|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran CodeIgniter 4 Video 1 – 9  (Tugas minggu ke - 17) | 95 |

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

1. CodeIgniter adalah sebuah framework yang terdiri dari class class yang dibuat dengan tujuan untuk memudahkan waktu pengerjaan dari sebuah web dan agar dapat membuat tampilan dari web dengan lebih baik.
2. CodeIgniter menggunakan 3 sistem, yaitu controller, model, dan view.
3. Fungsi dari controller adalaha menerima semua permintaan dari user. Apabila pada controller membutuhkan database maka akan diambil dari model, fungsi dari model adalah kode kode untuk model bisnis dan data serta disambungkan ke dalam database, untuk insert data, update data, delete data, dan search data. Lalu data akan ditampilkan menggunakan view.
4. Melakukan proses instalasi dari CodeIgniter dengan mendownload file pada web codeigniter.com lalu file download akan berupa zip, dan setelah itu, membuat folder restoran-ci4 di dalam htdocs lalu mengekstrak file zip ke dalam restoran-ci4.
5. Mengatasi error pada saat membuka file ci-4 pada codeigniter, error diatasi dengan membuka System>View>CodeIgniter.php.

Lalu mencari code perintah :

locale\_set\_default($this->config->defaultLocale ?? 'en');

Dan mengganti code perintahnya dengan seperti ini :

*if*( function\_exists('locale\_set\_default' ) ) :

     locale\_set\_default($this->config->defaultLocale ?? 'en');

*endif*;

Save halaman dan refresh page, maka setelah itu, halaman yang muncul adalah welcome to codeigniter 4.0.4

1. Mengcopy file yang ada di dalam folder ci4>public lalu mengcopy nya ke dalam folder restoran ci-4.
2. Mengatasi error yang terjadi setelah mengcopy file public dan mempaste nya ke dalam restoran-ci4. Error terjadi karena pemanggilan path yang dilakukan salah, awalnya pemanggilan path seperti ini, di dalam file index.php path yang error adalah seperti ini,

$pathsPath = realpath(FCPATH . '../app/Config/Paths.php');

karena file index.php sebelumnya berada di dalam folder public dimana folder app dan folder public ada di folder ci4, tetapi karena isi dari public diletakkan diluar folder ci4 dan folder app masih di dalam folder ci4, maka pemanggilannya akan menjadi seperti ini,

$pathsPath = realpath(FCPATH . 'ci4/app/Config/Paths.php');

yaitu mengganti (..) menjadi (ci4) karena letak folder app ada di dalam folder ci4 sementara index.php berada di luar ci4.

1. Cara kerja adalah semua yang ada di dalam URL akan masuk ke dalam controller, dan controller itulah yang akan menampilkan data.

* Cth : Di dalam folder app terdapat folder controller, dan di dalam controller terdapat file home.php, file home dicopy dan dipaste kedalam folder controller dengan nama Kategori.php, di dalam Kategori.php terdapat class yang sebelumnya bernama Home, nama dari class tersebut diganti sesuai dengan nama dari folder yaitu Kategori. Class tersebut akan menampilkan welcome message.

Jadi ketika URL memanggil localhost/restoran-ci4/Kategori, maka akan memanggil file Kategori.php, dan akan memanggil class Kategori, dimana nama class harus sesuai dengan nama dari filenya, lalu hasilnya akan menampilkan welcome message.

1. Membuat function select() di dalam menu dan menampilkannya dengan menambahkan /select ke dalam URL.

* Cth : code penulisan function :

class Menu extends BaseController

{

    public function select()

          {

             echo "<h3>Untuk menampilkan data</h3>";

           }

 }

Artinya, di dalam class Menu, terdapat function select yang berisi tulisan Untuk menampilkan data dan dijalankan menggunakan echo.

pemanggilan URL :

http://localhost/restoran-ci4/Menu/select

Artinya, untuk memanggil / menampilkan function select, maka di dalam URL diberi /select setelah Menu, karena function select terletak di dalam class menu.

1. Membuat function update dengan mengisi parameter, $id=null, dan menampilkannya dengan menambahkan /update ke dalam URL.

* Cth : code penulisan function :

class Menu extends BaseController

{

   public function update($id=null)

      {

            echo "Untuk update data dengan id ".$id;

       }

 }

Artinya, di dalam class Menu, terdapat function yang bernama update dengan nilai parameter dari function adalah $id=null, yang berarti nilai dari $id adalah kosong / belum terisi. Di dalam function update terdapat echo yang akan menjalankan tulisan untuk update data dengan id yang memanggil $id, karena $id masih kosong, maka $id tidak tampil.

pemanggilan URL :

http://localhost/restoran-ci4/Menu/update

Artinya, localhost akan memanggil restoran-ci4, lalu memanggil file dengan nama Menu dan memanggil update yang di dalam Menu terdapat function update, lalu function update akan menjalankan nilai paramater yaitu $id=null.

1. Untuk melakukan pemanggilan dari $id, maka pada URL bisa dituliskan dengan nilai variabel yang diinginkan, secara otomatis nilai dari $id akan berubah.

* Cth : pemanggilan URL :

http://localhost/restoran-ci4/Menu/update/2

Artinya, localhost memanggil restoran-ci4, lalu memanggil file dengan nama Menu dan di dalam file memanggil update yang di dalam update akan menjalankan parameter dari update dan memanggil nilai 2.

1. null / NULL adalah tipe data yang menunjukkan bahwa variabel kosong, variabel belum memiliki nilai / data. Penulisan null dapat menggunakan huruf besar / huruf kecil karena unll merupakan case-insensitive.
2. Case-insensitive artinya bahwa huruf besar dan huruf kecil diartikan sama.
3. Case-sensitive artinya bahwa huruf besar dan huruf kecil diartikan berbeda, maka jika salah penulisan yang jika seharusnya menggunakan huruf kecil tetapi ditulis menggunakan huruf besar, akan terjadi error.
4. Menambahkan $nama di dalam parameter pada function update dan melakukan pemanggilan di dalam URL.

* Cth : code penulisan function :

class Menu extends BaseController

{

   public function update($id=null, $nama=null)

      {

            echo "<h3>Untuk update data dengan id  : $id   , $nama  </h3>";

       }

 }

Artinya, di dalam class Menu terdapat function update dengan nilai pada parameter adalah $id=null dan $nama=null, nilai dari $id dan $nama diisi menggunakan null agar ketika function dipanggil tidak terjadi error, lalu di dalam function update, echo akan menjalankan tulisan Untuk update data dengan id : $id, mengambil nilai dari $id , $nama dan mengambil nilai dari $nama. Nilai dari $id dan $nama akan terisi ketika pada URL dimasukkan nilai tersebut.

pemanggilan URL :

* Cth : http://localhost/restoran-ci4/Menu/update/2/Pisang

Artinya, localhost akan memanggil restoran-ci4, lalu mengambil file yang bernama Menu pada Menu terdapat class dengan nama Menu, di dalam class Menu terdapat function update dengan nilai dalam parameter $id=null dan $nama=null, maka pemanggilan dari niali $id dan $nama dilakukan pada URL, yaitu /2/Pisang, dengan nilai dari $id adalah 2 dan nilai dari $nama adalah Pisang, maka setelah memasukkan nilai pada URL, nilai pada $id dan $nama akan ditampilkan pada echo Untuk update data dengan id : 2(mengambil nilai dari $id) , Pisang(mengambil nilai dari $nama).

1. Pada codeigniter, yang pertama kali dikerjakan adalah controller, dan perintah di dalam controller yang ditulis di dalam alamat URL akan dikerjakan yang selanjutnya akan menampilkan hasil dari perintah tersebut.
2. Php native adalah pembuatan projek mulai dari nol. Sementara framework adalah, kerangka kerja yan gsudah siap pakai, dan kita hanya perlu memanggil fungsinya saja
3. Perbedaan menggunakan php native dan codeigniter adalah ketika membuat aplikasi restoran menggunakan php native, pada menu kategori jika dilihat di dalam folder terdapat beberapa file-file seperti delete, update, insert, dan select. Tetapi jika di dalam codeigniter file file seperti delete, update, insert, dan select akan diwakili menggunakan function.
4. Membuat beberapa function di dalam file Kategori.php yaitu function select untuk menampilkan semua data, function selectwhere untuk menampilkan data yang dipilih, function formInsert untuk menampilkan form insert, function formUpdate untuk menampilkan form update, function update untuk proses update data, dan function delete untuk proses delete data.
5. Pemanggilan function pada Kategori.php adalah :

* Cth : http://localhost/restoran-ci4/Kategori/update

Artinya, localhost akan memanggil folder restoran-ci4, lalu memanggil file Kategori yang di dalam file Kategori terdapat class Kategori, pada class Kategori terdapat function update, dan controller akan menampilkan function update yang berisi tulisan Proses update data. Dan jika di dalam update terdapat parameter maka parameter dipanggil melalui URL.

1. Terjadi error ketika menampilkan file yang berada di dalam folder pada Controllers.
2. Mengatasi error dengan cara memberi tanda titik di depan file env pada ci4, lalu menambahkan CI\_ENVIRONMENT = development di bawah dari # CI\_ENVIRONMENT = production. CI\_ENVIRONTMENT = development berfungsi untuk menampilkan pesan error dan memperbaiki bug ketika adanya kesalahan.
3. Sementara fungsi dari CI\_ENVIRONTMENT = production adalah tidak menampilkan pesan error ketika terjadi kesalahan.
4. namespace dapat digunakan untuk menghindari penamaan class yang sama dengan namespace dapat memberi nama yang sama pada function tetapi pada grup yang berbeda. Karena nama function sama, agar dapat tampil maka perlu dilakukan pengaturan pada pemanggilannya.
5. use pada php digunakan untuk import namespace ke namespace lain dan dapat digunakan untuk aliansing / menyingkat atau memberi nama lain kepada namespace.
6. Memindahkan file kategori.php ke dalam folder Admin, untuk menampilkan file pada Controllers yang terletak di dalam folder, maka harus diatur pengaturan pemanggilannya.

* Cth :

<?php namespace App\Controllers\Admin;

use App\Controllers\BaseController;

Artinya, pada php terdapat namespace dengan direktori file App\Controllers\Admin artinya file tersebut diambil dari App lalu ke Controllers lalu ke Admin dan terdapat use karena class yang digunakan adalah extends dari BaseController maka menggunakan use agar class dapat menggunakan dari BaseController. Perlu ditambahkan use App\Controller\BaseController agar class Kategori dapat menggunakan base dari BaseController.

1. extends digunakan untuk membuat child class (class turunan) dari parent class (class induk).

* Cth :

class Kategori extends BaseController{

}

Artinya, jadi class Kategori adalah class turunan yang mempunyai class induk yaitu BaseController.

1. Routes adalah sebuah permintaan atau request di dalam URL yang kemudian bisa diatur / custom, custom adalah bisa mengatur sesuai dengan keinginan kita.
2. Membuat file Login.php yang berisi echo LOGIN PAGE.
3. Mengatur routes di dalam file routes.php pada folder config>Routes.php, lalu mengedit defaultcontroller pada $routes, yaitu dengan mengubah home menjadi login.

* Cth :

$routes->setDefaultController('Home');  -> Sebelum diedit

$routes->setDefaultController('Login');  -> Sesudah diedit

$routes->get('/', 'Login::index');

Artinya, pada $routes terdapat get yang akan membaca URL yang dituliskan di dalam browser. Lalu terdapat ('/') artinya adalah mengakses halaman utama dari web. Dan yang terakhir adalah penulisan controller dan method / function yang digunakan. Pada $routes, maka akan menjalankan method index yang ada di dalam controller Login.

Pemanggilan :

http://localhost/restoran-ci4/

Artinya, ketika pemangillan pada URL kosong atau pada URL hanya dipanggil restoran-ci4, dalam defaultcontroller jika sebelum diedit adalah tampilan akan diambil dari file Home.php karena itu adalah default pengaturan apabila URL kosong. Tetapi karena defaultcontroller diedit menjadi Login, maka ketika URL kosong / hanya memanggil restoran-ci4 saja, otomatis akan menampilkan Login.php.

1. Melakukan pengaturan tambahan pada $routes, dengan mengatur jalan dari Kategori, ketika Kategori diketikkan ke dalam URL.

* Cth :

$routes->get('Kategori', 'Admin\Kategori::select');

Artinya, di dalam $routes terdapat get, jika diisi dengan get maka get akan membaca URL yang dituliskan di URL browser. Lalu terdapat 'Kategori', kategori adalah halaman utama dari web, Kategori adalah URL yang memanggil, maka ketika Kategori dituliskan di dalam URL akan menjalankan controller dan method. Dan controller yang digunakan adalah Admin\Kategori, atau Kategori yang berada di dalam Admin, sementara method yang digunakan adalah select. Jadi codeigniter akan membaca / memanggil method / function select yang berada pada controller Kategori yang ada di dalam Admin ketika Kategori ditulis di dalam URL.

1. Melakukan pengaturan routing pada $route apabila terdapat parameter yang diisi di dalamnya.

* Cth :

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

Artinya, di dalam $routes pada pengaturan routing, terdapat get yang akan membaca URL yang dituliskan di dalam URL browser. Lalu terdapat Kategori yang merupakan halaman utama dari web, pada Kategori terdapat (:any) untuk menentukan segmen yang berupa semua karakter angka, huruf, atau spesial karakter pada URL. Dan controller Kategori yang terdapat pada Admin dengan method selectwhere, pada selectwhere terdapat /$1 artinya, memasukkan nilai variabel pada parameter ke dalam URL.

1. Untuk memasukkan nilai variabel maka pada routing menggunakan slash (/).

* Cth :

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

Artinya, di dalam $routes yang merupakan nilai dari variabel adalah /$1.

1. Untuk memasukkan Controller maka pada routing menggunakan backslash (\).

* Cth :

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

Artinya, di dalam $routes yang merupakan Controller adalah Admin\Kategori, artinya Kategori yang berada di dalam Admin.

1. (:any) pada routing berfungsi untuk menentukan segmen di dalam URL boleh berupa angka, huruf, atau spesial karakter yang diizinkan ke dalam URL.
2. (:num) pada routing berfungsi untuk menentukan segmen di dalam URL berupa number / angka, sehingga tidak boleh menggunakan huruf / karakter, tetapi hanya menggunakan angka.
3. Menambahkan $nama pada parameter di dalam selectwhere.

* Cth :

public function selectwhere($id = null, $nama = null)

    {

        echo "<h3>Menampilkan data yang dpilih, id yang dipilih adalah : $id, dan nama adalah : $nama</h3>";

    }

Artinya, terdapat function dengan nama selectwhere yang mempunyai nilai parameter di dalamnya adalah $id dengan nilai kosong, dan $nama dengan nilai kosong. Apabila selectwhere dipanggil, maka akan memunculkan tulisan yaitu Menampilkan data yang dipilih, id yang dipilih adalah $id, jadi nilai $id akan diambil dan dimasukkan ke dalam tulisan tersebut, dan nama adalah $nama, hasil dari nama mengambil dari nilai pada $nama.

1. Pengaturan pemanggilan pada route apabila nilai di dalam parameter terdapat 2 variabel.

* Cth :

$routes->get('Kategori/(:any)', 'Admin\Kategori::selectwhere/$1/$p');

Artinya, di dalam $routes pada pengaturan routing, terdapat get yang akan membaca URL yang dituliskan di dalam URL browser. Lalu terdapat Kategori yang merupakan halaman utama dari web, pada Kategori terdapat (:any) untuk menentukan segmen yang berupa semua karakter angka, huruf, atau spesial karakter pada URL. Dan controller Kategori yang terdapat pada Admin dengan method selectwhere, pada selectwhere terdapat /$1 dan /$p artinya, memasukkan nilai variabel pada parameter ke dalam URL, jadi ketika terdapat dua variabel di dalam parameter, untuk pemanggilannya hanya dengan menambahkan /$variabel saja.

1. Di dalam routes terdapat nilai yang dipanggil, pada nilai yang dipanggil dapat dirubah dan terdapat hasilnya.

* Cth :

$routes->get('Kategori/(:any)', 'Admin\Kategori::selectwhere/$1/$p');

Artinya, Kategori adalah nilai yang dipanggil, tulisan dari Kategori dapat dipanggil menjadi Menu atau lainnya, tidak harus dengan nama file atau class tetapi bisa dicustom dengan mangatur dari routes. Dan pada Admin\Kategori::selectwhere/$1/$p adalah hasil dari yang dipanggil, jadi ketika Kategori dituliskan, maka hasil yang dipanggil adalah Admin\Kategori::selectwhere/$1/$p atau memanggil method selectwhere dengan nilai parameter adalah $1 dan $p pada controller Kategori di dalam Admin.

1. Group routes digunakan untuk mengelompokkan routes yang sudah dibuat.
2. Membuat group route pada Admin.

* Cth :

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

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

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

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

});

Artinya, di dalam $route terdapat group route untuk menyambungkan beberapa $routes yang memanggil Admin dan dengan hasil yang dipanggil berdasarkan function ($routes), di dalam function $routes, terdapat beberapa $routes, fungsi add pada setiap $routes adalah untuk memanggil apa yang ada di sebelah kanan dari add, pada $routes pertama, add memanggil Kategori/insert, apabila pada URL diketikkan Kategori/insert maka hasilnya adalah menampilkan method formInsert yang ada pada controller Kategori di dalam Admin. Pada $routes kedua, add memanggil Kategori, jika pada URL diketikkan Kategori saja, maka hasilnya adalah menampilkan method select yang ada di dalam controller Kategori di dalam Admin. Dan $routes yang ketiga adalah, add memanggil Kategori\Updatea, apabila pada URL dituliskan Kategori\update, maka hasilnya akan menampilkan method formUpdate yang ada pada controller Kategori di dalam Admin.

1. Apabila memanggil function yang parameternya terisi, maka $routes yang terdapat nilai variabel di dalam parameter, maka pada $routes di dalam group routes diberi :

* Cth:

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

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

});

Artinya, pada group route memanggil Admin dengan menampilkan hasil dari function $routes yang di dalamnya terdapat $routes yang memanggil Kategori/update, jadi jika dituliskan Kategori/update pada URL akan menampilkan hasil dari method formUpdate pada controller Kategori di dalam Admin. Pada $routes terdapat (:any), karena di dalam parameter pada function terdapat nilai variabel yang diisi, jadi (:any) digunakan untuk menentukan segmen di dalam URL bisa berupa angka, huruf, atau karakter lainnya. Dan /$1 pada $routes adalah variabel pertama yang dipanggil.

1. Membuat folder kategori di dalam views, app>views>kategori, pada folder kategori berisi file select.php dan forminsert.php.
2. Membuat folder template di dalam views, app>views>template, pada folder template berisi file header.php dan footer.php.
3. view() digunakan untuk memanggil file view.

* Cth :

echo view("template/header");

Artinya, view akan memanggil file header pada folder template yang ada di dalam views. Pemanggilan file pada view() tidak menggunakan .php lagi karena pada saat memanggil view() otomatis codeigniter akan langusng memanggil app/views. Penulisan view() tidak menggunakan s, karena view yang dipanggil tunggal.

1. Melakukan pemanggilan view di dalam function select dalam kategori.php yang ada di folder Admin.

* Cth :

public function select()

{

    echo view("template/header");

    echo view("kategori/select");

    echo view("template/footer");

}

Artinya, terdapat function yang bernama select, di dalam function terdapat echo yang melakukan pemanggilan file - file menggunakan view, karena file - file tersebut berada di folder views maka pemanggilan file menggunakan view(). function select memanggil view pada header yang ada di dalam template, isi dari header adalah tag html untuk mengedit bagian title web. Lalu setelah itu memanggil view pada select yang berada di dalam folder kategori, isi dari select adalah tulisan untuk body. Dan yang terakhir memanggil view pada footer yang ada di dalam template, isi dari footer adalah tag penutup /body dan tag penutup /html yang merupakan sambungan dari file header.

1. Menambahkan nav di dalam file header.php dan memberi hyperlink pada setiap nav nya.

* Cth :

<nav>

   <ul>

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

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

     <li><a href="<?= base\_url() ?>/Admin/Kategori/update/3">Update</a>

</li>

    </ul>

</nav>

Artinya, terdapat nav yang memmpunyai <ul> dan terdapat 3 <li>, yaitu Kategori, yang di dalamnya terdapat hyperlink untuk memanggil base\_url() yang memanggil localhost/restoran-ci4, lalu mengarahkan ke dalam /Admin/Kategori/insert. <li> kedua adalah Insert yang di dalamnya terdapat hyperlink untuk memanggil base\_url() yang di dalamnya memanggil localhost/restoran-ci4, lalu mengarahkan ke dalam /Admin/Kategori/insert. Dan terdapat <li> ketiga yaitu Update, di dalam update terdapat base\_url() yang memanggil localhost/restoran-ci4, lalu mengarahkan ke dalam /Admin/Kategori/update

1. Mengedit baseURL yang ada di dalam file .env.

* Cth :

#app.baseURL = ' ' => sebelum diedit

app.baseURL = 'http://localhost/restoran-ci4' => sesudah diedit

Artinya, mengedit jalannya URL, agar ketika memanggil base\_url maka otomatis akan memanggil format localhost/restoran-ci4, dan tag atau hyperlink yang diberi tambahan base\_url() akan menambahkan localhost/restoran-ci4 pada awalnya.

1. Cara lain penulisan dari <?php echo base\_url() ?> sama dengan <?= base\_url()?>, tulisan php echo diganti menjadi = untuk mempersingkat.
2. Mengisi function formInsert pada file Kategori.php di dalam Admin.

* Cth :

public function formInsert()

{

    echo view("template/header");

    echo view("kategori/forminsert");

    echo view("template/footer");

}

Artinya, di dalam function formInsert memanggil beberapa view, karena memanggil file yang ada di dalam views, maka memanggilnya menggunakan view(), view memanggil file header yang ada pada template, file header adalah tempat untuk mengedit header dari web, seperti title, dan di dalam header.php terdapat nav. Setelah itu diarahkan pada view yang memanggil forminsert pada kategori, yang berisi form insert. Lalu pada view yang memanggil footer di dalam template, file footer berisi tag penutup dari /body dan /html yang merupakan sambungan dari file header.php.

1. Mengisi function formUpdate pada filekategori.php di dalam Admin.

* Cth :

public function formUpdate($id = null)

    {

        echo view("template/header");

        echo view("kategori/update");

        echo view("template/footer");

    }

Artinya, di dalam function formUpdate memanggil beberapa view, karena memanggil beberapa file yang ada di dalam views, maka memanggilnya menggunakan view(), view memanggil file header yang ada pada template, file header berisi html yang mengedit title pada web, dan terdapat nav di dalam header. Lalu diarahkan ke dalam view yang memanggil update yang ada pada kategori, berisi tulisan update data. Dan akhirnya terdapat footer yang berada di dalam template dan di dalam file footer terdapat tag penutup /body dan /html yang merupakan sambungan dari file header.php.

1. Semua pengiriman data di dalam codeigniter harus di dalam array assosiatif.
2. Menambahkan array assosiatif ke dalam function select() dan melakukan pemanggilan menggunakan print\_r.

* Cth :

$data = [

    'judul' => 'SELECT DATA',

    'kategori' => ['Makanan' , 'Minuman' , 'Jajan' , 'Gorengan' , 'Buah']

];

echo "<pre>";

print\_r($data);

echo "</pre>";

Artinya, di dalam $data, terdapat array assosiatif yang berisi, key 'judul' dengan value 'SELECT DATA', dan key 'kategori' dengan value array yang hasilnya adalah Makanan, Minuman, Jajan, Gorengan, dan Buah. Lalu code akan ditampilkan menggunakan print\_r, artinya code yang ditampilkan akan sama seperti code yang ditulis, dan untuk merapikan code nya menggunakan echo <pre> agar tampilan mudah dibaca.

1. Menampilkan array assosiatif / memasukkan controller ke dalam view select.php.

* Cth : pada file kategori.php :

echo view("kategori/select", $data);

Artinya, view memanggil file select pada kategori dan juga memanggil $data, $data adalah array assosiatif yang sudah dibuat yang berisi key judul dan key kategori.

pada file select.php :

h3> <?php echo $judul;?></h3>

Artinya di dalam php, memanggil $judul, untuk memanggil key dari array assosiatif, maka memanggilnya menggunakan $ dan setelah itu diberi nama key nya. Jadi di dalam $data terdapat key judul, dan key judul dipanggil ke dalam select.php, maka dari itu, untuk memanggil key judul adalah dengan menulis $judul.

1. Menampilkan array assosiatif jika di dalam value nya berupa array pada view select.php.

* Cth : pada file kategori.php :

echo view("kategori/select", $data);

Artinya, view memanggil file select pada kategori dan juga memanggil $data, $data adalah array assosiatif yang sudah dibuat yang berisi key judul dan key kategori. Pada value dari key kategori, valuenya adalah berupa array, jadi terdapat beberapa data yang menjadi value pada kategori.

pada file select.php :

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

    <h4><?= $key.' => '.$value ?></h4>

<?php *endforeach*; ?>

Artinya, di dalam php terdapat foreach yang menjalankan $kategori yang berasalh dari key kategori, key kategori dirubah menjadi $kategori karena memanggil array assosiatif dari $data. Di daam foreach, $kategori menjadi $key dengan array yang berada di dalam $kategori menjadi $value. Lalu, memanggil $key dan memanggil $value, karena yang dipanggil adalah $key dan $value, maka hasilnya akan menampilkan nilai index dan value nya. Lalu foreach diakhiri menggunakan endforeach.

1. Menampilkan satu data pada array di dalam $kategori.

* Cth :

<h4><?php echo $kategori[2]; ?></h4>

Artinya, di dalam php memanggil hasil dari array pada $kategori dengan index 2.

1. Dengan menggunakan View layout tidak perlu membuat file header dan footer, karena isi dari header dan footer sudah diambil dari dalam admin.php yang isinya terdapat renderSection.

* Cth : pada admin.php :

<?= $this->renderSection('content') ?>

pada kategori.php :

echo view("kategori/select", $data);

pada select.php :

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

<?= $this->section('content') ?>

    <h3> <?php echo $judul;?></h3>

    <h4><?php echo $kategori[2]; ?></h4>

<?= $this->endSection() ?>

Artinya, di dalam admin.php terdapat function renderSection, lalu pada kategori.php menjalankan view yang memanggil file select pada kategori dan menjalankan $data yang isinya berupa array assosiatif. Dan pada select.php terdapat extend yang mencari file admin yang berada di dalam template, lalu terdapat section yang bernama content, dan di dalam section content memanggil key judul pada $data , yaitu $judul dan memanggil nilai satuan pada array dalam key kategori pada $data, lalu section diakhiri dengan endSection.

Jadi urutannya adalah, di dalam admin.php terdapat title web View Layout dan terdapat readerSection() lalu di dalam file kategori.php menjalankan view yang memanggil file select pada folder kategori dan menjalankan $data, dan di dalam file select.php memanggil template/admin yang di dalamnya terdapat section content serta memanggil key kategori dan judul yang berada di dalam $data.

1. Model berhubungan dengan database.
2. Untuk menyambungkan codeigniter ke dalam databse, maka di dalam file .env bagian DATABASE, dinyalakan / dihilangkan tanda #, pada hostname, database, username, password, dan DBDriver.
3. Membuat file Kategori\_M.php untuk tempat menyimapn model.

* Cth :

<?php namespace App\Models;

    use CodeIgniter\Model;

        class Kategori\_M extends Model

        {

            protected $table = 'tblkategori';

        }

?>

Artinya, di dalam php terdapat namespace App\Models dengan use Codeigniter\Model yang mengambil Model dari extends Model, lalu nama class sama dengan nama file, yaitu Kategori\_M, di dalam class terdapat protected $table, dengan memanggil tblkategori.

1. Fungsi dari use dalam codeigniter adalah seperti require\_once pada php, yaitu berfungsi untuk memanggil semua isi dari file dan memasukkan nya ke dalam file lain yang diberi use.
2. Memasukkan dan menampilkan model dari Kategori\_M.php di dalam function select pada file kategori.php di dalam Admin.

* Cth :

public function select()

    {

        $model = new Kategori\_M();

        $kategori = $model -> findAll();

           }

Artinya, di dalam function select terdapat $model yang berisi objek yang memanggil class Kategori\_M, dan di dalam $kategori akan menjalankan $model yang menggunakan perintah findAll yaitu perintah dari codeigniter untuk menampilkan seluruh data yang diambil dari tblkategori yang sudah ditulis pada Kategori\_M.

1. findAll adalah perintah dari codeigniter yang digunakan untuk menampilkan seluruh data yang ada pada sebuah tabel yang dipilih.
2. Menampilkan nilai dari kategori pada tblkategori di dalam view.

* Cth : pada function select :

$data = [

    'judul' => 'SELECT DATA dari controller',

    'kategori' => $kategori

];

Artinya di dalam $data terdapat key judul dengan value SELECT DATA dari controller dan key kategori dengan value yang mengambil nilai pada $kategori.

pada select.php :

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

    <h4><?= $key.' => '.$value['kategori'] ?></h4>

<?php *endforeach*; ?>

<h4><?php echo $kategori[2]['kategori']; ?></h4>

Artinya, untuk menampilkan semua nilai pada array, di dalam foreach terdapat $kategori sebagai $key dan $value, lalu $key ditampilkan dan $value memanggil dari nilai pada kategori di dalam tblkategori. Untuk menampilkan satu nilai dari kategori, maka dapat menggunakan $kategori, memanggil index dan nama dari kolom yang ingin ditampilkan.

**Saya Belum Mengerti**