

## MODUL 5 : Bekerja Dengan Helper

### 0.1 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Mengetahui penggunaan Helper.
2. Mengetahui perluasan Helper.
3. Mengetahui pembuatan helper 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 CodeIgniter, pustaka hanya ditulis dalam bentuk kelas. CodeIgniter menyebutnya sebagai library. Disisi lain, CodeIgniter juga menyediakan kumpulan fungsi(non-kelas). CodeIgniter menyebut kumpulan fungsi tersebut sebagai helper.

Helper (penolong). Sesuai dengan namanya, berguna untuk membantu tugas-tugas pemrograman web yang kita hadapi. Helper merupakan file yang berisi kumpulan fungsi procedural, yang dikelompokkan berdasarkan kategori tertentu. Maksud fungsi procedural disini adalah fungsi yang didefinisikan diluar kelas, tidak ditulis dalam format Object Oriented. Fungsi yang didefinisikan dalam kelas tersebut metod(method). Helper merupakan kumpulan fungsi, bukan kumpulan kelas. Helper dapat dimuat didalam model, controller maupun view; tapi pada umumnya pemuatan helper dilakukan di bagian konstruktor di dalam controller. Setiap fungsi helper digunakan untuk menyelesaikan tugas spesifik dan tidak tergantung dengan fungsi helper lainnya.

Lokasi helper(daftar file helper) ditempatkan dalam folder khusus, yaitu system\helper yang berisi file helper yang telah didefinisikan oleh code igniter, sedangkan application\helper digunakan untuk menempatkan file helper yang akan kita definisikan sendiri. Nama file helper selalu diakhiri dengan teks \_helper.

#### 0.3.1 Mengetahui Penggunaan Helper

Sebelum menggunakan Helper, ada kalanya kita mengetahui tetnagn pemuatan helper pada kode program CI. CodeIgniter tidak memuar file helper secara default baik dalam direktori sistem atau aplikasi. Untuk memuat helper maka kode program harus disimpan pada controller pada bagian konstruktor dengan kode program sebagai berikut `$this->load->helper('nama_helper');`. Memuta helper bisa dengan cara lain yaitu dengan memuat helper secara otomatis pada konfigurasi. Untuk melakukannya cari file autoload pada direktori application/config. Kemudian cari helper, tambahkan helper yang akan dimuat sesuai kebutuhan. Adapun contohnya sebagai berikut `:$autoload['helper'] = array('file','url','form','text','html');`

#### 1. Contoh penggunaan helper array

Untuk menggunakan helper, pertama kita buat controllernya sebgai berikut dengan nama Demo\_helper sebagai berikut :

```

<?php
class Demo_helper extends CI_Controller{
    public function __construct(){
        parent::__construct();

        //memuat file helper 'array_helper.php'
        $this->load->helper('array');
    }
    public function index(){
        $data =['python', 'Perl', 'PHP', 'Ruby'];
        //mengirim $data ke view
        $this->load->view('demo_helper_view', ['data'=>$data]);
    }
}

```

Kemudian buat untuk viewnya dengan nama demo\_helper\_view sebagai berikut.

```

<html>
<head><title>Demo helper</title></head>
<body>
<h2>Demo menggunakan Helper</h2>
<p>
Elemen-elemen Array:<br/>
<?php
foreach ($data as $elemen) {
    echo $elemen . " ";
}
?>
</p>

Pengambilan Acak Pertama
<?php echo random_element($data); ?><br />

Pengambilan Acak Kedua
<?php echo random_element($data); ?><br />

Pengambilan Acak Ketiga
<?php echo random_element($data); ?>
</body>
</html>

```

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

### 0.3.2 Memperluas Helper

Memperluas helper disini dalam artian menambah atau mengganti fungsi-fungsi didalam helper. Untuk membuat perluasan helper dari helper yang sudah ada file yang diperluasnya di simpan pada direktori application/helper dengan nama sama seperti nama file helper yang asal yang diperluas, namun harus menambahkan awalan MY\_. Contoh kasus jika kita akan mengembangkan helper array. Kita copy file array\_helper.php pada direktori system/helper ke application/helper. Tambahkan awalan MY\_. Sehingga file perubahannya nanti akan berubah menjadi MY\_Array\_helper.php pada direktori application/helper.

### 0.3.3 Pembuatan helper sendiri

Untuk membuat helper sendiri caranya hamper sama dengan memeperluas helper dari helper yang sudah ada, namun untuk membuat helper sendiri tidak perlu menambahkan awalan MY\_ seperti pada perluasan helper, namun cukup menambahkan akhiran \_helper saja. Contoh kita akanmembuat fungsi-fungsi aritmatika dasar matematika dengan nama matematika. Maka file tersebut harus disimpan pada directory application/helper dengan nama matematika\_helper.php. adapun kode program untu helpernya sendiri sebagai berikut :

application\helper\matematika\_helper.php

```
<?php
// fungsi untuk menghitung penjumlahan bilangan
function tambah($a, $b){
    if (!is_numeric($a) OR !is_numeric($b)) return NULL;
    return $a + $b;
}
// fungsi untuk menghitung perkalian bilangan
function kali($a, $b){
    if (!is_numeric($a) OR !is_numeric($b)) return NULL;
    return $a * $b;
}
// fungsi untuk menghitung pembagian bilangan
function bagi($a, $b){
    if (!is_numeric($a) OR !is_numeric($b)) return NULL;
    return $a / $b;
}
// fungsi untuk menghitung perpangkatan bilangan
function pangkat($a, $b){
    if (!is_numeric($a) OR !is_numeric($b)) return NULL;
    return pow($a, $b);
}
// fungsi untuk menghitung akar kuadrat dari suatu bilangan
function akar($a){
    if (!is_numeric($a)) return NULL;
    return sqrt($a);
}
```

Fungsi-fungsi diatas merupakan fungsi dasar pada matematika. Untuk implementasi dari helper yang kita buat tadi. Kita akan deklarasikan pemuatannya pada controller di bagian konstruktor sebagai berikut :

application\controller\Demo\_helper1.php

```
<?php
class Demo_helper1 extends CI_Controller{
    public function __construct(){
        parent::__construct();

        //memuat file helper 'matematika_helper.php'
        $this->load->helper('matematika');
    }
    public function index(){
        $data = [
            'a'=> 10,
            'b'=> 3,
            'c'=> 25,
        ];
        //mengirim $data ke view
        $this->load->view('demo_helper_view1', $data);
    }
}
```

Dalam controller tersebut helper yang di muat adalah helper matematika, meskipun dalam file-nya bernama matematika\_helper.php, kita cukup memuat nama helper-nya saja tanpa akhiran helper. Dalam public function index tersebut, deklarasi variabel untuk operasi matematika sudah di tetapkan nilainya, kemudian variabel tersebut di kirim ke form demo\_helper\_view1.php yang harus disimpan pada folder application/view/form\_helper\_view1 yang akan dibuat sebagai berikut :

```
<html>
<head><title>Demo helper</title></head>
<body>
<h2>Demo menggunakan Helper</h2>

<?php echo "$a + $b = " . tambah($a,$b); ?><br />
<?php echo "$a * $b = " . kali($a,$b); ?><br />
<?php echo "$a / $b = " . bagi($a,$b); ?><br />
<?php echo "$a ^ $b = " . pangkat($a,$b); ?><br />
<?php echo "Akar $c = " . akar($c); ?><br />

</body>
</html>
```

Pada view diatas fungsi-fungsi yang ada pada helper yang dibuat tadi dipanggil dan diinputkan parameternya untuk dilakukan operasinya.

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

## 0.4 Latihan

1. Buat helper sendiri dengan nama bilangan untuk menentukan bilangan ganjil, genap, prima, bilangan kelipatan dua sesuai yang diinputkan di parameter. Implementasikan pada controller dan viewnya. Contoh jika pada inputannya angka 7 maka hasil yang ada pada layar bilangan 7 merupakan : bilangan Ganjil, bukan genap, Prima, bukan kelipatan 2.
2. Buat helper sendiri dengan nama harga untuk menghitung pajak (ppn 10% dari total belanja - discount), discount(dinput pada parameter), total harga setelah discount(total belanja-discount), total harga setelah pajak(total harga setelah discount + pajak) . Contoh : Jika pada inputan total belanja 1000 discount 10%, maka pada viewnya akan muncul Total belanja 1000 dengan ppn sebesar 100, discount 100, total harga setelah discount 900, total harga setelah pajak 1000.
3. Cari helper yang lain minimal 3 helper pengertian dan kegunaan helper tersebut beserta cara menggunakannya.

## DAFTAR PUSTAKA

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