|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| PHP FRAMEWORK CI minggu ke 17 (video 01-09 Tutorial Codeigniter-4) | 95 |

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

1. Instalasi Codeigniter dari codeigniter.com dengan menekan download lalu memindahkan file framework ini ke dalam htdocs dan memasukkannya ke dalam folder restoran-ci4 baru di ekstrak agar muncul folder dari file WinRAR Zip. Untuk menggunakan codeigniter versi 4.0.4 Php yang harus digunakan minimal versi 7.2 jadi apabila versi php dibawah 7.2 harus melakukan instalasi ulang php atau update php.

2. Di dalam folder framework terdapat folder app yang berisi folder Controllers, folder ini memiliki file home.php yang digunakan untuk controller dari navbar atau menu yang ada di aplikasi restoran seperti menu kategori, untuk itu dibuatkan file controller yang isinya seperti file home.php untuk setiap menunya. Di dalam file ini terdapat class yang namanya harus sesuai dengan nama filenya, misalkan file Kategori.php untuk menu kategori maka nama class nya Kategori.

3. Untuk pembuatan controller disetiap menunya, pertama adalah pembuatan controller untuk ‘Kategori’, jadi dibuatkan file Kategori.php yang isinya sama seperti Home.php, di dalam filenya nama class untuk menu Kategori adalah class Kategori, di dalam class terdapat function-function yang memiliki beberapa fungsi, karena di function ini berada di dalam class, maka harus diberi tanda public atau private untuk functionnya. Untuk class Kategori functionnya ada 7 yaitu:

1. public function index(): berfungsi untuk menampilkan halaman utama dari class Kategori
2. public function select(): berfungsi untuk menampilkan semua data
3. public function selectWhere($id=null): berfungsi untuk menampilkan data yang dipilih
4. public function formInsert(): berfungsi untuk menampilkan form insert
5. public function formUpdate(): berfungsi untuk menampilkan form update
6. public function update($id=null): berfungsi untuk proses update data $id atau sesuai dengan id yang dipilih user
7. public function delete($id=null): berfungsi untuk proses delete data

Untuk pemanggilan function melalui URL browsernya, cukup menggunakan nama folder di dalam htdocs yang digunakan untuk codeigniter misalkan restoran-ci4/kategori/select maka dia akan membuka isi dari function select yang ada di dalam class kategori, jadi ibaratnya class ini sebagai folder dan functionnya sebagai file yang bisa langsung dipanggil secara singkat melalui url. Selanjutnya adalah membuat file Menu.php untuk menu ‘Menu’, untuk menu ini maka class nya diberi nama Menu juga. Di dalam class Menu ada 3 function yang digunakan yaitu:

1. public function index(): berfungsi untuk menampilkan halaman utama dari class Menu
2. public function select(): berfungsi untuk menampilkan data
3. public function update($id=null,$nama=null): berfungsi untuk mengupdate data sesuai dengan id dan nama dari data yang dipilih oleh user.

4. Setelah mengisi class tadi dengan function-function, selanjutnya di dalam folder controllers dibuatkan 2 folder bernama admin dan front untuk mengelompokkan antara file-file yang digunakan di dalam backend(admin) dan frontend nya. untuk file kategori.php dan menu.php yang dibuat tadi diletakkan di dalam folder admin, untuk folder front diisi dengan file controller juga yang dicopas dari home.php lalu direname menjadi homepage.php dengan class nya diganti homepage juga. Karena kategori.php dipindah ke dalam folder admin, tentu saja pasti terjadi eror karena kesalahan pembacaan alamat tujuan yang baru, untuk itu di dalam file Kategori.php pada bagian namespace nya tambahkan \Admin agar dia masuk ke dalam folder Admin dulu baru mendapatkan file Kategori.php, selanjutnya karena setiap controller membutuhkan basecontroller dan letak basecontroller ini diluar admin atau tidak satu tempat dengan Kategori.php maka dibawah namespace tambahkan code use App\Controllers\BaseController; yang artinya dia akan menggunakan BaseController yang ada di dalam folder App->folder Controllers->file BaseController. Dengan begitu file controller yang dibuat seperti file Kategori.php yang telah dipindahkan ke folder Admin tidak akan terjadi eror dan bisa berjalan. Jadi misalkan mau mengambil function select di dalam file Kategori.php maka di url nya dari folder retoran-ci4/Admin/kategori/select karena dia sudah berada di folder Admin.

5. Apabila terjadi eror dan tidak diberitahu eror nya dibagian apa, untuk itu di dalam file env folder app, rename menjadi .env lalu di dalam file ini cari bagian #CI\_ENVIRONMENT = production, copas code itu tadi dibawahnya lalu ubah production menjadi development dan hilangkan tanda # untuk mengaktifkannya, dengan begitu ketika terjadi eror dia akan memberi tau dibagian mananya yang membuat eror.

6. Routes adalah sebuah permintaan atau request di dalam url yang bisa diatur atau dicustom sesuai keinginan user. Jadi misalkan di dalam url dituliskan alamat yang harusnya menampilkan file home.php tetapi yang kita inginkan untuk muncul adalah file login.php dengan url home.php atau nama url lainnya maka menggunakan routes. Untuk sebuah percobaan, di dalam folder controllers buat file baru yang bernama Login.php dengan class nya juga Login, di dalam class pada bagian function index, isinya yang bagian return view untuk menampilkan halaman codeigniter di comment saja lalu bawahnya buat echo h1 LOGIN, jadi tampilan file ini nanti hanya sebuah tulisan LOGIN, selanjutnya buka file Routes.php yang sudah disediakan dari codeigniter di dalam folder config, pada bagian $routes->setDefaultController(‘Home’); ubah Home nya menjadi Login agar dia mengatur controller file Login.php, lalu pada bagian $routes->get(‘/’,’Home::index’); yang berarti ketika pada url setelah tanda / setelah restoran-ci4 apabila kosong maka dia akan menampilkan class Home dengan mengambil function index. Jadi apabila ingin mengubah ketika url nya hanya localhost/restoran-ci4/ dan dia tampil function index dari class Login maka ubah alamatnya di ‘’ kedua di dalam $routes->get. Dan apabila ketika mengetikkan url localhost/restoran-ci4/Kategori agar yang tampil adalah index dari class login maka pada ‘/’ di $routes->get ubah menjadi ‘kategori’ yang artinya apabila setelah restoran-ci4/ adalah kategori maka dia akan menampilkan ‘’ setelah koma yaitu ‘Login::index’. Apabila di dalam $routes->get menampilkan functionnya sekaligus menampilkan $id di dalam sebuah class maka untuk url nya atau ‘’ yang pertama ketikann nama class nya dulu (mengetik class apa? Misal kategori) lalu tambahkan /(:any) yang berarti apapun yang diketikkan setelah kategori/ dia akan tampil ke halaman browsernya, agar dia tampil ke browsernya maka setelah nama functionnya diberi tanda / bukan \ lalu $1 yang berarti dia menggunakan parameter /$1. Setelah itu yang ditambahkan dan tampil di browser ada di sebelah mana? Dia akan berada di sebelah isi dari function yang ditampilkan, jadi misalkan alamat nya atau di dalam ‘’ yang kedua adalah Admin\Kategori::selectWhere/$1 berarti dia akan masuk ke dalam folder Admin (karena letak controller Login dengan Kategori harus masuk ke admin dulu) lalu mengambil class Kategori dan memilih function selectWhere dan setelah dia menampilkan isi dari function selectWhere dia akan menampilkan parameter atau $1 atau yang ditambahkan di url tadi ke browsernya.

7. Routes Group digunakan untuk menyimpan function yang berisi banyak $routes agar bisa mengatur lebih dari satu $routes, jadi dia seperti mengaktifkan banyak $routes yang ada di dalam functionnya itu, missal:

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

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

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

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

});

Berarti ada 1 Routes Group yang apabila setelah restoran-ci4/ dituliskan admin seperti restoran-ci4/admin maka dia akan menuju alamat sesuai dengan function yang ada, misalkan diketikkan code

* localhost/restoran-ci4/kategori/form dia akan menampilkan isi dari function formInsert yang ada di dalam Class Kategori di dalam folder Admin
* localhost/restoran-ci4/kategori dia akan menampilkan isi dari function select yang ada di dalam Class Kategori di dalam folder Admin
* localhost/restoran-ci4/kategori/update/sembarang maka dia akan menampilkan isi dari function formUpdate yang ada di dalam class Kategori di dalam folder Admin lalu menampilkan $1 nya yaitu sembarang atau (:any) itu (sesuai yang diketikkan setelah update/)
* Dan apabila hanya localhost/restoran-ci4/admin saja maka dia akan eror karena di dalam function tidak tersedia untuk ‘/’ saja atau yang berarti setelah admin/ kosong.

8. Penggunaan View CodeIgniter, view ini berfungsi untuk tempat file-file yang nanti bisa dibuka dari controllersnya. Jadi view ini berupa folder yang letaknya ada di dalam folder app. Isi dari view yaitu folder error dan file welcome\_message.php. untuk menggunakan view maka di dalam folder view ini dibuatkan folder baru misalkan folder kategori dan folder template. Di dalam folder kategori diisi dengan file sesuai nama function yang ada di class kategori pada controllers misalkan select.php, forminsert.php, dan update.php. lalu di dalam folder template diisi dengan file header.php dan footer.php. di dalam file header.php diisi dengan html sampai bagian open bodynya, lalu untuk close body dan close html diletakkan di footer.php. setelah itu di dalam controller bagian Kategori.php pada bagian function select isinya diganti dengan code

echo view("template/header");

echo view("kategori/select");

echo view("template/footer");

yang berarti nanti ketika di url diketikkan sesuai yang sudah diatur pada routes.php yaitu apabila mengetikkan localhost/restoran-ci4/admin/kategori maka akan membuka function select dan yang tampil adalah pada bagian pertama file header pada folder template yang ada di view, pada bagian kedua file select pada folder kategori yang ada di view, dan bagian terakhir akan tampil file footer pada folder header. Jadi echo view menampilkan sebuah file di dalam broser tetapi view ini khusus menampilkan file yang ada di folder view.

9. Di dalam folder app ada file .env, di dalam file ini ada bagian app.baseURL ini dihilangkan tanda # yang berarti untuk mengaktifkan lalu diisi dengan <http://localhost/restoran-ci4> menjadi app.baseURL = 'http://localhost/restoran-ci4' yang nantinya ketika ada hyperlink atau href dan diberi php echo base\_url() artinya link yang dituju pertama adalah localhost/restoran-ci4/ lalu diluar php lanjutan dari link yang ditujunya. Jadi link yang dituju tidak akan terjadi error karena sudah

10. Jika ada php dan didalamnya hanya berisi 1 echo saja untuk sebuah $variable misalkan, maka php itu bisa diganti dengan <?= $misal ?> jadi yang sebelumnya <?php echo $misal?> kata php dengan echo bisa disingkat dengan =.

11. Mengirim data dari controller ke view. Jadi misalkan di dalam class Kategori di controller ada sebuah variable yang berisi sebuah data di dalam function select seperti ini

$data = [

            'judul' => 'SELECT DATA DARI CONTROLLER',

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

        ];

Lalu data ini nanti akan dikirimkan ke dalam file yang ada di view yaitu file select.php di folder kategori maka di dalam function select echo view untuk select nya ditambah , $data seperti ini echo view("kategori/select", $data); lalu di dalam file select.php yang ada di folder kategori di dalam view nya apabila ingin menampilkan data dari controller tadi caranya seperti ini <h1><?= $judul; ?></h1> atau <?php echo $judul?> dan judul ini adalah key dari $data array assosiatif. Dengan mengecho key judul dari $data menjadi $judul yang tampil adalah value dari key nya yaitu ‘SELECT DATA DARI CONTROLLER’. Jadi bisa disimpulkan apabila mau menampilkan value dari key yang ada di $data tadi cukup meng echo key nya dengan mengubah nama key menjadi variable, tidak menyertakan $data karena sudah dicantumkan di echo view function selectnya. Dan apabila ingin menampilkan salah satu data dari array indeks milik key kategori caranya sama seperti sebelumnya yaitu mengecho key nya dengan mengubahnya menjadi variable, tetapi karena value dari key ini adalah sebuah array indeks maka ditambahkan [] lalu nomor indeks yang dipilih untuk ditampilkan seperti ini <h1><?= $kategori[1]  ?></h1>. Sedangkan jika ingin menampilkan semua data dari array indeks dari key kategori maka bisa menggunakan foreach atau perulangan.

12. Dengan menggunakan View Layout kita dapat membuat tampilan atau aplikasi lebih sederhana karena tinggal menyiapkan satu file tambahan di dalam folder template yaitu bernama file admin.php dan file ini diisi dengan html yang title nya View Layout lalu di dalam body bisa dicopas nav ul li yang sebelumnya sudah dibuat di dalam file header.php. kemudian ditambahkan dibawah nav code seperti ini <?= $this->renderSection('content') ?> yang berarti dia akan menyimpan file ini dengan section bernama content. Lalu di bagian file select.php ditambahkan <?= $this->extend('template/admin')  ?> untuk mencari letak file nya dulu lalu ditambahkan <?= $this->section('content') ?> sebelum code yang ada di select.php untuk menggunakan section nya dan setelah code dari select.php ditambahkan <?= $this->endSection() ?> untuk mengakhiri section. Lalu di dalam file forminsert.php dan update.php juga ditambahkan $this->extend, $this->section, $this->endSection. Lalu ke bagian controller class Kategori di function select, formInsert, dan formUpdate dihapus saja echo view header dan footer nya. Ketika dicek di browser nya nav ul li masih tetap berfungsi padahal echo view header dan footer nya sudah dihapus itu karena dari file admin.php atau tempat dibuatnya section content, sectionnya digunakan di file select.php, forminsert.php, dan update.php dengan menggunakan $this->section(‘content’) dan memanggil letak file adminnya dulu. Jadi ketika di dalam function select, formInsert, dan formUpdate langsung memanggil folder kategori, nav ul li nya juga otomatis tampil. Intinya admin.php membuat section content lalu di dalam file select.php, forminsert.php dan update.php memakai section contentnya dan 3 file itu tadi sudah disambungkan di dalam class Kategori sesuai dengan function masing-masing.

13. Terdapat sebuah folder bernama Models di dalam folder App, folder Models ini yang nantinya berhubungan dengan database. Untuk itu sebelum menggunakan database di dalam file .env pada bagian database default aktifkan semua dengan menghilangkan tanda # nya. Lalu mengganti isi dari database.default sesuai dengan database yang akan digunakan. Setelah itu kembali ke folder Models buat sebuah file baru bernama Kategori\_M.php lalu isi file ini dengan php yang didalamnya:

namespace App\Models;

    use CodeIgniter\Model;

    class Kategori\_M extends Model

    {

        protected $table = 'tblkategori';

    }

Jadi dia akan disambungkan dengan tblkategori dari database dbrestoran. Untuk bisa tersambung dengan file Kategori.php maka ditambahkan use App\Models\Kategori\_M; agar dia menampilkan file sesuai alamat tujuannya yaitu Kategori\_M.php karena di dalam file Kategori\_M.php ini terdapat class Kategori\_M yang isinya menyambungkan dengan tblkategori maka kembali lagi ke file Kategori.php ke function select nya ditambahkan variable $model = new Kategori\_Model() agar selain menyambungkan dengan file juga masuk ke dalam class yang bernama Kategori\_M itu dengan begitu function select ini bisa tersambung ke tblkategori juga. Lalu di bawahnya ditambahkan $kategori = $model->findALL(); yang berarti $kategori ini adalah salah satu kolom yang ada di tblkategori, dengan adanya findALL() ini berfungsi untuk mencari semua data atau semua baris yang ada di kolom kategori nya, jadi data dari kolom kategori ini bisa ditampilkan di browser itu. Karena sebelumnya tidak menggunakan model ini dan menuliskan secara manual, agar tidak terjadi eror kembali lagi di dalam select.php pada bagian <h2><?= $value ?></h2> ditambahkan ['kategori'] menjadi <h2><?= $value[‘kategori’] ?></h2> karena $kategori yang di function select bagian $data nya key ‘kategori’ diganti value nya menjadi 'kategori' => $kategori maka ditambahkan [‘kategori’] sesuai nama key nya.

**Saya Belum Mengerti**

1.