|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Laravel 8 & Tailwind CSS | 95 |

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

1. Sebelum mengisntal laravel harus menginstal composer terlebih dahulu.

2. Xampp php yang harus dipakai adalah versi 7.4 keatas.

3. Xampp control harus dinyalakan terlebih dahulu apache dan mysql nya baru bisa digunakan.

4. Ketik composer di dalam cmd untuk melihat composer yang sudah diinstal.

5. Ketik create-project laravel/laravel namaproject untuk membuat/install laravel.

6. Php artisan serve untuk menjalankan laravel.

7. Ctrl+click pada development server untuk langsung membukanya di dalam browser.

8. Sebelum meingstall tailwind-css harus menginstal node terlebih dahulu.

9. Node-v untuk melihat versi dari node js yang diinstal.

10. Npm install untuk menginstal npm.

11. Npm install –D tailwindcss untuk menginstal tailwind css.

12. Npx tailwindcss init untuk membuat tailwind.config.js.

13. Masukkan tailwind layer ke dalam app.css.

14. Di dalam mix.js masukkan require(‘tailwindcss’). Untuk mengupdate config dan memasukkan tailwind ke dalam framework.

15. Npm run watch untuk menjalankan dan melihat semua css dan javascript yang berkaitan.

16. Tambahkan link di dalam welcome.blade.php untuk mengambil css yang sudah ada.

17. Di dalam routes yang sudah disediakan oleh laravel terdapat 4 php yaitu: api.php, channels.php, console.php, dan web.php.

18. Api.php digunakan untuk merouting API.

19. Channels.php untuk merouting broadcast.

20. Console.php untuk merouting artisan.

21. Web.php untuk merouting web.

22. Route::get('login', function () {

return view();

}); Untuk melakukan routing bagian mengambil data.

23. Didalam views ketika membuat file harus dengan .blade.php.

24. Return view() untuk mengambil view yang ada di resources.

25. Ketika file yang di view berada di dalam folder dipanggil dengan “.” Atau “/”.

26. Membuat template dari blade.php.

27. {{asset(‘css/app.css’)}} untuk memanggil css dari tailwind di dalam blade.php.

28. @extends() untuk memilih layout mana yang akan di inherit.

29. @section() untuk menaruh html yang ada di dalam blade.php setelah melakukan extends. @endsection adalah penutup dari @section.

30. @yield sebagai penghubung dan menampilkan data dari extends.

31. Php artisan make:controller namaController untuk membuat controller.

32. Di dalam controller membuat function.

33. Membuat route untuk memanggil function yang sudah dibuat di controller.

34. Membuat database di phpMyadmin terlebih dahulu untuk laravel.

35. Masukkan nama database yang sudah di buat tadi ke dalam .env DB\_Database.

36. Php artisan make:migration namaMigration untuk membuat migration.

37. Data yang ingin di migrasi masukkan di dalam function up sedangkan function down untuk ketika di rollback akan hilang juga.

38. Php artisan migrate untuk melakukan migrasi ke dalam database.

39. Php artisan migrate:rollback untuk memanggil kembali atau menghapus migrasi yang sudah dilakukan.

40. Php artisan make:seeder namaSeeder untuk membuat seeder.

41. DB::table('kategoris')->insert([

'kategori' => 'Makanan',

]); Untuk membuat isi dari seeder.

42. Install php namespace resolver untuk membantu dalam import class yang belum bisa digunakan.

43. Di dalam DatabaseSeeder.php masukkan seeder yang akan dipanggil.

44. Php artisan db:seed untuk melakukan seeding data ke database.

45. Str::random() untuk membuat string acak.

46. Php artisan make:model namaModel untuk membuat model.

47. protected $table = 'kategoris';

protected $fillable = ['kategori']; Model untuk menghubungkan dengan database.

48. Ketika menggunakan factory harus sudah ada model.

49. Php artisan make:factory namaFactory untuk membuat factory.

50. Factory digunakan untuk mengisi data melalui cara seeding.

51. Faker digunakan untuk membuat data yang fake/palsu.

52. $kategoris = Namamodel::all() untuk menampilkan semua data yang ada.

53. Compact mengambil variabel yang ada di parameter.

54. @foreach ($kategoris as $isi)

<li>{{ $isi->kategori }} -- <a href="/store/{{ $isi->idkategori }}">Hapus</a></li>

@endforeach untuk menampilkan data dari database yang sudah diambil tadi.

55. @csrf untuk membuat token field.

56. Membuat form untuk menampilkan data dari database.

57. Ketika mengganti method dalam form maka membuat routing post juga.

58. Membuat variabel untuk memanggil model setelahnya dipanggil untuk menambah data.

59. Save() untuk menyimpan data.

60. Return redirect(‘/’) untuk mengembalikan lagi ke tempat yang dituju.

61. Membuat routing untuk merubah/memperbarui data.

62. $kategoris = [

'kategori'=>$request->kategori,

];

Kategori::where('idkategori', $id)

->update($kategoris);

return redirect('/'); Untuk membuat function merubah/memperbarui data yang sudah ada.

63. <a href="/store/{{ $isi->idkategori }}">Hapus</a> digunakan untuk menghapus data yang ada di database.

64. Delete() Untuk menghapus data.

65. Update() untuk merubah data yang sudah ada.

66. $this->validate($request, [

'kategori' => 'required | max:30',

]); Untuk membuat validasi data.

67. @error('kategori')

<p>{{ $message }}</p>

@enderror Untuk menampilkan pemberitahuan jika ada yang salah.

68. Membuat folder terlebih dahulu untuk menaruh file image nantinya.

69. Input bertipe file untuk menambahkan file seperti foto.

70. $name = $request->file('image')->getClientOriginalName();

$request->image->move(public\_path('images'), $name);

dd('sukses'); Untuk menambahkan data berupa file ke dalam project.

71. module.exports = {

content: [

'./storage/framework/views/\*.php',

'./resources/\*\*/\*.blade.php',

'./resources/\*\*/\*.js',

'./resources/\*\*/\*.vue',

],

theme: {

extend: {

colors: {

"ungu":"#5267DF",

"merah":"#FA5959",

"biru":"#242A45",

"abu-abu":"#9194A2",

"putih":"#f7f7f7",

},

container: {

center:true,

padding:"1rem",

screens:{

lg:"1124px",

xl:"1124px",

"2xl":"1124px",

},

},

},

},

plugins: [],

} Untuk menkonfigurasi apa saja customize yang akan digunakan dari tailwindcss.

72. –help untuk melihat apa saja opsi yang ada.

73. Php artisan make:model namaModel –a yang berarti selain membuat model maka semua yang ada akan langsung dibuat juga.

74. Membuat data apa saja yang akan dimasukkan ke dalam database di dalam migrations.

**Saya Belum Mengerti**

1.