

**Laporan Mata Kuliah Pemograman Web
JOBSHEET 12 PHP – OOP**



Oleh:

Suhatta (2241760060)

Kelas SIB-2E / 23

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

65141

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><u>Engine started!Honda</u></p> <p>membuat class Car yang berisikan attribute brand dan fungsi startEngine, dan membuat objek mobil1 dan mobil2, dan car1 dengan brand Toyota dan car 2 brand honda, kemudian car1 memanggil fungsi startEngine dan akan menghasilkan hasil.</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)

	<p>Whiskers is eating. Buddy is sleeping. Whiskers says meow! Buddy says woof!</p> <p>membuat class Animal yang terdapat atribut nama dan fungsi eat(), sleep(), lalu kita membuat class turunan dengan nama Cat dengan menambahkan extends sebagai tanda kalau dia class turunan dari animal</p>
6	<p>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.</p> <p>Berikut adalah contoh sederhana penggunaan polymorphism dalam PHP menggunakan antarmuka:</p>

```

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)

	<hr/> <p>Area: 78.539816339745 Area: 24</p> <p>dimana kita membuat class interface Shape dengan berisikan fungsi calculateArea(), lalu kita membuat class Circle dan Rectangle dengan atribut dan juga rumus untuk menghitung</p>
7	<p>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.</p> <p>Berikut adalah contoh sederhana encapsulation dalam PHP:</p>

```

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)

Model: Toyota

Color: Blue

Updated Color: Red

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

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)

Area of Circle: 78.539816339745

Area of Rectangle: 24

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

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)

Area of Circle: 78.539816339745
Area of Rectangle: Blue

membuat fungsi di dalamnya yang mengimplementasikan fungsi pada interface Shape dan Color Kemudian kita membuat objek \$circle dengan parameter 5 untuk radiusnya dan Blue sebagai Color lalu kita memanggil kedua fungsi tersebut.

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.

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)

	<pre> A new car is created. Brand: Toyota The car is destroyed. </pre> <p>fungsi getter getBrand untuk mendapatkan brand, dan juga destructor. Kemudian kita membuat object baru dengan variable \$car dengan parameter(Toyota) sebagai brandnya. Karena kita meletakkan fungsi destruct() yang berisikan statement “The car is destroyed” pada kode program, maka object tadi akan dihapuskan</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)

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 data-bbox="427 212 1519 651"><?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 data-bbox="368 683 1581 757">Buat file baru pada folder oop dengan nama baru bernama crud.php. Ketikkan kode seperti di bawah ini.</p>

4	<pre> <?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; } } </pre>
5	<p>Buat file baru pada folder oop dengan nama baru bernama index.php. Ketikkan kode seperti di bawah ini.</p>

```

<?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>

2

Tambah

ID	Jabatan	Keterangan	Aksi
3	Manager		Edit Delete

Edit Jabatan

Jabatan:

Karyawan

Keterangan:

Update

Dari kode di atas kita membuat sama seperti project sebelumnya, namun dengan konsep OOP. Database digunakan untuk koneksi ke database prakwebdb, edit untuk mengedi