

Modul 7 : Master Data metode CRUD

7.1 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Membuat Database dan table.
2. Mengetahui koneksi antara database dengan framework CI.
3. Membuat master barang.

7.2 Alat & Bahan

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

7.3 Dasar Teori

Meskipun aplikasi web sebenarnya bisa saja tidak memiliki database, tapi pada kenyataannya sebagian besar aplikasi web memilikinya. Sistem database digunakan untuk menyimpan dan mengelola data yang dibutuhkan oleh aplikasi. Kali ini akan dibahas tentang interaksi antara database dengan aplikasi yang dibuat menggunakan CodeIgniter.

CodeIgniter menyediakan library khusus yang digunakan untuk keperluan akses database. Untuk mengakses database, kita dapat menggunakan cara tradisional, yaitu dengan menuliskan sintaks-sintaks SQL standar, atau dengan memanfaatkan fasilitas *query builder*.

CodeIgniter mendukung koneksi ke beberapa jenis server database, seperti MySQL, PostgreSQL, Oracle, dan sebagainya. Dalam praktikum ini akan mendemonstrasikan koneksi database menggunakan MySQL.

7.3.1 Membuat database dan table

Untuk mencoba mendemonstrasikan Database MySQL dengan framework CodeIgniter terlebih dahulu kita harus membuat database dan tablenya terlebih dahulu. Langkah pertama membuat database MySQL dengan nama **posdb**, kemudian buat table dengan nama **barang** dengan field sebagai berikut :

Nama field	Tipe data	Key
Kode	Char(4)	Primary key
Nama	Varchar(30)	
Harga	Decimal	
Stok	integer	

7.3.2 Mengetahui koneksi antara database dengan framework CI

Untuk membangun koneksi ke server database MySQL, kita perlu melakukan konfigurasi terhadap file `application\config\database.php`

```
$db['default'] = array(  
    'dsn'       => '',  
    'hostname' => 'localhost',  
    'username' => 'root',
```

```

        'password' => '',
        'database' => 'posdb',
        'dbdriver' => 'mysqli',
        'dbprefix' => '',
        'pconnect' => TRUE,
        'db_debug' => TRUE,
        'cache_on' => FALSE,
        'cachedir' => '',
        'char_set' => 'utf8',
        'dbcollat' => 'utf8_general_ci',
        'swap_pre' => '',
        'encrypt' => FALSE,
        'compress' => FALSE,
        'stricton' => FALSE,
        'failover' => array(),
        'save_queries' => TRUE
    );

```

Tambahkan pada application\config\autoload.php.

```

$autoload['libraries'] = array('database', 'table');

$autoload['helper'] = array('file','url','form','text','html');

```

7.3.3 Membuat master barang metode Create Read Update Delete (CRUD)

Untuk implementasi master data terlebih dahulu perlu membuat kode program untuk me-model-kannya dengan nama file Barang_model.php sebagai model, adapun kode program di maksud adalah sebagai berikut :

```

<?php
class Barang_model extends CI_model{
    public $kode;
    public $nama;
    public $harga;
    public $stok;

    //array untuk menyimpan label dari masing-masing atribut
    public $labels=[];

    //konstuktur model
    public function __construct(){
        parent::__construct();
        $this->labels = $this->_attributelabels();

        //memuat librari database
        $this->load->database();
    }
    //untuk menambah baris data dalam tabel
    public function insert(){
        $sql=sprintf("INSERT INTO barang VALUES('%s','%s', %f, %d)",
            $this->kode,
            $this->nama,
            $this->harga,
            $this->stok
        );
        $this->db->query($sql);
    }
    public function update(){
        $sql=sprintf("UPDATE barang SET nama ='%s', harga= %f, stok= %d WHERE
kode='%s'",

```

```

        $this->nama,
        $this->harga,
        $this->stok,
        $this->kode
    );
    $this->db->query($sql);
}
public function delete(){
    $sql=sprintf("DELETE FROM barang WHERE kode='%s'", $this->kode);
    $this->db->query($sql);
}
public function read(){
    $sql="SELECT * FROM barang ORDER BY kode";
    $query = $this->db->query($sql);
    return $query->result();
}

//metode untuk menentukan label dari masing-masing atribut

public function _attributelabels(){
    return [
        'kode'=>'Kode:',
        'nama'=>'Nama Produk:',
        'harga'=>'Harga Produk:',
        'stok'=>'Stok:'
    ];
}

}

```

Buat Controllernya(sementara untuk melihat data terlebih dahulu) dengan nama Crud.php adapun kode programnya sebagai berikut :

```

<?php
class crud extends CI_controller{
    public $model = NULL;

    public function __construct(){
        parent::__construct();
        //memuat model
        $this->load->model('Barang_model');
        $this->model = $this->Barang_model;

        //memuat librari database
        $this->load->database();
    }

    public function index(){
        $this->read();
    }
    public function create() {
        //belum implementasi
        if (isset($_POST['btnSubmit'])) {
            $this->model->kode=$_POST['kode'];
            $this->model->nama=$_POST['nama'];
            $this->model->harga=$_POST['harga'];
            $this->model->stok=$_POST['stok'];
            $this->model->insert();
            redirect('crud');
        }else{
            $this->load->view('crud_create_view', ['model'=>$this->model]);
        }
    }
}

```

```

public function read() {
    $rows=$this->model->read();
    $this->load->view('crud_read_view', ['rows'=>$rows]);
}
public function update($id) {
    //belum implementasi
    if (isset($_POST['btnSubmit'])) {
        $this->model->kode=$_POST['kode'];
        $this->model->nama=$_POST['nama'];
        $this->model->harga=$_POST['harga'];
        $this->model->stok=$_POST['stok'];
        $this->model->update();
        redirect('crud');
    }else{
        $query = $this->db->query("SELECT * FROM barang WHERE
        kode='$id'");
        //if ($query->num_row() ==0) exit(1);
        $row =$query->row();
        $this->model->kode=$row->kode;
        $this->model->nama=$row->nama;
        $this->model->harga=$row->harga;
        $this->model->stok=$row->stok;
        $this->load->view('crud_update_view',
        ['model'=>$this->model]);
    }
}
public function delete($id) {
    //menentukan kode yang akan di hapus
    $this->model->kode = $id;
    //menghapus baris data di dalam tabel barang
    $this->model->delete();
    //mengarahkan kembali ke halaman utama/index
    redirect('crud');
}
}

```

Kemudian buat untuk view untuk tampilkan barang dengan nama crud_read_view.php, adapun kode programnya sebagai berikut :

```

<html>
<head><title>Demo Crud</title></head>
<body>
<h2>Demo Crud dengan metode Query()</h2>
<p><a href="Crud/create">tambah</a></p>
<table border="1">
<tr>
    <th width="80">Kode</th>
    <th width="120">Nama barang</th>
    <th width="80">Harga</th>
    <th width="80">Stok</th>
    <th width="100">Ubah</th>
    <th width="100">Hapus</th>
</tr>
<?php
foreach ($rows as $row){
?>
<tr>
    <td><?php echo $row->kode;?></td>
    <td><?php echo $row->nama;?></td>
    <td><?php echo $row->harga;?></td>
    <td><?php echo $row->stok;?></td>
    <td align="center">
    <a href="Crud/update/<?php echo $row->kode; ?>">Ubah</a>

```

```

        </td>
        <td align="center">
        <a href="Crud/delete/<?php echo $row->kode; ?>">Hapus</a>
        </td>
    </tr>
<?php
}
?>

</table>
</body>
</html>

```

Kemudian untuk input barang dengan nama crud_create_view.php adalah sebaai berikut :

```

<html>
<head><title>Demo CRUD</title></head>
<body>
<h2>Demo CRUD dengan metode query()</h2>
<p><strong>Tambah Data</strong></p>

<form action="create" method="post">
<?php echo $model->labels['kode']; ?> <br />
<input type="text" name="kode" size="10" maxlength="4" /><br /><br />

<?php echo $model->labels['nama']; ?> <br />
<input type="text" name="nama" size="30" maxlength="25" /><br /><br />

<?php echo $model->labels['harga']; ?> <br />
<input type="text" name="harga" size="20" /><br /><br />

<?php echo $model->labels['stok']; ?> <br />
<input type="text" name="stok" size="10" /><br /><br />

<input type="submit" name="btnSubmit" value="Simpan">
<input type="button" value="Batal" onclick="javascript:history.go(-1);"/>
</form>

</body>
</html>

```

Kemudian untuk update barang dengan nama crud_update_view.php adalah sebaai berikut :

```

<html>
<head><title>Demo CRUD</title></head>
<body>
<h2>Demo CRUD dengan metode query()</h2>
<p><strong>Update Data</strong></p>

<form action="update" method="post">
<?php echo $model->labels['kode']; ?> <br />
<input type="text" name="kode" size="10" maxlength="4"
readonly value="<?php echo $model->kode;?>" /><br /><br />

<?php echo $model->labels['nama']; ?> <br />
<input type="text" name="nama" size="30"
maxlength="25" value="<?php echo $model->nama;?>" /><br /><br />

<?php echo $model->labels['harga']; ?> <br />
<input type="text" name="harga" size="20"
value="<?php echo $model->harga;?>" /><br /><br />

```

```

<?php echo $model->labels['stok']; ?> <br />
<input type ="text" name="stok" size="10"
value="<?php echo $model->stok;?>" /><br /><br />

<input type ="submit" name="btnSubmit" value="Simpan">
<input type ="button" value="Batal" onclick="javascript:history.go(-1);"/>
</form>

</body>
</html>

```

7.4 Latihan

1. Buat database dengan nama bengkel buat table dengan kendaraan adapun filed yang dibuat adalah sebagai berikut :

Nama field	Tipe data	Key
Nomor_tnkb	Char(9)	Primary key
Nama_pemilik	Varchar(30)	
Nomor_mesin	Varchar(30)	
Nomor_sasis	Varchar(30)	
Jenis_mobil	Varchar(30)	
Kapasitas_mesin_mobil	Integer	
Tahun_pembuatan_mobil	Char(9)	

2. Buat aplikasi crud-nya untuk table nomor 1. Untuk jenis mobil pada saat input data gunakan combobox dengan value : Sedan, minibus, bus, truck.

DAFTAR PUSTAKA

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