

TUGAS PEMOGRAMAN FRAMEWORK

Pertemuan 3

NAMA : TAUFIK ALIF SHALAHUDDIN
NIM : G.211.21.0064

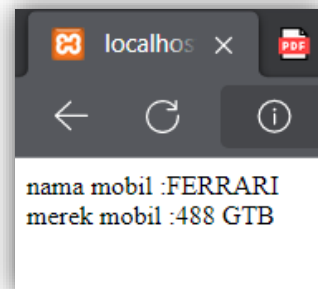
Latihan 1 : Class dan Object + modifier

a. Bagaimana tampilan diatas

```
<?php
// class mobil
class mobil{
    public $nama;
    public $merek;

    function getinfo(){
        echo "nama mobil :".$this->nama."<br/>";
        echo "merek mobil :".$this->merek."<br/>";
    }
}

// bagian main
$ferrari=new mobil();
$ferrari->nama="FERRARI";
$ferrari->merek="488 GTB";
$ferrari->getinfo($ferrari);
?>
```

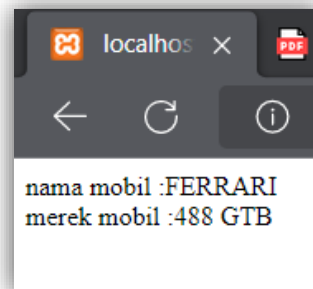


b. Buat method overload getinfo dengan \$a

```
<?php
// class mobil
class mobil{
    public $nama;
    public $merek;

    function getinfo($a){
        echo "nama mobil :".$a->nama."<br/>";
        echo "merek mobil :".$a->merek."<br/>";
    }
}

// bagian main
$ferrari=new mobil();
$ferrari->nama="FERRARI";
$ferrari->merek="488 GTB";
$ferrari->getinfo($ferrari);
?>
```



c. Kesimpulan pada latihan 1 adalah jika method tidak memiliki isi atau kosong () maka pemanggilan menggunakan \$this, dan jika diisi seperti (\$a) maka pemanggilan juga harus menggunakan (\$a) juga.

Latihan 2

- a. Terdapat eror karena dalam satu class hanya boleh ada 1 kontruksi saja

```
<?php
class mahasiswa {
    public $nama;
    public $nim;

    function __construct($a,$b){
        $this->nama=$a;
        $this->nim=$b;
        echo "Kelas telah dibuat <br/><br/>";
    }
    function __construct($c,$d){
        $this->nama=$c;
        $this->nim=$d;
        echo "Kelas telah dibuat <br/><br/>";
    }
    function cetak(){
        echo $this->nama."<br/>".$this->nim."<br/><br/>";
    }
    function __destruct(){
        echo "Kelas telah dihancurkan";
    }
}
?>

<!-- jika satu class terdapat 2 construct tidak akan bisa
berjalan -->
```

localhost/semester_3/fram... Fatal error: Cannot redeclare mahasiswa::__construct() in C:\xampp\htdocs\semester_3\framework\TUGAS_2\materi_praktikum\lat4_2a.php on line 11

Latihan 3

- a. Terdapat eror karena private hanya bisa digunakan pada class tertentu saja.

```
<?php
class mahasiswa{
    private $nama;
    private $nim;

    function __construct(){

    }

    function setnama ($a){
        $this->nama=$a;
    }
    function setnim ($b){
        $this->nim=$b;
    }

    function getnama (){
        return $this->nama;
    }
    function getnim (){
        return $this->nim;
    }
}
?>
```

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

$mhs1=new mahasiswa ();
$mhs1->nama="udin";
echo $mhs1->nama;

?>
```

localhost/semester_3/fram... Fatal error: Uncaught Error: Cannot access private property mahasiswa::\$nama in C:\xampp\htdocs\semester_3\framework\TUGAS_2\materi_praktikum\lat4_3b.php:5 Stack trace: #0 {main} thrown in C:\xampp\htdocs\semester_3\framework\TUGAS_2\materi_praktikum\lat4_3b.php on line 5

- b. Merubah modifier dari variable nama dan nim menjadi protected dan public.
- Jika private diubah menjadi public maka akan tetap berjalan dengan lancar
 - Jika protected dijalankan akan terjadi error karena protected sama dengan private hanya dapat digunakan di dalam class tertentu saja.
- c. Modifikasi mencetak nama dan nim

```
<?php
class mahasiswa{
    private $nama;
    private $nim;

    function __construct(
        $a = "Taufik Alif Shalahuddin <br/>",
        $b = "G.211.21.0064"){
        $this->nama = $a;
        $this->nim = $b;
    }

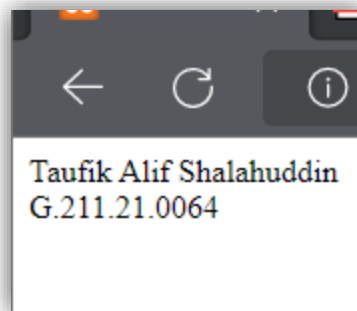
    function setnama ($a){
        $this->nama=$a;
    }
    function setnim ($b){
        $this->nim=$b;
    }

    function getnama (){
        return $this->nama;
    }
    function getnim (){
        return $this->nim;
    }
}
?>
```

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

$mhs1=new mahasiswa ();
echo $mhs1->getnama();
echo $mhs1->getnim();

?>
```



- d. Kesimpulannya adalah private dan protected dapat dipanggil bagian class nya saja yaitu method nya yang ada didalam class tersebut

Latihan 4

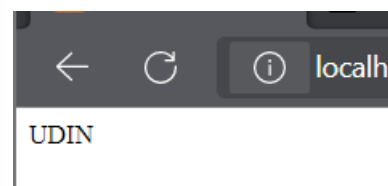
- Tidak terjadi error

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

class asisten extends mahasiswa
{
    function __construct(){}
```

```
<?php
require_once("lat4_4a.php");

$as = new asisten ();
$as->setnama ("UDIN");
echo $as->getnama();
```



Latihan 5

a. Bagaimana hasil tampilan dari kode diatas

```
<?php
abstract class Mahasiswawa
{
    abstract protected function getTugasAkhir();
    abstract protected function getProgram($postfix);

    public function tugasAkhir()
    {
        print $this->getTugasAkhir() . "<br>";
    }
}

class sarjana extends Mahasiswawa
{
    protected function getTugasAkhir()
    {
        return " Skripsi";
    }

    public function getProgram($postfix)
    {
        print "{$postfix} S1";
    }
}

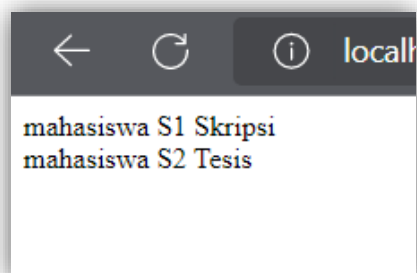
class magister extends Mahasiswawa
{
    public function getTugasAkhir()
    {
        return " Tesis";
    }

    public function getProgram($postfix)
    {
        print "{$postfix} S2";
    }
}
```

```
<?php
require_once("lat4_5a.php");

$s = new sarjana;
$s->getprogram('mahasiswa')."<br/>";
$s->tugasakhir();

$magister = new magister;
$magister->getprogram('mahasiswa')."<br/>";
$magister->tugasakhir();
```



b. Hapus pada baris 29 – 32

Terjadi error karena method `gettugasakhir()` tidak memiliki turunan sedangkan method abstract hanya dapat dipanggil apabila turunannya terdapat `gettugasakhir()`

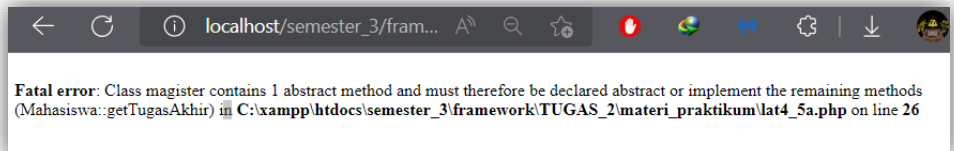
```
<?php
abstract class Mahasiswa
{
    abstract protected function getTugasAkhir();
    abstract protected function getProgram($postfix);

    public function tugasAkhir()
    {
        print $this->getTugasAkhir() . "<br>";
    }
}

class sarjana extends Mahasiswa
{
    protected function getTugasAkhir()
    {
        return " Skripsi";
    }

    public function getProgram($postfix)
    {
        print "{$postfix} S1";
    }
}

class magister extends Mahasiswa
{
    public function getProgram($postfix)
    {
        print "{$postfix} S2";
    }
}
```



c. Kesimpulan

Dapat disimpulkan bahwa class dan method dengan keyword abstract hanya bisa diakses melalui turunannya saja

Latihan 6

```
<?php
interface a
{
    public function foo ();
}
interface b
{
    public function bar();
}
interface c extends a, b
{
    public function baz();
}

class d implements c
{
    public function foo()
    {
    }
    public function bar()
    {
    }
    public function baz()
    {
    }
}
?>
```

- Maksud dari coding diatas adalah a, b dan c adalah class interface dan memiliki method foo(), bar(), dan baz(), dan method tersebut harus bentuk abstract, sedangkan class d adalah implementasi dari class c dimana class c memiliki turunan a dan b.
- Menghapus kode baris 27-29.

```
<?php
interface a
{
    public function foo ();
}
interface b
{
    public function bar();
}
interface c extends a, b
{
    public function baz();
}
class d implements c
{
    public function foo()
    {
    }
    public function bar()
    {
    }
}
```

localhost/semester_3/fram... A 🔍 🌐 🔄 ⬇️ 👤

Parse error: Unclosed '{' on line 15 in C:\xampp\htdocs\semester_3\framework\TUGAS_2\materi_praktikum\lat4_6.php on line 22

- c. Dari contoh kode diatas, buatlah class baru dengan nama “e” yang mempunyai method foo dan bar.

```
class e implements a
{
    public function foo()
    {
    }
}
```

- d. Kesimpulannya adalah interface dengan method abstract harus mempunyai class dan implements yang sama dengan parentnya.

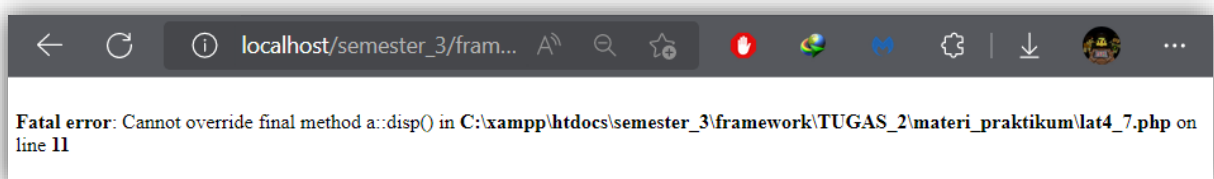
Latian 7

```
<?php
class a
{
    final public function disp(){
        echo "inside the final function";
    }
}

class b extends a{
    function disp(){
        echo "inside the final function";
    }
}

$obj=new b();
$obj->disp();
?>
```

- a. Tampilan kode diatas dan jelaskan
Error karena method pada class A tidak dapat override



- b. Modifikasi program diatas dengan menghapus kata final pada kode baris 5 dan menambah kata final pada baris 2

```
<?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();
?>
```

localhost/semester_3/fram... Fatal error: Class b cannot extend final class a in C:\xampp\htdocs\semester_3\framework\TUGAS_2\materi_praktikum\lat4_7.php on line 10

- c. Kesimpulannya adalah apabila memberikan status final pada deklarasi class maka class tersebut tidak dapat diturunkan
- d.

Latian 8

```
<?php
class one{
    private static $var=2;
    function __construct(){}
    static function disp(){
        print self::$var;
    }
    function __distruct(){}
}
one::disp();
?>
```

localhost/se
2

- a. Kesimpulannya adalah static digunakan untuk mencetak method tanpa melakukan instansiasi yaitu dengan memanggil class yang kemudian titik dua dua kali lalau method nya dipanggil.

Latian 9

Dari contoh kode MVC pada penjelasan diatas buatlah :

1. Model berhubungan langsung dengan database mysql dengan spesifikasi sebagai berikut:
 - a. Database : Library
 - b. Table : book

Username dan password menyesuaikan.

2. Buatlah view menjadi sebuah object, susun class view dimana fungsinya adalah mengambil file-file template yang sudah disediakan!
3. Tuliskan masing-masing kodenya, buatlah simulasinya