**LAPORAN PRAKTIKUM PEMROGRAMAN WEB**

**PRAKTIKUM 12 – Migration, Seeder, Factory**



Disusun oleh:

|  |  |  |
| --- | --- | --- |
| Nama | : | Alfath Roziq Widhayaka |
| Nim | : | L0122012 |
| Kelas | : | A |

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA**

**UNIVERSITAS SEBELAS MARET**

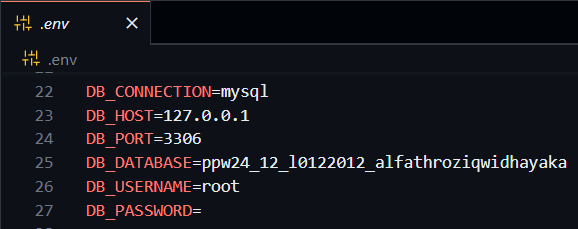
**2024**

**BAB I**

1. **Penjelasan Source Code**

Pada praktikum minggu ke-12 ini dilakukan praktikum pengembangan web untuk membuat website menggunakan Laravel. Kali ini menerapkan materi **Migration, Seeder,** dan **Factory**. Fungsi dari ketiga materi tersebut adalah untuk mengelola database, mengisi database, dan untuk membuat database berisikan dummy. Pada praktikum kali ini saya membuat website laravel berjudul **Daftar Film**. Berikut penjelasan masing-masing source codenya.

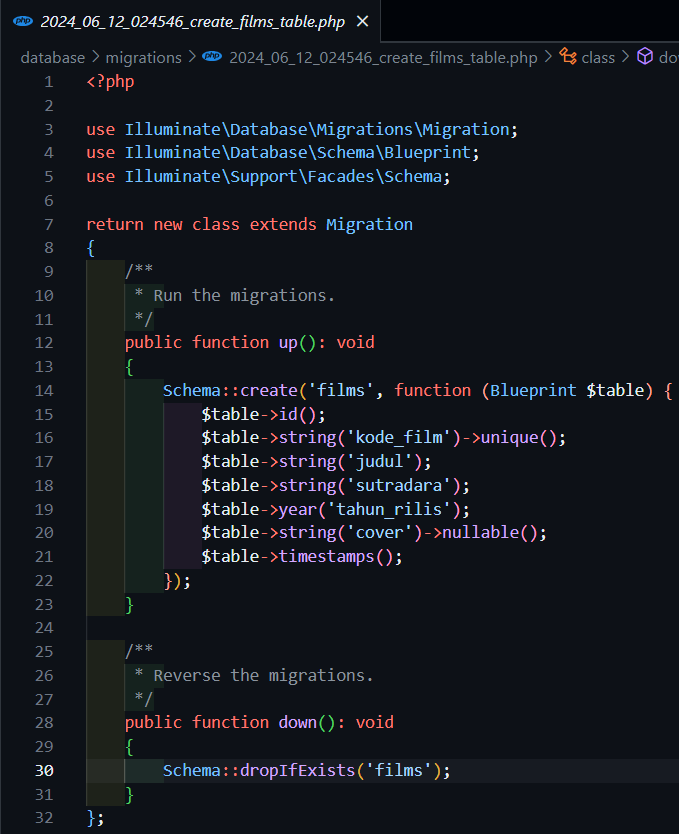
1. **Implementasi Migration**
2. **env**

****

*Gambar 1 env*

Kode **.env** di atas mengatur koneksi database untuk aplikasi Laravel, menggunakan MySQL sebagai database dengan host **127.0.0.1,** port **3306**, dan database bernama **ppw24\_12\_l0122012\_alfathroziqwidhayaka**. Aplikasi terhubung menggunakan username root tanpa password.

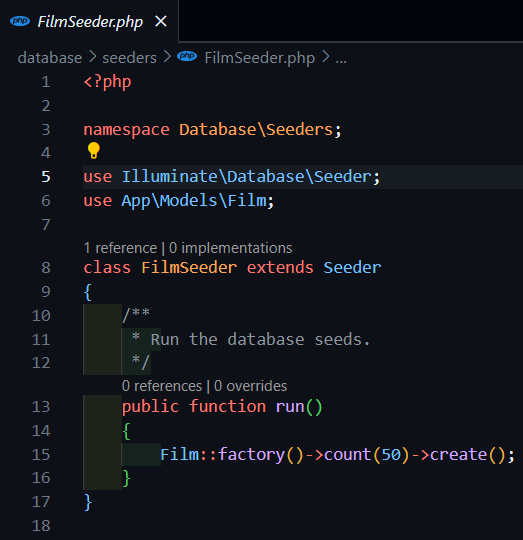
1. **film\_table.php**

****

*Gambar 2 films\_table.php*

Kode di atas adalah sebuah migrasi Laravel yang membuat tabel **films** di database. Dalam metode up, tabel films didefinisikan dengan kolom-kolom **id** sebagai **primary key, kode\_film yang unik, judul, sutradara, tahun\_rilis bertipe tahun, cover** yang dapat bernilai null, serta kolom timestamp (created\_at dan updated\_at). Metode down mendefinisikan tindakan rollback dengan menghapus tabel films jika migrasi dibatalkan. Migrasi ini memungkinkan manajemen struktur tabel films secara otomatis melalui perintah **php artisan migrate**.

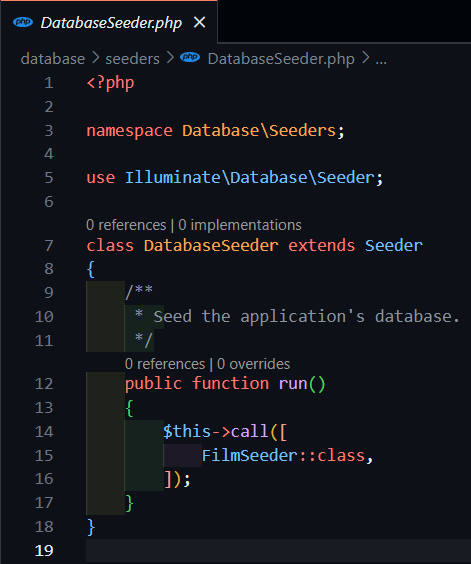
1. **Implementasi Seeder**
2. **FilmSeeder.php**

****

*Gambar 3 FilmSeeder.php*

Kode di atas adalah sebuah seeder Laravel yang mengisi tabel films dengan data contoh. Seeder ini berada dalam namespace Database\Seeders dan menggunakan model Film. Dalam metode run, seeder memanggil factory Film untuk membuat 50 entri data baru di tabel films dengan menggunakan **Film::factory()->count(50)->create();**. Factory ini otomatis menghasilkan data acak sesuai dengan definisi factory untuk model Film, sehingga memudahkan pengujian dan pengembangan dengan menyediakan banyak data contoh yang realistis di database.

1. **DatabaseSeeder.php**

****

*Gambar 4 DatabaseSeeder.php*

Kode di atas adalah seeder utama Laravel yang disebut **DatabaseSeeder**, berada dalam namespace Database\Seeders. Seeder ini bertugas mengisi database aplikasi dengan data awal. Metode run memanggil seeder lain, dalam hal ini **FilmSeeder**, menggunakan **$this->call([FilmSeeder::class])**. Dengan melakukan ini, DatabaseSeeder menjalankan FilmSeeder, yang pada gilirannya mengisi tabel films dengan 50 entri data contoh. Seeder ini membantu mengatur data awal aplikasi dengan mengumpulkan dan menjalankan berbagai seeder spesifik lainnya.

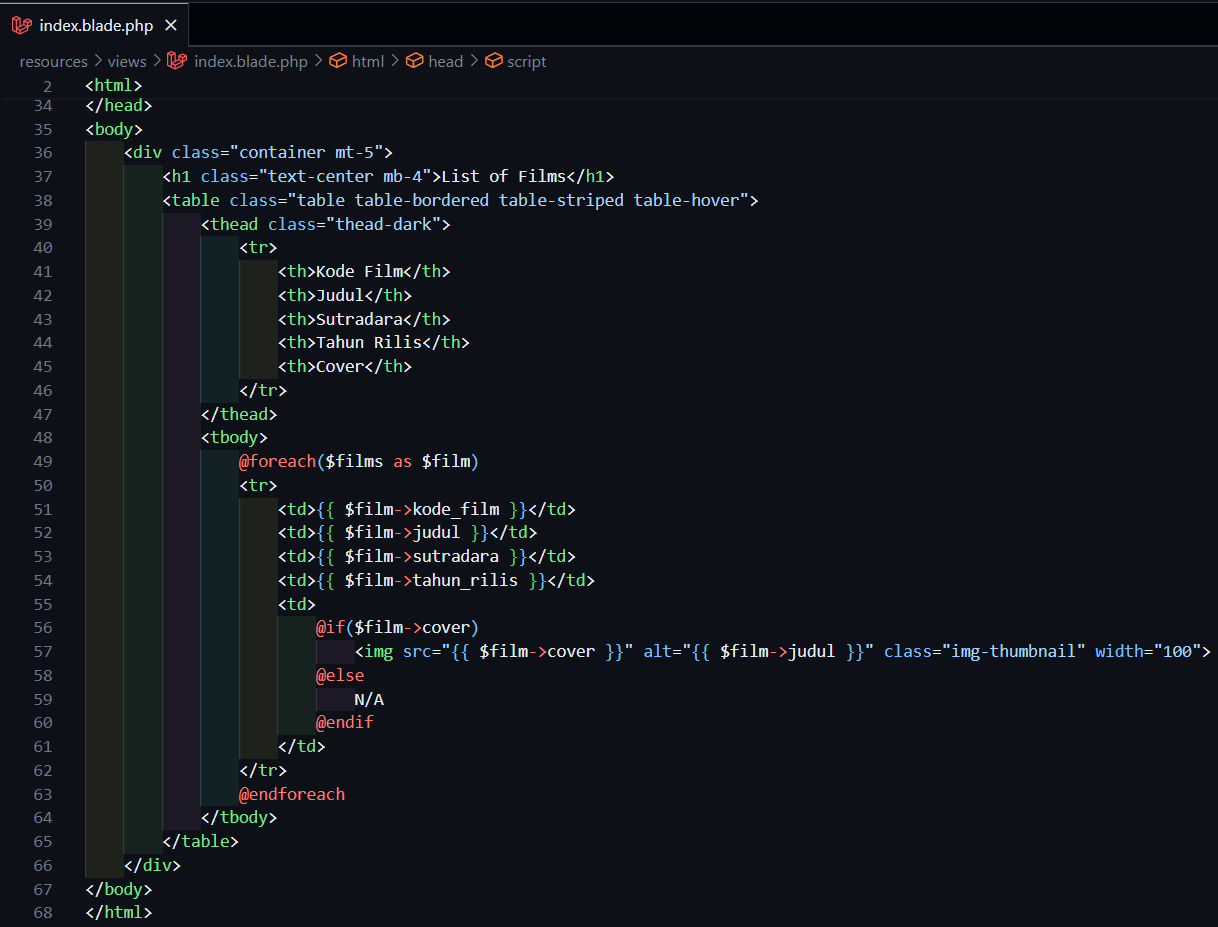
1. **Implementasi Factory**
2. **FilmFactory.php**

****

*Gambar 5 FilmFactory.php*

Kode di atas adalah sebuah factory Laravel yang digunakan untuk menghasilkan data acak untuk model Film. **Factory** ini berada dalam namespace Database\Factories dan menggambarkan struktur data yang akan dibuat untuk setiap entri Film. Dalam metode definition, factory mendefinisikan aturan pembuatan data, seperti menghasilkan kode unik untuk kode\_film, menghasilkan judul dengan tiga kata acak, menetapkan nama sutradara secara acak, menetapkan tahun rilis secara acak, dan opsional menyertakan URL gambar cover. Dengan menggunakan factory ini, Laravel dapat dengan mudah membuat banyak data contoh untuk digunakan dalam pengujian dan pengembangan aplikasi.

1. **Views**

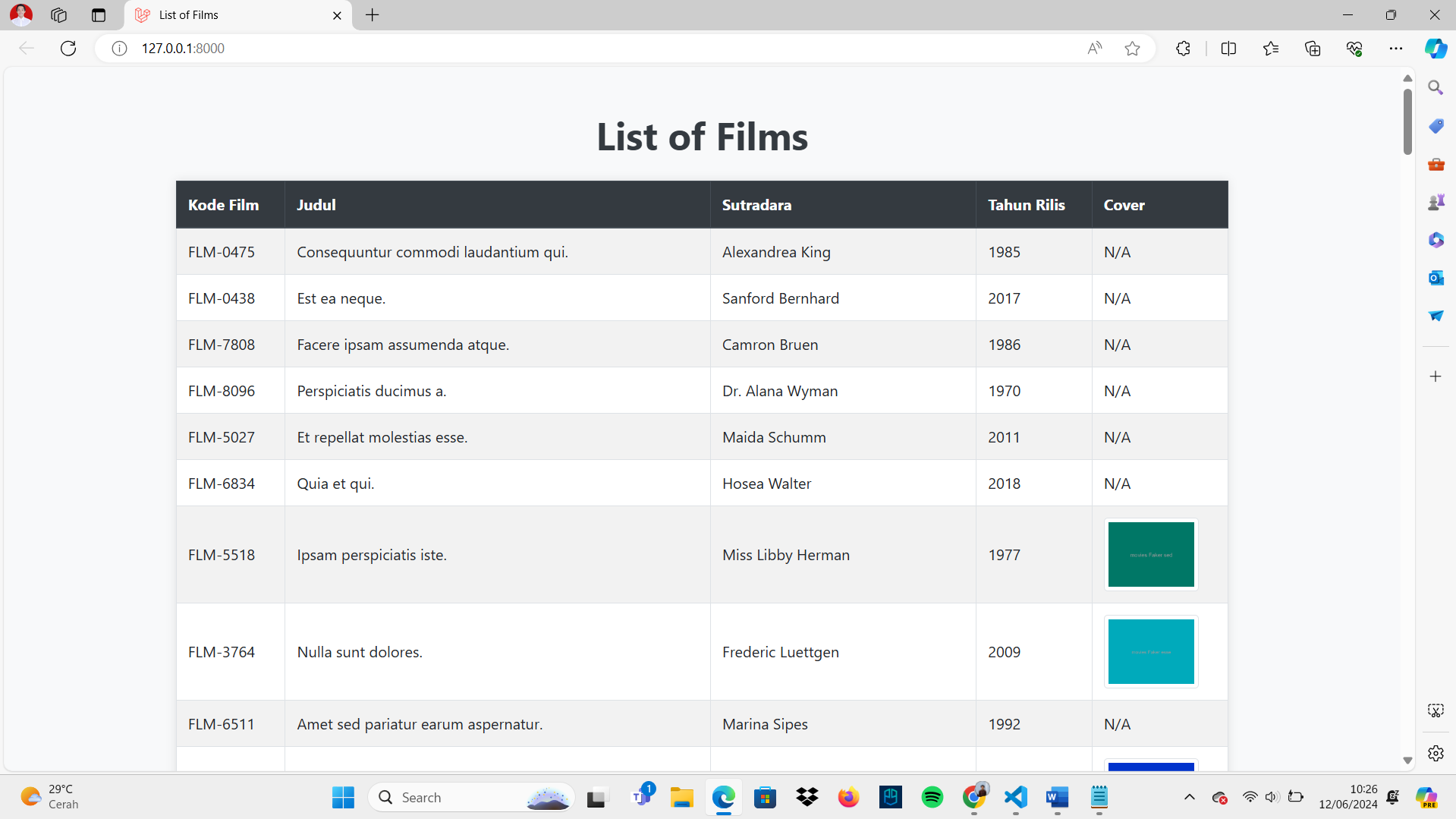


*Gambar 6 index.blade.php*

Kode view di atas adalah halaman web sederhana yang menampilkan daftar film dalam bentuk tabel. Halaman ini menggunakan HTML dan CSS dengan framework Bootstrap untuk tata letak dan tampilan yang responsif. Dalam tabel, setiap baris menampilkan informasi tentang sebuah film, termasuk kode film, judul, sutradara, tahun rilis, dan gambar cover jika tersedia. Data film disediakan melalui variabel **$film**s yang diteruskan ke view dari controller. Jika gambar cover tidak tersedia untuk suatu film, teks "N/A" akan ditampilkan.

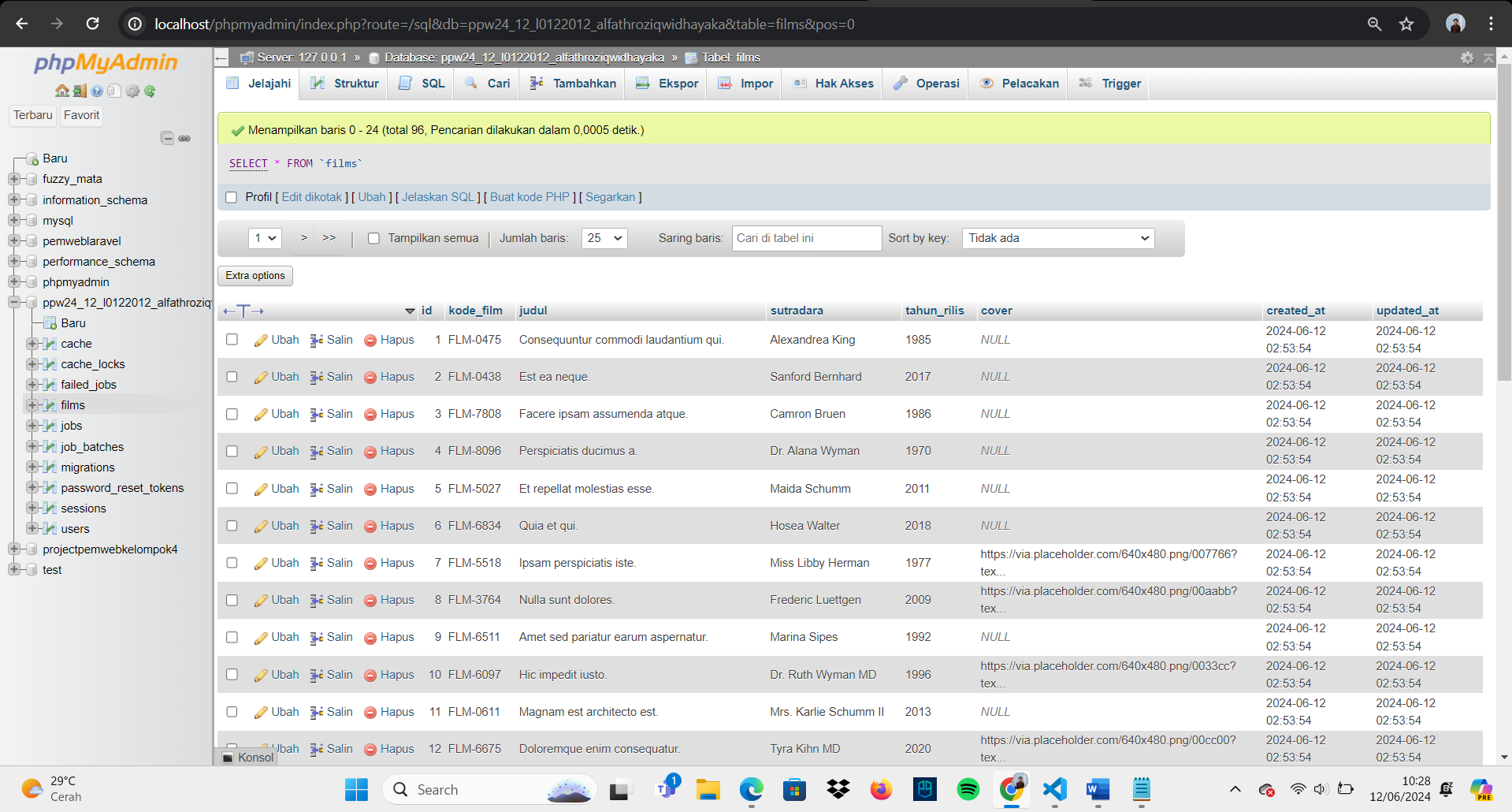
**BAB II**

1. **Hasil Output**

****

*Gambar 7 Tampilan Website*

Gambar diatas merupakan hasil ketika pengguna melakukan perintah **php artisan serve**. Hasilnya sama seperti yang dibuat pada **index.blade.php** yaitu pada website menampilkan daftar film yang mengimplementasikan data dummy dari pembuatan **Migration, Seeder,** dan **Factory.**

****

*Gambar 8 Tampilan phpMyAdmin*

Gambar diatas merupakan tampilan pada phpMyAdmin ketika user melakukan **Migration, Seeder**, dan **Factory**. Terlihat bahwa 50 data dummy otomatis terbentuk ketika menjalankan **php artisan db:seed**.

**BAB III**

1. **Kesimpulan**

Dalam praktikum ini, implementasi **Migration, Seeder, dan Factory** pada framework Laravel digunakan untuk mengelola **database, mengisi data,** dan **menciptakan data dummy** secara efisien. Melalui **Migration**, struktur tabel films didefinisikan dengan jelas, termasuk kolom-kolomnya. **Seeder** memanfaatkan **Factory** untuk mengisi tabel dengan 50 entri data contoh, secara otomatis memberikan data yang realistis untuk pengembangan dan pengujian aplikasi. Dengan DatabaseSeeder sebagai seeder utama, pengaturan data awal aplikasi dapat dengan mudah dilakukan dengan memanggil seeder-seeder spesifik lainnya, seperti FilmSeeder. Selain itu, **Factory** memungkinkan pembuatan data acak yang sesuai dengan aturan yang telah ditetapkan, seperti pembuatan judul film dengan tiga kata acak dan penentuan tahun rilis secara acak, memberikan fleksibilitas dalam pengembangan aplikasi.

Hasil output praktikum menunjukkan bahwa implementasi ini berhasil, dengan tampilan website yang menampilkan daftar film yang diperoleh dari data dummy yang telah disiapkan. Penggunaan phpMyAdmin juga memperlihatkan bahwa 50 data dummy telah berhasil terbentuk setelah melakukan Migration, Seeder, dan Factory. Dengan demikian, praktikum ini memberikan pemahaman yang baik tentang bagaimana menggunakan Migration, Seeder, dan Factory dalam pengembangan web menggunakan Laravel, serta menunjukkan pentingnya pengelolaan data yang efisien dan efektif dalam pembangunan aplikasi web.