|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| PHP FRAMEWORK CI minggu ke 19 (video 20-34 Tutorial Codeigniter-4) | 95 |

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

1. Permulaan dari membuat sebuah project baru yaitu Project Restoran menggunakan CI-4 dari Tutorial CodeIgniter-4 pada bagian Menu Aplikasi. Untuk yang pertama adalah menyiapkan menu-menu yang belum dibuat seperti menu pelanggan, order, dan order detail. Lalu merapikan tampilan awalnya pada browser dengan menambah row agar terbagi menjadi seperti memiliki ruangan-ruangan sendiri juga mengatur marginnya agar terjarak dengan element lainnya.

2. Tutorial CodeIginiter-4 pada bagian Menu Kategori. Mengunduh sebuah icon dari bootstrap->documents->extend->icon->octions lalu pilih icon yang akan digunakan dan mendownload nya dalam bentuk file.svg, lalu icon-icon ini diletakkan di dalam folder icon di dalam restoran ci-4. Mengganti tulisan menjadi sebuah icon dengan code <img src="<?= base\_url('/icon/trash.svg') ?>"> yang berarti dia mengambil gambar dari folder base\_url (localhost/restoran-ci4) masuk ke dalam folder icon dan memilih file trash.svg yang berarti icon dari sampah. Mengatur tampilan form dengan menambahkan div col untuk judul dan form nya, menambahkan form-group disetiap input text, submit, dll. Nya dan menambahkan class form-control agar tampilannya lebih bagus disetiap input text nya. Membuat form update agar data yang diupdate tidak boleh sama dengan sebelumnya, tidak boleh sama dengan data yang sudah ada, tidak boleh dibawah 3 huruf, dan tidak boleh menggunakan simbol maka di dalam controllers class Kategori pada bagian function update nya diganti dengan

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"));

        }

    }

Yang berarti jika sudah menekan simpan setelah melakukan update dan apabila data yang dimasukkan memenuhi kategori error yang sama seperti ketika memasukkan insert data maka dia akan menjalankan error lalu kembali ke tampilan form update data sesuai idkategori yang mau diupdate. Lalu apabila data yang dimasukkan tidak eror dia akan menjalankan update data. Agar tau kesalahan eror nya apa maka di dalam views update.php ditambahkan <?php echo session()->getFlashdata('info'); ?> Agar dia menampilkan eror nya apa

3. Tutorial CodeIgniter-4 pada bagian Penomoran Baris. Jadi nomor yang ada di kolom pada menu kategori itu tiap halamannya masih diulang dari nomer 1 lagi, agar tampilan nomornya pada halaman pertama ada nomor 1 sampai 3, dan halaman kedua adalah lanjutannya yaitu 4 dst. Maka di dalam file select.php ditambahkan

<?php

if (isset($\_GET['page\_page'])) {

    $page = $\_GET['page\_page'];

    $jumlah = 3;

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

} else {

    $no = 1;

}

?>

Yang berarti apabila dia menekan salah satu halaman yaitu nomer page\_page nya (page awal ini adalah nama yang ada di dalam controllers kategori yaitu bagian 'kategori' => $model->paginate(3, **'page'**) dan di dalam views select bagian <?= $pager->links(**'page'**, 'bootstrap') ?>) misalkan menekan halaman 1 maka $page nya adalah 1, jika menekan page 2 maka $page nya 2, dst. Setelah itu pada $no adalah tempat penjumlahan yang misalkan menekan halaman 2, $jumlah nya 3 dikali 2 sama dengan 6 lalu dikurangi 3 sama dengan 3 dan ditambah 1, maka nomor yang dimulai di halaman 2 adalah mulai nomor 4. Dan apabila belum memilih page atau tampilan awal menu kategori dia akan menampilkan halaman pertama. Agar $no ini berjalan maka di bawah tr th ditambahkan <?php $no ?> yang berarti menjalankan nomor yang akan tampil disetiap halaman berdasarkan $no nya.

4. Tutorial CodeIgniter-4 pada bagian Merapikan Tampilan Kategori. Pertama yaitu menambahkan alert yang diletakkan untuk pesan error nya, jadi pada php yang menampilkan error dalam file unsert dan update ditambahkan div class untuk alertnya seperti ini

<div class="col">

    <?php

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

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

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

        echo '</div>';

    }

    ?>

</div>

Jadi div col ini berisi sebuah php yang apabila ada eror, maka dia akan menampilkan error yang sudah dilengkapi dengan div class alert, jadi info error nya berada di dalam div class alert. Untuk alert nya pilih yang alert-danger karena pesan error biasanya berwarna merah seperti sebuah peringatan keras untuk tidak memasukkan data yang membuat error.

5. Tutorial CodeIgniter-4 pada bagian Menu Restoran (1). Mengatur menu “Menu” yang pertama yaitu membuat models nya dulu, model ini diberi nama Menu\_M yang berisi protected $table = ‘tblmenu’; yang berarti dia mengambil data dari tblmenu yang ada dalam database. Lalu membuat controller yang bernama menu.php, untuk controller menu yang sebelumnya diubah saja menjadi menu2 dan buat menu yang baru yaitu menu.php, menu.php ini mula-mula ditambahi namespace\controllers\admin karena dia berada di dalam controllers bagian folder admin, lalu ditambahi use App\Controllers\BaseController agar dia tersambung dengan basecontroller, lalu ditambahi use App\Models\Menu\_M agar dia tersambung dengan models dengan class Menu\_M, Controller Menu ini berisi sebuah class bernama menu yang isinya masih 1 function yaitu index, dimana function ini berisi seperti function read yang ada di dalam class Kategori. Codenya

public function index()

    {

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

        $model = new Menu\_M();

        $data = [

            'judul' => 'DATA MENU',

            'menu' => $model->paginate(3, 'page'),

            'pager' => $model->pager

        ];

        return view("menu/select", $data);

    }

Jadi function ini membuat sebuah object $model untuk class Menu\_M yang ada di dalam model lalu membuat variable yang bisa mengambil data dari class Menu\_M dari tblmenu lalu diletakkan atau ditampilkan di select.php yang ada folder menu di views. Untuk views menu select nya bisa diambil atau copas dari select kategori yang ada di views lalu mengganti semua kata kategori menjadi menu dengan ctrl+shift+L, lalu memperbaiki kolom tabel nya karena tabel kategori dengan tabel menu berbeda, jadi menyesuaikan tabel yang sudah ada di database. Untuk menampilkan gambar di dalam kolom sesuai dengan data nya maka code nya:

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

Jadi gambar ini berukuran 100px, lalu dia diambil dari base\_url (localhost/restoran-ci4) masuk ke folder upload dan mengambil gambar sesuai dengan data yang ada di database disetiap barisnya.

6. Tutorial CodeIgniter-4 pada bagian Menu Restoran(2). Pada tutorial ini untuk menambahkan select option dari kategori yang diletakkan di select.php Menu. Caranya dengan menambahkan div row diantara div row judul dan form, row ini berisi seperti ini

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

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

</form>

Jadi form ini menyambungkan dengan function option yang ada di controller dimana function ini menyambungkan dengan views option.php di dalam folder template.

Lalu apabila setelah menekan atau memilih option dia akan menyambungkan dengan function read di dalam controller Menu. Di dalam option.php ditambahkan onchange="this.form.submit()" pada bagian select optionnya. Penambahan onchange ini berfungsi untuk menambahkan url berupa nama dari select form nya lalu = idkategorinya. Jadi dari url yang menyambungkan ke function read, dia juga menyertakan nama select form dan idkategori dari data yang dipilih. Jadi url nya seperti ini <http://localhost/restoran-ci4/admin/menu/read?idkategori=30>.

7. Tutorial CodeIgniter-4 pada bagian Menu Restoran(3). Pada bagian ini adalah menampilkan menu dari pencarian atau dari pilihan di select option berdasarkan idkategori nya yg sama. Jadi di dalam models Menu\_M pada bagian function read nya diisi dengan

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

            ];

            return view("menu/cari", $data);

        }

    }

Yang berarti ketika memilih dari select option dia akan mengambil data idkategori nya, misalkan memilih makanan maka dia akan mengambil idkategori makanan yaitu 28, setelah itu idkategori yang diambil ini dijadikan variable $id, setelah itu membuat objek untuk menyambungkan dengan model Menu\_M, setelah itu dari model Menu\_M dia akan mengambil semu data yang idkategori nya sama dengan $id yang dipilih. Untuk data yang akan ditampilkan, dia tergantung dengan jumlah $tampil nya, jika sudah dia akan menampilkan file cari yang ada didalam folder menu di views. Di dalam file cari ini berisi seperti select.php yang nanti tampilannya akan diisi dengan $data dari function read.

8. Tutorial CodeIgniter-4 pada bagian Delete Data dari data menu. Untuk itu di dalam models controller Menu ditambahkan function delete yang sudah dibuat di dalam controler Kategori, jadi nanti isi dari function ini tinggal diubah objek nya untuk models Menu\_M dan return view nya mengarah ke menu.php. Agar delete ini berjalan pada controller menu, maka di dalam models Menu\_M ditambahi protected $primaryKey nya, primarykey yang ada di dalam tblmenu yaitu idmenu, jadi protected $primaryKey = ‘idmenu’;

9. Tutorial CodeIgniter-4 pada bagian form insert menu. Untuk menginsert data ke dalam tblmenu, yaitu memperbaiki tampilan form insert data agar sesuai dengan kolom apa saja yang ada di dalam tblmenu. Karena insert data ini juga memilih kategori untuk menu nya, maka ditambahkan form group seperti ini

<div class="form-group">

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

            <select class="form-control" name="idkategori" id="kategori">

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

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

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

                    </option>

                <?php endforeach; ?>

            </select>

</div>

Form ini diambil dari file form.php yang berfungsi untuk memilih kategori nya. Setelah itu menambahkan form lainnya untuk kolom lainnya juga, jika sudah selanjutnya adalah ke controller menu pada bagian function create diisi seperti ini

public function create()

    {

        $model = new Kategori\_M();

        $kategori = $model->findAll();

        $data = [

            'kategori' => $kategori

        ];

        return view("menu/insert", $data);

    }

Tujuannya agar data dari tblkategori yaitu kategori-kategori nya bisa ditampilkan di insert data milik menu.

10. Tutorial CodeIgniter-4 pada bagian Insert Menu. Karena sebelumnya sudah membuat form untuk insert datanya, selanjutnya adalah proses menginsert data dari form ke tblmenu nya. Caranya yaitu dengan menggunakan request di dalam function insert pada controller menu untuk mengambil data yang sudah diinsert dari form nya. Codenya seperti ini $request = \Config\Services::request(); di dalam function insert berisi seperti ini

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"));

}

Jadi setelah dia mengambil data dari form, selanjutnya adalah membuat variable untuk menyimpan request gambar dan selanjutnya akan diberi nama sesuai dengan gambar yang telah dipilih. Setelah itu di dalam $data ini adalah tempat untuk menambahkan data sesuai nama form nya, di dalam tblmenu untuk kolom idkategori akan diisi sesuai dengan idkategori yang dipilih, untuk kolom menu, gambar, dan harga juga seperti itu. Setelah itu menyambungkan dengan models Menu\_M untuk menjalankan code-code diatasnya tadi yang pertama ada untuk menginsert data maka di dalam Menu\_M nya harus ada protected $allowedFields = ['idkategori', 'menu', 'gambar', 'harga']; yang berarti dia akan menginsert $data ke dalam situ sesuai dengan nama kolomnya. Selanjutnya ada move file gambar ke folder upload yang berfungsi untuk mengupload gambar yang dipilih ke dalam folder upload. Setelah melakukan insert dia akan kembali ke tampilan awal menu.

11. Tutorial CodeIgniter-4 pada bagian Menu Restoran(7) yaitu Update Data. Yang pertama dilakukan adalah di dalam controller menu ditambahkan function find yang berfungsi untuk mengambil data dari salah satu data menu yang dipilih untuk diupdate. Maka harus ada code nya seperti ini $model = new Menu\_M(); $menu = $model->find($id); yang berarti mengambil data dari tblmenu berdasarkan id yang dipilih. Dan $kategorimodel = new Kategori\_M(); $kategori = $kategorimodel->findAll(); yang berfungsi untuk mengambil semua data dari tblkategori. Setelah itu

 $data = [

            'judul' => 'UPDATE DATA',

            'menu' => $menu,

            'kategori' => $kategori

];

return view("menu/update", $data);

yang berfungsi untuk masuk ke dalam file update.php di dalam folder menu views lalu dengan data menu dan kategori berdasarkan menu dan kategori yang dipilih. Setelah itu di dalam file update.php bisa diisi dengan mengcopas file insert.php, pada bagian select optionnya diubah menjadi seperti ini

<option <?php if ($value['idkategori'] == $menu['idkategori']) echo "selected" ?> value="<?= $value['idkategori'] ?>">

         <?= $value['kategori'] ?>

</option>

Yang berarti jika idkategori dari kategori nya sama dengan idkategori dengan menu nya maka tampilan awal select optionnya sesuai dengan data yang dipilih atau selected. Lalu untuk form menu, harga, dll ditambahkan value="<?= $menu['menu'] ?>" di input type nya dan di dalam kurung staples nya sesuai dengan nama input type nya. dan terakhir ditambahkan

<input type="hidden" name="gambar" value="<?= $menu['gambar'] ?>" required class="form-control">

<input type="hidden" name="idmenu" value="<?= $menu['idmenu'] ?>" required class="form-control">

Yang berfungsi untuk menyimpan datanya nanti. Setelah itu di dalam form actionnya diubah menjadi action="<?= base\_url('/admin/menu/update') ?>" yang nantinya setelah menekan simpan dia akan mengaktifkan function update yang ada di dalam controller menu. Di dalam function update berisi seperti ini

public function update()

    {

        $id = $this->request->getPost('idmenu');

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

        $name = $file->getName();

        if (empty($name)) {

            $name = $this->request->getPost('gambar');

        } else {

            $file->move('./upload');

        }

        $data = [

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

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

            'gambar' => $name,

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

        ];

        $model = new Menu\_M();

        $model->update($id, $data);

        return redirect()->to(base\_url("/admin/menu"));

    }

Yang berarti dia akan mempost idmenu yang dipilih lalu apabila gambar tidak diupdate dan dibiarkan kosong maka dia akan tetap menggunakan gambar yang sebelumnya, tetapi apabila gambar diubah dengan gambar lain yang diambil dari folder selain upload maka dia akan mengupload gambar itu dan memasukkannya ke dalam folder upload. Selanjutnya data-data yang sudah dipilih di dalam form akan akan dipost sebagai perubahan data yang telah diupdate lalu tampilannya akan kembali ke tampilan menu yang telah diperbarui.

12. Tutorial CodeIgniter-4 pada bagian Menu Restoran(8) yaitu Coding Validation. Jadi validation ini berfungsi untuk mengoreksi data yang akan diinsert dan diupdate agar data tidak sama, tidak kurang dari 3 huruf, dan tidak menggunakan simbol. Untuk membuat error apa saja maka di dalam models Menu\_M ditambahkan

protected $validationRules = [

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

        'harga' => 'numeric'

    ];

    protected $validationMessages = [

        'menu' => [

            'alpha\_numeric\_space' => 'Tidak boleh menggunakan simbol',

            'min\_length' => 'Minimal 3 huruf',

            'is\_unique' => 'Data sudah digunakan'

        ],

        'harga' => [

            'numeric' => 'Harus angka',

        ]

    ];

Yang berarti pada form input menu dan harga harus memenuhi syarat yang ada di $validationRoules lalu $validationMessages akan mengirimkan pesan sebagai keterangan apabila ada error nanti. Setelah itu kembali ke controller Menu pada bagian function insert ditambahi

if ($model->insert($data) === false) {

            $error = $model->errors();

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

            return redirect()->to(base\_url("/admin/menu/create"));

        } else {

            $file->move('./upload');

            return redirect()->to(base\_url("/admin/menu"));

        }

yang berarti apabila data yang dimasukkan terjadi kesalahan, maka dia akan menjalankan error dan kembali ke tampilan insert sebelum diisi data. Dan apabila data sudah lolos dari uji error maka dia akan menginsert dan kembali ke tampilan awal menu. Begitu juga pada function update ditambahi if else seperti ini sesuai dengan nama-nama yang dipakai juga. Lalu agar error muncul di atas form ketika kembali ke tampilan awal form maka di dalam file insert.php ditambahkan

<div class="col">

    <?php

    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>';

    }

    ?>

</div>

Yang berarti apabila data sudah diisi dia akan menjalankan untuk diseleksi error nya, jika terjadi kesalahan maka dia akan muncul sebuah peringatan warna merah dengan pesan error sesuai dengan nama form dan error nya yang mana, misalkan pada form menu error karena data yang dimasukkan sudah ada maka dia akan muncul peringatan menu=>data sudah digunakan, jadi nama form ini sebagai key nya dan info error nya sebagai value nya. begitu juga di dalam update.php agar sama seperti insert maka ditambahkan div col yang berisi if else seperti itu tadi.

13. Tutorial CodeIgniter-4 pada bagian Select Pelanggan atau menu “Pelanggan”. Pertama adalah membuat models untuk pelanggan dulu di dalam models, model pelanggan ini diisi dengan

namespace App\Models;

use CodeIgniter\Model;

class Pelanggan\_M extends Model

{

    protected $table = 'tblpelanggan';

}

Yang berarti dia akan menyambungkan dengan tabel pelanggan yang ada di database. Selanjutnya adalah membuat controller pelanggan dulu di dalam folder Admin controller. Controller pelanggan ini diisi sama dengan controller kategori untuk awalannya (sebelum class) lalu class nya diberi nama Pelanggan dan isinya baru 1 function yaitu function index. Function index nya seperti ini

public function index()

    {

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

        $model = new Pelanggan\_M();

        $data = [

            'judul' => 'DATA PELANGGAN',

            'pelanggan' => $model->paginate(3, 'page'),

            'pager' => $model->pager

        ];

        return view("pelanggan/select", $data);

    }

Yang berarti dia membuat paging untuk data yang diambil dari tblpelanggan yang sebelumnya sudah disambungkan melalui models dan setelah itu dia akan menuju ke file select.php yang ada di dalam folder pelanggan di view dengan mengisi datanya dari function ini tadi. Untuk file select.php nya bisa dicopas dari select yang ada di folder kategori lalu diubah semua kata kategorinya menjadi pelanggan dengan Ctrl+Shift+L dan mengubah th td nya agar sama dengan tblpelanggan yang ada di database.

14. Tutorial CodeIgniter-4 pada bagian Pelanggan Restoran(2) yaitu Coding Delete dan Update. Jadi tujuannya adalah agar status aktif bisa di nonaktifkan dan status nonaktif bisa di aktifkan. Caranya di dalam select.php yang ada di pelanggannya pada bagian td atau kolom dari statusnya diubah seperti ini

<td>

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

</td>

Dan diatas td nya ditambahkan

<?php

if ($value['aktif'] == 1) {

$aktif = "AKTIF";

} else {

$aktif = "NON AKTIF";

}

?>

Jadi kolom status ini akan terisi dari $aktif yaitu apabila kolom aktif pada database nya 1 maka dia tertuliskan AKTIF dan apabila 0 maka dia tertuliskan NON AKTIF. Lalu jika link ini ditekan dia akan mengarahkan ke dalam function update yang ada di dalam controller pelanggan. Di dalam function update diisi dengan

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"));

    }

Karena function ini membutuhkan tempat untuk mengetahui primarykey nya maka di dalam models pelanggan ditambahkan protected $primaryKey = 'idpelanggan'; dan ditambahkan protected $allowedFields = ['aktif']; yang berarti kolom ini boleh diubah oleh models. Jika sudah maka di dalam function juga bisa dijalankan. Di dalam function dia akan menyambungkan dengan models pelanggan\_m lalu membuat if else yang berarti apabila $isi atau status dari pelanggan nya 0 maka dia akan berubah menjadi 1 dan begitu juga sebaliknya. Setelah itu buat variable array assosiatif yang berfungsi untuk menyimpan key dengan value nya yaitu untuk kolom aktif diisi dengan $isi yang berarti menggunakan if else tadi lalu models akan menjalankan id pelanggan yang dipilih agar terupdate $data nya atau kolom statusnya sesuai if else juga dan tampilannya akan kembali ke tampilan awal pelanggan. Dan untuk delete nya di dalam controller pelanggan ditambahkan function delete seperti ini

public function delete($id = null)

    {

        $model = new Pelanggan\_M();

        $model->delete($id);

        return redirect()->to(base\_url("/admin/pelanggan"));

    }

Yang berarti dia akan mendelete data sesuai id pelanggan yang dipilih dengan menekan tanda hapus di kolom aksinya.

15. Tutorial CodeIgniter-4 pada bagian Order Restoran(1) yaitu Query Builder. Jadi cara ini adalah menyambungkan ke database tanpa menggunakan model. Untuk itu di dalam function yang kita buat misalkan function index yang berada di class Order kita tambahkan

$db = \Config\Database::connect();

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

$db itu berfungsi untuk menyambungkan dengan database dan $sql nya berfungsi untuk memilih table yang ada di dalam database yaitu tblorder dan apabila tampilkan datanya dia akan mengurutkan data berdasarkan statusnya dari yang terkecil. Agar kedua code ini berjalan maka ditambahkan $result = $db->query($sql); yang berfungsi untuk menyambungkan antara $db dengan $sql menggunakan query. Untuk menampilkan data dari tblorder tadi ada 2 cara yaitu

$row = $result->getResult('array');

echo $row[0]['idorder'];

echo "<br>";

foreach ($row as $key) {

       echo $key['tglorder'];

       echo "<br>";

}

$row ini dijadikan result nya berupa array, jadi data tblorder ini adalah array yang memiliki banyak indeks yaitu data, dan setiap data akan dibuat menjadi array yang berisi kolom-kolom dari data tadi. Untuk itu apabila hanya ingin menampilkan 1 kolom dari sebuah data maka menggunakan echo $row[0][‘idorder’]; yang berarti menampilkan data dari indeks ke 0 dan memilih kolom idordernya, lalu apabila ingin menampilkan kolom yang sama dari semua data maka menggunakan foreach seperti itu yang berarti $row atau data menjadi $key dan nanti jika di echo $key[‘tglorder’]; maka dia akan menampilkan kolom tglorder dari semua data yang ada. Cara yang kedua untuk menampilkan kolom dari data adalah

$row = $result->getResult();

echo $row[0]->idorder;

echo "<br>";

echo $row[0]->total;

foreach ($row as $key) {

     echo $key->tglorder;

     echo "<br>";

}

Jadi getResult nya kosong atau tidak diisi dengan array seperti sebelumnya, maka untuk menampilkan kolom dari 1 data yaitu dengan mengecho $row atau data dan memilih data yang keberapa dengan menambahkan kurung staples sebagai tanda indeks ke berapa yang mau ditampilkan missal $row[0] berarti indeks ke 0 lalu ->idorder yang berarti dia akan memilih kolom idorder dari data indeks ke 0. Lalu apabila ingin menampilkan salah satu kolom dari semua data maka menggunakan foreach seperti itu.

**Saya Belum Mengerti**

1.