

PRAKTIKUM PEMROGRAMAN BERBASIS WEB

Untuk Memenuhi Tugas 8 P.PBW



Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

1. Setup Laravel

Pada tahap pertama, dilakukan instalasi Laravel menggunakan Composer dan pengaturan awal untuk proyek. Setelah instalasi, lingkungan proyek Laravel disiapkan dengan menjalankan php artisan serve untuk melihat aplikasi berjalan di server lokal. Tahap ini bertujuan untuk memastikan instalasi berhasil dan memastikan semua dependensi Laravel terpasang dengan benar.

Langkah-langkah:

1. Instal Laravel menggunakan Composer.
2. Konfigurasi environment file (.env).
3. Jalankan server lokal untuk melihat hasil instalasi.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=pertemuan8
DB_USERNAME=root
DB_PASSWORD=
```

2. Routing dan View

Pada tahap ini, dipelajari dasar-dasar routing di Laravel untuk mengarahkan URL ke halaman yang diinginkan. Selain itu, view sederhana dibuat menggunakan Blade templating engine untuk menampilkan halaman tertentu sesuai dengan route yang ditentukan.

Langkah-langkah:

1. Membuat route sederhana di file web.php.
2. Menambahkan view menggunakan Blade untuk halaman utama.
3. Menguji route dengan menjalankan server lokal.

```
<?php

use App\Http\Controllers\AboutController;
use App\Http\Controllers>ContactController;
use App\Http\Controllers\GalleryController;
use App\Http\Controllers\HomeController;
use Illuminate\Support\Facades\Route;

Route::get( uri: '/', action: [HomeController::class, 'index']);
Route::get( uri: '/about', action: [AboutController::class, 'index']);
Route::get( uri: '/contact', action: [ContactController::class, 'index']);
Route::get( uri: '/gallery', action: [GalleryController::class, 'index']);

Route::get(uri: 'users', action: function () {

    $users = [

        ['id' => 1, 'name' => 'John Doe', 'email' => 'dzaki@gmail.com'],
        ['id' => 2, 'name' => 'John Doe', 'email' => 'dzaki@gmail.com'],
    ];

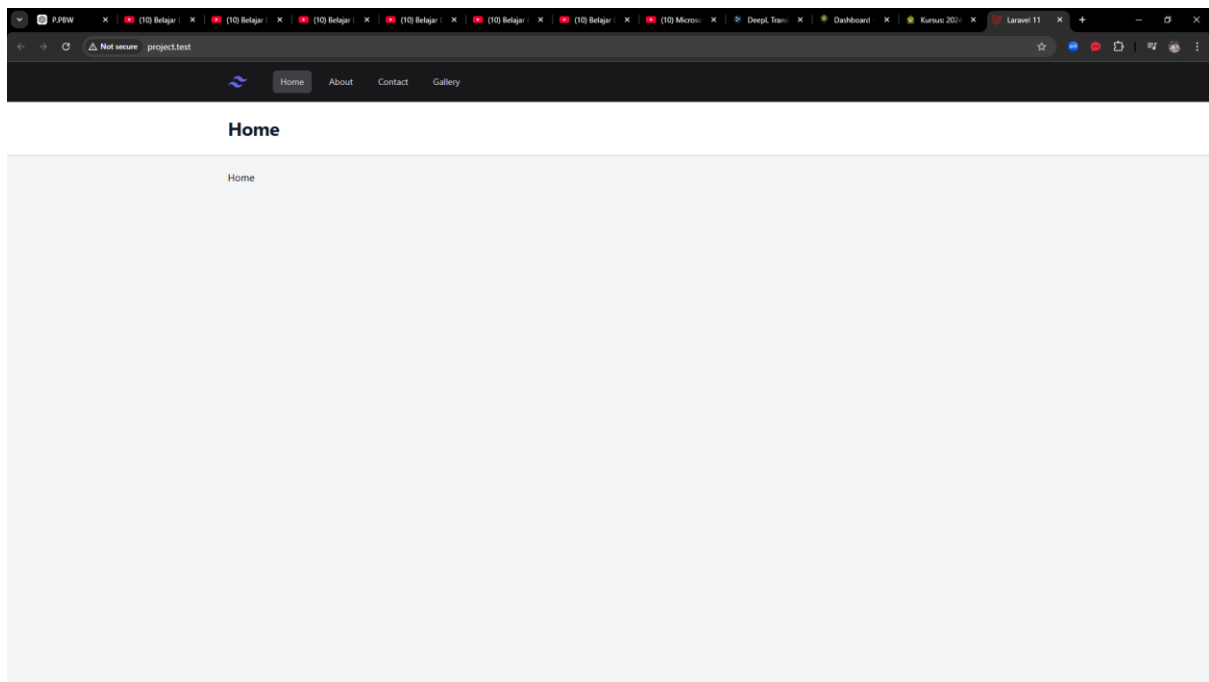
    return view( view: 'users.index', data: compact( var_name: 'users' ));
});
```

3. Styling

Pada tahap ini, dilakukan penambahan styling pada halaman dengan mengintegrasikan Tailwind CSS. Penyesuaian desain dilakukan di setiap view, termasuk pengaturan layout agar lebih rapi dan responsif.

Langkah-langkah:

1. Integrasikan Tailwind CSS ke dalam proyek Laravel.
2. Terapkan styling pada halaman utama dan beberapa komponen lain.
3. Verifikasi tampilan pada berbagai ukuran layar untuk memastikan responsivitas.



4. Active State

Menambahkan fitur “active state” pada navigasi untuk menunjukkan halaman mana yang sedang aktif. Fitur ini menandai link aktif dengan gaya tertentu, sehingga pengguna dapat melihat posisi mereka di dalam aplikasi.

Langkah-langkah:

1. Modifikasi link navigasi dengan kondisi untuk menentukan kelas aktif.
2. Terapkan gaya berbeda pada link aktif menggunakan Tailwind CSS atau CSS khusus.
3. Uji tampilan untuk memastikan active state bekerja sesuai harapan.

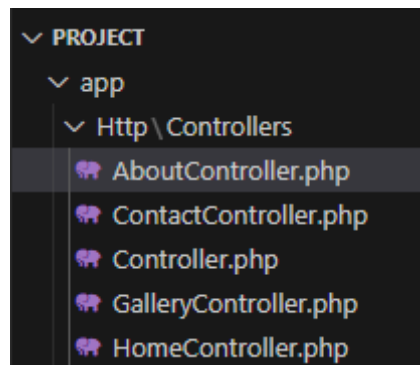
```
<!-- Current: "bg-gray-900 text-white", Default: "text-gray-300"
<a {{ $attributes }} class="{{ request()->fullUrlIs(url($href))
| {{ $slot }}"
</a>
```

5. Controller

Pada tahap ini, konsep Controller diperkenalkan untuk memisahkan logika bisnis dari view. Sebuah controller dasar dibuat untuk mengelola data dan menampilkan view sesuai kebutuhan.

Langkah-langkah:

1. Membuat controller baru menggunakan perintah php artisan make:controller.
2. Menyusun metode-metode dalam controller untuk menangani logika dan data.
3. Menghubungkan controller dengan route untuk menampilkan view yang sesuai.



6. Migration 101

Tahap ini memperkenalkan dasar-dasar migration di Laravel untuk mengelola struktur database. Migration digunakan untuk membuat tabel baru, menambahkan kolom, serta mengatur relasi antar tabel.

Langkah-langkah:

1. Membuat migration baru menggunakan perintah php artisan make:migration.
2. Mengedit file migration untuk menentukan struktur tabel.
3. Menjalankan migration menggunakan php artisan migrate untuk menerapkan perubahan ke database.

