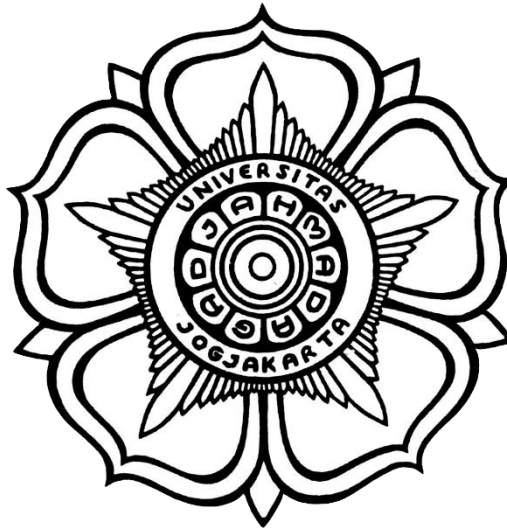


LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB 2
PERTEMUAN 4
CONTROLLER DAN MIGRATION



Disusun Oleh:

Ghita Najmi Naqasy

22/496466/SV/20961

Dosen Pengampu:

Dinar Nugroho Pratomo, S.Kom., M.IM., M.Cs.

Faza Maula Azif, S.Kom., M.Eng.

PROGRAM STUDI TEKNOLOGI REKAYASA PERANGKAT LUNAK
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2023

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	1
1.3. Tujuan	2
BAB II PEMBAHASAN.....	3
2.1. Controller	3
2.2. Migration.....	3
2.3. Seedder.....	4
BAB III LATIHAN PRAKTIKUM.....	5
3.1. Controller	5
3.2. Migration.....	6
3.3. Link Github.....	10
BAB IV KESIMPULAN.....	11
4.1. Kesimpulan	11
DAFTAR PUSTAKA.....	12

DAFTAR GAMBAR

Gambar 3.1. 1. Test Controller.....	5
Gambar 3.1. 2. Routes	6
Gambar 3.2. 1. Membuat database.....	6
Gambar 3.2. 2. Melakukan Migration.....	6
Gambar 3.2. 3. Function Up	7
Gambar 3.2. 4. Function Down	7
Gambar 3.2. 5. Create Perumahan	7
Gambar 3.2. 6. Insert Data	8
Gambar 3.2. 7. Folder Migration	8
Gambar 3.2. 8. Menambahkan Data.....	8
Gambar 3.2. 9. Tabel Perumahan	9
Gambar 3.2. 10. Seeder	10

BAB I

PENDAHULUAN

1.1. Latar Belakang

Controller adalah bagian penting dalam pengembangan perangkat lunak, terutama dalam aplikasi web. Ini adalah bagian dari kerangka kerja seperti Laravel yang digunakan untuk mengelola permintaan HTTP dari pengguna dan mengatur tindakan yang harus diambil dalam meresponsnya. Controller bertindak sebagai penghubung antara permintaan pengguna dan model (data) dalam aplikasi. Tugas utamanya adalah mengontrol aliran data, memproses permintaan, dan merespons dengan mengirimkan tampilan atau data yang sesuai kembali ke pengguna. Dalam controller, mendefinisikan metode-metode atau fungsi-fungsi yang akan dijalankan ketika pengguna mengakses berbagai URL atau rute dalam aplikasi web. Controller juga berperan dalam memvalidasi input pengguna, mengakses basis data, dan mengembalikan hasil kepada pengguna.

Migration adalah proses yang digunakan dalam pengembangan perangkat lunak untuk mengelola perubahan struktur basis data ini memungkinkan pengembang untuk membuat, mengubah, atau menghapus tabel dan kolom dalam basis data tanpa harus menghapus seluruh basis data dan memulainya dari awal, konteks kerangka kerja web seperti Laravel, migration digunakan untuk mendefinisikan dan mengatur perubahan struktur basis data. Setiap kali Anda ingin menambahkan tabel baru, mengubah tipe data kolom, atau melakukan perubahan struktural lainnya, Anda dapat membuat migrasi untuk melakukan tugas tersebut. Migrasi biasanya ditulis dalam kode yang dapat dijalankan sehingga perubahan basis data dapat diimplementasikan dengan mudah.

1.2. Rumusan Masalah

Berikut rumusan masalah dalam praktikum ini:

1. Bagaimana cara mengaplikasikan Controller dan Migration.
2. Apa saja manfaat yang dapat diambil dari mengaplikasikan Controller dan Migration.
3. Bagaimana proses Controller dan Migration bekerja.

1.3. Tujuan

Berikut adalah tujuan yang ingin dicapai pada praktikum kali ini:

1. Mampu menerapkan Controller dan Migration.
2. Mengenali Controller dan Migration.
3. Mampu memahami Controller dan Migration.

BAB II

PEMBAHASAN

2.1. Controller

Controller adalah bagian dari perangkat lunak yang bertugas menerima permintaan dari pengguna dan memberikan tanggapan yang sesuai. Ini mirip dengan operator yang mengatur aliran lalu lintas di persimpangan jalan. Pertama, pengguna membuat permintaan melalui web browser atau aplikasi. Permintaan ini berisi informasi tentang apa yang pengguna inginkan, seperti membuka halaman tertentu atau menyimpan data. Controller adalah yang mengambil permintaan ini dan memutuskan apa yang harus dilakukan selanjutnya. Misalnya, jika Anda mengklik tautan ke halaman profil pengguna, controller akan menentukan bahwa itu adalah permintaan untuk menampilkan halaman profil, dan kemudian mengambil data yang diperlukan dan menampilkan halaman tersebut. Jadi, controller adalah pengatur utama dalam menangani interaksi antara pengguna dan aplikasi Anda. Ini memastikan bahwa permintaan dari pengguna diinterpretasikan dengan benar dan ditindaklanjuti dengan benar.

2.2. Migration

Migration adalah alat yang digunakan untuk mengelola versi dari basis data aplikasi, bisa menganggapnya sebagai cara untuk mengubah struktur basis data tanpa kehilangan data yang ada atau merusak aplikasi. Bayangkan jika aplikasi toko online dan ingin menambahkan kolom baru ke tabel produk untuk menyimpan informasi tambahan. Migration memungkinkan untuk membuat perubahan ini dengan cara yang terorganisir dan aman. Ini seperti mengubah desain interior toko tanpa harus menutup toko selama renovasi. Setiap kali membuat perubahan pada struktur basis data, seperti menambahkan tabel baru, menghapus kolom lama, atau mengubah tipe data, membuat "migrasi" yang merekam perubahan tersebut. Kemudian, ketika ingin menerapkan perubahan ini, menjalankan migrasi ini, dan basis data akan diperbarui sesuai dengan perubahan tersebut. Ini memastikan bahwa aplikasi selalu berjalan dengan baik, bahkan ketika struktur basis data berubah seiring waktu. Jadi, migration adalah alat yang membantu mengelola perubahan struktur basis data tanpa merusak aplikasi, mirip dengan renovasi toko tanpa menutupnya

2.3. Seedder

Seeder digunakan untuk memasukkan data sampel, data awal, atau data uji ke dalam tabel-tabel database. Ini memiliki beberapa tujuan utama: Pengembangan dan Pengujian: Seeder digunakan saat mengembangkan atau menguji perangkat lunak. Mereka membantu pengembang untuk mengisi database dengan data palsu atau data uji sehingga mereka dapat menguji fungsi-fungsi perangkat lunak dengan data yang realistis tanpa harus memasukkan data manual setiap kali. Mengisi Basis Data Awal: Ketika Anda memulai proyek perangkat lunak baru atau sistem, Anda seringkali perlu mengisi basis data dengan beberapa data awal yang dibutuhkan. Seeder membantu Anda melakukannya secara otomatis dengan menghasilkan entri data yang sudah ada dalam jumlah yang dibutuhkan. Memastikan Konsistensi: Dengan menggunakan seeder, Anda dapat memastikan bahwa basis data awal selalu memiliki data yang konsisten dan sesuai dengan kebutuhan aplikasi Anda. Ini membantu mencegah kesalahan dalam memasukkan data manual yang mungkin tidak selalu konsisten. Migrasi Data: Seeder juga berguna saat Anda memindahkan atau mengganti versi basis data. Dalam hal ini, Anda dapat menggunakan seeder untuk mengisi data baru ke dalam skema database yang baru. Cara kerja seeder biasanya melibatkan penulisan kode atau skrip yang akan memasukkan data ke dalam tabel-tabel database. Seeder dapat dijalankan satu kali saat Anda membangun atau memperbarui basis data, atau Anda dapat menjalankannya secara berkala jika Anda perlu memperbarui data uji atau data awal dalam pengembangan perangkat lunak Anda.

BAB III

LATIHAN PRAKTIKUM

3.1. Controller

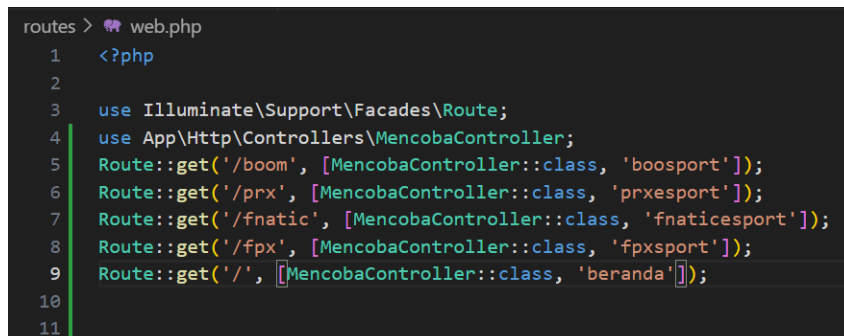
Class TestController extends Controller: Ini adalah deklarasi kelas TestController yang meng-extend (mewarisi) kelas Controller dari Laravel, controller adalah komponen dalam Laravel yang digunakan untuk mengelola permintaan HTTP selanjutnya, terdapat beberapa fungsi dalam TestController, boomesport(): Method ini mengembalikan tampilan dengan nama 'boom'. Ketika method ini dipanggil, akan mengarahkan aplikasi web ke tampilan 'boom'. prxesport(): Method ini mengembalikan tampilan dengan nama 'prx'. Ketika method ini dipanggil, akan mengarahkan aplikasi web ke tampilan 'prx'. fnaticesport(): Method ini mengembalikan tampilan dengan nama 'fpx'. Ketika method ini dipanggil, akan mengarahkan aplikasi web ke tampilan 'fpx'. beranda(): Method ini mengembalikan tampilan dengan nama 'layout/home'. Ini digunakan untuk menampilkan tampilan beranda dengan mengorganisasi kode dalam controller seperti ini, dapat dengan mudah mengelola berbagai permintaan HTTP dan mengarahkan mereka ke tampilan yang sesuai dalam aplikasi web Laravel setiap metode dalam controller ini merespons permintaan yang berbeda dengan cara yang sesuai.

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class TestController extends Controller
8  {
9      //Mencoba Controller lain
10     public function boomesport(){
11         return view('boom');
12     }
13     public function prxesport(){
14         return view('prx');
15     }
16     public function fnaticesport(){
17         return view('fpx');
18     }
19     public function beranda(){
20         return view('layout/home');
21     }
22 }
```

Gambar 3.1. 1. Test Controller

Penggunaan kelas MencobaController yang terdapat dalam namespace App\Http\Controllers. Controller ini akan digunakan untuk menangani permintaan yang datang melalui rute-rute yang didefinisikan di bawahnya. Route::get('/boom', [MencobaController::class, 'boosport']); Ini adalah definisi rute dengan metode HTTP GET. Rute ini mengarahkan permintaan yang datang

ke URL /boom ke metode boosport dalam controller MencobaController. Demikian pula, ada definisi rute lain yang mengarahkan URL tertentu ke metode-metode dalam MencobaController. `Route::get('/', [MencobaController::class, 'beranda']);` Ini adalah definisi rute akar (/) yang mengarahkan ke metode beranda dalam MencobaController rute yang akan diakses ketika pengguna membuka halaman utama aplikasi mendefinisikan rute-rute ini, memberi tahu Laravel bagaimana cara menangani permintaan yang masuk ke berbagai URL. Setiap rute mengarahkan permintaan ke metode yang sesuai dalam controller, yang kemudian dapat mengambil tindakan yang sesuai atau mengembalikan tampilan yang diperlukan.



```

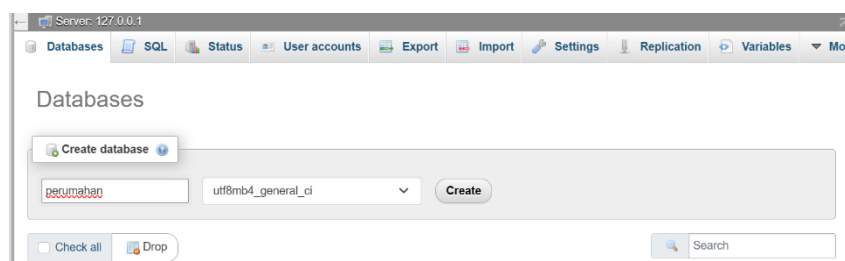
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\MencobaController;
5  Route::get('/boom', [MencobaController::class, 'boosport']);
6  Route::get('/prx', [MencobaController::class, 'prxesport']);
7  Route::get('/fnatic', [MencobaController::class, 'fnaticesport']);
8  Route::get('/fpx', [MencobaController::class, 'fpxsport']);
9  Route::get('/', [MencobaController::class, 'beranda']);
10
11

```

Gambar 3.1. 2. Routes


3.2. Migration

Buat sebuah database baru dengan nama perumahan.



Gambar 3.2. 1. Membuat database

Perintah php artisan make:migration digunakan dalam kerangka kerja Laravel untuk membuat berkas migrasi baru. Migrasi adalah cara untuk mengelola perubahan struktur database secara terstruktur.



```

PS C:\xampp\htdocs\instalasi> php artisan make:migration create_perumahan
INFO Migration [C:\xampp\htdocs\instalasi\database\migrations\2023_09_05_070742_create_perumahan.php] created successfully.

```

Gambar 3.2. 2. Melakukan Migration

Kemudian buat sebuah kolom dengan tipe data yang sesuai seperti contoh di bawah ini terdapat kolom “id_perumahan” “luas_perumahan” “harga_perumahan”.

```

7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('perumahan', function (Blueprint $table) {
15               $table->increments('id_perumahan');
16               $table->integer('luas_perumahan');
17               $table->integer('harga_perumahan');
18               $table->timestamps();
19           });
20       }

```

Gambar 3.2. 3. Function Up

Kode ini adalah definisi migrasi dalam bentuk anonymous class (kelas tanpa nama) yang meng-extend kelas Migration. Migrasi ini digunakan untuk mengelola tabel perumahan dalam database dengan menggunakan migrasi seperti ini, dapat dengan mudah mengelola struktur database termasuk pembuatan dan penghapusan tabel serta perubahan-perubahan lain dalam database selama pengembangan aplikasi.

```

22       /**
23        * Reverse the migrations.
24        */
25       public function down(): void
26       {
27           Schema::dropIfExists('perumahan');
28       }
29     };
30

```

Gambar 3.2. 4. Function Down

Perintah php artisan migrate adalah perintah yang digunakan dalam kerangka kerja Laravel untuk menjalankan migrasi database. Migrasi adalah cara untuk mengelola perubahan struktur database Anda dengan cara yang terstruktur dan terdokumentasi.

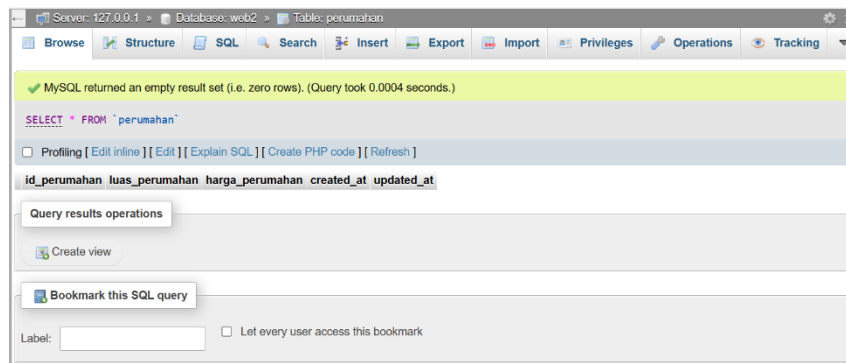
```

PS C:\xampp\htdocs\instalasi> php artisan migrate
INFO: Running migrations.
2023_09_05_070742_create_perumahan ... 13ms DONE
PS C:\xampp\htdocs\instalasi>

```

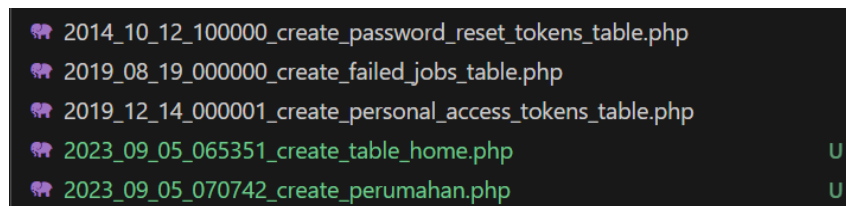
Gambar 3.2. 5. Create Perumahan

Jika dilihat pada gambar di bawah sudah terisi beberapa kolom

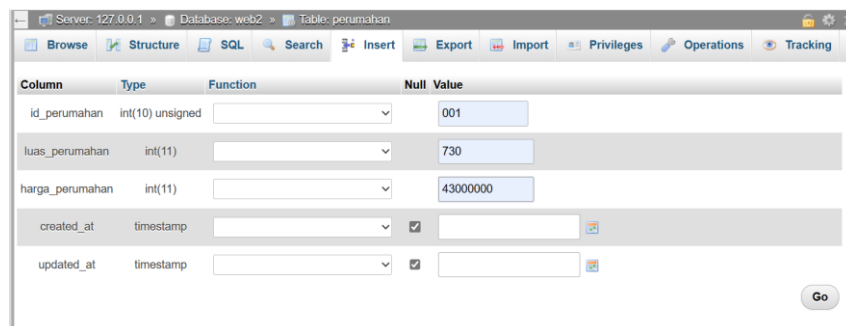


Gambar 3.2. 6. Insert Data

Folder migration dalam Laravel berisi berkas-berkas migrasi yang digunakan untuk mengelola struktur database aplikasi. Setiap berkas migrasi memiliki nama unik dan berisi kode yang mendefinisikan perubahan yang akan diterapkan pada database saat migrasi dijalankan. Berkas migrasi dalam folder migration adalah catatan tentang perubahan struktur database sepanjang waktu. Migrasi memungkinkan untuk secara konsisten mengelola skema basis data dan membuatnya lebih mudah untuk bekerja dalam tim atau mengelola perubahan struktur database seiring waktu.



Gambar 3.2. 7. Folder Migration



Gambar 3.2. 8. Menambahkan Data

Server: 127.0.0.1 » Database: web2 » Table: perumahan

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

SELECT * FROM `perumahan`

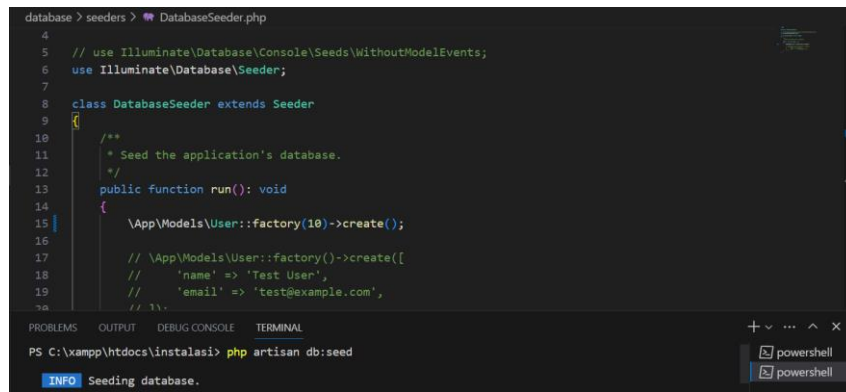
Number of rows: 25 Filter rows: Search this table

	id_perumahan	luas_perumahan	harga_perumahan	created_at	updated_at
	1	730	43000000	NULL	NULL

Gambar 3.2. 9. Tabel Perumahan

DatabaseSeeder yang terletak dalam direktori database/seeder dalam proyek Laravel kode ini digunakan untuk mengisi data awal (seed data) ke dalam database aplikasi deklarasi kelas DatabaseSeeder, yang mewarisi kelas Seeder. Kelas Seeder adalah bagian dari kerangka kerja Laravel yang digunakan untuk menulis kode penanaman data (seeding).

Metode run: Metode ini digunakan untuk menentukan tindakan yang akan dijalankan ketika penanaman data dilakukan kode ini, metode run memanggil `\App\Models\User::factory(10)->create();` adalah cara untuk menggunakan Factory Laravel untuk membuat dan menyimpan 10 data pengguna (users) ke dalam tabel users dalam database. User Factory: `\App\Models\User::factory(10)` adalah pemanggilan metode factory yang digunakan untuk membuat 10 objek data pengguna (users). Factory adalah cara untuk mendefinisikan cara membuat data yang realistis untuk pengujian atau seeding. `create()` digunakan untuk menyimpan data yang telah dibuat ke dalam database kode ini akan membuat 10 pengguna (users) dengan data acak menggunakan factory dan menyimpannya ke dalam tabel users dalam database ketika menjalankan perintah penanaman data (`php artisan db:seed`) dalam Laravel. Ini adalah cara yang efisien untuk mengisi database dengan data awal atau data tes.



The image shows a screenshot of a code editor with a dark theme. The main editor window displays a PHP file named `DatabaseSeeder.php`. The code is as follows:

```
database > seeders DatabaseSeeder.php
4
5 // use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7
8 class DatabaseSeeder extends Seeder
9 {
10     /**
11      * Seed the application's database.
12      */
13     public function run(): void
14     {
15         \App\Models\User::factory(10)->create();
16
17         // \App\Models\User::factory()->create([
18         //     'name' => 'Test User',
19         //     'email' => 'test@example.com',
20         // ]);
21     }
22 }
```

Below the code editor is a terminal window. The prompt is `PS C:\xampp\htdocs\instalasi>`. The command entered is `php artisan db:seed`. The output of the command is `INFO Seeding database.`. To the right of the terminal, there are two tabs labeled `powershell`.

Gambar 3.2. 10. Seeder

3.3. Link Github

<https://github.com/ghitanajminaqasy/PPW3.git>

<https://github.com/ghitanajminaqasy/PPW4.git>

BAB IV

KESIMPULAN

4.1. Kesimpulan

Berikut adalah kesimpulan singkat tentang controller, migration, dan seeder dalam konteks kerangka kerja Laravel. Controller adalah komponen dalam Laravel yang digunakan untuk mengelola permintaan HTTP ketiganya menghubungkan permintaan yang datang dari rute ke tindakan yang sesuai, seperti menampilkan tampilan atau mengelola data. Controller membantu dalam pemisahan logika bisnis dari tampilan, meningkatkan skalabilitas dan pemeliharaan kode. Migration adalah cara terstruktur dalam mengelola perubahan struktur database dalam aplikasi Laravel. Migrasi digunakan untuk membuat dan memodifikasi tabel database, menambahkan atau menghapus kolom, dan lainnya. Migrasi dapat dilacak dan di-rollback, memastikan bahwa database tetap konsisten sepanjang waktu. Seeder adalah cara untuk mengisi data awal ke dalam tabel database semua digunakan untuk mengisi tabel dengan data default atau data tes. Seeder membantu mengisi database dengan data yang diperlukan untuk pengujian atau pengembangan awal. Ketiga konsep ini, yaitu controller, migration, dan seeder, merupakan bagian penting dalam pengembangan aplikasi web Laravel membantu dalam mengelola permintaan, memelihara struktur database, dan mengisi data awal ke dalam aplikasi dengan cara yang terstruktur dan efisien.

DAFTAR PUSTAKA

- Ahmad, T. (2020, April 10). *Belajar Laravel (Chapter 7 | Tutorial Database: Migrations)*. Medium; dotlocal. <https://medium.com/dotlocal/belajar-laravel-chapter-7-tutorial-database-migrations-5e73753349b8>
- SEW. (2019, October 23). *Laravel Model & Migrations (Belajar Laravel #8) - Informatika Universitas Ciputra*. Informatika Universitas Ciputra. <https://informatika.uc.ac.id/2019/10/laravel-model-2/>
- Menggunakan Seeder di Laravel*. (2017). Codepolitan.com. <https://codepolitan.com/blog/menggunakan-seeder-di-laravel-59f7249589e2f>
- Dedy Yugo Purwanto. (2016, October 24). *Database Seeder di Laravel - Laravel Indonesia - Medium*. Medium; Laravel Indonesia. <https://medium.com/laravel-web-id/database-seeder-di-laravel-c839c34bfe0>
- SeventhQueen. (2019, November 14). *Lab-Informatika | Menggunakan Controller pada Laravel*. Lab-Informatika. <https://www.lab-informatika.com/menggunakan-controller-pada-laravel>