

LAPORAN PRAKTIKUM PEMROGRAMAN WEB 2

PERTEMUAN 4

CONTROLLER DAN MIGRATION



Disusun Oleh:

Wildan Dzaky Ramadhani

22/505766/SV/21917

Dosen Pengampu:

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

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

PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT LUNAK

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2023

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
BAB I.....	4
1.1. Latar Belakang.....	4
1.2. Rumusan Masalah.....	5
BAB II.....	6
2.1. Controller.....	6
2.2. Migration.....	6
BAB III.....	7
3.1. Membuat Controller.....	7
3.2. Membuat Migrasi Modul.....	8
BAB IV.....	13
4.1. Kesimpulan.....	13
DAFTAR PUSTAKA.....	14

DAFTAR GAMBAR

Gambar 1 Perintah membuat controller.....	7
Gambar 2 File Controller.....	7
Gambar 3 Fungsi Boomersport.....	7
Gambar 4 view boom.....	8
Gambar 5 route ke boom.....	8
Gambar 6 tampilan XAMPP.....	8
Gambar 7 Membuat database baru.....	9
Gambar 8 file environment.....	9
Gambar 9 connecting database.....	9
Gambar 10 Method up dan down.....	10
Gambar 11 membuat file migrasi.....	10
Gambar 12 file migrasi.....	10
Gambar 13 fungsi membuat tabel database.....	11
Gambar 14 perintah migrasi.....	11
Gambar 15 hasil migrasi.....	12

BAB I

PENDAHULUAN

1.1. Latar Belakang

Laporan praktikum pemrograman web ini membahas topik yang sangat penting dalam pengembangan aplikasi Laravel, yakni "Controller" dan "Migration". Dalam konteks pengembangan perangkat lunak, seringkali muncul permasalahan terkait manajemen basis data dan pengendalian aliran aplikasi. Permasalahan ini mencakup aspek seperti bagaimana mengelola perubahan skema database secara efisien serta mengendalikan logika aplikasi yang semakin kompleks.

Pentingnya penggunaan "Controller" dan "Migration" dapat dijelaskan oleh fakta bahwa keduanya menyediakan solusi terstruktur dan terorganisir untuk mengatasi permasalahan tersebut. "Migration" memungkinkan manajemen perubahan struktur database dengan baik, termasuk penambahan kolom baru, penghapusan tabel, atau perubahan tipe data. Ini memastikan aplikasi selalu sesuai dengan kebutuhan bisnis tanpa menimbulkan kompleksitas yang tidak perlu. Di sisi lain, "Controller" memisahkan logika aplikasi secara terstruktur, mempermudah pemeliharaan dan pengembangan lebih lanjut serta meningkatkan kejelasan dalam kode.

Praktikum ini sangat penting karena memperkenalkan konsep dasar dalam pengembangan aplikasi web menggunakan Laravel. Mahasiswa akan memahami cara efisien mengelola struktur database dan memisahkan logika aplikasi ke dalam komponen yang terstruktur dengan "Controller". Pengetahuan ini akan menjadi dasar yang kuat untuk memahami konsep lanjutan dalam pengembangan web serta memberikan mahasiswa keterampilan berharga di dunia kerja.

Diharapkan bahwa mahasiswa yang mengikuti praktikum ini akan memperoleh pemahaman mendalam tentang penggunaan "Controller" dan "Migration" dalam pengembangan aplikasi web dengan Laravel. Praktikum ini bertujuan agar mahasiswa mampu mengembangkan keterampilan teknis yang kuat dan mengaplikasikan pengetahuan tersebut dalam proyek nyata. Selain itu, praktikum ini juga diharapkan dapat meningkatkan minat mahasiswa dalam pengembangan web serta memberikan dasar yang kokoh untuk eksplorasi lebih lanjut dalam bidang ini.

1.2. Rumusan Masalah

- A. Bagaimana cara membuat *Controller* sederhana pada Laravel
- B. Bagaimana cara melakukan Migrasi pada Laravel

BAB II

LANDASAN TEORI

2.1. Controller

Controller adalah komponen yang sangat penting dalam pola desain Model-View-Controller (MVC). Fungsinya adalah menjadi perantara antara permintaan pengguna (View) dan data yang akan ditampilkan sebagai respons. Controller ini mengandung logika yang bertanggung jawab untuk menjalankan fungsi-fungsi khusus, seperti operasi Create, Read, Update, dan Delete (CRUD). Dalam pengembangan aplikasi web, Controller memiliki beberapa peran kunci.

2.2. Migration

Migration adalah salah satu fitur yang sangat berguna dalam Laravel. Anda bisa menganggapnya sebagai alat kontrol versi untuk basis data dalam aplikasi Anda. Fitur ini memungkinkan tim pengembangan web untuk bekerja secara kolaboratif dalam mengelola dan memodifikasi struktur basis data aplikasi mereka.

Migration sering digunakan bersama dengan Schema Builder yang disediakan oleh Laravel. Hal ini memungkinkan pengguna untuk merancang dan memodifikasi skema basis data aplikasi dengan mudah. Anda dapat membuat tabel, menambahkan kolom baru, menghapus tabel, atau mengubah tipe data dengan mudah.

BAB III

DOKUMENTASI PRAKTIKUM

3.1. Membuat Controller

Disini pertama kita akan membuat *Controller*. Kita dapat menuliskan perintah sebagai berikut untuk membuat.

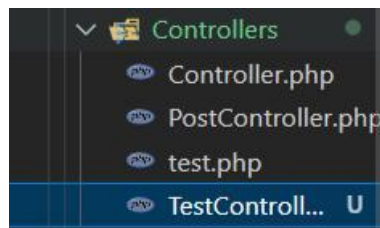
Php artisan make:controller PostController --resource

atribut *--resources* diatas berguna agar *controller* men-*generate* fungsi-fungsi dasar sejak dibuat.

```
PS C:\xampp\htdocs\example-app> php artisan make:controller TestController --resource
```

Gambar 1 Perintah membuat controller

Setelah perintah tersebut dijalankan, maka dalam *project* Laravel kita akan muncul file sebagai berikut.



Gambar 2 File Controller

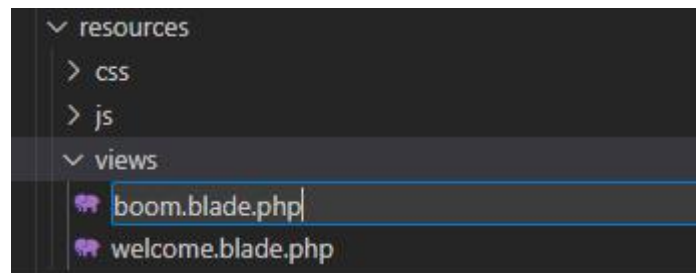
Selanjutnya kita buat fungsi baru pada *controller* yang telah kita buat seperti pada gambar dibawah ini.

```
0 references | 0 overrides
public function boomesport(string $id)
{
    return view('boom');
}
```

Gambar 3 Fungsi Boomersport

Fungsi diatas mengembalikan nilai yaitu *view boom*. Sehingga, ketika kita memanggil fungsi tersebut maka akan menampilkan halaman *boom*.

Sebelum kita memanggil view boom tersebut, kita buat terlebih dahulu file blade dengan nama “boom.blade.php” dalam folder *resources/view*.



Gambar 4 view boom

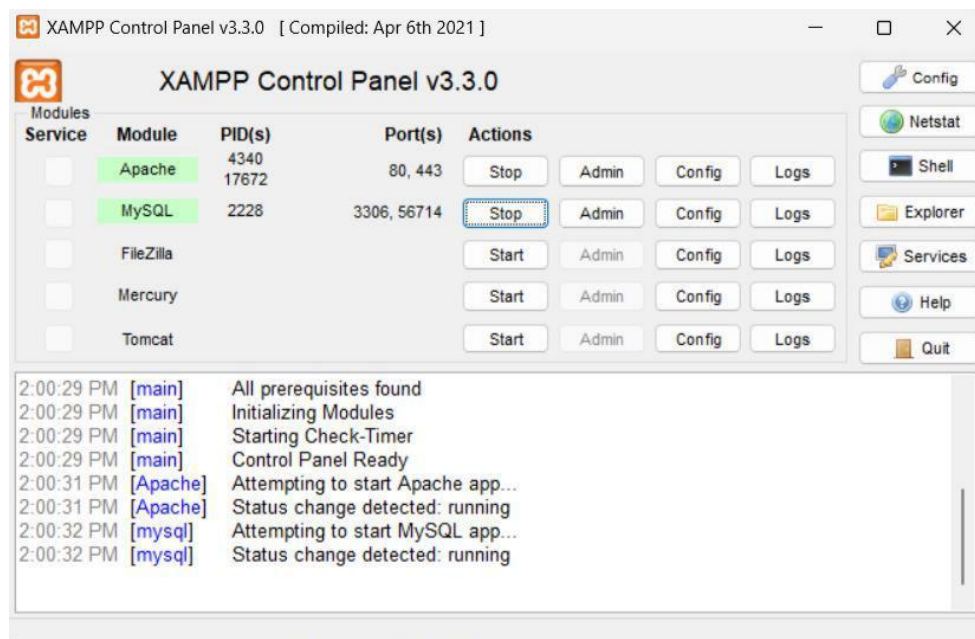
Selanjutnya kita buat route pada file web.php dalam folder *routes*. Sehingga ketika Alamat ‘boom’ dicapai, maka akan memanggil fungsi *boomsport* pada *PostController* yang akan mengembalikan nilai berupa view boom yang telah kita buat tadi.

```
Route::get('/boom',[PostController::class,'boomesport']);
```

Gambar 5 route ke boom

3.2. Membuat Migrasi Modul

Untuk langkah pertama yang kita lakukan adalah membuka aplikasi XAMPP dan start pada apache dan mysql untuk dapat mengakses database melalui phpMyAdmin.



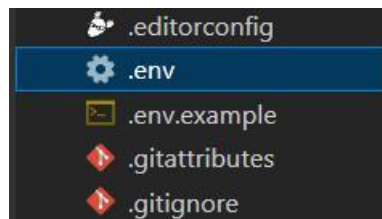
Gambar 6 tampilan XAMPP

Selanjutnya kita buat database baru, disini saya akan menggunakan nama 'ppw2'



Gambar 7 Membuat database baru

Selanjutnya kita melakukan setting pada directory .env yang kita gunakan pada Laravel. Kita dapat mengubah value dari variabel DB_DATABASE pada file .env dengan nama database yang ingin kita gunakan seperti pada gambar dibawah ini.



Gambar 8 file environment

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ppw2
DB_USERNAME=root
DB_PASSWORD=
```

Gambar 9 connecting database

Untuk membuat file migrasi pada *project* Laravel kita dapat menuliskan perintah sebagai berikut.

Php artisan make:migration create_ppw2

Dengan menambahkan atribut *create* sebelum nama tabel, maka dalam file tabel yang terbentuk akan ada method *default* yaitu *up* yang berfungsi untuk membentuk tabel dan *down* untuk menghapus tabel. Contohnya adalah sebagai berikut

```

public function up(): void
{
    Schema::create('perumahan', function (Blueprint $table) {
        $table->increments("id_perumahan");
        $table->integer("luas_perumahan");
        $table->integer("harga_perumahan");
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('peumahan');
}
};

```

Gambar 10 Method up dan down

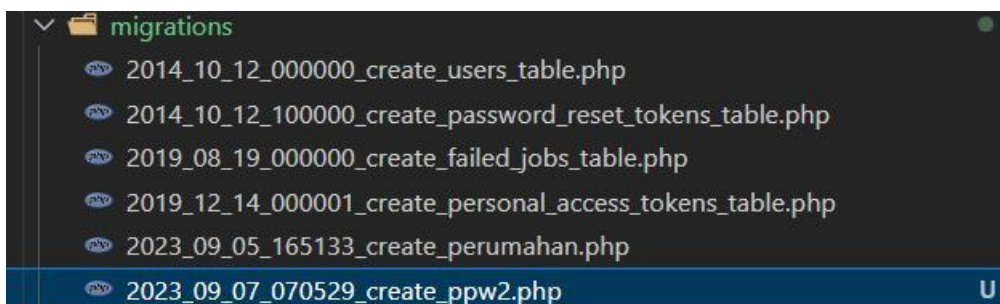
```

PS C:\xampp\htdocs\example-app> php artisan make:migration create_ppw2

```

Gambar 11 membuat file migrasi

Perintah tersebut setelah dijalankan maka pada project Laravel kita akan muncul file seperti pada gambar berikut.



Gambar 12 file migrasi

File tersebut merupakan tempat dimana kita membuat file database yang *default*-nya memiliki fungsi untuk membuat dan menghapus tabel. Sesuai modul kita buat file database tabel perumahan dengan isi sebagai berikut.

```

database > migrations > 2023_09_07_070529_create_ppw2.php
1  k?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('ppw2', function (Blueprint $table) {
15             $table->id();
16             $table->timestamps();
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::dropIfExists('ppw2');
26     }
27 };
28

```

Gambar 13 fungsi membuat tabel database

Selanjutnya kita dapat menuliskan perintah seperti pada gambar dibawah ini.

Php artisan migrate

Perintah tersebut bertujuan untuk melakukan migrasi sehingga kita dapat mengintegrasikan tabel pada Laravel kita dengan database yang kita gunakan.

```

PS C:\xampp\htdocs\example-app> php artisan migrate

INFO  Preparing database.

Creating migration table ..... 29ms DONE

INFO  Running migrations.

2014_10_12_000000_create_users_table ..... 30ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 41ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 28ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 45ms DONE
2023_09_05_165133_create_perumahan ..... 13ms DONE
2023_09_07_070529_create_ppw2 ..... 18ms DONE

```

Gambar 14 perintah migrasi

Setelah kita menjalankan perintah tadi, maka pada database kita akan otomatis terbarukan seperti pada gambar dibawah ini.

id_perumahan	luas_perumahan	harga_perumahan	created_at	updated_at
--------------	----------------	-----------------	------------	------------

Gambar 15 hasil migrasi

BAB IV

KESIMPULAN

4.1. Kesimpulan

Praktikum ini berfokus pada penggunaan "Controller" dan "Migration" dalam Laravel adalah langkah penting dalam memahami pengembangan aplikasi web modern. "Controller" berfungsi sebagai otak aplikasi yang mengatur alur logika bisnis dan merespons permintaan pengguna, sementara "Migration" memungkinkan pengelolaan struktur basis data dengan terstruktur. Dalam praktikum ini, mahasiswa belajar cara mengelola dan mengatur basis data dengan efisien, serta bagaimana memisahkan logika aplikasi menjadi komponen yang terstruktur. Kemampuan ini menjadi fondasi yang kuat untuk mengembangkan aplikasi web yang kuat, modular, dan siap berkembang. Praktikum ini membekali mahasiswa dengan keterampilan penting dalam pengembangan web dan mempersiapkan mereka untuk menghadapi tuntutan dunia kerja yang terus berkembang di bidang teknologi informasi.

DAFTAR PUSTAKA

<https://ojs.uajy.ac.id/index.php/jbi/article/view/766/727>

<https://centraldatatech.com/id/news-detail/cek-seluk-beluk-migrasi-database-dan-keuntungannya-untuk-bisnis/>