|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Code Igniter Vid (35 – 51) | 90 |

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

1. Dapat menampilkan view dari order lewat controller order

2. Dapat menambahkan section pada view order

3. Memahami fungsi dari syntax $db = \Config\Database::connect(); yaitu untuk mengkoneksi ke database dan juga dapat menerapkannya dalam program

4. Dapat membuat table untuk menampilkan data dari database order ke dalam view untuk ditampilkan ke page web

5. Memahami & Dapat menampilkan data order dari database pada table dengan menggunakan foreach lalu memanggil data berdasarkan kolom-nya dengan menyimpannya pada variable.

6. Memahami & Dapat menampilkan status order sudah dibayar atau belum dengan memanfaatkan statement if else yang juga akan berguna sebagai button nantinya untuk pembayaran

7. Memahami & dapat membuat paging halaman untuk view table data dari order,yaitu :

* Pada controller order menmabahkan syntax untuk pager dari code igniter
* Menambahkan syntax pager makeLinks pada view order juga template yang sudah dibuat dan digunakan sebelumnya dengan start 1
* Pada controller menambahkan data dari syntax pager makeLinks pada array karena unutk mengirim data (pager,perPage,total)
* Membuat variable untuk total data $ariable = count($row) karena $row digunakan untuk memanggil data dari database
* Juga membuat variable untuk jumlah data yang akan ditampilkan di tiap halaman
* Lalu membuat statement if else untuk menampilkan paging halaman. If isset $\_GET[‘page’] variable untuk halamannya bernama page,Lalu membuat operasi aritmatika untuk menghitung halaman $mulai = ($tampil \* $page) - $tampil; dan menggunakna sql untuk memanggil data dan mengurutkannya sesuai dari status nya sudah lunas/belum bayar dan menggunakan limit $mulai/variable untuk halaman paging, $variable untuk jumlah halaman data yang tampil tiap halaman
* Dalam else isi dengan sql namun start dengan 0 karena paging nya belum terisi data

8. Memahami & dapat membuat form untuk pembayaran order,yaitu :

* Menambahkan hyperlink pada button bayar di kolom status yg mengarahkan ke base\_url untuk folder Admin dan controller order lalu function Find
* Menambahkan juga idorder untuk ditampilkan di dalam server link untuk nantinya di ambil guna untuk inisialisasi proses pembayaran dengan $value[‘idorder’] tinggal mengambil nilainya karena sudah dalam area php
* Lalu membuat function find dalam controller order untuk mengambil data order dan order detail dari database dengan memanfaatkan sql SELECT\*FROM dan code igniter method query & getResult
* Membuat array untuk mengirim data kedalam view untuk pembayaran (judul,order yg berisi data yg diambil oleh getReslut yaitu data dari order juga data dari order detail (menggunakan 2 database)
* Menghubungkan view dengan echo view ke dalam file untuk view pembayaran pada folder order dalam folder Views juga menambahkan $array untuk mengirim datanya agar di tampilkan
* Di dalam file views untuk pembayaran mmebuat container2 dan kolom2 untuk identitas pelanggan yg akan membayar pesanan order (nama pelanggan diambilkan dengan $variable[0][‘pelanggan’] , tanggal dengan $variable[0][‘tglorder’] dan total yang harus dibayar $variable[0][‘total’] semua diambilkan dari database yang datanya di simpan dalam variable agar dapat di tampilkan yang berbentuk array).
* Untuk total menggunakan function php number\_format agar harga tampil seperti format harga
* Untuk tanggal diatur dengan format date d-m-y tanggal-bulan-tahun lalu menggunakan strtotime($variable[0][‘tglorder’) untuk menyeting waktu yang sesuai
* Membuat table untuk menampilkan rincian dari order dengan tag <table>,<tr>,<td>.untuk head table menggunakan th lalu untuk isi menggunakann td yang berada di dalam foreach untuk looping data dari database
* Ambil data dari vorderdetail yang sudah dikirim dari array dan di transform menjadi $value pada foreach ,jadi untuk mengambil data menu $value[‘menu’] dan begitu juga untuk harga,jumlah sesuai pada kolom database.Untuk total menggunakan jumlah yang dikalikan dengan harga. Tidak lupa diberi endforeach untuk menutup looping foreach
* Pada view untuk pembayaran,menambahkan form untuk pembayaran juga 2 input untuk idorder dan total yang di hidden kan karena akan digunakan fungsinya untuk proses pembayaran namun tampilannya tidak perlu jadi di hide saja.
* Menambahkan session()->getFlashdata(‘info) pada file untuk update/pembayaran
* Membuat function update pada controller order,lalu mengirim data untuk pembayaran $bayar,$idorder & $total ->yg diambil oleh hidden input.
* Membuat statement if else untuk menjalankan proses pembayaran.If $bayar < $total maka akan menjalankan session getFlashdata untuk menjalankan warning message PEMBAYARAN KURANG lalu gunakan function find untuk menemukan datanya berdasarkan idorder yang sudah di kirim datanya $this->find($idorder);
* Sedangkan pada else maka akan menjalankan proses pembayaran untuk menyimpan nilai dari kembalian,pembayaran dan status pembayaran.
* Untuk kembalian membuat operasi aritamtika $kembali = $bayar - $total;
* Untuk meng-update dat ke database maka menggunakan $sql UPDATE tabel SET bayar=$bayar,Kembali=$kembali,status=1/ber-arti lunas lalu WHERE idorder=$idorder/menggunakan idorder karena untuk inisialisasi order mana yg dilakukan pembayaran
* Untuk men-koneksikan ke database menggunakan syntax dari code igniter $db = \Config\Database::connect(); dan menggunakan function query $db->query($sql/variable untuk sql);
* Menambahkan return ke base url order,yang ber arti saat proses pembayaran selesai maka page akan ter-load otomatis ke tabel order
* Menambahkan if isset $\_POST[‘bayar’] lalu masukkan semua isi dari function update ke dalam if isset.

9. Memahami & dapat menampilkan data order detail,yaitu :

* Membuat controller untuk order detail dalam folder app\controllers\Admin lalu menyambungkan dengan BaseController dengan use case juga untuk mengkoneksikan dgn model-nya.Untuk namescape juga lengkapi App\Controllers\*Admin*
* Membuat model untuk orderdetail lalu sambungkan dengan tabel vorderdetail,karena akan menggunakan data dari view order detail
* Membuat folder untuk view order detail dan file untuk view,copy isi dari kategori lalu paste dan sesuaikan dengan menampilkan tglorder,menu,jumlah,harga,total
* Pada controller panggil view dengan return view untuk ditampilkan
* Pada controller menambahkan paging halaman untuk view order detail yg datanya di ambil dengan menggunakan model yang berbentuk objek $objek = new model order detail;
* membuat array asosiatif untuk judul dari tabel dan paging nya yg menggunakan function paginate $objek->paginate(jumlah data perhalaman,’page’/link pada paging server yg sudah di tentukan)
* Kemudian mengirim data objek lewat array ‘pager’ => $objek->pager

10. Memahami & dapat membuat opsi untuk search engine data order dalam tabel order detail,yaitu :

* Membuat view cari dalam folder views orderdetail,copy dari view utama order detail lalu Menambahkan form untuk pencarian tanggal berdasarkan awal dan sampai
* Pada controller untuk orderdetail membuat function cari
* Pada function cari,menambahkan if isset $\_POST[‘awal] lalu di isi untuk mengirim data dari form awal dan sampai untuk mengirimkan data ke server untuk dicari.
* Menggunakan sql untuk men-select data dari vorderdetail untuk menemukan data yang akan di cari berdasarkan tanggal $sql = "SELECT \* FROM vorderdetail WHERE tglorder BETWEEN '$awal' AND '$sampai'";
* Men-koneksikan dengan database dengan syntax dari code igniter $db = \Config\Database::connect(); $result = $db->query($sql); untuk menselect data $row = $result->getResult('array'); untuk mengambil data
* Membuat array untuk menampilkan hasil pencarian pada view dari $row yg menggunakan function getResult
* Menambahkan return view untuk menampilkan view cari dari folder view orderdetail

11. Memahami & dapat menampilkan data user dari database pada view table,yaitu :

* Membuat model untuk user dan memanggil isi dari database user
* Pada controller buat function untuk menampilkan data dari database user.Membuat objek untuk memanggil model,menambahkan syntax untuk paging dari code igniter,membuat array untuk menampilkan data pada view yang berisi judul,data user yang sudah di bungkus objek juga paging nya lalu menampilkan pager untuk paging halaman.
* Menambahkan return view untuk menampilkan view dari folder views\user
* Pada folder user pada view,membuat file untuk view isinya copy dari kategori lalu mengubah dengan menyesuaikannya untuk user (menampilkan user,email,posisi/level,status,dan aksi).
* Untuk status gunakan statement if else terlebih dahulu utuk uji apakah user tersebut bernilai 1 jika iya maka akan tertanda bahwa user tersebut AKTIF dan apabila tidak maka akan tertanda BANNED

12. Memahami & dapat membuat opsi insert user,yaitu :

* Di dalam model untuk user menambahkan protected $allowFields untuk mengisi kolom pada database
* Membuat function create pada controller user untuk menampilkan view dari insert data dan mengirim data level menggunakan array
* Mmebuat view untuk insert data pada folder views user,copy dari kategori lalu paste dan di sesuaikan untuk insert user dgn menambahkan form (user,email,password) dan level nya.
* Untuk mengatur level nya copy dari form untuk menampilkan kategori lalu ubah value menjadi key yg sudah diatur sesuai user,juga sesuaikan label dan name untuk user
* Membuat function insert pada controller untuk menggunakan model yang dibungkus oleh objek,lalu menggunakan function insert dan untuk return page saat proses insert data berhasil.

13. Memahami & dapat membuat opsi delete user,yaitu :

* Pada model user tambahkan protected $primaryKey = ‘kolom id user’
* Buat function delete pada controller user isi dengan membuat objek untuk menggunakan model dan menggunakan function delete untuk men-delete data user
* Lalu tambahkan redirect agar page ter-load saat proses delete data berhasil

14. Memahami & dapat membuat opsi AKTIF BANNED untuk user,yaitu :

* Copy td untuk aktif banned dari pelanggan lalu paste pada user dan sesuaikan base\_url dan value nya
* Pada controller user membuat function update untuk proses banned aktif,yaitu statement if else dan juga function update
* Membuat objek untuk menggunakan model dan juga redirect page.

15. Memahami & dapat membuat proses validasi data untuk user,yaitu :

* Membuat validation rules dan validation rules pada model user dari code igniter
* Untuk user alpha\_numeric,min\_length,Is\_Unique
* Untuk email valid\_email
* Pada function insert di controller User,menghapus function insert
* Membuat if isset untuk mengirim data dari for yang di insert kan dgn $\_POST user,email,password,level,aktif.Untuk password langsung di enkripsi menggunakan function php password\_hash($\_POST[‘password’, PASSWORD\_DEFAULT]
* Masukkan symbol pada if
* Copy if else dari session flash data untuk error pada saat proses insert data dari menu,lalu paste di dalam if isset dan sesuaikan untuk User.
* Pada file insert user copy session flash data untuk indikasi message error dari menu lalu paste kan di awal code

16. Memahami & dapat membuat proses update data user untuk email dan level saja :

* Pada controller user membuat function ubah,lalu isi dengan : mengirim data dari iduser dgn $\_POST,membuat array untuk mengirim data update yaitu email dan level dgn $\_POST.
* Membuat objek untuk menggunakan model user dan function update,menambahkan return redirect base\_url user
* Membuat view update copy dari insert user lalu hapus kolom password dan beri judul degan variable pada array
* Pada form user hapus label dan ubah value menjadi iduser lalu type form menjadi hidden,karena berfungsi untuk inisialisasi iduser namun tidak perlu tampil.
* Pada form email dan level,tambahkan value untuk menampilkan data yang akan di update
* Pada option di form level tambahkan if untuk menguji apabila level dari user sesuai maka akan required/tampil dalam form

17. Memahami & dapat membuat form untuk login admin pada web,yaitu :

* Membuat controller untuk login admin,meng-koneksi dengan base\_url dan juga namescpace nya
* Membuat view yang berisi form login (email password buton login) untuk login pada folder template di folder views
* Dalam function index di controller login,membuat statement if else untuk mengambil data dari form login sekaligus menge-cek ketersediaan data dalam database apkaah bisa untuk login. Syntax $this->request->getMethod() untuk if nya & $this->request->getPost() untuk mengambil data dari form nya.
* Membuat objek untuk menggunakan model dari user & menggunakan method where->kolom form login & first menambahkan function untuk set session
* Membuat function untuk set session untuk mengirim data dalam bentuk array user,email,level dan loggedin apabila bernilai true // function ini untuk mengeset data login
* Meletakkan tampilan username email dan level dari admin yang log in di homepage dengan get session
* Mengatur navbar untuk tampilan admin yg sedang log in juga posisi nya dgn margin menggunakan bootstrap
* Menambahkan warning message/validation apabila proses log in salah dgn menambahkan if else pada If setelah proses where first.Pada if jika form/yang datanya akan di ambil dengan array kosong maka akan mengeluaran warning message yg sudah di bungkus array dalam sebuah variable.Lalu pada else nya akan menjalankan function untuk set Session dan redirect ke homepage admin
* Pada view log in menambahkan warning message yg sudah di simpan dalam sebuah variable lalu di tampilkan.
* Mengatur routes untuk menjalankan page log In saat pertama kali membuka admin page,yaitu open syntax $routes->get dan mengatur link nya (dari apay g saya tangkap function first yg digunakan seelumnya berperan di sini)
* Menambahkan opsi untuk logout pada view.Membuat function untuk logout dengan memanfaatkan session destroy
* Membuat file pada folder filters untuk redirect ke base\_url login apabila belum log in
* Pada file filters pada folder config tambahkan aturan class dari file baru yg dibuat pada folder filters ke function before
* Pada routes tambahkan opsi untuk akses menjadi ke semua aspek menu,kategori,order,pelanggan,order detail,user dgn $routes->add(‘controller’,’folder/file:: function pertama dari masing masing controller’).Jadi tidak akan bisa di akses apabila belum log in.
* Mengatur akses web dengan filter namun sebelumnya pada routes di atur terlebih dahulu, akses find,read,delete,create dsb. Dgn $routes->add agar akses terhadap data web lebih aman
* Mengatur statement pada view home page sesuai level dari admin yg masuk,apabila admin maka akan tampil semua aspek,apabila kasir hanya order & order detail dan apabila koki maka hanya order detail saja. Pada View admin untuk card menu buat if $level === ‘Level nya’ maka akan keluar opsi apa saja .
* Pada controller log in mengubah if dgn mengisi password\_verify untuk verifikasi password dari database dan system,Lalu menghapus password dari function where kemudian isi degan warning message email salah dan redirect ke homepage untuk else nya warning message password salah.

18. Memahami cara pemanggilan dari routes yaitu Folder/File::function

19. Memahami bahwa di dalam filters terdapat 2 function bawaan yaitu before & after

20. Memahami bahwa setiap filters/controller filters harus di daftarkan di dalam config filters terlebih dahulu(class nya)

21. Memahami fungsi routes untuk pem-filteran a.k.a pembatasan akses terhadap suatu page sesuai dengan yg di tentukan sebagai bentuk perlindungan data dsb.

22. Memahami bahwa tanpa mengatur routes maka sebuah controller tidak dapat di filter.