|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran CodeIgniter 4 Video 10 – 19  (Tugas minggu ke - 18) | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Memahami jika sudah dilakukan pemanggilan melalu route maka tidak bisa dipanggil manual karena akan muncul error objek tidak ditemukan / object not found.
2. Melakukan percobaan pada file Menu.php, pertama, sebelum memasukkan function select ke dalam routes pemanggilan bisa dilakukan secara manual, yaitu dengan localhost/restoran-ci4/Admin/Menu/select, artinya saya memanggil function select di dalam file Menu. Setelah itu saya memasukkan function select ke dalam route, $route->add('Menu', 'Admin\Menu::select'), apabila dituliskan Menu, maka akan dilakukan pemanggilan pada function select di dalam file Menu.php.
3. Melakukan percobaan kedua, apabila route tertulis seperti ini, $routes->add('Kategori/create', 'Admin\Kategori::create');, maka jika dituliskan Kategori/create akan memanggil function create di dalam file Kategori. Dan jika route ditulis seperti ini, $routes->add('Kategorii', 'Admin\Kategori::create');, maka yang akan dipanggil adalah function create di dalam file Kategori. Apabila route sudah diatur jika dituliskan Kategorii maka akan memanggil function create dalam file Kategori, maka tidak bisa melakukan pemanggilan dengan cara manual, yaitu dengan menuliskan localhost/restoran-ci4/Admin/Kategori/create, jika dituliskan seperti ini, maka akan muncul tulisan object not found, karena pengaturan pemanggilan sudah diatur apabila ingin menampilkan function create, maka dengan menuliskan localhost/restoran-ci4/Admin/Kategorii saja.
4. Menambahkan kolom keterangan di dalam tblkategori.
5. Mengganti beberapa nama pada function, function select() diganti menjadi function read(), menghapus function selectwhere, mengganti function formInsert menjadi function insert saja, dan mengganti function formUpdate menjadi function find.
6. Memperbaiki routes karena terdapat pergantian nama dari function.

* Cth :

$routes->group('Admin', function($routes){

$routes->add('Kategori', 'Admin\Kategori::read');

$routes->add('Kategori/create', 'Admin\Kategori::create');

$routes->add('Kategori/find/(:any)', 'Admin\Kategori::find/$1');

});

Artinya, di dalam $routes, terdapat grouproute untuk menyambungkan beberapa $routes di dalamnya, grouproute memanggil Admin dengan hasil yang dipanggil berdasarkan function $routes, di dalam function $routes terdapat beberapa $routes, fungsi add pada setiap $routes adalah untuk memanggil apa yang ada di dalam add, $routes pertama, add memanggil Kategori, dengan hasil yang dipanggil yaitu, function read yang ada di dalam file Kategori dan folder Admin, jadi apabila Kategori diketikkan di dalam URL, maka akan menampilkan function/method read pada controller Kategori di dalam Admin. $routes kedua, add memanggil Kategori/create, dengan hasil yang dipanggil yaitu, function create yang ada pada controller Kategori di dalam Admin. Dan $routes ketiga, add memanggil Kategori/find/(:any), dengan hasil yang dipanggil yaitu method find pada controller Kategori di dalam Admin dengan nilai memanggil nilai dari parameter yaitu $1. Karena nilai parameter pada method find diisi dengan $1, maka untuk pemanggilannya diberi (:any), untuk menentukan segmen pada URL boleh berupa angka, huruf, atau karakter lain yang diizinkan.

1. Membuat tabel di dalam file select.php pada folder kategori di views.

* Cth :

<table border=1px>

    <tr>

        <th>No</th>

        <th>Kategori</th>

        <th>Keterangan</th>

    </tr>

<?php $no = 1 ?>

    <?php *foreach*($kategori as $key => $value): ?>

    <tr>

        <td><?= $no++?></td>

        <td><?= $value['kategori'] ?></td>

        <td><?= $value['keterangan']?></td>

    </tr>

    <?php *endforeach*; ?>

</table>

Artinya, terdapat tabel dengan garis tepi berukuran 1px, di dalam tabel terdapat 3 th, yaitu No, Kategori. dan Keterangan. Terdapat $no yang bernilai 1. Foreach digunakan untuk melakukan looping pada nilai yang diambil, yaitu pada $no, akan dilakukan looping secara otomatis untuk setiap penambahan baris karena pada $no sudah diberi tanda ++, kolom yang kedua yaitu kolom kategori, kolom kategori akan mengambil nilai dari $value['kategori'], dan kolom keterangan akan mengambil nilai dari $value['keterangan'].

1. Membuat form dengan method post di dalam file insert.php.

* Cth :

<form action="<?= base\_url()?>/Admin/Kategori/insert" method="post">

    Kategori : <input type="text" name="kategori" required>

    <br>

    Keterangan : <input type="text" name="keterangan" required>

    <br>

    <input type="submit" name="simpan" value="SIMPAN">

</form>

Artinya, terdapat form dengan action di dalamnya yaitu memanggi base\_url()/Admin/Kategori/insert, yang artinya jika form dikirimkan / submit SIMPAN di klik, maka akan diarahkan ke dalam base\_url yang memanggil localhost/reastoran-ci4, lalu disambungkan dengan /Admin/Kategori/insert, jadi tidakan dari form akan dikirimkan ke dalam method insert pada controller Kategori di dalam Admin. Pada form dengan method post ini terdapat form Kategori dengan input bertipe text dan name nya adalah kategori dan wajib diisi karena terdapat required di dalamnya. Pada form Keterangan dengan input bertipe text dan name adalah keterangan, form Keterangan ini wajib diisi karena terdapat tulisan required di dalamnya. Dan terdapat input dengan tipe submit dan name simpan serta value SIMPAN. Type submit digunakan untuk mengirimkan form.

1. Membuat function insert yang digunakan untuk apabila submit SIMPAN di klik, maka tindakan pada form akan diarahkan pada function insert.

* Cth :

public function insert()

{

    echo " <h4>INSERT DATA</h4>"

}

Artinya, terdapat function insert yang berisi tulisan INSERT DATA di dalamnya.

1. Mengecek apakah $\_POST yang ada pada form berhasil masuk ke dalam function insert atau tidak menggunakan print\_r.

* Cth :

public function insert()

{

    print\_r($\_POST);

}

Artinya, di dalam function insert akan menjalankan print\_r dari $\_POST dari form pada file insert.php, karena di dalam form tersebut sudah diberi action apabila submit SIMPAN di klik, maka tindakan pada form akan dikrimkan ke dalam function insert.

1. $table berfungsi untuk mengisi nama dari table yang digunakan pada model di dalam database, jadi nama tabel dari database yang diisi akan disambungkan ke dalam Model

* Cth :

class Kategori\_M extends Model

{

    protected $table = 'tblkategori';

 }

Artinya, di dalam class Kategori\_M terdapat $table yang berfungsi untuk mengisi nama tabel yang disambungkan ke dalam database. Di dalam class Kategori\_M, tabel yang dihubungkan ke dalam database adalah tblkategori.

1. $allowedFields digunakan untuk menentukan kolom mana saja yang dapat diinsert / dimasukkan data.

* Cth :

class Kategori\_M extends Model

{

    protected $table = 'tblkategori';

    protected $allowedFields = ['kategori', 'keterangan'];

}

Artinya, di dalam Kategori\_M terdapat $table yang digunakan untuk mengisi nama dari tabel yang digunakan dalam model di dalam database, tabel yang dihubungkan ke dalam database adalah tblkategori, dan $allowedFields yang digunakan untuk menentukan kolom mana yang diinsert, di dalam $allowedFields kolom yang akan diisi adalah kolom kategori dan kolom keterangan yang ada pada tblkategori.

1. Memasukkan data / insert data pada function insert di dalam Kategori.php.

* Cth :

public function insert()

{

    $model = new Kategori\_M();

    $model -> insert($\_POST);

}

Artinya, di dalam function insert terdapat objek $model yang memanggil Kategori\_M, dan di dalam $model terdapat function insert yang berfungsi untuk memasukkan data / insert data yang ada pada $\_POST. Karena di dalam form pada insert.php apabila submit SIMPAN di klik, maka tindakan pada form akan diarahkan pada function insert, sehingga di dalam function insert terdapat nilai dari $\_POST yang diisi dari form, lalu menjalankan objek $model yang mengambil nilai dari Kategori\_M, di dalam Kategori\_M menjalankan $table yaitu tblkategori dan menjalankan $allowedFileds, kolom yang diisi yaitu kolom kategori dan kolom keterangan. Lalu $model menjalankan function insert untuk memasukkan data, insert akan memasukkan data yang diambil dari $\_POST, lalu data akan dikirimkan ke dalam tblkategori di database.

1. Fungsi perintah insert() di dalam CodeIgniter 4, yaitu digunakan untuk memasukkan data / insert data.
2. redirect() digunakan untuk pengalihan alamat URL ke halaman lain yang dimasukkan.

* Cth :

public function insert()

    {

        $model = new Kategori\_M();

        $model -> insert($\_POST);

*return* redirect()->to(base\_url()."/Admin/Kategori");

    }

Artinya, setelah melakukan insert data / memasukkan data pada kolom keterangan dan kolom kategori pada tblkategori, maka halaman akan diarahkan / dialihkan ke dalam base\_url() yang berisi alamat localhost/restoran-ci4 lalu alamat disambung menggunakan titik, alamat disambung ke /Admin/Kategori, jadi setelah melakukan insert, maka halaman yang akan ditampilkan adalah halaman dari localhost/restoran-ci4/Admin/Kategori dimana jika URL tersebut ditulis akan menjalankan function read().

1. $primaryKey digunakan untuk menentukan primarykey dari tabel yang dihubungkan oleh model pada database, sebelum melakukan perintah untuk delete data, terlebih dahulu kita menentukan primarykey.

* Cth :

class Kategori\_M extends Model

{

    protected $table = 'tblkategori';

    protected $primaryKey = 'idkategori';

}

Artinya, di dalam class Kategori\_M terdapat $table yang digunakan untuk mengisi nama dari tabel yang dihubungkan dari model ke dalam database, jadi tabel pada model yang dihubungkan ke dalam database adalah tblkategori, lalu terdapat $primariKey yang digunakan untuk menentukan primarykey dari tabel, jadi primary key dari tblkategori adalah idkategori.

1. Membuat function delete dan mencoba untuk memanggilnya.

* Cth :

public function delete($id = null)

{

    echo "Proses delete data $id";

}

Artinya, di dalam function delete, terdapat nilai dari parameter yaitu, $id bernilai null, artinya isi dari $id adalah kosong, jadi apabila tidak diisi tidak menghasilkan error. Isi dari function delete adalah tulisan Proses delete data $id, nilai dari $id adalah nilai yang sesuai dengan nilai $id yang dimasukkan di dalam URL.

1. Melakukan delete data untuk tblkategori.

* Cth :

public function delete($id = null)

{

    $model = new Kategori\_M();

    $model -> delete($id);

}

Artinya, di dalam function delete, terdapat nilai dari parameter yaitu, $id bernilai null, artinya $id bernilai kosong, jadi ketika $id tidak diisi maka tidak menghasilkan error. Di dalam function delete terdapat objek $model yang memanggil Kategori\_M. Dan di dalam $model berisi function perintah yaitu delete / perintah untuk menghapus data. Jadi di dalam Kategori\_M terdapat $table yang berisi tblkategori, artinya tabel di dalam model yang dihubungkan ke dalam database adalah tblkategori, di dalam Kategori\_M juga terdapat $primaryKey yang berisi idkategori, artinya primarykey dari tblkategori adalah idkategori, jadi kolom yang nanti diambil untuk dihapus adalah mengambil nilai dari idkategori. Lalu $model akan menjalankan perintah untuk delete data / menghapus data, perintah delete menjalankan nilai dari $id dan akan menghapus $id yang dimasukkan.

1. Mengembalikan halaman ke dalam Admin/Kategori setelah dilakukan delete data.

* Cth :

public function delete($id = null)

{

    echo "Proses delete data $id";

    $model = new Kategori\_M();

    $model -> delete($id);

*return* redirect()->to(base\_url()."/Admin/Kategori");

}

Artinya, setelah melakukan delete data / menghapus data sesuai dengan $id sesuai dengan tblkategori yang dimasukkan ke dalam URL, maka halaman akan diarahkan ke dalam base\_url() yang berisi alamat localhost/restoran-ci4 lalu URL disambung dengan titik, alamat yang disambung adalah /Admin/Kategori, jadi apabila setelah melakukan delete data, maka halaman akan dikembalikan ke dalam halaman localhost/restoran-ci4/Admin/Kategori dimana jika alamat URL tersebut ditulis, maka akan memanggil function read().

1. Menambahkan kolom Hapus dan mengisi nilai kolom nya dengan mengambil nilai dari $value['idkategori'].

* Cth :

<th>Hapus</th>

<?php *foreach*($kategori as $key => $value): ?>

        <tr>

            <td><?= $value['idkategori']?></td>

        </tr>

        <?php *endforeach*; ?>

Artinya, terdapat kolom baru yaitu Hapus, dan isi dari kolom Hapus adalah, terdapat foreach yang digunakan untuk melakukan perulangan pada nilai dari kolom Hapus, kolom Hapus mengambil nilai dari $value['idkategori'], jadi kolom hapus berisi nilai dari idkategori pada setiap data.

1. Menambahkan hyperlink ke dalam isi dari kolom hapus dan mengarahkan ke dalam Admin/Kategori/delete/$id, agar ketika hyperlink di klik, maka akan langsung menghapus data pada idkategori.

* Cth :

<td>

    <a href="<?= base\_url()?>/Admin/Kategori/delete/<?= $value['idkategori']?>">

Hapus

</a>

</td>

Artinya, di dalam isi dari kolom hapus terdapat hyperlink, hyperlink memanggil base\_url() yang berisi alamat localhost/restoran-ci4, lalu /Admin/Kategori/delete/$value['idkategori'], artinya apabila hyperlink di klik, maka akan diarahkan pada alamat localhost/restoran-ci4/Admin/Kategori/delete/$value['idkategori'], yang artinya akan menjalankan function delete pada controller Kategori di dalam Admin, lalu di dalam parameter berisi $id, dan $value['idkategori'] akan mengisi nilai dari parameter untuk $id.

1. Update data digunakan untuk merubah isi data dari kategori dan keterangan pada tblkategori.
2. Menambah kolom Update dan memberi hyperlink di dalam kolomnya.

* Cth :

<th>Update</th>

<td>

    <a href="<?= base\_url()?>/Admin/Kategori/find/<?= $value['idkategori']?>">

        Update

    </a>

</td>

Artinya, terdapat kolom dengan thead Update, dan di dalam kolom update terdapat hyperlink yang memanggil base\_url(), base\_url berisi alamat localhost/restoran-ci4, lalu disambung dengan alamat /Admin/Kategori/find/$value['idkategori'], artinya apabila hyperlink di klik, maka akan diarahkan pada alamat localhost/restoran-ci4/Admin/Kategori/find/$value['idkategori'], yang artinya akan menjalankan function find di dalam controller Kategori pada Admin dengan nilai di dalam parameter mengambil nilai pada $value['idkategori'] sebagai isi dari $id karena $id bernilai null / kosong.

1. Perintah find() digunakan untuk mengambil / memanggil data.

* Cth :

$kategori = $model ->find($id);

Artinya, di dalam $kategori menjalankan $model dengan perintah find, yaitu mengambil data yang sesuai dengan $id.

1. Mengambil data dari $id menggunakan find.

* Cth :

public function find($id = null)

    {

        $model = new Kategori\_M();

        $kategori = $model ->find($id);

        print\_r($kategori);

    }

Artinya, pada function find terdapat parameter yang bernilai $id sama dengan null, atau $id mempunyai nilai kosong. Di dalam function find terdapat objek $model yang mengambil nilai dari Kategori\_M(), Kategori\_M berisi $table tblkategori, yaitu mengambil nilai tabel yang dihubungkan ke dalam database adalah tblkategori, dan mempunyai $allowedFileds yaitu kolom kategori dan kolom keterangan. Lalu $kategori akan memanggil $model yang menjalankan $model yang berisi Kategori\_M dan juga menjalankan perintah find() untuk mengambil data pada $id. Nilai dari $id ini diambil dari $value['idkategori'] karena pada penulisan hyperlink $value['kategori'] mengisi nilai dari $id. Mengecek nilai dari $kategori menggunakan print\_r. Apabil hasil dari print\_r pada $kategori sudah sesuai dan tidak error, maka pengambilan data melalui $id sudah betul.

1. Menyambungkan function find agar dapat terhubung ke dalam view update pada kategori setelah dilakukan pengambilan data melalui $id dan menambahkan $data.

* Cth :

public function find($id = null)

    {

        $model = new Kategori\_M();

        $kategori = $model ->find($id);

        $data = [

            'judul' => 'UPDATE DATA',

            'kategori' => $kategori

        ];

*return* view("kategori/update", $data);

    }

Artinya, pada function find terdapat parameter dengan nilai $id yang berisi null atau $id artinya kosong. Di dalam function find terdapat objek $model yang mengambil nilai dari Kategori\_M() yang di dalamnya terdapat$table tblkategori, artinya tabel yang dihubungkan pada model ke dalam database adalah tblkategori, lalu pada $kategori menjalankan $model yang melakukan perintah mengambil data / find data dari $id. Di dalam $data terdapat 'judul' yang berisi tulisan UPDATE DATA dan 'kategori' yang mengambil nilai dari $kategori. Lalu memanggil view yang terdapat pada file update yang ada di dalam folder kategori dan juga menjalankan $data yang berisi 'judul' dan 'kategori' yang mengambil nilai dari $kategori.

1. Menambahkan value di dalam form Kategori dan form Keterangan dengan mengambil data dari $kategori.

* Cth :

<form action="" method="post">

    Kategori : <input type="text" name="kategori" value="

<?= $kategori['kategori']?>" required>

    <br>

    Keterangan : <input type="text" name="keterangan" value="

<?= $kategori['keterangan']?>" required>

    <br>

   <input type="hidden" name="idkategori" value="

<?= $kategori['idkategori']?>">

    <br>

    <input type="submit" name="simpan" value="SIMPAN">

</form>

Artinya, di dalam form dengan method post terdapat form Kategori dengan input bertipe text dengan name kategori dan value yang mengambil nilai dari $kategori['kategori'] sesuai dengan $id yang diambil, form Kategori ini wajib diisi karena bersifat required. Form Keterangan dengan input bertipe text dengan name keterangan dan value yang mengambil nilai dari $kategori['keterangan'] sesuai dengan $id yang diambil, form Keterangan wajib diisi karena bersifat required. Dan terdaapat input terakhir yaitu form Idkategori dengan input bertipe hidden agar idkategori yang dipilih tidak tampil, lalu name nya adalah idkategori dan value mengambil nilai dari ?kategori['idkategori'] sesuai dengan $id yang dipilih. Setelah itu terdapat input bertipe submit dengan name SIMPAN. Karena di dalam form Kategori dan form Keterangan terdapat value yang mengambil nilai dari $kategori, maka di dalam form sudah terdapat nilai dari $kategori['kategori'] dan $kategori['keterangan'] sesuai dengan $id yang dipilih. Mengambil nilai menggunakan $kategori karena di dalam $kategori pada file Kategori.php di dalam Admin menjalankan $model yang memanggil Kategori\_M dan menjalankan perintah untuk mengambil data sesuai dengan $id yang dipilih.

1. Menyambungkan form yang ada di dalam update.php pada views ke dalam function update.

* Cth :

<form action="<?= base\_url()?>/Admin/Kategori/update" method="post">

    Kategori : <input type="text" name="kategori" value="

<?= $kategori['kategori']?>" required>

    <br>

    Keterangan : <input type="text" name="keterangan" value="

<?= $kategori['keterangan']?>" required>

    <br>

    <input type="hidden" name="idkategori" value="

<?= $kategori['idkategori']?>">

    <br>

    <input type="submit" name="simpan" value="SIMPAN">

</form>

Artinya, terdapat form dengan action yang mengarahkan pada base\_url(), base\_url berisi alamat localhost/restoran-ci4 lalu disambung dengan /Admin/Kategori/update, jadi form tersebut akan menjalankan action localhost/restoran-ci4/Admin/Kategori/update. Form ini juga mempunyai method post yang di dalamnya terdapat, form Kategori dengan input bertipe text dan name kategori serta value yang mengambil nilai dari $kategori['kategori'], lalu form Keterangan dengan input bertipe text name keterangan dan value yang mengambil nilai dari $kategori['keterangan'], dan input bertipe hidden artinya hasil dari form ini diesembunyikan dengan name idkategori dan value yang mengambil nilai dari $kategori['idkategori']. Serta input bertipe submit yang name nya adalah simpan dan value SIMPAN. Jadi ketika form dikirimkan / ketika submit SIMPAN di klik, maka form akan mengarahkan / menjalankan tindakan ke halaman action / localhost/restoran-ci4/Admin/Kategori/update, yang artinya halaman yang ditampilkan adalah isi dari function update pada Kategori di dalam Admin.

1. Membuat function update di dalam controller Kategori.php yang membaca nilai dari $\_POST.

* Cth :

public function update()

    {

        echo "<pre>";

        print\_r($\_POST);

        echo "</pre>";

    }

Artinya, di dalam function update terdapat print\_r untuk membaca nilai dari $\_POST dan menampilkan apa yang tertulis di dalam $\_POST. Lalu <pre> digunakan untuk merapikan format tulisan agar lebih mudah dibaca.

1. save() adalah perintah yang dapat digunakan untuk insert data / memasukkan data dan update data / mengubah data. Apabila di dalam model menambahkan primari key maka akan dianggap sebagai update data dan jika di dalam model tidak terdapat primary key, maka akan dianggap sebagai insert data.
2. Menjalankan perintah update data / mengubah data di dalam function update.

* Cth :

public function update()

    {

        $model = new Kategori\_M();

        $model -> save($\_POST);

*return* redirect()->to(base\_url()."/Admin/Kategori");

    }

Artinya, karena action / tindakan pada form diarahkan ke Admin/Kategori/update atau diarahkan ke function update, maka nilai dari $\_POST tersimpan di dalam function update. Di dalam function update terdapat objek $model yang menjalankan Kategori\_M, di dalam Kategori\_M berisi $table tblkategori, artinya tblkategori adalah tabel yang dihubungkan oleh model ke dalam database. Dan terdapat $primaryKey adalah idkategori, karena pada tblkategori idkategori berperan sebagai primarykey. Lalu $model akan menjalankan perintah untuk save, karena terdapat primary key pada Kategori\_M di dalam objek $model, maka save akan dianggap sebagai update data. Sehingga save akan menjalankan perintah update data yang nilainya diambil dari $\_POST. Apabila selesai mengupdate data, maka halaman akan dikembalikan ke dalam base\_url() yang berisi alamat localhost/restoran-ci4, lalu disambungkan ke halaman /Admin/Kategori, sehingga setelah update data maka akan menjalankan halaman localhost/restoran-ci4/Admin/Kategori.

1. Merapikan link yang dipanggil pada <a href>.

* Cth :

<li>

    <a href="<?= base\_url() ?>/Admin/Kategori">Select</a>

</li>  -> sebelum dirapikan

<li>

    <a href="<?= base\_url('/Admin/Kategori') ?>">Select</a>

</li>  -> setelah dirapikan

Artinya, di dalam <li> terdapat hyperlink yang memanggil base\_url, pada <li> sebelum diedit, /Admin/Kategori diletakkan di luar dari base\_url(), tetapi di dalam link yang sudah diedit /Admin/Kategori diletakkan di dalam base\_url(), dan untuk peletakannya diberi petik satu karena petik dua sudah digunakan untuk <a href>.

1. Membuat file baru yaitu AdminPage.php. Lalu menambahkan \Admin ke dalam namespace untuk pemanggilan karena file AdminPage.php diletakkan di dalam folder Admin, sehingga menjadi <?php namespace App\Controllers\Admin; . Karena memanggil BaseController yang digunakan, sehinggga menambahkan use\App\Controllers\BaaseController; . Serta nama dari class harus sama seperti nama file, yaitu AdminPage.
2. Mengatur route dari AdminPage.

* Cth :

$routes->add('/', 'Admin\AdminPage::index');

Artinya, di dalam $routes terdapat add yang memanggil '/' dengan hasil Admin/AdminPage::index. Artinya apabila tanda '/ ' yang mengakses halaman utama web diketikkan ke dalam URL, maka URL akan memanggil Admin\AdminPage::index, memanggil method index pada controller AdminPage di dalam Admin.

* pemanggilan URL : localhost/restoran-ci4/Admin

Artinya, karena URL yang dituliskan adalah localhost/restoran-ci4/Admin, maka URL akan memanggil method index yang ada pada controller AdminPage di dalam Admin. Setelah /Admin tidak perlu diketikkan tanda '/' karena tanpa tanda '/' sudah membaca tanda '/', jadi jika diketikkan tanda '/' setelah Admin, maka tanda '/' akan menjadi dobel sehingga muncul error, object not found.

1. Mengedit beberapa tampilan menu di dalam admin.php dengan menambahkan menu Home, Kategori, dan Menu. Serta memindahkan menu <li> yang berisi Insert ke dalam file select.php

* Cth :

<nav>

        <ul>

            <li><a href="<?= base\_url('/Admin') ?>">Home</a></li>

            <li><a href="<?= base\_url('/Admin/Kategori') ?>">Kategori</a></li>

            <li><a href="<?= base\_url('/Admin/Menu/select') ?>">Menu</a></li>

        </ul>

    </nav>

Artinya, di dalam admin.php mempunyai <ul> yang terdapat 3 <li> / menu yaitu Home, Kategori, dan Menu. Di dalam menu Home, terdapat hyperlink yang memanggil base\_url dengan alamat localhost/restoran-ci4 lalu disambungkan ke dalam /Admin, karena URL yang dipanggil adalah localhost/restoran-ci4/Admin dan di dalam route sudah diatur jika URL yang dituliskan seperti itu, maka akan menjalankan file AdminPage dalam folder Admin. Jadi menu Home akan menjalankan file AdminPage. Di dalam menu Kategori terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 dan disambungkan ke dalam /Admin/Kategori, sehingga URL akan memanggil localhost/restoran-ci4/Admini/Kategori, dimana sudah diatur jika URL ditulis seperti itu, maka akan menjalankan function read pada file Kategori di dalam Admin. Dan pada menu Menu terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 yang disambungkan dengan /Admin/Menu/Select, sehingga URL akan memanggil alamat localhost/restoran-ci4/Admin/Menu/select, dimana jika URL yang dituliskan seperti itu, maka akan memanggil select pada file Menu di dalam Admin.

* pada file select.php :

<?= $this->extend('template/admin') ?>

<?= $this->section('content') ?>

<li><a href="<?= base\_url('/Admin/Kategori/create') ?>">Insert</a></li>

Artinya, di dalam select.php memanggil admin yang berada di dalam template yang berisi menu list Home, Kategori, dan Menu. Karena di dalam admin.php pada template diberi <?= $this->section('content') ?>, sehingga apabila memanggil template/admin dan untuk isi setelah template/admin, maka harus diberi section('content'), sehingga semua isi dibawah template/admin akan dijalankan. Dan di dalam menu Insert terapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 dan disambungkan dengan /Admin/Kategori/create, sehingga jika URL dituliskan seperti itu, maka akan menjalankan function create yang ada pada controller Kategori di dalam Admin.

1. enctype=multipart/form-data berfungsi untuk apabila ingin menambahkan form dengan input yang bertipe file, maka harus menggunakan enctype=multipart/form-data untuk file yang akan dikirim melalui form method post, jika tidak menambahkan enctype, maka file tidak dapat dikirim.
2. Menambahkan form dengan input yang bertipe file untuk mengirim file.

* Cth :

<form action="<?= base\_url('/Admin/Menu/insert')?>" method="post" enctype="multipart/form-data">

    Gambar : <input type="file" name="gamber" required>

    <br>

    <input type="submit" name="simpan" value="SIMPAN">

</form>

Artinya, di dalam form terdapat action yang memanggil base\_url() dengan alamat localhost/restoran-ci4 yang disambungkan dengan /Admin/Menu/insert, jadi artinya form tersebut akan menjalankan localhost/restoran-ci4/Admin/Menu/insert. Di dalam form dengan method post, terdapat enctype="multipart/form-data", yang artinya di dalam form tersebut ada form Gambar dengan input yang bertipe file, dan enctype ini digunakan untuk mengirimkan file tersebut. Dan terdapat input dengan type submit yang mempunyai name simpan dan value SIMPAN. Jadi apabila submit SIMPAN di klik, maka form akan mengarahkan / menjalankan tindakan ke halaman action yaitu localhost/restoran-ci4/Admin/Menu/insert, dimana halaman yang ditampilkan adalah function insert dari Menu pada Admin.

1. Membaca file yang telah diupload oleh form.

* Cth :

public function insert()

    {

       $file = $this->request->getFile('gambar');

       var\_dump($file);

    }

Artinnya, di dalam function insert terdapat $file yang menjalankan request berisi perintah getFile / mengambil file oleh gambar. Karena action dari form yang mempunyai input bertipe file dan name gambar dijalankan ke dalam function insert, maka untuk mengambil file yang dikirimkan / getFile akan menjalankan name dari input gambar. Lalu nilai dari $file akan dibaca dan ditampilkan menggunakan var\_dump.

1. getFile() adalah perintah yang digunakan untuk membaca file yang dikirimkan.
2. getName() adalah perintah yang digunakan untuk membaca nama dari file yang dikirimkan.
3. Membaca nama dari gambar yang dimasukkan.

* Cth :

public function insert()

{

   $file = $this->request->getFile('gambar');

   $name = $file->getName();

    echo $name."<br>";

   var\_dump($file);

}

Artinya, di dalam function insert terdapat $file yang membaca file yang dikirimkan oleh form. Lalu terdapat $name yang menjalankan $file yang berisi perintah untuk membaca file yang dikirimkan oleh form, setelah membaca file yang dikirimkan oleh form, maka $name akan menjalankan perintah membaca nama dari file yang dikirim, jadi $name akan membaca nama dari file yang dibaca oleh $file yang dikirimkan dengan form. $name akan dibaca menggunakan echo dan $file akan ditampilkan nilainya menggunakan var\_dump.

1. move('./nama\_folder'), move digunakan untuk mengupload file ke dalam folder yang diatur.
2. Mengupload gambar ke dalam folder upload.

* Cth :

public function insert()

{

   $file = $this->request->getFile('gambar');

   $name = $file->getName();

   $file->move('./upload');

   echo $name."SUDAH DIUPLOAD";

}

Artinya, di dalam function insert terdapat $file yang membaca file yang dikirimkan oleh form, lalu di dalam $name akan membaca $file yang menjalankan function getName yaitu memanggil nama dari file pada $file, lalu $file akan menjalankan function perintah move untuk mengupload gambar yang ada di dalam $file, gambar akan diupload ke dalam folder upload, karena folder upload diletakkan di luar dari folder ci4, maka pada move diberi (./) untuk menunjukkan bahwa folder nya berada di luar dari ci4 dan membuat folder baru yaitu upload. Apabila $file sudah dipindahkan ke dalam upload, maka akan tampil $name SUDAH DIUPLOAD.

1. session adalah data yang disimpan di dalam server dan dapat digunakan secara global di server tersebut.
2. Membuat file User.php di dalam folder Admin, lalu menambahkan menu User di dalam admin.php pada folder template.

* Cth :

<li><a href="<?= base\_url('/Admin/User') ?>">User</a></li>

Artinya, di dalam menu User terdapat hyperlink yang memanggil base\_url() berisi alamat localhost/restoran-ci4, lalu disambungkan dengan /Admin/User, jadi apabila hyperlink di klik, sehingga akan menjalankan halaman localhost/restoran-ci4/Admin/User.

1. Untuk pembuatan session pada codeigniter harus diawali dengan inisialisasi, inialisasi artinya pemberian nilai awal atau menyalakan session, pemberian nilai awal pada pembuatan session yaitu dengan menambahkan code yaitu :

$session = \Config\Services::session();

1. Memasukkan data pada session.

* Cth :

public function create()

    {

        $session = \Config\Services::session();

        $tbluser = [

            'user'  => 'koki',

            'email' => 'koki@gmail.com',

            'level' => 'koki'

        ];

        $session->set($tbluser);

    }

Artinya, di dalam function create, terdapat $session yang digunakan untuk inisialisasi session / menyalakan session, lalu memasukkan data - data di dalam $tbluser, terdapat data user dengan value koki, data email dengan value koki@gmail.com, dan data level dengan value koki. Lalu $session menjalankan perintah set, yaitu untuk memasukkan data yang ada pada $tbluser.

1. get() di dalam session digunakan untuk mengambil nilai dari session.

* Cara penulisan : $session -> get('nama\_key'); , jadi $session akan menjalankan perintah get yaitu untuk mengambil nilai dari key yang dimasukkan.

1. Mengambil nilai dan menampilkan nilai dari session.

* Cth :

public function read()

    {

        $session = \Config\Services::session();

        echo $session -> get('user');

    }

Artinya, di dalam function read terdapat $session untuk menyalakan session / inisialisasi session, lalu echo digunakan untuk menampilkan $session yang menjalankan perintah mengambil data dari key 'user' yang mempunyai value koki, jadi session akan mengambil nilai dari user dan akan menampilkan valuenya yaitu koki. Pada pengambilan nilai, harus ditambahkan inisialisasi agar tidak terjadi error. Dan di dalam get() nilai yang diambil adalah key, jadi key akan diambil lalu akan menampilkan value.

1. Inisialisasi session wajib digunakan ketika akan menjalankan session, apabila tidak menambahkan inisialisasi waktu menjalankan session, maka akan terjadi error.
2. remove() pada session digunakan untuk menghapus data dar key yang dimasukkan.

* Cara penulisan : $session -> remove('nama\_key'); , jadi $session akan menjalankan perintah remove yaitu untuk menghapus data dari key yang dimasukkan.

1. Melakukan penghapusan data sesuai key pada session.

* Cth :

public function hapus()

    {

        $session = \Config\Services::session();

        $session -> remove('level');

    }

Artinya, di dalam function hapus terdapat inisialisasi pada $session yang digunakan untuk menyalakan session, lalu terdapat $session yang menjalankan perintah remove / perintah menghapus data pada level. Apabila jika data sudah diremove / dihapus, maka data akan terhapus, tetapi jika data sudah di set lagi, maka data akan muncul kembali.

1. destroy() pada session yaitu perintah yang digunakan untuk menghapus semua session yang disimpan di session saat ini. Jadi apabila session destroy() digunakan, maka session akan kosong, karena sudah dihapus oleh session.
2. Perbedaan destroy() dan remove() pada session, jika destroy() artinya menghapus semua session yang disimpan pada session saat ini, tetapi jika remove(), remove() akan menghapus session sesuai dengan key yang diremove / dihapus.
3. Menghapus semua session yang disimpan menggunakan destroy().

* Cth :

public function destroy()

    {

        $session = \Config\Services::session();

        $session -> destroy();

    }

Artinya, di dalam function destroy() terdapat $session yang berisi inisialisasi session / untuk menyalakan session. Lalu pada $session akan menjalankan perintah destroy, yaitu perintah untuk menghapus semua session yang disimpan, berarti session yang dihapus adalah session yang mempunyai key user, email, dan level karena session tersebut yang disimpan di dalam session.

1. Membuat variabel dengan visibility protected.

* Cth : protected $session = null;

Artinya, terdapat variabel $session yang bernilai kosong dengan visibility protected, jadi jika ingin menggunakan nya harus dengan menambahkan $this->session.

1. function \_\_construct, artinya jika class tersebut dijalankan maka otomatis variabel tersebut akan dijalankan.
2. Membuat function \_\_construct yang berisi inisialisasi dari session / untuk menyalakan session.

* Cth :

public function \_\_construct()

    {

        $this->session = \Config\Services::session();

    }

Artinya, terdapat function construct, function construct akan otomatis jalan ketika class dipanggil, di dalam function \_\_construct terdapat $thist->session artinya, mengambil nilai dari $session tetapi letaknya di luar dari function \_\_construct, jadi nilai dari $session = null diisi dengan inisialisi session, lalu mengisi nilai dari $this->session dengan inisialisasi session / menyalakan session. Karena setiap session akan digunakan perlu untuk dinyalakan, agar tidak menulis berulang kali, maka $this->session diisi dengan inisialisasi session. Jadi setiap session akan digunakan dan untuk menyalakan sessio hanya dengan menulis $this->session, maka session telah dinyalakan.

1. Validasi digunakan untuk apakah data diizinkan masuk ke dalam database atau tidak.
2. Membuat $validationRules yang berisi peraturan - peraturan yang digunakan.

* Cth :

protected $validationRules = [

            'kategori' => 'alpha\_numeric\_space|min\_length[3]|

is\_unique[tblkategori.kategori]'

    ];

Artinya, $validationRules adalah variabel yang berisi peraturan - peraturan untuk validasi, di dalam $validationRules, mengatur pada kolom kategori, rulesnya adalah, alpha\_numeric\_space artinya, kategori tidak boleh menggunakan simbol, hanya boleh menggunakan alpha / huruf, numeric / nomor, space /spasi. Lalu min\_length['3'], artinya isi minimal adalah 3. is\_unique artinya, unik dan tidak boleh sama, yang diatur adalah pada tblkategori kolom kategori.

1. Membuat $validationMessages yang berisi pesan - pesan apabila data yang dimasukkan tidak sesuai dengan peraturan atau data yang dimasukkan memenuhi $validationRules sehingga akan ditampilkan pesan error.

* Cth :

protected $validationMessages = [

    'kategori' =>[

        'alpha\_numeric\_space' => 'mohon maaf, di dalam pengisian kategori, tidak boleh menggunakan simbol,

        harus diisi dengan huruf, angka, dan spasi.',

        'min\_length' => 'Isi minimal adalah 3 huruf',

        'is\_unique' => 'Nama kategori sudah terdaftar'

    ]

];

Artinya, $validationMessages berisi pesan - pesan apabila data yang dimasukkan tidak sesuai dengan peraturan, mengatur kolom kategori, pesan error adalah apabila data yang dimasukkan menggunakan simbol, maka pesan yang ditampilkan adalah pesan pada alpha\_numeric\_space yaitu mohon maaf, di dalam pengisian kategori tidak boleh menggunakan simbol. Apabila data yang dimasukkan kurang dari 3, maka pesan yang ditampilkan adalah pesan pada min\_length, yaitu Isi minimal adalah 3 huruf. Dan apabila data yang dimasukkan sudah terdaftar di dalam kolom kategori, maka pesan error yang ditampilkan adalah Nama kategori sudah terdaftar.

1. Menyambungkan error ke dalam function insert yang digunakan untuk insert data.

* Cth :

public function insert()

    {

        $model = new Kategori\_M();

*if* ($model -> insert($\_POST)===false) {

            $error = $model->errors();

            echo $error['kategori'];

        } *else* {

*return* redirect()->to(base\_url("/Admin/Kategori"));

        }

    }

Artinya, di dalam function insert terdapat objek $model yang menjalankan Kategori\_M, lalu terdapat pengujian, yaitu $model yang menjalankan perintah untuk memasukkan data pada $\_POST tidak diterima / false, karena data yang dimasukkan tidak memenuhi peraturan, maka akan menjalankan, terdapat $error yang menjalankan $model dengan memanggil perintah errors(), lalu $error akan mengambil nilai error pada kategori, nilai yang diambil adalah nilai pada $validationRules dan $validationMessages. Tetapi apabila data yang dimasukkan benar / sesuai dengan peraturan, maka akan menjalankan resirect, untuk mengembalikan halaman ke dalam tampilan base\_url yang memanggil alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Kategori, sehingga apabila berhasil insert data / memasukkan data, maka halaman akan dialihkan / diarahkan pada localhost/restoran-ci4/Admin/Kategori dimana jika URL tersebut ditulis akan menjalankan function read pada Kategori di dalam Admin.

1. Melakukan instalasi / download pada bootstrap karena bootstrap akan digunakan secara offline jadi harus diinstal terlebih dahulu.
2. Untuk melakukan instalasi harus membuka web getbootstrap.com lalu menekan tombol download lalu di dalam compiled CSS and JS ada tombol download dan mendownload bootstrap itu. File bootstrap sudah diinstal.
3. Menyambungkan link bootstrap ke dalam admin.php di folder template.

* Cth :

<link rel="stylesheet" href="<?=base\_url('/bootstrap/css/bootstrap.min.css') ?>">

Artinya, terdapat link yang berisi base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan link bootstrap, yaitu /bootstrap/css/bootstrap.min.css, jadi, karena file bootstrap juga berada di dalam folder restoran-ci4, maka untuk menjalankannya dengan menggunakan base\_url() yang berisi alamat localhost/restoran-ci4, sehingga pada file admin.php ini sudah disambungkan ke dalam bootstrap dengan link. Karena link bootstrap diletakkan di dalam admin.php yang di dalamnya terdapat section('content'), jadi file yang disambungkan menggunakan template/admin, maka isi harus diletakkan di dalam section('content'), maka dari itu semua file yang disambungkan ke dalam template/admin menjalankan bootstrap karena isi di pada file diletakkan di dalam section('content').

1. Menambahkan navbar di dalam admin.php.

* Cth :

<div class="col">

     <nav class="navbar navbar-light bg-light">

        <a class="navbar-brand" href="<?=base\_url('/Admin') ?>">

Admin Page

</a>

       </nav>

 </div>

Artinya, di dalam <div> col terdapat <nav> dengan class navbar-light, yang artinya tulisan di dalam navbar adalah gelap / hitam dan bg-light yang artinya background di dalam navbar berwarna putih. Lalu di dalam navbar dengan tulisan Admin Page terdapat hyperlink yang menyambungkan ke dalam base\_url() dengan alamant localhost/restoran-ci4 lalu disambungkan ke /Admin, jadi jika navbar di klik, maka akan menjalankan localhost/restoran-ci4/Admin , dimana jika dituliskan seperti itu, maka akan memanggil file AdminPage yang ada di dalam Admin.

1. Membuat <div> class row dengan <div> class col 4 dan <div> class col 8.
2. Di dalam <div> class col-4 berisi list group dengan item Kategori, Menu, dan User.

* Cth :

<div class="col-4 mt-4">

    <div class="card" style="width: 18rem;">

        <ul class="list-group list-group-flush">

            <li class="list-group-item"><a href="<?= base\_url('/Admin/Kategori') ?>">Kategori</a></li>

            <li class="list-group-item"><a href="<?= base\_url('/Admin/Menu') ?>">Menu</a></li>

            <li class="list-group-item"><a href="<?= base\_url('/Admin/User') ?>">User</a></li>

        </ul>

    </div>

</div>

Artinya, terdapat <div> class dengan kolom 4 dan margin top 4, yang di dalamnya terdapat <div> card yang berisi <ul> list-group sebagai tempat untuk item yang berada di dalamnya. Di dalam list-group pada card terdapat 3 item, yaitu, Kategori, yang di dalamnya terdapat hyperlink dan memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Kategori, dimana jika item Kategori di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Kategori.Item Menu, yang di dalamnya terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Menu, dimana jika item Menu di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Menu. Item User, yang di dalamnya terdapat hyperlink dan memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/User, dimana jika item User di klik, maka akan menjalankan localhost/restoran-ci4/Admin/User.

1. Menambahkan button TAMBAH DATA.

* Cth :

<a class="btn btn-primary" href="<?= base\_url('/Admin/Kategori/create') ?>" role="button">

    TAMBAH DATA

</a>

Artinya, terdapat tag <a> yang berisi button dengan class btn-primary, yaitu button akan berwarna biru, di dalam button TAMBAH DATA terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu diambungkan dengan /Admin/Kategori/create, jadi jika button TAMBAH DATA di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Kategori/create.

1. Paging adalah membagi data menjadi beberapa halaman.
2. Cara untuk melakukan paging pada codeigniter yang pertama adalah memasukkan library nya ke dalam controller yang akan dipasangi pagingnya. Setelah memasukkan library, lalu mengisi data di dalam variabel yang akan dijalankan pada view. Lalu untuk menampilkan pagination, di dalam view ditambahkan <?= $pager -> links() ?>

* Code untuk paging adalah : $pager = \Config\Services::pager();
* Data yang dimasukkan pada $data =

$data = [

    'kategori' => $model -> paginate(3),

    'pager' => $model -> pager

];

Artinya, di dalam $data, ada $model yang menjalankan Kategori\_M, dimana terdapat $table tblkategori, artinya tabel yang dihubungkan ke dalam database adalah tblkategori, dan terdapat key 'kategori' dan key 'pager', 'kategori' dan 'pager' adalah data yang akan dijalankan pada view. Di dalam key kategori berisi $model yang menjalankan perintah untuk paginate / paging, sehingga halaman akan diatur dengan menampilkan 3 data pada setiap halamannya, lalu terdapat key pager berfungsi untuk menampung library paginationnya.

* Menampilkan pagination di dalam view = <?= $pager -> links() ?>

Artinya, karena data yang ditampilkan adalah data pada view select.php, maka di dalam view select.php ditambahkan <?= $pager -> links()?> yang berfungsi untuk menampilkan pagination di dalam view.

1. Mengcustom paging dan mengatur tampilan paging adalah dengan menambahkan nama dari group ke dalam $data yang menyimpan key yang dijalankan ke dalam view, lalu di dalam view select yang menjalankan $data, ditambahkan nama group yang ditambahkan di dalam $data ke dalam <?= $pager->links(group1)?>, lalu codeigniter sudah menyediakan code untuk paging, jadi kita mengcopas code ke dalam file beru di dalam template, yaitu file paging.php, setelah membuat file paging.php, untuk menjalankan file nya, maka ditambahkan nama apabila nama tersebut dipanggil maka akan menjalankan file paging.php, dan memasukkan nama yang menjalankan file paging ke dalam <?= $pager -> links()?> yang ada di dalam view select.

* Menambahkan nama dari group ke dalam key =

$data = [

    'kategori' => $model -> paginate(3, 'group1'),

    'pager' => $model -> pager

];

Artinya, di dalam $data, ada $model yang menjalankan Kategori\_M, dimana terdapat $table tblkategori, artinya tabel yang dihubungkan ke dalam database adalah tblkategori, dan terdapat key 'kategori' dan key 'pager', 'kategori' dan 'pager' adalah data yang akan dijalankan pada view. Di dalam key kategori berisi $model yang menjalankan perintah untuk paginate / paging, sehingga halaman akan diatur dengan menampilkan 3 data pada setiap halamannya, dan menjalankan 'group 1', grup digunakan untuk menentukan links() nya mau dimodel seperti apa,lalu terdapat key pager berfungsi untuk menampung library paginationnya.

* Menjalankan pagination di dalam view = <?= $pager -> links('group1', 'bootstrap') ?>

Artinya, di dalam $pager berisi links yang menjalankan group1 dengan dimodel seperti 'bootstrap'. Apabila bootstrap dituliskan, maka akan memanggil file / menjalankan file App\Views\template\paging, jadi, apabila setelah membuat file paging.php yang berisi pengaturan dari paging, dan setelah selesai melakukan pengaturan di dalam Pager.php, pengaturan yang dilakukan adalah jika 'bootstrap' dipanggil ke dalam links, maka akan menjalankan/ memanggil paging.php yang ada di dalam template.

* Pengaturan pada Pager.php =

public $templates = [

    'bootstrap'     => 'App\Views\template\paging',

];

Artinya, di dalam $templates terdapat 'bootstrap' dengan hasil App\Views\template\paging, jadi ketika 'bootstrap' dituliskan, maka akan menjalankan paging yang ada di dalam template. Karena 'bootstrap' dijalankan pada links() maka links akan menjalankan paging yang berisi pengaturan paging.

1. View cell adalah bagian terkecil dari view yang dapat digunakan untuk memotong motong bagian dari view.
2. Penggunaan dari view cells tidak boleh menggunakan echo apabila ingin memanggil view, tetapi menggunakan return.
3. Memanggil function option karena function option memanggil file yang menjalankan view\_cells.

* Cth :

<div class="row">

    <?= view\_cell('\App\Controllers\Admin\Menu::option')?>

</div>

Artinya, di dalam class row, terdapat function view\_cell yang memanggil function option yang ada di dalam App\Controllers\Admin\Menu. Jadi, function option itu memanggil view\_cell yang menjalankan view.

1. Membuat function option yang memanggil view yang menjalankan view\_cell.

* Cth :

public function option()

    {

      $model = new Kategori\_M();

      $kategori = $model -> findAll();

      $data = [

         'kategori' => $kategori

      ];

*return* view('template/option', $data);

    }

Artinya, karena option Menu dipanggil oleh file form.php di dalam views untuk menjalankan view\_cell, maka di dalam function option pada file Menu.php akan memanggil view. Pada objek $model menjalankan Kategori\_M yang berisi $table yaitu tblkategori, artinya table yang dihubungkan oleh model ke dalam database adalah tblkategori. Dan di dalam $kategori berisi $model yang menjalankan perintah findAll() atau perintah untuk mengambil data yang ada pada tblkategori. Untuk mengirimkan data ke dalam view, harus mempunyai variabel yang berisi data agar data dapat dikirim ke dalam view. Maka dari itu, di dalam $data terdapat key 'kategori' dengan value yaitu $kategori, $kategori menjalankan perintah untuk memilih data yang diambil dari $model. Lalu menjalankan view yang memanggil option yang ada di dalam template dan juga memanggil $data.

1. Mengambil nilai pada view karena function option yang berisi view\_cell dan di dalam function option menjalankan view yaitu pada option.php di dalam template.

* Cth :

<select name="kategori" id="kategori">

    <?php *foreach* ($kategori as $key => $value) : ?>

        <option value="<?= $value['idkategori']?>"><?=$value['kategori'] ?></option>

    <?php *endforeach*; ?>

</select>

Artinya, di dalam select dengan name kategori dan id kategori, terdapat foreach yang melakukan perulangan untuk option dengan $kategori sebagai $key dan terdapat $value. Lalu di dalam option mengambil value dari nilai pada $value['idkategori'] dan mengambil nilai untuk list dari $value['kategori'].

Penjelasan untuk saya agar ingat, untuk menggunakan view\_cell harus membuat file untuk dipanggil, isinya boleh belum sempurna, tetapi ada file yang dapat dipanggil, lalu menyempurnakan isi ketika pemanggilan sudah berhasil, seperti, membuat file option.php pada folder template di dalam views, lalu membuat function option di dalam Menu.php pada Admin, di dalam function option pada Menu, awalnya berisi memanggil view('template/option') saja, setelah itu di dalam file form.php, memanggil function option pada menu dengan menggunakan view\_cell, jadi function option itu adalah function yang berisi view\_cell yang menjalankan view di option di dalamnya, setelah menentukan pemanggilan dari view\_cell, pada function option melakukan penambahan isi, yaitu dengan membuat objek $model dan mengambil data dari Kategori\_M, lalu memberi perintah untuk mengambil data / findAll pada $model. Karena untuk mengirimkan data harus mempunyai variabel dengan array assosiatif di dalamnya yang berisi data yang akan dikirimkan ke dalam view, maka di dalam function option terdapat $data yang berisi key 'kategori' yang mengambil nilai dari $kategori. Lalu menjalankan view, karena untuk menjalankan view\_cell harus memanggil menggunakan return tidak boleh menggunakan echo, jadi untuk menjalankan function option yang berisi view\_cell untuk menjalankan view, maka harus menggunakan return view('template/option'), jadi view\_cell memanggil view option pada template. Di dalam option pada template berisi select dengan list option, yang mengambil nilai value dari $value['idkategori'] dan dengan list yang mengambil nilai dari $value['kategori'].

1. session flashdata digunakan untuk menampilkan pesan / notifikasi saat melakukan suatu proses.
2. $session -> setFlashdata() digunakan untuk mengatur pesan yang ditampilkan ketika melakukan proses.
3. $session -> getFlashdata() digunakan untuk mendapatkan / mengambil data pesan ketika suatu proses dilakukan.
4. Penulisan dari $session -> getFlashdata() bisa diganti menjadi session() -> getFlashdata() karena apabila menjalankan session, maka harus menyalakan session terlebih dahulu, maka jika menggunakan $session -> getFlashdata, session masih berupa variabel, jadi harus dinyalakan dulu / di inisialisasi terlebih dulu, sehingga jika mengganti nya ke dalam bentuk fungsi / session() maka tidak perlu untuk menyalakan session karena function sudah mencari data nya sendiri.
5. Mengatur pesan yang ditampilkan apabila error menggunakan session() -> setFlashdata().

* Cth :

public function insert()

    {

        $model = new Kategori\_M();

*if* ($model -> insert($\_POST)===false) {

            $error = $model->errors();

            session() -> setFlashdata('info', $error['kategori']);

*return* redirect()->to(base\_url("/Admin/Kategori/create"));

        } *else* {

*return* redirect()->to(base\_url("/Admin/Kategori"));

        }

    }

Artinya, di alam function insert terdapat objek $model yang memanggil Kategori\_M, di dalam Kategori\_M terdapat $table tblkategori, sehingga tabel yang dihubungkan oleh model ke dalam database adalah tblkategori. Lalu terdapat pengujian yang memanggil Kategori\_M dan menjalankan perintah memasukkan data dari $\_POST, jika data yang dikirimkan salah, maka akan menjalankan. $error yang memanggil $model yang menjalankan perintah errors(), lalu setealh itu akan menjalankan function session() yang menjalankan perintah untuk mengatur pesan apabila dengan nama key 'info' lalu menjalankan $error yang mengambil nilai error yang sudah diatur pada key 'kategori', lalu setelah selesai mengambil nilai error, maka halaman akan dikembalikan ke dalam base\_url() dengan alamat localhost/restoran-ci4 yang disambungkan dengan /Admin/Kategori/create, jadi jika selesai mengambil nilai error, maka halaman akan diarahkan ke localhost/restoran-ci4/Admin/Kategori/create. Tetapi apabila data yang dimasukkan benar / sesuai dengan peraturan, maka akan menjalankan redirect dengan mengarahkan ke halaman base\_url() dengan alamat localhost/restoran-ci4 dan disambungkan dengan /Admin/Kategori, sehingga jika berhasil insert data / memasukkan data, maka halaman akan dialihkan pada halaman localhost/restoran-ci4/Admin/Kategori dimana jika URL tersebut ditulis, maka akan menjalankan function read pada Kategori di dalam Admin dan menampilkan data - data menu.

1. Menampilkan pesan error dengan menggunakan session() -> getFlashdata().

* Cth :

echo session() -> getFlashdata('info');

Artinya, echo akan menampilkan function session yang menjalankan perintah untuk mendapatkan pesan yang sudah diatur di dalam session() -> setFlashdata dengan nama key 'info'.

**Saya Belum Mengerti**