TUGAS PHP OOP DAN MVC FRAMEWORK

NAMA : M.ZIBRAN AKBAR

NIM : G.211.21.0084

PRODI : TEKNIK INFORMATIKA

III. MATERI PRAKTIKUM

Latihan 1: Class dan Object + Modifier

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

        function getInfo(){
                echo "Nama mobil: ".$this->nama."<br/>";
                echo "Merk: ".$this->merk."<br/>";
        }
}

//bagian main
$ferari=new Mobil();
$ferari->nama="ferrari portofino";
$ferari->merk="ferrari";
$ferari->getInfo();
?>
```

A. Bagaimana hasil tampilan di atas?
B. Buatlah sebuah method overload getInfodengan parameter $a. Lalujalankan dan amati perubahan yang terjadi.
C. Lalu simpulkan apa yang Anda peroleh dari Latihan 1!

JAWAB :

A.

B. Error, karena di php tidak bisa mengoverload method
C. Kesimpulan:
    1. Cara membuat class pada php

<?php
Class nama_class{
}
?>

2. Cara penulisan property
   Modifier $nama_properti;
3. Penulisan method
   Modifier function nama_method(){
   Isi_method;
   }
4. Cara inisiasi object
   $nama_object = new nama_class();
5. Cara mengisi property atau mendefinisikan property
   $nama_object->properties=”aaa”;
6. Cara memanggil/menjalankan method pada suatu class
   $nama_object->nama_methode();

LATIHAN 2 :

CODE:

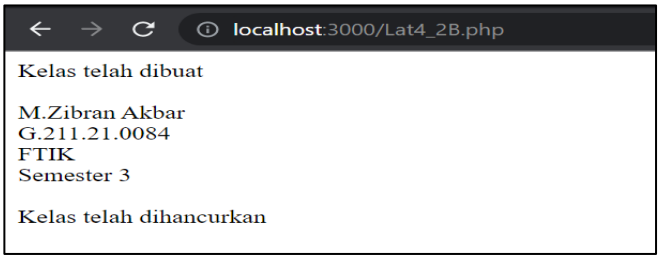
```php
<?php
class mahasiswa
{
    public $nama;
    public $nim;
    public $prodi;
    public $semester;
    function __construct($a, $b, $c, $d)
    {
        $this->nama = $a;
        $this->nim = $b;
        $this->prodi = $c;
        $this->semester = $d;

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

        echo "Kelas telah dibuat<br/><br/>";
    }
    function cetak()
    {
        echo $this->nama . "<br/>" . $this->nim . "<br/>";
        echo $this->prodi . "<br/>" . $this->semester . "<br/></br>";
    }
    function __destruct()
    {
        echo "Kelas telah dihancurkan";
    }
}
```

```php
<?php
require_once("lat4_2a.php");
$mhs2 = new mahasiswa("M.Zibran Akbar", "G.211.21.0084", "FTIK", "Semester 3");
$mhs2->cetak();
```

Modifikasilah latihan 4_2a dengan menambahkan 1 konstruktor lagi di lat 4_2a baris 11. Lalu jalankan. Bagaimana hasil tampilan di atas sebelum dan sesudah dimodifikasi? Lalu simpulkan apa yang Anda peroleh dari Latihan 2!

JAWAB :

- EROR



- Contructor dalam php oop tidak bisa di override**.**

LATIHAN 3 :

CODE :

```php
<?php
class mahasiswa
{
    public $nama;
    public $nim;
    function __construct()
    {
    }
    function setNama($a)
    {
        $this->nama = $a;
    }
    function setNim($b)
    {
        $this->nim = $b;
    }
    function getNama()
    {
        return $this->nama;
    }
    function getNim()
    {
        return $this->nim;
    }
    function destruct()
    {
    }
}
```

```php
<?php
require_once("lat4_3a.php");
$mhs1 = new mahasiswa();
$mhs1->nama = "M.Zibran Akbar </br></br>";
$mhs1->nim = "G.211.21.0084";
echo $mhs1->nama;
echo $mhs1->nim;
```

A. Apakah program error? Jika error mengapahalitudapatterjadi?
B. Rubahlah modifier dari variable namadannimmenjadi protected dan public.Laluamatiperubahan yang terjadi.
C. Modifikasilah Lat4_3b sehingga dapat member dan mencetak isi dari nim dan namadengan modifier private
D. Simpulkanapa yang andaperolehdarilatihan 3!

JAWAB :

A. Error, karena property yang bermodifier private hanya bisa digunakan pada class mahasiswasendiri.
B. Protected tetep error, public hasilnya m.zibran akbar
C. Kesimpulan :
    1. Pengunaan Modifier

Modifier	Keterangan
Public	Untuk mendefinisikan data atau metode yang akan terlihat dari luar oleh siapapun dan dimanapun.
Private	Untuk mendefinisikan data atau metode agar hanya terlihat pada class/object itu sendiri.
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 mengimplementasikansetter-getter.

LATIHAN 4 :

CODE :

```php
<?php
require_once("Lat4_3a.php");
class asisten extends mahasiswa
{
    public $nama;
    function __construct()
    {
    }
    function setNama($a)
    {
        $this->nama = $a;
    }
    function getNama()
    {
        return $this->nama;
    }
    function destruct()
    {
    }
}
```

```php
<?php
require_once("lat4_4a.php");
$as = new asisten();
$as->setNama("slebew");
echo $as->getNama();
```

Simpulkan apa yang Anda peroleh dari Latihan 4!

JAWAB :

Pada php oop, class asisten (child) bisa memanggil method dari mahasiswa (parent)

LATIHAN 5 :

CODE :

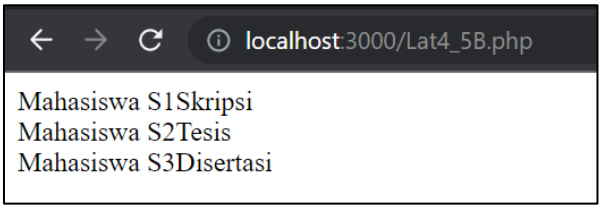
```php
<?php
abstract class mahasiswa
{
    abstract protected function getTugasAkhir();
    abstract protected function getProgram($postfix);
    public function tugasAkhir()
    {
        print $this->getTugasAkhir() . "<br>";
    }
}
class sarjana extends mahasiswa
{
    protected function getTugasAkhir()
    {
        return "Skripsi";
    }
    public function getProgram($postfix)
    {
        print "{$postfix} S1";
    }
}
class magister extends mahasiswa
{
    public function getTugasAkhir()
    {
        return "Tesis";
    }
    public function getProgram($postfix)
    {
        print "{$postfix} S2";
    }
}
class Doctor extends mahasiswa
{
    public function getTugasAkhir()
    {
        return "Disertasi";
    }
    public function getProgram($postfix)
    {
        print "{$postfix} S3";
    }
}
```

```php
<?php
require_once("lat4_5a.php");
$s = new sarjana;
$s->getProgram('Mahasiswa') . "<br>";
$s->tugasAkhir();
$m = new magister;
$m->getProgram('Mahasiswa') . "<br>";
$m->tugasAkhir();
$d = new Doctor;
$d->getProgram('Mahasiswa') . "<br>";
$d->tugasAkhir();
```

A. Bagaimana hasil tampilan dari program di atas?
B. Hapuslah kode baris 29 – 32 pada lat4_5a.php, lalu jalankan lat4_5b.php. Bagaimana hasil tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?
C. Simpulkan apa yang anda peroleh pada latihan 4?

JAWAB :

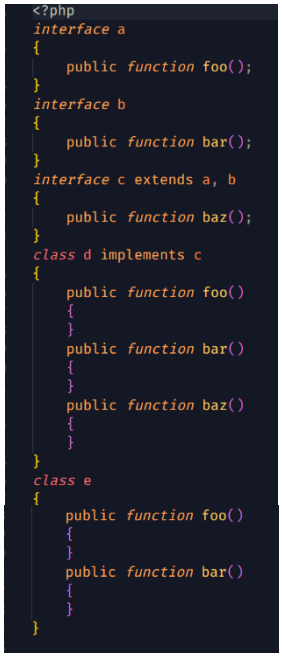
A.



B. Error, pada class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.
C. Kesimpulan:
class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.

LATIHAN 6 :

CODE :



```php
<?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()
    {
    }
}
class e
{
    public function foo()
    {
    }
    public function bar()
    {
    }
}
```

A. Jelaskan maksud dari program di atas?
B. Hapuslah kode baris 27–29, lalu jalankan lat4_6.php. Bagaimana tampilan program di atas?Jelaskan mengapa hal tersebut dapat terjadi?
C. Dari contoh kode diatas, buatlah class baru dengan nama “e” yang mempunyai methodfoo dan **bar**.
D. Simpulkan apa yang anda peroleh darilatihan 6!

JAWAB :

A. Maksud dari program tersebut adalah penggunaan object interface
B. Error, karena pada object interface, ketika kita mengimplementasikan object tersebut, seluruhmethod pada interface harus diimplementasikan seluruhnya. Karena class d mengimplementinterface c, maka method-method pada interface c harus diimplementasikan seluruhnya.
C. 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 :

- Interface didefinisikan dengan “Interface” keyword, mirip dengan deklarasi class biasa,hanya saja definisi atau detail method tidak dituliskan.
- Seluruh method yang dideklarasikan pada interface harus memiliki modifier “public”.
- Untuk mengimplementasikan sebuah interface, kita dapat menggunakan “implement” keyword.
- Seluruh method yang ada pada interface harus diimplementasikan seluruhnya. Sebuahclass bisa mengimplementasikan lebih dari satu interface.

- Class tidak bisa mengimplementasikan dua interface yang mempunyai nama methodyang sama.
- Interface bisa diwariskan seperti class menggunakan “extends”.
- Class yang mengimplementasikan interface harus menggunakan method-method yangada pada interface tersebut dengan nama dan spesifikasi yang sama persis.

LATIHAN 7 :

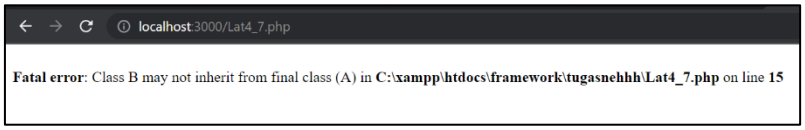
CODE :

```php
<?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();
```

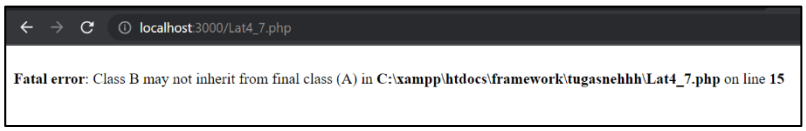
A. Bagaimana tampilan program di atas?Jelaskan mengapa hal tersebut terjadi?b.
B. Modifikasi program di atas dengan menghapus kata final pada kode baris 5 danmenambahkan kata final pada baris 2. Bagaimana tampilan program di atas? Jelaskanmengapa hal tersebut terjadi?c.
C. Simpulkan apa yang anda peroleh dari latihan 7!

JAWAB :

A. Error karena method pada class A tidak dapat di override



B. Tampilan

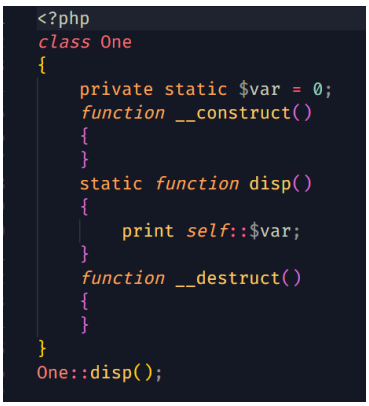


karena tujuan digunakannya keyword final adalah agar class tidak akan di override.

C. KESIMPULAN :
   - final, akan mencegah proses overriding method pada class anak (sub-class)
   - Apabila metode kita berikan status final, maka metode tersebut tidak akan bisadioverr ide, begitu juga pada class, apabila kita berikan status “final” pada deklarasi class maka class tersebut tidak bisa diperpanjang (diwariskan).

LATIHAN 8 :

CODE :



```php
<?php
class One
{
    private static $var = 0;
    function __construct()
    {
    }
    static function disp()
    {
        print self::$var;
    }
    function __destruct()
    {
    }
}
One::disp();
```

Simpulkan apa yang anda peroleh dari latihan 8!

JAWAB :

A. Lat4_8.php mengimplementasikan penggunaan property static dengan modifierprivate.Properti statis dideklarasikan dengan menggunakan katakunci statis sebelum modifier
B. Sifat statis dapat diakses tanpa perlu sebuah contoh objek dari kelas
C. Untuk mengakses metode statis atau properti dari dalam kelas yang sama dapatmenggunakan kata kunci self.
   self :: $ properti;