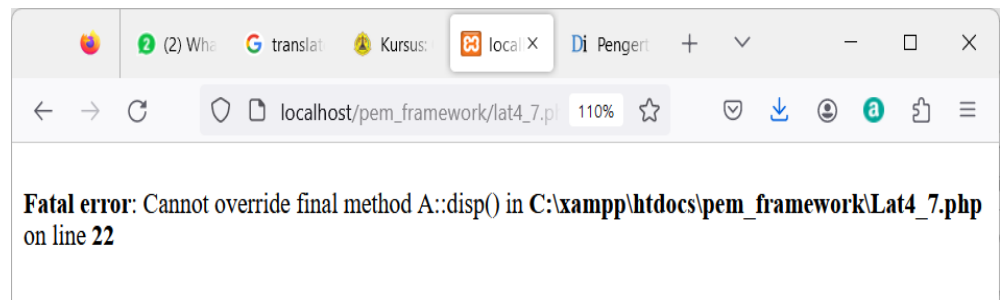
- c. Menambahkan class e dengan method **foo** dan **bar**
Tampilan



- d. Kesimpulan, Interface secara konsep hampir saja dengan abstract, hanya saja interface digunakan untuk aturan terhadap implemetansi class yang mewarisinya, artinya class yang menggunakan interface harus menuliskan ulang method yang yang sama. Bisa di katakan menuliskan method yang berasal dari signature / method interface.

LATIHAN 7.

a. Tampilan Script



Hal ini terjadi karena Dengan menambahkan **keyword final** kepada sebuah *method*, maka *method* tersebut tidak dapat didefinisikan ulang di dalam *child class*. Dan jika sebuah **class** ditambahkan keyword *final*, maka *class* tersebut tidak bisa diturunkan sama sekali.

b. Merubah script kata final baris ke 5 dan menambahkan kata final pada baris ke 2

Script :

```
<?php
final class A

{

    public function disp(){

        echo "Inside the final function";

    }

}

class B extends A {

    function disp(){

        echo "Inside the final function";

    }

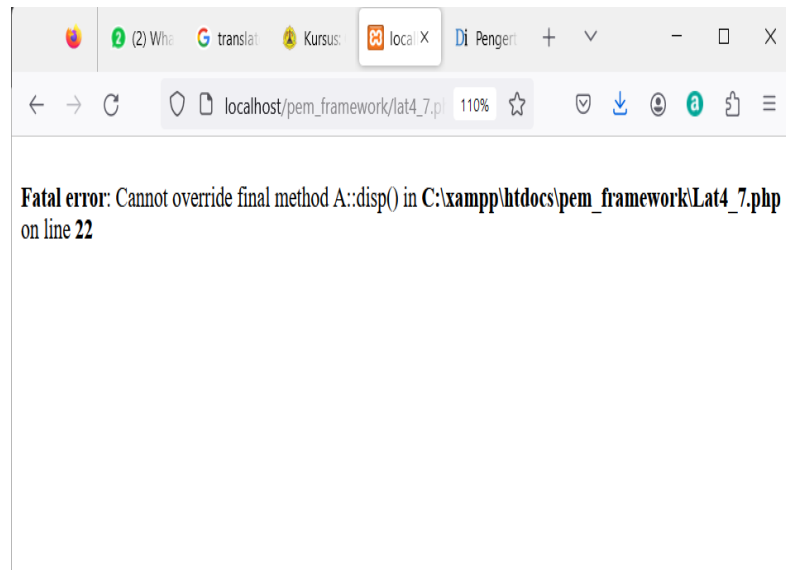
}

$obj=new B();

$obj->disp();

?>
```

Screenshot :



Hal ini terjadi karena Final class adalah class yang tidak bisa diwarisi ([inheritance](#)) oleh class lain. Dengan kata lain class ini merupakan end-class yang tidak bisa diturunkan kepada class lain. Dan jika kita mencoba membuat class dengan inherit kepada final class akan muncul error.

Kesimpulan :

Baik Final Class maupun Final method sama-sama tidak bisa diwarisi oleh class lain.

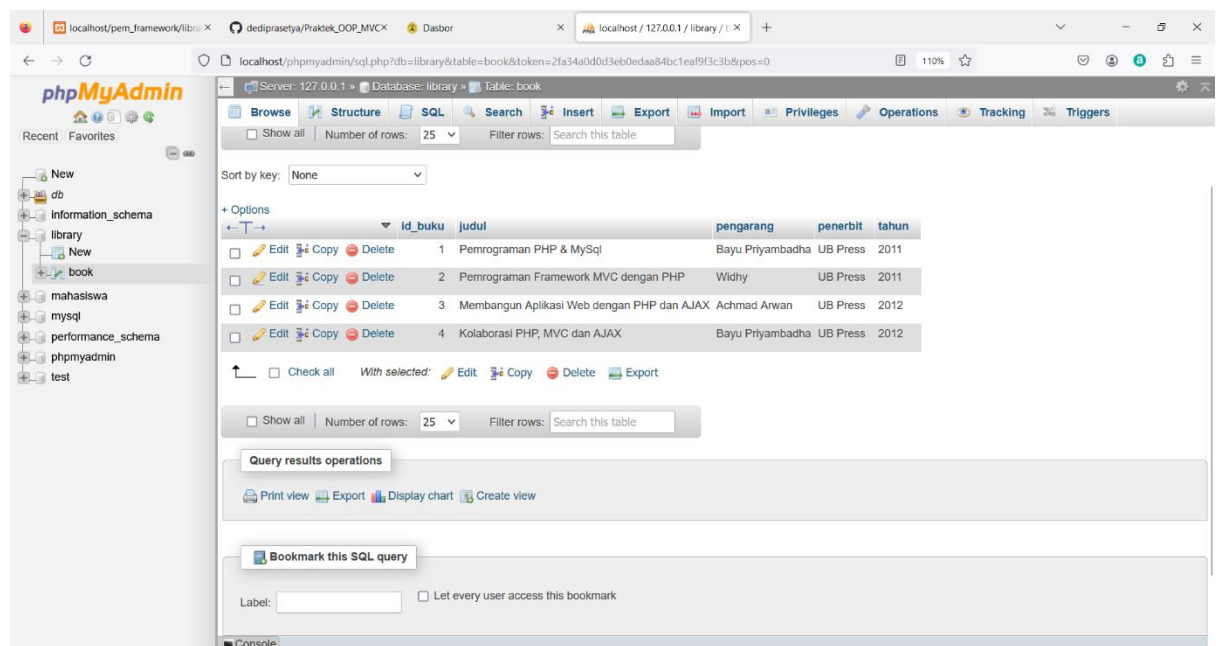
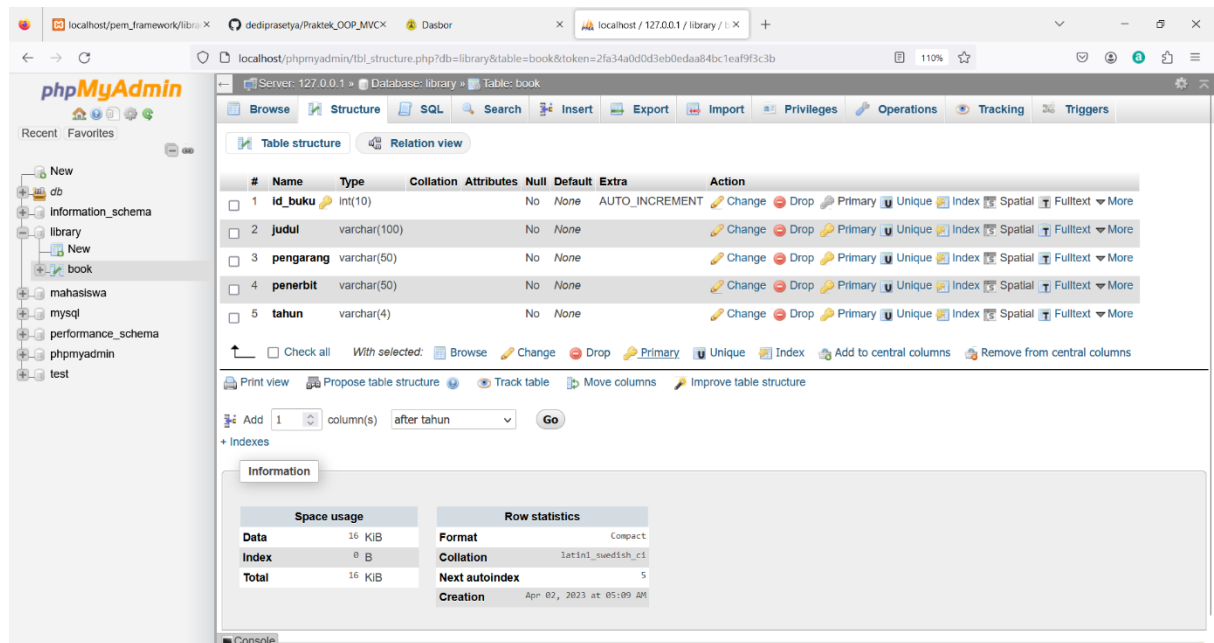
LATIHAN 8

a. Kesimpulan

Keyword Static **Method** dan **Property** cocok digunakan ketika data komunikasi dengan class terhadap **method** dan **property** nya. Keyword static juga membuat seolah-olah kode kita menjadi procedural namun memangkas penulisan kode panjang seperti membuat object baru.

LATIHAN 9.

a. Screenshoot database Library yang berisi table book



b. Script Database.php

```
<?php
```

```
class database{
```

```

var $host = "localhost";
var $username = "root";
var $password = "";
var $database = "library";
var $koneksi = "";
function __construct(){
    $this->koneksi = mysqli_connect($this->host, $this->username, $this-
>password,$this->database);
    if (mysqli_connect_errno()){
        echo "Koneksi database gagal : " . mysqli_connect_error();
    }
}

function tampil_data()
{
    $data = mysqli_query($this->koneksi,"select * from book");
    while($row = mysqli_fetch_array($data)){
        $hasil[] = $row;
    }
    return $hasil;
}
}

?>

```

c. Script index.php

```

<?php
include 'database.php';
$db = new database();
$data_buku = $db->tampil_data();
?>
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
<table border="1">
    <tr>
        <th>No</th>
        <th>Judul</th>
        <th>Pengarang</th>
        <th>Penerbit</th>
        <th>Tahun</th>
        <th>Action</th>
    </tr>
    <?php
    $no = 1;
    foreach($data_buku as $row){

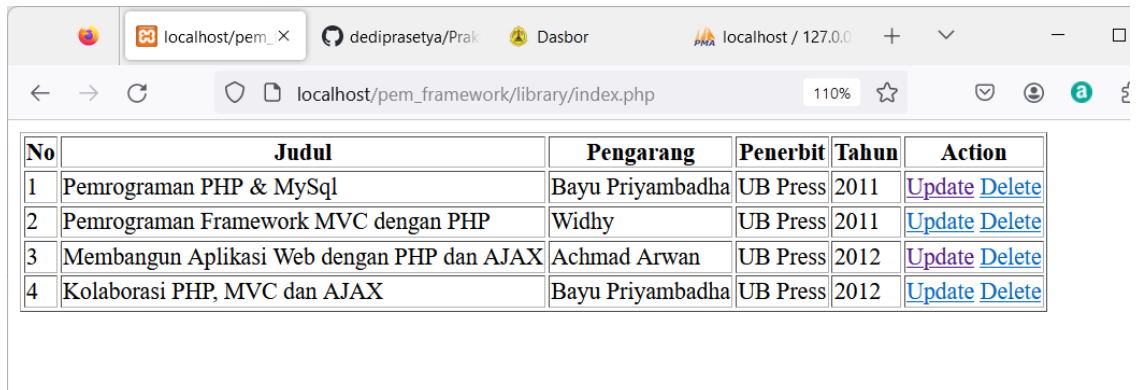
```

```

?>
<tr>
    <td><?php echo $no++; ?></td>
    <td><?php echo $row['judul']; ?></td>
    <td><?php echo $row['pengarang']; ?></td>
    <td><?php echo $row['penerbit']; ?></td>
    <td><?php echo $row['tahun']; ?></td>
    <td>
        <a href="edit.php?id=<?php echo $row['id_buku'];
?>">Update</a>
        <a href="hapus.php?id=<?php echo $row['id_buku'];
?>">Delete</a>
    </td>
</tr>
<?php
}
?>
</table>
</body>
</html>

```

d. Screenshot simulasi di browser



No	Judul	Pengarang	Penerbit	Tahun	Action
1	Pemrograman PHP & MySql	Bayu Priyambadha	UB Press	2011	Update Delete
2	Pemrograman Framework MVC dengan PHP	Widhy	UB Press	2011	Update Delete
3	Membangun Aplikasi Web dengan PHP dan AJAX	Achmad Arwan	UB Press	2012	Update Delete
4	Kolaborasi PHP, MVC dan AJAX	Bayu Priyambadha	UB Press	2012	Update Delete