|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| ***Project Restoran CI 4 (Front End)*** | 90 |

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

**#BAGIAN TEMPLATE**

1. Saya membuat template home terlebih dahulu untuk halaman utama yang akan saya isi nantinya dengan berbagai view seperti Login, logout, daftar pelanggan, menu, detail, dan histori. Untuk membuatnya saya isi menggunakan html yang nantinya kita tambahkan dengan php. Saya sambungkan terlebih dahulu dengan bootstrapnya untuk merapikan web. Lalu, Kita masukkan tittle untuk webnya. Lalu, kita membuat container untuk membungkus item yang akan di masukkan. Kemudian, membuat div class row dan col. Untuk headernya, menggunakan navbar yang nantinya di isi dengan nama restoran, email pelanggan dan button untuk memudahkan pelanggan yang menggunakannya.

2. Untuk bagian menampilkan kategori menu, di tambahkan div col dan row kemudian kita tambahkan ul dan li yang nantinya akan kita isi menggunakan controller.

3. Lalu untuk render section untuk template home menambahkan col lalu <?= $this->redersession('content')?>. Karena template home ini yang nantinya digunakan untuk setiap view dari contoller yang diperlukan.

4. Untuk footer pada template kita buat div class footer lalu menambahkan isi footer yang diperlukan.

5. Untuk bagian nav yang ingin diisi linknya, menggunakan base\_url() untuk setiap pemberian link urlnya.

5. Untuk menampilkan email pelanggan kita gunakan <?= session()->get('email') ?>.

6. Agar pelanggan yang belum terdaftar atau belum melakukan login tidak bisa melihat bagian histori atau cart adalah dengan menambahkan if di atas bagian li seperti <?php if (!empty(session()->get('email')) && !empty(session()->get('pelanggan'))) { ?>. Pada bagian ini kita isikan dengan jika session terisi maka akan tampil bagian yang boleh diakses oleh pelanggan <?php } else { ?> jika belum maka akan tampil bagian untuk daftar dan login.

**#BAGIAN DAFTAR**

1. Untuk function daftar kita masukkan ke dalam controller homepage. Lalu, kita akan menampilkan kategori menu yang membutuhkan model kategori untuk mengambil datanya sehingga kita membuat objek terlebih dahilu seperti $model\_kategori = new Kategori\_M(). Kemudian kita masukkan datanya menggunakan array seperti $data = ['kategori' => $model\_kategori->findAll()]. Kemudian, kita kirimkan datanya ke view untuk daftar seperti, return view('Home/daftar', $data).

2. Pada view untuk daftar kita gunakan <?= $this->extend('Template/home')?> dan <?= $this->section('content')?> dan pada akhri untuk isi dari daftar kita gunakan <?= $this->endsection()?>. Untuk isi daftar kita menggunkana form untuk mengisikan data yang diperlukan untuk menjadi pelanggan seperti Nama, Alamat, Nomor telepon, Email, Password, Konfirmasi password dan button untuk menyimpan data. Untuk, tipe form yang digunakan password dan konfirmasi adalah password agar ketika pelanggan memasukkan passwordnya maka akan muncul simbol untuk melindungi keamanan password tersebut.

3. Untuk function daftar pada homepage kita melakukan request. Dalam CI terdapat function khusus untuk request dengan cara $request = \Config\Services::request(). Lalu, kita memasukkan datanya satu persatu menggunakan array seperti misal untuk nama pelanggan $data = [ 'pelanggan' => $request->getPost('pelanggan')] begitu juga untuk alamat,telp, dan email. Untuk bagian password dan konfirmasi kita gunakan untuk mengenkripsi password dan konfirmasi untuk menjaga keamanan password caranya dengan menggunakan password\_hash($request->getPost('password'), PASSWORD\_DEFAULT). Kita juga menggunakan 'aktif' => 1 untu menunjukkan bahwa pelanggan tersebut aktif jika tanda aktif pelanggan adalah 0 maka pelanggan tersebut tidak bisa login.

4. Untuk memasukkan data kita memerlukan model pelanggan untuk bisa memasukkan data yang di daftarkan seperti $model\_pelanggan = new Pelanggan\_M().

5. Untuk mengecek apakah data yang didaftarkan pelanggan sudah memenuhi atau tidak kita menggunakan if dan else di dalam if dan else untuk mengecek datanya dua kali seperti if($request->getPost('password'==$request->getPost('konfirmasi'))) { if($model\_pelanggan->insert($data)===false) {

$error = $model\_pelanggan->errors();

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

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

} else {

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

}

} else {

$error = ['password'=>'Password Dan Konfirmasi harus sama!'];

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

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

}.

6. Pada bagian view daftar kita tampilkan perintah jika data pelanggan tidak terpenuhi menggunakan if seperti 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>';

}.

**#BAGIAN LOGIN**

1. Untuk melalukan login aktifkan terlebih dahulu controller login, kemudian di isikan dengan model yang nanti akan digunakan seperti model kategori dan model pelanggan karena kita membutuhkan data dari dalam model tersebut.

2. Pada function index kita akan menggunakan model kategori sehingga kita membuat objek terlebih dahulu seperti $model\_kategori = new Kategori\_M(). Untuk mengambil data kategori menu untuk ditampilkan kita menggunakan array $data = ['kategori' => $model\_kategori->findAll()]. Kemudian untuk mengirim data array tersebut, kita return view('Home/login', $data).

3. Lalu, pada view login kita berikan form method post untuk login pelanggan yang sudah memenuhi syarat dengan mengunakan email dan password untuk melakukan pemesanan dan lain - lain.

4. Lalu, pada controller login pada function index untuk mengisi form kita gunakan if seperti if ($this->request->getMethod() == 'post') {

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

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

echo $email. Karena kita akan mencari data yang ada di dalam model pelanggan maka kita membuat objek baru seperti $model\_pelanggan = new Pelanggan\_M(). Kemudian, untuk mencarinya kita gunakan variabel baru untuk mencarinya menggunakan $pelanggan = $model\_pelanggan->where(['email' => $email, 'aktif' => 1])->first(). Lalu, jika email atau password yang dimasukkan pelanggan salah akan muncul tulisan "Email Salah!" dan "Password salah!" caranya dengan menggunakan if dan else seperti if (empty($pelanggan)) { $data['info'] = "Email Salah!";

} else {

if (password\_verify($password, $pelanggan['password'])) { $this->setSession($pelanggan);

return redirect()->to(base\_url());

} else {

$data['info'] = "Password Salah!";}

}.

**#BAGIAN PRODUK**

1. Untuk mengambil data menu kita membuat function read dalam homepage untuk menampilkan produk. Cara untuk menampilkannya adalah dengan mengambil data dari model menu dan kategori. Lalu, membuat array untuk mengambil datanya seperti $data = [ 'kategori' => $model\_kategori->findAll(), 'menu'=>$model\_menu ]. Kemudian, untuk mengirim datanya view adalah dengan menggunakan return view('Home/produk', $data).

2. Pada view produk kita menggunakan card untuk menampilkan setiap nama menunya dengan cara membuat div row dan col terlebih dahulu kemudian untuk menampilkannya kita memasukkan data yang telah dikirim terlebih dahulu dari function read dengan menggunakan if dan foreach untuk memasukkan data yang dari controller agar ditampilkan datanya seperti

<?php if (!empty($menu)){ ?>

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

<div class="card ml-2" style="width: 13rem; float:left; margin:10px;">

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

<div class="card-body">

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

<p class="card-text">Rp.<?=

number\_format($key['harga']) ?></p>

<a class="btn btn-info" href="<?= base\_url('/front/beli/index/'.$key['idmenu'].'')?>" role="button">Beli</a>

</div>

</div>

<?php endforeach ?>

<?php } ?>

3. Untuk menampilkan paging kita menggunakan pager pada controller read seperti $pager = \Config\Services::pager(). Lalu, pada array $data di bagian menu menu tambahkan paginate(3, 'page'). Lalu 'pager' => $model\_menu->pager.

4. Agar kategori menu dapat tampil per kategorinya kita membuat function baru dengan nama read dengan menggunakan para meter $id = null untuk mengambil id setiap kategori menu adalah dengan menggunakan if seperti if (isset($id)) {

$model\_menu = new Menu\_M();

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

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

$count = count($jumlah). Untuk mendapatkan datanya kita gunakan $menu = $model\_menu->where('idkategori', $id)->findAll();. Untuk mengirim datanya kita menggunakan array $data seperti

$data = [

'menu' => $menu,

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

'pager' => $pager,

'tampil' => $tampil,

'total' => $count,

'id' => $id

];. Untuk mengirim datanya pada view kita gunakan return view ('Home/cari', $data).

5. Pada view cari kita gunakan cara yang sama seperti view produk hanya saja dalam pemangilan kategori menunya menggunakan if seperti

<?php if (!empty($kategori)){ ?>

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

<?php if($id == $key['idkategori']) { ?>

<h2>Daftar Menu <?= $key['kategori']?></h2>

<?php } ?>

<?php endforeach ?>

<?php } ?>.

6. Pada button beli kita arahkan pada controller beli untuk menampilkan setiap menu yang di klik akan tampil di keranjang belanja.

**#BAGIAN CART**

1. Untuk meletakkan produk yang ingin dibeli oleh pembeli kita harus mengambil id dari masing masing menu tersebut dengan membuat function index dengan parameter $id = null pada controller beli . Untuk mengambil id menu kita memerlukan model kategori dan model menu. Caranya kita membuat objek baru untuk menggunakannya seperti $model\_kategori = new Kategori\_M() dan$model\_menu = new Menu\_M(). Kemudian $total = 0 agar setiap total pembelanjaan dimulai dari 0 rupiah. Kemudian, untuk mendapatkan idnya kita menggunakan if isset seperti if (isset($id)) {

$this->isi($id);

return redirect()->to(base\_url('/front/beli')); }. Kemudian else untuk mengforeach seperti

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;

}

}

2. Untuk function isi yang kita gunakan di function index tadi untuk mengambil idnya kita isi dengan parameter $id. Lalu, menggunakan if dan else untuk mengset session id seperti if (session()->has('\_'.$id)) {

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

} else {

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

}.

3. Agar menu yang belum ada isinya di dalam keranjang kosong maka menggunakan if seperti if (!isset($menu)) {$menu = []; $jumlah = []; }. Kemudian kita isi datanya dengan array seperti $data = [

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

'menu' => $menu,

'jumlah' => $jumlah,

'total' => $total

]. Lalu, untuk mengirim datanya kita gunakan return view ke view untuk keranjang seperti return view('Home/cart', $data).

4. Pada view cart kita gunakan tabel untuk menampung datanya. Pada bagian isinya kita lakukan foreach sebanyak 2 kali untuk memperoses datanya pertama untuk mengambil value dari menu kemudian value tersebut diambil lagi valuenya untuk mendapatkan isi dari menu. Pada bagian isi tabel kita isi dengan nama menu, harga, kemudian tombol untuk tambah dan kurang pemesanan dari menu, total pembelanjaan pembeli, kemudian tombol untuk menghapus pesanan.

Untuk pada bagian total pembayaran kita menggunakan rumus $jumlah[$key] \* $v['harga']. Untuk memasukkan jumlah pembayaran ke dalam $total untuk checkout kita buat rumus dalam $total = $total + ($jumlah[$key] \* $v['harga']). Kemudian kita letakkan pada base url untuk menuju ke function checkout pada tombol checkout seperti <?= base\_url('/front/beli/checkout/'.$total')?>.

5. 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); }.

6. 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'));.

**#BAGIAN CHECKOUT**

1. Untuk bagian checkout kita buat function checkout dalam controller beli dengan menambahkan di dalam parameternya $total = null. Lalu, kita sambungkan ke dalam database dengan menggunakan perintah bawaan dari CI yaitu $db = \Config\Database::connect(). Kemudian, menggunakan if untuk memasukkan datanya yang diperlukan jika $total terisi seperti idorder yang nantinya kita isi menggunakan function idorder, idpelanggan yang mengambil idpelanggan menggunakan session, tanggal order, Lalu menggunakan $sql untuk mengambil id di dalam tabel order. Untuk menyambungkan sqlnya kita gunakan variabel 4result yang diisi dengan $db->query($sql). Lalu, variabel detail untuk memasukkan detail pemesanannya yang diisi dengan $result->getResult('array') kemudian $count untuk menghitung data yang ada di dalam variabel detail tadi.

2. Jika $count belum terisi atau 0 maka akan menjalankan fungtion dari insert order dan function insert order detail . Kemudian, jika $count sudah ada isinya maka akan hanya mengisi function dari insert order detail. dan jika sudah memenuhi syarat maka akan otomatis session akan terhapus menggunakan function dari kosongkanSession. Lalu, menggunakan return redirect ke function checkout agar function tersebut berjalan.

3. Lalu, menggunakan else untuk menambahkan idorder yang sudah terisi dengan menggunakan $id lalu menjalanakan function idorder yang ditambahkan satu. kemudian kita membutuhkan model kategori untuk mengambil isi dari model kategori. Maka, kita memerlukan objek baru untuk menjalankan model kategori. Untuk memasukkan datanya kita menggunakan array $data untuk mengirim datanya ke view info dengan cara return view('Home/info', $data).

4. 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.

5. Untuk 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).

6. 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).

7. 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).

**#BAGIAN HISTORI**

1. 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 mentimpan 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("Home/histori", $data).

2. 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'].'') ?>.

**#BAGIAN DETAIL PEMESANAN**

1. 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).

2. Pada view detail kita isi dengan tanggal order, menu yang dipesan, harga menu, dan jumlah. Untuk bagian isi dari tabel tersebut kita menggunakan foreach untuk measukkan data yang sudah kita kirimkan melalui unction tadi. Data yang di foreach adalah $detail yang kita ambil adalah valuenya untuk menampilkannya ke dalam isi dari masing masing kolom yang dibutuhkan adalah dengan menggunakaan <?php echo $value['tglorder'] ?> untuk tanggal order, <?php echo $value['menu'] ?> untuk nama menu yang dipesan, <?php echo $value['harga'] ?> untuk harga masing masing menu, dan <?php echo $value['jumlah'] ?> untuk jumlah menu yang dipesan.

**#BAGIAN FILTER**

1. Menggunakan filter adalah untuk mengatur bagian bagian mana saja yang tidak boleh di akses selain pelanggan yang telah melakukan login, sehingga ketika ia ingin melakukan pembelian atau transaksi dalam web pasti tidak bisa dan selalu mengarah ke form untuk login pelanggan.

2. Untuk membuat file filter kita membuat file baru pada folder filter bawaan dari ci kemudian kita tambahkan file baru, untuk setiap class harus sama dengan nama file filternya. Lalu, kita isi dengan pengaturan bawaan untuk filter yang telah disediakan oleh ci yaitu function before dan after. Yang kita gunakan hanya function before yaitu dengan melakukan if seperti if (!session()->get('logIn')) {

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

}. Ketika jika session login tidk terisi maka akan menuju ke form login.

3. Pada file filter didalam config kita tambahkan filter yang telah kita buat tadi di dalam array public $alianses agar filter yang kita buat itu berjalan dan berfungsi.

4. Pada bagian routes kita tambahkan $routes->group('front', ['filter' => 'Puth(sesuai dengan nama filternya)'], function($routes). Lalu, kita isikan dengan bagian mana saja yang tidak boleh diakses sebelum melakukan login.

**Saya Belum Mengerti**

1.

2.

3.