|  |  |
| --- | --- |
| Materi | Nilai |
| Front End Aplikasi Restoran  (Tugas minggu ke – 21) | 95 |

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

1. Menampilkan data – data menu apa saja yang ada di dalam kategori. Terdapat dua cara,

Cara pertama :

public function index()

    {

        $model = new Kategori\_M();

        $data = [

            'menu' => $model -> findAll()

        ];

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

    }

Artinya, cara pertama yaitu dengan membuat objek $model yang membaca Kategori\_M, setelah itu, $data berisi data yang akan dikirimkan ke dalam view, di dalam $data terdapat key menu dengan value $model yang menjalankan perintah findAll / mengambil semua data. Setelah itu, membaca template/pelanggan dan menjalankan $data.

Cara kedua :

public function index()

    {

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

        $sql = "SELECT kategori FROM tblkategori ORDER BY kategori ASC";

        $result = $db -> query($sql);

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

        $data = [

            'menu' => $row

        ];

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

    }

Artinya, cara kedua yaitu dengan menyambungkan function ke dalam database menggunakan aliasi, $db digunakan untuk menyambungkan function ke dalam database, $sql berisi perintah untuk memilih data / SELECT kategori dari tblkategori yang diurutkan dari kategori dengan urutan ASC, yaitu mengurutkan data dari yang terkecil ke yang terbesar. $result berisi $db dengan perintah query yang menjalankan perintah dari $sql. $row berisi $result yang di dalamnya terdapat perintah untuk mengambil data / getResult dengan hasil menjadi berbentuk array. $data berisi data data yang akan dikirimkan ke dalam view, $data berisi key menu dengan value yaitu $row. Setelah itu, membaca template/pelanggan dan mengirimkan $data.

1. Melakukan pemanggilan dari $data yang sudah dikirimkan ke dalam view.

Cth :

<?php *if*(!empty($menu)) { ?>

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

          <?php *foreach*($menu as $key) : ?>

                   <li class="list-group-item">

<a href="#"><?= $key['kategori']?></a>

</li>

            <?php *endforeach* ?>

 </ul>

  <?php } ?>

Artinya, terdapat pengujian, yaitu apabila $menu tidak kosong, maka akan menjalankan ul dengan option yang diambil dari foreach, yaitu terdapat foreach dengan $menu as $key, di dalam foreach terdapat li yang memanggil $key[‘kategori’] dengan terdapat hyperlink di dalamnya.

1. Karena data yang dikirimkan ke dalam template/admin adalah memanggil data dari database, maka dalam setiap file yang dipasangkan homepage harus mengirimkan data yang memanggil kategori lagi.
2. Menampilkan daftar menu dengan menyambungkan ke dalam database dan membaca menu menu.

Cth :

Di dalam Front/Menu :

ublic function index($id = null)

    {

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

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

        $sql = "SELECT \* FROM tblkategori ORDER BY kategori ASC";

        $result = $db -> query($sql);

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

        $model = new Menu\_M();

        $data = [

            'judul' => 'DAFTAR MENU',

            'menu' => $row,

            'daftar' => $model -> paginate(3, 'info'),

            'pager' => $model -> pager

        ];

*return* view('front/menu', $data);

    }

Artinya, di dalam function index, parameter nya berisi $id = null, yaitu adalah $id bernilai kosong, jika tidak diisi maka tidak error. Lalu $db digunakan untuk menyambungkan ke dalam database, $pager digunakan untuk inialisasi / menyalakaan paging. $sql berisi perintah yaitu memilih data / SELECT data dari tblkategori diurutkan dari katgori dengan urutan ASC, yaitu diurutkan dari yang terkecil ke yang terbesar, $result berisi $db yang menjalankan perintah query dari $sql, $row berisi $result yang menjalankan perintah untuk mendapatkan hasil / getResult dan dengan hasil berbentuk array. Pemanggilan dari tblkategori ini digunakan untuk mengirimkan data ke dalam template yang membaca tblkategori, data harus dikirimkan kembali ke dalam setiap view yang dipasang rendersection, apabila data tidak dikirimkan kembali, walaupun view sudah dipasang rendersection dan di homepage sudah dikirimkan data untuk membaca tblkategori, tetap saja, karena data yang dibaca adalah data dari database, dimana harus dikirimkan lagi ke dalam view template agar data dari tblkategori dapat ditampilkan. Terdapat objek $model berisi perintah untuk membaca model Menu\_M, dimana di dalam model Menu\_M terdapat $table tblmenu, artinya data yang disambungkan oleh model ke dalam database adalah tblmenu.m. $data berisi data – data yang akan dikirimkan ke dalam view, $data berisi, key judul dengan value DAFTAR MENU, key menu dengan value $row, key daftar dengan value $model yang menjalankan perintah paginate dan data yang ditampilkan adalah 3 dengan menggunakan key info, lalu key pager dengan value $model yang menjalankan perintah pager. Setelah itu, terdapat perintah return yaitu untuk mengeluarkan nilai, lalu membaca front/menu dan mengirimkan $data ke dalam front/menu.

1. Menampilkan menu sesuai dengan kategori yang dipilih.

Di dalam pelanggan/template :

<?php *foreach*($menu as $key) : ?>

    <li class="list-group-item">

<a href="

<?= base\_url('/Front/Menu/option/'.$key['idkategori'])?>

"><?= $key['kategori']?></a>

</li>

<?php *endforeach* ?>

Artinya, terdapat foreach dengan $menu sebagai $key, $menu itu dikirimkan dari data dalam tblkategori, foreach melakukan pengulangan pada <li>, <li> berisi hyperlink dengan hasil yang mengambil nilai dari $key[‘kategori’], lalu di dalam hyperlink tersebut, apabila link di klik, maka akan diarahkan ke dalam base\_url() lalu disambungkan dengan /Front/Menu/option/$key[‘idkategori’], jadi akan diarahkan ke dalam function option di dalam controller menu pada folder Front, lalu idkategori akan sesuai dengan kategori yang di klik.

Di dalam function option pada Menu :

public function option($id)

    {

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

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

           $sql = "SELECT \* FROM tblkategori ORDER BY kategori ASC";

           $result = $db -> query($sql);

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

*if* (isset($id)) {

            $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;

            }

            $data = [

                'judul' => 'DAFTAR MENU',

                'menu' => $row,

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

                'tampil' => $tampil,

                'total' => $count,

                'pager' => $pager

            ];

*return* view('front/option', $data);

       }

    }

Artinya, di dalam function option, parameter nya adaalh $id, $id mengambil nilai dari $key[‘idkategori’] di dalam view pelanggan/template. Terdapat $db yang digunakan untuk menyambungkan ke dalam database, $pager digunakan inialisasi / menyalakan paging. $sql berisi perintah untuk memiih data / SELECT data dari tblkategori yang diurutkan dari kategori dengan urutan ASC yatu dari urutan kecil ke urutan besar, $result berisi $db yang menjalankan perintah query dari $sql, $row berisi $result yang menjalankan perintah untuk mendapatkan hasil / getResult dan dengan hasil berbentuk array. Terdapat pengujian yaitu apabila $id terisi maka akan membaca objek $model yang berisi Menu\_M, dimana terdapat $table tbluser, artinya table yang disambungkan ke dalam database adalah tbluser, $jumlah berisi $model yang menjalankan perintah seleksi data / where dari kolom idkategori dan id nya sesuai dengan $id, lalu menjalankan perintah findAll, yaitu perintah untuk mengambil semua data setelah seleksi, $count berisi perintah untuk menghitung banyak nya data yang adal di dalam $jumlah, $tampil berisi 3, $mulai berisi 0. Setelah itu terdapat pengujian, yaitu apabial $\_GET[‘page’] di klik, maka $mengambil nilai dari $\_GET[‘page’], $mulai menjalankan operasi aritmatika, yaitu $mulai berisi hasil dari$tampil dikalikan $page lalu dikurangi $tampil. Setelah itu, terdapat $data yang berisi data data yang dikirimkan ke dalam view, $data berisi, key judul dengan value DAFTAR MENU, key menu dengan value $row, key hasil dengan value yaitu $model yang menjalankan perintah seleksi / where idkategori dan nilai dari id nya sesuai dengan $id, lalu menjalankan perintah findAll dengan $tampil untuk berapa data yang ditampilkan dan $mulai untuk data yang ditampilkan dimulai dari data nomor berapa, key tampil dengan value $tampil, key total dengan value $count dan key pager dengan value $pager. Setelah itu, terdapat perintah return yaitu untuk mengeluarkan nilai dan menjalankan perintah untuk membaca front/option dan mengirimkan $data ke dalam front/menu.

Di dalam view front/option :

<div class="col-4 mt-4">

    <h4> <?= $judul?></h4>

</div>

<?php *if*(!empty($hasil)) { ?>

<?php *foreach*($hasil as $key):  ?>

    <div class="card" style="width: 14rem; float:left; margin:10px;">

        <img style="height:160px;" src="<?= base\_url('upload/'.$key['gambar'].'')?>" class="card-img-top" alt="...">

        <div class="card-body">

            <h5 class="card-title"><?php echo $key['menu'] ?></h5>

            <p class="card-text"><?php echo $key['harga'] ?></p>

            <a class="btn btn-primary" href="<?= base\_url('Front/Menu/beli/'.$key['idmenu'])?>" role="button">BELI</a>

        </div>

    </div>

<?php *endforeach* ?>

<?php } ?>

Artinya, terdapat tulisan dengan ukuran h4 yang menampilkan $judul, lalu terdapat pengujian menggunakan if, yaitu apabila $hasil tidak kosong, maka akan menjalankan pengulangan menggunakan foreach dengan $hasil sebagai $key yang berisi, img dengan mengambil gambar dari base\_url() lalu disambungkan dengan upload dengan gambar sesuai pada $key[‘gambar’], tulisan berukuran h5 mengambil tulisan dari $key[‘menu’], lalu mengambil harga sesuai dengan $key[‘harga’], dan terdapat hyperlink dengan button BELI yang apabila link di klik, maka akan diarahkan ke dala, base\_url() lalu disambungkan dengan Front/Menu/beli, jadi apabila link di klik maka akan diarahkan ke dalam halaman dari folder Front controller Menu dan function beli.

1. Menampilkan semua menu.

Di dalam view front/menu :

<div class="col">

            <?php *if*(!empty($daftar)) { ?>

                    <?php *foreach*($daftar as $key):  ?>

                        <div class="card" style="width: 14rem; float:left; margin:10px;">

                            <img style="height:160px;" src="<?= base\_url('upload/'.$key['gambar'].'')?>" class="card-img-top" alt="...">

                            <div class="card-body">

                                <h5 class="card-title"><?php echo $key['menu'] ?></h5>

                                <p class="card-text"><?php echo $key['harga'] ?></p>

                                <a class="btn btn-primary" href="<?= base\_url('Front/Menu/beli')?>" role="button">BELI</a>

                            </div>

                        </div>

                    <?php *endforeach* ?>

                    <?php } ?>

            </div>

<div class="col-6 px-1"><?= $this->renderSection('content') ?></div>

Artinya, di dalam View front/menu terdapat pengujian yaitu apabila $daftar terisi / tidak kosong, maka akan melakukan foreach / pengulangan dengan $daftar sebagai $key, lalu terdapat div yang berisi img yang mengambil gambar dari folder upload sesuai dengan $key[‘gambar’], dengan title yang mengambil text sesuai dengan $key[‘menu’], dan harga yang mengambil harga sesuai dengan $key[‘harga’], setelah itu, terdapat hyperlink dengan button BELI yang apabila link di klik, maka akan diarahkan ke dala, base\_url() lalu disambungkan dengan Front/Menu/beli, jadi apabila link di klik maka akan diarahkan ke dalam halaman dari folder Front controller Menu dan function beli.

1. Di dalam function beli

public function beli($id = null)

    {

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

        $sql = "SELECT \* FROM tblkategori ORDER BY kategori ASC";

        $result = $db -> query($sql);

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

        $model\_kategori = new Kategori\_M();

        $model\_menu = new Menu\_M();

        $total = 0;

*if* (isset($id)) {

            $this -> isi($id);

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

        } *else* {

*foreach* (session() -> get() as $key => $value) {

*if* ($key<>'\_\_ci\_last\_regenerate' && $key<>'\_ci\_previous\_url' && $key<>'pelanggan' && $key<>'email' && $key<>'idpelanggan') {

                   $id = substr($key, 1);

                   $menu[] = $model\_menu -> where('idmenu', $id)-> findAll();

                   $jumlah[] = $value;

                }

            }

        }

*if* (!isset($menu)) {

            $menu = [];

            $jumlah = [];

        }

        $data = [

            'kategori' => $model\_kategori -> findAll(),

            'menu' => $row,

            'menuu' => $menu,

            'jumlah' => $jumlah,

            'total' => $total

        ];

*return* view('front/cart', $data);

    }

Artinya, di dalam function beli terdapat parameter yang berisi $id = null, artinya $id bernilai kosong. $db berfungsi untuk menyambungkan function ke dalam database, lalu $sql berisi perintah untuk memilih data / SELECT data dari tblkategori yang diurutkan dari kategori dengan urutan ASC, yaitu dari kecil ke besar, $result berisi menjalankan $db dengan perintah query pada $sql, lalu $row berisi untuk membuat hasil dari data yang diambil menjadi array dengan perintah getResult. Lalu terdapat objek $model\_kategori mengambil data dari Kategori\_M dam objek $model\_menu yang mengambil data dari Menu\_M. $total berisi nilai 0.

Di dalam function beli terdapat pengujian menggunakan if, yaitu apabila $id terisi maka akan menjalankan function isi yang membaca $id, lalu setelah membaca $id maka tampilan akan dikembalikan ke halaman base\_url() dan disambungkan dengan Front/Menu/beli. Tetapi jika $id belum terisi, terdapat foreach dengan session() -> get as $key dan terdapat $value. Di dalam foreach juga terdapat if, dengan pengujian yaitu, apabila $key tidak sama dengan \_\_ci\_last\_regenerate & $key tidak sama dengan \_\_ci\_previous\_url & $key tidak sama dengan pelanggan & key tidak sama dengan email & $key tidak sama dengan idpelanggan, maka akan menjalankan $id yang di dalam nya terdapat perintah substr($key, 1), yaitu perintah untuk memotong string pada $key dan menampilkan tulisan pada index 1, berarti substr hanya memotong index ke 0. $menu[] $model\_menu yang menjalankan perintah untuk mengambil data menggunakan fungsi where sesuai dengan idmenu dan id, setelah menemukan data yang dicari yaitu sesuai dengan idmenu dan $id maka menjalankan perintah untuk mengambil data dengan fungsi findAll. $jumlah[] mengambil nilai dari $value.

Pada pengujian if yang lain, yaitu apabila $menu tidak terisi, maka $menu berisi kosong, dan $jumlah juga berisi kosong.

$data berisi data – data yang akan dikirimkan ke dalam view, $data berisi key kategori dengan value mengambil nilai dari $model\_kategori yang menjalankan perintah findAll untuk mengambil semua data. Key menu dengan value $row, key menu dengan value $menu, key jumlah dengan value $jumlah dan key total dengan value $total. Setelah itu akan menjalankan view front/cart dan mengirimkan $data.

1. Di dalam function isi

public function isi($id)

    {

*if* (session() -> has('\_'.$id)) {

            session() -> set('\_'.$id, session() -> get('\_'.$id)+1);

        }*else* {

            session() -> set('\_'.$id, 1);

        }

    }

1. Di dalam function tambah dan function kurang

public function tambah($id = null)

    {

        session()->set('\_' . $id, session()->get('\_' . $id) + 1);

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

    }

public function kurang($id = null)

    {

        $idmenu = '\_' . $id;

        session()->set($idmenu, session()->get($idmenu) - 1);

*if* (session()->get($idmenu) == 0) {

            session()->remove($idmenu);

        }

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

    }

Artinya, pada tombol tambah dan kurang untuk pesanan pada view cart adalah kita menambahkan link url untuk tambah seperti <?= base\_url('/front/beli/tambah/'.$v['idmenu']) ?>, dan untuk kurang adalah dengan <?= base\_url('/front/beli/kurang/'.$v['idmenu']) ?>. pada controller beli kita buat function untuk tambah dengan menggunakan parameter $id = null lalu kita isi untuk session id seperti session()->set('\_' . $id, session()->get('\_' . $id) + 1);

return redirect()->to(base\_url('/front/beli')). Untuk function kurang sama seperti isi dari function tambah hanya saja kita kirangi satu seperti session()->set('\_' . $id, session()->get('\_' . $id) - 1);

return redirect()->to(base\_url('/front/beli'));. Karena pada function kuarng hasilnya bisa kurang dari 0 maka kita tambahkan if untuk menu yang di kurangi sampai ke angka 0 maka menu akan otomatis hilang seperti membuat variabel baru misal $idmenu terlebih dahulu untuk memasukkan id yang tersimpan kemudian tambahkan if (session()->get($idmenu) == 0) {

session()->remove($idmenu); }.

1. Di dalam function hapus

public function hapus($id = null)

    {

        $idmenu = '\_'.$id;

        session()->remove($idmenu);

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

    }

Artinya, Untuk tombol hapus menu kita tambahkan base url untuk view hapus dengan <?= base\_url('/front/beli/hapus/'.$v['idmenu']) ?>. Kemudian, pada function hapus kita isi dengan parameter $id = null lalu untuk isi function hapus adalah $idmenu = '\_'.$id;

session()->remove($idmenu);

return redirect()->to(base\_url('/front/beli'));.

1. Di dalam function idOrder

public function idorder()

    {

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

        $sql = "SELECT idorder FROM tblorder ORDER BY idorder DESC";

        $result = $db->query($sql);

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

        $count = count($detail);

*if* ($count == 0) {

            $id = 1;

        } *else* {

            $id = $count + 1;

        }

*return* $id;

    }

Artinya, pada function idorder kita gunakan untuk memilih data yang sudah tersimpan lalu menambahkannya dengan menggunakan if. Untuk menjalankannya sambungkan terlebih dahulu dengan database dengan menggunakan perintah bawaan dari CI yaitu $db = \Config\Database::connect(). Kemudian, menggunakan $sql untuk memilih data idorder dari tabel order lalu mengurutkan data id order dari yang terbesar ke yang terkecil. Kemudian, menyambungkan$ sql tersebut dengan menggunakan variabel baru misal $result yang diisi dengan $db->query($sql). Lalu, menggunakan $detail untuk menampilkan hasil dari $result tadi lalu memasukkan datanya menggunakan array. lalu, menggunakan $count untuk menghitung jumlah dari $detail. Kemudian, jika $count berisi kosong atau 0 maka $id berisikan 1, namun jika $id sudah terisi maka $count ditambah 1. Kemudian, return $id.

1. Di dalam function insertOrder

public function insertOrder($idorder, $idpelanggan, $tgl, $total)

    {

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

        $sql = "INSERT INTO tblorder VALUES ($idorder, $idpelanggan, '$tgl', $total, 0, 0, 0)";

        $db->query($sql);

    }

Artinya, pada function insert order yang digunakan untuk memasukkan data yang tersimpan pada tabel order yang berada di database dengan menggunakan parameter $idorder, $idpelanggan, $tgl, $total. Di dalam function insert order kita sambungkan terlebih dahulu dengan database. Kemudian, kita menggunakan $sql untuk memasukkan data yang di terima ke dalam tbl order kemudian kita jalankan $sql tersebut dengan menggunakan perintah $db->query($sql).

1. Di dalam function insertOrderDetail

public function insertOrderDetail($idorder)

    {

        $model\_menu = new Menu\_M();

        $model\_orderdetail = new OrderDetail2\_M();

*foreach* (session()->get() as $key => $value) {

*if* (

                $key<>'\_\_ci\_last\_regenerate' && $key<>'\_ci\_previous\_url' && $key<>'pelanggan'

                && $key<>'email' && $key<>'idpelanggan'

            ) {

                $id = substr($key, 1);

                $produk = $model\_menu->where('idmenu', $id)->findAll();

*foreach* ($produk as $key) {

                    $idmenu = $key['idmenu'];

                    $harga = $key['harga'];

                    $data = [

                        'idorder' => $idorder,

                        'idmenu' => $idmenu,

                        'jumlah' => $value,

                        'hargajual' => $harga

                    ];

                    $model\_orderdetail->insert($data);

                }

            }

        }

    }

Artinya, Untuk function insert order detail dengan menggunakan parameter $idorder. Kemudian menggunakan model menu dan order detail sehingga memerlukan objek untuk mengunakan modelnya dan menyambungkannya untuk disambungkan datanya.

lalu, melakukan foreach untuk mendapatkan session yang diubah menjadi value untuk diambil isinya. Kemudian menggunakan if yang menggunakan kondisi $key<>'\_\_ci\_last\_regenerate' && $key<>'\_ci\_previous\_url' && $key<>'pelanggan'

&& $key<>'email' && $key<>'idpelanggan'. Lalu, mengisikan ifnya dengan $id yang di substr $keynya, pada baris ke 1. Lalu, menggunakan variabel $produk untuk mengambil isi model dari model menu yang dimana idmenu, $id kemudian findAll untuk mencari semua id datanya. Kemudian, melakukan foreach untuk menjadikan $produk menjadi $key kemudian memforeach idmeu dan harganya. Kemudian untuk menyimpan datanya kita membuat array $data yang beridi dengan idorder dari $idorder, idmenu dari $idmenu, jumlah dari $value, dan harga jual dari $harga. Agar data yang disimpan oleh array akan terkirim dan dimasukkan ke model order detail maka menggunakan perintah $model\_orderdetail->insert($data).

1. Di dalam function kosonganSession

public function kosonganSession()

    {

*foreach* (session()->get() as $key => $value) {

*if* (

                $key<>'\_\_ci\_last\_regenerate' && $key<>'\_ci\_previous\_url' && $key<>'pelanggan'

                && $key<>'email' && $key<>'idpelanggan'

            ) {

                $id = substr($key, 1);

                session()->remove('\_'.$id);

            }

        }

    }

Artinya, Pada function untuk mengosongkan session ketika kita sudah selesai checkout maka kita isi dengan foreach session sebagati $key yang kemudian diambil valuenya. Lalu, melakukan if dengan kondisi $key<>'\_\_ci\_last\_regenerate' && $key<>'\_ci\_previous\_url' && $key<>'pelanggan'

&& $key<>'email' && $key<>'idpelanggan'. Kemudian mensubstr $key baris pertama kita masukkan de dalam $id. Lalu, $id tersebut akan terpotong dan akan terhapus dengan menggunakan session()->remove('\_'.$id).

1. Di dalam function history

 public function history()

    {

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

        $sql = "SELECT \* FROM tblkategori ORDER BY kategori ASC";

        $result = $db -> query($sql);

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

        $model\_kategori = new Kategori\_M();

        $vorder = $db->table('vorder');

        $email = session()->get('email');

        $result = $vorder->getWhere(['email' => $email]);

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

        $count = count($halaman);

        $result = $vorder->getWhere(['email' => $email]);

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

        $data = [

            'menu' => $row,

            'vorder' => $vo,

            'total' => $count

        ];

*return* view("front/history", $data);

    }

Artinya, Untuk pada bagian function hinstori kita gunakan model kategori dengan membuatkan objek terlebih dahulu. Kemudian, kita smabungkan dengan databasenya menggunakan $db = \Config\Database::connect(). Kemudian kita mengambil tabel dari vorder langsung dengan menggunakan $db->table('vorder') dan kita masukkan ke dalam variable $vorder untuk menyimpan datanya. Lalu kita menggunakan session untuk mengambil email pelanggan. Lalu, kita membuat $result untuk mengambil data yang ada di dalam $vorder lalu mengambil email dengan $email. Lalu, membuat variabel halaman untuk mengirim data dari $result menggunakan array. Lalu, menggunakan $count untuk menghitung jumlah isi dari $halaman menggunakan function bawaan yaitu count().

Kemudian, kita gunakan kembali $result untuk mengambil data dari $vorder. Lalu kita gunakan $vo untuk mengambil data dari $result dan mengirim datanya menggunakan array. Array yang digunakan untuk mengirim datanya adalah dengan menggunakan $data = [

'kategori' => $model\_kategori->findAll(),

'vorder' => $vo,

'total' => $count

]. Lalu, kita kirim datanya menggunakan return view ke view histori seperti return view("front/histori", $data).

1. Di dalam view history

Dalam view history kita gunakan tabel untuk menampung data yang akan ditampilkan. Data yang ditampilkan adalah tanggal order, total pembayaran dan detail pembelian. Untuk bagian isinya kita gunakan foreach untuk mengambil value dari $vorder yang datanya kita kirim melalui array $data. Untuk bagian tanggal dan total kita gunakan <?php echo $value['tglorder'] ?> untuk tanggal order dan <?php echo $value['total'] ?> untuk total pembayaran. Pada bagian detail kita arahkan ke function detail yang kita ambil idorder dari value $vorder tadi, seperti <?= base\_url('/front/homepage/detail/'.$value['idorder'].'') ?>.

1. Di dalam function detail

public function detail($id)

    {

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

        $sql = "SELECT \* FROM tblkategori ORDER BY kategori ASC";

        $result = $db -> query($sql);

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

        $model\_orderdetail = new OrderDetail\_M();

        $jumlah = $model\_orderdetail->where('idorder', $id)->findAll();

        $count = count($jumlah);

        $detail = $model\_orderdetail->where('idorder', $id)->findAll();

        $data = [

            'detail' => $detail,

            'menu' => $row

        ];

*return* view('front/detail', $data);

    }

Artinya, pada function detail kita menggunakan parameter $id. Untuk menyambungkan database kita gunakan $db = \Config\Database::connect(). Karena detail pemesanan membutuhkan model dari model kateori dan order detail maka, kita membuat objek baru seperti $model\_kategori = new Kategori\_M() dan $model\_orderdetail = new OrderDetail\_M(). Lalu, kita membuat variabel untuk mengambil data dari model order detail dimana idorder sebagai $id kemudian findAll() data yang berasal dari id order. Kemudian, membuat $count untuk menghitung jumlah dari isi $jumlah. Lalu, membuat variabel $detail untuk mengambil id order di dalam model order detail lalu findAll(). Kemudian, kita membuat variabel untuk menyimpan data menggunkan array yaitu $data yang kita isi dengan detail dari $detail dan kategori dari model kategori kemudian findAll(). Lalu kita kirimkan data array tersebut dengan melakukan return view ke view untuk detail pemesanan seperti return view(front/detail', $data).

1. Di dalam view detail

Artinya, pada function detail kita menggunakan parameter $id. Untuk menyambungkan database kita gunakan $db = \Config\Database::connect(). Karena detail pemesanan membutuhkan model dari model kateori dan order detail maka, kita membuat objek baru seperti $model\_kategori = new Kategori\_M() dan $model\_orderdetail = new OrderDetail\_M(). Lalu, kita membuat variabel untuk mengambil data dari model order detail dimana idorder sebagai $id kemudian findAll() data yang berasal dari id order. Kemudian, membuat $count untuk menghitung jumlah dari isi $jumlah. Lalu, membuat variabel $detail untuk mengambil id order di dalam model order detail lalu findAll(). Kemudian, kita membuat variabel untuk menyimpan data menggunkan array yaitu $data yang kita isi dengan detail dari $detail dan kategori dari model kategori kemudian findAll(). Lalu kita kirimkan data array tersebut dengan melakukan return view ke view untuk detail pemesanan seperti return view('Home/detail', $data).

**Saya belum mengerti**