# LAPORAN PRAKTIKUM MATA KULIAH PRAKTIKUM PEMROGAMAN WEB

# PRAKTIKUM JOBSHEET 10 "PHP – OOP"

Dosen Pengampu: Dimas Wahyu Wibowo, S.T., M.T.



Disusun oleh:

Fifi Novitasari 2241760035

SIB 2E/09

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

TAHUN 2024



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-12: PHP - OOP

Mata Kuliah Desain dan Pemrograman Web

Pengampu: Tim Ajar Desain dan Pemrograman Web

Oktober 2023

#### **Topik**

Konsep Pembuatan Web Dinamis dengan OOP

#### **Tujuan**

Mahasiswa diharapkan dapat:

- 1. Mahasiswa mampu membuat class dan object, inheritance, polymorphism, encapsulation, abstraction, interfaces, constructors and destructors, dan encapsulation and access modifier
- 2. Mahasiswa mampu membuat CRUD dengan OOP

#### **Perhatian**

Jobsheet ini harus dikerjakan step-by-step sesuai langkah-langkah praktikum yang sudah diberikan. Soal dapat dijawab langsung di dalam kolom yang disediakan dengan menggunakan PDF Editor.

#### Pendahuluan

#### OOP

Pemrograman Berorientasi Objek (OOP) adalah paradigma pemrograman yang sangat penting dalam dunia pengembangan perangkat lunak. Ini memungkinkan para pengembang untuk mengorganisasi kode mereka menjadi objek-objek yang memiliki atribut (data) dan metode (fungsi) yang terkait.

#### Pengenalan Pemrograman Berorientasi Objek (OOP)

Pemrograman Berorientasi Objek didasarkan pada konsep objek, yang mewakili entitas dalam dunia nyata. Setiap objek memiliki karakteristik yang disebut atribut (properti), dan dapat melakukan tindakan tertentu yang disebut metode (fungsi). OOP membantu dalam memecah kode menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola.

# **Kenapa OOP Penting?**

Dalam dunia pengembangan website yang semakin kompleks dan dinamis, penggunaan Konsep Pemrograman Berorientasi Objek (OOP) telah menjadi landasan yang esensial. OOP membawa keefektifan, kemudahan pemeliharaan, dan skalabilitas yang tak ternilai harganya untuk proyek-proyek website. Artikel ini akan membahas mengapa OOP begitu penting dalam pengembangan proyek website dan manfaat utamanya.

# Modularitas dan Pengelolaan Kode yang Lebih Baik

Salah satu manfaat utama OOP adalah kemampuannya untuk memecah kode menjadi modul atau objek yang independen. Dalam pengembangan website, setiap komponen seperti formulir, tampilan, database, dan lainnya dapat diwakili sebagai objek yang terpisah. Ini memungkinkan tim pengembangan untuk bekerja secara terpisah pada komponen-komponen ini, mempercepat proses pengembangan dan memungkinkan pemeliharaan yang lebih mudah di masa depan.

# Penggunaan Ulang (Reusability) dan Efisiensi

Dalam OOP, objek-objek dapat digunakan ulang di berbagai bagian proyek. Ini mengurangi jumlah kode yang perlu ditulis, menghemat waktu dan usaha pengembang. Misalnya, jika Anda telah membuat objek "Formulir" yang memiliki metode untuk memvalidasi input, Anda dapat menggunakannya di berbagai halaman website tanpa perlu menulis ulang kode validasi tersebut.

# Pengelolaan Kesalahan yang Lebih Baik

Ketika terjadi kesalahan dalam kode OOP, Anda dapat dengan mudah mengisolasi dan menemukan sumber kesalahan tersebut karena setiap objek memiliki tanggung jawab yang jelas. Ini memungkinkan Anda untuk memperbaiki masalah lebih cepat dan lebih akurat, mengurangi waktu yang dihabiskan untuk debugging.

# Skalabilitas dan Pengembangan Kolaboratif

Proyek website cenderung berkembang seiring waktu. Dengan OOP, Anda dapat dengan mudah menambahkan fitur baru atau memperbarui komponen yang ada tanpa mengganggu fungsi lainnya. Tim pengembangan juga dapat bekerja secara paralel pada berbagai komponen, karena setiap objek berdiri sendiri dan tidak terlalu bergantung pada yang lain.

#### Pemeliharaan Lebih Mudah

Ketika proyek website tumbuh, pemeliharaan menjadi sangat penting. OOP membantu dalam memisahkan perubahan yang diperlukan pada suatu komponen tanpa mempengaruhi yang lain. Jika Anda ingin mengubah tampilan halaman tertentu, Anda hanya perlu mengedit objek tampilan tanpa perlu khawatir tentang dampaknya pada komponen lain.

#### Enkapsulasi dan Keamanan

Konsep enkapsulasi dalam OOP memungkinkan Anda untuk menyembunyikan detail implementasi dari komponen lainnya. Ini berarti bahwa komponen lain hanya dapat berinteraksi dengan objek melalui antarmuka yang ditentukan, mengurangi potensi kesalahan atau manipulasi yang tidak diinginkan.

# Fleksibilitas dan Peningkatan Kualitas Kode

OOP memungkinkan Anda untuk membuat abstraksi yang tinggi untuk mengelola kerumitan dan mendefinisikan pola umum. Ini meningkatkan kualitas kode karena mengikuti prinsip-prinsip yang terbukti dalam desain perangkat lunak, seperti DRY (Don't Repeat Yourself) dan SOLID (Prinsip-responsibilitas terpisah, Terbuka-Tertutup, Substitusi Liskov, Segregasi Antarmuka, Ketergantungan Inversi).

# Konsep Utama dalam OOP PHP

Dalam PHP, OOP memungkinkan Anda untuk mengorganisir dan mengelompokkan kode menjadi unit-unit yang lebih terstruktur dan mudah dikelola. Berikut adalah konsep-konsep utama OOP dalam PHP:

Praktikum 1. Basic OOP		
Langkah	Keterangan	
1	Kelas adalah blueprint atau cetak biru yang mendefinisikan struktur dan perilaku suatu objek. Kelas berisi atribut (data) dan metode (fungsi) yang berkaitan dengan objek tersebut. Objek, di sisi lain, adalah instance konkret dari suatu kelas, memiliki nilai nyata untuk atribut dan mampu menjalankan metode yang didefinisikan dalam kelas. Dalam PHP, Anda dapat membuat kelas dengan kata kunci class dan kemudian membuat objek dari kelas tersebut dengan kata kunci new. Berikut adalah contoh sederhana:	
2	Buatlah folder oop dalam folder dasarWeb/ dengan file baru yaitu oop.php.	

Ketikkan ke dalam file oop.php tersebut kode di bawah ini.

3

5

6

```
<?php
class Car
{
    public $brand;

    public function startEngine()
    {
        echo "Engine started!";
     }
}

$car1 = new Car();
$car1->brand = "Toyota";

$car2 = new Car();
$car2->brand = "Honda";

$car1->startEngine();
echo $car2->brand;
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.1)

```
class Car
{
public $brand;
    public function startEngine()
    {
        echo "Engine started!";
    }
}
$car1 = new Car();
$car1->brand = "Toyota";

$car2 = new Car();
$car2->brand = "Honda";

$car1->startEngine();
echo "<br/>echo $car2->brand;
```

**Jawab:** Kode diatas adalah kode membuat class Car yang berisikan attribute brand dan fungsi startEngine, dimana lalu akan dibuat objek mobil1 dan mobil2, dengan car1 dengan brand Toyota dan car2 brand Honda. Kemudian car1 memanggil fungsi startEngine dan akan menghasilkan output Engine started!Honda.

#### Engine started!Honda

Inheritance adalah salah satu konsep dasar dalam pemrograman berorientasi objek (OOP) yang memungkinkan sebuah class untuk mewarisi properti dan metode dari class lain. Class yang mewarisi disebut subclass atau child class, sedangkan class yang memberikan warisan disebut superclass atau parent class. Konsep ini memungkinkan kita untuk menggunakan kembali kode, memperpanjang fungsionalitas, dan membangun hierarki class.

Berikut ini adalah contoh sederhana penerapan inheritance dalam PHP:

```
class Animal
{
    protected $name;
    public function __construct($name)
        $this->name = $name;
    }
   public function eat()
        echo $this->name . " is eating.<br>";
    public function sleep()
        echo $this->name . " is sleeping.<br>";
class Cat extends Animal
   public function meow()
        echo $this->name . " says meow!<br>";
    }
}
class Dog extends Animal
   public function bark()
        echo $this->name . " says woof!<br>";
    }
}
$cat = new Cat("Whiskers");
$dog = new Dog("Buddy");
$cat->eat();
$dog->sleep();
$cat->meow();
$dog->bark();
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.2)

Jawab: Sama seperti praktikum sebelumnya kita membuat class Animal yang terdapat atribut nama dan fungsi eat(), sleep(), lalu kita membuat class turunan dengan nama Cat dengan menambahkan extends sebagai tanda kalua dia class turunan dari Animal, kita tidak perlu menuliskan fungsi/method eat dan sleep lagi karena akan otomatis terinclude karena merupakan kelas turunan. Whiskers is eating. Buddy is sleeping. Whiskers says meow. Buddy says woof! Polymorphism adalah konsep dalam pemrograman berorientasi objek yang memungkinkan objek dari class yang berbeda untuk merespon pada pemanggilan metode dengan cara yang sama. Ini dapat diwujudkan dalam PHP melalui penggunaan antarmuka (interface) dan penggunaan overriding metode. Dengan polymorphism, Anda dapat memperlakukan objek dari 6 class yang berbeda dengan cara yang seragam. Berikut adalah contoh sederhana penggunaan polymorphism dalam PHP menggunakan antarmuka:

```
interface Shape
    public function calculateArea();
class Circle implements Shape
    private $radius;
    public function __construct($radius)
    {
        $this->radius = $radius;
    public function calculateArea()
        return pi() * pow($this->radius, 2);
    }
class Rectangle implements Shape
    private $width;
   private $height;
    public function __construct($width, $height)
    {
        $this->width = $width;
        $this->height = $height;
    public function calculateArea()
        return $this->width * $this->height;
function printArea(Shape $shape)
{
    echo "Area: " . $shape->calculateArea() . "<br>";
$circle = new Circle(5);
$rectangle = new Rectangle(4, 6);
printArea($circle);
printArea($rectangle);
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.3)

Jawab: Kode di atas mengimplementasikan polimorfisme dalam php, dimana membuat class interface Shape dengan berisikan fungsi calculateArea(), lalu membuat class circle dan Rectangle dengan atribut dan juga rumus untuk menghitung luasnya, dengan polymorphism kita bisa menimpa fungsi pada class Shape untuk digunakan pada class Lain, yang kemudian

kita masukkan parameter pada saat pembuatan object.

Area: 78.539816339745

Area: 24

Encapsulation adalah salah satu konsep dalam pemrograman berorientasi objek (OOP) yang mengizinkan pembungkusan (encapsulation) properti dan metode dalam sebuah class sehingga akses ke mereka dapat dikontrol. Hal ini dapat membantu dalam menerapkan prinsip-prinsip pengelolaan akses dan memastikan bahwa properti dan metode yang mungkin berubah di kemudian hari tidak merusak integritas class atau program secara keseluruhan.

Berikut adalah contoh sederhana encapsulation dalam PHP:

```
class Car
{
    private $model;
    private $color;
    public function __construct($model, $color)
        $this->model = $model;
        $this->color = $color;
    public function getModel()
        return $this->model;
    public function setColor($color)
        $this->color = $color;
    public function getColor()
        return $this->color;
$car = new Car("Toyota", "Blue");
echo "Model: " . $car->getModel() . "<br>";
echo "Color: " . $car->getColor() . "<br>";
$car->setColor("Red");
echo "Updated Color: " . $car->getColor() . "<br>";
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.4) **Jawab:** Membuat konsep enkapsulasi, kita mengumpulkan semua fungsi/method dan attribute

	yang memiliki 1 tujuan dan kegunaan yang sama ke dalam satu class yaitu pada class Car dengan atribut model dan color, serta ada geter dan seter untuk getModel getColor dan setColor, lalu kita membuat objek car dengan parameter Toyota Blue untuk model dan warna, kemudian kita panggil dengan getModel dan getColor terhadap parameter tadi, kemudian kita setColor menjadi Red untuk update color menjadi red.  Model: Toyota Color: Blue Updated Color: Red
8	Abstraction adalah salah satu konsep dasar dalam pemrograman berorientasi objek (OOP) yang memungkinkan Anda menyembunyikan detail internal dan hanya mengekspos fungsionalitas yang diperlukan. Ini membantu dalam menciptakan class dan metode yang bersifat umum dan fleksibel, memungkinkan pengguna untuk berinteraksi dengan objek tanpa perlu mengetahui implementasi internalnya.
	Berikut adalah contoh sederhana abstraksi dalam PHP menggunakan abstract class dan method:

```
abstract class Shape
    abstract public function calculateArea();
class Circle extends Shape
   private $radius;
   public function __construct($radius)
        $this->radius = $radius;
   public function calculateArea()
        return pi() * pow($this->radius, 2);
class Rectangle extends Shape
   private $width;
   private $height;
   public function __construct($width, $height)
        $this->width = $width;
        $this->height = $height;
   public function calculateArea()
        return $this->width * $this->height;
$circle = new Circle(5);
$rectangle = new Rectangle(4, 6);
echo "Area of Circle: " . $circle->calculateArea() . "<br>";
echo "Area of Rectangle: " . $rectangle->calculateArea() . "<br>";
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.5)

**Jawab:** Membuat class Abstract dengan nama Shape, yang kemudian kita isikan fungsi/method abstract juga dengan nama calculateArea(), lalu kita buat class turunan dari class Abstract tadi dengan nama Circle lalu ditambahkan extends Shape, dan kita mengisikan function yang berisikan perhitungan untuk mencari luasnya, tak lupa untuk menambahkan constructor.

Area of Circle: 78.539816339745 Area of Rectangle: 24

Interface adalah konsep dalam pemrograman berorientasi objek yang memungkinkan definisi kontrak atau kerangka yang harus diikuti oleh class-class yang mengimplementasikannya. Interface tidak memiliki implementasi sendiri, tetapi hanya menyediakan deklarasi metode dan properti yang harus diimplementasikan oleh class yang menggunakannya. Hal ini

memungkinkan untuk mencapai polimorfisme tanpa memerlukan pewarisan tunggal, sehingga sebuah class dapat mengimplementasikan beberapa interface.

Berikut adalah contoh penggunaan interface dalam PHP:

```
interface Shape
{
    public function calculateArea();
}
interface Color
    public function getColor();
class Circle implements Shape, Color
    private $radius;
    private $color;
    public function __construct($radius, $color)
        $this->radius = $radius;
        $this->color = $color;
    }
    public function calculateArea()
        return pi() * pow($this->radius, 2);
    public function getColor()
        return $this->color;
}
$circle = new Circle(5, "Blue");
echo "Area of Circle: " . $circle->calculateArea() . "<br>";
echo "Color of Circle: " . $circle->getColor() . "<br>";
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.6)

**Jawab:** Membuat interface untuk Shape dan Color yang kemudian keduanya meimplementasikan pada class Circle dengan menambahkan kode implements Shape, Color, lalu kita membuatfungsi di dalamnya yang mengimplementasikan fungsi pada interface Shape dan Color Kemudian membuat objek \$circle dengan parameter 5 untuk radiusnya dan Blue sebagaiColor lalu kita memanggil kedua fungsi tersebut.

Area of Circle: 78.539816339745

Color of Cicle: Blue

Constructor (Metode Pembuat)

Constructor menggunakan nama khusus \_\_construct dalam PHP. Constructor ini akan dipanggil secara otomatis setiap kali objek baru dibuat dari class yang mengandung constructor tersebut. Destructor (Metode Penghancur)

Destructor menggunakan nama khusus \_\_destruct dalam PHP. Destructor ini akan dipanggil secara otomatis ketika objek dihapus atau program selesai dieksekusi.

Berikut adalah contoh constructor dan destructor:

```
class Car
{
    private $brand;

    public function __construct($brand)
    {
        echo "A new car is created.<br>";
        $this->brand = $brand;
    }

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

    public function __destruct()
    {
        echo "The car is destroyed.<br>";
    }
}

$car = new Car("Toyota");
echo "Brand: " . $car->getBrand() . "<br>";
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.7)

**Jawab:** Kode di atas membuat class Car{} dengan berisikan atribut \$brand dan ada constructor(\$brand), fungsi getter yaitu getBrand untuk menadapatkan brand, dan juga destructor. Kemudian membuat object baru dengan variable \$car dengan parameter(Toyota) sebagai brandnya.

A new car is created.

Brand: Toyota

The car is destroyed.

Karena meletakkan fungsi \_\_destruct() yang berisikan statement "The car is destroyed"pada kode program, maka object tadi akan dihapuskan.

# **Encapsulation and Access Modifiers**

Encapsulation adalah salah satu konsep utama dalam pemrograman berorientasi objek (OOP), dan itu melibatkan pembungkusan data (variabel) dan metode (fungsi) dalam sebuah class. Ini membantu dalam menyembunyikan implementasi internal suatu class dan hanya mengekspos fungsionalitas yang diperlukan. Access modifiers adalah bagian dari encapsulation yang memungkinkan Anda mengontrol tingkat akses ke properti dan metode dalam sebuah class.

PHP memiliki tiga access modifiers utama yang dapat digunakan dalam class:

11 Public (public): Properti atau metode yang dideklarasikan sebagai public dapat diakses dari luar class, sehingga mereka bersifat terbuka untuk diakses dari mana saja.

Protected (protected): Properti atau metode yang dideklarasikan sebagai protected hanya dapat diakses dari dalam class itu sendiri dan dari class turunannya (inheritance).

Private (private): Properti atau metode yang dideklarasikan sebagai private hanya dapat diakses dari dalam class itu sendiri. Mereka tidak dapat diakses dari luar class, bahkan oleh class turunannya.

Berikut adalah contoh penggunaan access modifiers dalam PHP:

```
class Animal
   public $name;
   protected $age;
    private $color;
   public function __construct($name, $age, $color)
        $this->name = $name;
        $this->age = $age;
        $this->color = $color;
   public function getName()
        return $this->name;
   protected function getAge()
        return $this->age;
   private function getColor()
       return $this->color;
$animal = new Animal("Dog", 3, "Brown");
echo "Name: " . $animal->name . "<br>";
echo "Age: " . $animal->getAge() . "<br>";
echo "Color: " . $animal->getColor() . "<br>";
```

Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.8) **Jawab:** Kode di atas akan menampilkan nama, umur dan warna, kita menuliskan access modifier padagetName dan name sebagai public, age dan getAge sebagai protected, color dan getColor sebagai private

Name: Dog Age: 3

Color: Brown

# Praktikum 2. CRUD dengan OOP

Langkah	Keterangan
1	Buat file baru pada folder oop dengan nama baru bernama database.php. Ketikkan kode seperti di bawah ini.
2	<pre> <?php class Database {     private \$host = "localhost";     private \$username = "root";     private \$password = "";     private \$database = "prakwebdb";     public \$conn;      public functionconstruct()     {         \$this->conn = new mysqli(\$this-&gt;host, \$this-&gt;username, \$this-&gt;password, \$this-&gt;database);          if (\$this-&gt;conn-&gt;connect_error) {</pre>
3	Buat file baru pada folder oop dengan nama baru bernama crud.php. Ketikkan kode seperti di bawah ini.

```
require_once 'Database.php';
                 class Crud
                     private $db;
                     public function __construct()
                     public function create($jabatan, $keterangan)
                         $query = "INSERT INTO jabatan (jabatan, keterangan) VALUES $jabatan', '$keterangan')";
                         $result = $this->db->conn->query($query);
                         return $result;
                     public function read()
                         $query = "SELECT * FROM jabata ";
                        $result = $this->db->conn->query($query);
                            while ($row = $result->fetch_assoc()) {
                                $data[] = $row;
4
                         return $data;
                     public function readById($id)
                         $query = "SELECT * FROM jabatan WHERE i =$id";
                         $result = $this->db->conn->query($quedy);
                         if ($result->num_rows == 1) {
                            return $result->fetch_assoc();
                     public function update($id, $jabatan, $keterangan)
                         $query = "UPDATE jabatan SET jabata ='$jabatan', keterangan='$keterangan' WHERE i =$id";
                         $result = $this->db->conn->query($query);
                     public function delete($id)
                         $query = "DELETE FROM jabatan WHERE i =$id";
                         $result = $this->db->conn->query($qdery);
                         return $result;
```

Buat file baru pada folder oop dengan nama baru bernama index.php. Ketikkan kode seperti di bawah ini.

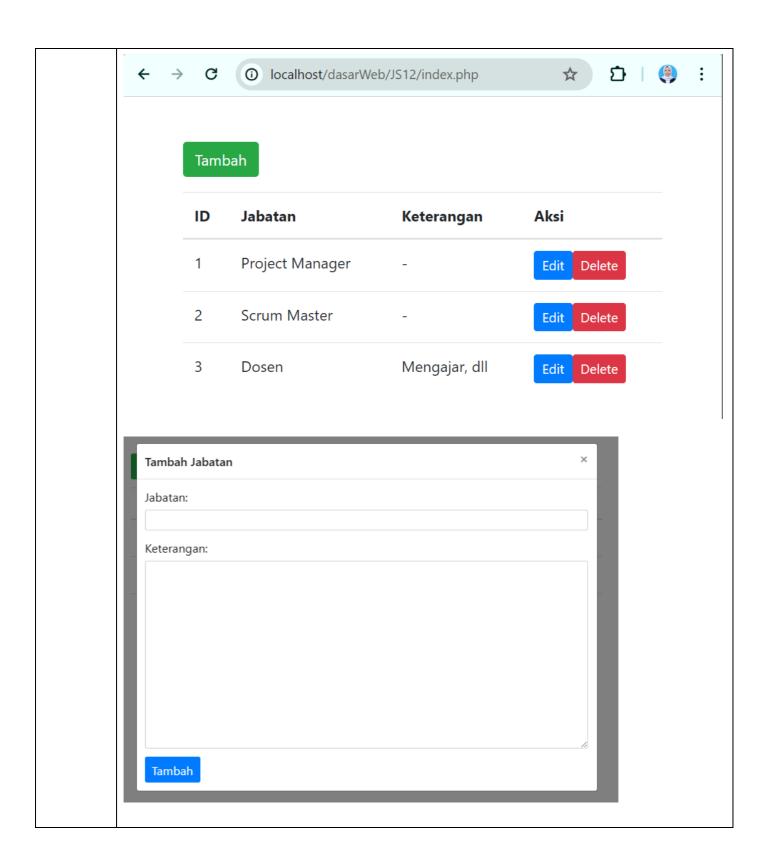
```
6
```

```
<?php
require_once 'Crud.php';</pre>
$crud = new Crud();
           $jabatan = $_POST['jabatan'];
           $keterangan = $_POST['keterangan'];
           $crud->create($jabatan, $keterangan);
if (isset($_GET['action']) && $_GET['action'] === 'delete') {
    $id = $_GET['id'];
           $crud->delete($id):
$tampil = $crud->read():
  <!DOCTYPE html>
 <html lang="en"
           <title>CRUD Jabatan</title
           Jabatan
                                                    Keterangan
                                                 each ($tamprt as \show\)
echo "" . \$show\]'id'] . "";
echo "" . \$show\]'jabatan'] . "";
echo "" . \$show\]'keterangan'] . "";
                                                  ctol '\tau',
echo "\tau href='edit.php?id=" . \$show['id'] . "' class='btn btn-primary btn-sm'>Edit</a> ";
echo "\tau href='index.php?action=delete&id=" . \$show['id'] . "' class='btn btn-danger btn-sm'>Delete</a>";
echo "\tau';
           <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></
```

```
7
```

```
$tampil = $crud->readById($id);
            $jabatan = $_POST['jabatan'];
$keterangan = $_POST['keterangan'];
            $crud->update($id, $jabatan, $keterangan);
<html lang="en"
              <meta charset="UTF-8">
            <title>Edit Jabatan</title>
<tink rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
                       <form method="post" action="">
                                          <label for="keterangan">Keterangan:</label>
 <input type="hidden" name="id" value="<?php echo $tampil['id']; ?>">
<button type="submit" class="btn btn-primary">Update</button>
             <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></
```

Jalankan code pada praktikum 2. Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 2.1)



Dosen	
Keterangan:	
Mengajar, dll	
	le
Update	

Database digunakan untuk koneksi ke database prakwebdb, edit untuk mengedit.