

## Lanjutan Web Programming

### Materi Pembahasan Lanjutan Web Programming

(Hari Ke-4)

#### A. PENGERTIAN DAN PENGENALAN OOP

OOP adalah singkatan dari Object Oriented Programming. OOP merupakan metode pemrograman yang lebih berorientasi pada objek. Maksudnya pemrograman yang lebih terpusat pada objek, sehingga akan lebih sangat memudahkan kita di dalam membuat aplikasi. sebenarnya OOP lebih di dukung pada pemrograman JAVA dan C++. Akan tetapi di PHP sudah sangat di dukung pada versi PHP5. Kenapa OOP sangat bagus jika di terapkan pada pembuatan aplikasi ? Berikut ini merupakan uraian kelebihan-kelebihan penggunaan OOP.

##### ➤ Kelebihan OOP Pada PHP

Berikut ini adalah kelebihan menggunakan OOP dalam membuat aplikasi :

1. Syntax lebih terstruktur.
2. Terekomendasi.
3. Sangat efektif jika di gunakan untuk membuat aplikasi yang berskala besar.
4. Lebih mudah dan menghemat waktu, karena fungsi/function bisa di panggil berulang-ulang kali tergantung keperluan.
5. Lebih menghemat waktu.
6. Aplikasi lebih mudah di kembangkan.
7. Dan lebih banyak lagi kelebihan dari OOP yang akan dirasakan sendiri pada saat mulai menggunakannya.

Salah satu framework PHP yang termasuk paling banyak di gunakan adalah Framework Codeigniter. framework codeigniter juga menggunakan metode OOP. Jadi jika ingin belajar dan membuat aplikasi dengan framework codeigniter, maka harus memahami dulu konsep OOP PHP.

➤ Memahami Konsep OOP PHP Contoh PHP.

Agar lebih mudah memahami konsep OOP PHP, akan saya contohkan pada kehidupan sehari-hari. Saya memiliki sebuah robot yang bisa dikendalikan dengan remot. Jika di pemrograman dengan metode struktural maka dibaratnya memerintahkan robot tersebut dengan sekali-sekali perintah. Misalnya pertama memerintahkan untuk mengangkat tangan. lalu melangkah. dan jika perintah nya ingin di ulangi maka harus mengulangi perintah tersebut lagi. Tetapi jika dengan pemrograman OOP anda bisa memerintahkan nya dengan berulang kali tanpa menekan tombol berulang kali.

➤ Apa saja yang di pelajari untuk pada konsep OOP

Adapun yang akan kita pelajari dan pahami untuk menggunakan metode OOP adalah sebagai berikut :

1. Pengenalan struktur OOP PHP Dasar.
2. Class, Property, Method, Object Pada OOP PHP.
3. Function Constructor dan Destructor di OOP PHP.
4. Inheritance dan Encapsulation OOP PHP. dan lainnya.

## **B. PENGERTIAN CLASS, OBJECT, PROPERTY DAN METHOD**

Class, object, property dan method adalah pondasi dasar dari membangun aplikasi menggunakan struktur OOP. Jika di ibaratkan membangun sebuah rumah, maka class, object, property dan method adalah pilar-pilar dan bahan penyokong nya. selain penjelasannya, akan di sertakan juga contoh dan cara penulisan nya.

Jika kita ibaratkan pada kehidupan nyata dari pengertian class, object, property dan method. Ibaratnya class nya mobil maka property nya adalah warna warna mobil, bentuk mobil, merek mobil dan lain-lain. dan method atau aksi yang bisa di lakukan oleh mobil adalah maju, berhenti, belok dan lain-lain.

❖ Pengertian Class Dalam Pemrograman Berorientasi Objek

Class di dalam OOP di gunakan untuk membuat sebuah kerangka kerja, bisa di katakan sebagai library. Class berisi property dan method. Jadi ibaratnya class adalah sebuah wadah yang menyimpan property dan method. dan object yang di

hasilkan biasanya berdasarkan isi dari class. Jika kita ibaratkan lagi di dalam OOP PHP, class di tulis dengan awalan syntax class. dan kemudian baru di ikuti dengan nama class nya.

Contoh penulisan class seperti berikut :

```
<?php

//Cara penulisan class OOP PHP - www.yourproject.com
class nama_class{

    //isi dari class ini

}

?>
```

#### ❖ Pengertian dan cara penulisan Property di dalam OOP PHP

Property adalah data-data yang terdapat di dalam class. data nya biasanya berupa sifat. contoh nya seperti yang saya jelaskan pengertian class, object, property dan method di atas. yaitu penjelasan sifat dari class. contoh property pada class mobil adalah warna, ukuran, merek dan lain-lain. contoh property pada class manusia adalah warna kulit, warna rambut, nama, dan lain-lain.

Kegunaan property pada sebuah class sama dengan kegunaan variabel di php. bisa di gunakan untuk meyimpan data dan lain-lain. cara penulisan property pada class adalah dengan di awali syntax var.cara penamaan property sama dengan aturan penamaan variabel.

Berikut ini adalah contoh penulisan property di dalam class :

```
<?php

//Cara penulisan class dan property OOP PHP - www.yourproject.com
class mobil{

    var $warna;
    var $merek;
    var $ukuran;

}

?>
```

❖ Pengertian Method dalam pemrograman OOP PHP

Method adalah sebuah aksi yang terdapat di dalam class. seperti yang sudah di jelaskan. misalnya method pada class mobil adalah contoh nya maju, berhenti, belok dan lain-lain.

Penulisan method pada class oop adalah dengan menuliskan syntax function di awalnya. lalu di ikuti dengan nama method tersebut. kita dapat memberikan nilai pada parameter method oop dan dapat juga mengembalikan nilai nya.

Berikut ini adalah contoh penulisan method pada oop php :

```
<?php

//Cara penulisan class dan property OOP PHP - www.yourproject.com
class mobil{
    // property oop
    var $warna;
    var $merek;
    var $ukuran;

    //method oop
    function maju(){
        //isi method
    }

    function berhenti(){
        //isi mehod
    }
}

?>
```

❖ Pengertian dan Contoh object pada oop php

Object adalah output dari class. dan object dapat menampilkan atau mengelola isi class. seluruh isi class akan kita instansiasikan menjadi object. Contoh penulisan object pada oop sebagai berikut :

```
<?php

//Cara penulisan class dan property OOP PHP - www.yourproject.com
class mobil{

    //isi class
```

```

}

$mobil = new mobil();

?>

```

perhatikan pada contoh penulisan object pada oop di atas. penulisan object diawali dengan syntax new. dan kemudian diikuti dengan nama class yang ingin diinstansiasi.

#### ❖ Contoh Pemrograman Berorientasi Object pada PHP(OOP)

Berikut adalah contoh syntax penulisan metode oop di php. buat sebuah file dengan nama oop.php atau terserah teman-teman. Kemudian ketik syntax berikut :

```

<?php
//PHP OOP Part 2  Pengertian Class, Object, Property dan Method
//class manusia
class manusia{
    //property
    var $nama;
    var $warna;

    //method manusia
    function tampilkan_nama(){
        return "Nama saya ishamashi <br/>";
    }

    function warna_kulit(){
        return "Warna kulit saya hitam <br/>";
    }
}
//instansiasi classmanusia
$manusia = new manusia();

//memanggil method tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();

//memanggil method warna_kulit dari class manusia
echo $manusia->warna_kulit();

```

pada contoh ini kita membuat sebuah class dengan nama manusia.

```

class manusia{

```

dan kemudian contoh penulisan property nama dan warna

```
//property
var $nama;
var $warna
```

dan membuat dua buah method oop php untuk contoh

```
//method manusia
function tampilkan_nama(){
    return "Nama saya ishamashi <br/>";
}

function warna_kulit(){
    return "Warna kulit saya hitam <br/>";
}
```

di sini kita membuat method atau fungsi tampilkan\_nama yang berisi perintah untuk mengembalikan string nama saya ishamashi dan contoh objek nya. Kita menginstansiasikan dulu class manusia nya. dan memanggil method tampilkan nama dan method warna kulit. pemanggilan method dari object menggunakan tanda “->”

```
//instansiasi class manusia
$manusia = new manusia();

//memanggil method tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();

//memanggil method warna_kulit dari class manusia
echo $manusia->warna_kulit();
```

dan hasilnya :



```
← → ↻ localhost/oop.php
Nama saya ishamasi
Warna kulit saya hitam
```

### C. PENGENALAN CONSTRUCT DAN DESTRUCT PADA OOP PHP

- Pengertian Method Construct(konstruktor)

Method Construct adalah method yang di jalankan pertama kali pada saat sebuah class di jalankan. Jadi apabila kita menjalanka atau menginstansiasi sebuah

class, maka jika terdapat method atau function construct di dalam nya maka yang pertama kali di jalankan adalah method construct ini.

Cara penulisan method construct diawali dengan syntax function. lalu di ikuti dengan \_\_construct.(double underscore dan construct).

Contoh penulisannya :

```
<?php

class nama_class{

    function __construct(){
        //isi fungsi construct
    }

}
```

perlu di ketahui, kita tidak perlu lagi memanggil fungsi construct pada saat instansiasi. karena construct dan destruct akan di jalankan otomatis jika class nya yang di instansiasi. Perhatikan contoh penggunaan method atau fungsi \_\_construct() pada contoh oop php berikut :

```
<?php

//class manusia
class manusia{
    //property
    var $nama;
    var $warna;

    function __construct(){
        echo "ini adalah isi method construct <br/>";
    }

    //method manusia
    function tampilkan_nama(){
        return "Nama saya ishamashi <br/>";
    }

}

//instansiasi class manusia
$manusia = new manusia();

//memanggil method tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();
```

dapat dilihat pada contoh di atas. method construct akan di panggil pertama kali. Padahal kita tidak memanggilnya. karena method construct di jalankan pertama kali sebelum method yang lainnya secara otomatis.

- Pengertian Method Destruct (destruktor)

Jika construct adalah method yang pertama kali di jalankan pada saat sebuah class di akses, maka destruct adalah method yang terakhir di jalankan setelah semua method di classnya selesai di jalankan. jadi method destruct adalah kebalikan dari method construct.

Cara penulisannya sama seperti penulisan method construct. diawali dengan syntax function. dan di awalnya menggunakan underscore ganda(\_\_). contohnya \_\_destruct()

```
<?php
class nama_class{
    function __destruct(){
        //isi method destruct
    }
}
```

dan perhatikan contoh penulisan dan penggunaan method destruct pada oop php. Disini saya juga menyertakan method construct agar bisa teman-teman bedakan perbedaan dari construct dan destrcut. sesuai dengan yang di jelaskan di atas. method construct adalah method yang di jalankan pertama kali. dan method destruct adalah method yang terakhir di jalankan.

Contohnya seperti ini :

```
<?php
//www.yourproject.com
//class manusia
class manusia{
    //property
    var $nama;
    var $warna;

    //method construct di jalankan pertama kali
    function __construct(){
        echo "ini adalah isi method construct <br/>";
    }
}
```



```

//method destruct di jalankan terakhir
function __destruct(){
    echo "ini adalah isi method destruct <br/>";
}

//method manusia
function tampilkan_nama(){
    return "Nama saya ishamashi <br/>";
}
}
//instansiasi class manusia
$manusia = new manusia();

//memanggil method tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();

```

dapat di lihat pada contoh di atas. method destruct di jalankan yang terakhir.PHP

#### D. PENGERTIAN ENKAPSULASI (PUBLIC, PRIVATE, PROTECTED)

Enkapsulasi lah yang menentukan pemberian hak akses pada setiap property atau method. enkapsulasi atau encapsulation pada OOP terdiri dari 3. yaitu : Public, Private dan Protected.

Tentu ketiganya memiliki kegunaan nya masing-masing. adapun kegunaan public, private dan protected adalah :

- Public : Dengan menggunakan public pada property dan method, berarti kita memberikan akses pada property dan method agar bisa di akses dari luar atau dalam class.
- Private
- Protected

kita bisa menentukan property dan method menjadi public, protected dan private. sesuai dengan keinginan dan kebutuhan.

#### Pengertian Enkapsulasi (Encapsulation)

Enkapsulasi (encapsulation) adalah metode untuk mengatur struktur pada class dengan tujuan untuk menentukan hak akses pada property atau method. Ini

bertujuan untuk menyembunyikan informasi dari method dan property dengan alasan keamanan.

### Enkapsulasi Public, Private dan Protected

Untuk dapat memberikan hak akses pada method dan property pada sebuah class, di OOP terdapat 3 teknik yaitu, public, private dan protected. dan akan kita jelaskan satu persatu. di mulai dengan hak akses public.

### Pengertian Hak Akses Public

Jika sebuah method dan property menggunakan public, maka method dan property tersebut dapat di akses dari luar atau dari dalam class tersebut. perlu di ketahui, pada method dan property yang tidak di tetapkan hak aksesnya menggunakan public, private dan protected, maka hak akses pada method atau property tersebut adalah secara default menjadi public. seperti penulisan oop yang kita pelajari sebelumnya.

Contoh penggunaan hak akses public.

```
<?php

//class manusia
class manusia{
    //prop
    public $nama;
    public $warna;

    //method manusia
    function tampilkan_nama(){
        return "Nama saya ishamashi <br/>";
    }
}

//instansiasi class manusia
$manusia = new manusia();

//memanggil method tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();
```

nah, coba perhatikan pada syntax di atas. terdapat property dan method yang kita tetapkan dengan public. Karena method dan property nya kita tetapkan dengan public, maka bisa kita akses dari luar class. ataupun class turunan seperti yang

terdapat pada contoh public di atas. kita dapat memanggil method dan property public dari luar class.

#### Pengertian Hak Akses Private

Hak akses kedua yang akan kita bahas adalah private. private adalah hak akses yang melarang method atau property yang menggunakan nya di larang di akses dari luar class. jadi hak akses private hanya bisa di akses dari dalam class itu sendiri.

```
<?php

//class manusia
class manusia{
    //menentukan property dengan private
    private $nama = "ishamashi";

    //method public
    public function tampilkan_nama(){
        return "Nama saya " . $this->nama;
    }
}

//instansiasi class manusia
$manusia = new manusia();

//memanggil method public tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();
```

#### Pengertian Hak Akses Protected

Setelah memahami penjelasan tentang public dan private, selanjutnya kita akan membahas tentang hak akses protected pada oop php. jika property dan method di tetapkan dengan protected, berarti property dan method tersebut tidak bisa di akses dari luar class. tetapi bisa di akses dari dalam class itu sendiri.

Contohnya :

```
<?php

//class manusia
class manusia{
    //menentukan property dengan protected
    protected $nama = "ishamashi";
```

```

//method protected
protected function nama(){
    return "Nama saya " . $this->nama;
}

public function tampilkan_nama(){
    return $this->tampilkan_nama;
}
}

//instansiasi class manusia
$manusia = new manusia();

//memanggil method public tampilkan_nama dari class manusia
echo $manusia->tampilkan_nama();

```

## E. PEWARISAN SIFAT (INHERITANCE) PADA OOP PHP

Inheritance atau pewarisan sifat merupakan class turunan dari class yang lainnya. inheritance di buat dengan membuat class baru dengan tujuan di hubungkan dengan class yang lain yang di inginkan. dengan tujuan untuk mendapatkan informasi atau data dari class yang di hubungkan tersebut. class yang baru ini di sebut juga dengan sub class atau child class. jadi subclass dan childclass adalah class turunan. dan class utama atau class yang di hubungkan di sebut juga parent class.

### ✓ Pengertian Inheritance atau pewarisan sifat

Contoh sederhana nya misalnya kita memiliki 2 buah class. kita beri nama dengan class a dan class b. misalnya pada class a ini terdapat property atau method. dan dengan inheritance atau class turunan, kita dapat mengakses method atau property dari class a langsung dari class b.

### ✓ Cara Membuat Class Turunan (Inheritance)

Untuk membuat class turunan, kita bisa menggunakan syntax “extends” pada class. perhatikan contoh berikut ini untuk membuat pewarisan sifat pada class(inheritance). Langsung saja untuk membuat class turuan atau pewarisan sifat(inheritance) buat langsung sebuah file php dengan nama inheritance.php. Lalu, diletakkan di localhost.

## inheritance.php

```

<?php
//www.yourproject.com
// class parent
class manusia{

    // property class manusia
    public $nama_saya;

    // method pada class manusia
    function berinama($saya){
        $this->nama_saya=$saya;
    }

}

// class turunan atau sub class dari class manusia
// kita menghubungkan class dengan syntax extends
class teman extends manusia{

    // property class teman
    public $nama_teman;

    // method pada class teman
    function berinamateman($teman){
        $this->nama_teman=$teman;
    }

}

// instansiasi class teman
$ishamashi = new teman;

// method beri nama adalah method pada class manusia, tapi kita bisa
mengaksesnya karena telah menghubungkan class teman dengan class ma
nusia
$ishamashi ->berinama(" ISHAMASHI ");
$ishamashi ->berinamateman(" Diki ");

// menampilkan isi property
echo "Nama Saya : " . $ishamashi->nama_saya . "<br/>";
echo "Nama Teman Saya : " . $ishamashi->nama_teman;

?>

```

perhatikan pada contoh di atas. sebagai contoh saya membuat 2 buah class. yaitu class manusia dan class teman. di class manusia saya membuat sebuah method dan sebuah property. yang dapat teman-teman lihat sendiri pada contoh syntax di atas.

Kemudian saya membuat pewarisan sifat pada class teman, dengan cara menambahkan syntax extends lalu mengisi nama class mana yang ingin di hubungkan. di sini saya menghubungkan class teman dengan class manusia.

```
class teman extends manusia{

// property class teman
public $nama_teman;

// method pada class teman
function berinamateman($teman){
    $this->nama_teman=$teman;
}
}
```

Sampai di sini kita telah memiliki class yang telah di hubungkan dan memiliki pewarisan sifat. selanjutnya tinggal di instansiasi.

```
// instansiasi class teman
$ishamashi = new teman;

// method beri nama adalah method pada class manusia, tapi kita bisa
mengaksesnya karena telah menghubungkan class teman dengan class ma
nusia
$ishamashi->berinama(" ISHAMASHI ");
$ishamashi->berinamateman(" Diki ");
```

Perhatikan pada syntax di atas. kita hanya menginstansiasi kan class teman. tetapi kita bisa mengakses method “berinama()”. Padahal method berinama adalah method milik class manusia. Kita bisa mengaksesnya karena sudah terhubung. jadi sifat pada class manusia sudah turun ke class teman juga.