|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Tugas coding minggu – 18-20 web CI4 | 90 |

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

1. Menampilkan data order yang telah di buat sebelumnya di dalam database dan di tampilkan ke dalam bentuk tabel
2. Penggunaan query builder di dalam ci4 untuk mengambil seluruh data yang ada di dalam database
3. Penggunaan tag $db = \Config\Database::connect(); untuk mengambil data dan menghubungkan db dengan database yang telah di buat di phpmyadmin
4. Penggunaan paging manual
5. Penggunaan tag $sql = "SELECT \* FROM vorder ORDER BY status ASC LIMIT
6. Penggunaan tag $sql = "SELECT \* FROM vorder ORDER BY status ASC LIMIT 0,
7. Penggunaan tanggal bulan dan tahun pada ci4 menggunakan tag
8. Menampilkan isi data lain dari database yang telah di buat ke dalam bentuk array asosiatif
9. Mengupdate atau menambahkan atau mengubah sebuah data yang ada di dalam database melalui function yang telah di buat
10. Penggunaan setflashdata pada codeIgniter
11. Melakukan perubahan nilai pada sebuah variable data di dalam database melalui codeIgniter dengan menggunakan tag sql
12. Penggunaan tag if($value['aktif']==1)$aktif="Aktif"; else $aktif="Banned";
13. Penggunaan validation di dalam codeIgniter untuk menampilkan sebuah pesan apabila sebuah statement tidak sesuai berjalan dengan perintah yang di buat sebelumnya
14. Menampilkan sebuah statement di dalam statement melalui proses form dan di tampilkan dalam kedalam sebuah variable lalu di tampilkan kembali isi data yang ada di dalam variable tersebut
15. Penggunaan tag $user = $model->where(['email'=>$email, 'password'=>$password, 'aktif'=>1])->first();
16. Pembuatan page login yang terhubung dengan database yang telah di buat sebelumnya
17. Penggunaan tag echo session()->get('level'); untuk menampilkan isi data session yang telah di buat sebelumnya
18. Penggunaan tag if (!empty(session()->get('level'))) di gunakan untuk apabila sebuah statement tidak kosong maka akan mengambil data dari session yang telah di buat sebelumnya
19. Penggunaan filter yang telah di sediakan oleh codeIgniter untuk antisipasi agar sebuah page tidak bisa di akses secara langsung tanpa harus melalui statement awal terlebih dahulu
20. Pengaturan seluruh routes di dalam folder routes yang telah di sediakan oleh codeIgniter
21. Penggunaan hak ases di dalam codeIgniter untuk mengelompokkan beberapa kategori sesuai level statement yang telah di tentukan
22. Penggunaan code enkripsi untuk password pada codeIgniter
23. Menambahkan sebuah icon dari bootstrap ke dalam sebuah hyperlink atau sebuah statement yang telah di buat sebelumnya
24. Merapikan sistem update data pada bagian statement update agar teratur dan sesuai dengan urutan yang di inginkan saat sebuah elemen di inputkan ke dalam database
25. Menambahkan function form-group yang telah di sediakan oleh bootstrap ke dalam form table insert dan update yang telah di buat sebelumnya
26. Menampilkan pesan eror yang telah id buat sebelumnya ke dalam satu statement update yang telah di buat sebelumnya
27. Penggunaan tag $no = ($jumlah \* $page) - $jumlah+1; pada statement if agar mengatur sistem nomor pada sebuah data yang di tampilkan dalam bentuk table agar ter urut sesuai urutannya
28. Penggunaan tag if (isset($\_GET['page\_page'])) untuk di saat sebuahkondisi statement page\_page maka akan menjalankan sebuah statement di dalamnya
29. Merapikan tampilan eror yang telah di sediakan oleh bootstrap ke dalam sbeuah statement saaat sebuah statement tersebut berjalan
30. Merapikan tampilan table yang telah di buat
31. Menampilkan sebuah gambar yang ada di dalam sebuah database yang telah di buat sebelumnya
32. Penggunaan tag number\_format($value['harga']) untuk membuat agar sebuah angka yang mencapai ribuan terdapat koma setelah 3 angka 0 di depannya
33. Menampilkan seluruh gambar yang ada di dalam database dengan statement looping menggunakan foreach
34. Menampilkan beberapa data dalam sebuah beberapa kategori yang telah di buat sebelumnya melalui statement select option
35. Menampilkan jumlah data yang di inginkan dalam satu kategori tiap page melalui statement findALL melalui id tiap menu yang akan di tampilkan
36. Menghapus beberapa data melalui icon hapus di dalam table yang telah di buat sebelumnya dengan statement function delete yang telah di buat sebelumnya
37. Menginsertkan sebuah data ke dalam database melalui table form yang telah di buat
38. Menampilkan kembali sebuah insert data yang telah di lakukan sebelumnya ke dalam tampilan table
39. Penggunaan tag $request->getPost('menu') untuk menampilkan kembali data yang telah di inputkan di dalamnya
40. Penggunaan tag if (empty($name)) { $name = $this->request->getPost('gambar'); di gunakan untuk dalam sebuah statement apabila saat terjadi perbuahan namun pada bagian yg di maksud di dalam statement getPost tidak di ubah maka akan memakai data yang lama,namun jika menggunakan tag else { $file->move('./upload'); maka jika pada bagian tersebut ada perubahan maka akan otomatis masuk ke dalam folder tersebut saat di klik simpan
41. Penggunaan tag $file = $this->request->getFile('gambar');
42. Penggunaan tag <?php if ($value['idkategori'] == $menu['idkategori'])
43. Penggunaan tag $model->paginate untuk menampilkan data yang di paging
44. Mengganti status sebuah pelanggan atau dari isi element 1 ke 0 dan sebaliknya melalui statement if variable isi nya adalah 1 maka akan menjalankan statement variable 0,dan sebaliknya
45. Penggunaan query pada codeIgniter yang telah di sediakan oleh codeIgniter dalam pengambilan data di dalam database
46. Membuat sebuah table pada codeIgniter yang terhubung dengan database yang telah di buat
47. Menampilkan isi data dari sebuah table database ke dalam table yang telah di buat sebelumnya dengan menggunakan tag menggunakan statement foreach agar seluruh data yang ada dalam table di dalam database dapat di tampilkan
48. Penggunaan tag protected $allowedFields
49. Penggunaan tag return redirect()->to(base\_url()."/Admin/kategori"); untuk apabila sebuah statement telah selesai dan menekan tombol submit maka akan kembali ke halaman awal sesuai dengan url yang telah di atur di dalam statement return redirect
50. Menginputkan sebuah data ke dalam table yang ada pada database yang tela di buat melalui statement insert yang telah di sediakan oleh codeIgniter
51. Menghapus sebuah data yang ada dalam sebuah table di dalam database dengan menggunakan tag delete()
52. Penggunaan tag protected $primaryKey = 'idkategori'; untuk mengambil data yang telah di buat sebagai primary key di dalam sebuah table di database yang telah di buat
53. Mengubah data yang ada dalam database melalui statement function find
54. Menampilkan data yang akan di ubah ke dalam form yang telah di buat
55. Penggunaan tag $model->save($\_POST);
56. Penggunana input type hidden agar data yang di panggil tidak aan di tampilkan
57. Mengupload sebuah gambar ke dalam database dengan menggunakan statement form dengan input type file
58. Penggunaan tag $name = $file->getName(); untuk mengambil data nama file yang telah di upload untuk di tampilkan
59. Penggunaan tag $file->move('./upload'); untuk memindahkan file gambar atau apapun ke dalam sebuah folder upload atau folder apapun yang telah di atur di dalam url function move yang telah di sediakan oleh codeIgniter
60. Mengatur seluruh session yang telah di buat di dalam codeIgniter
61. Penggunaan tag $this->session->set($tbluser); untuk memproses session yang mengambil data tbluser yang telah di buat di dalam di database
62. Penggunaan protected $session=null; untuk sebagai awalan seluruh session agar penulisan tag session lebih efisien dan fleksibel
63. Penggunaan public function create() untuk membuat sebuah session baru dari elemen variable yang telah di buat
64. Penggunaan public function read() untuk membaca elemen dari variable sebuah session yang telah di buat sebelumnya
65. Penggunaan public function delete() untuk menghapus beberapa elemen dari variable sebuah session yang telah di buat sebelumnya
66. Penggunaan tag public function destroy() untuk menghapus seluruh elemen dari variable sebuah session yang telah di buat sebelumnya
67. Penggunaan tag $session = \Config\Services::session();
68. Penggunaan validation pada codeIgniter agar di saat pengisian data ke dalam database melalui form tidak ada yang double atau sama dengan data yang lain
69. Penggunaan tag validationRules di gunakan untuk sebagai aturan agar inputan sebuah data tidak terjadi double di dalam database
70. Penggunaan tag validationMessages di gunakan untuk memberikan notifikasi atau pesan bahwa user telah menginputkan sebuah data yang tidak sesuai dengan aturan yang telah di buat
71. Penggunaan tag alpha\_numeric\_space pada statement validaton agar data input an ke dalam database tidak ada yang menggunakan symbol
72. Penggunaan tag min\_length pada statement validation agar data input an ke dalam database minimal harus 3 huruf atau sesuai dengan aturan validation yang telah di buat
73. Penggunaan tag is\_unique pada statement validation agar data input an ke dalam database harus tidak ada yang sama atau sesuai dengan aturan yang telah di buat sebelumnya di dalam validationRules
74. Menambahkan bootstrap ke dalam codeIgniter
75. Membuat sebuah navbar di dalam codeIgniter melalui tag yang telah di sediakan oleh bootstrap
76. Penggunaan paging dalam codeIgniter maupun bootstrap ke dala web yang akan di buat
77. Penggunaan tag 'kategori' => $model->paginate(3,'group1'),i
78. Penggunaan tag 'bootstrap' => 'App\Views\template\paging'
79. Pembuatan view cell di dalam codeIgniter dengan menggunakan tag view\_cell yang telah di sediakan oleh codeIgniter
80. Pengggunaan flash data di dalam codeIgniter untuk menampilkan sbeuah statement di dalam statement menggunakan tag session()->setFlashdata
81. Penggunaan tag echo session()->getFlashdata('info');