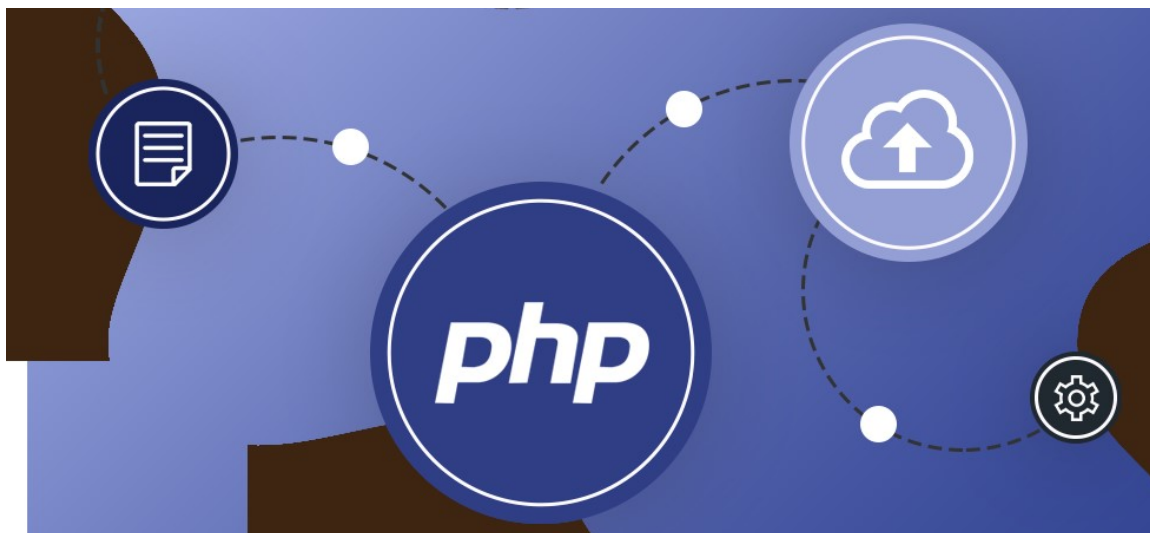


PEMROGRAMAN BERBASIS WEB 2

PERTEMUAN 4



DESKRIPSI MATAKULIAH

Mahasiswa mampu mengembangkan aplikasi web dinamis menggunakan PHP dan framework CodeIgniter dengan konsep MVC. Mahasiswa mempelajari implementasi CRUD, autentikasi, pengelolaan database, serta integrasi Bootstrap untuk tampilan responsif. Melalui pendekatan berbasis proyek, mahasiswa diharapkan mampu merancang dan membangun aplikasi web secara utuh dan profesional.

CAPAIAN PEMBELAJARAN

DOSEN PENGAMPU

Taufikqurrohman

INSTITUT WIDYA PRATAMA

IWIMA

KONSEP OOP

CAPAIAN PEMBELAJARAN

Mahasiswa dapat memahami konsep object oriented programming pada PHP

MATERI PEMBELAJARAN

- Class dan Object
- Method Constructor dan Destructor
- Encapsulation
- Inheritance
- Praktikum

MATERI

Pengertian OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. OOP lebih sering kita dengar dan didukung pada pemrograman JAVA, tetapi sekarang konsep OOP sudah dapat digunakan dalam pemrograman PHP.

1. Class dan Object

Class adalah blueprint atau cetak biru yang mendefinisikan struktur dan perilaku suatu objek. Kelas berisi atribut (data) dan metode (fungsi) yang berkaitan dengan objek tersebut.

```
<?php
class Fruit {
    // Properties
    public $name;
    public $color;
    // Methods
    function set_name($name) {
        $this->name = $name; /* $this merupakan built in variable
        yang digunakan untuk memanggil properti atau method
        yang ada dalam class tersebut.*/
    }
    function get_name() {
        return $this->name;
    }
}
?>
```

Dalam sebuah class didalamnya dapat berisi properties atau method, function yang berada dalam class disebut method. Method merupakan semua proses yang berkaitan dengan objek.

Jika disimpulkan class bisa juga dapat diartikan sebagai berikut:

- Class adalah penampung sekumpulan elemen data (variable) dan kode program (function) yang digunakan oleh pengolahan datanya. Sifat ini disebut sebagai enkapsulasi (Encapsulation) .
- Class disusun berdasarkan karakteristik sebuah objek benda. Sifat ini disebut sebagai Abstraksi (Abstraction)
- Class dapat disusun secara hierarki sehingga suatu class dapat mewariskan beberapa atau semua karakteristiknya ke class lain (class anak/child class). Sifat ini disebut sebagai Pewarisan (Inheritance)

Aturan pembuatan class, diantaranya:

- Definisi suatu class dimulai dengan keyword Class, diikuti dengan nama classnya.
- Isi suatu class diapit dengan kurung kurawal ({ }).
- Umumnya isi class diawali dengan menuliskan properti-propertinya dan diikuti dengan method-methodnya.

Berdasarkan sifat abstraksi, suatu class menyimpan karakteristik suatu benda dalam suatu variable. Variabel ini sering disebut sebagai property dari Class, sedangkan Method biasa dalam bentuk fungsi. Pada Class person kita akan menambahkan sebuah properti baru dengan nama name, pada konsep OOP dalam bahasa pemograman PHP untuk menyatakan sebuah properti bisa dengan menggunakan keyword var lalu diikuti dengan nama propertinya seperti dibawah ini :

```
<?php
class rumah{
    // properties / variabel / atribut
    var $pemilik;
    var $warna;
    var $tipe;
}
?>
```

Method adalah suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object. untuk membuat method bisa dengan dimulai dengan keyword function lalu di ikuti oleh nama function nya. berikut ini adalah contoh method Setter dan Getter untuk memberikan value kepada properti dan mengakses properti tersebut :

```
<?php
class rumah{
    // properties / variabel / atribut
    var $tipe;
    var $lantai;

    // method setting properties
    function set_tipe($new_tipe){
        $this->tipe = $new_tipe;
    }

    // method ambil data properties
    function get_tipe(){
        return $this->tipe;
    }
}
?>
```

Object adalah instance kelas atau bentuk konkret dari suatu kelas yang memiliki nilai nyata untuk atribut dan mampu menjalankan metode yang didefinisikan dalam kelas serta dapat membuat beberapa objek dari sebuah kelas.

```
$apple = new Fruit();
$apple->set_name('Apple');
$apple->set_color('Red');
```

Contoh

Asumsikan Buah adalah nama kelas yang akan dibuat. Buah dapat memiliki properti seperti nama, warna, berat, dll. maka dapat didefinisikan variabel seperti \$nama, \$warna, dan \$berat untuk menyimpan nilai properti ini.

Ketika objek individu (apel, pisang, dll.) dibuat, objek tersebut mewarisi semua properti dan perilaku dari kelas, tetapi setiap objek akan memiliki nilai properti yang berbeda.

```
<?php
class Fruit {
    // Properties
    public $name;
    public $color;

    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
    function set_color($color) {
        $this->color = $color;
    }
    function get_color() {
        return $this->color;
    }
}

$apple = new Fruit();
$apple->set_name('Apple');
$apple->set_color('Red');
echo "Name: " . $apple->get_name();
echo "<br>";
echo "Color: " . $apple->get_color();
?>
```

Dari contoh diatas Class Fruit Mempunyai property name dan color, Properti pada Class dapat diartikan sifat atau informasi yang melekat pada sebuah obyek.

2. Method Constructor dan Destructor

Constructor adalah fungsi khusus / fungsi unik yang akan di panggil secara otomatis pada saat object dibuat. Secara umum, pada saat pembuatan object, class harus meng-inisialisasikan variabelnya untuk menghindari error yang tidak diinginkan sedangkan Destructor adalah method khusus yang dijalankan secara otomatis pada saat sebuah objek dihapus. Di dalam PHP, seluruh objek secara otomatis dihapus ketika halaman PHP dimana objek itu berada selesai diproses. berikut ini adalah contoh nya :

```
<?php
class rumah{
    // properties / variabel / atribut
    var $tipe;
    var $lantai;

    // method constructor
    function __constructor($tipe){
        $this->tipe = $tipe;
    }

    // method ambil data properties
    function get_tipe(){
        return $this->tipe;
    }

    // method destructor
    function __destruct(){
        echo "Tipe telah dihapus";
    }
}

// instansiasi objek (pembuatan objek)
$rumah = new rumah('21');

// ambil data properties
echo $rumah->get_tipe();

?>
```

3. Encapsulation

Encapsulation membuat pembatasan akses kepada property dan method, sehingga hanya property dan method tertentu saja yang bisa diakses dari luar class. Enkapsulasi juga dikenal dengan istilah 'information hiding'. ada 3 jenis enkapsulasi yang bisa anda manfaatkan yaitu Public, Private Protected.

a. Public

Jika menggunakan enkapsulasi jenis public pada properti atau method dalam sebuah class, itu artinya method dan propertinya bisa di akses secara bebas baik dari dalam class, dari luar class bahkan dari class turunan nya sekalipun. berikut ini adalah contoh nya :

```
<?php
class rumah{
    // properties / variabel / atribut
    public $tipe;
    public $lantai;

    // method setting properties
    public function set_tipe($tipe){
        $this->tipe = $tipe;
    }

    // method ambil data properties
    public function get_tipe(){
        return $this->tipe;
    }
}
// instansiasi objek (pembuatan objek)
$rumah = new rumah();

// properti bisa di akses secara langsung
$rumah->tipe = 45;

// method bisa di akses secara langsung
echo $rumah->get_tipe();

// method bisa di akses secara langsung
$rumah->set_tipe('21');

// method bisa di akses secara langsung
echo $rumah->get_tipe();

?>
```


b. Private

Lain halnya jika menggunakan enkapsulasi jenis private, itu artinya method dan propertinya hanya bisa di akses dari dalam class itu sendiri dan dari class turunan nya, berikut ini adalah contoh nya:

```
<?php
class rumah{
    // properties / variabel / atribut
    private $tipe;
    public $lantai;

    // method setting properties
    private function set_tipe($tipe){
        $this->tipe = $tipe;
    }

    // method ambil data properties
    public function get_tipe(){
        return $this->tipe;
    }
}
// instansiasi objek (pembuatan objek)
$rumah = new rumah();

// properti tidak bisa di akses secara langsung
$rumah->tipe = 45;

// method bisa di akses secara langsung
echo $rumah->get_tipe();

// method tidak bisa di akses secara langsung
$rumah->set_tipe('21');

// method bisa di akses secara langsung
echo $rumah->get_tipe();

?>
```

c. Protected

Lalu yang terakhir adalah enkapsulasi jenis protected, jenis enkapsulasi ini biasanya digunakan untuk melindungi informasi yang bersifat rahasia, dimana jika menggunakan protected pada method atau properti maka hanya bisa mengakses properti tersebut hanya dari dalam class tersebut, berikut contoh script nya :

```
<?php
class rumah{
    // properties / variabel / atribut
    protected $tipe;
    public $lantai;

    // method setting properties
    protected function set_tipe($tipe){
        $this->tipe = $tipe;
    }

    // method ambil data properties
    public function get_tipe(){
        return $this->tipe;
    }
}
// instansiasi objek (pembuatan objek)
$rumah = new rumah();

// properti tidak bisa di akses secara langsung
$rumah->tipe = 45;

// method bisa di akses secara langsung
echo $rumah->get_tipe();

// method tidak bisa di akses secara langsung
$rumah->set_tipe('21');

// method bisa di akses secara langsung
echo $rumah->get_tipe();

?>
```

4. Inheritance

Inheritance adalah konsep di mana sebuah class dapat mewarisi properti dan metode dari class lain. Class yang mewarisi disebut sebagai subclass atau class turunan, sedangkan class yang diwarisi disebut sebagai superclass.

```
// kelas hewan adalah kelas yang diturunkan
// kelas parent
class Hewan {
    function makan() {
        return "Hewan ini makan";
    }
}

// extends key untuk membuat kelas turunan / kelas child
// yaitu kelas kucing
class Kucing extends Hewan {
    function bersuara() {
        return "Meong";
    }
}

// kucingPersia adalah objek dari kelas kucing
$KucingPersia = new Kucing();

// makan adalah method dari kelas parent yaitu Hewan
echo $KucingPersia->makan(); // Output: Hewan ini makan
```

5. Praktikum:

A. Alat dan Bahan

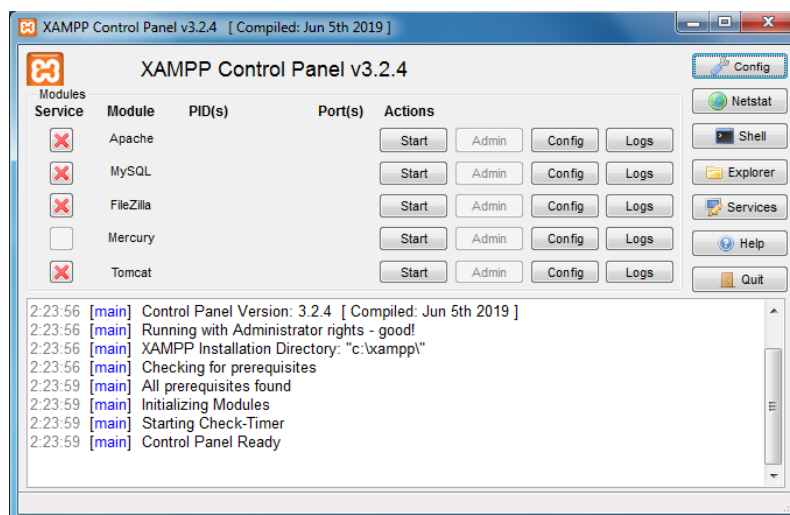
Praktikum 4 berisi materi tentang Konsep OOP pada PHP. Serta peralatan yang akan digunakan selama kuliah/praktikum, mencakup:

- 1) Server Lokal (XAMPP)
- 2) Teks Editor (VS Code)
- 3) Web Browser

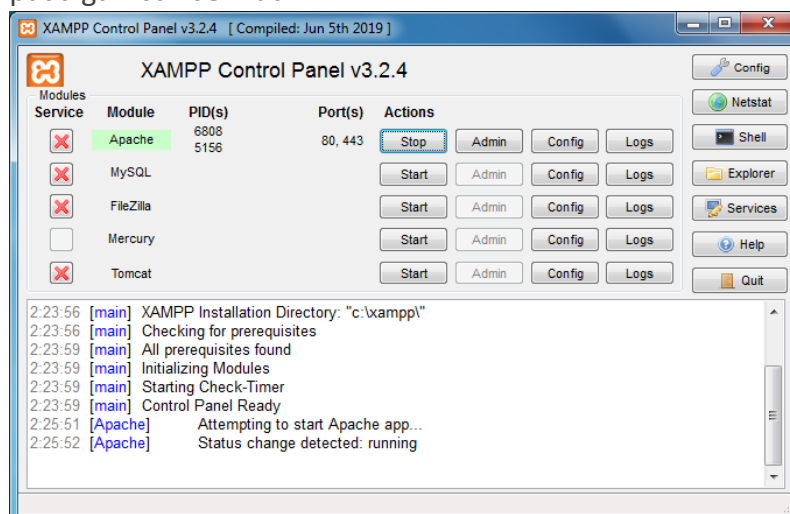
B. Praktikum

1. install dan nyalakan XAMPP

- a) Yang belum instal silahkan diinstal dulu
- b) Yang sudah ada silahkan buka file XAMPP sampai tampil seperti gambar dibawah



- c) Jalankan Apache dengan cara tekan tombol start sehingga tampil seperti pada gambar berikut



2. Buat folder **Pertemuan 4** pada folder **XAMPP/htdocs/PBW2**
3. Ketikkan kode berikut dan simpan pada folder
XAMPP/htdocs/PBW2/Pertemuan 4 dengan nama file **index.html**

```
<!DOCTYPE html>
<html>
<head>
  <title>PBW2 - Pertemuan 4</title>
</head>
<body>
  <h3>Konsep OOP</h3>
  <ol>
    <li>Contoh 1</li>
    <li>Contoh 2</li>
    <li>Contoh 3</li>
    <li>Contoh 4</li>
  </ol>
</body>
</html>
```

4. Tambahkan kode berikut pada file **index.html** yang ada di folder
XAMPP/htdocs/PBW2

```
<!DOCTYPE html>
<html>
<head>
  <title>PBW2</title>
</head>
<body>
  <h2>Selamat Datang di Pemrograman Berbasis Web 2</h2>
  <ol>
    <li><a href="Pertemuan 1">Pertemuan 1 - Mengenal PHP</a></li>
    <li><a href="Pertemuan 2">Pertemuan 2 - PHP Lanjut</a></li>
    <li><a href="Pertemuan 3">Pertemuan 3 - Form Handling</a></li>
    <li><a href="Pertemuan 4">Pertemuan 4 - Konsep OOP</a></li>
  </ol>
</body>
</html>
```

5. Ketikkan kode berikut dan simpan pada folder
XAMPP/htdocs/PBW2/Pertemuan 4 dengan nama file **cth1.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li>Contoh 1</li>
        <li><a href="cth2.php">Contoh 2</a></li>
        <li><a href="cth3.php">Contoh 3</a></li>
        <li><a href="cth4.php">Contoh 4</a></li>
    </ol>

    <?php
        class rumah{
            // properties / variabel / atribut
            public $tipe;
            public $lantai;
            // public key untuk enkapsulasi / visibilitas member

            // method setting properties
            function set_tipe($tipe){
                $this->tipe = $tipe;
            }

            // method ambil data properties
            function get_tipe(){
                return $this->tipe;
            }
        }

        // instansiasi objek (pembuatan objek)
        $rumah = new rumah();

        // set properties
        $rumah->set_tipe('21');

        // ambil data properties
        echo $rumah->get_tipe();

    ?>
</body>
</html>
```

6. Tambahkan kode berikut pada file **index.html** yang ada di folder **XAMPP/htdocs/PBW2/Pertemuan 4**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li><a href="cth1.php">Contoh 1</a></li>
        <li>Contoh 2</li>
        <li>Contoh 3</li>
        <li>Contoh 4</li>
    </ol>
</body>
</html>
```

7. Ketikkan kode berikut dan simpan pada folder **XAMPP/htdocs/PBW2/Pertemuan 4** dengan nama file **cth2.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li><a href="cth1.php">Contoh 1</a></li>
        <li>Contoh 2</li>
        <li><a href="cth3.php">Contoh 3</a></li>
        <li><a href="cth4.php">Contoh 4</a></li>
    </ol>
    <?php
        class rumah{
            // properties / variabel / atribut
            public $pemilik;
            public $warna;
            public $tipe;
            // public key untuk enkapsulasi / visibilitas member

            // method setting properties
            function setData($pemilik,$warna,$tipe){
                $this->pemilik = $pemilik;
                $this->warna = $warna;
                $this->tipe = $tipe;
            }

            // method ambil data properties
            function tampilData(){
                echo "Rumah Bapak {$this->pemilik} Berwarna {$this->warna} dengan tipe {$this->tipe}";
            }
        }

        // instansiasi objek (pembuatan objek)
        $infoRumah = new rumah();

        // set properties
        $infoRumah->setData('Taufiq','Hijau','70');

        // ambil data properties
        $infoRumah->tampilData();
    ?>
</body>
</html>
```

8. Tambahkan kode berikut pada file **index.html** yang ada di folder **XAMPP/htdocs/PBW2/Pertemuan 4**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li><a href="cth1.php">Contoh 1</a></li>
        <li><a href="cth2.php">Contoh 2</a></li>
        <li>Contoh 3</li>
        <li>Contoh 4</li>
    </ol>
</body>
</html>
```

9. Ketikkan kode berikut dan simpan pada folder **XAMPP/htdocs/PBW2/Pertemuan 4** dengan nama file **cth3.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li><a href="cth1.php">Contoh 1</a></li>
        <li><a href="cth2.php">Contoh 2</a></li>
        <li>Contoh 3</li>
        <li><a href="cth4.php">Contoh 4</a></li>
    </ol>
<?php
    class rumah{
        // properties / variabel / atribut
        public $pemilik;
        public $warna;
        public $tipe;
        // public key untuk enkapsulasi / visibilitas member

        // method setting properties
        function setData($pemilik,$warna,$tipe){
            $this->pemilik = $pemilik;
            $this->warna = $warna;
            $this->tipe = $tipe;
        }

        // method ambil data properties
        function tampilData(){
            echo "Rumah Bapak {$this->pemilik} Berwarna {$this->warna} dengan tipe {$this->tipe}";
        }

        function statusRumah(){
            if($this->tipe > 60){
                $status = "Luas";
            }elseif($this->tipe > 45){
                $status = "Sedang";
            }else{
                $status = "Kecil";
            }
            echo $status;
        }
    }
}
```



```
// instansiasi objek (pembuatan objek)
$infoRumah = new rumah();

// set properties
$infoRumah->setData('Taufiq','Hijau','70');

// ambil data properties
$infoRumah->tampilData();
echo " ";
echo "Rumah terlihat ";
$infoRumah->statusRumah();

?>
</body>
</html>
```

10. Tambahkan kode berikut pada file **index.html** yang ada di folder **XAMPP/htdocs/PBW2/Pertemuan 4**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li><a href="cth1.php">Contoh 1</a></li>
        <li><a href="cth2.php">Contoh 2</a></li>
        <li><a href="cth3.php">Contoh 3</a></li>
        <li>Contoh 4</li>
    </ol>
</body>
</html>
```

11. Ketikkan kode berikut dan simpan pada folder

XAMPP/htdocs/PBW2/Pertemuan 4 dengan nama file **cth4.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>PBW2 - Pertemuan 4</title>
</head>
<body>
    <h3>Konsep OOP</h3>
    <ol>
        <li><a href="cth1.php">Contoh 1</a></li>
        <li><a href="cth2.php">Contoh 2</a></li>
        <li>Contoh 3</li>
        <li><a href="cth4.php">Contoh 4</a></li>
    </ol>
    <?php
        class rumah{
            // properties / variabel / atribut
            public $pemilik;
            public $warna;
            public $tipe;
            // public key untuk enkapsulasi / visibilitas member

            // method setting properties
            function setData($pemilik,$warna,$tipe){
                $this->pemilik = $pemilik;
                $this->warna = $warna;
                $this->tipe = $tipe;
            }

            // method ambil data properties
            function tampilData(){
                echo "Rumah Bapak {$this->pemilik} Berwarna {$this->warna} dengan tipe {$this->tipe}, ";
            }

            function statusRumah(){
                if($this->tipe > 60){
                    $status = "Luas";
                }elseif($this->tipe > 45){
                    $status = "Sedang";
                }else{
                    $status = "Kecil";
                }
                echo $status;
            }

            function kisaranHarga(){
                if($this->tipe > 60){
                    $harga = "Diatas 300 Juta";
                }elseif($this->tipe > 45){
                    $harga = "200 - 300 Juta";
                }else{
                    $harga = "150 - 200 Juta";
                }
                echo $harga;
            }
        }

        // instansiasi objek (pembuatan objek)
        $infoRumah = new rumah();

        // set properties
        $infoRumah->setData('Taufiq','Hijau','70');

        // ambil data properties
        $infoRumah->tampilData();
        echo " ";
        echo "Rumah terlihat ";
        $infoRumah->statusRumah();
        echo " dengan kisaran harga ";
        $infoRumah->kisaranHarga();

    ?>
</body>
</html>
```

12. Tambahkan kode berikut pada file **index.html** yang ada di folder **XAMPP/htdocs/PBW2/Pertemuan 4**

```
<!DOCTYPE html>
<html>
<head>
  <title>PBW2 - Pertemuan 4</title>
</head>
<body>
  <h3>Konsep OOP</h3>
  <ol>
    <li><a href="cth1.php">Contoh 1</a></li>
    <li><a href="cth2.php">Contoh 2</a></li>
    <li><a href="cth3.php">Contoh 3</a></li>
    <li><a href="cth4.php">Contoh 4</a></li>
  </ol>
</body>
</html>
```

Kunci sukses belajar pemrograman adalah **sering praktek**, jangan hanya membaca saja, mulailah tulis kodemu sendiri.

RANGKUMAN

- Dengan menerapkan konsep object oriented struktur folder dan pembagian fungsi-fungsi didalamnya menjadi lebih rapi.
- Setiap kelas dapat dibuat menjadi banyak objek.

TES FORMATIF

DAFTAR PUSTAKA

codeofaninja.com

w3schools.com

php.net/docs.php