

OOP in PHP, CMS and Java

- PHP also support OO (object oriented) eventhough a simple OO
- Recall concept of OO (class, inheritance...). Think about object. Ex: person, house,... anything
- OO adalah konsep programming, berbeda dgn konsep prosedural
- Pemahaman dasar tentang OOP plg penting, sebelum memulai

- OO bisa diartikan segala sesuatu yang bersifat modular. Modularisasi
- Object → gambaran di dunia nyata, ex: baju gaun
- Class → representasi object dalam program, ex: desain baju gaun
- Recall array (index and assosiatif), the complex data type as the prominent about class, all basic dari php

Class

- Class is a storage to store data and information of object, property or attributes of object as like as a variable
- In class not only for property but also a function. Function in a class show the process that used to in object
- *class classname{
//declare of property or/and function
}*

Ex:

```
class ClassCourse {  
    //think about course in IE dept as object  
    var $dept = "Informatics Engineering";  
    var $code;  
    var $name;  
}
```

- Class Course in IE dept
- Has 3 variables-properties (dept, code and name), general attributes in object course

- Or

```
class ClassCourse1{  
    //a general course not specific dept  
    var $dept;  
    var $code;  
    var $name;  
}
```

- U can think class as "a factory knife" can cut uniformly or cut with various style and keep the core properties
- Sederhananya, class memproduksi banyak object

```
//Web_Programming is variable which has object as data type
$Web_Programming = new ClassCourse();
//means Web_Programming will has a property from ClassCourse
//dept: Informatics Engineering var code and name
```

```
$International_Bussiness = new ClassCourse1();
//International_Bussiness is var which has ClassCourse1 as data type
```

- Or define value of var

•

```
//Web_Programming is variable which has object as data type
$Web_Programming = new ClassCourse();
//means Web_Programming will has a property from ClassCourse
//dept: Informatics Engineering var code and name
$Web_Programming->code = "TI01777";
$Web_Programming->name = "Web Programming";
echo $Web_Programming->dept."<br>";//value as the same as ClassCourse
echo $Web_Programming->code."<br>";
echo $Web_Programming->name."<br>";
```



Informatics Engineering
TI01777
Web Programming

- What about this one

```
//Web_Programming is variable which has object as data type
$Web_Programming = new ClassCourse();
//means Web_Programming will has a property from ClassCourse
//dept: Informatics Engineering var code and name
$Web_Programming->code = "TI01777";
$Web_Programming->name = "Web Programming";
$Web_Programming->dept = "Computer Science";
echo $Web_Programming->dept."<br>";//will be filled by new value
echo $Web_Programming->code."<br>";
echo $Web_Programming->name."<br>";
```



Computer Science
TI01777
Web Programming

- Class not only storage of properties but also functions/methods (process its own data)
- Ex: from prev class, modify

```
class ClassCourse1{  
    //a general course not specific dept  
    var $dept;  
    var $code;  
    var $name;  
  
    function showCourse() { //easy ex  
        echo "Department = ".$this->dept."<br>";  
        echo "Course_code = ".$this->code."<br>";  
        echo "Course_name = ".$this->name."<br>";  
    }  
}
```

- Use it as prev ex

```
$International_Bussiness = new ClassCourse1();  
//International_Bussiness is var which has ClassCourse1 as data type  
$International_Bussiness->dept = "Economics";  
$International_Bussiness->code = "EM23009";  
$International_Bussiness->name = "International Bussiness";  
$International_Bussiness->showCourse();
```

Department = Economics
Course_code = EM23009
Course_name = International Bussiness

Another ex

```
//Class_Student_lib.php
class Student{
    var $dept; var $nim; var $name; var $course;
    var $kkd1; var $kkd2; var $kkd3; var $kkd4;

    function marknumber(){
        $average = ($this->kkd1+$this->kkd2+$this->kkd3+$this->kkd4)/4;
        return $average;
    }
    function markletter(){
        if ($this->marknumber()>="85"){
            $letter = "A";
        }
        if ($this->marknumber()<"85" and $this->marknumber()>="70"){
            $letter = "B";
        }
        if ($this->marknumber()<"70" and $this->marknumber()>="69"){
            $letter = "C";
        }
        if ($this->marknumber()<"69" and $this->marknumber()>="60"){
            $letter = "D";
        }
        if ($this->marknumber()<"60"){
            $letter = "E";
        }
        return $letter;
    }

    function show(){
        echo "Department: ".$this->dept."<br>";
        echo "NIM: ".$this->nim."<br>";
        echo "Name: ".$this->name."<br>";
        echo "Course: ".$this->course."<br>";
        echo "Marknumber: ".$this->marknumber()."<br>";
        echo "Markletter: ".$this->markletter();
    }
}
```

- Dari contoh sebelumnya class yang telah dibuat dapat dikatakan sebagai sebuah modul.
- Lets name it Class_Student_lib.php
- Dan gunakan dengan modularisasi, include

```
//Course.php  
include 'Class_Student_lib.php';//include modul
```

- Use the class

```
//Course.php
include 'Class_Student_lib.php';//include modul

$if0001 = new Student(); //membuat object baru
$if0001->dept = "Informatika";
$if0001->nim = "IF001";
$if0001->name = "Raden Mas";
$if0001->course = "Web Programming";
$if0001->kkd1 = "88";
$if0001->kkd2 = "80";
$if0001->kkd3 = "78";
$if0001->kkd4 = "86";
$if0001->show();
```

```
Department: Informatika
NIM: IF001
Name: Raden Mas
Course: Web Programming
Marknumber: 83
Markletter: B
```

Constructor

- Simply, sebuah class yang telah terinisiasi nilai tertentu in property or method
- Secara otomatis, property or methods will be retrieved when the class is used

```
//Class_Student_lib.php  
class Student{  
    var $dept = "Informatika";  
}
```

Encapsulation

- Simply, make boundary of the class, membuat aturan batasan akses
- Modifiers: Public, Private, Protected
- Default: public
- "var" → public
- Public: have no access restrictions, meaning anyone can access them. Tidak ada pembatasan, if U need to call property from class, then property must be public

- Private: only the same class can access the property. Hanya dalam class tersebut
- Protected: only the same class and classes derived from that class can access the property. Hanya dalam class tersebut dan class turunanny
- Gunanya: pembatasan pengaksesan class

- Modify Class_Student_lib.php

```
// -----  
class Student{  
    var $dept; var $nim; var $name; var $course;  
    var $kkd1; var $kkd2; var $kkd3; var $kkd4; //all public  
    private $univ; // private
```

```
private function marknumber(){  
    $average = ($this->kkd1+$this->kkd2+$this->kkd3+$this->kkd4)/4;  
    return $average;  
}  
private function markletter(){  
    if ($this->marknumber()>="85"){  
        $letter = "A";  
    }  
    if ($this->marknumber()<"85" and $this->marknumber()>="70"){  
        $letter = "B";  
    }  
    if ($this->marknumber()<"70" and $this->marknumber()>="69"){  
        $letter = "C";  
    }  
    if ($this->marknumber()<"69" and $this->marknumber()>="60"){  
        $letter = "D";  
    }  
    if ($this->marknumber()<"60"){  
        $letter = "E";  
    }  
    return $letter;  
}
```

```
function show1(){
    echo "Universitas: ".$this->univ."<br>";
    echo "Department: ".$this->dept."<br>";
    echo "NIM: ".$this->nim."<br>";
}
private function show2(){ //private methods
    echo "Department: ".$this->dept."<br>";
    echo "NIM: ".$this->nim."<br>";
    echo "Name: ".$this->name."<br>";
    echo "Course: ".$this->course."<br>";
    echo "Marknumber: ".$this->marknumber()."<br>";
    echo "Markletter: ".$this->markletter();
}
```

Universitas:
Department: Informatika
NIM: IF001

Inheritance

- One of important concept in OO, modify prev ex

```
class Stud{ //reduce properties from Student
    var $dept;
    var $nim;
    var $name;
}
```

- Make inheritance, pewarisan
- **class classname2 extends classname{**
//add properties or functions
}

```

class DataStud extends Stud{ //new class DataStud inheritance from Stud
    //means DataStud has all properties or functions from Stud
    //now, add properties or functions only belongs to DataStud
    var $gender;
    var $address;
    var $birthday;
    var $email;
    var $mobilephone;
    function age(){//simple ex calculate age
        list($y,$m,$d)= explode("-", $this->birthday);
        $yearage = $y;
        $today = date(Y);
        $ag = $today - $yearage;
        return $ag;
    }
    function show(){
        echo "NIM = ".$this->nim."<br>";
        echo "Name = ".$this->name."<br>";
        echo "Age = ".$this->age()."<br>";
        echo "Gender = ".$this->gender."<br>";
        echo "Address = ".$this->address."<br>";
        echo "Email = ".$this->email."<br>";
        echo "Mobilephone = ".$this->mobilephone."<br>";
    }
}

```

- Then use it

```
$DataStudent = new DataStud();  
$DataStudent->nim = "EM1111";  
$DataStudent->dept = "Economics";  
$DataStudent->name = "Suryo";  
$DataStudent->birthday = "1988-08-08";  
$DataStudent->gender = "Male";  
$DataStudent->address = "Surakarta";  
$DataStudent->email = "suryo@suryo.net";  
$DataStudent->mobilephone = "081111111111";  
$DataStudent->show();
```

```
NIM = EM1111  
Department = Economics  
Name = Suryo  
Age = 22  
Gender = Male  
Address = Surakarta  
Email = suryo@suryo.net  
Mobilephone = 081111111111
```

Another one

```
class MarkStud extends Stud{//Stud has dept nim name
//add only belongs to MarkStud
var $course;
var $kkd1;var $kkd2; var $kkd3;var $kkd4;
//use prev functions
function marknumber(){
    $average = ($this->kkd1+$this->kkd2+$this->kkd3+$this->kkd4)/4;
    return $average;
}
function markletter(){
    if ($this->marknumber()>="85") {
        $letter = "A". "<br>";
    }
    if ($this->marknumber()>="70" and $this->marknumber()<"84") {
        $letter = "B". "<br>";
    }
    if ($this->marknumber()>="60" and $this->marknumber()<"69") {
        $letter = "C". "<br>";
    }
    if ($this->marknumber()>="50" and $this->marknumber()<"59") {
        $letter = "D". "<br>";
    }
    if ($this->marknumber()<="49") {
        $letter = "E". "<br>";
    }
    return $letter;
}
function show(){
    echo "Department = ".$this->dept."<br>";
    echo "NIM = ".$this->nim."<br>";
    echo "Name = ".$this->name."<br>";
    echo "Course = ".$this->course."<br>";
    echo "Mark_number = ".$this->marknumber(). "<br>";
    echo "Mark_letter = ".$this->markletter(). "<br>";
}
```

```
$Mark = new MarkStud();  
$Mark->dept = "Economics";  
$Mark->nim = "EM1111";  
$Mark->name = "Suryo";  
$Mark->course = "Management";  
$Mark->kkd1 = "80";  
$Mark->kkd2 = "68";  
$Mark->kkd3 = "72";  
$Mark->kkd4 = "70";  
$Mark->show();
```

Department = Economics
NIM = EM1111
Name = Suryo
Course = Management
Mark_number = 72.5
Mark_letter = B

- U can also do like this

```
//inheritance from prev inheritance
class MarkwithLabWork extends MarkStud{
    //MarkwithLabWork --> MarkStud --> Stud
    //a kind of level of inheritance
    //MarkwithLabWork has all properties and function from its parents
    var $lab;
    function marknumber(){//same func name with MarkStud but different content
        //it will change the same func of MarkStud
        $average = (((($this->kkd1+$this->kkd2+$this->kkd3+$this->kkd4)/4)*2)+$this->lab)/3;
        return $average;
    }
}
```



```
.  
$Mark2 = new MarkwithLabWork();  
$Mark2->dept = "Economics";  
$Mark2->nim = "EM1111";  
$Mark2->name = "Suryo";  
$Mark2->course = "Management";  
$Mark2->kkd1 = "80";  
$Mark2->kkd2 = "80";  
$Mark2->kkd3 = "72";  
$Mark2->kkd4 = "70";  
$Mark2->lab = "99";  
$Mark2->show();
```

Department = Economics

NIM = EM1111

Name = Suryo

Course = Management

Mark_number = 83.3333333333

Mark_letter = B

- Modify with form, ex:

```
<form action="ClassF.php" method="post">
```

```
<h2>Data of Student</h2>
```

```
<table>
```

```
<tr>
```

```
<td>NIM : <input type="text" value="EM1111" />
```

Data of Student

NIM :

Name :

Gender :

Birthday : format yyyy-mm-dd

NIM = EM1111

Name = Raden Mas Suryo

Gender = Male

Age = 22

- Protected modifier in inheritance, modify...

```
class Student{  
    var $dept; var $nim; var $name; var $course;  
    var $kkd1; var $kkd2; var $kkd3; var $kkd4; //all public  
    private $univ; // private  
    protected $city = "Surakarta"; //protected
```

```
class Student_Solo extends Student { // get inheritance from class Student  
    function show(){  
        echo "Universitas: ".$this->univ."<br>";  
        echo "Kota: ".$this->city."<br>";  
        echo "Department: ".$this->dept."<br>";  
        echo "NIM: ".$this->nim."<br>";  
        echo "Name: ".$this->name."<br>";  
        echo "Course: ".$this->course."<br>";  
        echo "Marknumber: ".$this->marknumber()."<br>";  
        echo "Markletter: ".$this->markletter();  
    }  
}
```

```
include 'Class_Student2_lib.php';//include modul

$if0001 = new Student_Solo(); //membuat object baru
$if0001->dept = "Informatika";
$if0001->nim = "IF001";
$if0001->name = "Raden Mas";
$if0001->course = "Web Programming";
$if0001->kkd1 = "88";
$if0001->kkd2 = "80";
$if0001->kkd3 = "78";
$if0001->kkd4 = "86";
$if0001->show();
```

Universitas: Kota: Surakarta Department: Informatika NIM: IF001 Name: Raden Mas Course: Web Programming Marknumber: 83 Markletter: B

- Sometimes (when using inheritance,) anda ingin customize methods dasar dari core class
- Lakukan "override" atas methods core
- Ex: function show1 di class Student will be "override"

```
function show1(){ //in class Student
    echo "Universitas: ".$this->univ."<br>";
    echo "Department: ".$this->dept."<br>";
    echo "NIM: ".$this->nim."<br>";
}
```

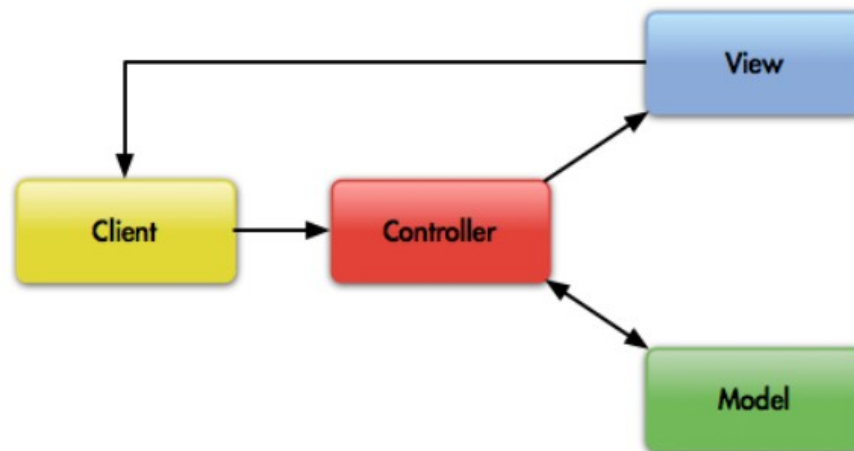
```
class Stu extends Student{
    protected function show1(){ //override, notice show1 here
                                //different within Student
        echo "Universitas: ".$this->univ."<br>";
        echo "Department: ".$this->dept."<br>";
        echo "NIM: ".$this->nim."<br>";
        echo "Name: ".$this->name."<br>";
        echo "Course: ".$this->course."<br>";
        echo "Marknumber: ".$this->marknumber."<br>";
        echo "Markletter: ".$this->markletter();
    }
}
```

- Then U can make Ur own class to manage database, to make connection, or to make validation, a template for website, etc
- OO in PHP make it more dynamic and easier to manage and program
- All U need only improve from the basic and make modularization of Ur OO idea

Simple idea OO

- Easiest one, aplikasi template webpage of website
- Think in OO approach, pemodulan dan dapat dipanggil berulang2 (generally)
- Object apa kira2?
 - Object homepage
 - Object header, footer, left bar, right bar
 - Object pages (selain homepage)

- PHP is going to be OOP
- PHP ext ex PEAR, is OO, Zend Framework also OO
- Framework is set of modules that perform various tasks
- is divided:



- Zend Framework has a bunch core PHP Applications
- Ex:
 - Database access (Zend_DB)
 - Google Data API's (Zend_Gdata)
 - OpenID (Zend_OpenId)
 - etc
- All U need is configure, know the basic of PHP and OO concept in PHP to learn and use it

CMS (Content Management System)

- CMS more about concept than product
- Tepatnya Web CMS
- Besar kecilnya relatif dengan its aim
- Used to for Large Scale Website
- Ex:
 - Moodle → for e-learning
 - Drupal → general website
 - Etc
- Must be dynamic, a bit complex
- Much modularization, OO

- Categories:
 - Authoring, penanganan user pengisi content
 - Workflow, pengaturan alur dari authoring ke publishing
 - Storage, pengiriman atau penyimpanan ke storage
 - Publishing, penyajian di webpage
- Core features:
 - User management
 - User Interface
 - Application
 - Data Source
 - Deployment

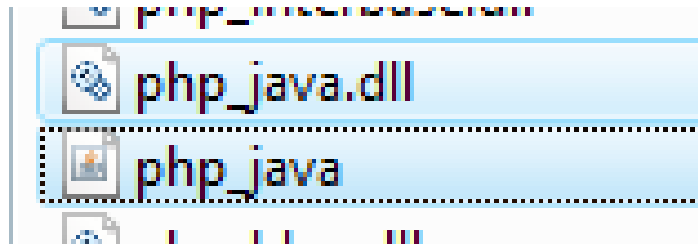
- Simple Ex: CMS Homepage Article
 - Installation
 - Log in Admin
 - Initiation of database/ data file
 - Interface Management article (list article, add,edit,delete-op)
 - Admin interface to add content of article (author, date/time, title, category, body article, etc)

Using Java in PHP

- PHP always improve its extensions for a wide variety of different technologies
- One of the coolest things about PHP has to be the wide range of extensions available for the language, also for wide range of database system
- Java is delicious, isnt it?
- One thing, It has a lot library and very usefull

- By default PHP isn't ready working with Java
- Only need to add configuration (depend on the platform), *.nix or windows
- Assumption in windows, U have installed properly java and take a look in PHP directory, make sure that U have php_java.dll and php_jar.dll.

-



- Then repair the configuration. Open PHP.INI under PHP directory
- Make all [Java] section enabled and correct
- ```
[Java]
java.class.path = C:\xampp\php\ext\php_java.jar
java.home = C:\Program Files\Java\jdk1.6.0_11\bin
java.library = C:\Program Files\Java\jre1.6.0_04\bin\client\jvm.dll
java.library.path = C:\xampp\php\ext
```

- After those steps, Java will be ready as a class for PHP
- And could access Java from PHP Scripts.