



PENGEMBANGAN DAN PENYELENGGARAAN  
PEMBELAJARAN DIGITAL (P3D)



# Modul Pembelajaran **SISTEM INFORMASI**

Pemrograman Web Framework



Disusun oleh : Wicaksono Yuli Sulistyio



# Modul Ajar: Database di Laravel

## Pengenalan Database dalam Laravel

Dalam pengembangan aplikasi web, manajemen data merupakan salah satu aspek yang paling krusial. Data dapat berfungsi sebagai inti dari aplikasi, mulai dari informasi pengguna, produk, hingga transaksi. Laravel, sebagai salah satu framework PHP yang paling populer, menawarkan solusi yang efisien dan elegan untuk mengelola database melalui fitur-fitur yang kuat seperti Eloquent ORM dan Query Builder. Dengan memanfaatkan kemampuan ini, pengembang dapat berfokus pada logika bisnis dan pengalaman pengguna, alih-alih terjebak dalam kerumitan pengelolaan data.

Laravel mendukung berbagai sistem database, termasuk MySQL, PostgreSQL, SQLite, dan SQL Server. Keberagaman ini memberikan fleksibilitas kepada pengembang untuk memilih solusi yang paling sesuai dengan kebutuhan aplikasi mereka. Selain itu, Laravel memudahkan integrasi dengan database melalui file konfigurasi dan penggunaan migration, yang akan dijelaskan lebih lanjut dalam modul ini.

## Konfigurasi Database

Sebelum memulai interaksi dengan database, langkah pertama yang harus dilakukan adalah mengonfigurasi koneksi database di file `.env`. File ini berisi parameter yang diperlukan untuk menghubungkan aplikasi Laravel dengan database yang dipilih. Misalnya, jika Anda menggunakan MySQL, Anda perlu menambahkan detail seperti nama database, username, dan password. Contoh konfigurasi untuk MySQL dalam file `.env` adalah sebagai berikut:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nama_database
DB_USERNAME=username_database
DB_PASSWORD=password_database
```

Setelah konfigurasi ini selesai, Laravel akan menggunakan informasi ini setiap kali melakukan operasi database. Ini adalah langkah penting yang memastikan aplikasi Anda dapat berkomunikasi dengan database dengan benar.

## Migrasi Database

Migrasi adalah fitur yang sangat berguna dalam Laravel, yang memungkinkan pengembang untuk mendefinisikan struktur tabel dalam database menggunakan kode PHP. Ini menghilangkan kebutuhan untuk menulis SQL secara manual dan memastikan bahwa semua anggota tim pengembangan memiliki akses ke struktur database yang konsisten. Dengan menggunakan



migrasi, Anda dapat membuat, mengubah, dan menghapus tabel dengan cara yang terorganisir dan dapat dipulihkan.

Untuk membuat migrasi baru, Anda dapat menggunakan perintah Artisan berikut:

```
bash  
php artisan make:migration create_users_table
```

Perintah ini akan membuat file migrasi baru di direktori `database/migrations`. File ini berisi dua metode: `up()` dan `down()`. Metode `up()` digunakan untuk mendefinisikan tabel dan kolom yang akan dibuat, sementara `down()` digunakan untuk menghapus tabel tersebut jika diperlukan. Sebagai contoh, berikut adalah cara mendefinisikan tabel `users`:

```
php  
  
use Illuminate\Database\Migrations\Migration;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Support\Facades\Schema;  
  
class CreateUsersTable extends Migration  
{  
    public function up()  
    {  
        Schema::create('users', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->string('email')->unique();  
            $table->string('password');  
            $table->timestamps();  
        });  
    }  
  
    public function down()  
    {  
        Schema::dropIfExists('users');  
    }  
}
```

Setelah mendefinisikan migrasi, Anda dapat menjalankannya dengan perintah:

```
bash  
php artisan migrate
```

Perintah ini akan membuat tabel `users` di database Anda. Jika Anda perlu mengubah struktur tabel, Anda cukup membuat migrasi baru, yang akan menjaga sejarah perubahan struktur database Anda.

## Eloquent ORM

Eloquent adalah Object-Relational Mapping (ORM) yang digunakan oleh Laravel, yang menyediakan cara yang intuitif dan sederhana untuk berinteraksi dengan database. Setiap tabel dalam database memiliki model Eloquent yang mewakilinya, dan model ini memudahkan pengembang untuk melakukan operasi CRUD (Create, Read, Update, Delete) tanpa harus menulis query SQL secara manual.

Sebagai contoh, untuk membuat model `User`, Anda dapat menggunakan perintah:

```
bash
```

```
php artisan make:model User
```

Model `User` ini akan terletak di `app/Models/User.php` dan dapat digunakan untuk berinteraksi dengan tabel `users` di database. Contoh kode untuk mengambil semua pengguna menggunakan Eloquent adalah:

```
php
```

```
$users = User::all();
```

Eloquent juga mendukung berbagai operasi yang lebih kompleks, seperti relasi antar model. Misalnya, jika Anda memiliki model `Post` yang memiliki relasi satu-ke-banyak dengan model `User`, Anda dapat mendefinisikan relasi tersebut di dalam model `User`:

```
php
```

```
public function posts()
{
    return $this->hasMany(Post::class);
}
```

Dengan relasi ini, Anda dapat dengan mudah mengambil semua post yang ditulis oleh pengguna tertentu:

```
php
```

```
$user = User::find(1);
$posts = $user->posts;
```

## Query Builder

Selain Eloquent, Laravel juga menyediakan **Query Builder**, yang memungkinkan pengembang untuk membangun query database menggunakan sintaksis PHP yang lebih dekat dengan SQL. Ini memberikan fleksibilitas untuk menulis query yang lebih rumit yang mungkin tidak mudah atau tidak efisien dilakukan dengan Eloquent.

Contoh penggunaan Query Builder untuk mengambil data pengguna dengan kondisi tertentu adalah sebagai berikut:



php

```
$users = DB::table('users')->where('active', 1)->get();
```

Query Builder mendukung berbagai operasi database, seperti penggabungan (join), pengelompokan (grouping), dan pemesanan (ordering). Jika Anda ingin mengurutkan pengguna berdasarkan nama, Anda dapat menggunakan:

php

```
$users = DB::table('users')->orderBy('name', 'asc')->get();
```

## Kesimpulan

Database adalah komponen penting dalam pengembangan aplikasi web, dan Laravel menyediakan berbagai alat dan fitur yang kuat untuk mengelola database dengan efisien. Dari konfigurasi koneksi, sistem migrasi yang memudahkan pengelolaan struktur database, hingga penggunaan Eloquent dan Query Builder untuk interaksi data, Laravel menawarkan pendekatan yang elegan dan terstruktur. Dengan pemahaman yang kuat tentang cara kerja database dalam Laravel, pengembang dapat membangun aplikasi yang kompleks, terstruktur, dan mudah dipelihara.