

JOB SHEET 5 PEMROGRAMAN WEB DINAMIS

CLASS DAN OBJEK PADA PHP

I. Tujuan Instruksional Khusus

1. Memahami konsep object oriented programming
2. Memahami dan menuliskan Class dan Objek
3. Memahami konsep Class Public, Protected dan Private
4. Memahami konsep Constructor dan Destructor

II. Landasan Teori

2. 1. Pengertian Pemrograman Berbasis Objek

Objek Oriented Programming atau dalam bahasa Indonesia diartikan Pemrograman Berbasis Objek, adalah salah satu cara membuat program (programming paradigm) dengan memecah alur program menjadi modul-modul sederhana yang disebut dengan objek. Setiap objek akan memiliki fungsi dan tugas tersendiri. OOP berbeda dengan prosedural programming yang memecah program menjadi fungsi-fungsi / prosedural.

Pemrograman Berbasis Objek adalah sebuah tata cara pembuatan program (programming paradigm) dengan menggunakan konsep “objek” yang memiliki data (atribut yang menjelaskan tentang objek) prosedur (function) yang dikenal dengan method.

Dalam pengertian sederhananya, OOP adalah konsep pembuatan program dengan memecah permasalahan program dengan menggunakan objek. Objek dapat diumpamakan dengan ‘fungsi khusus’ yang bisa berdiri sendiri. Untuk membuat sebuah aplikasi, berbagai objek akan saling bertukar data untuk mencapai hasil akhir.

Berbeda dengan konsep fungsi atau function di dalam pemrograman, sebuah objek bisa memiliki data dan function tersendiri. Setiap objek ditujukam untuk mengerjakan sebuah tugas, dan menghasilkan nilai akhir untuk selanjutnya dapat ditampilkan atau digunakan oleh objek lain.

2. 2. Pemrograman Berbasis Objek dengan PHP

Objek Oriented Programming telah menjadi standar dalam dunia pemrograman, termasuk PHP. Walaupun kita bisa membuat program PHP tanpa menggunakan OOP

sama sekali, namun untuk membuat aplikasi yang lebih fleksibel, programmer PHP akan beralih menggunakan OOP.

Jika kita telah menguasai pemrograman PHP dasar seperti tipe data, array, dan fungsi, mempelajari pemrograman objek PHP adalah langkah berikutnya. Fitur dan desain kode yang ditawarkan dengan membuat program menggunakan objek akan sangat memudahkan kita dalam merancang aplikasi website modern dan memiliki fleksibilitas yang tinggi. Framework PHP berbasis framework telah menggunakan OOP, seperti Code Igniter, Yii Framework, Symfony dan Laravel.

2. 3. Pengertian Class, Object, Property dan Method

Pemrograman berbasis objek tidak hanya berisi 'object'. OOP juga terdiri dari class, object, property dan method. Class adalah 'cetak biru' atau 'blueprint' dari object. Class hanya untuk membuat kerangka dasar. Yang akan kita pakai nantinya adalah hasil cetakan dari class, yakni object. Sebagai analogi, class bisa diibaratkan dengan laptop atau notebook. Kita tahu bahwa laptop memiliki ciri-ciri seperti merk, memiliki keyboard, memiliki processor dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah laptop. Selain memiliki ciri-ciri, sebuah laptop juga bisa dikenakan tindakan, seperti : menhidupkan laptop atau mematikan laptop.

Class dalam analogi ini gambaran umum tentang sebuah benda. Di dalam pemrograman nantinya, contoh class seperti: koneksi_database dan profile_user. Dalam PHP, penulisan class diawali dengan keyword class, kemudian diikuti dengan nama dari class. Aturan penulisan nama class sama seperti aturan penulisan variabel dalam PHP, yakni diawali dengan huruf atau underscore untuk karakter pertama, kemudian dapat diikuti dengan huruf, underscore atau angka untuk karakter kedua dan selanjutnya. Isi dari class berada dalam tanda kurung kurawal. Berikut adalah contoh penulisan class dalam PHP :

```
1  <?php
2  class laptop {
3      // isi dari class laptop...
4  }
5  ?>
```

Property (atau disebut dengan atribut) adalah data yang terdapat dalam sebuah class. Melanjutkan analogi tentang laptop, property dari laptop bisa berupa merk, warna, jenis processor, ukuran layar dan lain-lain. Jika telah terbiasa dengan program PHP, property sebenarnya hanyalah variabel yang terletak di dalam class. Seluruh aturan dan tipe data yang biasa diinput ke dalam variabel, bisa juga diinput kedalam property. Aturan tata cara penamaan property sama dengan aturan penamaan variabel. Berikut adalah contoh penulisan class dengan penambahan property :

```
1  <?php
2  class laptop {
3      var $pemilik;
4      var $merk;
5      var $ukuran_layar;
6      // lanjutan isi dari class laptop...
7  }
8  ?>
```

Dari contoh di atas, \$pemilik, \$merk dan \$ukuran_layar adalah property dari class laptop. Seperti yang dapat kita lihat, penulisan property di dalam PHP sama dengan cara penulisan variabel, yakni menggunakan tanda dollar (\$). Sebuah class tidak harus memiliki property.

Method adalah tindakan yang bisa dilakukan di dalam class. Jika menggunakan analogi class laptop kita, maka contoh method adalah : menghidupkan laptop, mengganti cover laptop, dan berbagai tindakan lain.

Method pada dasarnya adalah function yang berada di dalam class. Seluruh fungsi dan sifar fungsi bisa diterapkan ke dalam method, seperti argumen/parameter, mengembalikan nilai (dengan keyword return), dan lain-lain. Berikut adalah contoh penulisan class dengan penambahan method :

```

1  <?php
2  class laptop {
3      function hidupkan_laptop() {
4          //... isi dari method hidupkan_laptop
5      }
6
7      function matikan_laptop() {
8          //... isi dari method matikan_laptop
9      }
10
11     ... //isi dari class laptop
12 }
13 ?>

```

Dari contoh di atas, function `hidupkan_laptop()` dan function `matikan_laptop()` adalah method dari class `laptop`. Seperti yang kita lihat, bahwa penulisan method di dalam PHP sama dengan cara penulisan function. Sebuah class tidak harus memiliki method.

Object adalah hasil cetak dari class, atau hasil ‘konkrit’ dari class. Jika menggunakan analogi class `laptop`, maka objek dari class `laptop` bisa berupa : `laptop_anto`, `laptop_dina` dan lain-lain. Objek dari class `laptop` akan memiliki seluruh ciri-ciri `laptop`, yaitu property dan methodnya.

Proses ‘mencetak’ objek dari class ini disebut dengan ‘instansiasi’ (atau instantiation dalam bahasa inggris). Pada PHP, proses instansiasi dilakukan dengan menggunakan keyword ‘`new`’. Hasil cetakan class akan disimpan dalam variabel untuk selanjutnya digunakan dalam proses program. Sebagai contoh, berikut adalah cara membuat objek `laptop_andi` dan `laptop_anto` yang dibuat dari class `laptop` :

```

1  <?php
2  class laptop {
3      //... isi dari class laptop
4  }
5
6  $laptop_andi = new laptop();
7  $laptop_anto = new laptop();
8  ?>

```

Dari contoh di atas, `$laptop_andi` dan `$laptop_anto` merupakan objek dari class `laptop`. Kedua objek ini akan memiliki seluruh property dan method yang telah dirancang dari class `laptop`.

2. 4. Pengkapsulan (Enkapsulasi)

Enkapsulasi / pengkapsulan adalah konsep melindungi kode sumber yang ada pada Class dari pengeksesannya dari luar Class. Merupakan konsep pembatasan akses, terhadap property dan juga method yang ada pada sebuah Class, untuk diakses dari luar Class. Dengan proses pengkapsulan ini, seorang Developer dapat menentukan apakah property dan juga method yang dibuat dapat digunakan diluar Class atay hanya dapat digunakan oleh Class itu sendiri dan juga turunannya.

Hal ini dilakukan untuk melindungi code yang ada didalam Property dan juga Method tersebut. Agar nilai-nilai tidak dirubah-ubah oleh Developer lain, karena sudah didefinisikan sebagai nilai mutlak, dan biasanya kalau dirubah, maka program tidak akan berjalan sebagaimana yang diharapkan.

Proses pengkapsulan terbagi kedalam 3 cara, yakni Public, Protected dan juga Private. 3 Keyword ini didefinisikan sebelum pendefinisian Property dan Method.

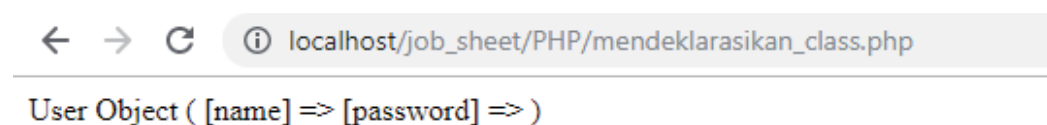
III. Peralatan yang Digunakan

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 Mendeklarasikan Class

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



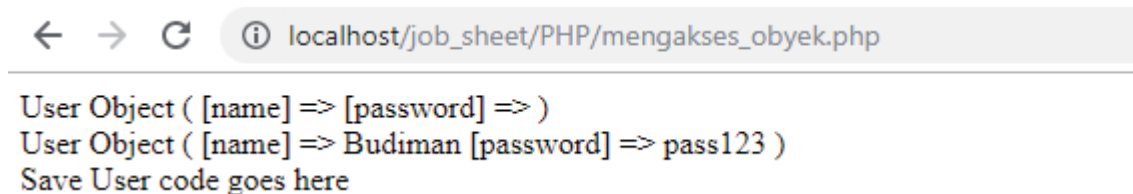
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama mendeklarasikan_kelas.php
4. Ketikkan barisan kode PHP berikut

```

c > xampp > htdocs > job_sheet > PHP > mendefinisikan_class.php
1  <html>
2      <head>
3          <title></title>
4      </head>
5      <body>
6          <?php
7              $object = new User;
8              print_r($object);
9              class User
10             {
11                 public $name, $password;
12                 function save_user()
13                 {
14                     echo "Save User code goes here";
15                 }
16             }
17          ?>
18      </body>
19  </html>

```

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



```

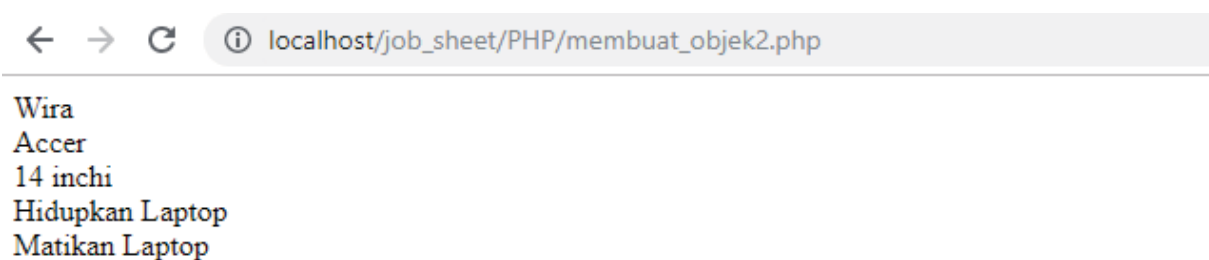
User Object ( [name] => [password] => )
User Object ( [name] => Budiman [password] => pass123 )
Save User code goes here

```

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

```
C: > xampp > htdocs > job_sheet > PHP > 🐞 mengakses_obyek.php
1  <html>
2  <head>
3  |   <title></title>
4  </head>
5  <body>
6  <?php
7  |   $object = new User;
8  |   print_r($object);echo "<br>";
9  |
10 |   $object -> name = "Budiman";
11 |   $object -> password = "pass123";
12 |   print_r($object);echo "<br>";
13 |
14 |   $object->save_user();
15 |
16 |   class User{
17 |       public $name, $password;
18 |       function save_user(){
19 |           echo "Save User code goes here";
20 |       }
21 |   }
22 |
23 |   ?>
24 </body>
```

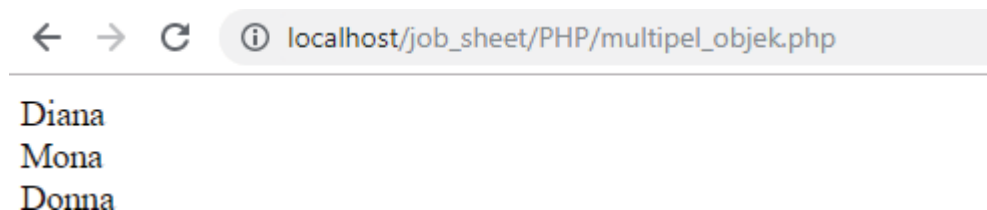
5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 3. Latihan Membuat dan Mengakses Objek 2
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama membuat_objek2.php
4. Ketikkan barisan kode PHP berikut

```
C:\xampp> htdocs > job_sheet > PHP > membuat_objek2.php
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <?php
7 // buat class laptop
8 class laptop {
9 // buat property untuk class laptop
10 var $pemilik;
11 var $merk;
12 var $ukuran_layar;
13 // buat method untuk class laptop
14 function hidupkan_laptop() {
15     return "Hidupkan Laptop";
16 }
17 function matikan_laptop() {
18     return "Matikan Laptop";
19 }
20 }
21 // buat objek dari class laptop (instansiasi)
22 $laptop_anto = new laptop();
23 // set property
24 $laptop_anto->pemilik="wira";
25 $laptop_anto->merk="Accer";
26 $laptop_anto->ukuran_layar="14 inchi";
27 // tampilkan property
28 echo $laptop_anto->pemilik;echo "<br />";
29 echo $laptop_anto->merk;echo "<br />";
30 echo $laptop_anto->ukuran_layar;echo "<br />";
31 // tampilkan method
32 echo $laptop_anto->hidupkan_laptop();echo "<br />";
33 echo $laptop_anto->matikan_laptop();
34 ?>
35 </body>
36 </html>
```

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



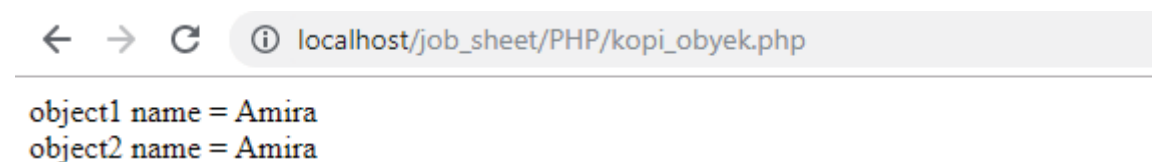
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama multipel_objek.php
4. Ketikkan barisan kode PHP berikut


```

C: > xampp > htdocs > job_sheet > PHP > multipel_objek.php
1  <html>
2  <head>
3  |   <title></title>
4  </head>
5  <body>
6  <?php
7  // buat class laptop
8  class laptop {
9  |   // buat property untuk class laptop
10 |   var $pemilik;
11 |   // buat method untuk class laptop
12 |   function hidupkan_laptop() {
13 |       return "Hidupkan Laptop";
14 |   }
15 | }
16 // buat objek dari class laptop (instansiasi)
17 $laptop_anto = new laptop();
18 $laptop_andi = new laptop();
19 $laptop_dina = new laptop();
20 // set property
21 $laptop_anto->pemilik="Diana";
22 $laptop_andi->pemilik="Mona";
23 $laptop_dina->pemilik="Donna";
24 // tampilkan property
25 echo $laptop_anto->pemilik; // Anto
26 echo "<br />";
27 echo $laptop_andi->pemilik; // Andi
28 echo "<br />";
29 echo $laptop_dina->pemilik; // Dina
30 echo "<br />";
31 ?>
32 </body>
33 </html>

```

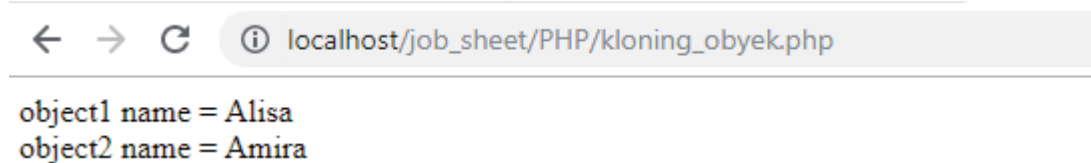
5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 5. Latihan Mengkopi Objek
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



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

```
C: > xampp > htdocs > job_sheet > PHP > ❏ kopi_obyek.php
1  <html>
2      <head>
3          <title></title>
4      </head>
5      <body>
6          <?php
7              $object1 = new User();
8              $object1->name = "Alisa";
9              $object2 = $object1;
10             $object2->name = "Amira";
11             echo "object1 name = " . $object1->name . "<br>";
12             echo "object2 name = " . $object2->name;
13
14             class User{
15                 public $name;
16             }
17         ?>
18     </body>
19 </html>
```

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



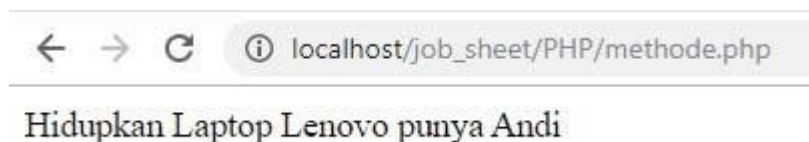
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama kloning_objek.php
4. Ketikkan barisan kode PHP berikut

```

c: > xampp > htdocs > job_sheet > PHP > kloning_obyek.php
1  <html>
2      <head>
3          <title></title>
4      </head>
5      <body>
6          <?php
7              $object1 = new User();
8              $object1->name = "Alisa";
9              $object2 = clone $object1;
10             $object2->name = "Amira";
11             echo "object1 name = " . $object1->name . "<br>";
12             echo "object2 name = " . $object2->name;
13
14             class User{
15                 public $name;
16             }
17         ?>
18     </body>
19 </html>

```

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



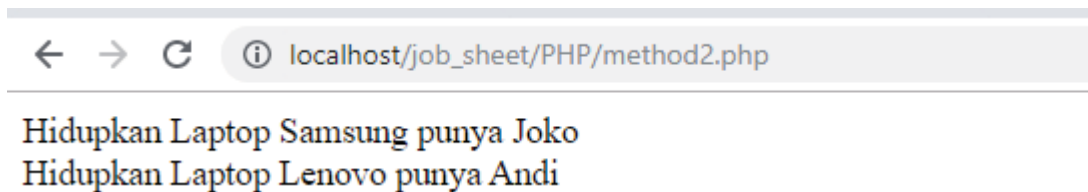
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama metode.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > method.php
1  <?php
2  // buat class laptop
3  class laptop {
4      // buat method untuk class laptop
5      public function hidupkan_laptop($pemilik,$merk) {
6          return "Hidupkan Laptop $merk punya $pemilik";
7      }
8  }
9  // buat objek dari class laptop (instansiasi)
10 $laptop_andi= new laptop();
11 echo $laptop_andi->hidupkan_laptop("Andi", "Lenovo");
12 // hasil: "Hidupkan Laptop Lenovo punya Andi";
13 ?>

```

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



← → ↻ ⓘ localhost/job_sheet/PHP/method2.php

Hidupkan Laptop Samsung punya Joko
Hidupkan Laptop Lenovo punya Andi

2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama metode2.php
4. Ketikkan barisan kode PHP berikut

```

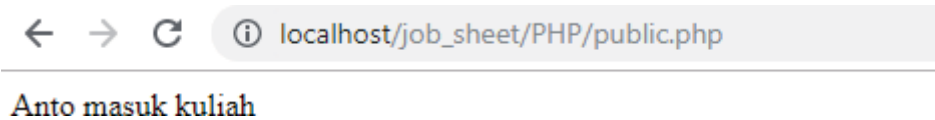
C: > xampp > htdocs > job_sheet > PHP > method2.php
1  <?php
2  // buat class laptop
3  class laptop {
4      // buat method untuk class laptop
5      public function hidupkan_laptop($pemilik="Joko",$merk="Samsung") {
6          return "Hidupkan Laptop $merk punya $pemilik";
7      }
8  }
9  // buat objek dari class laptop (instansiasi)
10 $laptop_andi= new laptop();
11 echo $laptop_andi->hidupkan_laptop();
12 // hasil: "Hidupkan Laptop Samsung punya Joko";
13 echo "<br />";
14 echo $laptop_andi->hidupkan_laptop("Andi", "Lenovo");
15 // hasil: "Hidupkan Laptop Lenovo punya Andi";
16 ?>

```

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

4. 9. Latihan Class Public

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 public.php
4. Ketikkan barisan kode PHP berikut

```

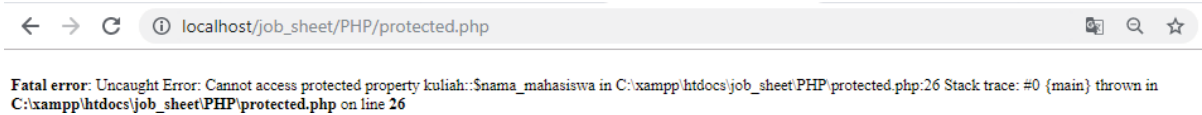
public.php ×
C: > xampp > htdocs > job_sheet > PHP > public.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10 <?php
11 // buat class kuliah
12 class kuliah {
13     // buat public property
14     public $nama_mahasiswa;
15     // buat public method
16     public function masuk_kuliah() {
17         return "masuk kuliah";
18     }
19 }
20 // buat objek dari class kuliah (instansiasi)
21 $kuliah_anto = new kuliah();
22 // set property
23 $kuliah_anto->nama_mahasiswa="Anto";
24 // tampilkan property
25 echo $kuliah_anto->nama_mahasiswa; // Anto
26 // tampilkan method
27 echo $kuliah_anto->masuk_kuliah(); // "masuk kuliah"
28 ?>
29 </body>
30 </html>

```

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

4. 10. Latihan Class Protected

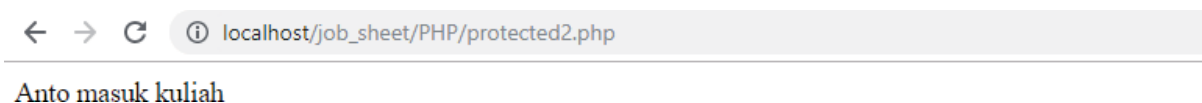
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama protected.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=<?php
6
7 // buat class kuliah
8 class kuliah {
9
10     // buat protected property
11     protected $nama_mahasiswa;
12
13     // buat protected method
14     protected function masuk_kuliah() {
15         return "masuk kuliah";
16     }
17 }
18 // buat objek dari class kuliah (instansiasi)
19 $kuliah_anto = new kuliah();
20 // set protected property akan menghasilkan error
21 $kuliah_anto->nama_mahasiswa="Anto";
22 // Fatal error: Cannot access protected property laptop::$nama_mahasiswa
23 // tampilkan protected property akan menghasilkan error
24 echo $kuliah_anto->nama_mahasiswa;
25 // Fatal error: Cannot access protected property kuliah::$nama_mahasiswa
26 // jalankan protected method akan menghasilkan error
27 echo $kuliah_anto->masuk_kuliah();
28 // Fatal error: Call to protected method laptop::hidupkan_laptop()
29 // from context
30 ?>, initial-scale=1.0">
31     <meta http-equiv="X-UA-Compatible" content="ie=edge">
32     <title>Document</title>
33 </head>
34 <body>
35 </body>
36 </html>
```

5. Buka file di peramban web
 6. Simpan tampilan pe
 7. ramban web dengan menekan Alt+PrtScn
4. 11. Latihan Class Protected 2
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



2. Hidupkan komputer

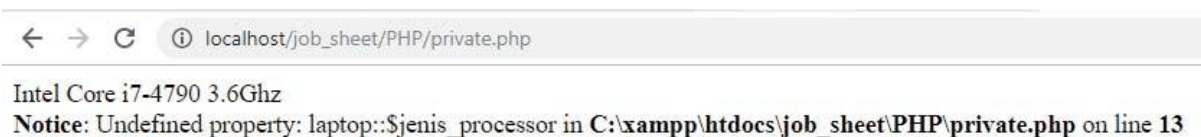
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama protected2.php
4. Ketikkan barisan kode PHP berikut

```
C:\xampp > htdocs > job_sheet > PHP > protected2.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="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 laptop
12 class kuliah {
13     // buat protected property
14     protected $nama_mahasiswa="Anto";
15
16     public function akses_mahasiswa() {
17         return $this->nama_mahasiswa;
18     }
19     protected function masuk_kuliah() {
20         return " masuk kuliah";
21     }
22     public function masuk_kelas() {
23         return $this->masuk_kuliah();
24     }
25 }
26 // buat objek dari class laptop (instansiasi)
27 $kuliah_anto = new kuliah();
28 // jalankan method akses_pemilik()
29 echo $kuliah_anto->akses_mahasiswa(); // "Anto"
30 // jalankan method paksa_hidup()
31 echo $kuliah_anto->masuk_kelas(); // "Hidupkan Laptop"
32 ?>
33 </body>
34 </html>
```

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

4. 12. Latihan Class Private

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



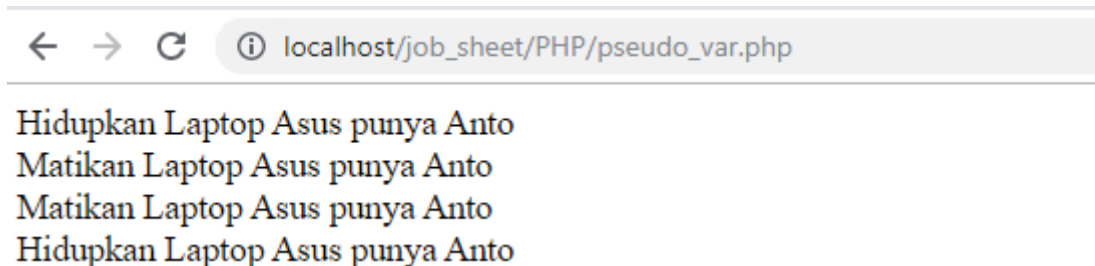
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama protected2.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > private.php
1  <?php
2  // buat class komputer
3  class komputer {
4      // property dengan hak akses protected
5      private $jenis_processor = "Intel Core i7-4790 3.6Ghz";
6      public function tampilkan_processor() {
7          return $this->jenis_processor;
8      }
9  }
10 // buat class laptop
11 class laptop extends komputer{
12     public function tampilkan_processor() {
13         return $this->jenis_processor;
14     }
15 }
16 // buat objek dari class laptop (instansiasi)
17 $komputer_baru = new komputer();
18 $laptop_baru = new laptop();
19 // jalankan method dari class komputer
20 echo $komputer_baru->tampilkan_processor(); // "Intel Core i7-4790 3.6Ghz"
21 // jalankan method dari class laptop (error)
22 echo $laptop_baru->tampilkan_processor();
23 // Notice: Undefined property: laptop::$jenis_processor
24 ?>

```

5. Buka file di peramban web
 6. Simpan tampilan peramban web dengan menekan Alt+PrtScn
4. 13. Latihan Pseudo-Variable \$this
1. Buat kode seperti di bawah sehingga menghasilkan seperti gambar di bawahnya :



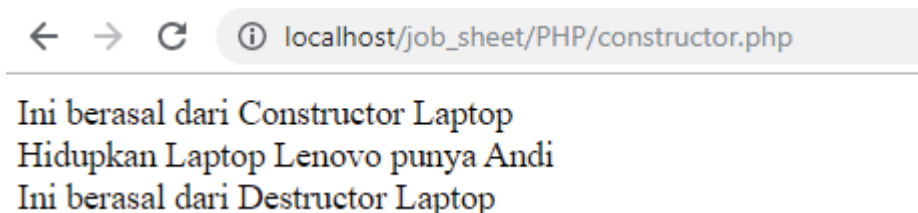
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama pseudo_var.php
4. Ketikkan barisan kode PHP berikut


```

C:\xampp > htdocs > job_sheet > PHP > pseudo_var.php
1  <?php
2  // buat class laptop
3  class laptop {
4      // buat property untuk class laptop
5      public $pemilik;
6      public $merk;
7      // buat method untuk class laptop
8      public function hidupkan_laptop() {
9          return "Hidupkan Laptop $this->merk punya $this->pemilik";
10     }
11     public function matikan_laptop() {
12         return "Matikan Laptop $this->merk punya $this->pemilik";
13     }
14     public function restart_laptop() {
15         $matikan=$this->matikan_laptop();
16         $hidupkan= $this->hidupkan_laptop();
17         $restart=$matikan."<br />".$hidupkan;
18         return $restart;
19     }
20 }
21 // buat objek dari class laptop (instansiasi)
22 $laptop_anto = new laptop();
23 // isi property objek
24 $laptop_anto->pemilik="Anto";
25 $laptop_anto->merk="Asus";
26 echo $laptop_anto->hidupkan_laptop();
27 // hasil: "Hidupkan Laptop Asus punya Anto";
28 echo "<br />";
29 echo $laptop_anto->matikan_laptop();
30 // hasil: "Matikan Laptop Asus punya Anto";
31 echo "<br />";
32 echo $laptop_anto->restart_laptop();
33 // hasil:
34 // "Matikan Laptop Asus punya Anto";
35 // "Hidupkan Laptop Asus punya Anto";
36 ?>

```

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



← → ↻ ⓘ localhost/job_sheet/PHP/constructor.php

Ini berasal dari Constructor Laptop
Hidupkan Laptop Lenovo punya Andi
Ini berasal dari Destructor Laptop

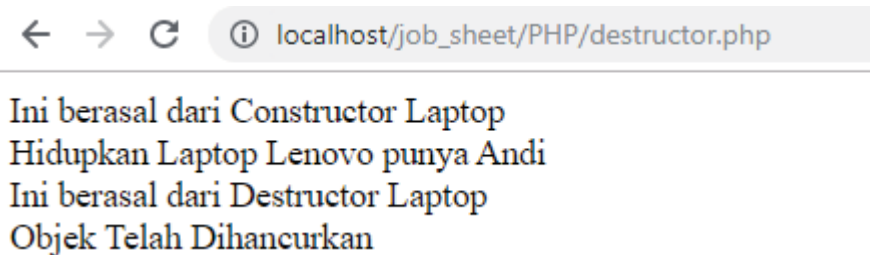
2. Hidupkan komputer
3. Buat folder di C:\xampp\htdocs\, lalu buatlah file dengan nama constructor.php
4. Ketikkan barisan kode PHP berikut

```

C: > xampp > htdocs > job_sheet > PHP > constructor.php
1  <?php
2  // buat class laptop
3  class laptop {
4      private $pemilik = "Andi";
5      private $merk = "Lenovo";
6      public function __construct(){
7          echo "Ini berasal dari Constructor Laptop";
8      }
9      public function hidupkan_laptop(){
10         return "Hidupkan Laptop $this->merk punya $this->pemilik";
11     }
12     public function __destruct(){
13         echo "Ini berasal dari Destructor Laptop";
14     }
15 }
16 // buat objek dari class laptop (instansiasi)
17 $laptop_andi= new laptop();
18 echo "<br />";
19 echo $laptop_andi->hidupkan_laptop();
20 echo "<br />";
21 ?>

```

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



← → ↻ ⓘ localhost/job_sheet/PHP/destructor.php

Ini berasal dari Constructor Laptop
Hidupkan Laptop Lenovo punya Andi
Ini berasal dari Destructor Laptop
Objek Telah Dihancurkan

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

```

C: > xampp >htdocs > job_sheet > PHP > unset.php
1  <?php
2  // buat class laptop
3  class laptop {
4      private $pemilik = "Andi";
5      private $merk = "Lenovo";
6      public function __construct(){
7          echo "Ini berasal dari Constructor Laptop";
8      }
9      public function hidupkan_laptop(){
10         return "Hidupkan Laptop $this->merk punya $this->pemilik";
11     }
12     public function __destruct(){
13         echo "Ini berasal dari Destructor Laptop";
14     }
15 }
16 // buat objek dari class laptop (instansiasi)
17 $laptop_andi= new laptop();
18 echo "<br />";
19 echo $laptop_andi->hidupkan_laptop();
20 echo "<br />";
21 // hapus objek $laptop_andi
22 unset($laptop_andi);
23 echo "<br />";
24 echo "Objek Telah Dihancurkan";
25 ?>

```

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	Nama: Novita Alya Ramadhani Umur: 20 Tahun Tempat, Tanggal Lahir: Kab. Semarang, 14 November 2000 Jenis Kelamin: Perempuan Jurusan: Teknik Elektro Prodi: D3-Teknik Informatika	

VI. Latihan Soal

1. Buatlah dengan konsep OOP sebuah Class yang menurunkan minimal 5 objek dengan menggunakan metode public, protected dan private.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JOB 5</title>
  </head>
  <body>
    <?php
    class DataMhs {
      public $namamhs;
      public $usia;
      private $ttl = "Kab. Semarang, 14 November 2000<br />";
      private $jk = "Perempuan<br />";
      protected $jursn = "Teknik Elektro<br />";
      protected $prodi = "D3-Teknik Informatika";

      //method
      function tampil_nama(){
        return "Nama</t/t/t>: Novita Alya Ramadhani <br />";
      }
      function tampil_usia(){
        return "Umur</t/t/t>: 20 Tahun <br />";
      }
      function tampil_ttl(){
        return "Tempat, Tanggal Lahir      : ".$this->ttl;
      }
      function tampil_jk(){
        return "Jenis Kelamin              : ".$this->jk;
      }
      public function tampil_jursn(){
        return "Jurusan                    : ".$this->jursn;
      }
    }
  </body>
</html>

```

```

      public function tampil_jursn(){
        return "Jurusan                    : ".$this->jursn;
      }
      public function tampil_prodi(){
        return "Prodi                      : ".$this->prodi;
      }
    }

    // instansiasi class
    $DataKu = new DataMhs();

    //Panggil method
    echo $DataKu->tampil_nama();
    echo $DataKu->tampil_usia();
    echo $DataKu->tampil_ttl();
    echo $DataKu->tampil_jk();
    echo $DataKu->tampil_jursn();
    echo $DataKu->tampil_prodi();
    ?>
  </body>
</html>

```

VII. Kesimpulan

Melalui job kali ini, dapat dimengerti Object Oriented Programming terdiri dari class, object, property dan method. Class ialah *blueprint* dari object. Di sini class hanya membuat kerangka dasar. Property ialah data, sedangkan method adalah tindakan yang dapat dilakukan dalam class, yang sebenarnya function dalam *class*.