|  |  |
| --- | --- |
| Materi | Nilai |
| PHP- CODEIGNITER–VID10-19-WEEK18 | **96** |

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

1. Saya bisa membuat dan menambahkan kolom baru pada tabel di dalam database dengan cara memencet tombol struktur kemudian dibawahnya masukkan jumlah kkolom dan letaknya kemudian kirim dan tuliskan nama kolom yang diinginkan lalu simpan.
2. Supaya lebih mudah dan lebih ringkas kita bisa mengganti nama tiap-tiap function dan isinya, kita juga bisa menghapus function yang tidak kita perlukan.
3. Saya bisa menggunakan elemen **print\_r()** untuk mengecek data yang telah masuk . Misalnya **print\_r($\_POST)** maka hasilnya adalah nilai yang dimasukkan di dalam form akan ditampilkan dalam bentuk array asosiatif, dengan begitu kita akan lebih mudah dalam mengeceknya
4. Saya bisa membuat tabel dengan menggunakan fitur dari html yaitu tag table, contohnya **( <table>…</table> ) .**
5. Saya sudah mengerti bahwa di dalam tabel ada dua bagian yaitu kepala tabel yang berisi nama kolom dan badan tabel yang mengandung isi tabel, Untuk membaginya hanya perlu menggunakan tag **(<tr>…</tr>)** .
6. Saya bisa membuat form dengan menggunakan tag form **(<form></form)** kemudian digabungkan dengan tag input **(<input type=” “ > )** .
7. Saya bisa membuat penomoran yang secara otomatis akan terus bertambah sesuai data yang masuk, caranya adalah yang pertama kita harus membuat php open dan close lalu buat variabel bernilai 1, jika sudah kita bisa menuliskan variabel tersebut dengan operator increment **(++)** dan letakkan sesuai kebutuhan.
8. Penamaan merupakan hal yang sangat penting dalam menghubungkan file atau program yang kita buat dengan database, jika ada 1 huruf yang tidak sesuai maka tak akan terhubung ke database.
9. Dengan menggunakan perintah **protected $allowedFields = ['name', 'email'**];kita bisa memasukkan data ke dalam database, jadi kita tak perlu menggunakan perintah **“INSERT INTO…”** Sehingga lebih ringkas dan singkat.
10. Supaya perintah **protected $allowedFields = ['name', 'email'**]; dapat berjalan kita memerlukan function insert, contohnya **$model->insert($\_POST) .**
11. Setelah melakukan proses insert data dengan menggunakan form biasanya kita akan langsung diarahkan ke halaman tertentu dengan mneggunakan elemen **header(),** tapi di codeigniter menyediakan cara lain yaitu menggunakan **redirect**, contohnya **return redirect()->to(base\_url().”/admin/kategori”;** .
12. Untuk menghapus atau mendelete data yang harus disiapkan adalah kita harus menentukan primary key, primary key ini dapat kita lihat di dalam database, ditandai dengan adanya simbol kunci. Jika sudah kita bisa membuat variabel seperti berikut **protected $primarykey = ‘idkategori’** .
13. Saya bisa menambhakan kolom baru pada tabel untuk kolom hapus, caranya tinggal menambahkan dua tag td **(<td></td>)** Untuk nama kolom dan isi kolom.
14. Saya sudah mengerti cara membuat object, yaitu tinggal membuat variabel yang isinya diawali dengan **new**, contohnya **$model = new Kategori\_M()** .
15. Saya sudah bia menghapus data dengan menggunakan kode perintah **$model-> delete($id)** . Kode tersebut mengandung function untuk menghapus data yang sudah disediakan oleh codeigniter, kemudian setelanya itu merupakan nomor id dari row yang akan dihapus, ditulis dalam bentuk variabel.
16. Saat kita sedang menggunakan codeigniter, sebaiknya kita selalu melihat panduan atau user guide supaya kita lebih mudah dalam menggunakan codeigniter, untuk membuka user guide kita hanya perlu masuk ke web resminya kemudian pilih guide.
17. Update data memiliki peran penting untuk memperbaiki atau mengubah data jika ada salah penulisan.
18. Saya sudah mengerti bahwa kita bisa memanfaatkan function dari codeigniter yaitu find() untuk digunakan di menu update atau ubah data.
19. Dengan menggunakan perintah  **return view("kategori/update",$data);** . Untuk menghubungkan file berisi form update dengan file controller dan memunculkannya di halaman utama.
20. Codeigniter menyediakan berbagai function atau function, contohnya function save() yang berguna untuk saving data atau menyimpan data.
21. Saya sudah bisa membuat pilihan update data, caranya yang pertama adalah kita harus mencari mana yang harus diubah dengan menggunakan function find(), jika sudah maka dapat ditampilkan di dalam view dari update dalam bentuk form, jika sudah diubah maka tinggal dikirim.
22. Saya sudah tahu bahwa jika kita membuart input dalam bentuk upload file kita juga harus menambahkan kode **enctype=”multipart/form-data”;** di dalam tag input.
23. Kita bisa memunculkan nama dari file yang kita upload yaitu dengan menggunakan function **getName(),** Contoh penggunanya adalah **$name = $file->getName; echo $name; .**
24. Untuk mendapatkan atau mengambil file kita bisa menggunakan function **getFile(),** Contoh penggunaanya adalah **$file = $this ->request->getFile(‘gambar’);** .
25. Codeigniter sudah menyediakan function **move()** untuk membuat gambar yang diupload secara otomatis masuk ke dalam foder upload, jadi kita hanya perlu menentukan di folder mana gambar tersebut akan dibawa setelah diupload, contoh penggunaanya yaitu **$file-> move(‘./upload’); .**
26. Yang harus diperhatikan dari upload adalah yang pertama kita harus pastikan bahwa type pada formnya merupakan file lalu jangan lupa untuk menambahkan enctype.
27. Saya sudah mengerti bahwa dalam membuat session langkah pertama yang harus kita lakukan ialah inisialisasi session. Contohnya **$session = \Config\Services::session(); .**
28. Setelah session di inisialisasikan atau dinyalakan maka session sudah siap untuk digunakan.
29. Untuk menampilkan session tersebut kita harus menggunakan get, misalnya

**echo $session->get($tbluser); ,**Dan di browser kita haus mengetikkan setelah user /create dan jalankan, jika sudah, ganti **/create** menjadi **/read** lalu jalankan, maka isi dari session akan tampil di halaman utama.

1. Saya sudah mengerti bahwa jika kita tidak menginisialisasikan session terlebih dahulu maka akan muncul eror, jadi kita harus menambahkan inisialisasi session di tiap function yang terdapat session.
2. Selain menggunakan unset, ternyata kita bisa menghapus atau me remove data dari session dengan menggunakan function bernama **remove(),** contonya **$session->remove('email');.** Namun sebelum itu kita harus membuat function bernama delete dulu supaya lebih mudah.
3. Jika kita menggunakan destro, maka akan membuat seluruh isi dari session tersebut hilang atau terhapus, contoh penggunaanya yaitu **$session->destroy();** .
4. Dengan adanya **\_construct**, maka saat kita membuat suatu class, secara otomatis akan mulai atau langsung berjalan mengisi variabel.
5. Validation atau validasi merupakan sesuatu yang digunakan untuk memeriksa apakah sebuah data itu diizinkan untuk masuk ke database atau tidak.
6. Saya sudah bisa membuat validation.
7. Saya sudah mengerti bahwa dalam menggunakan bootstrap itu ada dua cara yaitu offline dan online, Jika offline maka kita bisa menghubungkan file dengan folder bootstrap, jika online kita bisa mengh8bungkannya melalui link.
8. Saya sudah mengerti dan mengetahui bahwa kita bisa menggunakan lebih dari 1 framework, seperti codeigniter dan bootstrap, menurut saya itu terjadi karena masing-masing framework punya peran yang berbeda, jika codeigniter akan berperan di back- end dan bootstrap akan berperan di front end.
9. Saya bisa mengganti warna button dengan menggunakan clas dari bootstrap, sepertii success untuk warna hijau, danger untuk warna merah, primary untuk warna biru.
10. Saya sudah mengerti bahwa paging merupakan pembagian data menjadi beberapa bagian yang dipisahkan dengan halaman.
11. Codeigniter menyeiakan cara untukmembuat paging, caranya yang pertama adalah kita harus memasukkan librarynya ke dalam file controller, contoh nya **$pager = \Config\Services::pager(); .**
12. Untuk menentukan jumlah pembagian paging, biasanyakita harus membuat variabel dulu kemudian dikombinasikan dengan function,session ataupun ifelse, namu codeigniter sudah menyediakan cara yang jauh lebih singkat dan cepat, yaitu dengan menggunakan **kategori' => $model->paginate(2),** di dalam function pagination tersebut kita bisa memasukkan jumlah data yang kita inginkan.
13. Saya sudah bisa membuat paging atau pagination dengan menggunakan bootstrap, yaitu dengan class pagination, page-item, dan page-link.
14. View cells merupakan pembagian dari view supaya lebih sederhana.
15. Saya sudah bisa membuat function, contohnya public function **option(){ …}** .
16. Saat kita ingin menggunakan view, kita harus menggunakan return di dalam function, kemudian diikuti oleh function view, contohnya **return view(‘template/option;);** . jadi kita memang tidak boleh menggunakan echo.
17. View cells merupakan bagian terkecil dari view, view cell ini dapat menerima data dari parameter dan bisa juga mengakses data dari model, prinsip kerjanya mirip dengan controller
18. Saya sudah bisa membuat view cell, contohnya **<?= view\_cell('\App\Controllers\Admin\Menu::option') ?>** ..
19. Jika kita tidak membuat path atau inisialisasi seperti use App\Models\Kategori\_M; . Maka akan muncul error berupa **“WHOOPS!!”** yang sangat mengesalkan.
20. Sesion flashdata merupakan session yang hanya bersifat sementara. Flashdata ini dapat kita manfaatkan ketika kita ingin membuat suatu form yang jika tidak sesuai mka akan keluar alert atau peringatan.
21. Saat kita ingin membuat set flashdata, di petunjuk dari codeigniternya session akan berupa variabel, seperti **$session->setFlashdata('item', 'value');** sehingga perlu dinyalakan atau diaktifkan **.** Sehingga akan muncul error, untuk mengatasinya kita bisa mengubah variabel tersebut menjadi function, contohnya **session()->setFlashdata('item', 'value');**
22. Yang harus diperhatikan dari flashdata ialah yang pertama kita harus mengganti variabel session menjadi sebuah fungsi sehingga kita tak perlu menginisialisasikannya. Untuk mengisi flashdata kita bisa menggunakan perintah **setFlashdata(),** Untuk mengambil flashdata kita bisa menggunakan **getFlashdata()** .

**Saya kurang paham**

1. -