

Praktikum 4

OBJECT ORIENTED PROGRAMMING (OOP)

Kompetensi Dasar: Memahami tentang metode pemrograman berorientasi objek dalam bahasa pemrograman PHP serta mampu membuat program dengan metode pemrograman berorientasi objek

Pokok Bahasan:	1. Class dan Object 2. Atribut dan Method
Indikator:	1. Mampu menjelaskan prinsip-prinsip metode pemrograman berorientasi objek 2. Mampu membuat program sederhana dengan menerapkan pemrograman berorientasi objek dalam bahasa pemrograman PHP
Waktu:	2 x 50 menit (1 pertemuan)

A. RINGKASAN MATERI

1. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek adalah metode pemrograman yang memodelkan kompleksitas dalam dunia nyata sebagai sekumpulan **objek** yang memiliki karakteristik dan perilaku tertentu serta dapat saling berinteraksi untuk menghasilkan satu perilaku pada level yang lebih tinggi

2. Class

Class adalah elemen utama dalam pemrograman berorientasi objek. *Class* adalah definisi atau *template* dari objek. Karakteristik dari objek didefinisikan dalam *class*, dan semua objek yang dibentuk dari sebuah *class* akan memiliki karakteristik yang sama. Karakteristik dari sebuah *class* objek terdiri dari sejumlah atribut dan sejumlah method.

Atribut

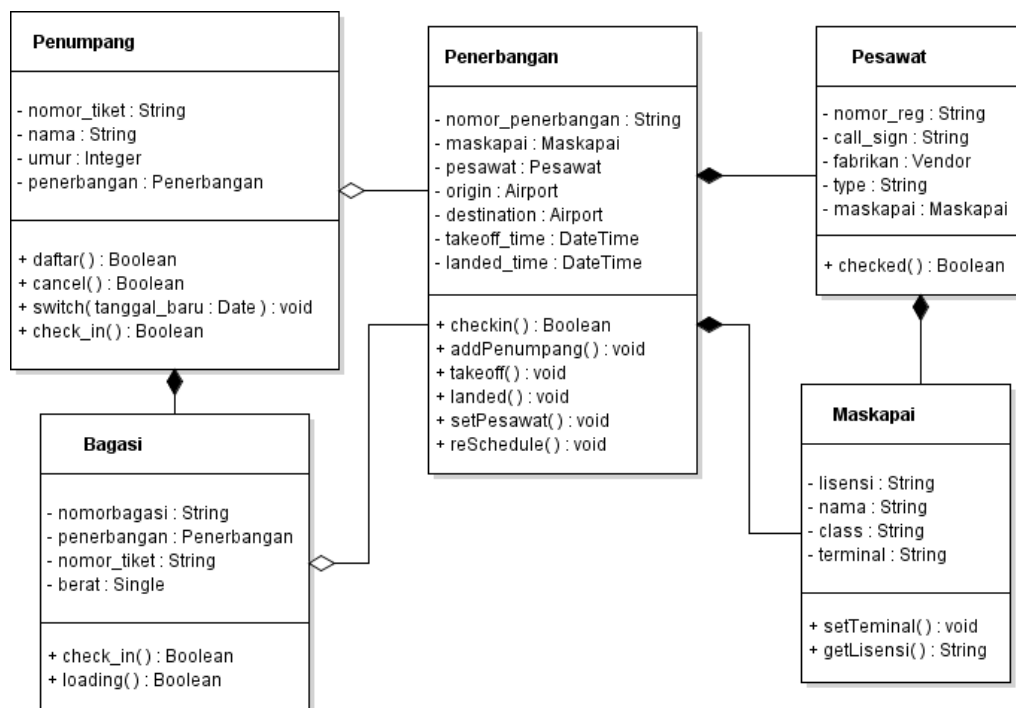
Atribut adalah variabel yang menjadi pengenalan, ciri, atau *property* dari sebuah objek. Atribut membedakan secara prinsip antara sebuah *class* objek dengan *class* objek yang lain, yang berarti dalam sebuah problem domain atau penyelesaian masalah, tidak ada *class* yang memiliki atribut yang persis sama. Atribut objek didefinisikan pada *class*, dan setiap objek yang dibentuk dari satu *class* yang sama akan mewarisi atribut yang sama tetapi dengan nilai atribut yang mungkin berbeda.

Method

Method merupakan modul-modul fungsi (*function*) atau prosedur (*procedure*) yang dimiliki oleh sebuah *class*. *Method* membuat sebuah *class* menjadi memiliki *behavior* atau kemampuan untuk melakukan aktivitas, baik untuk menangani atribut-atribut di dalam *class* itu sendiri ataupun untuk berinteraksi dengan *class* lain dalam sebuah sistem fungsional/aplikasi.

Class Diagram

Dalam merencanakan aplikasi dengan konsep berorientasi objek, *class* dirancang dengan menggunakan model Class Diagram yang menggambarkan rancangan *class* beserta atribut-atribut dan *method-method*nya serta relasinya dengan *class-class* yang lain. Lihat contoh Class Diagram pada Error! Reference source not found..



Gambar 1 Contoh Class Diagram

Implementasi Class

Menulis kode untuk implementasi *class* dimulai dengan deklarasi nama *class* menggunakan statement

```
class Mahasiswa
{
    ...
    ...
} //end class    (pastikan ini paling bawah)
```

lalu diikuti dengan menuliskan deklarasi atribut dan *method* didalam blok *class*.

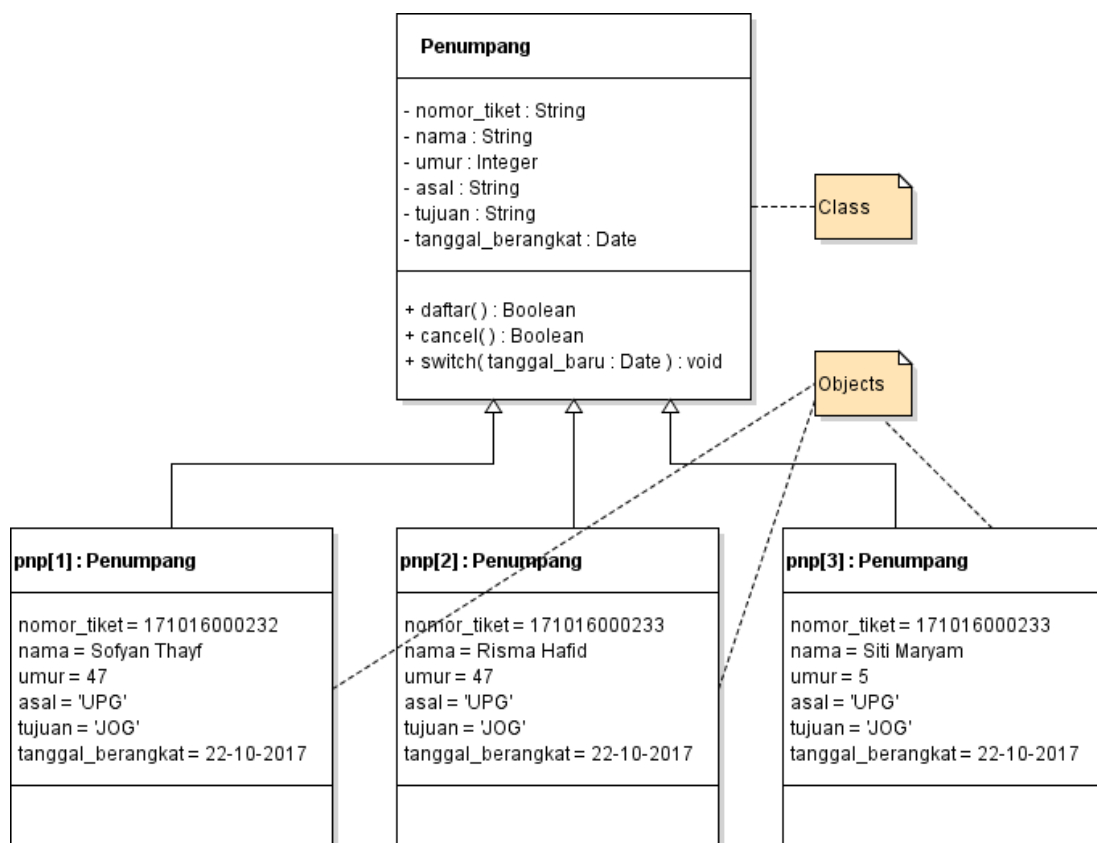
Class biasanya ditulis dalam satu file tersendiri dengan nama file yang sama dengan nama *class*, dan nama *class* biasanya ditulis dengan style Title Case (huruf pertama setiap kata menggunakan huruf besar). Seperti aturan penulisan variabel di hampir semua bahasa pemrograman, nama *class* juga harus menggunakan *alpha-numeric*, tetapi tidak dimulai dengan angka, dan tidak mengandung spasi atau tanda baca.

This

Class adalah template dari objek, saat class dirancang dan dideklarasikan, nama objek yang akan diturunkan dari class tersebut belum diketahui dan akan bisa sangat bervariasi, sehingga untuk merujuk pada diri sendiri, class menggunakan *keyword this*. Atribut atau method yang mengikuti *this*. berarti atribut dan *method* milik class itu sendiri.

3. Objek

Object atau objek merupakan turunan atau *instance* dari *class*. Sebagai turunan maka objek mewarisi semua sifat-sifat *class*-nya, berupa atribut dan *method*. Dari sebuah class dapat dibentuk sejumlah objek dengan atribut yang sama tetapi dengan nilai atribut yang berbeda-beda sehingga objek menjadi unik antara satu dengan yang lainnya (lihat **Gambar 2**).



Gambar 2 Class dan Objek turunannya

Implementasi Objek

Untuk mengimplementasikan objek, terlebih dahulu harus dideklarasikan variabel objek dengan menggunakan *class*-nya sebagai tipe dari variabel objek, lalu membentuk objek dengan perintah **new**. Pastikan sebelum membentuk objek, *class*-nya sudah didefinisikan terlebih dahulu.

```
$mahasiswa = new Mahasiswa();  
  
// atau  
  
$mahasiswa = new Mahasiswa( $nim ); //jika membutuhkan parameter
```

B. PERSIAPAN PRAKTIKUM

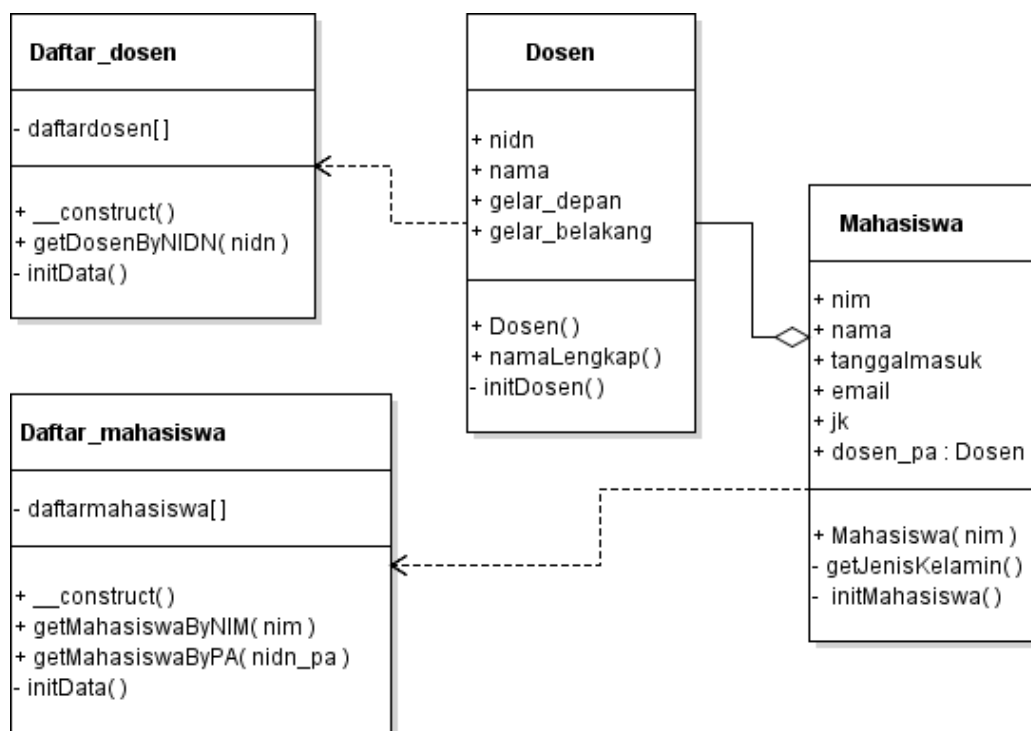
1. Perangkat Keras
 - a. Perangkat komputer yang terkoneksi ke Internet
2. Perangkat Lunak (ter-*install* di komputer, dan siap digunakan)
 - a. Text/code editor, dianjurkan menggunakan Atom
 - b. Virtual web server berbasis Docker atau XAMPP
 - c. Web Browser (Google Chrome, Mozilla Firefox)
3. Buat folder baru bernama **praktikum4** pada direktori server latihan Anda, lalu *copy* semua file dan folder bahan praktikum dari folder sharing GDrive untuk kelas Desain dan Pemrograman Web

shared folder: **20201 WebDev\Web Programming\Praktikum\bahan praktikum\praktikum 4**

direct link: <https://drive.google.com/drive/folders/1AvC-R3MNAXwt3EhGJFcDFMiaq8CA-i6Y?usp=sharing>

C. KEGIATAN PRAKTIKUM

Dalam praktikum ini, akan dibuat sebuah aplikasi informasi data mahasiswa sederhana yang menampilkan data mahasiswa dan dosen Pembimbing Akademik (PA)-nya. Untuk itu dibutuhkan dua *class* objek yang berelasi yaitu *class* Mahasiswa dan *class* Dosen. Perhatikan gambar *class diagram* dalam Gambar 3. Relasi antara Mahasiswa dengan Dosen adalah **relasi agregasi** di mana objek mahasiswa dapat berbagi objek dosen karena beberapa mahasiswa dapat memiliki dosen PA yang sama



Gambar 3 Rancangan aplikasi dalam Class Diagram

Class **Daftar_dosen** dan class **Daftar_mahasiswa** hanya merupakan simulasi dari sumber data karena dalam praktikum ini belum melibatkan *database* sebagai sumber data.

Class **Daftar_Dosen** dan class **Daftar_mahasiswa** sudah tersedia dan tidak perlu diketik ulang dalam praktikum ini

1. Buka file **Daftar_dosen.php** dan **Daftar_mahasiswa.php** pada text-editor (Atom), pelajari dan pahami struktur dan isinya.
2. Buat file baru bernama **Mahasiswa.php**, dan ketik dengan teliti kode program dalam **Listing 1**.

Bandingkan kode program *class* **Mahasiswa** dengan gambar diagram *class* **Mahasiswa** dalam **Gambar 3**

Listing 1

```
<?php

/**
 * class model data Mahasiswa
 */
class Mahasiswa
{
    public $nim;
    public $nama;
    public $tanggalmasuk;
    public $email;
    public $jk;
    public $dosen_pa;

    public function Mahasiswa($nim)
    {
        $this->nim = $nim;
        $this->initMahasiswa();
    }

    private function initMahasiswa()
    {
        $listmhs = new Daftar_mahasiswa();
        $mhs = $listmhs->getMahasiswaByNIM( $this->nim );

        $this->nama = $mhs['namamhs'];
        $this->tanggalmasuk = $mhs['tanggal_masuk'];
        $this->email = $mhs['email'];
    }

}

} //end class    (pastikan ini paling bawah)

?>
```

Pertanyaan

1. Apa maksud dari tanda + (plus) dan tanda – (minus) di depan nama atribut dan nama *method*, dalam penggambaran *class diagram* ?
3. Buat file baru bernama **program04.php** dan ketik struktur dasar HTML diikuti beberapa elemen *layout* (dua kolom, lebar 8 dan lebar 4), jangan lupa link ke file **style.css**, dan beberapa kode PHP seperti contoh dalam **Listing 2**.

Listing 2

```
<?php
    require_once("Daftar_mahasiswa.php");
    require_once("Daftar_dosen.php");
    require_once("Mahasiswa.php");
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
```

```
<meta charset="utf-8">
<link rel="stylesheet" href="css/style.css">
<title>Program 04</title>
</head>
<body>

  <div class="container">
    <div class="row">

      <div class="col8">

      </div>

      <div class="col4">

      </div>

    </div>
  </div>

</body>
</html>
```

Perhatikan baris-baris kode PHP berikut ini:

```
require_once("Daftar_mahasiswa.php");
require_once("Daftar_dosen.php");
require_once("Mahasiswa.php");
```

Baris-baris ini adalah perintah untuk melibatkan file PHP lain ke dalam sebuah file program PHP.

4. Lanjutkan dengan meng-edit file program04.php untuk mengisi kolom kiri (blok `<div class="col8"> .. </div>`) dengan kode program seperti pada Listing 3

Listing 3

```
...
<div class="col8">
  <?php
    if ( isset( $_GET['nim'] ) ) {

      // pembentukan object $mahasiswa dari class Mahasiswa
      // berdasarkan NIM
      $mahasiswa = new Mahasiswa( $_GET['nim'] );

    }
  >
  <h1>Mahasiswa</h1>
  <table>
```

```
<tr>
  <td class="field">N I M:</td>
  <td class="data"><?= $mahasiswa->nim ?></td>
</tr>
<tr>
  <td class="field">Nama Lengkap:</td>
  <td class="data"><?= $mahasiswa->nama ?></td>
</tr>
<tr>
  <td class="field">E-mail:</td>
  <td class="data"><?= $mahasiswa->email ?></td>
</tr>
<tr>
  <td class="field">Tanggal masuk:</td>
  <td class="data">
    <?= date('d F Y',strtotime($mahasiswa->tanggalmasuk)) ?>
  </td>
</tr>
</table>

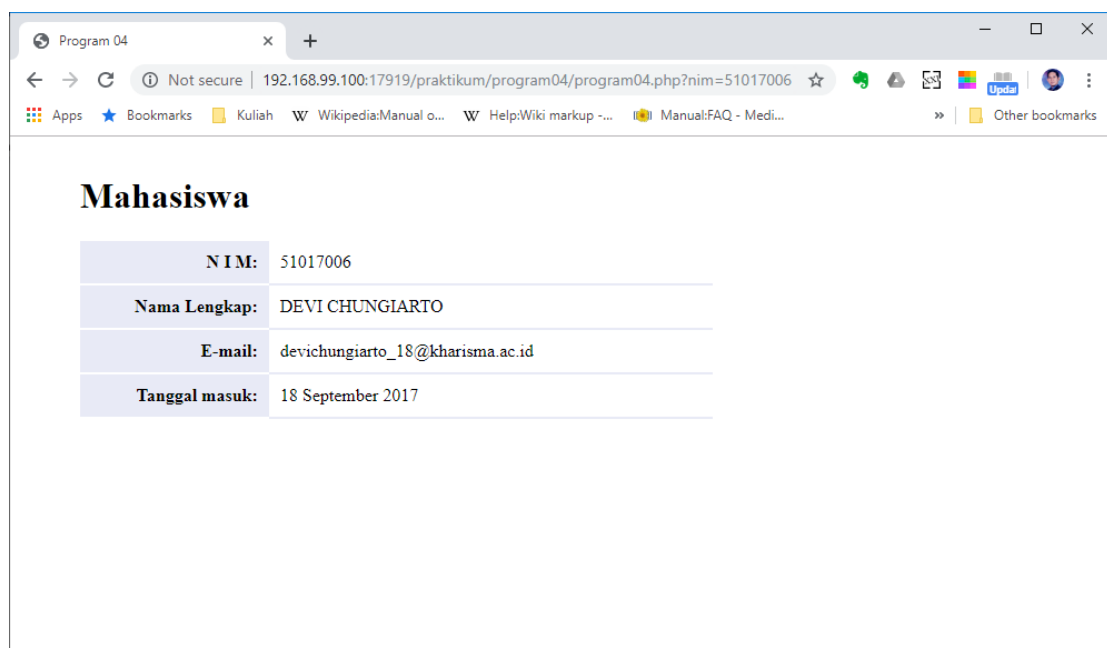
<?php
}
?>
</div>

<div class="col4">

</div>

...
```

Jika sudah, lihat hasilnya pada *web-browser* (lihat **Gambar 4**)



Gambar 4

5. Edit file class **Mahasiswa.php** untuk menambahkan fungsi `getJenisKelamin($kodep)` seperti pada **Listing 4**

Listing 4

```
...

private function initMahasiswa()
{
    $listmhs = new Daftar_mahasiswa();
    $mhs = $listmhs->getMahasiswaByNIM( $this->nim );

    $this->nama = $mhs['namamhs'];
    $this->tanggalmasuk = $mhs['tanggal_masuk'];
    $this->email = $mhs['email'];

    $this->jk = $this->getJenisKelamin($mhs['lp']);

}

private function getJenisKelamin( $kodep )
{
    $jk = [ 'L' => [ 'kode' => 'L', 'gender' => 'Laki-laki' ],
            'P' => [ 'kode' => 'P', 'gender' => 'Perempuan' ] ];
    return $jk[ $kodep ];
}
```

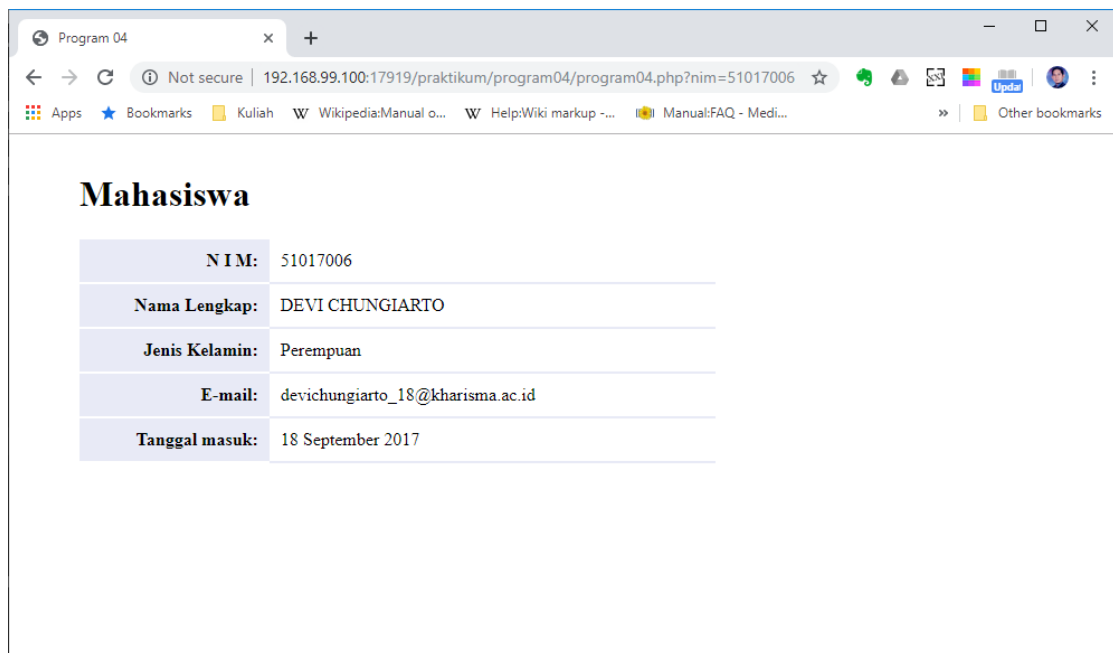
6. Lakukan modifikasi pada file **program04.php** seperti dalam **Listing 5** untuk menambahkan menampilkan data jenis kelamin mahasiswa. Lihat hasilnya pada browser (lihat **Gambar 5**)

Listing 5

```
...

<h1>Mahasiswa</h1>
<table>
    <tr>
        <td class="field">N I M:</td>
        <td class="data"><?= $mahasiswa->nim ?></td>
    </tr>
    <tr>
        <td class="field">Nama Lengkap:</td>
        <td class="data"><?= $mahasiswa->nama ?></td>
    </tr>
    <tr>
        <td class="field">Jenis Kelamin:</td>
        <td class="data"><?= $mahasiswa->jk['gender'] ?></td>
    </tr>
    <tr>
        <td class="field">E-mail:</td>
        <td class="data"><?= $mahasiswa->email ?></td>
    </tr>
</table>
```

```
</tr>
<tr>
  <td class="field">Tanggal masuk:</td>
  <td class="data">
    <?= date('d F Y',strtotime($mahasiswa->tanggalmasuk)) ?>
  </td>
</tr>
</table>
...
```



Gambar 5

7. Buat file baru bernama **Dosen.php** dan salin kode class **Dosen** dari Listing 6

Listing 6

```
<?php

/**
 * class model data Dosen
 */
class Dosen
{
    public $nidn;
    public $nama;
    public $gelar_depan;
    public $gelar_belakang;

    public function Dosen($nidn)
    {
        $this->nidn = $nidn;
        $this->initDosen();
    }
}
```

```
}

private function initDosen()
{
    $listdsn = new Daftar_dosen();
    $dsn = $listdsn->getDosenByNIDN($this->nidn);

    $this->nama = $dsn['nama'];
    $this->gelar_depan = $dsn['gelar_depan'];
    $this->gelar_belakang = $dsn['gelar_belakang'];
}

} //end class    (pastikan ini paling bawah)
?>
```

8. Modifikasi *method* / function `initMahasiswa()` dari class `Mahasiswa` dalam file `Mahasiswa.php` untuk menerapkan / memanggil class `Dosen` yang baru sebagai objek `dosen_pa`. Lihat **Listing 7**

Listing 7

```
...

private function initMahasiswa()
{
    $listmhs = new Daftar_mahasiswa();
    $mhs = $listmhs->getMahasiswaByNIM( $this->nim );

    $this->nama = $mhs['namamhs'];
    $this->tanggalmasuk = $mhs['tanggal_masuk'];
    $this->email = $mhs['email'];
    $this->jk = $this->getJenisKelamin($mhs['lp']);

    $this->dosen_pa = new Dosen( $mhs['dosen_pa'] );

}

...
```

9. Lakukan modifikasi pada file `program04.php` seperti dalam **Listing 5** untuk menambahkan menampilkan data dosen PA mahasiswa. Lihat hasilnya pada browser

Listing 8

```
...
<h1>Mahasiswa</h1>
<table>
  <tr>
    <td class="field">N I M:</td>
    <td class="data"><?= $mahasiswa->nim ?></td>
  </tr>
  <tr>
```

```
<td class="field">Nama Lengkap:</td>
<td class="data"><?= $mahasiswa->nama ?></td>
</tr>
<tr>
<td class="field">Jenis Kelamin:</td>
<td class="data"><?= $mahasiswa->jk['gender'] ?></td>
</tr>
<tr>
<td class="field">E-mail:</td>
<td class="data"><?= $mahasiswa->email ?></td>
</tr>
<tr>
<td class="field">Tanggal masuk:</td>
<td class="data">
<?= date('d F Y',strtotime($mahasiswa->tanggalmasuk)) ?>
</td>
</tr>
<tr>
<td class="field">Pembimbing Akademik:</td>
<td class="data"><?= $mahasiswa->dosen_pa->nama ?></td>
</tr>
</tr>
</table>
...
```

Temukan dan perbaiki *error* yang terjadi agar data mahasiswa bisa tampil seperti pada **Gambar 6**

Mahasiswa	
NIM:	51017006
Nama Lengkap:	DEVI CHUNGIARTO
Jenis Kelamin:	Perempuan
E-mail:	devichungiarto_18@kharisma.ac.id
Tanggal masuk:	18 September 2017
Pembimbing Akademik:	RENNY

Gambar 6

10. Edit file `class Dosen.php` untuk menambahkan fungsi `namaLengkap()` yang akan menghasilkan nama lengkap berikut gelar di depan dan di belakang nama dosen. Lihat kode program pada **Listing 9**

Listing 9

```
...  
  
public function namaLengkap()  
{  
    $namalengkap = $this->gelar_depan . ' '. $this->nama ;  
    if( $this->gelar_belakang != ' ' ){  
        $namalengkap .= ', ' . $this->gelar_belakang;  
    }  
    return $namalengkap;  
}  
  
...
```

Pertanyaan

2. Jelaskan fungsi dari baris-baris program berikut ini, dan mengapa harus ada

```
if( $this->gelar_belakang != ' ' ){  
    $namalengkap .= ', ' . $this->gelar_belakang;  
}
```

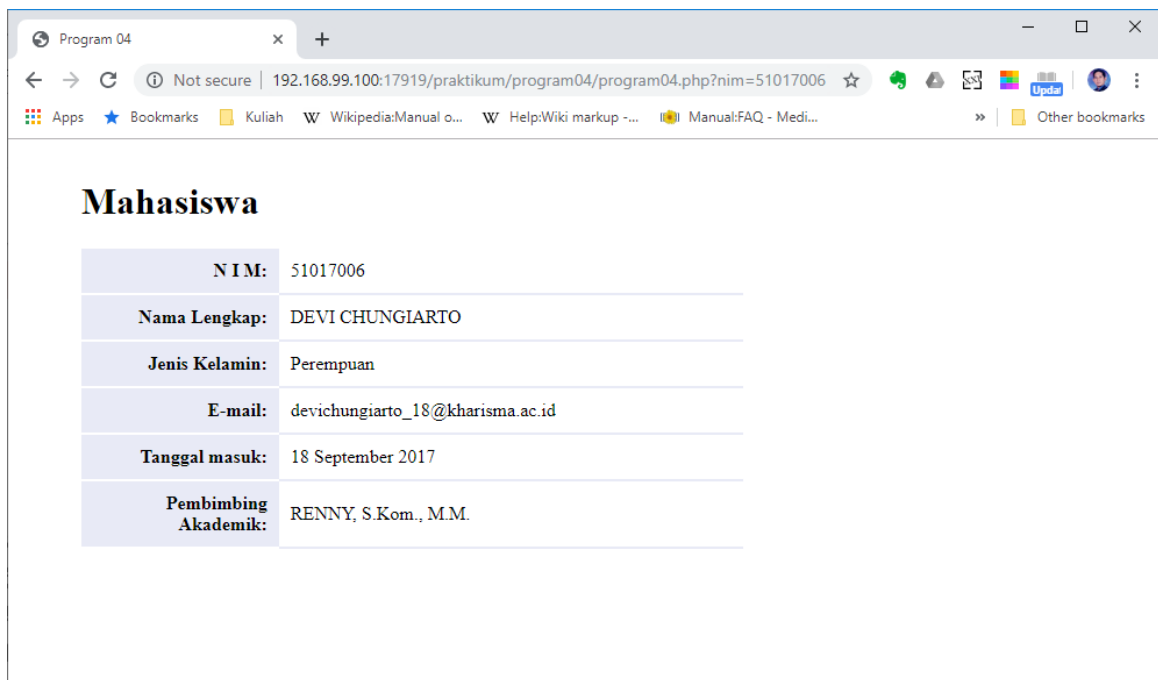
11. Modifikasi pada file `program04.php` seperti dalam **Listing 10** untuk memanggil *method / function* `namaLengkap()` yang baru dibuat, untuk menampilkan nama lengkap dosen PA mahasiswa beserta gelarnya. Lihat hasilnya pada browser (lihat **Gambar 7**)

Listing 10

```
...  
  
<h1>Mahasiswa</h1>  
<table>  
    <tr>  
        <td class="field">N I M:</td>  
        <td class="data"><?= $mahasiswa->nim ?></td>  
    </tr>  
    <tr>  
        <td class="field">Nama Lengkap:</td>  
        <td class="data"><?= $mahasiswa->nama ?></td>  
    </tr>  
    <tr>  
        <td class="field">Jenis Kelamin:</td>  
        <td class="data"><?= $mahasiswa->jk['gender'] ?></td>  
    </tr>  
</table>
```

```
<td class="field">E-mail:</td>
<td class="data"><?= $mahasiswa->email ?></td>
</tr>
<tr>
<td class="field">Tanggal masuk:</td>
<td class="data">
<?= date('d F Y',strtotime($mahasiswa->tanggalmasuk)) ?>
</td>
<tr>
<td class="field">Pembimbing Akademik:</td>
<td class="data"><?= $mahasiswa->dosen_pa->namaLengkap() ?></td>

</tr>
</tr>
</table>
...
```



Gambar 7

D. TUGAS SELANJUTNYA

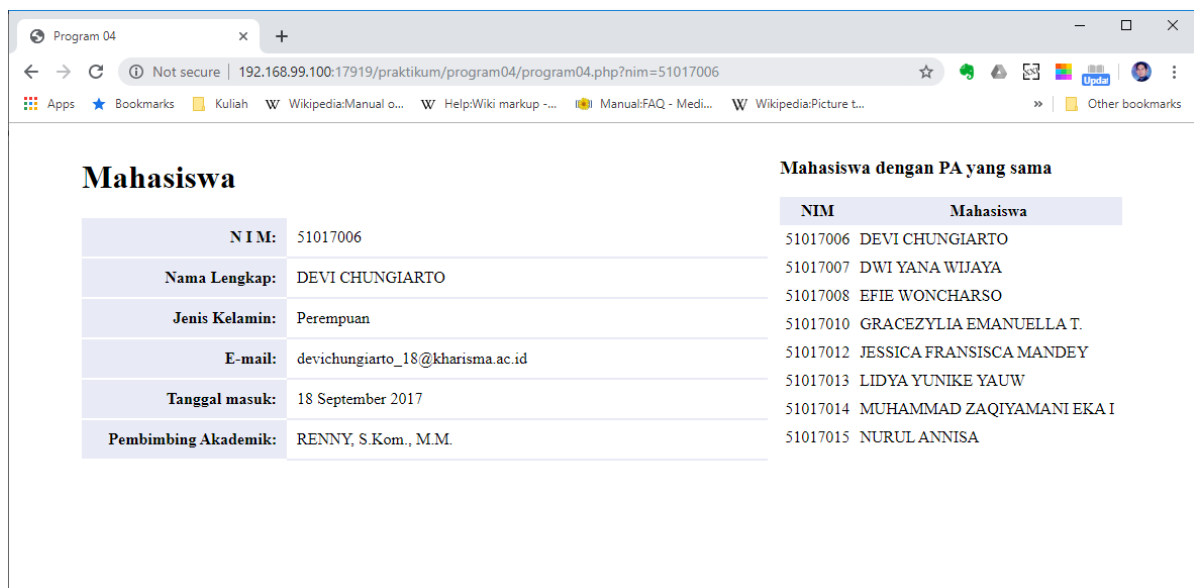
Perhatikan bahwa di dalam *class* `Daftar_mahasiswa` (file `Daftar_mahasiswa.php`) terdapat *method* berikut ini:

```
...

/**
 * fungsi pencarian daftar mahasiswa berdasarkan NIDN Dosen PA
 * @param $nidn_pa Nomor Induk Dosen PA
 * @return array dari objek mahasiswa dari PA yang sama
 */
public function getMahasiswaByPA($nidn_pa)
{
    $mahasiswa = array();
    foreach ($this->daftarmahasiswa as $mhs) {
        if ( $mhs['dosen_pa'] == $nidn_pa ) {
            $mahasiswa[] = new Mahasiswa( $mhs['nimhs'] ); //objek mahasiswa dalam
                                                         //array $mahasiswa
        }
    }
    return $mahasiswa;
}

...
```

Manfaatkan *method* tersebut untuk menghasilkan daftar mahasiswa yang dosen PA-nya sama dengan mahasiswa yang ditampilkan datanya, yang ditampilkan pada kolom sebelah kiri (lihat **Gambar 8**)



The screenshot shows a web browser window with the URL `192.168.99.100:17919/praktikum/program04/program04.php?nim=51017006`. The page displays two tables. The first table, titled "Mahasiswa", shows details for a student with NIM 51017006. The second table, titled "Mahasiswa dengan PA yang sama", lists other students who share the same advisor (PA).

Mahasiswa	
NIM:	51017006
Nama Lengkap:	DEVI CHUNGIARTO
Jenis Kelamin:	Perempuan
E-mail:	devichungiarto_18@kharisma.ac.id
Tanggal masuk:	18 September 2017
Pembimbing Akademik:	RENNY, S.Kom., M.M.

Mahasiswa dengan PA yang sama	
NIM	Mahasiswa
51017006	DEVI CHUNGIARTO
51017007	DWI YANA WIJAYA
51017008	EFIE WONCHARSO
51017010	GRACEZYLLA EMANUELLA T.
51017012	JESSICA FRANSISCA MANDEY
51017013	LIDYA YUNIKE YAUW
51017014	MUHAMMAD ZAQIYAMANI EKA I
51017015	NURUL ANNISA

Gambar 8

Lihat contoh dari Praktikum 3 untuk membuat tabel daftar mahasiswa