

# **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**

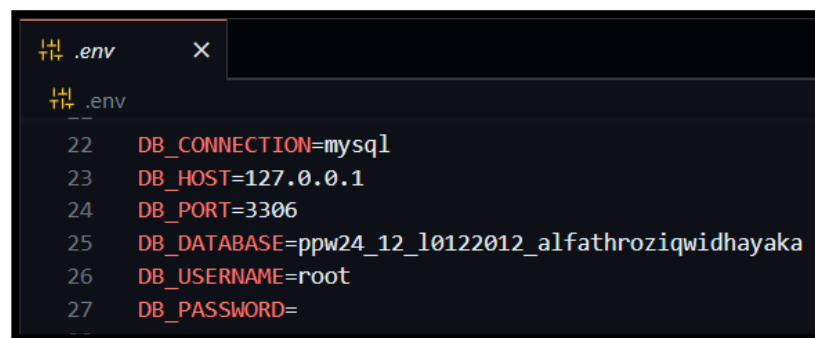
# BAB I

## A. 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

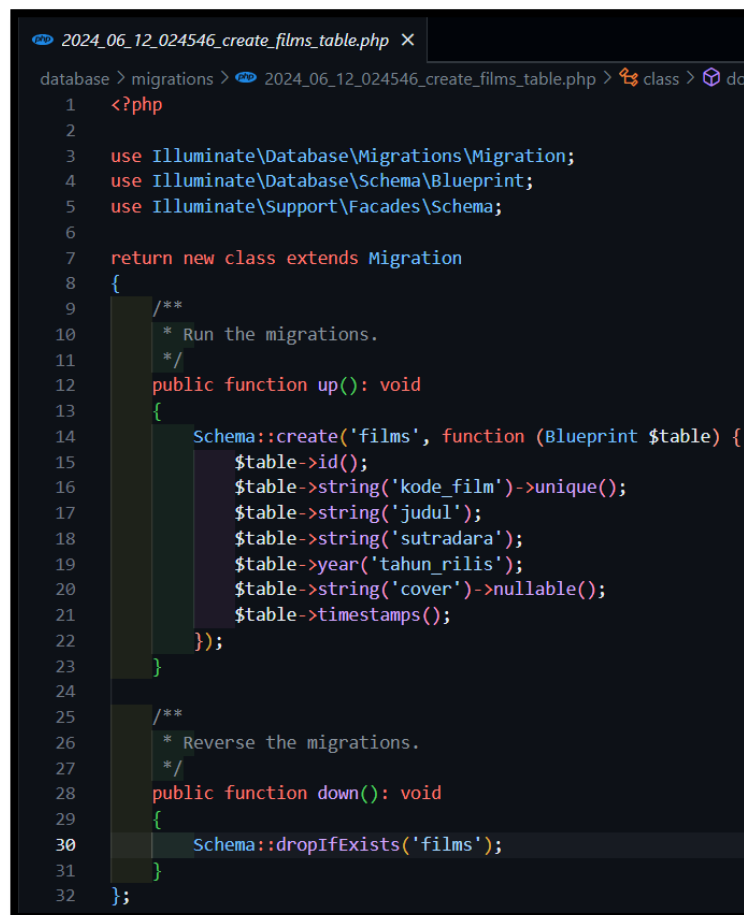
#### a. env



*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.

## b. film\_table.php



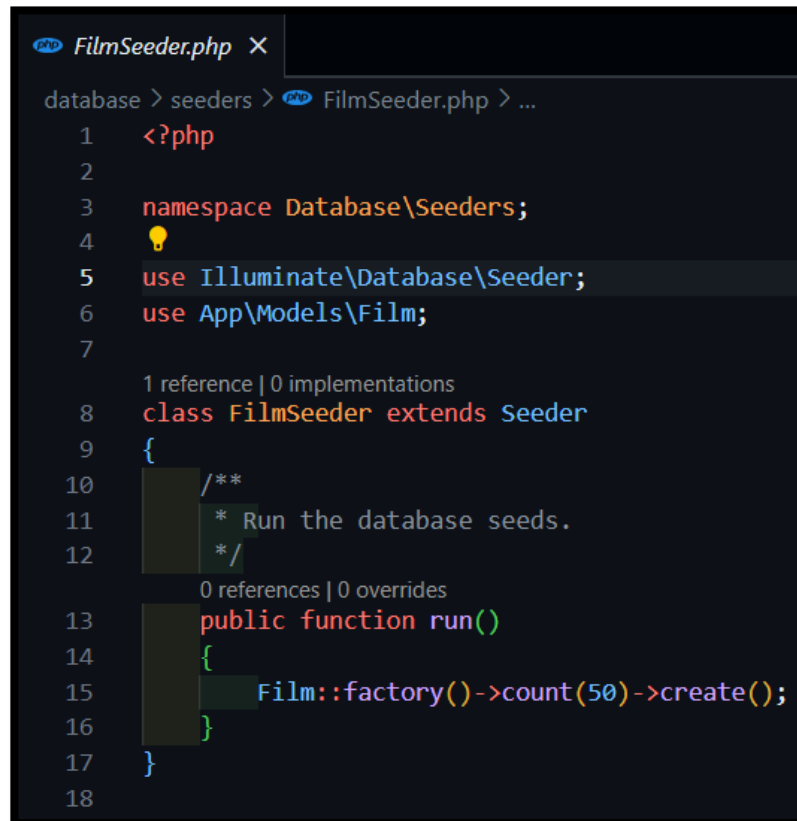
```
2024_06_12_024546_create_films_table.php X
database > migrations > 2024_06_12_024546_create_films_table.php > class > do
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('films', function (Blueprint $table) {
15             $table->id();
16             $table->string('kode_film')->unique();
17             $table->string('judul');
18             $table->string('sutradara');
19             $table->year('tahun_rilis');
20             $table->string('cover')->nullable();
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      */
28     public function down(): void
29     {
30         Schema::dropIfExists('films');
31     }
32 };
```

*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**.

## 2. Implementasi Seeder

### a. FilmSeeder.php

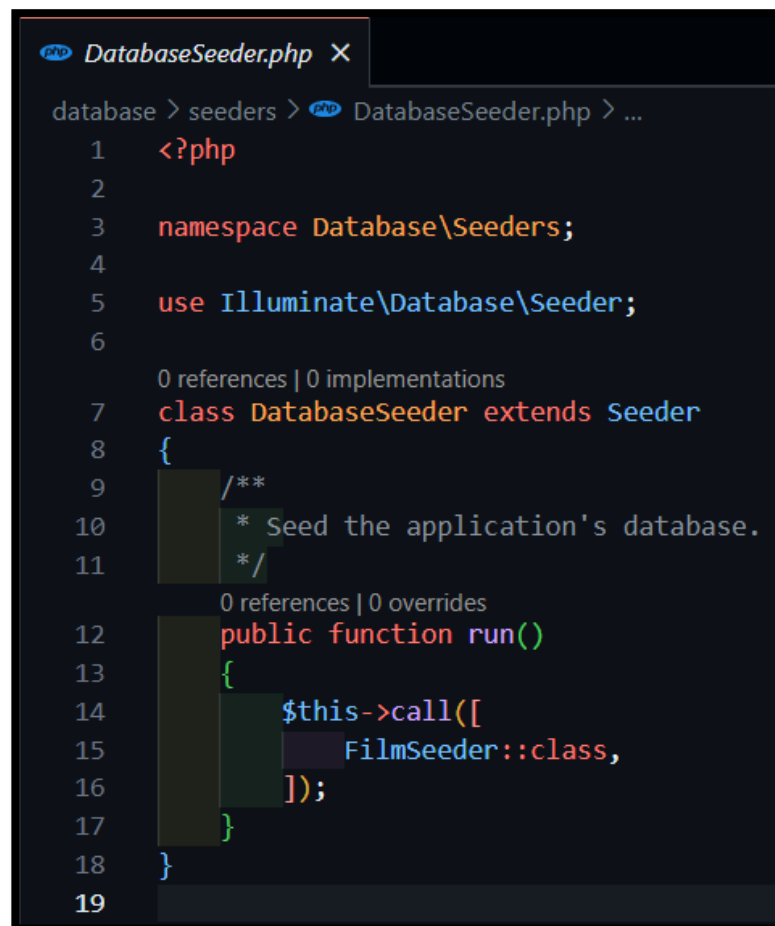


```
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Seeder;
6 use App\Models\Film;
7
8 class FilmSeeder extends Seeder
9 {
10     /**
11      * Run the database seeds.
12      */
13     public function run()
14     {
15         Film::factory()->count(50)->create();
16     }
17 }
18
```

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.

## b. DatabaseSeeder.php

A screenshot of a code editor showing the DatabaseSeeder.php file. The editor has a dark theme. The file path is 'database > seeders > DatabaseSeeder.php'. The code is as follows:

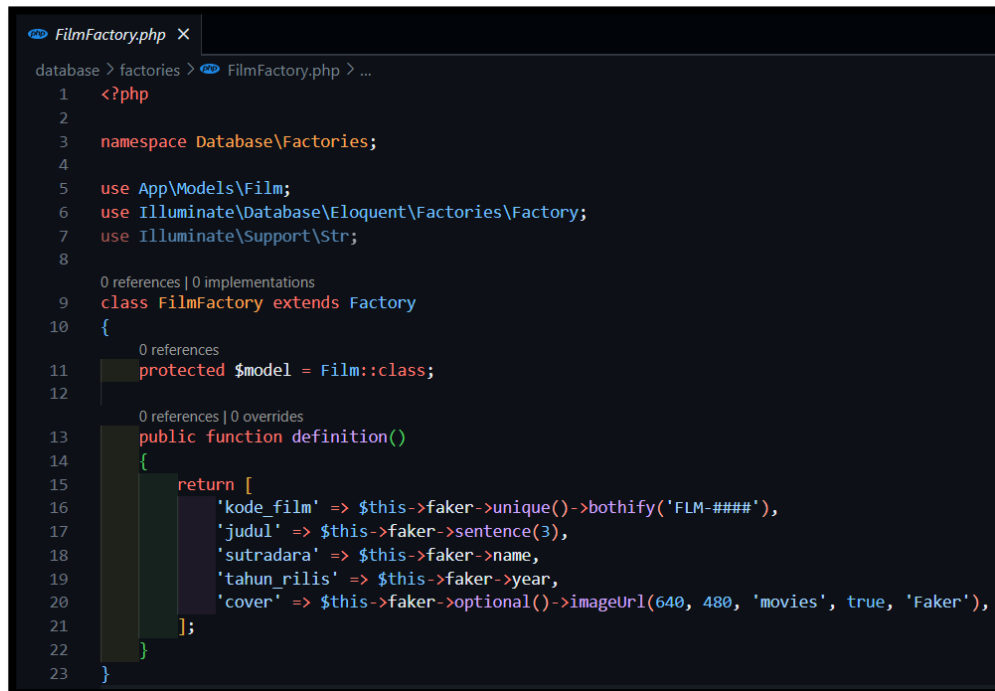
```
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Seeder;
6
7  class DatabaseSeeder extends Seeder
8  {
9      /**
10       * Seed the application's database.
11       */
12     public function run()
13     {
14         $this->call([
15             FilmSeeder::class,
16         ]);
17     }
18 }
19
```

*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.

### 3. Implementasi Factory

#### a. FilmFactory.php



```
1 <?php
2
3 namespace Database\Factories;
4
5 use App\Models\Film;
6 use Illuminate\Database\Eloquent\Factories\Factory;
7 use Illuminate\Support\Str;
8
9 class FilmFactory extends Factory
10 {
11     protected $model = Film::class;
12
13     public function definition()
14     {
15         return [
16             'kode_film' => $this->faker->unique()->bothify('FLM-###'),
17             'judul' => $this->faker->sentence(3),
18             'sutradara' => $this->faker->name,
19             'tahun_rilis' => $this->faker->year,
20             'cover' => $this->faker->optional()->imageUrl(640, 480, 'movies', true, 'Faker'),
21         ];
22     }
23 }
```

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.

## 4. Views

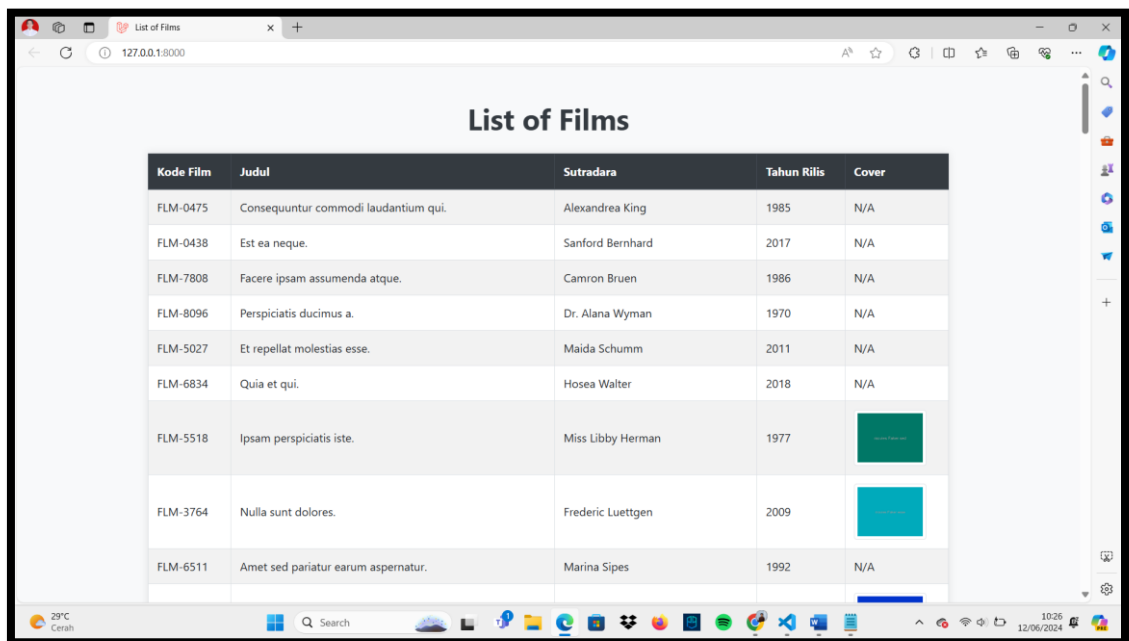
```
index.blade.php X
resources > views > index.blade.php > html > head > script
2  <html>
34 </head>
35 <body>
36     <div class="container mt-5">
37         <h1 class="text-center mb-4">List of Films</h1>
38         <table class="table table-bordered table-striped table-hover">
39             <thead class="thead-dark">
40                 <tr>
41                     <th>Kode Film</th>
42                     <th>Judul</th>
43                     <th>Sutradara</th>
44                     <th>Tahun Rilis</th>
45                     <th>Cover</th>
46                 </tr>
47             </thead>
48             <tbody>
49                 @foreach($films as $film)
50                 <tr>
51                     <td>{{ $film->kode_film }}</td>
52                     <td>{{ $film->judul }}</td>
53                     <td>{{ $film->sutradara }}</td>
54                     <td>{{ $film->tahun_rilis }}</td>
55                     <td>
56                         @if($film->cover)
57                             judul }}" class="img-thumbnail" width="100">
58                         @else
59                             N/A
60                         @endif
61                     </td>
62                 </tr>
63                 @endforeach
64             </tbody>
65         </table>
66     </div>
67 </body>
68 </html>
```



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 **\$films** yang diteruskan ke view dari controller. Jika gambar cover tidak tersedia untuk suatu film, teks "N/A" akan ditampilkan.

## BAB II

### A. Hasil Output



Kode Film	Judul	Sutradara	Tahun Rilis	Cover
FLM-0475	Consequuntur commodi laudantium qui.	Alexandrea King	1985	N/A
FLM-0438	Est ea neque.	Sanford Bernhard	2017	N/A
FLM-7808	Facere ipsam assumenda atque.	Camron Bruen	1986	N/A
FLM-8096	Perspiciatis ducimus a.	Dr. Alana Wyman	1970	N/A
FLM-5027	Et repellat molestias esse.	Maida Schumm	2011	N/A
FLM-6834	Quia et qui.	Hosea Walter	2018	N/A
FLM-5518	Ipsam perspiciatis iste.	Miss Libby Herman	1977	
FLM-3764	Nulla sunt dolores.	Frederic Luetngen	2009	
FLM-6511	Amet sed pariatur earum aspernatur.	Marina Sipes	1992	N/A

*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**.



id	kode_film	judul	sutradara	tahun_rilis	cover	created_at	updated_at
1	FLM-0475	Consequuntur commodi laudantium qui.	Alexandrea King	1985	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
2	FLM-0438	Est ea neque.	Sanford Bernhard	2017	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
3	FLM-7808	Facere ipsam assumenda atque.	Camron Bruen	1986	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
4	FLM-8096	Perspiciatis ducimus a.	Dr. Alana Wyman	1970	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
5	FLM-5027	Et repellat molestias esse.	Maida Schumm	2011	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
6	FLM-6834	Quia et qui.	Hosea Walter	2018	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
7	FLM-5518	Ipsam perspiciatis iste.	Miss Libby Herman	1977	https://via.placeholder.com/640x480.png/007786?tex...	2024-06-12 02:53:54	2024-06-12 02:53:54
8	FLM-3764	Nulla sunt dolores.	Frederic Luetigen	2009	https://via.placeholder.com/640x480.png/00aabb?tex...	2024-06-12 02:53:54	2024-06-12 02:53:54
9	FLM-6511	Amet sed pariat eorum aspernatur.	Marina Spies	1992	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
10	FLM-6097	Hic impedit iusto.	Dr. Ruth Wyman MD	1996	https://via.placeholder.com/640x480.png/0033cc?tex...	2024-06-12 02:53:54	2024-06-12 02:53:54
11	FLM-0611	Magnam est architecto est.	Mrs. Karlie Schumm II	2013	NULL	2024-06-12 02:53:54	2024-06-12 02:53:54
12	FLM-9975	Doloremque enim consequatur.	Tyra Kihn MD	2020	https://via.placeholder.com/640x480.png/00cc00?tex...	2024-06-12 02:53:54	2024-06-12 02:53:54

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

### A. 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.