|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| CodeIgniter 4 Vid 10-19 | 90 |

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

1. Kita bisa menambahkan data menggunakan codeigniter.

2. Pertama tama kita ubah terlebih dahulu controller yang kita buat dan juga kita mengubah routesnya.

3. Kemudian, kita menambahkan tabel yang ingin ditambah datanya melalui database yang kita punya.

4. Untuk merapihkan kita bisa menggunakan tabel agar data yang ditampilkan tidak berantakan.

5. Kita buatkan function untuk menambahkan data.

6. Pada bagian untuk menambahkan data kita menggunakan form dengan method post untuk menambahkan datanya.

7. Untuk mengarahkan formnya ke function tadi pada bagian form action. seperti, Form action="<?= base\_url()?>/admin(nama folder)/kategori(nama file)/insert(nama function)"(kita arahkan form tersebut pada function untuk menambah data tadi).

8. Agar bagian form kita atur untuk wajib diisi oleh user kita tambahkan perintah required pada bagian input form.

9. Untuk memasukkan data yang kita tambahkan tadi ke database adalah dengan menambahkan protected $allowedFields = ['kategori', 'keterangan']; (kita isikan dengan nama kolom yang ingin dimasukkan datanya). kita tambahkan perintah tersebut pada file model yang kita telah buat.

10. Agar file untuk menambah data tadi tersambung dengan file model tadi kita tambahkana pada function untuk menambah data tadi dengan objek model seperti, $model = new Kategori\_M();(nama file model).

11. Lalu kita tambahkan function bawaan CodeIgniter untuk menambahkan data pada variabel objek yang kita buat untuk Kategori\_M tadi

seperti, $model ->insert($\_POST);

12. Agar setelah kita menambahkan data lalu kembali lagi pada halaman utama adalah dengan menambahkan dengan perintah return redirect()->to(base\_url()./admin/kategori). Maka, otomatis akan kembali lagi ke halaman utama lalu data akan otomatis bertambah tanpa harus kita refresh halamannya terlebih dahulu.

13. Kita menghapus data yang tersimpan.

14. Caranya kita memasukkan perintah protected $primaryKey = 'idkategori'(kita menggunakan idkategori karena dalam database terdapat icon kunci yang menunjukkan kolom tersebut adalah primary key). Kita memasukkan perintah tersebut pada file untuk model yang sudah kita buat.

15. Untuk menyambungkan perintah tadi ke function delete untuk menghapus data adalah kita menambahkan objek model yang kita buat tadi pada function delete seperti $model = new Kategori\_M(), alu kita masukkan function bawaan dari CI untuk menghapus data seperti $model ->delete($id).

16. Lalu, agar setelah data terhapus akan otomatis kemali lagi ke halaman utama adalah dengan menambahkan redirect tadi yaitu return reirect()->to(base\_url()."/admin/kategori"); (maka otomatisa akan kembali ke halaman yang dituju dan halaman akan otomatis terefresh).

17. Untuk bagian view untuk delete adalah dengan menambahkan kolom tabel pada bagian select untuk perintah menghapus. lalu agar isi dari kolom tersebut berfungsi kita menambahkan hyperlink seperti <a href="<?=base\_url()?>/admin/kategori/delete/<?=$value['idkategori']?>">Hapus</a>(maka ketika kita klik bagian hapus otomatis data yang memiliki id tersebut akan terhapus).

18. Kita dapat memperbarui data yang sudah tersimpan.

19. Pada bagian view select untuk menambahkan kolom tambahan untuk mengubah data. lalu, untuk isinya hampir sama dengan hapus data tadi hanya berbeda dengan controller yang dituju yaitu controller find.

20. Dalam Pengubahannya kita perlu function find untuk mencari data yang kita ingin ubah. Lalu, dalam function tersebut kita isi dengan objek yang kita telah buat. Lalu, kita gunakan function bawaan dari CI untuk mencari data yaitu function find seperti $kategori = $model->find($id) (Maka, data yang berada dalam variabel kategori akan tampil sesuai dengan id yang dituju menggunakan function find).

21. Lalu, untuk view menampilkan data yang ingin diubah dengan menggunakan view update lalu kita gunakan form untuk menampilkannya, Codingannya hampir sama dengan view insert. Form action untuk menambah data adalah kita arahkan ke controller update. Lalu, kita menambahkan value pada bagian input untuk menampilkan nama data kategori dan data keterangan yang dipilih, agar ketika kita pilih jika kedua atau salah satu data tersebut terisi maka akan otomatis terisi di dalam masing-masing form. kemudian, kita tambahkan input form untuk id kategori tetapi tidak kita tampilkan maka typenya adalah hidden.

22. Lalu, agar file update tampil di controller find maka kita gunakan return view untuk menampilkan isi dari file update tadi yang sudah kita buat. seperti return view("kategori/update",$data). Untuk bagian variabel datanya kita gunakan sama dengan controller read dengan mengganti isi judulnya.

23. Pada function update kita tambahkan objek model lalu kita isi objek model tersebut dengan function bawaan CI yaitu function save untuk menyimpan data yang telah diubah. Lalu, agar ketika sudah klik tombol simpan maka otomatis halaman akan langsung mengarah ke halaman utama dengan menambahkan redirect pada function update.

24. Kita bisa mengupload gambar yang kita inginkan.

25. Kita menambahkan gambar ini untuk bagian menu, maka kita membuat membuat controller untuk menu, dan membuat folder view baru untuk menu dan kita tambahkan file dengan nama form, codingannya sama dengan insert pada bagian view, tetapi kita hanya mengisi form untuk mengupload gambar yaitu type formnya adalah file. Untuk pemberian file kita tambahkan enctype="multipart/form-data" pada bagian form.

26. Pada controller menu kita lakukan return view untuk file form.

27. Untuk mengambil gambar kita buat function baru dalam controller menu dalam CI untuk mengambil gambar kita gunakan perintah getFile, seperti $file = $this->request->getFile('gambar(nama form untuk mengupload gambar)').

28. Untuk menampilkan nama dari gambar kita gunakan perintah getName, seperti $name = $file->getName(). Lalu, untuk menampilkannya adalah dengan menggunakan perintah echo $name(nama variabel pengambil nama dari gambar tersebut).

29. Untuk memindahkan gambar ke dalam folder yang kita tuju kita gunakan perintah move, seperti $file(variabel pengambil gambar)->move('./upload(folder yang dituju)').

30. Kita bisa menggunakan session dalam CI.

31. Dalam setiap pembuatan session harus diawali dengan inisialisasi/menyalakan sessionnya terlebih dahulu yaitu dengan $session = \Config\Services::session($config).

32. Kita bisa membuat isi session dengan membuat function terlebih dahulu yang bernama create lalu kita tambahkan data yang ingin di buat atau di tambahkan, dengan menggunakan array asosiatif. lalu, untuk mengaktifkannya kita gunakan perintah $session->set($tbluser(nama arraynya)).

33. Perintah untuk membaca isi sessionnya kita membuat function yang bernama read untuk membaca isi session yang telah kita buat tadi dengan menggunakan perintah echo $session->get(user(nama key dari array yang kita tambahkan tadi)).

34. Perintah untuk mengapus session adalah dengan membuat function terlebih dahulu dengan nama delete untuk perintah menghapusnya kita menggunakan $session->remove('email(nama key yang ingin dihapus)').

35. Perintah untuk menghapus seluruh isi session yang ada adalah dengan membuat function bernama destroy terlebih dahulu dengan memasukkan perintah 4session->destroy().

36. Dalam sebuah class untuk membuat variabel harus diawali dengan protected, karena kita ingin membuat variabel session maka $session=null. lalu, kita buat function construct agar setiap pembuatan function kita tidak menulis ulang insialisasinya maka perintahnya seperti $this->session=\Config\Services::session(). Sehingga jika kita membuat session hanya perlu menuliskan $this->session pada setiap perintah session.

37. Validasi data adalah kita memberi ijin apakah data tersebut boleh masuk atau tidak.

38. Untuk mengaktifkannya kita gunakan perintah bawaan yaitu protected $validationRules(untuk aturannya) dan protected $validationMessages (untuk tulisan peringatannya).

39. Pada bagian validation rules kita masukkan perintah 'kategori' => 'alpha\_numeric\_space(hanya boleh diisi dengan alfabet, angka dan spasi)|min\_length[jumlah minimal huruf data](jumlah minimal huruf data)|is\_unique(data tidak boleh sama)[tblkategori(nama tabel yang ingin diberi ijin).kategori(kolom tabel yang akan diisi)]'.

40. Pada bagian validation messages kita isi dengan keterangan/pesan untuk masing masing perintah. Jika salah satu perintah tersebut ada yang dilanggar maka salah satu pesan tersebut akan muncul.

41. Untuk menggunakannya kita tambahkan fungsi if dan pada controller yang membutuhkan validasi data seperti if ($model->insert($\_POST)===false){

$error=$model->errors();

echo $error['kategori'];

} else{

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

}.

42. Kita bisa menggunakan bootstrap untuk merapikan web menggunakan CI.

43. Untuk membagi isi tabel kita bisa menggunakan paging/pagination.

44. Cara untuk mengaktifkannya kita aktifkan dulu perintah bawaan dari CI untuk mengaktifkan pagination ini dengan $pager = \Config\Services::pager() pada function yang ingin digunakan untuk paging. Dalam paging diperlukan array asosiatif untuk menginput data yang ingin di bagi, kita menggunakan perintah seperti 'kategori' => $model->paginate(3(jumlah membagi halamannya),'group1'(nama pagingnya)), 'pager' =>$model->pager. Lalu, pada bagian view nya untuk pagination kita masukkan perintah <?= $pager->links('group1', 'bootstrap')?>. Untuk mengcustom pagingnya kita bisa gunakan contoh codingan dari CI, lalu kita taruh di view dengan menambahkan file baru untuk view pagingnya. kita bisa mengcustom paging menggunakan bootstrap untuk bagian ulnya kita tambahkan claa="pagination", untuk linya kita tambahkan class="page-item", dan untuk a href kita juga tambahkan class="page-link". Pada file pager dalam folder config kita tambahkan perintah bawaan dari CI yaitu 'bootstrap'(nama pagingnya)=>'App\Views\Template(folder penyimpan file view untuk paging)\paging(nama filenya)'.

45. Kita bisa menggunakan view cells.

46. Dengan cara kita membuat file baru pada folder view untuk view cellsnya dengan menambahkan select dan option.

47. Untuk mengaktifkannya kita membuat function baru dengan nama option untuk view cellsnya pada bagian menu. Lalu, kita isi dengan return view('Template(tempat menyimpan folder untuk view cells)/option(nama fielnya)'). Karena kita membuat view cells untuk form insert maka kita masukkan perintah <?= view\_cell('\App\Controllers\Admin(letak controllernya)\Menu(nama controllernya)::option(nama function untuk view\_cellsnya)')?>.

48. Untuk mengambil isi dari kategori/isi untu view cellsnya adalah dengan mengisi use App\Models\Kategori\_M(file modelnya) pada file controllernya. Kemudian, pada function option kita tabahkan objek $model = new Kategori\_M(). Lalu, kita buat variabel dengan nama $kategori => $model->findAll(). Kemudian, untuk mengirimkan datanya adalah dengan menambahkan array asosiatif dengan nama $data=['kategori(nama keynya)'=>$kategori(nama valuenya yang kita ambil dari model tadi)]. Supaya datanya tampil kita gunakan foreach untuk array asosiatif pada file option. Lalu, agar isinya sama dengan yang kita ingin tampilkan kita tambahkan <?= $value['kategori']?> pada optionnya. untuk value optionnya kita gunakan <?= $value['idkategori']?> agar yang dikirim adalah idnya.

49. Agar dalam pengisian data jika error maka akan tetap berada di halaman yang sama tetapi akan memunculkan tulisan kesalahannya adalah dengan memindahkan return redirect yang sebelumnya pada file kategori dan pada bagian function insert kita pindahkan ke bagian ifnya.

50. Untuk menampilkan pemberitahuan errornya kita gunakan perintah bawaan yaitu $session->setFlashdata('item', 'value') ini akan terjadi error karena $session pasti akan memerlukan inisialisasi session maka kita ubah sessionnya menjadi function dengan cara tanda $ kita hapuskan dan kita tambahkan (), menjadi session()->setFlashdata('info(untuk keynya)', $error['kategori'](untuk veluenya)).

51. Untuk mengambil isi set flashdata tadi kita masukkan perintah bawaan yaitu get flash data pada bagian view insert data seperti <?= session()->getFlashdata('info')?>.

**Saya Belum Mengerti**

1.

2.

3.