

LAPORAN

TUGAS JOBSHEET XII

“PHP – OOP Mata Kuliah Desain dan Pemrograman Web ”

Mata Kuliah: Pemrograman Web

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



Disusun Oleh:

Nama : Syava Aprilia P

NIM : 2241760129

Absen : 25

**PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023**



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.

3	Ketikkan ke dalam file oop.php tersebut kode di bawah ini.
4	<pre> <?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; </pre>
5	<p>Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.1)</p> <p>Jawab:</p> <p>Kode tersebut adalah implementasi sederhana dari sebuah kelas Car dalam PHP. Kelas tersebut memiliki satu properti publik untuk merek mobil (\$brand) dan satu metode public (startEngine()) yang mencetak pesan "Engine started!".</p> <p>Objek dari kelas Car dibuat, masing-masing memiliki merek mobil yang berbeda. Metode startEngine() dipanggil pada objek pertama untuk menyalakan mesin, dan merek mobil dari objek kedua dicetak ke output.</p>
6	<p>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.</p> <p>Berikut ini adalah contoh sederhana penerapan inheritance dalam PHP:</p>

```

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:

Program tersebut menggunakan konsep pewarisan (inheritance) dalam pemrograman berorientasi objek dengan PHP. Ada tiga kelas yang terlibat: Animal, Cat, dan Dog.

- Kelas Animal memiliki properti \$name dan metode eat() serta sleep().
- Kelas Cat dan Dog mewarisi sifat dan perilaku dari Animal.
- Setiap kelas anak memiliki metode tambahan: meow() untuk Cat dan bark() untuk Dog, masing-masing mencetak suara hewan.

Objek Cat dan Dog diciptakan dengan nama dan tindakan yang sesuai dengan jenis hewan. Program mencetak apa yang hewan itu lakukan: makan, tidur, dan suara khasnya (meong untuk kucing, gonggong untuk anjing).

6

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 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:

Program menggunakan konsep antarmuka (interface) di PHP. Antarmuka **Shape** didefinisikan dengan satu metode, **calculateArea()**. Ada dua kelas, **Circle** dan **Rectangle**, yang mengimplementasikan antarmuka **Shape**.

- **Circle** memiliki properti radius dan menghitung luas lingkaran.
- **Rectangle** memiliki properti panjang dan lebar, menghitung luas persegi panjang.

Ada juga fungsi **printArea()** yang menerima objek yang mengimplementasikan **Shape**, dan mencetak luas area dari objek tersebut. Objek **Circle** dan **Rectangle** diciptakan,

ukurannya disesuaikan, dan kemudian fungsi **printArea()** dipanggil untuk mencetak luas area masing-masing objek.

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:

Program tersebut mendefinisikan kelas Car dengan dua properti private, yaitu \$model dan \$color. Terdapat konstruktor (_construct()) yang akan menginisialisasi nilai properti tersebut saat objek Car dibuat.

Kelas Car juga memiliki beberapa metode:

- getModel(): Mengembalikan nilai properti \$model.
- setColor(\$color): Mengubah nilai properti \$color menjadi nilai baru yang diberikan.
- getColor(): Mengembalikan nilai properti \$color.

	Objek \$car dibuat dengan merek "Toyota" dan warna "Blue". Kemudian, informasi model dan warna dicetak ke layar. Setelah itu, warna mobil diubah menjadi "Red" menggunakan metode setColor(), dan warna yang diperbarui dicetak ulang ke layar.
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

	<p>fleksibel, memungkinkan pengguna untuk berinteraksi dengan objek tanpa perlu mengetahui implementasi internalnya.</p> <p>Berikut adalah contoh sederhana abstraksi dalam PHP menggunakan abstract class dan method:</p> <pre> 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() . "
"; echo "Area of Rectangle: " . \$rectangle->calculateArea() . "
"; </pre> <p>Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.5)</p> <p>Jawab:</p> <p>Program di atas mendefinisikan sebuah kelas abstrak Shape dengan metode abstrak calculateArea(). Kemudian, terdapat dua kelas turunan (Circle dan Rectangle) yang mengimplementasikan metode calculateArea().</p>
--	--

	<p>Kelas Circle memiliki properti radius, dan mengimplementasikan metode calculateArea() untuk menghitung luas lingkaran dengan formula πr^2.</p> <p>Kelas Rectangle memiliki properti panjang dan lebar, dan mengimplementasikan metode calculateArea() untuk menghitung luas persegi panjang dengan formula panjang x lebar.</p> <p>Pada bagian akhir, objek \$circle dan \$rectangle dibuat menggunakan nilai yang sesuai (radius 5 untuk lingkaran, panjang 4 dan lebar 6 untuk persegi panjang). Kemudian, luas area dari masing-masing bentuk geometri dicetak ke layar.</p>
9	<p>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.</p> <p>Berikut adalah contoh penggunaan interface dalam PHP:</p>

	<pre> 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() . "
"; echo "Color of Circle: " . \$circle->getColor() . "
"; </pre> <p>Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.6)</p> <p>Jawab:</p> <p>Program di atas mendefinisikan dua antarmuka (interface), yaitu Shape dan Color. Antarmuka Shape memiliki satu metode calculateArea(), sedangkan antarmuka Color memiliki satu metode getColor().</p>
--	--

	<p>Kelas Circle mengimplementasikan kedua antarmuka Shape dan Color. Kelas ini memiliki properti private \$radius dan \$color, yang diinisialisasi melalui konstruktor. Kelas Circle juga memiliki metode calculateArea() untuk menghitung luas lingkaran dengan formula πr^2, dan metode getColor() untuk mengembalikan nilai warna lingkaran.</p> <p>Objek \$circle dibuat dengan radius 5 dan warna "Blue". Kemudian, luas area lingkaran dan warna lingkaran dicetak ke layar.</p> <p>Program ini menunjukkan penggunaan antarmuka untuk memperkenalkan kontrak metode yang harus diimplementasikan oleh kelas yang menggunakannya. Dalam hal ini, Circle harus mengimplementasikan metode calculateArea() dari antarmuka Shape dan metode getColor() dari antarmuka Color.</p>
10	<p>Constructors dan destructors adalah metode khusus dalam pemrograman berorientasi objek (OOP) yang digunakan dalam PHP untuk menginisialisasi dan membersihkan objek. Constructor adalah metode yang dipanggil secara otomatis ketika objek baru dibuat, sedangkan destructor adalah metode yang dipanggil secara otomatis ketika objek dihapus atau tidak lagi digunakan.</p> <p>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.</p> <p>Destructor (Metode Penghancur) Destructor menggunakan nama khusus __destruct dalam PHP. Destructor ini akan dipanggil secara otomatis ketika objek dihapus atau program selesai dieksekusi.</p> <p>Berikut adalah contoh constructor dan destructor:</p> <pre> class Car { private \$brand; public function __construct(\$brand) { echo "A new car is created.
"; \$this->brand = \$brand; } public function getBrand() { return \$this->brand; } public function __destruct() { echo "The car is destroyed.
"; } } \$car = new Car("Toyota"); echo "Brand: " . \$car->getBrand() . "
"; </pre> <p>Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 1.7)</p> <p>Jawab:</p> <p>Program tersebut mendefinisikan sebuah kelas Car dengan properti private \$brand. Terdapat sebuah konstruktor (__construct()) yang dipanggil saat objek Car dibuat. Konstruktor ini</p>

	<p>mencetak pesan "A new car is created." ke layar dan mengatur nilai properti \$brand sesuai dengan parameter yang diberikan.</p> <p>Kelas Car juga memiliki metode getBrand() yang mengembalikan nilai properti \$brand.</p> <p>Selain itu, terdapat sebuah metode destruct() yang dipanggil saat objek Car dihancurkan (destructed). Metode ini mencetak pesan "The car is destroyed." ke layar.</p> <p>Objek \$car dibuat dengan merek "Toyota", dan kemudian merek mobil tersebut dicetak ke layar menggunakan metode getBrand().</p>
11	<p>Encapsulation and Access Modifiers</p> <p>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.</p> <p>PHP memiliki tiga access modifiers utama yang dapat digunakan dalam class:</p> <p>Public (public): Properti atau metode yang dideklarasikan sebagai public dapat diakses dari luar class, sehingga mereka bersifat terbuka untuk diakses dari mana saja.</p> <p>Protected (protected): Properti atau metode yang dideklarasikan sebagai protected hanya dapat diakses dari dalam class itu sendiri dan dari class turunannya (inheritance).</p> <p>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.</p> <p>Berikut adalah contoh penggunaan access modifiers dalam PHP:</p>

```

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:

Program di atas mendefinisikan sebuah kelas Animal dengan tiga properti: \$name (publik), \$age (terlindung), dan \$color (pribadi). Terdapat konstruktor (__construct()) yang digunakan untuk menginisialisasi nilai properti saat objek Animal dibuat.

Kelas Animal memiliki tiga metode:

getName(): Mengembalikan nilai properti \$name.

getAge(): Mengembalikan nilai properti \$age.

getColor(): Mengembalikan nilai properti \$color.

Objek \$animal dibuat dengan nama "Dog", usia 3 tahun, dan warna "Brown". Kemudian, informasi nama dan usia hewan dicetak langsung menggunakan properti publik \$name dan metode getAge() yang terlindung. Namun, saat mencetak warna, terjadi kesalahan karena properti \$color adalah pribadi, sehingga mencoba mengaksesnya dari luar kelas akan menghasilkan kesalahan.

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 function __construct() { \$this->conn = new mysqli(\$this->host, \$this->username, \$this->password, \$this->database); if (\$this->conn->connect_error) { die("Connection failed: " . \$this->conn->connect_error); } } } </pre>
3	<p>Buat file baru pada folder <code>oop</code> dengan nama baru bernama <code>crud.php</code>. Ketikkan kode seperti di bawah ini.</p>

4

```

<?php
require_once 'Database.php';

class Crud
{
    private $db;

    public function __construct()
    {
        $this->db = new Database();
    }

    // Create
    public function create($jabatan, $keterangan)
    {
        $query = "INSERT INTO jabatan (jabatan, keterangan) VALUES ($jabatan, '$keterangan')";
        $result = $this->db->conn->query($query);

        return $result;
    }

    // Read
    public function read()
    {
        $query = "SELECT * FROM jabata ";
        $result = $this->db->conn->query($query);

        $data = [];
        if ($result->num_rows > 0) {
            while ($row = $result->fetch_assoc()) {
                $data[] = $row;
            }
        }

        return $data;
    }

    // Read By Id
    public function readById($id)
    {
        $query = "SELECT * FROM jabatan WHERE i =$id";
        $result = $this->db->conn->query($query);
        if ($result->num_rows == 1) {
            return $result->fetch_assoc();
        } else {
            return null;
        }
    }

    // Update
    public function update($id, $jabatan, $keterangan)
    {
        $query = "UPDATE jabatan SET jabata ='$jabatan', keterangan='$keterangan' WHERE i =$id";
        $result = $this->db->conn->query($query);

        return $result;
    }

    // Delete
    public function delete($id)
    {
        $query = "DELETE FROM jabatan WHERE i =$id";
        $result = $this->db->conn->query($query);

        return $result;
    }
}

```

5

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

```

<?php
require_once 'Crud.php';

$crud = new Crud();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $jabatan = $_POST['jabatan'];
    $keterangan = $_POST['keterangan'];
    $crud->create($jabatan, $keterangan);
}

if (isset($_GET['action']) && $_GET['action'] === 'delete') {
    $id = $_GET['id'];
    $crud->delete($id);
}

$stampil = $crud->read();
?>

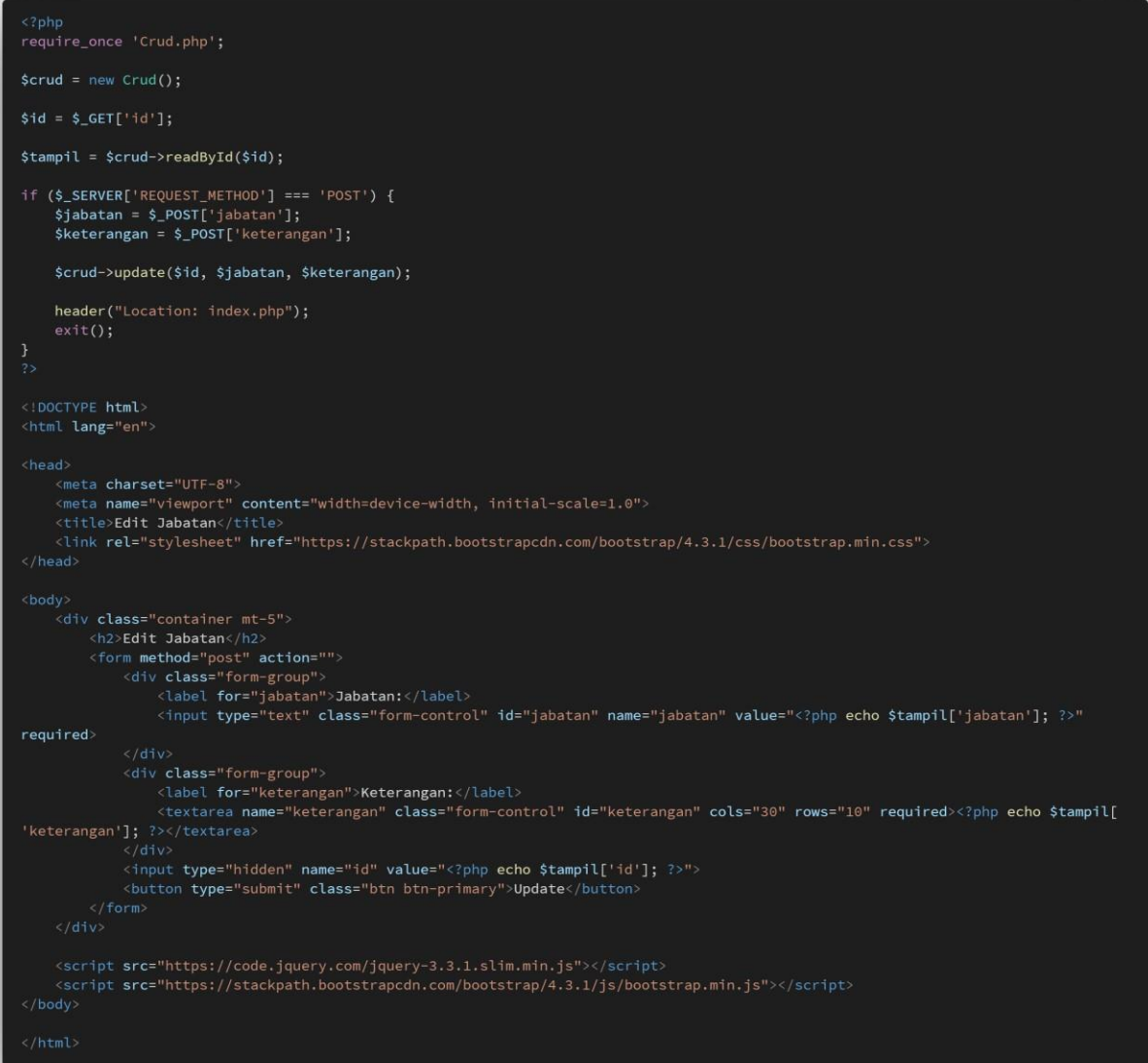
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CRUD Jabatan</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
</head>

<body>
    <div class="container mt-5">
        <button type="button" class="btn btn-success mb-3" data-toggle="modal" data-target="#tambahModal">Tambah</button>
        <table class="table">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Jabatan</th>
                    <th>Keterangan</th>
                    <th>Aksi</th>
                </tr>
            </thead>
            <tbody>
                <?php
                foreach ($stampil as $show) {
                    echo "<tr>";
                    echo "<td>" . $show['id'] . "</td>";
                    echo "<td>" . $show['jabatan'] . "</td>";
                    echo "<td>" . $show['keterangan'] . "</td>";
                    echo "<td>";
                    echo "<a href='edit.php?id=" . $show['id'] . "' class='btn btn-primary btn-sm'>Edit</a> ";
                    echo "<a href='index.php?action=delete&id=" . $show['id'] . "' class='btn btn-danger btn-sm'>Delete</a>";
                    echo "</td>";
                    echo "</tr>";
                }
                ?>
            </tbody>
        </table>
    </div>
    <div class="modal fade" id="tambahModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModalLabel">Tambah Data Jabatan</h5>
                    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">
                    <form method="post" action="">
                        <div class="form-group">
                            <label for="name">Jabatan:</label>
                            <input type="text" class="form-control" id="jabatan" name="jabatan" required>
                        </div>
                        <div class="form-group">
                            <label for="email">Keterangan:</label>
                            <textarea name="keterangan" class="form-control" id="keterangan" cols="30" rows="10" required></textarea>
                        </div>
                        <button type="submit" class="btn btn-primary">Tambah</button>
                    </form>
                </div>
            </div>
        </div>
    </div>

    <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>
</body>
</html>

```


7	Buat file baru pada folder oop dengan nama baru bernama edit.php. Ketikkan kode seperti di bawah ini.
8	 <pre> <?php require_once 'Crud.php'; \$crud = new Crud(); \$id = \$_GET['id']; \$stampil = \$crud->readById(\$id); if (\$_SERVER['REQUEST_METHOD'] === 'POST') { \$jabatan = \$_POST['jabatan']; \$keterangan = \$_POST['keterangan']; \$crud->update(\$id, \$jabatan, \$keterangan); header("Location: index.php"); exit(); } ?> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Edit Jabatan</title> <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"> </head> <body> <div class="container mt-5"> <h2>Edit Jabatan</h2> <form method="post" action=""> <div class="form-group"> <label for="jabatan">Jabatan:</label> <input type="text" class="form-control" id="jabatan" name="jabatan" value="<?php echo \$stampil['jabatan']; ?>" required> </div> <div class="form-group"> <label for="keterangan">Keterangan:</label> <textarea name="keterangan" class="form-control" id="keterangan" cols="30" rows="10" required><?php echo \$stampil['keterangan']; ?></textarea> </div> <input type="hidden" name="id" value="<?php echo \$stampil['id']; ?>"> <button type="submit" class="btn btn-primary">Update</button> </form> </div> <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> </body> </html> </pre> <p>Jalankan code pada praktikum 2. Apa yang anda pahami dari code di atas. Catat di bawah ini pemahaman anda. (soal no 2.1)</p> <p>Jawab:</p> <p>Kode tersebut adalah halaman untuk mengedit data jabatan. Saat halaman dimuat, data jabatan dengan ID tertentu dibaca dari database dan ditampilkan di dalam form. Saat form disubmit, data jabatan diupdate berdasarkan inputan baru, dan pengguna dialihkan kembali ke halaman utama.</p> <p>Berikut hasilnya:</p>

Tambah

ID	Jabatan	Keterangan	Aksi
1	Dosen	Bahasa Kewarganegaraan	Edit Delete
3	Mahasiswa	Sistem Informasi Bisnis	Edit Delete

Jika ingin mengubah data maka klik edit, lalu hasilnya seperti berikut:

Edit Jabatan

Jabatan :

Dosen

Keterangan :

Bahasa Indonesia

Update

Setelah pengeditan Dosen:

Tambah

ID	Jabatan	Keterangan	Aksi
1	Dosen	Bahasa Indonesia	Edit Delete
3	Mahasiswa	Sistem Informasi Bisnis	Edit Delete

Saya mencoba menambahkan data Jabatan Ketua HMTI:

Tambah

ID	Jabatan	Keterangan	Aksi
1	Dosen		Edit Delete
3	Mahasiswa		Edit Delete

Tambah Data Jabatan

Jabatan :< /label>

Ketua

Keterangan :< /label>

HMTI

Tambah

Setelah berhasil ditambahkan Jabatan Ketua:

Tambah

ID	Jabatan	Keterangan	Aksi	
1	Dosen	Bahasa Indonesia	Edit	Delete
3	Mahasiswa	Sistem Informasi Bisnis	Edit	Delete
4	Ketua	HMTI	Edit	Delete

Saya mencoba hapus data Jabatan Dosen:

Tambah

ID	Jabatan	Keterangan	Aksi	
3	Mahasiswa	Sistem Informasi Bisnis	Edit	Delete
4	Ketua	HMTI	Edit	Delete