**1. APA FUNGSI DARI MIGRATIONS, MODELS, CONTROLLER, VIEWS, ROUTES**

1. Migrations

Migrations adalah cara untuk mengelola skema database dalam Laravel. Fungsinya adalah:

Membuat dan Mengelola Tabel: Migrations memungkinkan pengembang untuk membuat, mengubah, dan menghapus tabel dalam database dengan cara yang terstruktur dan mudah dipahami.

Versi Kontrol Database: Dengan menggunakan migrations, pengembang dapat melacak perubahan pada skema database seiring waktu, mirip dengan sistem kontrol versi untuk kode.

2. Models

Models adalah representasi dari tabel dalam database. Fungsinya adalah:

Interaksi dengan Database: Models bertanggung jawab untuk berinteraksi dengan database, baik untuk mengambil data (read), menyimpan data (create), memperbarui data (update), atau menghapus data (delete).

Logika Bisnis: Models dapat menyimpan logika bisnis yang terkait dengan data, seperti validasi dan relasi antar model.

Abstraksi: Models memberikan cara yang lebih terstruktur untuk bekerja dengan data, mengabstraksi detail implementasi database.

3. Controllers

Controllers adalah penghubung antara models dan views. Fungsinya adalah:

Mengelola Permintaan: Controllers menangani permintaan HTTP yang masuk dan menentukan apa yang harus dilakukan dengan permintaan tersebut.

Mendapatkan Data: Controllers berinteraksi dengan models untuk mendapatkan data yang diperlukan untuk ditampilkan.

Mengembalikan Respons: Setelah mendapatkan data, controllers akan mengembalikannya ke view yang sesuai untuk ditampilkan kepada pengguna.

4. Views

Views adalah komponen yang bertanggung jawab untuk menampilkan data kepada pengguna. Fungsinya adalah:

Antarmuka Pengguna: Views berisi HTML dan kode terkait yang digunakan untuk menampilkan antarmuka pengguna.

Menerima Data dari Controllers: Views menerima data yang dikirim oleh controllers dan menggunakannya untuk menghasilkan tampilan yang dinamis.

Template: Views dapat menggunakan template untuk memisahkan logika presentasi dari logika aplikasi.

5. Routes

Routes adalah cara untuk mendefinisikan URL yang dapat diakses dalam aplikasi. Fungsinya adalah:

Menentukan URL: Routes mendefinisikan URL yang akan diakses oleh pengguna dan mengaitkannya dengan controllers yang sesuai.

Mengelola Permintaan: Routes menentukan bagaimana permintaan HTTP (GET, POST, dll.) akan ditangani oleh aplikasi.

Organisasi: Routes membantu dalam mengorganisir dan mendokumentasikan endpoint aplikasi dengan jelas.

2. TULISKAN KODE YANG BIASA DIKETIKAN PADA TERMINAL UNTUK PEMBUATAN CONTROLLER YANG SUDAH TERDAPAT FUNCTION YANG SIAP DIGUNAKAN

Dalam pengembangan aplikasi menggunakan framework Laravel, pembuatan controller dengan opsi `--resource` adalah salah satu cara efisien untuk menghasilkan struktur controller yang sudah dilengkapi dengan fungsi-fungsi standar CRUD (Create, Read, Update, Delete). Command ini dijalankan melalui terminal dengan sintaks `php artisan make:controller NamaController --resource`. Ketika perintah ini dieksekusi, Laravel secara otomatis membuat file controller di direktori `app/Http/Controllers`, lengkap dengan metode seperti `index()` untuk menampilkan daftar data, `create()` untuk menampilkan form penambahan data baru, `store()` untuk menyimpan data baru ke database, `show()` untuk menampilkan detail data tertentu, `edit()` untuk menampilkan form pengeditan data, `update()` untuk memperbarui data di database, dan `destroy()` untuk menghapus data.

Penggunaan resource controller ini sangat mempermudah developer karena fungsi-fungsi yang dihasilkan mengikuti pola RESTful, sehingga mempermudah pengaturan rute dan konsistensi dalam pengembangan aplikasi. Dengan resource controller, pengembang cukup menghubungkan rute menggunakan `Route::resource()` di file `routes/web.php` atau `routes/api.php`, dan seluruh metode CRUD dapat langsung diakses tanpa perlu mendefinisikan satu per satu rutenya. Hal ini menjadikan pengembangan lebih cepat, terorganisir, dan sesuai dengan standar industri.

3. APA PERBEDAAN DARI MIGRATE DENGAN MIGRATE FRESH

Dalam Laravel, terdapat dua perintah utama terkait migrasi database, yaitu `migrate` dan `migrate:fresh`. Berikut adalah perbedaan mendasar antara keduanya:

1. `php artisan migrate`

- Fungsi Utama: Menjalankan migrasi yang belum diterapkan ke database.

- Cara Kerja: Laravel akan memeriksa tabel `migrations` di database untuk melihat daftar migrasi yang sudah dijalankan. Jika ada file migrasi baru yang belum diterapkan, perintah ini akan menjalankannya.

- Efek pada Data: Tidak menghapus data yang ada di database, hanya menambahkan tabel, kolom, atau perubahan lain sesuai file migrasi baru.

- Kegunaan: Digunakan saat ingin menerapkan perubahan tambahan tanpa mengganggu data yang sudah ada.

2. `php artisan migrate:fresh`

- Fungsi Utama: Menghapus semua tabel di database dan menjalankan ulang semua file migrasi dari awal.

- Cara Kerja: Perintah ini akan:

1. Menghapus semua tabel di database, termasuk tabel sistem seperti `migrations`.

2. Menjalankan semua file migrasi dari awal, sehingga database kembali ke kondisi awal sesuai dengan file migrasi.

- Efek pada Data: Semua data yang ada di database akan hilang karena semua tabel dihapus terlebih dahulu.

- Kegunaan: Sangat berguna saat ingin memulai ulang database dari awal, misalnya dalam pengembangan atau pengujian aplikasi.

Contoh Penggunaan

- `migrate`:

- Ketika ada penambahan file migrasi baru untuk fitur tambahan dan Anda tidak ingin kehilangan data lama.

- `migrate:fresh`:

- Saat terjadi kesalahan besar pada struktur database selama pengembangan, atau Anda ingin menghapus semua data untuk memulai ulang dari awal.

Catatan Penting

- Gunakan `migrate:fresh` dengan hati-hati, terutama di lingkungan produksi, karena perintah ini akan menghapus semua data di database.

- Sebelum menggunakan `migrate:fresh`, pastikan data penting telah dicadangkan jika diperlukan untuk referensi atau pengujian.