**Project Membuat Aplikasi Gallery Foto ( UKK Paket III )**

**SMK Telkom Lampung**

**Step 1 – Mempersiapkan Database Migration**

1. **Membuat Project Baru**

Pertama, Anda harus memastikan bahwa Anda telah menginstal Composer, karena itu adalah alat yang diperlukan untuk menginstal Laravel. Setelah Anda memastikan Composer telah terinstal, buka terminal atau command prompt dan jalankan perintah berikut untuk membuat proyek baru:

composer create-project --prefer-dist laravel/laravel ukk\_telkom

Perintah ini akan membuat folder baru dengan nama **ukk\_telkom** dan menginstal Laravel versi terbaru di dalamnya.

1. **Konfigurasi Database di File .env**

Setelah proyek berhasil dibuat, navigasikan ke dalam folder proyek:

cd ukk\_telkom

Dalam folder proyek, Anda akan menemukan file .env. File ini digunakan untuk menyimpan konfigurasi lingkungan, termasuk detail database. Buka file .env menggunakan editor teks pilihan Anda dan cari bagian yang berhubungan dengan konfigurasi database.

Ubah baris yang sesuai untuk mencerminkan nama database, nama pengguna, dan kata sandi MySQL Anda seperti berikut:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=ukk\_telkom

DB\_USERNAME=root

DB\_PASSWORD=

Pastikan untuk mengganti DB\_USERNAME dan DB\_PASSWORD dengan nama pengguna dan kata sandi MySQL Anda. Jika Anda menggunakan environment pengembangan lokal dan belum mengatur kata sandi untuk MySQL, DB\_PASSWORD mungkin kosong.

1. **Membuat Migrasi Database**

Setelah mengkonfigurasi file .env, Anda bisa menjalankan migrasi untuk membuat tabel-tabel default yang dibutuhkan Laravel dalam database Anda. Pastikan server MySQL Anda sudah berjalan, kemudian jalankan perintah berikut dari terminal di dalam folder proyek:

php artisan migrate

Perintah ini akan membuat tabel-tabel seperti users dan password\_resets di database ukk\_telkom.

Dengan mengikuti langkah-langkah di atas, Anda telah berhasil membuat proyek baru Laravel dan mengkonfigurasi database MySQL. Anda sekarang bisa mulai mengembangkan galeri foto pada website Anda dengan Laravel. Jika Anda memiliki pertanyaan lebih lanjut atau memerlukan bantuan dengan fitur spesifik Laravel, jangan ragu untuk bertanya!

1. **Menambahkan Kolom Pada Tabel Users**

Untuk menambahkan kolom username dan alamat ke tabel users di Laravel, Anda harus membuat migration baru yang akan memodifikasi tabel tersebut. Berikut adalah langkah-langkah untuk melakukan hal tersebut:

1. Membuat File Migration Baru

Pertama, Anda perlu membuat migration baru yang bertujuan untuk menambahkan kolom baru ke tabel users. Buka terminal atau command prompt, pastikan Anda berada dalam direktori proyek Laravel Anda, dan jalankan perintah berikut:

php artisan make:migration add\_username\_and\_alamat\_to\_users\_table --table=users

Perintah ini akan membuat file migration baru dalam direktori database/migrations dengan nama yang dimulai dengan timestamp saat ini, diikuti dengan nama migration add\_username\_and\_alamat\_to\_users\_table. File ini digunakan untuk menambahkan atau mengubah skema tabel users.

1. Modifikasi File Migration

Setelah membuat file migration, buka file tersebut dalam editor teks Anda. File tersebut akan terlihat seperti ini:

***<?php***

**use** Illuminate**\**Database**\**Migrations**\**Migration**;**

**use** Illuminate**\**Database**\**Schema**\**Blueprint**;**

**use** Illuminate**\**Support**\**Facades**\**Schema**;**

class **AddUsernameAndAlamatToUsersTable** *extends* *Migration*

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

*public* function up()

    {

*// Menambahkan kolom 'username' dan 'alamat' ke tabel 'users'*

        Schema**::**table('users'**,** function (Blueprint **$table**) {

            $table**->**string('username')**->**unique()**;** *// Menambahkan kolom 'username' yang unik*

            $table**->**text('alamat')**;** *// Menambahkan kolom 'alamat'*

        })**;**

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

*public* function down()

    {

*// Menghapus kolom 'username' dan 'alamat' jika migration di-rollback*

        Schema**::**table('users'**,** function (Blueprint **$table**) {

            $table**->**dropColumn('username')**;** *// Menghapus kolom 'username'*

            $table**->**dropColumn('alamat')**;** *// Menghapus kolom 'alamat'*

        })**;**

    }

}

Dalam method up(), kita menambahkan dua baris untuk menambahkan kolom username dan alamat. Kolom username ditandai sebagai unik, yang berarti tidak boleh ada dua pengguna dengan username yang sama. Method down() mendefinisikan apa yang harus terjadi jika migration di-rollback, dalam hal ini, menghapus kolom username dan alamat.

1. Jalankan Migration

Setelah Anda memodifikasi file migration, jalankan migration tersebut dengan perintah berikut di terminal:

php artisan migrate

Perintah ini akan menerapkan perubahan ke database, menambahkan kolom username dan alamat ke tabel users.

Dengan langkah-langkah ini, Anda telah berhasil memodifikasi tabel users untuk menambahkan kolom username dan alamat. Sekarang, Anda dapat menggunakan kolom-kolom baru ini dalam aplikasi Laravel Anda.

1. **Membuat Tabel Gallery Album**

Untuk membuat tabel gallery\_album dengan kolom album\_id, nama\_album, deskripsi, user\_id yang merupakan foreign key dari tabel users, dan timestamp, Anda perlu membuat migration baru di Laravel. Berikut adalah langkah-langkah untuk melakukannya:

1. Membuat Migration

Pertama, buat migration untuk tabel gallery\_album dengan menjalankan perintah berikut di terminal atau command prompt Anda:

php artisan make:migration create\_gallery\_album\_table

Perintah ini akan membuat file migration baru di direktori database/migrations.

1. Menulis Schema Tabel

Selanjutnya, buka file migration yang baru saja Anda buat. Anda akan menemukan dua method dalam file tersebut: up() untuk menentukan pembuatan tabel, dan down() untuk menghapus tabel jika diperlukan. Modifikasi file tersebut sehingga terlihat seperti ini:

***<?php***

**use** Illuminate**\**Database**\**Migrations**\**Migration**;**

**use** Illuminate**\**Database**\**Schema**\**Blueprint**;**

**use** Illuminate**\**Support**\**Facades**\**Schema**;**

class **CreateGalleryAlbumTable** *extends* *Migration*

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

*public* function up()

    {

        Schema**::**create('gallery\_album'**,** function (Blueprint **$table**) {

            $table**->**id('album\_id')**;** *// Membuat primary key 'album\_id'*

            $table**->**string('nama\_album')**;** *// Membuat kolom 'nama\_album'*

            $table**->**text('deskripsi')**->**nullable()**;** *// Membuat kolom 'deskripsi' yang bisa null*

            $table**->**foreignId('user\_id')**->**constrained()**->**onDelete('cascade')**;** *// Membuat foreign key 'user\_id' yang merujuk ke 'id' di tabel 'users' dengan constraint onDelete cascade*

            $table**->**timestamps()**;** *// Membuat kolom 'created\_at' dan 'updated\_at'*

        })**;**

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

*public* function down()

    {

        Schema**::**dropIfExists('gallery\_album')**;** *// Menghapus tabel 'gallery\_album' jika migration di-rollback*

    }

}

Penjelasan Dalam kode ini:

* album\_id diatur sebagai primary key.
* nama\_album adalah kolom untuk menyimpan nama album.
* deskripsi adalah kolom untuk deskripsi album, yang boleh null.
* user\_id adalah foreign key yang merujuk ke kolom id di tabel users. Jika pengguna dihapus, maka album terkait juga akan dihapus (onDelete('cascade')).
* timestamps() menambahkan dua kolom: created\_at dan updated\_at.

1. Jalankan Migration

Setelah Anda selesai menulis schema tabel di migration file, simpan file tersebut dan jalankan migration dengan perintah berikut:

php artisan migrate

Perintah ini akan menerapkan perubahan ke database Anda, membuat tabel gallery\_album dengan kolom-kolom yang telah Anda definisikan.

Dengan mengikuti langkah-langkah ini, Anda telah berhasil membuat tabel gallery\_album di database Laravel Anda, yang siap untuk digunakan dalam proyek galeri foto Anda.

1. **Membuat Tabel Gallery Foto**

Untuk membuat tabel gallery\_foto dengan kolom-kolom yang Anda sebutkan, termasuk foto\_id, judul\_foto, deskripsi\_foto, lokasi\_file, foreign key album\_id yang merujuk ke gallery\_album, user\_id yang merujuk ke users, serta timestamp, Anda harus membuat migration baru di Laravel. Ikuti langkah-langkah berikut:

1. Membuat Migration

Buat migration baru untuk tabel gallery\_foto dengan menjalankan perintah berikut di terminal atau command prompt Anda:

php artisan make:migration create\_gallery\_foto\_table

Perintah ini akan membuat file migration baru di direktori database/migrations.

1. Menulis Schema Tabel

Buka file migration yang baru saja dibuat. Dalam file tersebut, Anda akan mengisi method up() untuk mendefinisikan struktur tabel, dan down() untuk menghapus tabel jika migration di-rollback. Modifikasi file tersebut seperti contoh berikut:

***<?php***

**use** Illuminate**\**Database**\**Migrations**\**Migration**;**

**use** Illuminate**\**Database**\**Schema**\**Blueprint**;**

**use** Illuminate**\**Support**\**Facades**\**Schema**;**

class **CreateGalleryFotoTable** *extends* *Migration*

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

*public* function up()

    {

        Schema**::**create('gallery\_foto'**,** function (Blueprint **$table**) {

            $table**->**id('foto\_id')**;** *// Membuat primary key 'foto\_id'*

            $table**->**string('judul\_foto')**;** *// Membuat kolom 'judul\_foto'*

            $table**->**text('deskripsi\_foto')**->**nullable()**;** *// Membuat kolom 'deskripsi\_foto' yang bisa null*

            $table**->**string('lokasi\_file')**;** *// Membuat kolom 'lokasi\_file' untuk menyimpan path file*

            $table**->**unsignedBigInteger('album\_id')**;** *// Gunakan tipe yang sama dengan 'album\_id' di 'gallery\_album'*

            $table**->**foreign('album\_id')**->**references('album\_id')**->**on('gallery\_album')**->**onDelete('cascade')**;** *// Membuat foreign key 'album\_id' yang merujuk ke tabel 'gallery\_album'*

            $table**->**foreignId('user\_id')**->**constrained()**->**onDelete('cascade')**;** *// Membuat foreign key 'user\_id' yang merujuk ke tabel 'users'*

            $table**->**timestamps()**;** *// Membuat kolom 'created\_at' dan 'updated\_at'*

        })**;**

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

*public* function down()

    {

        Schema**::**dropIfExists('gallery\_foto')**;** *// Menghapus tabel 'gallery\_foto' jika migration di-rollback*

    }

}

Penjelasan Dalam kode di atas:

* foto\_id diatur sebagai primary key.
* judul\_foto, deskripsi\_foto, dan lokasi\_file adalah kolom untuk informasi foto.
* album\_id dan user\_id adalah foreign keys yang merujuk ke tabel gallery\_album dan users, masing-masing. onDelete('cascade') menandakan bahwa jika album atau user terkait dihapus, maka foto juga akan dihapus.
* timestamps() akan otomatis menambahkan dua kolom: created\_at dan updated\_at.

1. Jalankan Migration

Simpan perubahan pada file migration dan jalankan migration dengan perintah berikut untuk menerapkan perubahan ke database Anda:

php artisan migrate

Setelah menjalankan perintah ini, tabel gallery\_foto akan dibuat di database Anda dengan semua kolom yang ditentukan, siap untuk digunakan dalam proyek galeri foto Anda.

1. **Membuat Tabel Gellery Komentar Foto**

Untuk membuat tabel gallery\_komentarfoto dengan kolom-kolom yang Anda sebutkan, termasuk komentar\_id, foto\_id yang merujuk ke gallery\_foto, user\_id yang merujuk ke users, isi\_komentar, serta timestamp, Anda harus membuat migration baru di Laravel. Ikuti langkah-langkah berikut:

1. Membuat Migration

Pertama, buat migration baru untuk tabel gallery\_komentarfoto dengan menjalankan perintah berikut di terminal atau command prompt Anda:

php artisan make:migration create\_gallery\_komentarfoto\_table

Perintah ini akan membuat file migration baru di direktori database/migrations.

1. Menulis Schema Tabel

Buka file migration yang baru saja dibuat. Anda akan mengisi method up() untuk mendefinisikan struktur tabel dan down() untuk menghapus tabel jika migration di-rollback. Modifikasi file tersebut seperti contoh berikut:

***<?php***

**use** Illuminate**\**Database**\**Migrations**\**Migration**;**

**use** Illuminate**\**Database**\**Schema**\**Blueprint**;**

**use** Illuminate**\**Support**\**Facades**\**Schema**;**

class **CreateGalleryKomentarfotoTable** *extends* *Migration*

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

*public* function up()

    {

        Schema**::**create('gallery\_komentarfoto'**,** function (Blueprint **$table**) {

            $table**->**id('komentar\_id')**;** *// Membuat primary key 'komentar\_id'*

            $table**->**unsignedBigInteger('foto\_id')**;** *// Gunakan tipe yang sama dengan 'foto\_id' di 'gallery\_foto'*

            $table**->**foreign('foto\_id')**->**references('foto\_id')**->**on('gallery\_foto')**->**onDelete('cascade')**;** *// Foreign key 'foto\_id' ke 'gallery\_foto' Secara eksplisit merujuk ke 'foto\_id' di 'gallery\_foto'*

            $table**->**foreignId('user\_id')**->**constrained()**->**onDelete('cascade')**;** *// Foreign key 'user\_id' ke 'users'*

            $table**->**text('isi\_komentar')**;** *// Kolom untuk isi komentar*

            $table**->**timestamps()**;** *// Membuat kolom 'created\_at' dan 'updated\_at' secara otomatis*

        })**;**

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

*public* function down()

    {

        Schema**::**dropIfExists('gallery\_komentarfoto')**;** *// Menghapus tabel 'gallery\_komentarfoto' jika migration di-rollback*

    }

}

Penjelasan Dalam kode di atas:

* komentar\_id diatur sebagai primary key.
* foto\_id dan user\_id adalah foreign keys yang merujuk ke tabel gallery\_foto dan users, masing-masing. Menggunakan constrained() secara otomatis mengatur referensi ke kolom primary di tabel terkait dan onDelete('cascade') berarti jika foto atau pengguna dihapus, maka komentar terkait juga akan dihapus.
* isi\_komentar adalah kolom untuk menyimpan teks komentar.
* timestamps() akan otomatis menambahkan dua kolom: created\_at dan updated\_at.

1. Jalankan Migration

Simpan perubahan pada file migration dan jalankan migration dengan perintah berikut untuk menerapkan perubahan ke database Anda:

php artisan migrate

Setelah menjalankan perintah ini, tabel gallery\_komentarfoto akan dibuat di database Anda dengan semua kolom yang ditentukan, siap untuk digunakan dalam proyek galeri foto Anda.

1. **Membuat Tabel Gallery Like Foto**

Untuk membuat tabel gallery\_likefoto dengan kolom like\_id, foto\_id yang merujuk ke gallery\_foto berdasarkan foto\_id, user\_id yang merujuk ke users berdasarkan id, dan menambahkan timestamp, Anda harus membuat migration baru di Laravel. Ikuti langkah-langkah berikut:

1. Membuat Migration

Buat migration baru untuk tabel gallery\_likefoto dengan menjalankan perintah berikut di terminal atau command prompt Anda:

php artisan make:migration create\_gallery\_likefoto\_table

Perintah ini akan membuat file migration baru di direktori database/migrations.

1. Menulis Schema Tabel

Buka file migration yang baru saja dibuat. Anda akan mengisi method up() untuk mendefinisikan struktur tabel dan down() untuk menghapus tabel jika migration di-rollback. Modifikasi file tersebut seperti contoh berikut:

***<?php***

**use** Illuminate**\**Database**\**Migrations**\**Migration**;**

**use** Illuminate**\**Database**\**Schema**\**Blueprint**;**

**use** Illuminate**\**Support**\**Facades**\**Schema**;**

class **CreateGalleryLikefotoTable** *extends* *Migration*

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

*public* function up()

    {

        Schema**::**create('gallery\_likefoto'**,** function (Blueprint **$table**) {

            $table**->**id('like\_id')**;** *// Membuat primary key 'like\_id'*

            $table**->**unsignedBigInteger('foto\_id')**;** *// Menentukan 'foto\_id' sesuai dengan tipe di 'gallery\_foto'*

            $table**->**foreign('foto\_id')**->**references('foto\_id')**->**on('gallery\_foto')**->**onDelete('cascade')**;** *// Membuat foreign key 'foto\_id' yang merujuk ke 'gallery\_foto'*

            $table**->**foreignId('user\_id')**->**constrained()**->**onDelete('cascade')**;** *// Membuat foreign key 'user\_id' yang merujuk ke 'users'*

            $table**->**timestamps()**;** *// Membuat kolom 'created\_at' dan 'updated\_at' secara otomatis*

        })**;**

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

*public* function down()

    {

        Schema**::**dropIfExists('gallery\_likefoto')**;**

    }

}

Penjelasan Dalam kode di atas:

* like\_id diatur sebagai primary key.
* foto\_id adalah foreign key yang merujuk ke gallery\_foto pada kolom foto\_id. Kita secara eksplisit menetapkan bahwa foto\_id harus merujuk ke kolom foto\_id di tabel gallery\_foto.
* user\_id adalah foreign key yang merujuk ke tabel users, mengasumsikan kolom primary key adalah id secara default.
* timestamps() akan otomatis menambahkan dua kolom: created\_at dan updated\_at.

1. Menjalankan Migration

Simpan perubahan pada file migration dan jalankan migration dengan perintah berikut untuk menerapkan perubahan ke database Anda:

php artisan migrate

Setelah menjalankan perintah ini, tabel gallery\_likefoto akan dibuat di database Anda dengan semua kolom yang ditentukan, siap untuk digunakan dalam proyek galeri foto Anda.