MODUL 6 : Bekerja dengan Library

0.1 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Mengetahui penggunaan library.
2. Mengetahui perluasan library.
3. Mengetahui pembuatan library sendiri.

0.2 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.

0.3 Dasar Teori

Dalam paket distribusinya, CodeIgniter hadir dengan disertai sekumpulan file kelas yang dapat digunakan untuk berbagai keperluan di dalam aplikasi. Kumpulan kelas ini disebut library. Dengan adanya library yang cukup lengkap, pengguna CodeIgniter hanya perlu berfokus pada permasalahan yang akan mereka buat atau kembangkan, bukan pada cara melakukannya. CodeIgniter untuk menulis kode yang lebih baik, mudah dan dapat dilakukan secara cepat.

Lokasi Library yang disediakan oleh CodeIgniter, ditempatkan di dalam direktori `system\libraries`. Jika kita ingin membuat library sendiri atau ingin menggunakan library yang dikembangkan oleh pihak ketiga dalam apliksi yang dibuat, maka library tersebut perlu di tempatkan di dalam direktori `application\libraries`.

Nama kelas library yang disediakan Oleh CodeIgniter selalu diawali dengan prefix CI_. Nama file juga sama dengan nama kelas, tapi prefiksnya dihilangkan. Sebagai contoh, library yang digunakan untuk proses pengiriman email didefinisikan ke dalam kelas Ci_email dan disimpan dalam file bernama Email.php. Contoh lainnya, library yang digunakan untuk keperluan upload file didefinsikan ke dalam kelas CI_upload dan disimpan dalam file dengan nama Upload.php

0.3.1 Mengetahui Penggunaan Library

Sebelum menggunakan library, ada kalanya kita mengetahui tentang pemuatan library pada kode program CI. Untuk memuat helper maka kode program harus disimpan pada controller pada bagian konstruktor dengan kode program sebagai berikut `$this->load->library('nama_library');`. Memuta library bisa dengan cara lain yaitu dengan memuat library secara otomatis pada configurasi. Untuk melakukannya cari file autoload pada direktori application/config. Kemudian cari library, tambahkan library yang akan dimuat sesuai kebutuhan. Adapun contohnya sebagai berikut : `$autoload['libraries'] = array('email','form_validation','table');`

1. Contoh penggunaan Library table

Untuk menggunakan library, pertama kita buat controllernya sebgai berikut dengan nama Demo_library sebagai berikut :

```
<?php
class Demo_library extends CI_controller{
    public function __construct(){
    parent::__construct();
    //memuat library Table.php
    $this->load->library('table');
    }
```

```
    public function index(){
        $data['heading'] = ['ID', 'Nama Produk', 'Harga'];
        $data['row1'] = ['P01', 'Spidol', 8500];
        $data['row2'] = ['P02', 'Tinta', 22000];
        $data['row3'] = ['P03', 'Pensil', 5000];
        $this->load->view('demo_library_view', ['data'=>$data]);
    }
}
```

Kemudian but untuk viewnya dengan nama demo_library_view sebagai berikut.

```
<html>
<head><title>Demo Menggunakan Library</title></head>
<body>
<h2>Demo Menggunakan Library</h2>
<?php
//Menggunakan Library table.php
$this->table->set_heading($data['heading']);

$this->table->add_row($data['row1']);
$this->table->add_row($data['row2']);
$this->table->add_row($data['row3']);

echo $this->table->generate();
?>


</body>
</html>
```

Apa hasilnya, silahkan tulis pada kotak berikut (tulis dalam kertas) setelah controllernya di panggil.

0.3.2 Memperluas library

Memperluas library disini dalam artian menambah atau mengganti metode didalam library. Untuk membuat perluasan library dari library yang sudah ada file yang diperluasnya di simpan pada direktori applicarion/libraries dengan nama sama seperti nama file helper yang asal yang diperluas, namun harus menambahkan awalan MY_. Contoh kasus jika kita akan mengembangkan library table. Kita copy file table.php pada direktori system/libraries ke application/libraries. Tambahkan awalan MY. Sehingga file perubahannya nanti akan berubah menjadi MY_table.php pada direktori application/libraries. Adapun contoh perubahannya sebagai berikut :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class MY_Table extends CI_Table {
    protected function _default_template()
    {
        return array(
                        'table_open'           => '<table border="1"
cellpadding="4" cellspacing="0">',

            'thead_open'          => '<thead>',
            'thead_close'         => '</thead>',

            'heading_row_start'   => '<tr>',
            'heading_row_end'     => '</tr>',
            'heading_cell_start'  => '<th>',
            'heading_cell_end'    => '</th>',

            'tbody_open'          => '<tbody>',
            'tbody_close'         => '</tbody>',

            'row_start'           => '<tr>',
            'row_end'             => '</tr>',
            'cell_start'          => '<td>',
            'cell_end'            => '</td>',

            'row_alt_start'             => '<tr>',
            'row_alt_end'         => '</tr>',
            'cell_alt_start'      => '<td>',
            'cell_alt_end'        => '</font></td>',

            'table_close'         => '</table>'
        );
    }

}
```

Apa hasilnya setelah controller demo_library-nya dipanggil, silahkan tulis pada kotak berikut (tulis dalam kertas).

0.3.3 Pembuatan Library sendiri

Untuk menangani permasalahan-permasalahan spesifik yang diperlukan oleh aplikasi, kita bisa saja membuat library sendiri. Dalam membuat library, anda perlu memperhatikan aturan-aturan berikut :

1. Nama file harus diawali dengan huruf besar
2. Nama kelas harus diawali dengan huruf besar
3. Nama file dan nama kelas harus sama
4. File harus disimpan di dalam direktori application\libraries.

Berbeda pada saat kita memperluas library, pada saat membuat library kita tidak perlu mengawali nama kelas maupun nama file dengan prefix `MY_`. Nama kelas bebas, bisa apa saja selama tidak melanggar aturan-aturan yang disebutkan diatas.

Cara membuat library yang dibuat sendiri dengan library yang telah disediakan oleh CodeIgniter pada dasarnya sama saja. Sebagai contoh, jika kita memiliki library dengan nama Arrays.php, maka proses peuatannya dilakukan menggunakan kode berikut :

```
$this->load->library(arrays);
```

Cara yang agak berbeda dijumpai ketika library yang kita buat memiliki parameter di dalam bagian konstruktornya. Sebagai contoh, jika kelas Arrays memiliki parameter di bagian konstruktor =, maka proses pemuatannya perlu ditulis seperti berikut :

```
$param ['param1' => 'nilai1', 'param2' =>'nilai2'];
$this->load->library('arrays', $param);
```

Pada bagian kelas (library yang kita buat), kodenya perlu ditulis seperti berikut :

```
Class Arrays {
   Public function __construct($param){
//$param digunakan disini
}
}
```

0.4 Latihan

1. Modifikasi library table sehingga tampilannya seperti berikut :

Demo Menggunakan Library

ID	Nama Produk	Harga
P01	Spidol	8500
P02	Tinta	22000
P03	Pensil	5000

2. Tambah data sehingga tampilannya akan seperti berikut :

Demo Menggunakan Library

ID	Nama Produk	Harga
P01	Spidol	8500
P02	Tinta	22000
P03	Pensil	5000
P04	Ballpoint	7000
P05	Kapur	2000
P06	Pensil	5000
P07	Ballpoint	7000
P08	Kapur	2000

DAFTAR PUSTAKA

- Programming PHP, Kevin Tatroe, Peter MacIntyre & Rasmus Lerdorf, 2013; Oreilly
- Raharjo Budi 2015. Belajar otodidak framework codeigniter. Informatika.Bandung