

Modul 4 : Implementasi PHP pada Object Oriented Programming (OOP)

4.1 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Mengetahui penggunaan Model.
2. Mengetahui penggunaan Controller.
3. Mengetahui penggunaan View.

4.2 Alat & Bahan

Alat & Bahan Yang digunakan adalah hardware perangkat PC beserta Kelengkapannya berjumlah 40 PC serta Software xampp, notepad++ yang telah terinstall pada masing-masing PC.

4.3 Dasar Teori

4.3.1 Mengetahui Penggunaan Model

Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Tipe data yang dimodelkan berupa apa saja; bisa table dalam database, file atau yang lainnya. Semua kelas model harus disimpan didalam direktori application\models.

1. Anatomi Model

Dalam codeigniter, setiap kelas model harus diturunkan dari kelas CI_Model. Bentuk umum pendefinisian model dituliskan sebagai berikut :

```
Class Nama_Model extends CI_Model{  
    //badan kelas  
}
```

Jika ingin memanggil konstruktor kelas CI_Model dari konstruktor kelas model yang didefinisikan sebagai berikut.

```
<?php  
class Nama_model extends CI_model{  
    //konstruktor kelas model  
    public function __construct(){  
        //memanggil konstruktor kelas CI_Model  
        parent::__construct();  
    }  
}
```

Nama_Model yang terdapat pada bentuk umum di atas merupakan nama kelas. Nama kelas harus diawali oleh huruf kapital/besar dan diikuti oleh huruf kecil. Nama file model harus sesuai dengan nama kelas. Sebagai contoh, jika nama kelasnya adalah `Lingkaran_model` maka file harus disimpan dengan nama `Lingkaran_model.php`.

2. Membuat Model

Untuk lebih memahami pendefinisian model, pada bagian ini kita akan membuat aplikasi sederhana yang digunakan untuk menghitung luas dan keliling lingkaran. Dalam kasus ini, model berperan untuk memodelkan data yang akan digunakan oleh aplikasi, yaitu objek lingkaran; controller berperan untuk menggunakan dan memproses model tersebut; sedangkan view berperan untuk membuat tampilan hasil akhir disajikan ke layar browser, sedangkan file lingkaran_form_view untuk input nilai jari-jari yang dinamis.

File : application\models\Lingkaran_model.php

```
<?php
class Lingkaran_model extends CI_Model{
    // mendefinisikan konstanta untuk Nilai PI
    const PI=3.14 ;
    //atribut model
    private $radius;
    //metode untuk menentukan nilai $radius
    public function set_radius($r){
        $this->radius = $r;
    }
    //metode untuk mengambil nilai $radius
    public function get_radius(){
        return $this->radius;
    }
    //metode untuk menghitung luas lingkaran
    public function hitung_luas (){
        return self::PI * $this->radius * $this->radius;
    }
    //metode untuk menghitung keliling lingkaran
    public function hitung_keliling(){
        return 2 * self::PI * $this->radius;
    }
}
```

Program 4-1 Program memuat model untuk menghitung luas lingkaran

Setelah memuat modelnya kita buat controller dan view yang akan dijelaskan pada sub bab selanjutnya.

4.3.2 Mengetahui Penggunaan Controller

Controller merupakan jantung dari aplikasi yang dibuat menggunakan codeigniter. Komponen ini yang akan menentukan bagaimana seharusnya permintaan user ditangani. sama halnya seperti model, controller juga merupakan kelas. Setiap Controller merupakan kelas turunan dari `CI_Controller`. Nama controller akan digunakan sebagai segmen pertama dari URI yang diminta oleh *user*. Segmen kedua URI nama metode di dalam controller yang akan dipanggil. Contoh URI :

```
http://localhost/index.php/demo_controller/aksi.
```

Pada URI diatas, `demo_controller` disebut segmen pertama dan `aksi` disebut segmen kedua. URI di atas akan memanggil atau mengeksekusi metode `aksi()` yang ada di dalam controller `demo_controller.php`. Jika pada saat pemanggilan URI segmen kedua tidak disertakan, maka metode yang akan dipanggil adalah metode `index()`. Dalam codeigniter metode `index()` berperan sebagai aksi default.

Kode program di bawah ini merupakan kelanjutan dari model yang sudah dibuat sebelumnya, namun ini merupakan controllernya. Adapun Kode program dimaksud adalah sebagai berikut :

```
<?php
class Lingkaran extends CI_Controller {
    public function index(){
        if (isset($_POST['btnsubmit'])){
            //memuat model Lingkaran Model
            $this->load->model('Lingkaran_model');
            //menangkap model yang telah dimuat oleh controller
            $model = $this->Lingkaran_model;
            //mengambil nilai dari form input
            $radius = $_POST['jari2'];

            //menentukan nilai $radius
            $model->set_radius($radius);

            //memuat view dan mengirimkan $model ke view
            $this->load->view('lingkaranview', array('model'=>$model));
        }else{
            //menampilkan form
            $this->load->view('lingkaran_form_view');
        }
    }
}
```

4.3.3 Mengetahui Penggunaan View

View pada dasarnya adalah sebuah halaman web, atau bisa juga berupa bagian tertentu dalam halaman web. Bagian dalam halaman web bisa disebut dengan fragmen. View tidak dapat langsung dipanggil di web browser, tetapi harus melalui controller. View ditulis menggunakan kode HTML dan kode-kode PHP yang ringan. Kode program di bawah ini merupakan kelanjutan dari kode program sebelumnya (model dan controllernya) dalam bentuk tampilan input dan output sebagai berikut :

application\models\Lingkaran_form_view.php

```
<html>
<head><title>Demo Model</title></head>
<body>
<h2>Luas dan Keliling lingkaran</h2>
<form action="http://localhost/ci/index.php/Lingkaran" method="post">
Masukan nilai jari-jari : <br />
<input type="text" name="jari2">
<input type="submit" name="btnsubmit" value="Hitung">
</form>

</body>
</html>
```

Untuk menampilkan hasil perhitungannya buat kode program di bawah ini

application\models\Lingkaran_form_view.php

```
<html>
<head><title>Controller, Model, View</title></head>
<body>
<h2>Model lingkaran</h2>
<!-- memanggil metode get_radius() dari objek $model -->
Nilai jari-jari: <?php echo $model->get_radius();    ?> <br />

<!-- memanggil metode hitung_luas() dari objek $model -->
Luas Lingkaran: <?php echo $model->hitung_luas();    ?> <br />
```

```
<!-- memanggil metode hitung_keliling() dari objek $model -->
Nilai jari-jari: <?php echo $model->hitung_keliling();    ?> <br />

</body>
</html>
```

4.4 Latihan

1. Buat program berbasis Ci menggunakan konsep MVC, untuk menghitung besarnya Ppn 10% dengan inputan total pembelian menggunakan textbox pada halaman input, sedangkan outputnya menghasilkan total pembelian sebelum pajak, besarnya pajak, dan total pembelian setelah ditambah ppn 10%
2. Buat program berbasis Ci menggunakan konsep MVC, untuk menghitung besarnya total bayar dengan ketentuan sebagai berikut :
 - a. Inputan terdiri dari total pembelajaan (pakai textbox) dan status member(member atau non member)(pakai combobox), di tampilkan dalam form.
 - b. Jika yang bersangkutan member, maka potongan bayarnya adalah Total belanja * 15%. Jadi total bayar = totalbayar+ppn10 –(discount member *totalbayar+ppn10%)
 - c. Jika yang bersangkutan non-member, maka tidak ada potongan bayarnya Jadi total bayar = totalbayar+ppn10 –(discount member *totalbayar+ppn10%)