

# Laporan Modul 7: Eloquent Relationship & Pagination

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** Maila Aziza

**NIM:** 2024573010024 **Kelas:** TI-2C

---

## Abstrak

Eloquent relationship menyediakan cara yang mudah untuk mendefinisikan hubungan antara tabel database dan model di Laravel. Pagination adalah teknik untuk mengatur konten di website dengan membaginya menjadi beberapa halaman berurutan, biasanya ditandai dengan nomor seperti 1, 2, 3, atau tombol "Next" dan "Previous". Biasanya, nomor ini muncul di atas atau bawah halaman situs. Tujuannya? Agar pengunjung mudah bernavigasi dan situs tidak menampilkan semua data di satu tempat.

---

## 1. Dasar Teori

Eloquent relationships merupakan method yang didefinisikan di dalam Model dan digunakan untuk menghubungkan antar table yang saling berhubungan. Dengan menggunakan fitur eloquent relationships maka dapat lebih mudah dalam melakukan maintenance. Pagination merupakan adalah sebuah fitur web yang digunakan untuk membatasi tampilan data agar tidak terlalu panjang dan lebih rapi. Penggunaan pagination banyak dipergunakan digunakan untuk menampilkan data dalam jumlah banyak, sehingga dapat dipisah/dipilih berapa data yang akan ditampilkan terlebih dahulu sesuai dengan urutan akan kriteria pencarian tertentu.

---

## 2. Langkah-Langkah Praktikum

### 2.1 Praktikum 1 – Eloquent ORM Relationships: One-to-One, One-to-Many, Many-to-Many

#### 1. Buat dan buka proyek laravel

```
laravel new complex-relationships
cd complex-relationships
code .
```

- Konfigurasi MySQL:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=eloquentrelation_db
DB_USERNAME=<username database anda>
DB_PASSWORD=<password database anda jika ada>
```

- `php artisan config:clear`

## 2. Buat migrasi untuk skema database

```
php artisan make:migration create_profiles_table
php artisan make:migration create_posts_table
php artisan make:migration create_tags_table
php artisan make:migration create_post_tag_table
```

- `php artisan migrate`

## 3. Mendefinisikan model eloquent

```
php artisan make:model Profile
php artisan make:model Post
php artisan make:model Tag
```

- `php artisan make:seeder DatabaseSeeder`
- `php artisan db:seed`

## 5. Membuat controller

```
php artisan make:controller UserController
php artisan make:controller PostController
```

## 6. Mendefinisikan web routes

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UserController;
use App\Http\Controllers\PostController;

// Routes untuk User
Route::get('/users', [UserController::class, 'index'])->name('users.index');
Route::get('/users/{user}', [UserController::class, 'show'])->name('users.show');

// Routes untuk Post
Route::get('/posts', [PostController::class, 'index'])->name('posts.index');
Route::get('/posts/{post}', [PostController::class, 'show'])->name('posts.show');
```

## 7. Membuat views menggunakan bootstrap

- ## 8. Testing aplikasi `php artisan serve` Kunjungi `http://127.0.0.1:8000/users` atau `http://127.0.0.1:8000/posts`

Hasil:

The screenshot displays a web application interface for 'Laravel 12 Complex Relationships'. It features two tabs: 'Users' and 'Posts'. The 'Users' tab is active, showing a list of users with their names and email addresses. The 'Posts' tab is also visible, showing a list of posts with titles and authors.

## Laravel 12 Complex Relationships

Users List

<a href="#">Laurine Gleichner Jr.</a> (kmiller@example.net)
<a href="#">Emily Jaskolski</a> (cameron21@example.net)
<a href="#">Callie Kiehn</a> (theresa94@example.com)
<a href="#">Bennie Wintheiser</a> (bernier.samson@example.com)
<a href="#">Maximillian Goldner</a> (lauretta.spencer@example.net)
<a href="#">Dr. Nick Breitenberg</a> (adrienne.jones@example.com)
<a href="#">Nikko Walter</a> (keebler.antoINETte@example.net)
<a href="#">Hazle McClure</a> (watson.gutmann@example.org)
<a href="#">Ruben Waters</a> (goldner@example.org)

## Laravel 12 Complex Relationships

All Posts

<a href="#">Judul Post untuk user 1</a> by Laurine Gleichner Jr.
<a href="#">Judul Post untuk user 2</a> by Emily Jaskolski
<a href="#">Judul Post untuk user 3</a> by Callie Kiehn
<a href="#">Judul Post untuk user 4</a> by Bennie Wintheiser
<a href="#">Judul Post untuk user 5</a> by Maximillian Goldner
<a href="#">Judul Post untuk user 6</a> by Dr. Nick Breitenberg
<a href="#">Judul Post untuk user 7</a> by Nikko Walter
<a href="#">Judul Post untuk user 8</a> by Hazle McClure
<a href="#">Judul Post untuk user 9</a> by Ruben Waters

## 2.2 Praktikum 2 – Paginasi dengan Eloquent ORM

### 1. Membuat proyek laravel baru

```
laravel new productpagination  
cd productpagination  
code .
```

- Install dependency MySQL: `composer require doctrine/dbal`

- Konfigurasi MySQL

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=pagination_db
DB_USERNAME=<username database anda>
DB_PASSWORD=<password database anda jika ada>
```

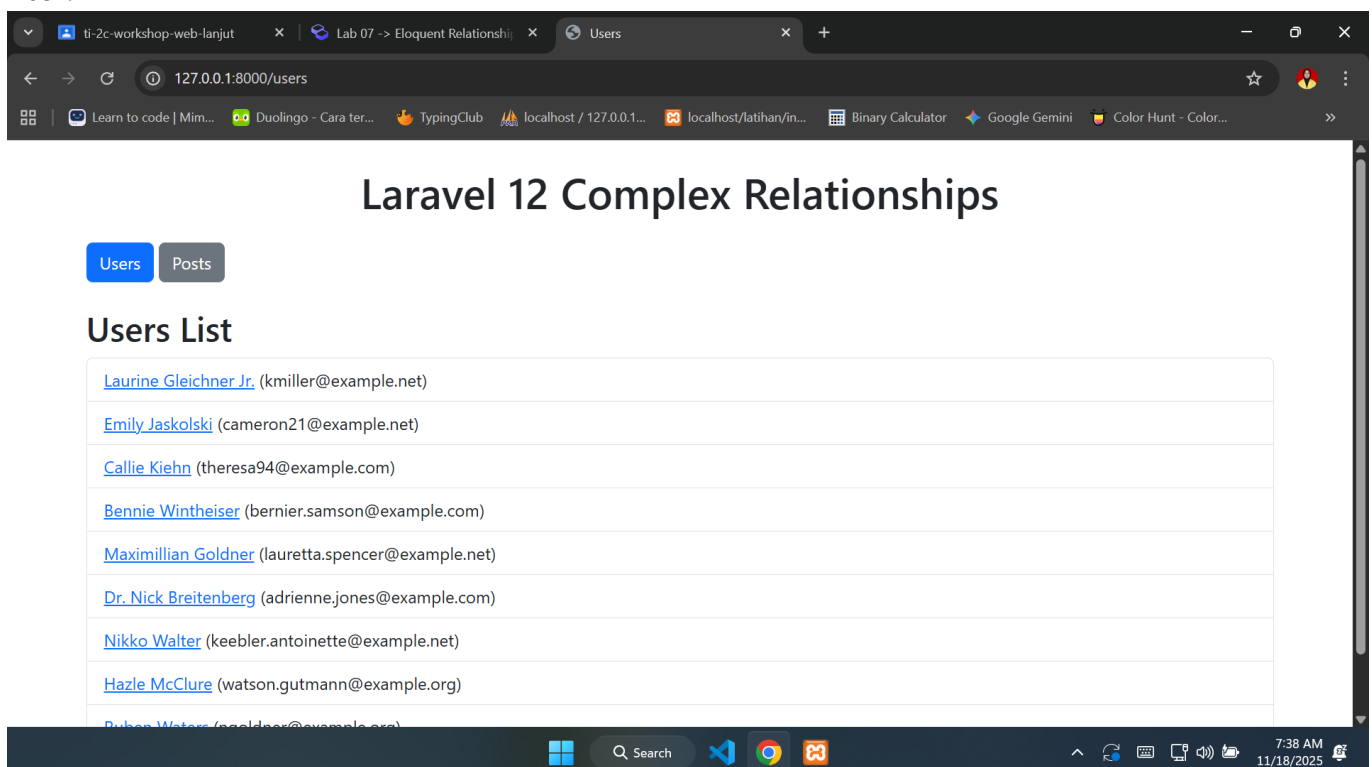
- Bersihkan config cache `php artisan config:clear`
2. Membuat model dan migrasi product `php artisan make:model Product -m`
  3. `php artisan migrate`
  4. `php artisan make:seeder ProductSeeder`
  5. `php artisan make:factory ProductFactory --model=Product`
  6. `php artisan db:seed`
  7. Buat Controller untuk paginasi `php artisan make:controller ProductController`
  8. Mendefinisikan route

```
use App\Http\Controllers\ProductController;

Route::get('/products', [ProductController::class, 'index'])-
>name('products.index');
```

9. Membuat View untuk Daftar Produk dengan Paginasi
10. Menjalankan dan menguji `php artisan serve` `http://localhost:8000/products`

Hasil:



### 3. Hasil dan Pembahasan

modul ini juga mengajarkan implementasi Pagination yang merupakan teknik penting untuk mengelola penampilan data dalam jumlah besar. Pagination tidak hanya meningkatkan pengalaman pengguna dengan menghindari penampilan data yang berlebihan dalam satu halaman, tetapi juga meningkatkan performa aplikasi dengan memuat hanya sebagian data yang diperlukan. Dalam praktikum pagination, developer belajar membuat sistem pagination lengkap mulai dari model Product, factory untuk data dummy, controller dengan method paginate(), hingga view yang menampilkan navigasi halaman secara otomatis menggunakan Tailwind CSS. Kedua konsep ini—Eloquent Relationship dan Pagination—merupakan keterampilan esensial yang sangat berguna dalam pengembangan aplikasi Laravel yang skalabel dan efisien.

---

### 4. Kesimpulan

Dari praktikum ini bisa disimpulkan Modul ini memberikan pemahaman komprehensif tentang dua konsep fundamental dalam pengembangan aplikasi Laravel, yaitu Eloquent Relationship dan Pagination. Eloquent Relationship memungkinkan developer untuk mendefinisikan hubungan antar tabel database dengan cara yang intuitif dan mudah dipahami, tanpa harus menulis query SQL yang kompleks. Modul ini mencakup tiga jenis relationship utama: One-to-One (seperti hubungan user dan profile), One-to-Many (seperti hubungan category dan post), serta Many-to-Many (seperti hubungan students dan courses) yang memerlukan tabel pivot. Melalui praktikum yang disajikan, developer dapat mempelajari cara implementasi masing-masing relationship mulai dari pembuatan migrasi, model, seeder, controller, hingga view dengan antarmuka yang user-friendly menggunakan Bootstrap.

---

### 5. Referensi

Laravel Eloquent Relationship & Pagination — <https://hackmd.io/@mohdrzu/r1RPvWaCxx> Tutorial Laravel Eloquent Relationships#1: Apa itu Eloquent Relationships? — <https://santrikoding.com/tutorial-laravel-eloquent-relationships-1-apa-itu-eloquent-relationships> Laravel Pagination (Belajar Laravel #11) — <https://informatika.ciputra.ac.id/2019/11/laravel-crud-pagination/>

---