PRAKTIKUM PEMROGRAMAN BERBASIS WEB

Untuk Memenuhi Tugas 9 P.PBW

Oleh :

Faathir Akbar Nugroho

4522210033

Kelas A

1. DB Facades

Tahap ini mempelajari penggunaan **DB Facades** untuk berinteraksi langsung dengan database tanpa menggunakan model. Teknik ini digunakan untuk query langsung seperti SELECT, INSERT, UPDATE, dan DELETE.

Langkah-langkah:

1. Gunakan DB facade untuk mengambil data dari tabel dengan query langsung.
    - Contoh: DB::table('users')->get();
2. Gunakan metode seperti insert(), update(), dan delete() untuk memanipulasi data.
3. Uji setiap operasi dengan melihat hasil perubahan pada database.

Hasil: Berhasil mengelola data secara langsung di database menggunakan DB Facades.

---

2. Model dan Fetching

Pada tahap ini, dipelajari cara membuat **Model** untuk merepresentasikan tabel database dalam Laravel. Fetching data dilakukan melalui Model untuk menjaga struktur kode yang lebih bersih.

Langkah-langkah:

1. Membuat model menggunakan perintah php artisan make:model.
2. Gunakan model untuk mengambil data dari database dengan cara:
    - Contoh: User::all(); atau User::find($id);
3. Tampilkan data yang di-fetch ke view menggunakan controller.

Hasil: Pengelolaan data lebih mudah dan terstruktur melalui model Laravel.

---

3. Kondisi Dalam Komponen

Tahap ini membahas penggunaan kondisi di dalam komponen Blade untuk mengatur tampilan dinamis berdasarkan data tertentu.

Langkah-langkah:

1. Gunakan statement Blade seperti @if, @else, dan @elseif untuk kondisi sederhana.
2. Gunakan @switch dan @case untuk kondisi lebih kompleks.
3. Uji kondisi dengan berbagai data untuk memastikan tampilan sesuai.

Hasil: Berhasil menampilkan elemen UI dinamis berdasarkan kondisi tertentu.

---

4. Fillable and Guarded

Pada tahap ini, dipelajari konsep **fillable** dan **guarded** untuk mengamankan input data ke dalam model.

Langkah-langkah:

1. Tambahkan properti $fillable di model untuk menentukan atribut yang dapat diisi.
    - Contoh: protected $fillable = ['name', 'email'];
2. Gunakan $guarded untuk menentukan atribut yang tidak boleh diisi.
    - Contoh: protected $guarded = ['is_admin'];
3. Uji keamanan dengan mencoba menyimpan data melalui model.

Hasil: Berhasil mengontrol input data ke database menggunakan fillable dan guarded.

---

5. Validasi Data

Tahap ini mempelajari validasi input data sebelum disimpan ke database menggunakan fitur **Validation** di Laravel.

Langkah-langkah:

1. Gunakan validate() di controller untuk memvalidasi input.
    - Contoh: $request->validate(['name' => 'required|max:255']);
2. Tangani validasi yang gagal dengan menampilkan pesan error ke pengguna.
3. Uji validasi dengan data yang benar dan salah.

Hasil: Berhasil melakukan validasi data untuk memastikan hanya data valid yang masuk ke database.

---

6. Route Wildcard

Pada tahap ini, dipelajari penggunaan wildcard di route untuk menangani URL dinamis.

Langkah-langkah:

1. Tambahkan wildcard di route dengan tanda {}.
    - Contoh: Route::get('/user/{id}', [UserController::class, 'show']);
2. Tangkap parameter wildcard di controller untuk diproses.
3. Tampilkan data berdasarkan parameter wildcard yang diterima.

Hasil: Berhasil menampilkan data dinamis berdasarkan parameter wildcard di URL.