

Analisis Proyek Laravel – Tema: Data Hewan

1. Deskripsi Proyek

Proyek ini merupakan sistem informasi berbasis web yang dibangun menggunakan **Laravel 12** dan **MariaDB** sebagai sistem basis datanya. Tema yang diangkat adalah **Data Hewan**, dengan tujuan memudahkan pengelolaan data hewan, jenis, lokasi kandang, perawatan, dan riwayat kesehatannya.

2. Entitas dan Relasi

◇ Entitas yang Dirancang:

- 1. **JenisHewan** – Menyimpan data jenis seperti Mamalia, Reptil, Burung.
- 2. **Kandang** – Menyimpan data kandang dan lokasi.
- 3. **Hewan** – Data utama hewan (nama, umur, jenis, kandang).
- 4. **Perawatan** – Mencatat tindakan perawatan terhadap hewan.
- 5. **RiwayatKesehatan** – Mencatat penyakit dan pengobatan hewan.

Relasi Antar Entitas:

Entitas	Relasi
JenisHewan → Hewan	One to Many (jenis_hewan_id)
Kandang → Hewan	One to Many (kandang_id)
Hewan → Perawatan	One to Many (hewan_id)
Hewan → RiwayatKesehatan	One to Many (hewan_id)

3. Teknologi dan Tools

Komponen	Teknologi Digunakan
Framework	Laravel 12.x
Database	MariaDB
PHP Version	8.3.x
Frontend Admin	Filament (jika diimplementasikan)
Package Manager	Composer
Proses Manual	DCM (tanpa php artisan)

4. Masalah yang Dihadapi

1. Tabel Tidak Ditemukan (**jenis_hewans**, **hewans**)

Laravel secara otomatis mengasumsikan bahwa nama tabel jamak (plural), contohnya:

- **JenisHewan** → **jenis_hewans**
- **Hewan** → **hewans**

Namun, tabel yang dibuat manual adalah **jenis_hewan** dan **hewan**, sehingga terjadi error:

```
SQLSTATE[42S02]: Table 'hewans' doesn't exist
```

Solusi:

Menambahkan properti **\$table** secara eksplisit pada model:

```
protected $table = 'hewan';  
protected $table = 'jenis_hewan';
```

5. Langkah Teknis yang Dilakukan

Struktur Implementasi:

- **Model** dibuat manual di folder **app/Models/**
- **Migration** dibuat manual dengan nama timestamp yang urut
- **Seeder** diisi data dummy dan dipanggil melalui **DatabaseSeeder**
- **Migrasi & Seed** dijalankan dengan **migrate:fresh --seed**

Validasi Relasi:

Semua foreign key menggunakan pendekatan:

```
$table->foreignId('xxx_id')->constrained('nama_tabel')->onDelete('cascade');
```

6. Hasil Akhir Sistem

Setelah seluruh struktur diperbaiki:

- Database berhasil dimigrasi dan diisi data awal
- Data hewan dapat dikelola secara lengkap (jenis, kandang, perawatan, kesehatan)
- Sistem siap digunakan untuk diintegrasikan dengan frontend admin seperti **Filament** atau custom frontend

7. Kesimpulan

Proyek Laravel "Data Hewan" ini menunjukkan pentingnya pemahaman relasi antar entitas, konvensi penamaan Laravel, serta bagaimana menangani error secara sistematis. Dengan pendekatan manual (DCM), developer dapat memahami struktur internal Laravel lebih dalam, walaupun membutuhkan ketelitian lebih besar.