|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Front-End Restoran CI4 minggu ke-21 sebagai nilai UAS | 95 |

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

1. Untuk membuat Front-end pada aplikasi Restoran CI-4, yang pertama adalah menyiapkan controller, lalu view dan models. Bagian yang pertama adalah mengatur untuk homepage nya terlebih dahulu, karena pada url browsernya yang diketikkan adalah nama class di dalam controller, maka dibuatkan folder front di dalam controller dan buat file Homepage.php yang berisi class homepage. Di dalam file ini dibagian atasnya class b=perlu ditambahkan

namespace App\Controllers\Front;

use App\Controllers\BaseController;

use App\Models\Kategori\_M;

use App\Models\Menu\_M;

yang berarti controller ini ada di dalam folder front dan menggunakan basecontroller agar controller berjalan. Setelah itu menyambungkan dengan model kategori dan menu menggunakan use App\Models\...

Di dalam class homepage dibuatkan function yang pertama yaitu index dimana index ini berisi

public function index()

    {

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

        $modelm = new Menu\_M();

        $modelk = new Kategori\_M();

        $data = [

            'judul' => 'DATA MENU',

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

            'pager' => $modelm->pager,

            'kategori' => $modelk->findAll()

        ];

        return view('front/menu', $data);

    }

Ada $pager sebagai paging untuk menu nanti, $modelm yang isinya untuk menggunakan model menu, $modelk yang isinya untuk menggunakan model kategori, lalu di dalam $data ada ‘judul’ untuk judul, ‘menu’ yang mengambil data dari model m yaitu tblmenu dengan membaginya menjadi 3 data setiap page nya, lalu ada ‘pager’ untuk menjalankan pager dari model menu, dan terakhir ‘kategori’ untuk mengambil semua data dari $modelk atau tblkategorinya, di return view ada $data yang dapat digunakan ke dalam view folder front file menu.

2. Langkah kedua membuat file front.php di dalam view yang berfungsi sebagai template yang selalu muncul dihalaman apapun pada restoran ci-4. Isi file front.php ini yaitu berupa html yang dibagi menjadi bagian head dan body, di head nya ada ada title yang diisi dengan nama restoran lalu ada link yang menyambungkan dengan bootstrap offline. Di bagian body ada div container dan didalam container dibagi menjadi 3 div row. Row pertama berisi 1 col untuk navbar atau kepala yang ada dibagian atas webnya. Navbar ini ada judul restoran yang jika ditekan akan ke halaman awal lalu ada div class yang berisi if (!empty(session()->get('emailp'))), di dalam if nya untuk mendeklarasikan nav pelanggan, email, button cart, button history, dan button logout apabila sudah login yang ada emailp nya, dan else nya berisi button daftar dan login. Setelah itu div row kedua dibagi menjadi 2 col, yang pertama untuk membuat navbar dengan model list-group-item seperti ini

<div class="col-md-3">

                <h3>Kategori</h3>

                <hr>

                <nav>

                    <?php

                    foreach ($kategori as $key) : ?>

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

                            <a href="<?= base\_url('front/kategori/menu/' . $key['idkategori']) ?>">

                                <?php echo $key['kategori'] ?>

                            </a>

                        </li>

                    <?php endforeach ?>

                </nav>

            </div>

Yang berarti ada judul nya yaitu Kategori lalu nav kebawah yang isinya diambil dari $data yaitu $kategori nya dan data ini dibuat menjadi hyperlink yang nanti akan menyambungkan ke controller kategori bagian function menu lalu diambil idkategorinya sesuai dengan data yang dipilih. Setelah itu col kedua berisi

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

Yang bisa digunakan sebagai template untuk file view lainnya pada bagian col yang ini. Terakhir bagian row ketiga berisi 1 col untuk footer berupa text 2020-copyright@suhailahnfsella\_ setelah itu tutup html.

3. karena pada langkah pertama sudah dibuatkan controller homepage yang mengarahkan pada view folder front dan file menu. Pertama ditambahkan

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

<?= $this->section('content') ?>

Ditelakkan dibagian atas agar file ini berada di bagian col dari renderSection yang bernama ‘content’ di file front.php tadi. Setelah itu baru nemabahkan div dan diisi dengan data menu dengan menggunakan key yang ada di $data pada controllernya. Seperti ini

<?php foreach ($menu as $key) : ?>

            <div class="card" style="width: 15rem; float: left; margin: 12px;">

                <img src="<?= base\_url('upload/' . $key['gambar'] . '') ?>" class="card-img-top" style="height: 150px;" alt="">

                <div class="card-body">

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

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

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

                        Beli

                    </a>

                </div>

            </div>

        <?php endforeach ?>

Yang berarti dia melooping $menu atau key dari $data dan mengisinya ke dalam card sehingga membentuk beberapa card yang jumlahnya sejumlah dengan jumlah data menu nya. adanya $key[‘menu’] dan lainnya berarti mengambil data dari kolom bernama menu pada tblmenu. Setelah itu ditambahkan div yang berisi <?= $pager->links('page', 'bootstrap') ?> untuk memberi page pada banyaknya card menjadi beberapa page. Terakhir menambahkan

<?= $this->endSection() ?> untuk mengakhiri section.

4. Setelah tampilan homepage sudah selesai, selanjutnya mengatur bagian login. Pertama adalah mengarahkan button login ke <?= base\_url('front/login') ?> atau ke dalam folder front dan file login pada controllernya, yang pertama adalah mengisi class login ini dengan function index

public function index()

    {

        $modelk = new Kategori\_M();

        $data = [];

        $data = [

            'kategori' => $modelk->findAll()

        ];

        if ($this->request->getMethod() == 'post') {

            $email = $this->request->getPost('emailp');

            $password = $this->request->getPost('password');

            $model = new Pelanggan\_M();

            $pelanggan = $model->where(['email' => $email])->first();

            if (empty($pelanggan)) {

                $data = [

                    'info' => "Email Salah",

                    'kategori' => $modelk->findAll()

                ];

            } else {

                if (password\_verify($password, $pelanggan['password'])) {

                    $this->setSession($pelanggan);

                    return redirect()->to(base\_url("Front/Homepage"));

                } else {

                    $data = [

                        'info' => "Password Salah",

                        'kategori' => $modelk->findAll()

                    ];

                }

            }

        } else {

            # code...

        }

return view('template/loginfront', $data);

Yang berarti $modelk untuk mengambil data dari model Kategori\_M yang sebelumnya sudah ditambahkan use App\Models\Kategori\_M; di awal file. Lalu ada $data[] untuk menyimpan semua data yang diambil dari semua $data. Ada if ($this->request->getMethod() == 'post') yang berarti apabila pengguna mengisi form yang bermethod post yang sudah diatur didalam view nya login ini nanti maka dia akan menjalankan $email sebagai variable untuk menyimpan data dari input bernama emailp dan begitu juga untuk $passwordnya. Setelah itu ada $model = new Pelanggan\_M(); yang mengambil data dari model Pelanggan\_M dan $pelanggan = $model->where(['email' => $email])->first(); nah $pelanggan ini untuk mencari email yang isinya sama dengan $email atau sebagai pencocokan data. Setelah itu ada

            if (empty($pelanggan)) {

                $data = [

                    'info' => "Email Salah",

                    'kategori' => $modelk->findAll()

                ];

            }

Yang berarti apabila $pelanggan atau pencocokan data ini kosong, maka dia akan menampilkan info ini lalu ada else nya

else {

          if (password\_verify($password, $pelanggan['password'])) {

               $this->setSession($pelanggan);

                return redirect()->to(base\_url("Front/Homepage"));

         }

Yang berarti apabila passwordnya sudah terenkripsi dan $password cocok dengan password yang ada di dalam tblpelanggan maka dia akan membuat session untuk login pelanggan sebagai penyimpanan data pelanggan sementara lalu tampilannya mengarah ke tampilan homepage dengan headernya ada nama pelanggan, nama email, button cart, button history, dan button logout karena sudah melakukan login. Terakhir ada

else {

$data = [

'info' => "Password Salah",

'kategori' => $modelk->findAll()

];

}

Yang berfungsi untuk apabila password tidak sama atau tidak cocok dengan data yang ada di tblpelanggan maka yang muncul adalah info password salah. Terakhir ada

return view('template/loginfront', $data); yang berfungsi untuk mengarahkan ke view loginfront dan pada view ini bisa menggunakan $data yang sudah dibuat di dalam controller ini. Di dalam view ada section content seperti view sebelumnya agar menggunakan template dari front.php dan ada form yang bermethod post dengan input untuk email, password, dan submit.

5. Mengatur daftar pelanggan dari button daftar pada tampilan awal homepage nya. yaitu dengan membuat controller sesuai dengan base\_url dari button daftar nya yaitu di bagian folder front dan class daftar. Function pertamanya yaitu

public function index()

    {

        $modelk = new Kategori\_M();

        $data = [

            'kategori' => $modelk->findAll()

        ];

        return view('template/daftar', $data);

    }

Yang nantinya akan mengarahkan ke file view di dalam folder template file daftar. Di file view daftar ini berisi sebuah form yang bermethod post dan nanti diarahkan ke dalam function insert yang ada di dalam controller daftar ini. Functionnya seperti ini

public function insert()

    {

        if (isset($\_POST['simpan'])) {

            if ($\_POST['password'] === $\_POST['konfirmpassword']) {

                $data = [

                    'pelanggan' => $\_POST['pelanggan'],

                    'alamat' => $\_POST['alamat'],

                    'telp' => $\_POST['telp'],

                    'email' => $\_POST['email'],

                    'password' => password\_hash($\_POST['password'], PASSWORD\_DEFAULT),

                    'aktif' => 1

                ];

                $model = new Pelanggan\_M();

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

                    $error = $model->errors();

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

                    return redirect()->to(base\_url("/front/daftar"));

                } else {

                    return redirect()->to(base\_url("/front/homepage"));

                }

            } else {

                $modelk = new Kategori\_M();

                $data = [

                    'kategori' => $modelk->findAll()

                ];

                return view('tes/error', $data);

            }

        }

    }

Yang berarti apabila form itu sudah diisi dan untuk data yang dimasukkan ke password dan confirm password sama maka dia akan menjalankan membuat $data yang menyimpan key dengan value nya dan bagian password juga otomatis terkenkripsi lalu apabila ada kesalahan atau error karena data yang dimasukkan tidak memenuhi syarat maka akan menjalankan info error dan mengulang form daftar dan jika daftar berhasil maka akan ke tampilan awal homepage nya.

6. Mengatur button menu agar login dulu baru bisa melakukan pemesanan. Pertama adalah membuat file filter di dalam folder filter misalkan diberi nama Login.php. file ini bisa dicopy dari file Auth.php dengan mengganti class nya menjadi class Login dan isinya

if (!session()->get('logIn')) {

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

        }

Yang berarti jika belum melakukan login dengan nama logIn milik login pelanggan maka ketika menekan beli akan diarahkan ke login dulu. Setelah itu menambahkan 'Login' => \App\Filters\Login::class ke dalam class Filters public $aliases yang ada di dalam file filters di folder config. Terakhir mengatur routes seperti ini

$routes->group('Front', ['filter' => 'Login'], function ($routes) {

    $routes->add('Beli/beli', 'Front\Beli::beli');

    $routes->add('Beli/beli/(:any)', 'Front\Beli::beli/$1');

    $routes->add('Beli/checkout/(:any)', 'Front\Beli::checkout/$1');

    $routes->add('Histori/histori', 'Front\Histori::histori');

    $routes->add('Histori/detail/(:any)', 'Front\Histori::detail/$1');

});

Yang berarti apabila pada browser mengarahkan url dari salah satu yang ada di atas dan kondisinya belum login pelanggan maka dia tidak akan bisa tersambung ke url itu tetapi bisa tersambung ke url itu ketika sudah melakukan login.

7. Jika tampilan homepage awal sudah jadi selanjutnya adalah mengatur tampilan setelah login, yang pertama yaitu cart yang bisa dilihat melalui button cart. Cart ini sama seperti keranjang belanja dimana jika menekan beli dari card menu nya maka akan menyimpan data yang akan dikirimkan dan ditampilkan ke dalam button cart. Pertama adalah mengatur button beli jika ditekan akan menuj ke controller beli function beli. Codenya

public function beli($id = null)

    {

        $modelk = new Kategori\_M();

        $modelm = new Menu\_M();

        $total = 0;

        $idk = $modelm->where('idmenu', $id)->findColumn('idkategori');

        if (isset($id)) {

            $this->isi($id);

            return redirect()->to(base\_url('/front/kategori/menu/' . $idk[0]));

        } else {

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

                if (

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

                    && $key <> 'emailp' && $key <> 'idpelanggan' && $key <> 'loggedIn' && $key <> 'password' && $key <> 'logIn'

                ) {

                    $id = substr($key, 1);

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

                    $jumlah[] = $value;

                }

            }

            if (!isset($menu)) {

                $menu = [];

                $jumlah = [];

            }

            $data = [

                'kategori' => $modelk->findAll(),

                'menu' => $menu,

                'jumlah' => $jumlah,

                'total' => $total

            ];

        }

        return view('front/beli', $data);

    }

Yang berarti dari data yang dipilih untuk dibeli akan diambil id nya, ada $modelk untuk mengambil data dari model Kategori\_M ada $modelm untuk mengambil data dari model Menu\_M lalu ada $total yang diisi dengan 0, da nada $idk yang berfungsi untuk mencari idkategori dari idmenu yang dipilih. Jika $id sudah didapatkan yang berarti jika button sudah ditekan maka $id ini akan diarahkan ke function bernama isi seperti ini

 public function isi($id)

    {

        if (session()->has('\_' . $id)) {

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

        } else {

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

        }

        var\_dump($\_SESSION);

    }

Yang berfungsi untuk mengambil id menu dan mengambil data dari menu yang dipilih untuk disimpan di sessionnya dan nanti ditampilkan isi session yang dipilih di dalam tabel cart nya. Lalu ketika menekan button kan diarahkan ke halaman yang satu kategori dengan menu yang dipilih. Setelah itu hasil dari code nya disimpan di dalam $data yang nantinya digunakan di dalam view beli nya. Karena tabel ini ada kurang, tambah, dan hapus untuk barang yang dipilih maka perlu function untuk ketiga nya.

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/Beli/beli'));

    }

    public function tambah($id = null)

    {

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

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

    }

    public function hapus($id = null)

    {

        $idmenu = '\_' . $id;

        session()->remove($idmenu);

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

    }

Setelah itu ada button checkout yang diarahkan ke function checkout nya

public function checkout($total = null, $id = null)

    {

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

        $modelm = new Menu\_M();

        $modelk = new Kategori\_M();

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

        $data = [

            'kategori' => $modelk->findAll()

        ];

        if (isset($total)) {

            $idorder = $this->idorder();

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

            date\_default\_timezone\_set('Asia/Jakarta');

            $tgl = Date('Y-m-d');

            $sql = "SELECT \* FROM tblorder WHERE idorder = $idorder";

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

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

            $count = count($detail);

            if ($count == 0) {

                $this->insertOrder($idorder, $idpelanggan, $tgl, $total);

                $this->insertDetail($idorder);

            } else {

                $this->insertDetail($idorder);

            }

            $this->kosongkanSession();

            return view('front/pesan', $data);

        } else {

            $id = $this->idorder() - 1;

            $modelk = new Kategori\_M();

            $modelod = new OrderDetail\_M();

            $data = [

                'kategori' => $modelk->findAll(),

                'menu' => $menu,

                'total' => $total,

                'order' => $modelod->where('idorder', $id)->findAll(),

                'id' => $id

            ];

            return view('front/beli', $data);

        }

    }

Yang berarti menu yang dipilih di cart akan diambil totalnya untuk bisa dicheckout, dengan function-function lainnya checkout ini nanti akan dikirim ke orderdetail sesuai dengan data yang ada di cart nya, jadi admin atau kasir bisa mengecek barang apa saja yang dipilih dari jumlah total dll nya dan bisa mengatur pembayarannya, jika pelanggan sudah menekan checkout akan keluar sebuah teks terimakasih sudah berbelanja dan lakukan pembayaran ke kasir karena diarahkan ke view pesan.php yang berisi teks itu tadi.

8. Setelah melakukan checkout, hasil checkout atau history nya bisa dilihat melalui button history, jadi button ini diarahkan ke controller history yang bagian function history nya.

public function histori()

    {

        $modelk = new Kategori\_M();

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

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

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

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

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

        $count = count($halaman);

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

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

        $data = [

            'kategori' => $modelk->findAll(),

            'vorder' => $vo,

            'total' => $count

        ];

        return view('front/histori', $data);

    }

Yang berarti dia mengambil data dari model Kategori\_M untuk template nya, $vorder untuk memilih vorder dari database dan memilih data dari email pelanggannya, jadi diambil data yang hanya milik pelanggan ini lalu dijadikan array dan semuanya dimasukkan ke dalam $data agar bisa digunakan ketika di view. Untuk view nya berisi tabel yang isinya diambil dari $data ini. Karena di dalam table ada hyperlink untuk detail, maka di dalam controller histori ditambahkan function detail untuk digunakan oleh detail. Isinya

public function detail($id)

    {

        $modelk = new Kategori\_M();

        $modelod = new OrderDetail\_M();

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

        $jumlah = $modelod->where('idorder', $id)->findAll();

        $count = count($jumlah);

        $detail = $modelod->where('idorder', $id)->findAll();

        $data = [

            'kategori' => $modelk->findAll(),

            'detail' => $detail

        ];

        return view('front/detail', $data);

    }

Cara kerjanya sama dengan function histori, hanya saja detail ini mengambil data dari order detailnya karena yang dibutuhkan untuk ditampilkan adalah data yang lebih rinci nya. dan untuk view nya isinya sama dengan view histori yaitu hanya tabel.

9. Terakhir bagian button logoutnya, button ini diarahkan ke controller login bagian function logout nya. isinya

public function logout()

    {

        session()->destroy();

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

    }

Yang berarti dia akan mendestroy atau menghapus atau mereset kosong semua session yang disimpan mulai dari loginnya. Lalu diarahkan ke tampilan awal homepage sebelum login dimana button yang ada diatas hanya daftar dan login karena sudah logout.

**Saya Belum Mengerti**

1.