|  |  |
| --- | --- |
| Materi | Nilai |
| Laravel & Tailwinds CSS | 95 |

Saya Sudah Belajar dan Mengerti dan Saya BISA

1. Sebelum menginstall Laravel kita harus menginstall composer terlebih dahulu melalui website composer
2. Setelah composer terinstall kita bisa memulai installasi Laravel dengan menggunakan CMD, kemudian masukkan composer create-project laravel/laravel example-app di lokasi file yang diinginkan
3. Menyalakan Laravel dengan terminal kemudian masukkan perintah php artisan serve
4. Kemudian install tailwind melalui terminal dengan memasukkan perintah npm install, npm install -D tailwindcss, npx tailwindcss init.
5. Kemudian masukkan content: [

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

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

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

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

didalam file tailwind.config.js

1. Masukkan juga

@tailwind base;

@tailwind components;

@tailwind utilities;

Kedalam file recources/css/app.css

1. Yang terakhir tambahkan require('tailwindcss'), pada file webpack.mix.js
2. Untuk mencoba apakah tailwinds sudah terinstall dengan benar atau belum, buka file views dan isi file welcome dengan html, kemudian nyalakan dengan php artisan serve
3. Route website dapat diatur melalui folder routes file web.php
4. Route::get('login',function(){

return view('admin.login');

});

Digunakan untuk route pada web.php yang berarti url/login akan menjalankan atau membuka file pada views/admin/login

1. Tanda . pada route dapat diganti dengan / begitu pula sebaliknya
2. Controller dapat dibuat dengan cara memasukkan php artisan make:controller NamaController pada terminal
3. Controller berisi function dan mereturn sebuah file agar dapat dipanggil
4. Migration digunakan untuk membuat tabel secara otomatis didalam database, namun sebelum itu kita harus mengubah nama database yang digunakan pada file .env
5. Cara membuat file migration dengan memasukkan php artisan make:migration NamaMigration
6. Didalam file migration yang barusan dibuat terdapat function up, isi function up dengan

Schema::create('NamaTabel', function (Blueprint $table) {

$table->tipedata('NamaKolom');

$table->tipedata('NamaKolom’);

$table->timestamps();

});

1. Seeder digunakan untuk mengisi data palsu pada tabel
2. Cara membuat file seeder adalah dengan memasukkan php artisan make:seeder NamaSeeder
3. Didalam file seeder terdapat function yang dapat dijalankan untuk mengisi tabel, isi function dengan

DB::table(NamaTabel)->insert([

'NamaKolom' => isi,

]);

1. Kemudian cara menjalankannya dengan memasukkan php artisan db:seed pada terminal
2. Buat file model dengan memasukkan php artisan make:model NamaModel pada terminal
3. Didalam class model tadi masukkan

protected $table = ‘NamaTabel’;

protected $fillable = ['NamaKolom'];

agar model dapat mengetahui tabel dan kolom apa saja yang dapat diisi

1. Factory digunakan untuk memasukkan data palsu seperti seeders, namun dapat lebih terstruktur dan dapat diatur isinya
2. Cara membuat file factory dengan memasukkan php artisan make:factory NamaFactory
3. Didalam function definition di file factory tadi, masukkan 'NamaKolom' => $this->faker->name(),
4. Kemudian pada file seeder ganti isi function dengan Kategori::factory(5)->create; jangan lupa juga tambahkan models pada file seeders tadi
5. Untuk mengupload file hanya perlu memasukkan php artisan db:seed pada terminal
6. Menampilkan data dengan cara mengambil data yang dikirimkan oleh controller kemudian di foreach pada views
7. Insert data dengan cara membuat form post pada views, kemudian masukkan url pada action. Jangan lupa buat route menggunakan post dan url form tadi, serta jalankan function controller yang berisi data tadi yang di post
8. Update data dengan cara mengirimkan id melalui url ke controller, kemudian diterima parameter, dan buat array yang berisi nama kolom dan value request tadi dan sambungkan ke model untuk menentukan $id yang diupdate kemudian jalankan function update
9. Cara delete data hampir sama dengan saat update data, yaitu kita hanya perlu mengirimkan $id ke controller lalu menjalankan function delete
10. validasi digunakan agar saat terjadi error tidak muncul pesan error pada website, seperti pada contoh kolom kategori harus selalu ada isinya, jika tidak akan terjadi error screen, oleh karena itu kita perlu memakai validate required pada controller, dan juga menampilkan pesan dengan memasang $message pada home
11. cara meng upload gambar dengan menambahkan form insert file terlebih dahulu

<input type="file" name="image"> kemudian ubah enctype pada form menjadi multipart/form-data lalu buka file controller dan tambahkan

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

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

Pada function store, setelah itu coba upload gambar dan lihat apakah gambar tersebut sudah terpindah di folder image

Saya Belum Mengerti