



UNS
UNIVERSITAS
SEBELAS MARET

MODUL

Pemrograman Web Lanjut



Agus Purbayu, S.Si, M.Kom

PROGRAM STUDI
D3 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET

Praktikum 1 : Review Pemrograman Web Dasar

A. TUJUAN PEMBELAJARAN

Mahasiswa dapat menggunakan perintah dasar pemrograman web dengan database

1. Mahasiswa mampu menggunakan php link dalam web
2. Mahasiswa mampu menggunakan database dalam web
3. Mahasiswa mampu menggunakan session dalam web

B. WAKTU : 2 x 200 menit

C. MATERI:

Mahasiswa menggunakan html form, php link, database dan session dalam membuat bagian aplikasi sederhana

D. PRAKTIKUM

Mahasiswa mengulang materi desain web dan pemrograman web dasar lewat latihan materi form, database dan session

E. TUGAS

1. Mahasiswa membuat form dengan html dengan type yang berbeda-beda (text, radio, checkbox, select, textarea) untuk inputan data di satu tabel
2. Menggunakan login (session) untuk CRUD untuk satu tabel di database



Praktikum 2 : Pengantar OOP dasar dengan PHP

A. TUJUAN PEMBELAJARAN

Mahasiswa mampu menggunakan OOP dasar php terkaitan aturan class, pembuatan object, pembuatan method dalam program PHP

B. WAKTU : 2 X 200 Menit

C. MATERI:

Penggunaan Atribut dan Class

Membuat OOP di php mempunyai aturan tersendiri sehingga sebelum mengenal framework lebih baik mempelajari terlebih dahulu bagaimana membuat class, atribut dan method/fungsi di php.

Membuat class di php awalnya dengan mendeklarasikan class itu sendiri. Dengan aturan identifier di php maka nama kelas haruslah satu kata dengan tanpa spasi dan awalnya tidak boleh angka (jika terdapat dua kata atau lebih dapat menggunakan tanda _ (underline))

Kelebihan Menggunakan OOP

1. Dengan OOP, kode-kode yang kita buat menjadi lebih rapih dan terstruktur.
2. Dengan OOP, proses reuse kode-kode yang kita buat untuk project yang hampir sama, mudah, karena kode yang kita buat rapih dan terstruktur. Sehingga untuk mengubah atau reuse kode jauh lebih mudah.
3. Konsepnya per modul. Fungsi atau prosedur misal di java atau di PHP tinggal panggil saja nama fungsinya. Jika bermasalah nantinya akan mudah diketahui karena terbaca dari fungsi yang kita panggil. Itulah yang dimaksud konsep per modul. Sehingga dengan OOP, kita dimudahkan untuk membuat dan membaca kode kita (efisiensi kode).
4. Konsep OOP juga memudahkan kita untuk menganalisa program yang kita akan buat. Ini akan sangat terasa kalau kita membuat program besar dan rumit.

Kekurangan Menggunakan OOP

1. Kesulitan dalam ide programnya yang dapat digunakan dalam program
2. Membangun awal perlu ketrampilan programing lebih

D. PRAKTIKUM

Contoh deklarasi class di php dengan atribut dan methodnya



```

1 <?php
2 //contoh class
3 class mahasiswa{
4 //property class
5 public $nim;
6 public $nama;
7 //method class
8 public function isidata($nim,$nama)
9 {
10    $this->nim=$nim;
11    $this->nama=$nama;
12 }
13 public function cetakdata()
14 {
15 echo "Cetak NIM Didalam Class :".$this->nim."</br>";
16 echo "Cetak Nama Didalam Class :".$this->nama."</br>";
17 }
18 }//end of class

```

PHP Hypertext Preprocessor file length : 1.086 lines : 59 Ln : 22 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Berikut adalah cara untuk memanggil menggunakan class dengan membuat file misal mhsaksi.php dengan script sebagai berikut :

```

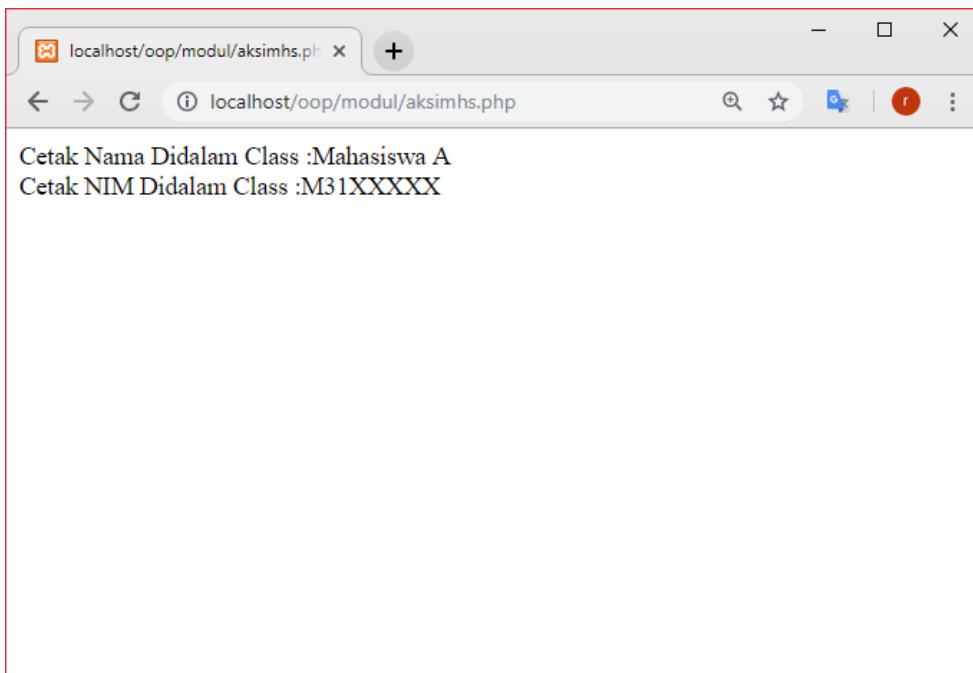
1 <?php
2 require('mhs1.php');
3 $objectmhs1=new mahasiswa();//create object
4 $objectmhs1->isidata("M31XXXXX","Mahasiswa A");//memanggil method
5 $objectmhs1->cetakdata();//memanggil method
6 ?>

```

PHP Hypertext Preprocessor file length : 188 lines : 6 Ln : 6 Col : 3 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Pada gambar diatas adalah contoh membuat object baru berdasarkan class yang sudah ada dan memanggil methodnya. Kemudian diakses browser lewat alamat servernya tampilannya menjadi :





Penggunaan selanjutnya untuk inputan lewat form melalui class adalah sebagai berikut (ditambahkan dari class sebelumnya)

```
16 echo "Cetak Nama Didalam Class :" . $this->nama . "</br>";
17 }
18
19 public function inputdata()
20 {
21     echo "
22 <form action=' ' method='post'
23 NIM :<input type='text' name='nim'><br>
24 Nama :<input type='text' name='nama'><br>
25 <input type='submit' name='tombol' value='Input Mhs'>";
26 }
27
28 public function terimadata()
29 {
30     $this->nim=$_POST['nim'];
31     $this->nama=$_POST['nama'];
32 }
33 }//end of class
```

Kemudian diterapkan dalam file mhsaksi2.php dengan script sebagai berikut:



D:\xampp\htdocs\oop\modul\aksimhs2.php - Notepad++

```
1 <?php
2 require('mhs1.php');
3 $objectmhs1=new mahasiswa(); //create object
4 if(!isset($_POST['tombol']))
5 {
6     $objectmhs1->inputdata();
7 } else
8 {
9     $objectmhs1->terimadata();
10    $objectmhs1->cetakdata();
11 }
12
13
14
15
16
17
```

PHP Hypertext Preprocessor file length : 229 lines : 17 Ln : 15 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Hasil dari penggunaan form lewat class adalah sebagai berikut

localhost/ooo/modul/aksimhs2.php

NIM :
Nama :

Kemudian jika data sudah diinput maka menjadi

localhost/ooo/modul/aksimhs2.php

Cetak NIM Didalam Class :M3111111
Cetak Nama Didalam Class :Paijo Bin Bejo



E. TUGAS

1. Buatlah form untuk menginput dan mencetak hasil 3 jenis inputan data (text, radio dan select).
2. Lanjutkan nomor 1 dengan menyimpan dan mencetak ke dan dari database (input dan cetak database menggunakan method terpisah)



Praktikum 3 : Constructor, Encapsulation dan Inheritance dengan PHP

A. TUJUAN PEMBELAJARAN

1. Mahasiswa mampu menggunakan constructor dalam program PHP
2. Mahasiswa mampu membuat aturan keamanan akses atribut dan method menggunakan encapsulation dalam program PHP
3. Mahasiswa mampu mengembangkan class menggunakan inheritance dalam program PHP

B. WAKTU : 2 x 200 menit

C. MATERI:

Pengertian OOP

OOP adalah cara pandang atau pola pikir dalam menganalisa suatu sistem atau permasalahan pemrograman, dalam OOP semua bagian program dianggap sebagai object

Beberapa konsep dasar OOP

1. Class, Object
2. Encapsulation (method dan atribut dalam class)
3. Inheritance (Penurunan sifat)
4. Polimorfisme

Mobil, hewan, orang adalah Class dan nantinya bisa diciptakan objectnya, bukan hanya fisik saja bisa juga berupa abstrak seperti account bank, file di komputer, pesan email, acara tv dan lain-lain. Object mempunyai properties (atribut) dan melakukan sesuatu (behavior / method)

Class adalah penjelasan dari object seperti untuk object mobil paling tidak mempunyai penjelasan properties tentang warna, tahun rakit dan penjelasan melakukan sesuatu seperti cara mengerem, cara parkir dll.

Contoh class : mobil, contoh object : mobil pak udin, mobil di depan rumah, penjelasan properties disebut atribut dan penjelasan melakukan sesuatu disebut method

D. PRAKTIKUM

Constructor

Constructor adalah memberikan method / fungsi default yang akan dijalankan ketika object diciptakan



D:\xampp\htdocs\oop\modul\mhs1.php - Notepad++

```
1 <?php
2 //contoh class
3 class mahasiswa{
4 //property class
5 public $nim;
6 public $nama;
7
8 public function __construct($nama='Belum Diketahui',$nim='M31XXXXXX')
9 {
10 //echo"Dipanggil ketika membuat object mahasiswa <br> ";
11 $this->nama=$nama;
12 $this->nim=$nim;
13 }
14 //method
15 public function isidata($nim,$nama)
16 {
17 $this->nim=$nim;
18 $this->nama=$nama;
19 }
```

PHP Hypertext Preprocessor file length : 1.246 lines : 62 Ln : 12 Col : 19 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

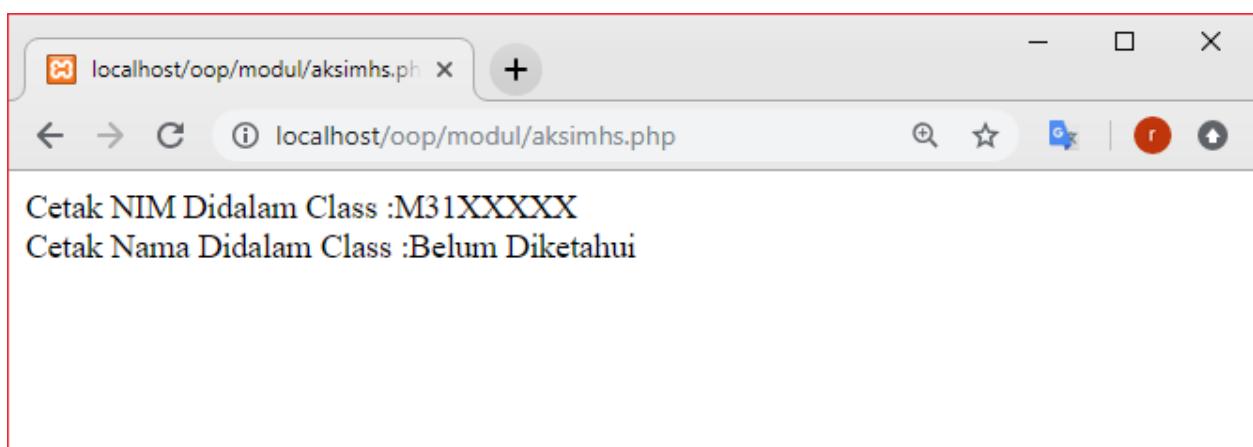
Kemudian digunakan pada mhsaksi.php dengan script berikut :

D:\xampp\htdocs\oop\modul\aksimhs.php - Notepad++

```
1 <?php
2 require ('mhs1.php');
3 $objectmhs1=new mahasiswa(); //create object
4 $objectmhs1->cetakdata(); //memanggil method
5
```

PHP Hypertext Preprocessor file length : 119 lines : 5 Ln : 5 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

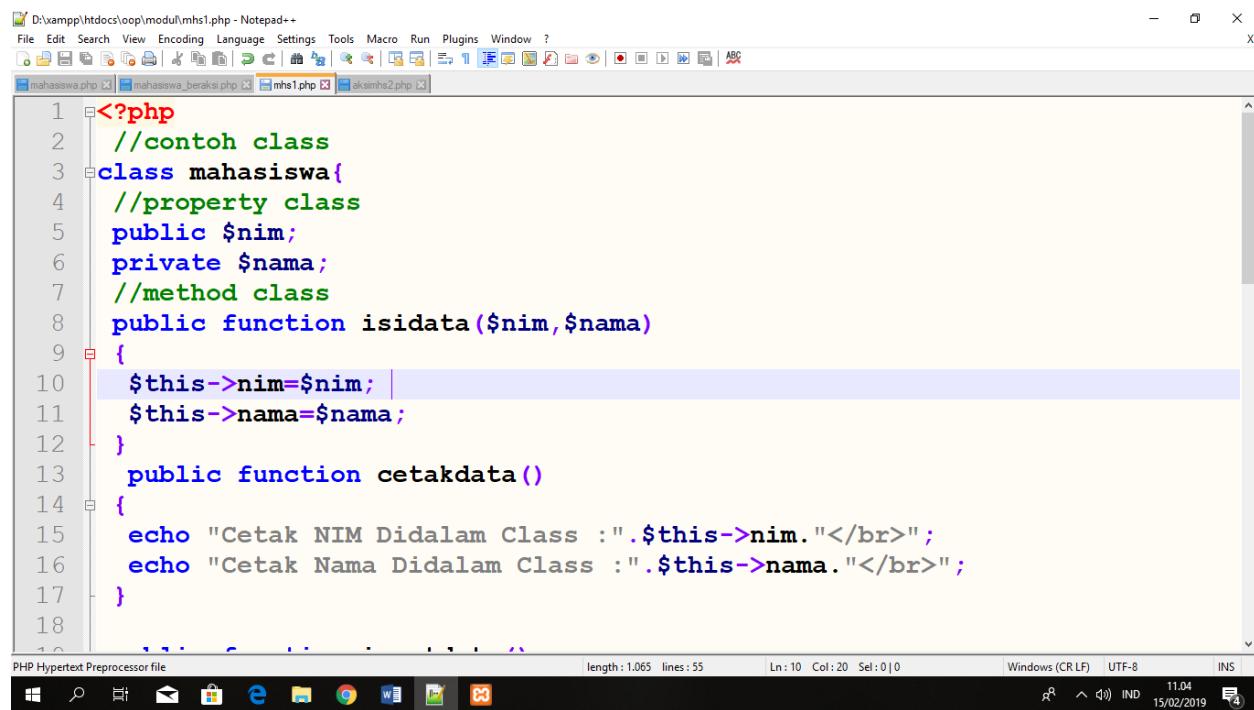
Dan mendapatkan tampilan sebagai berikut :



Encapsulation OOP

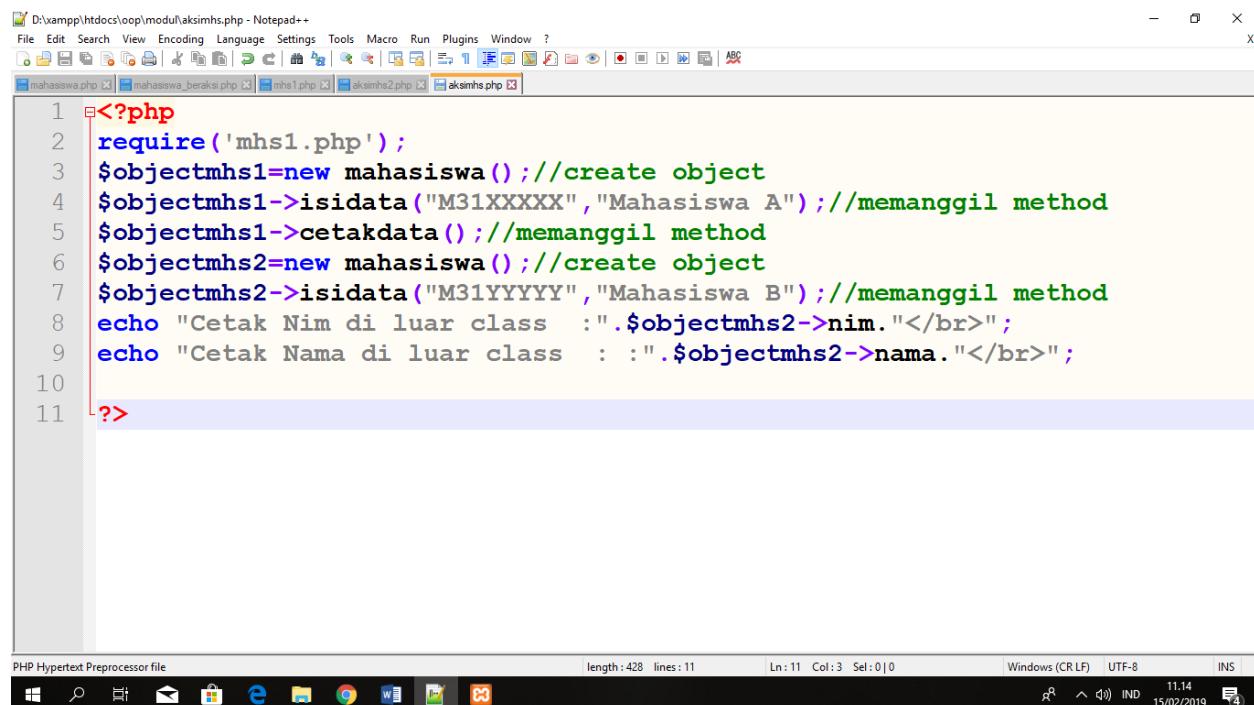


Dengan menambahkan private pada salah satu atribut sebagai berikut berikut



```
<?php
//contoh class
class mahasiswa{
//property class
public $nim;
private $nama;
//method class
public function isidata($nim,$nama)
{
    $this->nim=$nim;
    $this->nama=$nama;
}
public function cetakdata()
{
    echo "Cetak NIM Didalam Class :".$this->nim."</br>";
    echo "Cetak Nama Didalam Class :".$this->nama."</br>";
}
```

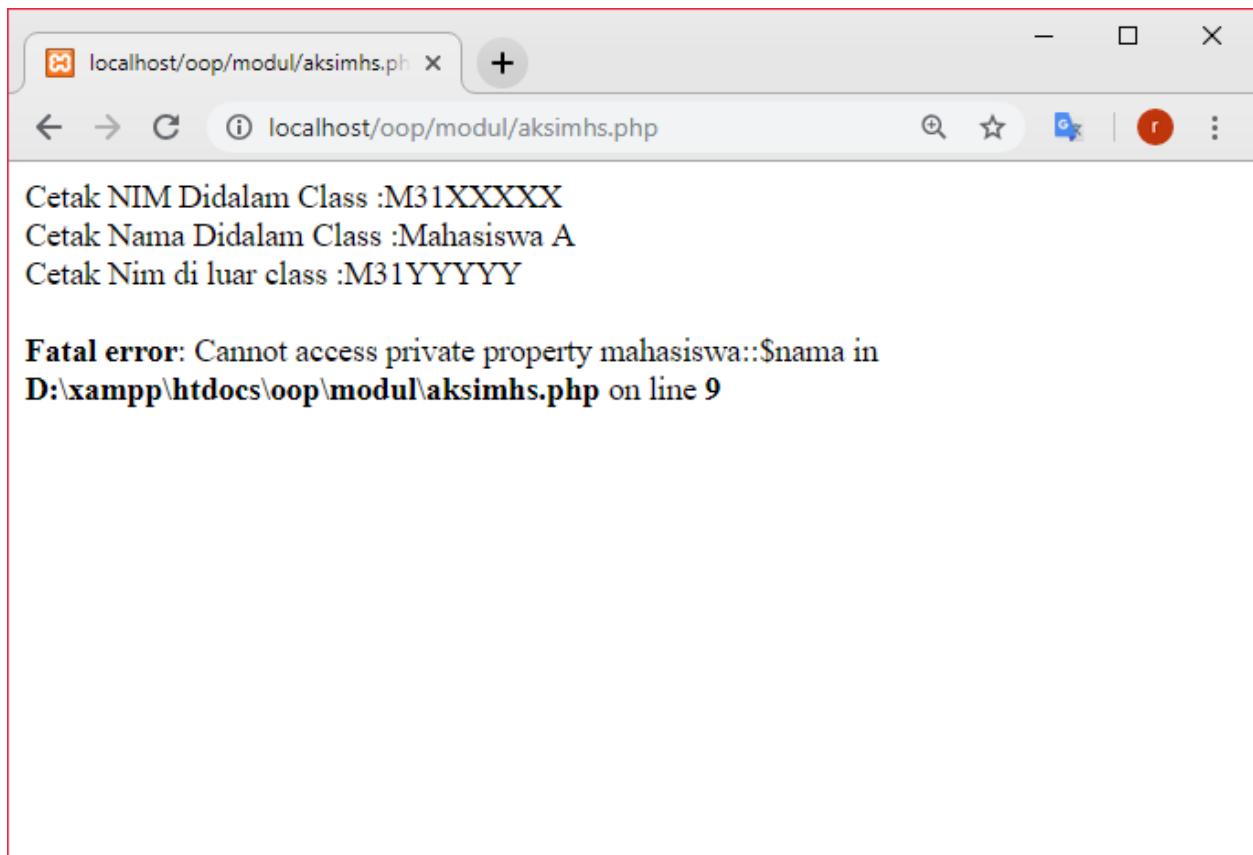
Kenudian diujicobakan dengan aksimhs.php sebagai berikut :



```
<?php
require('mhs1.php');
$objectmhs1=new mahasiswa();//create object
$objectmhs1->isidata("M31XXXXX","Mahasiswa A");//memanggil method
$objectmhs1->cetakdata();//memanggil method
$objectmhs2=new mahasiswa();//create object
$objectmhs2->isidata("M31YYYYY","Mahasiswa B");//memanggil method
echo "Cetak Nim di luar class :".$objectmhs2->nim."</br>";
echo "Cetak Nama di luar class : :".$objectmhs2->nama."</br>" ;
?>
```

Menghasilkan Sebagai berikut





Untuk atribut private tidak diperbolehkan diakses diluar class dan harus diakses lewat class (lewat method seperti contoh berikut)

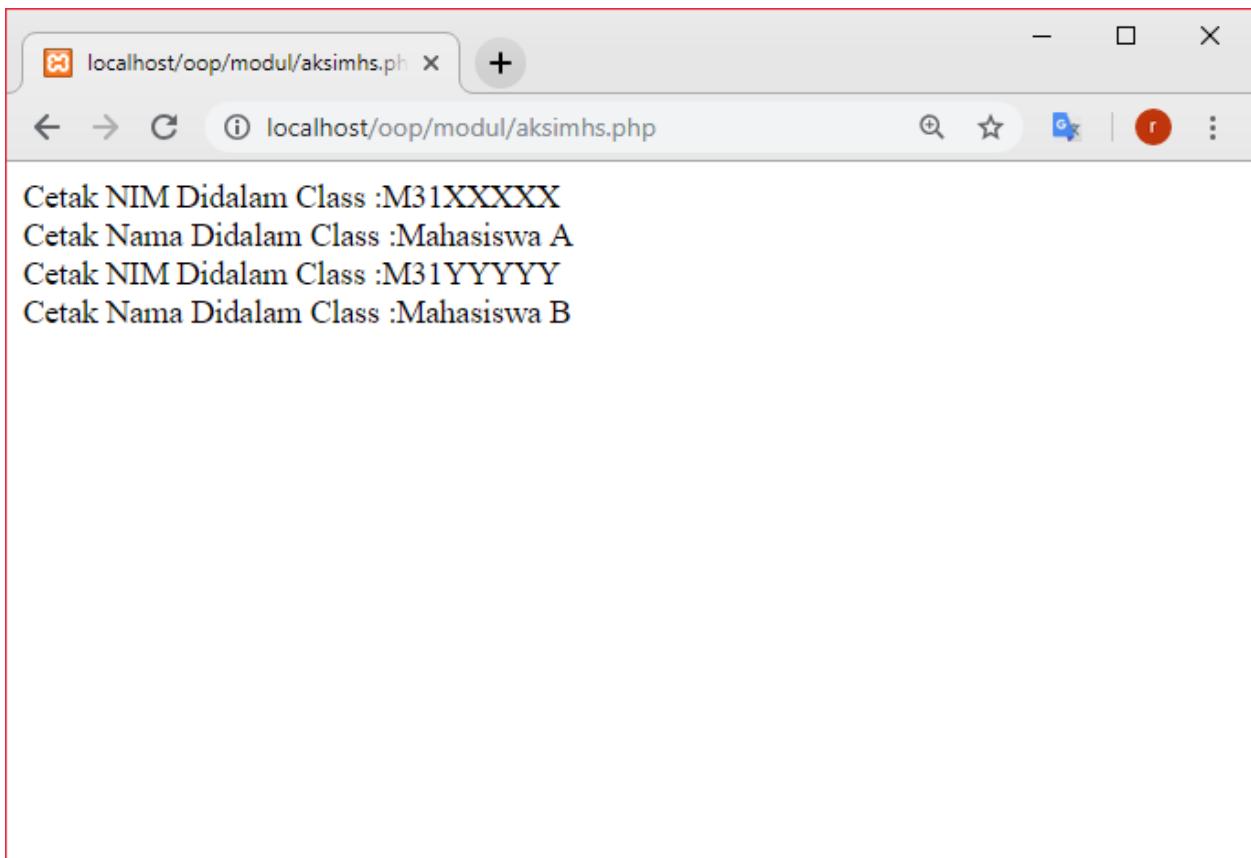
A screenshot of Notepad++ showing a PHP script named 'aksimhs.php'. The code uses the 'require' statement to include 'mhs1.php'. It then creates two objects, '\$objectmhs1' and '\$objectmhs2', both of type 'mahasiswa'. Both objects are used to call the 'isidata' and 'cetakdata' methods. The code is as follows:

```
1 <?php
2 require('mhs1.php');
3 $objectmhs1=new mahasiswa(); //create object
4 $objectmhs1->isidata("M31XXXXX","Mahasiswa A"); //memanggil method
5 $objectmhs1->cetakdata(); //memanggil method
6 $objectmhs2=new mahasiswa(); //create object
7 $objectmhs2->isidata("M31YYYYY","Mahasiswa B"); //memanggil method
8 $objectmhs2->cetakdata(); //memanggil method
9 ?>
```

The status bar at the bottom of the Notepad++ window shows file statistics: length: 345, lines: 9, and encoding: UTF-8.

Kemudian diakses sebagai berikut





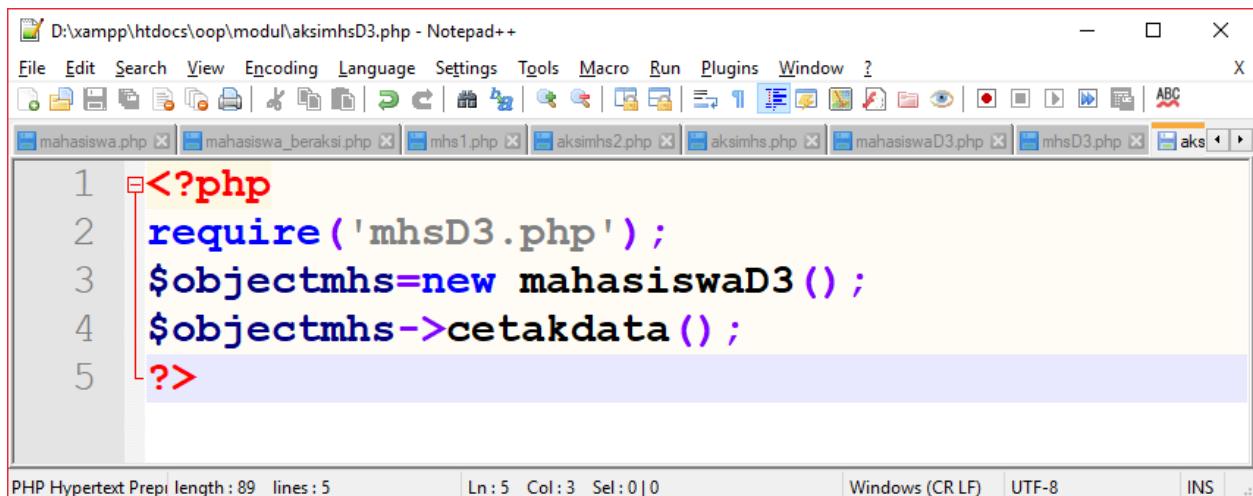
Penurunan Class (Inheritance)

Penurunkan class mahasiswa menjadi mahasiswaD3 seperti contoh berikut :

```
<?php
require('mhs1.php');
class mahasiswaD3 extends mahasiswa{
    //property class
    private $TA;
    //method class
    public function __construct()
    {
        parent::__construct();
        $this->TA = "Judul TA ";
    }
    public function cetakdata() //overriding
    {
        echo "Cetak Nama Didalam Class :".$this->nama."<br>";
        echo "Cetak NIM Didalam Class :".$this->nim."<br>";
        echo "Cetak TA Didalam Class :".$this->TA."<br>";
    }
}
```

Kemudian class turunan dapat digunakan sebagai berikut





D:\xampp\htdocs\oop\modul\aksimhsD3.php - Notepad++

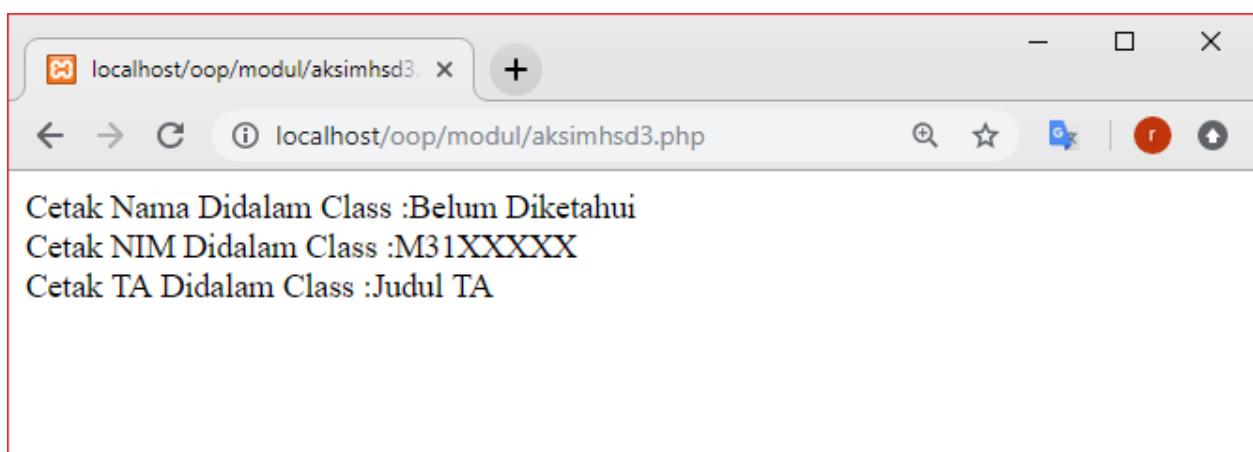
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

mahasiswa.php mahasiswa_beraksi.php mhs1.php aksimhs2.php aksimhs.php mahasiswaD3.php mhsD3.php aks

```
1 <?php
2 require('mhsD3.php');
3 $objectmhs=new mahasiswaD3();
4 $objectmhs->cetakdata();
5 ?>
```

PHP Hypertext Preprocessor length : 89 lines : 5 Ln : 5 Col : 3 Sel : 0 | 0 Windows (CR LF) UTF-8 INS .

Diakses browser sebagai berikut :



E. TUGAS

- Buatlah Superclass Circle dan Subclass Cylinder
- Buat fungsi Menghitung Luas (getArea()) pada Class Circle
- Buat fungsi Menghitung Volume dan Luas Permukaan (menggunakan nama fungsi yang sama : getArea()) pada Class Cylinder yang menggunakan gabungan fungsi yang telah ada pada Class Circle (getArea())



Praktikum 4 : Menggunakan Ide OOP class form

A. TUJUAN PEMBELAJARAN

1. Mahasiswa mampu menggunakan class form (salah satu ide OOP) dalam program PHP
2. Mahasiswa mampu mengembangkan class form dalam program PHP

B. WAKTU : 2 x 200 menit

C. MATERI:

Penggunaan Class Form

Mahasiswa mempelajari salah satu ide programing class form dalam program PHP

D. PRAKTIKUM

Mencoba Ide Class Form

Berikut adalah salah satu contoh ide dari class. Script untuk class form ini dapat digunakan untuk menampilkan form secara umum

```
<?php  
class Form{  
var $fields = array();  
var $action;  
var $submit = "";  
var $jumField=0;  
  
function __construct($action, $submit){  
$this->action = $action;  
$this->submit = $submit;  
}  
  
function displayForm(){  
echo "<form action=\"".$this->action." method='post'>";  
echo "<table width='100%'>";  
for($i=0;$i<count($this->fields);$i++)  
{
```



```

echo"<tr>
    <td align='right'>".$this->fields[$i]['label']."</td>";
echo"<td><input type='text' name='".$this->fields[$i]['name']."'></td>
</tr>";
}

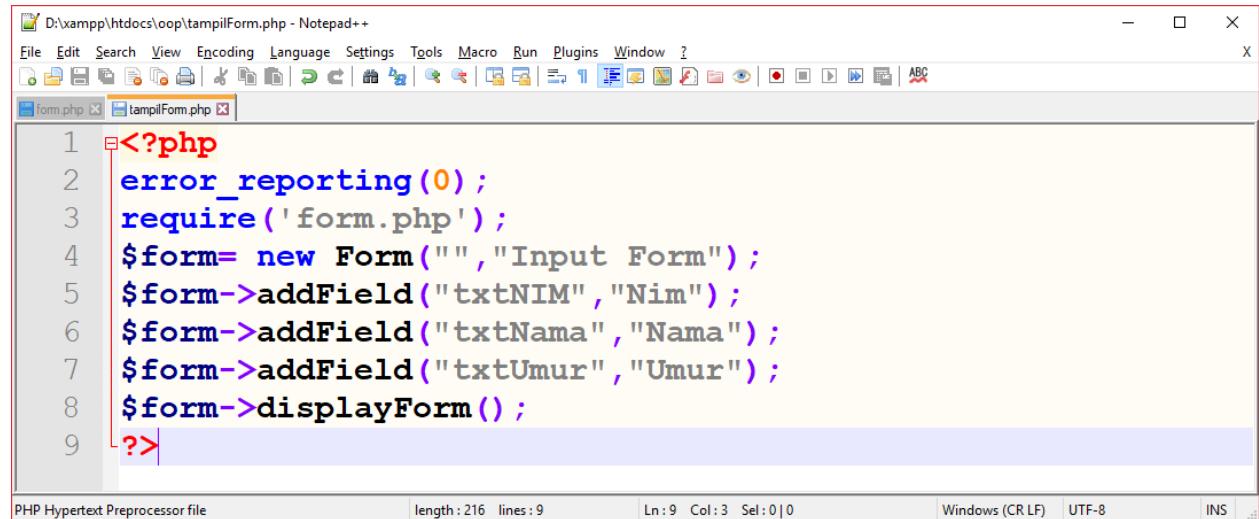
echo"<tr><td></td><td><input type='submit' name='tombol'
value='".$this->submit."' ></td></tr>";
echo"</table>";
}

function addField($name,$label){
    $this->fields[$this->jumField]['name']=$name;
    $this->fields[$this->jumField]['label']=$label;
    $this->jumField++;
}
}

?>

```

Kemudian untuk menampilkan hasilnya dengan memanggil dalam tampilForm sebagai berikut



```

1 <?php
2 error_reporting(0);
3 require('form.php');
4 $form= new Form("", "Input Form");
5 $form->addField("txtNIM", "Nim");
6 $form->addField("txtNama", "Nama");
7 $form->addField("txtUmur", "Umur");
8 $form->displayForm();
9 ?>

```

PHP Hypertext Preprocessor file length : 216 lines : 9 Ln : 9 Col : 3 Sel : 0 | 0 Windows (CR LF) UTF-8 INS .

Dan ditampilkan sebagai berikut



A screenshot of a web browser window titled "localhost/oop/tampilForm.php". The page displays a form with three text input fields: "Nim", "Nama", and "Umur", followed by a submit button labeled "Input Form".

E. TUGAS

1. Lengkapilah ide form dengan hasil inputan dari form
2. Tambahkan form dengan type inputan selain text misalkan radio, checkbox, select dan textarea.



Praktikum 5 : Menggunakan Patern MVC

A. TUJUAN PEMBELAJARAN

1. Mahasiswa mampu menggunakan OOP dan MVC dalam program PHP
2. Mahasiswa mampu mengembangkan OOP dan MVC dalam program PHP

B. WAKTU : 2 x 200 menit

C. MATERI:

MVC adalah singkatan dari Model View Controller. MVC adalah teknik pemrograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) da presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses.

Model

Berhubungan dengan data dan interaksi ke text base, database atau webservice. Biasanya di dalam model berisi akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data web.

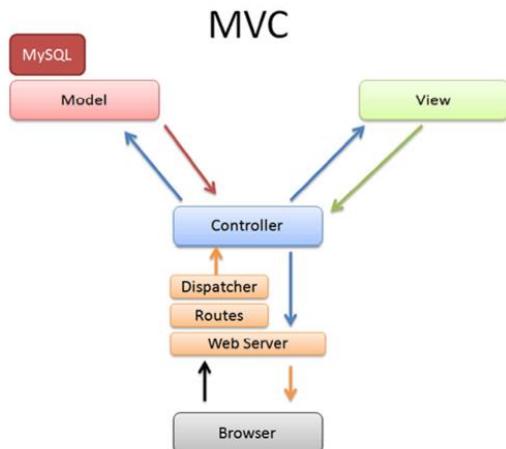
View

Berhubungan dengan segala sesuatu yang ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. View bisa dikatakan sebagai halaman yang dibuat dengan HTML dan bantuan CSS atau Javascript.

Controller

Controller bertindak sebagai penghubung data dan view. Di Dalam controller inilah terdapat class-class dan fungsi-fungsi yang dapat memproses permintaan dari view ke dalam struktur data di dalam model. Controller juga tidak boleh berisi kode utuk mengakses basis data. Tugasnya adalah menyediakan berbagai variabel yg akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan, mengerjakan proses logika dari aplikasi dan melakukan validasi atau cek terhadap input.





D. PRAKTIKUM

Berikut adalah salah satu contoh program web OOP dan MVC

The screenshot shows a Notepad++ window displaying a PHP file named 'book.php'. The code defines a class 'Book' with properties \$judul, \$pengarang, \$penerbit, and \$tahun, and a constructor __construct() that initializes these properties. The code is color-coded, and line numbers 1 through 14 are visible on the left.

```

1 <?php
2 class Book {
3     public $judul;
4     public $pengarang;
5     public $penerbit;
6     public $tahun;
7     public function __construct($judul,$pengarang,$penerbit,$tahun) {
8         $this->judul=$judul;
9         $this->pengarang=$pengarang;
10        $this->penerbit=$penerbit;
11        $this->tahun=$tahun;
12    }
13 } //end class
14 ?>
  
```

At the bottom of the Notepad++ interface, there is a status bar showing file information like length: 284, lines: 14, and file details like Windows (CRLF), UTF-8, and INS. The taskbar at the very bottom of the screen also displays various application icons.



D:\xampp\htdocs\mvc\bookmodel.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

form.php mvcForm.php mvccontroller.php mvc.php mvcmodel.php book.php bookmodel.php bookview.php bookcontroller.php bookberaksip.php

```
1 <?php
2 include_once('book.php');
3 class BookModel{
4 public function getData(){
5     return array(
6         new Book('buku 1', 'pengarang 1', 'penerbit 1', 'tahun 1'),
7         new Book('buku 2', 'pengarang 2', 'penerbit 2', 'tahun 2'),
8         new Book('buku 3', 'pengarang 3', 'penerbit 3', 'tahun 3'),
9         new Book('buku 4', 'pengarang 4', 'penerbit 4', 'tahun 4'));
10    }
11 }
12 ?>
```

PHP Hypertext Preprocessor file length: 336 lines: 12 Ln:1 Col:1 Sel:0|0 Windows (CR LF) UTF-8 19.17 18/02/2010 INS

The screenshot shows a Notepad++ window with the following code:

```
<html>
<head>
</head>
<body>
<center>
<?= $data ?>
</center>
</body>
</html>
```

The code is syntax-highlighted, with tags in blue and variables in red. The variable `$data` is highlighted in red and enclosed in a light purple selection box, indicating it is selected or being edited.

D:\xampp\htdocs\mvc\bookcontroller.php - Notepad++

```
1 <?php
2 include_once('bookmodel.php');
3 class BookController{
4     function invoke(){
5         $model_data=new BookModel();
6         $row_data=$model_data->getData();
7         $no=1; $data="";
8         $data.= "<table border='1'>
9             <tr> <th>No</th><th>Judul</th><th>Pengarang</th>
10            <th>Penerbit</th><th>Tahun</th> </tr>";
11         foreach ($row_data as $key =>$value){
12             $data.= "<tr>
13                 <td>$no</td><td>" . $value->judul. "</td>
14                 <td>" . $value->pengarang. "</td><td>" . $value->penerbit. "</td>
15                 <td>" . $value->tahun. "</td> </tr>";
16         }
17         $no++;
18     }
19 }
```

PHP Hypertext Preprocessor file length : 539 lines : 19 Ln : 18 Col : 2 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

D:\xampp\htdocs\mvc\bookberaksi.php - Notepad++

```
1 <?php
2 include_once('bookcontroller.php');
3 $main_ctrl=new BookController();
4 $main_ctrl->invoke();
5 ?>
```

PHP Hypertext Preprocessor file length : 103 lines : 5 Ln : 1 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

E. TUGAS

1. Buatlah tampilan list data buku dari tabel database menggunakan OOP dan MVC
2. Buatlah inputan data buku ke database menggunakan OOP dan MVC



Praktikum 6 : Menggunakan OOP MVC untuk CRUD

A. TUJUAN PEMBELAJARAN

Mahasiswa mampu menggunakan OOP dengan ide class form dan MVC dalam CRUD tabel database

B. WAKTU : 3 x 200 menit

C. MATERI:

project kelompok CRUD OOP dan MVC

D. PRAKTIKUM

Mahasiswa membuat program secara berkelompok membuat CRUD tabel di database menggunakan OOP dengan ide class form dan pattern MVC dalam program PHP

E. TUGAS

Pilih tema (tidak boleh sama dalam satu kelas) untuk isian 1 tabel CRUD produk barang maupun jasa sebagai tugas kelompok menggunakan OOP dan ide class form yang ditambahkan pattern MVC.

Contoh tema :

Data Buku Perpustakaan, Data Tiket Kereta Api, Data Paket Outbond, Data Kamar Hotel, Data Kelas Materi Kursus Komputer, Data Menu Restoran Cepat Saji.

