

PEMROGRAMAN WEB DINAMIS PHP OOP II



Dosen : Wahyu Sulistiyo, S. T., M. Kom.

Disusun oleh

Nama : Novita Alya Ramadhani

NIM : 3.34.19.1.17

Kelas : IK-2B

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG
TAHUN 2020**

JOB SHEET 6 PEMROGRAMAN WEB DINAMIS

CLASS DAN OBJEK PADA PHP (LANJUTAN)

I. Tujuan Instruksional Khusus

1. Memahami konsep pewarisan pada object oriented programming
2. Memahami konsep enkapsulasi dalam pewarisan
3. Memahami polimorfisme dalam objek oriented programming

II. Landasan Teori

2. 1. Pewarisan (Inheritance)

Inheritance atau pewarisan dalam OOP dalam PHP adalah mewariskan property dan method baik public maupun protected kepada class lain. Konsep ini sangat berguna jika kita ingin membuat class lagi yang memiliki fungsi mirip dengan class yang sudah ada, sehingga tidak usah membuat lagi method atau property yang memiliki fungsi mirip atau sama. Dengan inheritance kita bisa menghindari duplikasi kode program, atau disebut juga code reuse.

Konsep inheritance membuat sebuah struktur atau hierarki class dalam code program. Class yang akan ‘diturunkan’ bisa disebut sebagai class induk (parent class), super class atau base class. Sedangkan class yang ‘menerima penurunan’ bisa disebut sebagai class anak (child class), sub class, derived class atau heir class.

Tidak semua property dan method dari class induk akan diturunkan. Property dan method dengan hak akses private, tidak akan diturunkan kepada class anak. Hanya property dan method dengan hak akses protected dan public saja yang bisa diakses dari class anak.

Di dalam PHP, inheritance / penurunan dari sebuah class kepada class lain menggunakan kata kunci : “extends”.

```

C: > xampp > htdocs > jobsheet1 > OOP > extends.php
1  <?php
2      class induk
3      {
4          // ... class induk
5      }
6      class anak extends induk
7      {
8          // ..class anak bisa mengakses
9          // ...property dan method class induk
10     }
11  ?>

```

2. 2. Polimorfisme

Dari segi bahasa, polimorfisme berasal dari dua kata bahasa latin yakni poly dan morph. Poly berarti banyak, dan morph berarti bentuk. Polimorfisme berarti banyak bentuk. Di dalam pemrograman berbasis objek, polimorfisme adalah konsep dimana terdapat banyak class yang memiliki signature method yang sama. Implementasi dari method-method tersebut diserahkan kepada tiap class, akan tetapi cara pemanggilan method harus sama. Agar kita dapat “memaksakan” signature method yang sama pada banyak class, class tersebut harus diturunkan dari sebuah abstract class atau object interface.

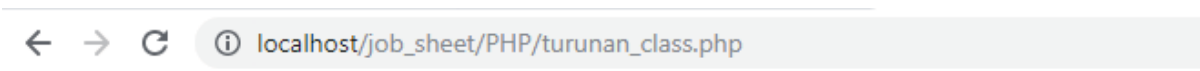
III. Peralatan yang Digunakan

1. PC
2. Peramban web : Internet Explorer, Firefox, Google Chrome, Opera atau Safari
3. Aplikasi server local : XAMPP
4. Aplikasi text editor : Visual Studio Code

IV. Langkah Kerja

4. 1. Latihan Turunan Class

1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



4 jam pertemuan

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama protected2.php

4. Ketikkan barisan kode PHP berikut

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8 </head>
9 <body>
10 <?php
11 // buat class kuliah
12 class kuliah{
13     // property dengan hak akses protected
14     protected $jam_perkuliah = "4 jam pertemuan";
15 }
16 // buat class praktikum
17 class praktikum extends kuliah{
18     public function tampilkan_pertemuan() {
19         return $this->jam_perkuliah;
20     }
21 }
22 // buat objek dari class praktikum (instansiasi)
23 $kuliah_baru = new praktikum();
24 // jalankan method
25 echo $kuliah_baru->tampilkan_pertemuan(); // "Intel Core i7-4790 3.6Ghz"
26 ?>
27 </body>
28 </html>
```

5. Buka file di peramban web

6. Simpan tampilan peramban web dengan menekan Alt+PrtScn

4. 2. Latihan Inheritance

1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



2. Hidupkan komputer

3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama inheritance.php

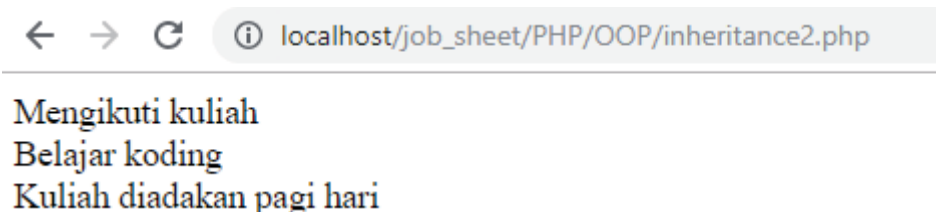
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > inheritance.php
1  <?php
2  // buat class induk: mahasiswa
3  class mahasiswa {
4      public $nama;
5      public $nim;
6      public $alamat;
7      public function kuliah() {
8          return "Mengikuti perkuliahan";
9      }
10 }
11 // turunkan class mahasiswa ke mhs_informatika
12 class mhs_informatika extends mahasiswa{
13     public function belajar_pemrograman() {
14         return "nama: $this->nama, nim: $this->nim,
15         alamat: $this->alamat";
16     }
17 }
18 // buat objek dari class mhs_informatika (instansiasi)
19 $mhs_informatika_baru = new mhs_informatika();
20 // isi property objek
21 $mhs_informatika_baru->nama = "Sofia";
22 $mhs_informatika_baru->nim = "12345678";
23 $mhs_informatika_baru->alamat = "Semarang";
24 //panggil method objek
25 echo $mhs_informatika_baru->kuliah();
26 echo "<br />";
27 echo $mhs_informatika_baru->belajar_pemrograman();
28 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 3. Latihan Inheritance dengan 3 Class
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama inheritance.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > inheritance2.php
1  <?php
2  // buat class mahasiswa
3  class mahasiswa {
4      protected function kuliah() {
5          return "Mengikuti kuliah";
6      }
7  }
8  // turunkan class mahasiswa ke mhs_informatika
9  class mhs_informatika extends mahasiswa {
10     protected function belajar_pemrograman() {
11         return "Belajar koding";
12     }
13 }
14 // turunkan class mhs_informatika ke kelas_reguler
15 class kelas_reguler extends mhs_informatika {
16     protected function kuliah_pagi() {
17         return "Kuliah diadakan pagi hari";
18     }
19     public function kuliah_semua(){
20         $a = $this->kuliah();
21         $b = $this->belajar_pemrograman();
22         $c = $this->kuliah_pagi();
23         return "$a <br /> $b <br /> $c";
24     }
25 }
26 // buat objek dari class mhs_informatika (instansiasi)
27 $masuk_kuliah = new kelas_reguler();
28 //panggil method objek
29 echo $masuk_kuliah->kuliah_semua();
30 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 4. Latihan Cara Mengakses Property dan Method Parent Class
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ ⓘ localhost/job_sheet/PHP/OOP/property_parent.php

Nama mahasiswa: Jerry, nim : 987654321, Alamat : Malang

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama property_parent.php

4. Ketikkan barisan kode PHP berikut

```
C: > xampp > htdocs > job_sheet > PHP > OOP > property_parent.php
1  <?php
2  // buat class mahasiswa
3  class mahasiswa {
4      public function data_mahasiswa() {
5          return "Nama mahasiswa: Mariska,
6              nim : 12345678, Alamat : Yogyakarta";
7      }
8  }
9  // turunkan class mahasiswa ke mhs_informatika
10 class mhs_informatika extends mahasiswa {
11     public function data_mahasiswa() {
12         return "Nama mahasiswa: Jerry,
13             nim : 987654321, Alamat : Malang";
14     }
15 }
16 // buat objek dari class laptop (instansiasi)
17 $mahasiswa_baru = new mhs_informatika();
18 //panggil method lihat_spec()
19 echo $mahasiswa_baru->data_mahasiswa();
20 ?>
```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 5. Latihan Cara Memanggil Method Parent Class dari Child Class
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



← → ↻ ⓘ localhost/job_sheet/PHP/OOP/method_parent.php

Nama bunga: anggrek, Warna : ungu, Lokasi : di pohon mangga
Nama bunga: mawar, Warna : merah, Lokasi : taman depan

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama method_parent.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > scope_resolution_operator.php
1  <?php
2  // buat class bunga
3  class bunga {
4      public function lihat_bunga() {
5          return "Nama bunga: mawar,
6              Warna : merah, Lokasi : taman depan";
7      }
8  }
9  // turunkan class bunga ke anggrek
10 class anggrek extends bunga {
11     public function lihat_bunga() {
12         return "Nama bunga: anggrek,
13             Warna : ungu, Lokasi : di pohon mangga";
14     }
15     public function lihat_jenis_bunga() {
16         return parent::lihat_bunga();
17     }
18 }
19 // buat objek dari class anggrek (instansiasi)
20 $anggrek_baru = new anggrek();
21 //panggil method lihat_bunga()
22 echo $anggrek_baru->lihat_bunga();
23 echo "<br />";
24 //panggil method lihat_jenis_bunga()
25 echo $anggrek_baru->lihat_jenis_bunga();
26 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 6. Latihan Cara Memanggil Property Parent Class dari Child Class
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ 📄 localhost/job_sheet/PHP/OOP/parent_property.php

Nama mahasiswa: Farina, nim : 87654321, Alamat : Kudus
 Nama mahasiswa: Wira, nim : 12345678, Alamat : Solo

Fatal error: Uncaught Error: Access to undeclared static property: mahasiswa::\$nama in C:\xampp\htdocs\job_sheet\PHP\OOP\parent_property.php:21 Stack trace: #0 C:\xampp\htdocs\job_sheet\PHP\OOP\parent_property.php(32): mhs_informatika->lihat_nama_mahasiswa() #1 {main} thrown in C:\xampp\htdocs\job_sheet\PHP\OOP\parent_property.php on line 21

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama parent_property.php
4. Ketikkan barisan kode PHP berikut


```

C: > xampp > htdocs > job_sheet > PHP > OOP > parent_property.php
1  <?php
2  // buat class mahasiswa
3  class mahasiswa {
4      public $nama = "Wira";
5      public function data_mahasiswa() {
6          return "Nama mahasiswa: Wira,
7              nim : 12345678, Alamat : Solo";
8      }
9  }
10 // turunkan class mahasiswa ke mhs_informatika
11 class mhs_informatika extends mahasiswa {
12     public $nama = "Farina";
13     public function data_mahasiswa() {
14         return "Nama mahasiswa: Farina,
15             nim : 87654321, Alamat : Kudus";
16     }
17     public function lihat_data_mahasiswa() {
18         return parent::data_mahasiswa();
19     }
20     public function lihat_nama_mahasiswa() {
21         return parent::$nama;
22     }
23 }
24 // buat objek dari class mhs_informatika (instansiasi)
25 $mahasiswa_baru = new mhs_informatika();
26 //panggil method data_mahasiswa()
27 echo $mahasiswa_baru->data_mahasiswa();
28 echo "<br />";
29 //panggil method lihat_spec_komputer()
30 echo $mahasiswa_baru->lihat_data_mahasiswa();
31 //panggil method lihat_merk_komputer()
32 echo $mahasiswa_baru->lihat_nama_mahasiswa();
33 ?>

```

5. Buka file di peramban web
6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 7. Latihan Cara Memanggil Constructor dan Destructor Parent
 1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ ⓘ localhost/job_sheet/PHP/OOP/parent_constructor.php

Constructor dari class komputer

Belajar OOP PHP

Destructor dari class komputer

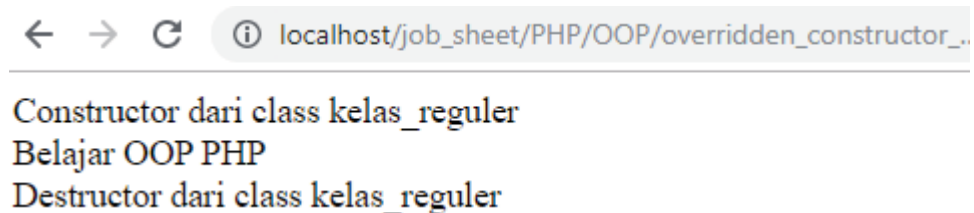
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama parent_constructor.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > parent_constructor_destructor.php
1  <?php
2  // buat class komputer
3  class mahasiswa {
4      // buat constructor class komputer
5      public function __construct() {
6          echo "Constructor dari class mahasiswa <br />";
7      }
8      // buat destructor class komputer
9      public function __destruct() {
10         echo "Destructor dari class mahasiswa <br />";
11     }
12 }
13 // turunkan class mahasiswa ke mhs_informatika
14 class mhs_informatika extends mahasiswa {
15 }
16 // turunkan class mhs_informatika ke kelas_reguler
17 class kelas_reguler extends mhs_informatika {
18 }
19 // buat objek dari class kelas_reguler (instansiasi)
20 $mahasiswa_baru = new kelas_reguler();
21 echo "Belajar OOP PHP <br />";
22 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 8. Latihan Overridden Constructor dan Destructor Parent
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



← → ↻ ⓘ localhost/job_sheet/PHP/OOP/overridden_constructor_..

Constructor dari class kelas_reguler
 Belajar OOP PHP
 Destructor dari class kelas_reguler

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama overridden_constructor_destructor.php
4. Ketikkan barisan kode PHP berikut

```

c: > xampp > htdocs > job_sheet > PHP > OOP > overridden_constructor_destructor.php
1  <?php
2  // buat class komputer
3  class mahasiswa {
4      // buat constructor class komputer
5      public function __construct() {
6          echo "Constructor dari class mahasiswa <br />";
7      }
8      // buat destructor class komputer
9      public function __destruct() {
10         echo "Destructor dari class mahasiswa <br />";
11     }
12 }
13 // turunkan class mahasiswa ke mhs_informatika
14 class mhs_informatika extends mahasiswa {
15     // buat constructor class mhs_informatika
16     public function __construct() {
17         echo "Constructor dari class mhs_informatika <br />";
18     }
19     // buat destructor class mhs_informatika
20     public function __destruct() {
21         echo "Destructor dari class mhs_informatika";
22     }
23 }
24 // turunkan class mhs_informatika ke kelas_reguler
25 class kelas_reguler extends mhs_informatika {
26     // buat constructor class kelas_reguler
27     public function __construct() {
28         echo "Constructor dari class kelas_reguler <br />";
29     }
30     // buat destructor class kelas_reguler
31     public function __destruct() {
32         echo "Destructor dari class kelas_reguler <br />";
33     }
34 }
35 // buat objek dari class kelas_reguler (instansiasi)
36 $mhs_baru = new kelas_reguler();
37 echo "Belajar OOP PHP <br />";
38 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 9. Latihan Enkapsulasi pada Pewarisan
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↺ ① localhost/job_sheet/PHP/OOP/enkapsulasi_pewarisan.php

Notice: Undefined property: mhs_informatika::\$nim in C:\xampp\htdocs\job_sheet\PHP\OOP\enkapsulasi_pewarisan.php on line 18
Halo NIM saya

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama enkapsulasi_pewarisan.php

4. Ketikkan barisan kode PHP berikut

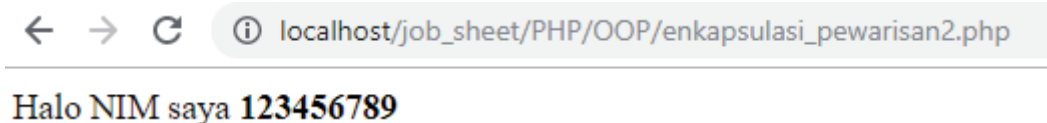
```
C: > xampp > htdocs > job_sheet > PHP > OOP > enkapsulasi_pewarisan.php
1  <?php
2  // The parent class
3  class Mahasiswa {
4      //The $model property is private, thus it can be accessed
5      // only from inside the class
6      private $nim;
7      //Public setter method
8      public function setNim($nim)
9      {
10         $this->nim = $nim;
11     }
12 }
13 // The child class
14 class mhs_informatika extends Mahasiswa{
15     //Tries to get a private property that belongs to the parent
16     public function hello()
17     {
18         return "Halo NIM saya <b>" . $this->nim . "</b><br />";
19     }
20 }
21 //Create an instance from the child class
22 $mhs_informatika1 = new mhs_informatika();
23 //Set the class model name
24 $mhs_informatika1->setNim('123456789');
25 //Get the class model name
26 echo $mhs_informatika1->hello();
27 ?>
```

5. Buka file di peramban web

6. Simpan tampilan peramban web dengan menekan Alt+PrtScn

4. 10. Latihan Enkapsulasi pada Pewarisan 2

1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



← → ↻ ⓘ localhost/job_sheet/PHP/OOP/enkapsulasi_pewarisan2.php

Halo NIM saya **123456789**

2. Hidupkan komputer

3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama enkapsulasi_pewarisan2.php

4. Ketikkan barisan kode PHP berikut


```

C: > xampp > htdocs > job_sheet > PHP > OOP > enkapsulasi_pewarisan2.php
1  <?php
2  // The parent class
3  class Mahasiswa {
4      //The $model property is private, thus it can be accessed
5      // only from inside the class
6      protected $nim;
7      //Public setter method
8      public function setNim($nim)
9      {
10         $this->nim = $nim;
11     }
12 }
13 // The child class
14 class mhs_informatika extends Mahasiswa{
15     //Tries to get a private property that belongs to the parent
16     public function hello()
17     {
18         return "Halo NIM saya <b>" . $this->nim . "</b><br />";
19     }
20 }
21 //Create an instance from the child class
22 $mhs_informatika1 = new mhs_informatika();
23 //Set the class model name
24 $mhs_informatika1->setNim('123456789');
25 //Get the class model name
26 echo $mhs_informatika1->hello();
27 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 11. Latihan Getter dan Setter
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ ⓘ localhost/job_sheet/PHP/OOP/getter_setter.php

Tinggi seorang anak adalah 120 cm

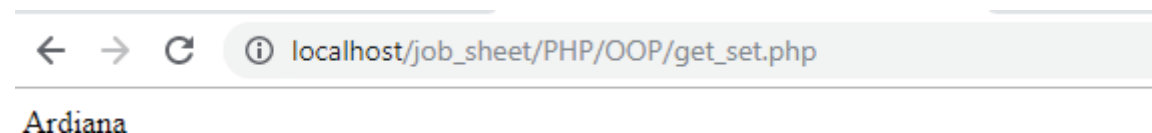
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama getter_setter.php
4. Ketikkan barisan kode PHP berikut


```

C: > xampp > htdocs > job_sheet > PHP > OOP > getter_setter.php
1  <?php
2  class AnakAnak
3  {
4      private $tinggi;
5      public function setTinggi($tinggi)
6      {
7          if ($tinggi > 30)
8          {
9              $this->tinggi= $tinggi;
10         }
11     }
12     public function getTinggi()
13     {
14         echo "Tinggi seorang anak adalah " . $this->tinggi . " cm";
15     }
16 }
17 $anak1= new AnakAnak;
18 $anak1->setTinggi(120);
19 $anak1->getTinggi();
20 ?>

```

5. Buka file di peramban web
6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 12. Latihan Getter dan Setter 2
7. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



8. Hidupkan komputer
9. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama get_set.php
10. Ketikkan barisan kode PHP berikut

```

1  <?php
2  class Kelasku
3  {
4      /* atribut bersifat private, tidak dapat diakses secara langsung */
5      private $nama;
6      /* Constructor */
7      public function __construct()
8      {
9          $this->nama = '';
10     }
11     /* Getter function digunakan untuk membaca method */
12     public function get_nama()
13     {
14         return $this->nama;
15     }
16     /* Setter function untuk merubah method */
17     public function set_nama($new_nama)
18     {
19         if ($this->is_valid_name($new_nama))
20         {
21             $this->nama = $new_nama;
22         }
23     }
24     private function is_valid_name($nama)
25     {
26         $valid = TRUE;
27         /* cek jika panjang string diantara 3 hingga 16 */
28         if (mb_strlen($nama) < 3)
29         {
30             $valid = FALSE;
31         }
32         else if (mb_strlen($nama) > 16)
33         {
34             $valid = FALSE;
35         }
36         return $valid;
37     }
38 }
39 $mc = new Kelasku();
40 $mc->set_nama('Ardiana');
41 echo $mc->get_nama();
42 ?>

```

11. Buka file di peramban web
12. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 13. Latihan Final Method
 1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ ① localhost/job_sheet/PHP/OOP/final_method.php

Fatal error: Cannot override final method Mahasiswa::belajar() in C:\xampp\htdocs\job_sheet\PHP\OOP\final_method.php on line 15

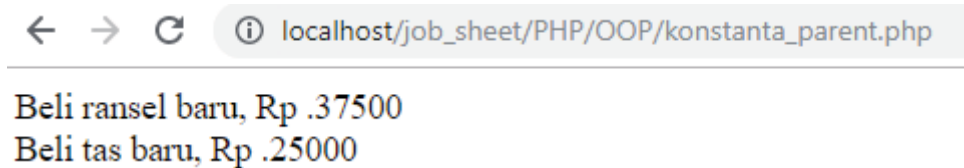
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama final_method.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > final_method.php
5   {
6       return "Belajar";
7   }
8   }
9   //The child class has hello method that tries to override the hello method in the parent
10  class mhs_informatika extends Mahasiswa {
11      public function belajar()
12      {
13          return "PHP";
14      }
15  }
16  //Create a new object
17  $mhs_informatika1 = new mhs_informatika();
18  //Get the result of the hello method
19  echo $mhs_informatika1 -> belajar();
20  ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 14. Latihan Mengakses Konstanta Parent
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



← → ↻ ⓘ localhost/job_sheet/PHP/OOP/konstanta_parent.php

Beli ransel baru, Rp .37500
Beli tas baru, Rp .25000

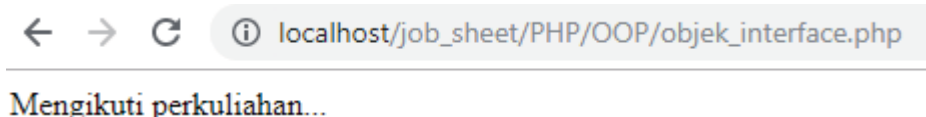
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama konstanta_parent.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > 📁 konstanta_parent.php
1  <?php
2  // buat class tas
3  class tas {
4      // buat konstanta class tas
5      const DISKON = 0.1;
6  }
7  // turunkan class tas ke class ransel
8  class ransel extends tas {
9      // buat konstanta class ransel
10     const DISKON = 0.15;
11
12     // buat method dengan konstanta class tas
13     public function beli_tas($harga){
14         return "Diskon tas baru, Rp ".$harga*parent::DISKON;
15     }
16
17     // buat method dengan konstanta class ransel
18     public function beli_ransel($harga){
19         return "Diskon ransel baru, Rp ".$harga*self::DISKON;
20     }
21 }
22 // buat objek dari class ransel (instansiasi)
23 $ransel_baru=new ransel();
24 echo $ransel_baru->beli_ransel(250000);
25 echo "<br />";
26 echo $ransel_baru->beli_tas(250000);
27 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 15. Latihan Objek Interface
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



← → ↻ ⓘ localhost/job_sheet/PHP/OOP/objek_interface.php

Mengikuti perkuliahan...

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama objek_interface.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > objek_interface.php
1  <?php
2      interface mahasiswa{
3          public function masuk_kuliah();
4          public function praktikum();
5      }
6      class mhs_informatika implements mahasiswa{
7          public function masuk_kuliah(){
8              return "Mengikuti perkuliahan...";
9          }
10         public function praktikum(){
11             return "Melakukan prkatikum...";
12         }
13     }
14     $mhs_baru = new mhs_informatika();
15     echo $mhs_baru->masuk_kuliah();
16     ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 16. Latihan Objek Interface 2
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ ⓘ localhost/job_sheet/PHP/OOP/objek_interface2.php

Fatal error: Class mhs_informatika contains 1 abstract method and must therefore be declared abstract or implement the remaining methods (mahasiswa:praktikum) in C:\xampp\htdocs\job_sheet\PHP\OOP\objek_interface2.php on line 6

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama objek_interface2.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > objek_interface2.php
1  <?php
2      interface mahasiswa{
3          public function masuk_kuliah();
4          public function praktikum();
5      }
6      class mhs_informatika implements mahasiswa{
7          public function masuk_kuliah(){
8              return "Mengikuti perkuliahan...";
9          }
10     }
11     $mhs_baru = new mhs_informatika();
12     ?>

```

5. Buka file di peramban web

6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 17. Latihan Objek Interface dengan Method Protected
 1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

← → ↻ ⓘ localhost/job_sheet/PHP/OOP/objek_interface_protected.php

Fatal error: Access type for interface method mahasiswa::praktikum() must be omitted in C:\xampp\htdocs\job_sheet\PHP\OOP\objek_interface_protected.php on line 4

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama objek_interface_protected.php
4. Ketikkan barisan kode PHP berikut

```
C: > xampp > htdocs > job_sheet > PHP > OOP > objek_interface_protected.php
1  <?php
2      interface mahasiswa{
3          public function masuk_kuliah();
4          protected function praktikum();
5      }
6      class mhs_informatika implements mahasiswa{
7          public function masuk_kuliah(){
8              return "Mengikuti perkuliahan...";
9          }
10         public function praktikum(){
11             return "Melakukan prkatikum...";
12         }
13     }
14     $mhs_baru = new mhs_informatika();
15     echo $mhs_baru->masuk_kuliah();
16     ?>
```

5. Buka file di peramban web
6. Simpan tampilan peramban web dengan menekan Alt+PrtScn.
4. 18. Latihan Objek Interface dengan Method Protected dan Private
 1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :

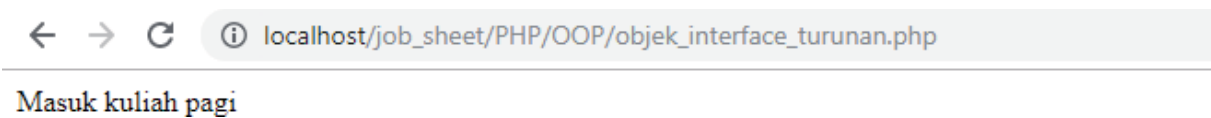
← → ↻ ⓘ localhost/job_sheet/PHP/OOP/objek_interface_protected_and_private.php

Fatal error: Access level to mhs_informatika::masuk_kuliah() must be public (as in class mahasiswa) in C:\xampp\htdocs\job_sheet\PHP\OOP\objek_interface_protected_and_private.php on line 6

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama objek_interface_protected_and_private.php
4. Ketikkan barisan kode PHP berikut

```
C: > xampp > htdocs > job_sheet > PHP > OOP > objek_interface_protected_and_private.php
1  <?php
2      interface mahasiswa{
3          public function masuk_kuliah();
4          public function praktikum();
5      }
6      class mhs_informatika implements mahasiswa{
7          protected function masuk_kuliah(){
8              return "Mengikuti perkuliahan...";
9          }
10         private function praktikum(){
11             return "Melakukan prkatikum...";
12         }
13     }
14     $mhs_baru = new mhs_informatika();
15     echo $mhs_baru->masuk_kuliah();
16     ?>
```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 19. Latihan Objek Interface yang diturunkan
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



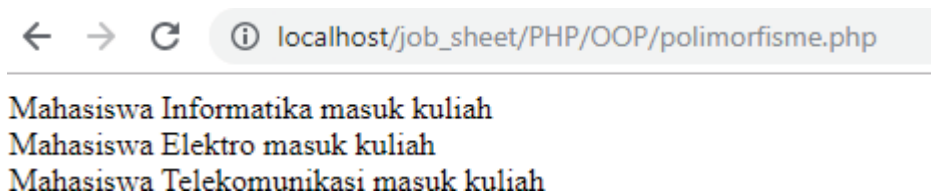
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama objek_interface_turunan.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > objek_interface_turunan.php
1  <?php
2  interface mahasiswa{
3      public function masuk_kuliah();
4      public function praktikum();
5  }
6  interface mahasiswa_reguler extends mahasiswa{
7      public function kelas_pagi();
8  }
9  class mhs_informatika implements mahasiswa_reguler{
10     public function masuk_kuliah(){
11         return "Mengikuti kuliah...";
12     }
13     public function praktikum(){
14         return "Melakukan praktikum...";
15     }
16     public function kelas_pagi(){
17         return "Masuk kuliah pagi";
18     }
19 }
20 $mhs_baru = new mhs_informatika();
21 echo $mhs_baru->kelas_pagi();
22 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 20. Latihan Polimorfisme
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



← → ↻ ⓘ localhost/job_sheet/PHP/OOP/polimorfisme.php

Mahasiswa Informatika masuk kuliah
 Mahasiswa Elektro masuk kuliah
 Mahasiswa Telekomunikasi masuk kuliah

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama poliformisme.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > OOP > polimorfisme.php
1  <?php
2  abstract class mahasiswa{ // buat abstract class
3  |   abstract public function masuk_kuliah();// buat abstract method
4  | }
5  class mhs_informatika extends mahasiswa{
6  |   public function masuk_kuliah(){
7  |   |   return "Mahasiswa Informatika masuk kuliah";
8  |   }
9  | }
10 class mhs_elektro extends mahasiswa{
11 |   public function masuk_kuliah(){
12 |   |   return "Mahasiswa Elektro masuk kuliah";
13 |   }
14 | }
15 class mhs_telekomunikasi extends mahasiswa{
16 |   public function masuk_kuliah(){
17 |   |   return "Mahasiswa Telekomunikasi masuk kuliah";
18 |   }
19 | }
20 // buat objek dari class diatas
21 $mhs_informatika_baru = new mhs_informatika();
22 $mhs_elektro_baru = new mhs_elektro();
23 $mhs_telekomunikasi_baru = new mhs_telekomunikasi();
24 // buat fungsi untuk memproses objek
25 function mhs_masuk_kuliah($objek_komputer){
26 |   return $objek_komputer->masuk_kuliah();
27 | }
28 // jalankan fungsi
29 echo mhs_masuk_kuliah($mhs_informatika_baru);echo "<br />";
30 echo mhs_masuk_kuliah($mhs_elektro_baru);echo "<br />";
31 echo mhs_masuk_kuliah($mhs_telekomunikasi_baru);
32 ?>

```

5. Buka file di peramban web
6. Simpan tampilan peramban web dengan menekan Alt+PrtScn.

V. Lembar Kerja

No	Nama File	Hasil Program	Keterangan
1.	latihan.php	<gambar hasil program>	

VI. Latihan Soal

1. Buatlah dengan konsep OOP minimal 6 Class yang diwarisi dari sebuah Class dengan menggunakan method dengan enkapsulasi.

Prodi : D3 Teknik Informatika
Ada 5 Kelas

Prodi : D3 Teknik Elektro
Ada 4 Kelas

Prodi : D3 Teknik Telekomunikasi
Ada 2 kelas

Prodi : D4 Teknik Telekomunikasi
Ada 2 kelas

Prodi : D3 Teknik Listrik
Ada 2 kelas

Prodi : D4 Teknik Rekayasa Komputer
Ada 2 kelas

2. Dari tugas nomor 1 buatlah minimal 3 interface yang menerapkan konsep poliformisme.

```
localhost/job_sheets/PHP%20OOP/job6_2.php

IK Ada lab
LT ada lab
TI ada lab
```

3. Buatlah program yang yang dapat berbagi method yang bersifat private.

```
localhost/job_sheets/PHP%20OOP/job6_3.php

Pohon memiliki ciri berbunga dan berbauh
```

```
job_sheets > PHP OOP > job6_3.php > Pohon > $ciri
1  <?php
2  class Pohon{
3      private $ciri = "Pohon memiliki ciri berbunga dan berbauh";
4
5      public function sifat(){
6          return $this->ciri;
7      }
8  }
9
10     class Mangga extends Pohon{
11         public function sifat(){
12             return $this->ciri;
13         }
14     }
15     $muda = new Mangga();
16     $muda = new Pohon();
17     echo $muda->sifat();
18
19  ?>
```

VII. Kesimpulan

Usai mengerjakan job sheet ini, dapat dimengerti konsep pewarisan dimana membuat struktur parent class yang menurunkan class lain (child class), enkapsulasi dalam pewarisan, dan polimorfism dalam OOP yang mana ada banyak class yang memiliki method sama..