

Nama : Nafiati Rosidah

NIM : G.211.21.0092

MATERI PRAKTIKUM

Latihan 1 : Class dan Object+Modifier

a. Bagaimana hasil tampilan diatas?

```
Lat4_1.php x Lat4_2a.php Lat4_2b.php
C: > xampp > htdocs > LATIHAN_OOP > Lat4_1.php
1  <?php
2  //class mobil
3  Class Mobil {
4      public $nama;
5      public $merk;
6
7      function getInfo() {
8          echo "Nama Mobil : ".$this->nama. "<br/>";
9          echo "Merk : " . $this->merk. "<br/>";
10     }
11 }
12
13 //bagian main
14 $ferrari=new Mobil();
15 $ferrari->nama="Ferrari 488";
16 $ferrari->merk="Ferrari";
17
18 $ferrari->getInfo() ;
19 ?>
```

localhost/LATIHAN_OOP/Lat4_1.p... A

Nama Mobil : Ferrari 488
Merk : Ferrari

b. Buatlah sebuah method overload getInfo dengan parameter \$a. Lalu jalankan dan amati perubahan yang terjadi.

```
Lat4_1.php x Lat4_2a.php Lat4_2b.php
C: > xampp > htdocs > LATIHAN_OOP > Lat4_1.php
1  <?php
2  //class mobil
3  Class Mobil {
4      public $nama;
5      public $merk;
6      private array $properties = [];
7
8      public function __get($a) {
9          echo "Nama Mobil : $a";
10         return " Ferrari 488";
11     }
12 }
13
14 //bagian main
15 $mobil=new Mobil();
16 echo $mobil->ferrari
17 ?>
```

localhost/LATIHAN_OOP/Lat4_1.php

Nama Mobil : ferrari Ferrari 488

c. Lalu simpulkan apa yang Anda peroleh dari Latihan 1!

Kesimpulan : Pada script diatas untuk menggunakan method overload saya menggunakan magic function dengan menggunakan "private array \$properties = [];" yang fungsinya untuk

mengecek data __get ada atau tidak. Jadi kesimpulanya jika tidak ada property Honda maka otomatis akan mengakes function __get dengan parameter \$a.

Latihan 2 :

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!

Sebelum dimodifikasi:

```
Lat4_1.php x Lat4_2a.php x Lat4_2b.php
C: > xampp > htdocs > LATIHAN_OOP > Lat4_2a.php

1  <?php
2  Class Mahasiswa{
3      public $nama;
4      public $nim;
5
6      function __construct($a,$b) {
7          $this->nama=$a;
8          $this->nim=$b;
9          echo "Kelas telah dibuat<br/><br/>";
10 }
11
12 function cetak() {
13     echo $this->nama. "<br/>".$this->nim."<br/><br/>";
14 }
15
16 function __destruct() {
17     echo "Kelas telah dihancurkan";
18 }
19 }
20 ?>
```

localhost/LATIHAN_OOP/Lat4_2b.php

Kelas telah dibuat

Pennyka
0910683073

Sesudah dimodifikasi:

```
Lat4_1.php Lat4_2a.php x Lat4_2b.php
C: > xampp > htdocs > LATIHAN_OOP > Lat4_2a.php

1  <?php
2  Class Mahasiswa{
3      public $nama;
4      public $nim;
5
6      function __construct($a,$b) {
7          $this->nama=$a;
8          $this->nim=$b;
9          echo "Kelas telah dibuat<br/><br/>";
10 }
11 function __construct(){
12
13 }
14 function cetak() {
15     echo $this->nama. "<br/>".$this->nim."<br/><br/>";
16 }
17
18 function __destruct() {
19     echo "Kelas telah dihancurkan";
20 }
21 }
22 ?>
23
```

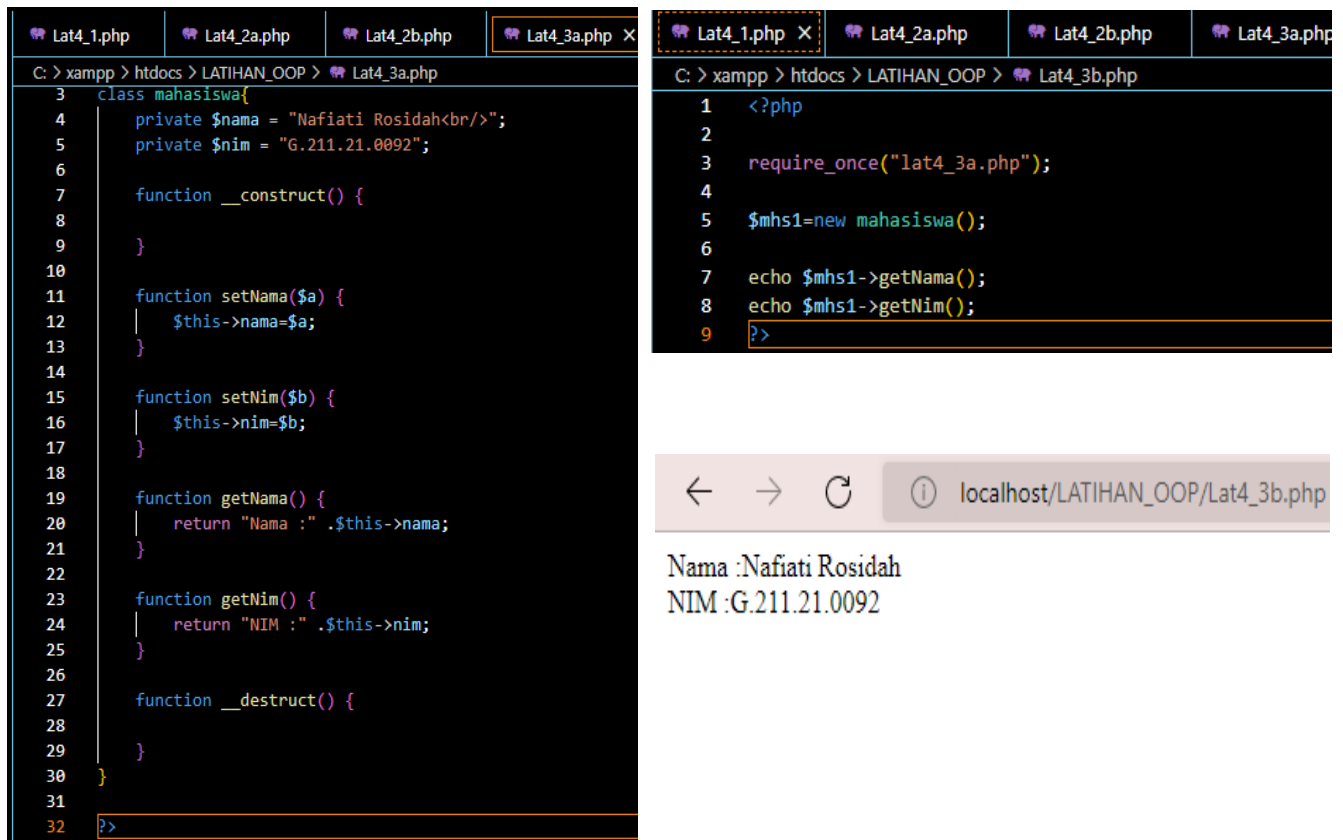
localhost/LATIHAN_OOP/Lat4_2b.php

Fatal error: Cannot redeclare Mahasiswa::__construct() in C:\xampp\htdocs\LATIHAN_OOP\Lat4_2a.php on line 11

Kesimpulan : Pada script yang belum dimodifikasi ketika di eksekusi akan berjalan dengan normal karena script tersebut terdapat 1 construct dan destruct, setelah saya tambahkan construct pada line 11 yang terjadi erorr. Jadi kesimpulanya construct hanya dapat digunakan didalam satu objek tidak bisa lebih.

Latihan 3 :

- Apakah program error? Jika error mengapa hal itu dapat terjadi?
Jawab : Iya program error, karena property nama dan nim menggunakan modifier private jadi hak akses private hanya bisa diakses dari dalam class itu sendiri.
- Rubahlah modifier dari variable nama dan nim menjadi protected dan public. Lalu amati perubahan yang terjadi.
- Modifikasilah Lat4_3b sehingga dapat memberi dan mencetak isi dari nim dan nama dengan modifier private.



```
Lat4_1.php Lat4_2a.php Lat4_2b.php Lat4_3a.php X
C: > xampp > htdocs > LATIHAN_OOP > Lat4_3a.php
3 class mahasiswa{
4     private $nama = "Nafiati Rosidah<br/>";
5     private $nim = "G.211.21.0092";
6
7     function __construct() {
8
9     }
10
11    function setName($a) {
12        | $this->nama=$a;
13    }
14
15    function setNim($b) {
16        | $this->nim=$b;
17    }
18
19    function getName() {
20        | return "Nama :". $this->nama;
21    }
22
23    function getNim() {
24        | return "NIM :". $this->nim;
25    }
26
27    function __destruct() {
28
29    }
30 }
31
32 ?>
```

```
Lat4_1.php X Lat4_2a.php Lat4_2b.php Lat4_3a.php
C: > xampp > htdocs > LATIHAN_OOP > Lat4_3b.php
1 <?php
2
3     require_once("lat4_3a.php");
4
5     $mhs1=new mahasiswa();
6
7     echo $mhs1->getName();
8     echo $mhs1->getNim();
9     ?>
```

localhost/LATIHAN_OOP/Lat4_3b.php

Nama :Nafiati Rosidah
NIM :G.211.21.0092

- Simpulkan apa yang anda peroleh dari latihan 3!
Kesimpulan : pada script modul 3 jika menggunakan modifier private harus mengisi hak akses di dalam class itu sendiri.

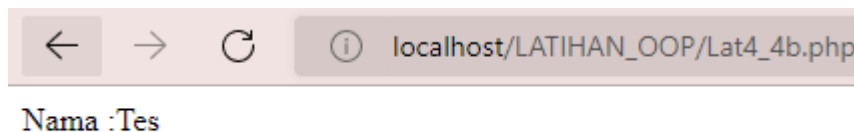
Latihan 4:

Lat4_4a.php

```
C: > xampp > htdocs > LATIHAN_OOP > Lat4_4a.php
1  <?php
2
3  require_once("lat4_3a.php");
4
5  class asisten extends mahasiswa {
6      function __construct() {
7
8      }
9  }
```

Lat4_4b.php

```
C: > xampp > htdocs > LATIHAN_OOP > Lat4_4b.php
1  <?php
2
3  require_once("lat4_4a.php");
4
5  $as = new asisten();
6  $as->setNama("Tes");
7  echo $as->getNama();
8
9  ?>
```



← → ↻ ⓘ localhost/LATIHAN_OOP/Lat4_4b.php

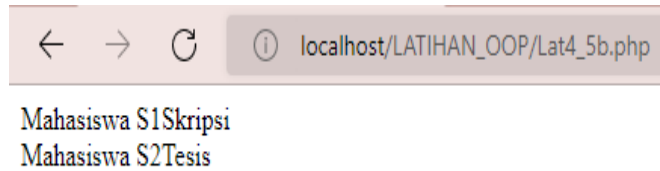
Nama :Tes

Simpulkan apa yang anda peroleh dari latihan 4!

Kesimpulan : Pada latihan 4 file Lat4_4a itu scriptnya adalah membuat class asisten yang terhubung dengan class mahasiswa yang ada di latihan 3. Dan pada Lat4_4b itu terhubung dengan Lat4_4a jadi kita bisa memanggil function dari class asisten yang terhubung dengan class mahasiswa

Latihan 5:

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 diatas? Jelaskan mengapa hal tersebut terjadi?

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

Kesimpulan : class anak yang mewarisi super class harus menuliskan semua method abstrak dari super classnya.

Latihan 6:

a. Jelaskan maksud dari program di atas?

Jawab : pada script latihan 6 itu adalah interface, interface mirip dengan abstract class, yang membedakan adalah di interface semua method otomatis abstract. Selain itu juga di dalam interface tidak boleh memiliki property. Dan juga bisa implements lebih dari satu interface sangat berbeda dengan extends pada class.

b. Hapuslah kode baris 27 - 29, lalu jalankan lat4_6.php. Bagaimana tampilan program diatas? Jelaskan mengapa hal itu terjadi?

```
C:\> xampp > htdocs > LATIHAN_OOP > Lat4_6.php
1  <?php
2  interface a {
3  |   public function foo();
4  |   }
5
6  interface b {
7  |   public function bar();
8  |   }
9
10 interface c extends a, b {
11 |   public function baz();
12 |   }
13
14 class d implements c {
15 |   public function foo() {
16 |   }
17 |
18 |   public function bar() {
19 |   }
20 |
21 |   public function baz() {
22 |   }
23 |
24 |   }
25 }
26 ?>
```

Parse error: syntax error, unexpected token ";", expecting "function" or "const" in C:\xampp\htdocs\LATIHAN_OOP\Lat4_6.php on line 26

Hal ini terjadi karena salah satu function yang implementasi dari abstract nya yang ada di interface tidak tertulis, jadi 3 function dari interface tersebut harus di tulis semuanya.

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

```
19     public function bar() {
20
21     }
22
23     public function baz() {
24
25     }
26 }
27 class e {
28     public function foo() {
29
30     }
31
32     public function bar() {
33
34     }
35 }
36
37 ?>
```

d. Simpulkan apa yang ada peroleh dari latihan 6!

Kesimpulan : jadi kesimpulanya adalah jika kita menggunakan interface maka akan lebih mudah untuk menghubungkan beberapa class. Metode ini juga cukup simple jika sepenuhnya script menggunakan abstract lebih baik menggunakan interface.

Latihan 7 :

a. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut terjadi? Yang terjadi akan error karena kata “final” sudah diterapkan pada function pertama, maka artinya function tersebut tidak bisa di override lagi di class child nya.

```

C: > xampp > htdocs > LATIHAN_OOP > Lat4_7.php
1  <?php
2  class A {
3      final public function disp() {
4          echo "Inside the final function";
5      }
6  }
7
8  class B extends A {
9      final function disp() {
10         echo "Inside the final function";
11     }
12 }
13
14 $obj=new B();
15 $obj->disp();
16
17 ?>

```

Fatal error: Cannot override final method A::disp() in
C:\xampp\htdocs\LATIHAN_OOP\Lat4_7.php on line 9

- b. Modifikasi program di atas dengan menghapus kata final pada kode baris 5 dan menambahkan kata final pada baris 2. Bagaimana tampilan program di atas? Jelaskan mengapa hal tersebut terjadi?

```

C: > xampp > htdocs > LATIHAN_OOP > Lat4_7.php
1  <?php
2  class A {
3      public function disp() {
4          echo "Inside the final function";
5      }
6  }
7
8  class B extends A {
9      final function disp() {
10         echo "Inside the final function";
11     }
12 }
13
14 $obj=new B();
15 $obj->disp();
16
17 ?>

```

← → ↻ ⓘ localhost/LATIHAN_OOP/Lat4_7.php

Inside the final function

- c. Simpulkan apa yang anda peroleh dari latihan 7!

Kesimpulan : pada script latihan 7 itu menurut saya sangat cocok jika kita ingin mengunci implementasi dari sebuah method agar tidak bisa diubah lagi oleh class child nya

Latihan 8:

```
C: > xampp > htdocs > LATIHAN_OOP > Lat4_8.php
1  <?php
2
3  class One {
4      private static $var=0;
5      function __construct() {
6
7      }
8
9      static function disp() {
10         |   print self::$var;
11     }
12
13     function __destruct() {
14
15     }
16 }
17
18 One::disp();
19
20 ?>
```

← → ↻ ⓘ localhost/LATIHAN_OOP/Lat4_8.php