|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran CodeIgniter 4 Video 20 – 34  (Tugas minggu ke - 19) | 95 |

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

1. Menambahkan beberapa menu item, yaitu menu Pelanggan, Order, dan Order-detail ke dalam admin.php.

Cth :

<li class="list-group-item">

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

</li>

<li class="list-group-item">

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

</li>

<li class="list-group-item">

<a href="<?= base\_url('/Admin/Orderdetail') ?>">Order-detail</a>

</li>

Artinya, terdapat 3 <li> dengan menu item di dalamnya, pada <li> pertama, terdapat menu item Pelanggan, di dalam Pelanggan terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4, lalu disambungkan dengan /Admin/Pelanggan, jadi di dalam menu pelanggan, apabila di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Kategori. Pada <li> kedua, terdapat menu item Order, dan di dalam Order terdapat hyperlink yang memanggil base\_url() dengan alamat, localhost/restoran-ci4, lalu disambungkan dengan /Admin/Order, jadi jika menu Order di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Order. Dan di dalam <li> ketiga, terdapat hyperlink yang memanggil base\_url() dengan alamat base\_url() dan disambungkan dengan /Admin/Orderdetail, jadi di dalam menu Order-detail, jika di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Orderdetail.

1. Mengatur kerapian dari button tambah data dan tulisan data kategori agar sejajar dengan memasukkan ke dalam row.

Cth :

<div class="row">

    <div class="col-4">

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

    </div>

    <div class="col-4">

        <h4 class="mt-2"> <?php echo $judul;?></h4>

    </div>

</div>

Artinya, terdapat <div> row dengan 2 <col-4>, yang artinya pada setiap col, terdapat 4 kolom di dalamnya, <col-4> pertama berisi <a> yang di dalamnya terdapat button TAMBAH DATA dengan warna button primary, lalu terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4, lalu disambungkan ke dalam /Admin/Kategori/create, jadi di dalam button TAMBAH DATA apabila button di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Kategori/create, atau menjalankan create yang ada di dalam Kategori pada Admin. Pada <col-4> kedua berisi, h4 dengan margin-top / batas atas 2, dan di dalamnya memanggil $judul yang berisi tulisan DATA KATEGORI. Karena masing – masing sudah diatur ke dalam col, sehingga antara button TAMBAH DATA dan tulisan DATA KATEGORI akan sejajar.

1. Mendownload octions di dalam bootstrap, oction digunakan sebagai icon. Cara melakukan download adalah, pertama dengan membuka web getbootstrap.com lalu klik documentation, dan search icon, di dalam icon terdapat pilihan octions, lalu klik oction dan di dalam oction tersedia berbagai icon, lalu pilih icon yang ingin diambil setelah itu download svg, dan icon sudah didownload, tinggal dimasukkan, dengan cara memanggil filenya.
2. Memasukkan dan memanggil file oction dengan icon hapus dan pensil.

Cth :

 <th>Aksi</th>

<td>

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

<img src="<?= base\_url('/icon/can.svg')?>"></a>

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

<img src="<?= base\_url('/icon/pen.svg') ?>"></a>

</td>

Artinya, terdapat <th> yang berisi Aksi, dan pada <td> yang di dalamnya terdapat 2 tag<a> dengan tag <a> pertama berisi icon hapus, base\_url() yang dengan alamat localhost/restoran-ci4 yang disambungkan dengan /Admin/Kategori/delete, jadi ketika link di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Kategori/delete/$value[‘idkategori’] artinya value yang diambil adalah dari idkategori yang dipilih. Lalu memanggil file img dengan src adalah base\_url(‘/icon/can.svg’), artinya di dalam hyperlink tersebut menjalankan localhost/restoran-ci4/icon/can.svg, yang di dalam can.svg adalah berisi icon hapus, jadi hyperlink tersebut akan menampilkan icon hapus yang ketika di klik iconnya, maka akan menjalankan perintah pada localhost/restoran-ci4/Admin/Kategori/delete. Tag <a> yang kedua berisi icon pensil digunakan untuk update, di dalam tag <a> memanggil base\_url() dengan alamat localhost/restoran-ci4 yang disambungkan dengan /Admin/Kategori/find, jadi ketika link di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Kategori/find/$value[‘idkategori’], artinya value yang diambil adalah dari idkategori yang dipilih. Lalu memanggil file img dengan srcnya adalah base\_url(‘/icon/pen.svg’), artinya, di dalam hyperlink tersebut menjalankan localhost/restoran-ci4/icon/pen.svg, yang di dalam pen.svg adalah berisi icon pensil, jadi hyperlink tersebut akan menampilkan icon pensil yang apabila jika iconnya di klik, maka akan menjalankan perintah yaitu pada localhost/restoran-ci4/Admin/Kategori/find.

1. Melakukan edit pada form.

Cth :

<div class="col-8">

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

        <div class="form-group">

            <label for="Kategori">Kategori</label>

            <input type="text" class="form-control" name="kategori" required>

        </div>

        <div class="form-group">

            <label for="Keterangan">Keterangan</label>

            <input type="text" class="form-control" name="keterangan" required>

        </div>

        <div class="form-group">

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

        </div>

    </form>

</div>

Artinya terdapat div dengan col-8, jadi isi di dalam div hanya seukuran kolom berisi 8, di dalam div terdapat form yang menjalankan action base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Kategori/insert, jadi ketika form dikirimkan dengan mengklik submit, maka tindakan form akan dikirimkan kepada function insert yang ada di dalam Kategori. Pada form terdapat form-group pada setiap inputan nilai, di dalam form-group dengan input Kategori, terdapat label Kategori dan input bertipe text dengan class form-control dan name kategori, form Kategori ini wajib diisi karena bersifat required. Pada form dengan input Keterangan, terdapat label Katerangan dan input yang bertipe text dengan class form-control dan name keterangan, form Keterangan juga wajib diisi karena formnya bersifat required. Dan pada form terakhir, yaitu form dengan input bertipe submit, dengan name simpan dan value SIMPAN. Maka hasil dari form adalah berupa baris panjang dengan ukuran col-8.

1. Menampilkan error yang terjadi jika data yang di update tidak memenuhi syarat.

Di dalam /Admin/Kategori/find :

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, di dalam function find terdapat objek $model yang menjalankan Kategori\_M, dimana di dalam Kategori\_M terdapat $table tblkategori, artinya table yang dihubungkan oleh model ke dalam database adalah tblkategori dan $primarykey [‘idkategori’] artinya primary key dari tblkategori adalah idkategori. $kategori berisi $model yang menjalankan peritah mengambil $id yaitu dengan find($id), lalu untuk memasukkan data ke dalam view maka harus menggunakan variable yang berisi key dan value berupa data yang akan dimasukkan ke dalam view, di dalam $data dengan key judul dan value UPDATE DATA dan key kategori dengan value $kategori. Apabila selesai mengambil $id, maka halaman akan diarahkan kedalam update/kategori pada view dan menjalankan $data.

Di dalam view kategori/update :

<?php

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

?>

Di dalam kategori/update berisi form dengan action /Admin/Kategori/update, jadi apabila submit SIMPAN di klik, maka tindakan pada form akan dikirimkan kepada /Admin/Kategori/update. Apabila data yang dimasukkan tidak sesuai denagn peraturan, maka function update pada /Admin/Kategori/update akan mengambil nilai error, dan karena di dalam /Admin/Kategori/find menjalankan view kategori/update, maka ketika error, pesan error akan ditampilkan di dalam view kategori/update. Dan untuk menampilkan pesan error, maka perintah session dibaca menggunakan echo, di dalam session() menjalankan perintah untuk mendapatkan data / getFlashdata dari key info.

Di dalam /Admin/Kategori/update :

public function update()

    {

        $model = new Kategori\_M();

        $id = $\_POST['idkategori'];

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

            $error = $model->errors();

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

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

        } *else* {

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

        }

    }

Artinya, di dalam function update terdapat $model yang menjalankan Kategori\_M, dimana di dalam Kategori\_M terdapat $table tblkategori, artinya table yang dihubungkan oleh model ke dalam database adalah tblkategori dan $primarykey [‘idkategori’] artinya primary key dari tblkategori adalah idkategori. Lalu terdapat $id yang mengambil nilai dari $\_POST[‘idkategori’]. Dan terdapat pengujian apabila $model yang menjalankan perintah save, karena di dalam objek $model yang menjalankan Kategori\_M terdapat primary key, jadi save disini digunakan untuk update data. Pengujian yang dilakukan adalah apabila $model yang menjalankan perintah update data dengan mengambil nilai dari $\_POST adalah salah, maka di dalam $error terdapat $model yang menjalankan perintah error. Lalu di dalam session() terdapat perintah untuk mengambil setFlashdata yang digunakan untuk mengatur pesan error dengan mengambil nilai dari key info lalu menjalankan $error[‘kategori’] yang sudah diatur dalam key ‘kategori’. Setelah mengambil nilai error, maka halaman akan dikembalikan ke dalam base\_url dengan alamat localhost/restoran-ci4 dan disambungkan ke dalam /Admin/Kategori/find/$id. Jadi ketika selesai mengambil nilai error, maka halaman akan dikembalikan ke dalam localhost/restoran-ci4/Admin/Kategori/find/$id. Dan jika data yang dimasukkan sesuai dengan aturan / sesuai dengan peraturan, maka akan menjalankan else dengan mengambalikan halaman ke dalam localhost/restoran-ci4/Admin/Kategori.

Jadi, urutannya, apabila link update di klik maka akan diarahkan ke dalam /Admin/Kategori/find, lalu di dalam /Admin/Kategori/find menjalankan view kategori/update, apabila form pada view kategori/update dikirim maka akan diarahkan action form ke dalam /Admin/Kategori/update, dan di dalam update terdapat pengujian jika $\_POST yang dikirimkan benar maka halaman akan dikembalikan pada halaman /Admin/Kategori, tetapi jika $\_POST yahg dikirimkan salah, maka akan mengambil nilai error lalu diarahkan kembali ke dalam /Admin/Kategori/find/$id, dan di dalam /Admin/Kategori/find/$id akan menjalankan view kategori/update yang menampilkan pesan error dengan getFlashdata.

1. Melakukan pengaturan pada penomoran baris.

Cth :

<?php

*if* (isset($\_GET['page\_page'])) {

        $page = $\_GET['page\_page'];

        $jumlah = 3;

        $no = ($jumlah \* $page) - $jumlah + 1;

    }*else*{

        $no = 1;

    }

?>

Artinya, di dalam view kategori/select, terdapat pengujian menggunakan if yang berisi, apabila $\_GET[‘page\_page’] di klik, maka akan menjalankan $page yang mengambil nilai dari $\_GET[‘page\_page’], di dalam $jumlah nilai nya sama dengan data yang ditampilkan, yaitu 3, dan pada $no terdapat operasi aritmatika, yaitu $jumlah dikalikan dengan $page lalu dikurangi $jumlah dan ditambah 1. Dan jika apabila $\_GET[‘page\_page’] tidak di klik, maka akan menjalankan else yaitu $no yang berisi 1.

Contoh : apabila page\_page yang di klik adalah 2, maka 3 \* 2 – 3 + 1 = 4, jadi data yang ditampilkan akan dimulai dari nomor 4.

1. Menambahkan warna pada pesan error di dalam insert.php.

Cth :

<div class="col mt-2">

    <?php

*if* (!empty(session() -> getFlashdata('info'))) {

            echo '<div class="alert alert-danger" role="alert">';

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

            echo '</div>';

        }

    ?>

</div>

Artinya, terdapat div dengan class col dan margin-top 2, di dalam div terdapat pengujian menggunakan if, yaitu apabila isi dari session() -> getFlashdata yang mangambil nilai dari key info tidak kosong / !empty, maka akan menjalankan div yang berisi pengaturan pada pesan untuk pemberitahuan error. Lalu, session() -> getFlashdata(‘info’) akan ditampilkan menggunakan echo. Maka apabila terdapat pesan error, session() -> getFlashdata akan ditampilkan dengan tulisan dan background berwarna merah

1. Membuat file menu.php serta menambahkan namespace dan use di dalamnya.

Cth :

<?php namespace App\Controllers\Admin;

use App\Controllers\BaseController;

class Menu extends BaseController

{

    public function index()

    {

*return* view('welcome\_message');

    }

*//--------------------------------------------------------------------*

}

Artinya, pada php terdapat namespace dengan direktori file App\Controller\Admin yang artinya file tersebut diambil dari App lalu ke Controllers dan masuk ke dalam Admin. Use digunakan untuk memanggil semua isi pada file seperti require\_once. Dan terdapat use yang mengambil nilai dari App\Controllers\BaseController, memanggil BaseController karena letak dari BaseController adalah di luar dari folder Admin, jadi untuk memanggil nya digunakan use App\Controllers\BaseController agar BaseController dapat digunakan. Karena nama file adalah Menu, maka nama yang digunakan dalam class adalah nama yang sama seperti nama file, yaitu Menu, di dalam class terdapat function index, jadi ketika file dipanggil /Admin/Menu, maka akan langsung dijalankan function index yang berisi welcome message.

1. Membuat file Menu\_M di dalam Models.

Cth :

<?php namespace App\Models;

use CodeIgniter\Model;

class Menu\_M extends Model

{

    protected $table = 'tblmenu';

}

?>

Artinya, pada php terdapat namespace dengan direktori file App\Models, yang artinya file tersebut diambil dari App lalu masuk ke Models, di dalam Models terdapat use yang mengambil nilai dari CodeIgniter\Model. Karena nama file adalah Menu\_M, maka nama dari class adalah sama dengan nama dari file yaitu Menu\_M. Di dalam class Menu\_M terdapat variable $table dengan visibility protected, $table tblmenu, artinya, table yang dihubungkan oleh model ke dalam database adalah tblmenu.

1. number\_format() adalah fungsi bawaan dari php yang digunakan untuk mengatur tampilan angka, jadi dapat mengatur tampilan angka ribuan, dan lainnya.

Cth :

<td><?= number\_format($value['harga'])?></td>

Artinya, di dalam td terdapat number\_format, number\_format mengatur tampilan angka dari $value[‘harga’].

1. Melakukan pemanggilan gambar pada tabel.

Cth :

 <td>

<img style="width:100px;" src="<?= base\_url('upload/'.$value['gambar'].'')?>"

alt="">

</td>

Artinya, di dalam td terdapat img dengan pengaturan besar gambar adalah 100px dan mengambil gambar dari base\_url dengan alamat localhost/restoran-ci4 lalu disambung dengan upload dan mengambil value dari $value[‘gambar’].

Jadi, awal untuk pembuatan Menu adalah dengan membuat modelsnya dulu, yaitu Menu\_M di dalam Models, lalu membuat file controller Menu.php yang berisi $data yang akan dikirimkan ke dalam view, lalu memanggil view yang menjalankan menu/select dan menjalankan $data. Di dalam menu/select akan menjalankan beberapa perintah, yaitu paging, pengaturan judul, pemanggilan data dalam tabel pada tblmenu dan memanggil nya di dalam tabel menggunakna foreach karena isi dari tblmenu itu lebih dari satu, jadi dilakukan perulangan untuk mengambil nilai dari tblmenu.

1. Menambahkan form yang memanggil view\_cell dan mengarahkan view\_cell ke dalam function option.

Cth :

<div class="row">

    <div class="col-5">

        <form action="<?= base\_url('/Admin/Menu/read')?>" method="get">

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

        </form>

    </div>

</div>

Artinya, di dalam div row terdapat div class dengan col-5 dan di dalamnya terdapat form yang menggunakan action base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Menu/read, jadi tindakan dari form diarahkan ke dalam localhost/restoran-ci4/Admin/Menu/read, form ini mempunyai method get, pada form ini terdapat view\_cell yang memanggil function option pada \App\Controllers\Admin\Menu .

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 :

<div class="form-group">

<select class="form-control mt-2" onchange="this.form.submit()" name="idkategori" id="idkategori"

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

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

    <?php *endforeach*; ?>

</select>

Artinya, di dalam select terdapat class form-control dengan margin-top adalah 2 dengan name idkategori dan id idkategori, onchange this.form.submit digunakan untuk mengirimkan tindakan dari select tanpa tombol submit, jadi ketika option pada list di klik, maka akan mengirimkan tindakan yang akan memroses data, dan di dalam select 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 jangan lupa membuat function di dalam controller yang berisi pengaturan data yang akan dikirimkan kepada view yang dijalankan oleh view\_cell, dan 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 select.php, memanggil function option pada menu dengan menggunakan view\_cell, jadi function option itu adalah function yang berisi view\_cell yang menjalankan view pada file template/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. Onchange digunakan untuk melakukan suatu tindakan apabila nilai / value berubah. Pada form, onchange = “this.form.submit” digunakan untuk mengirimkan tindakan form tanpa menggunakan submit.
2. Mengambil dan menampilkan nilai $id dari $\_GET[‘idkategori’].

Cth :

public function read()

    {

*if* (isset($\_GET['idkategori'])) {

           $id = $\_GET['idkategori'];

           echo $id;

       }

    }

Artinya, di dalam function read terdapat pengujian menggunakan if yaitu, apabila $\_GET[‘idkategori’] di klik, maka $id akan mengambil nilai dari $\_GET[‘idkategori’], dan nilai dari $id akan ditampilkan menggunakan echo.

1. Perintah untuk mengambil data sesuai dengan kolom pada codeigniter.

Cth :

$users **=** $userModel**->**where('active', 1)

**->**findAll();

Artinya, $users adalah nama dari variable yang dibuat, $userModel adalah nama dari variable yang menyimpan model yang akan digunakan, where(‘active’, 1), where berfungsi untuk mencari dimana data yang dicari adalah kolom dan id nya, lalu menjalankan perintah untuk findAll atau perintah untuk mengambil data.

1. Mengambil dan menampilkan data yang mempunyai idkategori sesuai dengan $id yang dipilih pada tblmenu.

Cth :

public function read()

    {

*if* (isset($\_GET['idkategori'])) {

           $id = $\_GET['idkategori'];

           echo $id."<br>";

           $model = new Menu\_M();

           $menu = $model -> where('idkategori', $id) -> findAll();

           echo "<pre>";

           print\_r($menu);

           echo "</pre>";

       }

    }

Artinya, di dalam function read terdapat pengujian menggunakan if, yaitu apabila $\_GET[‘idkategori’] di klik, maka $id akan mengambil nilai dari $\_GET[‘idkategori’], lalu $id akan ditampilkan menggunakan echo. Dan terdapat objek $model yaitu menjalankan model Menu\_M, dimana di dalam Menu\_M ini berisi $table tblmenu, yang artinya tabel yang dihubungkan oleh Model ke dalam database adalah tblmenu, lalu $menu berisi $model yang menjalankan perintah untuk mencari data dengan kolom idkategori dan id nya sesuai dengan $id, lalu menjalankan perintah findAll untuk mengambil data. Dan $menu ditampilkan menggunakan print\_r untuk menampilkan hasil sesuai dengan apa yang diketik dan echo “<pre>” digunakan untuk merapikan tampilan code dengan format yang lebih mudha dibaca.

1. Menampilkan data yang diambil dan mengirimkan data ke dalam view.

Cth :

public function read()

    {

*if* (isset($\_GET['idkategori'])) {

           $id = $\_GET['idkategori'];

           $model = new Menu\_M();

           $menu = $model -> where('idkategori', $id) -> findAll();

           $data = [

            'judul' => 'DATA PENCARIAN MENU',

            'menu' => $menu,

            'pager' => $model -> pager

           ];

           echo view('menu/cari', $data);

       }

    }

Artinya, di dalam function read terdapat pengujian menggunakan if, yaitu apabila $\_GET[‘idkategori’] di klik, maka $id akan mengambil nilai dari $\_GET[‘idkategori’], lalu terdapat objek $model yang menjalankan model Menu\_M, dimana di dalam Menu\_M berisi $table tblmenu, yang artinya tabel yang dihubungkan oleh model ke dalam database adalah tblmenu. $menu berisi $model yang menjalankan perintah mencari data dari kolom idkategori dan id sesuai dengan $id, lalu menjalankan perintah findAll untuk mengambil data. $data berisi array assosiatif dimana $data inii akan dikirimkan ke dalam view. $data berisi keu judul DATA PENCARIAN MENU, key menu yang mengambil nilai dari $menu, d key pager yang berisi $model untuk menjalankan paginate. Lalu menjalankan view menu/cari dan juga menjalankan $data.

1. count() adalah function pada php yang digunakan untuk emnghitung data yang ditampilkan.

Cth :

$jumlah = $model -> where('idkategori', $id) -> findAll();

$count = count($jumlah);

Artinya, di dalam $jumlah berisi $model yang menjalankan perintah where untuk mencari data dari kolom idkategori dan id yang diambil dari $id, lalu menjalankan perintah findAll data, lalu $count berisi perintah count yaitu perintah yang digunakan untuk menghitung data yang diambil, dan count menghitung data dari $jumlah.

1. Mengatur isi dari function read dan mengirimkan data ke dalam view menu/cari.

Cth :

public function read()

    {

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

*if* (isset($\_GET['idkategori'])) {

           $id = $\_GET['idkategori'];

           $model = new Menu\_M();

           $jumlah = $model -> where('idkategori', $id) -> findAll();

           $count = count($jumlah);

           $tampil = 3;

           $mulai = 0;

*if* (isset($\_GET['page'])) {

               $page = $\_GET['page'];

               $mulai = ($tampil \* $page) - $tampil;

           }

           $menu = $model -> where('idkategori', $id) -> findAll($tampil, $mulai);

           $data = [

            'judul' => 'DATA PENCARIAN MENU',

            'menu' => $menu,

            'pager' => $pager,

            'tampil' => $tampil,

            'total' => $count

           ];

           echo view('menu/cari', $data);

       }

    }

Artinya, di dalam function read terdapat $pager digunakan untuk menyalakan pager pada paging, lalu terdapat pengujian yang dilakukan menggunakan if, yaitu apabila $\_GET[‘idkategori’] di klik, maka $id mengambil nilai dari $\_GET[‘idkategori’]. Lalu terdapat objek $model yang menjalankan Menu\_M dimana di dalam Menu\_M berisi $table tblmenu, yang artinya tabel yang dihubungkan oleh model ke dalam database adalah tblmenu. Dan di dalam $jumlah berisi $model yang menjalankan perintah where untuk mencari data dari kolom idkategori dan id yang diambil dari $id, lalu menjalankan perintah findAll data, lalu $count berisi perintah count yaitu perintah yang digunakan untuk menghitung data yang diambil, dan count menghitung data dari $jumlah. $tampil berisi 3 dan $mulai berisi 0. Lalu terdapat pengujian yang kedua, apabila $\_GET[‘page’] di klik, maka $page mengambil nilai dari $\_GET[‘page’], dan $mulai menjalankan operasi artimatika yaitu $tampil dikalikan $page dikurangi $tampil. Lalu di dalam $menu berisi $model yang menjalankan perintah mencari data dari kolom idkategori dan id sesuai dengan $id, lalu menjalankan perintah findAll untuk mengambil data, dan untuk mengatur berapa data yang ditampilkan, maka di dalam findAll() berisi $tampil dan $mulai. $data berisi data – data yang akan dikirimkan ke dalam view, data data nya berada di dalam array assosiatif. $data berisi key judul dengan value DATA PENCARIAN MENU, key menu berisi value yang mengambil nilai dari $menu, key pager mengambil nilai dari $pager, lalu key tampil mengambil nilai dari $tampil dan total mengambil nilai dari $count. Lalu view menu/cari dan $data akan dijalankan.

1. Mengatur tampilan urutan nomor dari data pada tabel dan menambahkan paging.

Cth :

Pengaturan urutan nomor pada tabel :

<?php

*if* (isset($\_GET['page'])) {

        $page = $\_GET['page'];

        $jumlah = 3;

        $no = ($jumlah \* $page) - $jumlah + 1;

    }*else*{

        $no = 1;

    }

?>

Artinya, terdapat pengujian dengan if, yaitu apabila $\_GET[‘page’] di klik, maka $page akan mengambil nilai dari $\_GET[‘page’], $jumlah berisi 3, artinya jumlah data yang ditampilkan adalah 3, lalu $no melakukan operasi aritmatika, yaitu $jumlah dikalikan dengan $page lalu dikurang dengan $jumlah ditambah 1, jika $page tidak diklik, maka akan menjalankan else, yaitu $no yang berisi 1.

Pengaturan paging :

<?= $pager->makeLinks(1, $tampil, $total, 'bootstrap') ?>

Artinya, terapat $pager digunakan untuk paging dan di dalam $pager menjalankan function makeLinks dengan menjalankan 1, artinya halaman dimulai dari 1, lalu $tampil yang berisi 3, artinya data yang ditampilkan adalah 3, $total yang menjalankan $count, artinya berapa banyak data pada $count, lalu menjalankan key bootstrap yang digunakan untuk merapikan paging / css

1. Melakukan hapus data pada menu menggunakan perintah delete.

Cth :

Di dalam hyperlink yang memanggil function delete :

<a href="<?= base\_url()?>/Admin/menu/delete/<?= $value['idmenu']?>"><img src="<?= base\_url('/icon/can.svg')?>"></a>

Artinya, terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Menu/delete/?value[‘idmenu’], jadi ketika link tersebut di klik, maka akan menjalankan localhost/restoran-ci4/Admin/Menu/delete. Lalu memanggil file img dengan src yang memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan ke dalam icon/can.svg, jadi artinya gambar yang diambil berada di dalam localhost/restoran-ci4 lalu masuk ke dalam folder icon dan file nya mempunyai nama can.svg.

Di dalam /Admin/Menu/delete :

public function delete($id = null)

    {

        $model = new Menu\_M();

        $model -> delete($id);

*return* redirect()->to(base\_url('/Admin/Menu'));

    }

Artinya, terdapat function delete yang mempunyai $id null maka $id bernilai kosong artinya tidak error jika nilai dari $id tidak diisi karena $id mempunyai nilai kosong. Lalu di dalam function delete objek $model menjalankan Menu\_M, yang di dalam nya terdapat $table tblmenu, artinya tabel yang dihubungkan oleh model ke dalam database adalah tblmenu, di dalam Menu\_M juga terdapat $primaryKey idmenu, artinya primarykey dari tblmenu adalah idmenu, apabila melakukan delete data / update data, di dalam model harus dideskripsikan primary key nya agar dapat menjalankan perintah delete data / update data. Lalu $model berisi perintah untuk delete data yang menjalankan $id, sehingga data yang di delete sesuai dengan $id yang dipilih. Apabila telah selesai melakukan delete data, maka menjalankan perintah untuk redirect / perintah untuk mengembalikan halaman, halaman akan dikembalikan ke dalam base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan ke dalam /Admin/Menu. Jadi, ketika setelah selesai melakukan delete data, maka halaman dikembalikan ke dalam halaman dari localhots/restoran-ci4/Admin/Menu.

1. Menambahkan form untuk insert data.

Di dalam button tambah data pada select/php :

<div class="col-4 mt-2">

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

    </div>

Artinya, terdapat <a> yang berisi button TAMBAH DATA yang di dalam nya terdapat hyperlink yang memanggil base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Menu/create, jadi ketika button di klik, maka akan mengarahkan kepada localhost/restoran-ci4/Admin/Menu/create.

Membuat function create di dalam /Admin/Menu :

public function create()

    {

        $model = new Kategori\_M();

        $kategori = $model -> findAll();

        $data = [

            'kategori' => $kategori

        ];

*return* view('menu/insert', $data);

    }

Artinya, terdapat function create dengan objek $model yang menjalankan Menu\_M yang di dalam nya terdapat $table tblmenu, artinya tabel yang dihubungkan oleh model ke dalam database adalah tblmenu. Lalu $kategori berisi $model yang menjalankan perintah untuk mengambil data / findAll. $data berisi data data yang akan dikrimkan ke dalam view, di dalam $data terdapat key kategori yang mengambil nilai dari $kategori. Lalu menjalankan view menu insert dan menjalankan $data.

1. Request digunakan untuk mengambil nilai dari form. Untuk menggunakan request, harus dinyalakan dulu request nya.

Code menyalakan request :

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

1. Menggunakan request untuk mengambil nilai dari form dan mengisi $data yang akan diinsert kan ke dalam database.

Cth :

public function insert()

    {

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

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

        $name = $file -> getName();

        $data = [

            'idkategori' => $request -> getPost('idkategori'),

            'menu' => $request -> getPost('menu'),

            'gambar' => $name,

            'harga' => $request -> getPost('harga')

        ];

        $model = new Menu\_M();

        $model -> insert($data);

        $file->move('./upload');

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

    }

Artinya, pada function insert terdapat $request yang digunakna untuk menyalakan request apabila request akan digunakan. Lalu $file berisi perintah request untuk mengambil nilai data / getFile pada gambar, lalu $name berisi $file yang menjalankan perintah untuk mengambil nama data / getName. Di dalam $data berisi data data yang akan diinsert kan / dimasukkan, yaitu pada key idkategori dengan value mengambil nilai dari form post menggunakan $request dan menjalankan perintah getPost untuk mengambil data pada form dengan name idkategori, jadi nilai pada form dengan name id kategori akan diambil dan disimpan ke dalam value dari key idkategori. Key menu berisi value yang mengambil nilai pada form post menggunakakn $request dan menjalankan perintah getPost untuk mengambil data pada form dengan name menu. Key gambar berisi value $name yang di dalam $name sudah berisi perintah untuk getFile pada form dengan name gambar. Key harga berisi $request dengan value yang mengambil nilai dari form menggunakan $request dan menjalankan perintah mengambil data / getPost pada form dengan name harga. Objek $model menjalankan Menu\_M yang di dalam nya sudah berisi data $table tblmenu, yaitu tabel yang dihubungkan oleh model ke dalam database adalah tblmenu, dan terdapat $allowedFields berisi kolom – kolom yang digunakan untuk memasukkan data. Lalu $model menjalankan perintah insert / memasukkan data dari $data. Setelah data berhasil diinsert, lalu $file akan memindahkan gambar ke dalam folder upload, jadi secara otomatis gambar akan dipindahkan ke dalam folder upload yang berada di luar dari folder ci4. Setelah melakukan insert data dan memasukkan gambar ke dalam folder upload, maka halaman akan dikembalikan ke dalam base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengna /Admin/Menu.

1. Mengisi $allowedFields di dalam Menu\_M, digunakan untuk memasukkan data pada kolom yang sudah ditentukan di dalam $allowedFileds untuk menginsert data.

Cth :

protected $allowedFields = ['idkategori', ' menu', 'gambar', 'harga'];

Artinya, di dalam $allowedFields terdapat kolom – kolom yang sudah ditentukan untuk menginsert data, yaitu kolom idkategori, kolom menu, kolom gambar, dan kolom harga.

1. Data yang dikirim tidak masuk, karena di dalam $allowedFields pada model yaitu Menu\_M, terdapat jarak pada tanda petik dan kolom yang dipanggil, jadi pada tanda petik dan nama kolom yang dipanggil, harus mepet tidak boleh ada spasi.
2. Update() adalah perintah yang digunakan untuk mengupdate data / merubah data.
3. Mengatur error pada update data.

Cth :

Mengatur pesan apabila terjadi error pada model Menu\_M :

protected $validationRules = [

     'menu' => 'alpha\_numeric\_space|min\_length[3]|is\_unique[tblmenu.menu]',

     'harga' => 'numeric'

    ];

    protected $validationMessages = [

        'menu' =>[

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

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

            'min\_length' => 'Isi minimal dari data yang dimasukkan adalah 3 huruf',

            'is\_unique' => 'Nama menu sudah terdaftar, silahkan menggunakan nama yang lain'

        ],

        'harga' =>[

            'numeric' => 'Mohon maaf, di dalam pengisian harga, tidak boleh menggunakan simbol,

            harus diisi dengan angka.'

        ]

    ];

Artinya, di dalam $validationRules yang mengatur 2 kolom, yaitu pada kolom menu, rulesnya adalah, alpha\_numeric\_space artinya, menu 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 tblmenu kolom menu. Dan mengatur kolom harga dengan peraturan numeric, yaitu kolom harus diisi dengan angka, apabila diisi selain angka, maka akan terjadi error.

Di dalam $validationMessages berisi pesan – pesan yang ditampilkan apabila update data yang dimasukkan tidak sesuai dengan peraturan, $validastionMessages mengatur kolom menu dan kolom harga, dimana pesan di dalam kolom menu adalah, apabila data yang dimasukkan menggunakan symbol, maka pesan yang ditampilkan adalah pesan pada key alpha\_numeric\_space, lalu apabila data yang dimasukkan berisi kurang dari 3, maka akan menampiilkan pesan error pada min\_length, dan apabila data yang dimasukkan sudah terdaftar, maka akan menampilkan pesan error yang ada pada is\_unique. Dan di dalam kolom harga, apabila data yang dimasukkan tidak berupa angka, maka akan menampilkan pesan error pada numeric, yaitu data yang dimasukkan harus berupa angka.

Mengatur error di dalam function update pada Menu :

*if* ($model -> update($id, $data)===false) {

            $error = $model->errors();

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

*return* redirect()->to(base\_url("/Admin/Menu/find/$id"));

        }*else* {

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

        }

Artinya, terdapat pengujian yaitu apabila $model perintah update data pada $id dan update pada $data yang dilakukan salah / tidak mengikuti peraturan, maka akan menampilkan $error yang berisi $model menjalankan perintah errors(), lalu di dalam session() berisi perintah untuk mengatur pesan yang ditampilkan ketika melakukan proses dengan key info dan menjalankan $error. Tetapi apabila update yang dijalankan mengikuti peraturan, maka akan mengarahkan halaman ke pada base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Menu.

Mengatur pemanggilan dari error pada file update.php di dalam folder views :

*if* (!empty(session() -> getFlashdata('info'))) {

            echo '<div class="alert alert-danger" role="alert">';

            $error = session() -> getFlashdata('info');

*foreach* ($error as $key => $value) {

                echo $key." : " .$value;

                echo "<br>";

            }

            echo '</div>';

        }

Artinya, apabila session() getFlashdata yang menjalankan key info tidak kosong, karena data pada key info yang akan diambil sudah diatur, jadi ketika mau mengambil data key info, datanya sudah ada, jadi tidak kosong, maka akang menjalankan tampilan pesan yang mempunyai background merah, lalu menjalankan $error dengan session() yang berisi perintah untuk mengambil data / getFlashdata yang menjalankan key info, karena kolom yang diatur apabila terjadi error lebih dari satu, maka untuk menampilkan pesan error menggunakan foreach dengan $error sebagai key dan menampilkan $value.

1. Membuat file Pelanggan.php di dalam folder admin lalu di dalam function index berisi $model yang memanggil Pelanggan\_M yang di dalam nya terdapat $table tblpelanggan, lalu $data berisi data – data yang akan dikirimkan ke view, dan menjalankan view pelanggan/select. Di dalam view pelanggan/select terdapat pengaturan tabel yang akan ditampilkan yang datanya diambil dari tblpelanggan.
2. Melakukan pengambilan data apabila akun aktif, maka akan menampilkan tulisan aktif dan apabila akun tidak aktif, maka akan menampilkan tulisan non-aktif.

Cth :

<?php

*if* ($value['aktif']==1) {

                            $aktif = "AKTIF";

                        }*else* {

                            $aktif = "NON-AKTIF";

                        }

                    ?>

                    <td>

                        <a href="<?= base\_url()?>/Admin/Pelanggan/update/<?= $value['idpelanggan']?> /<?= $value['aktif']?>"><?= $aktif?></a>

                    </td>

Artinya, terdapat pengujian yaitu, apabila $value[‘aktif’] nilai nya sama dengan 1, maka akan menjalankan $aktif = AKTIF, tetapi apabbila $value[‘aktif’] bernilai 0 atau tidak bernilai 1, maka akan menjalankan $aktif = NON-AKTIF. Terdapat td berisi hyperlink yang memanggil base\_url() dengan alamat localhpst/restoran-ci4 lalu disambungkan dengan /Admin/Pelanggan/update/dengan mengambil value dari $value[‘idpelanggan’]/lalu disambungkan dengan mengambil nilai status aktif. , jadi artinya apabila link di klik, maka akan mengarahkan ke tampilan method update di dalam file pelanggan pada folder Admin, dan di dalam hyperlink memanggil nilai dari $aktif dan menampilkan status akun dari setiap data pelanggan.

1. Melakukan update data pada kolom aktif.

Cth :

public function update($id = null, $isi = 1)

    {

        $model = new Pelanggan\_M();

*if* ($isi == 0) {

            $isi = 1;

        } *else* {

            $isi = 0;

        }

        $data = [

            'aktif' => $isi

        ];

        $model -> update($id, $data);

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

    }

Artinya, function update terdapat $id yang bernilai kosong dan $isi yang bernilai 1, lalu di dalam function update terdapat objek $model yang memanggil Pelanggan\_M, yang di dalamnya terdapat $table tblpelanggan, artinya tabel yang dihubungkan oleh model ke dalam database adalah tblpelanggan, dan di dalam Pelanggan\_M terdapat $allowedFields yaitu kolom [‘aktif’], lalu terdapat pengujian, yaitu apabila $isi bernilai 0, maka $isi akan diubah nilai nya menjadi 1, dan apabila $isi bernilai 1, maka nilainya akan dirubah menjadi 0. Lalu $dasta berbentuk array digunakan untuk mengirimkan data yang akan diupdate, $data berisi key aktif dengan value mengambil nilai dari $isi. $model menjalankan perintah update pada $data sesuai dengan $id. Setelah melakukan update data, maka halaman akan dikembalikan ke dalam halaman base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Pelanggan.

1. Melakukan delete data di dalam pelanggan.php

Cth :

  public function delete($id = null)

    {

        $model = new Pelanggan\_M();

        $model -> delete($id);

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

    }

Artinya, function delete terdapat $id dengan nilai null berarti $id bernilai kosong, lalu objek $model yang memanggil Pelanggan\_M yang berisi $table tblpelanggan, artinya table yang dihubungkan oleh model ke dalam database adalah tblpelanggan, lalu terdapat $primaryKey [‘idpelanggan’], $primaryKey digunakan untuk melakukan delete data. $model berisi perintah untuk delete data pada $id. Apabila selesai melakukan delete data, maka halaman akan dikembalikan ke dalam halaman base\_url() dengan alamat localhost/restoran-ci4 lalu disambungkan dengan /Admin/Pelanggan.

1. Query builder adalah class yang disediakan oleh codeigniter yang digunakan untuk melakukan perintah pada database, seperti select, update, insert, dan delete.
2. Perintah koneksi ke database.

Cth :

$db = \Config\Database::connect();

1. Memilih data dari tblorder menggunakan perintah select pada query.

Cth :

public function index()

    {

        $db = \Config\Database::connect();

        $sql = "SELECT \* FROM tblorder ORDER BY status ASC";

        $result = $db -> query($sql);

        $row = $result -> getResult();

        echo "<pre>";

        print\_r($row);

        echo "</pre>";

    }

Artinya, function index berisi $db yang digunakan untuk menyambungkan ke dalam database, lalu $sql yang berisi perintah untuk memilih data / SELECT data dari tblorder yang diurutkan dari status dan diurutkan dengan metode ASC / dari kecil ke besar. Lalu $result berisi $db yang menjalankan perintah query pada sql. $row berisi $result yang menjalankan perintah getResult / perintah untuk mengambil data. $row ditampilkan menggunakan print\_r dan dirapikan tampilannya menggunakan echo “pre”.

1. Menampilkan salah satu kolom.

Cth :

public function index()

    {

        $db = \Config\Database::connect();

        $sql = "SELECT \* FROM tblorder ORDER BY status ASC";

        $result = $db -> query($sql);

        $row = $result -> getResult();

        echo $row[0] -> tglorder;

}

Artinya, function index berisi $db yang digunakan untuk menyambungkan ke dalam database, lalu $sql yang berisi perintah untuk memilih data / SELECT data dari tblorder yang diurutkan dari status dan diurutkan dengan metode ASC / dari kecil ke besar. Lalu $result berisi $db yang menjalankan perintah query pada sql. $row berisi $result yang menjalankan perintah getResult / perintah untuk mengambil data, karena hasil dari pengambilan nilai adalah array yang terdapat objek di dalamnya, maka untuk menambil nilai salah satu kolom yaitu dengna memanggil index yang akan dipanggil dan kolom yang akan ditampilkan. $row memanggil index 0 dan memilih kolom tglorder untuk ditampilkan.

1. Menampilkan banyak data di dalam query menggunakan foreach.

Cth :

public function index()

    {

        $db = \Config\Database::connect();

        $sql = "SELECT \* FROM tblorder ORDER BY status ASC";

        $result = $db -> query($sql);

        $row = $result -> getResult();

*foreach* ($row as $key ) {

            echo $key -> total;

            echo "<br>";

        }

}

Artinya, function index berisi $db yang digunakan untuk menyambungkan ke dalam database, lalu $sql yang berisi perintah untuk memilih data / SELECT data dari tblorder yang diurutkan dari status dan diurutkan dengan metode ASC / dari kecil ke besar. Lalu $result berisi $db yang menjalankan perintah query pada sql. $row berisi $result yang menjalankan perintah getResult / perintah untuk mengambil data, lalu untuk menampilkan banyak data, maka menggunakan foreach, di dalam foreach $row sebagai $key, $key memanggil kolom total. Maka semua hasil dari kolom total akan ditampilkan. Karena hasil dari getResult adalah berbentuk object, maka untuk memanggil kolom diberi tanda panah / ->.

1. Menampilkan hasil dengan menggunakan array.

Cth :

public function index()

    {

        $db = \Config\Database::connect();

        $sql = "SELECT \* FROM tblorder ORDER BY status ASC";

        $result = $db -> query($sql);

        $row = $result -> getResult('array');

        echo $row[2]['idpelanggan'];

}

Artinya, function index berisi $db yang digunakan untuk menyambungkan ke dalam database, lalu $sql yang berisi perintah untuk memilih data / SELECT data dari tblorder yang diurutkan dari status dan diurutkan dengan metode ASC / dari kecil ke besar. Lalu $result berisi $db yang menjalankan perintah query pada sql. $row berisi $result yang menjalankan perintah getResult / perintah untuk mengambil data, hasil data akan berbentuk array karena di dalam getResult diberi array, maka hasilnya akan menjadi array, sehingga untuk pemanggilan dari data, maka memanggil index dan kolom yang ingin ditampilkan. Seperti, $row[2][‘idpelanggan’], maka $row akan menampilkan data dari index 2 dengan kolom idpelanggan.

1. Menampilkan hasil getResult berbentuk array menggunakan foreach.

Cth :

public function index()

    {

        $db = \Config\Database::connect();

        $sql = "SELECT \* FROM tblorder ORDER BY status ASC";

        $result = $db -> query($sql);

        $row = $result -> getResult('array');

*foreach* ($row as $key) {

            echo $key['tglorder'];

            echo "<br>";

        }

}

Artinya, function index berisi $db yang digunakan untuk menyambungkan ke dalam database, lalu $sql yang berisi perintah untuk memilih data / SELECT data dari tblorder yang diurutkan dari status dan diurutkan dengan metode ASC / dari kecil ke besar. Lalu $result berisi $db yang menjalankan perintah query pada sql. $row berisi $result yang menjalankan perintah getResult / perintah untuk mengambil data, di dalam foreach terdapat $row sebagai $key, lalu $key memanggil kolom tglorder dan ditampilkan menggunakan echo.

**Saya Belum Mengerti**