

	MODUL PRAKTIKUM PEMROGRAMAN WEB LANJUT FITUR PADA LARAVEL	JTI Hal 1 of 11
---	--	-------------------------------

TUJUAN PRAKTIKUM

Tujuan :

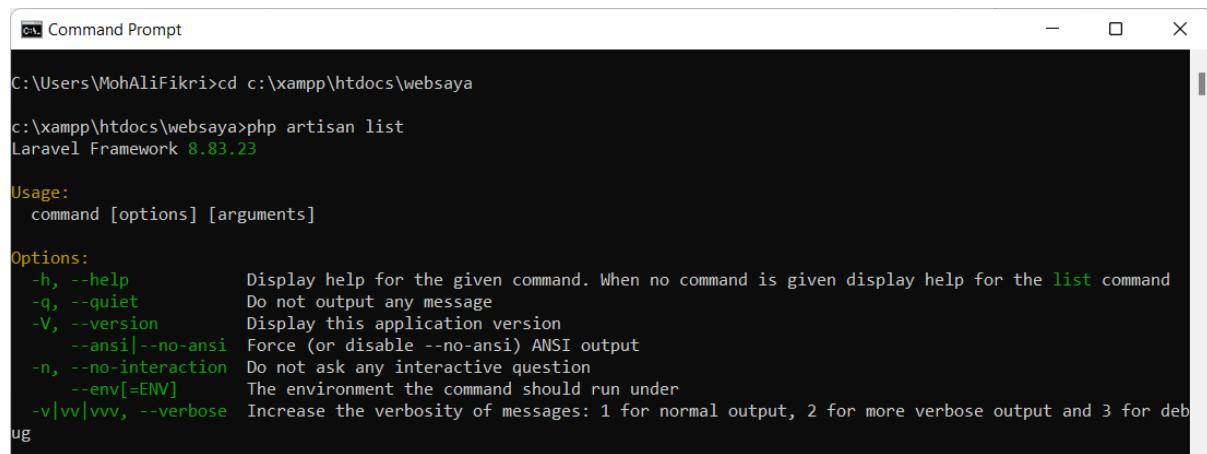
1. Mahasiswa dapat mengetahui fitur dalam framework php
2. Mahasiswa dapat memahami perintah artisan pada framework php
3. Mahasiswa dapat memahami perintah migration pada framework php
4. Mahasiswa dapat memahami perintah model pada framework php
5. Mahasiswa dapat memahami perintah seeder pada framework php

TEORI SINGKAT

Artisan merupakan command-line interface atau perintah-perintah yang diketikan pada command prompt untuk melakukan tugas tertentu saat proses pembuatan suatu aplikasi pada Laravel. Adapun cara untuk menggunakan perintah artisan adalah masuk ke dalam command prompt atau cmd pada windows atau terminal pada linux, lalu arahkan ke dalam folder projek laravel yang telah dibuat. Selanjutnya kita dapat mengetikan perintah-perintah artisan.

ARTISAN

Untuk melihat perintah-perintah artisan yang dapat digunakan, kita dapat mengetikan perintah `php artisan list` pada cmd atau command prompt, maka akan muncul list dari perintah-perintah artisan yang disediakan oleh laravel seperti pada gambar dibawah:



```

Command Prompt

C:\Users\MohAliFikri>cd c:\xampp\htdocs\websaya
c:\xampp\htdocs\websaya>php artisan list
Laravel Framework 8.83.23

Usage:
  command [options] [arguments]

Options:
  -h, --help          Display help for the given command. When no command is given display help for the list command
  -q, --quiet         Do not output any message
  -V, --version       Display this application version
  --ansi|--no-ansi   Force (or disable --no-ansi) ANSI output
  -n, --no-interaction Do not ask any interactive question
  --env[=ENV]          The environment the command should run under
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
  ug
  
```

Pada gambar di atas dapat dilihat sebagian perintah dari perintah-perintah yang disediakan oleh artisan berikut dengan penjelasnya. Kita dapat menampilkan penjelasan yang lebih detail dari sebuah perintah artisan dengan mengetikan perintah dalam format `php artisan help (nama perintah)` yang pada contohnya



nyatanya untuk melihat penjelasan dari perintah migrate yang ada pada perintah artisan , maka perintahnya adalah php artisan help migrate.

PRAKTIKUM

MEMBUAT MIGRATION

Migration dapat dikatan sebagai versi control untuk skema database pada sebuah aplikasi. Dengan bantuan migration, skema database dapat diperbaharui sehingga tetap up to date.

Migration juga memungkinkan kita tidak perlu membuka database client seperti PHPMyadmin untuk mengubah struktur database. Cukup mengubah kode pada migration lalu menjalankan perintah artisan, maka skema database akan secara otomatis akan berubah.

Kelebihan lain dari fitur ini adalah ketika ingi pindah database, misalnya dari Mysql ke Sqllite, kita tidak perlu membuat struktur database dari awal namun hanya perlu mengubah konfigurasi database Laravel, lalu menjalan perintah artisan migration. Untuk dapat mempraktekan fitur migration pada Laravel, terlebih dahulu kita harus membuat sebuah database melalui PHPMyAdmin misalnya dalam contoh ini database yang bernama “**websaya**”.

Selanjutnya ubah konfigurasi database dalam projek laravel pada file .env yang berada pada folder **websaya/.env** menjadi seperti berikut:

```
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=websaya
15 DB_USERNAME=root
16 DB_PASSWORD=
```



MEMBUAT TABEL

Untuk membuat migration, kita menggunakan artisan, jadi silahkan buka command prompt dan arahkan ke folder Laravel yang telah diinstal yang dalam contoh pada modul ini adalah projek Laravel dengan nama penjualan yang telah dibuat pada pertemuan sebelumnya, lalu ketikan script berikut:

```
c:\xampp\htdocs\websaya>php artisan make:migration create_mahasiswa_table
Created Migration: 2022_09_13_081701_create_mahasiswa_table
```

Hasil dari perintah artisan tersebut adalah akan ada file migration baru dengan nama **2022_09_13_081701_create_mahasiswa_table** pada folder **database/migration** sebagai berikut:

Selanjutnya, buka file **2022_09_13_081701_create_mahasiswa_table**, lalu ubah scriptnya menjadi seperti berikut:

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateMahasiswaTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('mahasiswa', function (Blueprint $table) {
17             $table->integer('nim')->primary();
18             $table->string('nama');
19             $table->string('prodi');
20             $table->year('angkatan');
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      *
27      * @return void
28      */
29     public function down()
30     {
31         Schema::dropIfExists('mahasiswa');
32     }
33 }
```



Pada file migration selalu ada dua method atau fungsi, yaitu up() dan down(). Method up() biasanya diisi script untuk membuat tabel, kolom, atau index pada database. Sedangkan method down() diisi script untuk mengembalikan operasi yang dilakukan oleh method up(). Pada script diatas kita membuat tabel dengan nama produks. Untuk membuat kolom tabelnya. Kita sebutkan dulu tipe datanya baru menuliskan nama kolomnya di dalam tanda kurung (dengan tanda petik karena merupakan string).

Sebelum menjalankan perintah migration kita harus melakukan sedikit konfigurasi pada file database.php yang ada pada folder projek laravel yang baru dibuat atau pada folder penjualan/config/database.php pada baris ke 53 ubahlah value strict dari true menjadi false seperti pada script dibawah :

```
46     'mysql' => [
47         'driver' => 'mysql',
48         'url' => env('DATABASE_URL'),
49         'host' => env('DB_HOST', '127.0.0.1'),
50         'port' => env('DB_PORT', '3306'),
51         'database' => env('DB_DATABASE', 'websaya'),
52         'username' => env('DB_USERNAME', 'root'),
53         'password' => env('DB_PASSWORD', ''),
54         'unix_socket' => env('DB_SOCKET', ''),
55         'charset' => 'utf8mb4',
56         'collation' => 'utf8mb4_unicode_ci',
57         'prefix' => '',
58         'prefix_indexes' => true,
59         'strict' => true,
60         'engine' => null,
61         'options' => extension_loaded('pdo_mysql') ? array_filter([
62             PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
63         ]) : [],
64     ],
65 ];
```

Untuk membuat tabel dari file migration diatas ketikan perintah artisan php artisan migrate pada commad prompt, dan hasil dari perintah artisan tersebut dapat dilihat pada gambar dibawah.

```
php artisan migrate
```

Perintah di atas akan menjalankan method run() pada file migration sehingga akan menghasilkan tabel pada database websaya. Untuk membuktikan hasilnya silahkan buka <http://localhost/phpmyadmin/> pilih database penjualan maka hasilnya akan tampak seperti gambar dibawah.



Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
mahasiswa		0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations		5	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_resets		0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
personal_access_tokens		0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
users		5	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
6 tables		Sum			176.0 KiB	0 B

Akan ada 4 tabel baru yang salah satunya adalah tabel produk yang dibuat dari file migration yang baru saja dibuat. Untuk tabel migrations, password_reset dan users adalah tabel-tabel default yang telah disediakan oleh laravel untuk melakukan autentikasi yang akan dijelaskan pada bagian selanjutnya.

ROLLING BACK MIGRATION

Untuk menghapus tabel, kita dapat menjalankan perintah berikut:

```
php artisan migrate:rollback
```

perintah artisan tersebut akan atau mengembalikan satu operasi atau operasi terakhir yang telah dilakukan seperti pada contoh:

```
c:\xampp\htdocs\websaya>php artisan migrate:rollback
Rolling back: 2022_09_13_081701_create_mahasiswa_table
Rolled back: 2022_09_13_081701_create_mahasiswa_table (6.71ms)
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back: 2019_12_14_000001_create_personal_access_tokens_table (2.95ms)
Rolling back: 2019_08_19_000000_create_failed_jobs_table
Rolled back: 2019_08_19_000000_create_failed_jobs_table (2.93ms)
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table (2.67ms)
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table (2.75ms)
```

Jika diperiksa pada localhost/phpmyadmin maka tabel yang baru saja dibuat menggunakan fitur migration akan terhapus. Untuk mengembalikan tabel-tabel tersebut jalankanlah perintah artisan.

```
php artisan migrate sekali lagi.
```

MODIFIKASI TABEL

Sebelumnya kita telah membuat tabel dengan migration menggunakan perintah **schema::create()**. Selain membuat tabel, kita juga dapat mengubah nama tabel menggunakan perintah **schema::rename()**. Untuk mencobanya, kita harus membuat migration dengan nama `ubah_tabel_produk` dengan script artisan sebagai berikut:

```
php artisan make:migration ubah_tabel_mahasiswa
```

Selanjutnya ubah method `up()` pada migration yang baru dibuat menjadi seperti berikut:

```
14   public function up()
15   {
16     Schema::rename('mahasiswa', 'mhs');
17   }
18
```

Pada script diatas akan, akan mengubah nama tabel mahasiswa menjadi tabel mhs. Untuk mengeksekusinya, jalankan perintah berikut:

```
c:\xampp\htdocs\websaya>php artisan migrate
Migrating: 2022_09_13_090550_ubah_tabel_mahasiswa
Migrated: 2022_09_13_090550_ubah_tabel_mahasiswa (5.78ms)
```

Sekarang nama tabel produk sudah berubah menjadi barang. Bisa dilihat pada gambar dibawah.

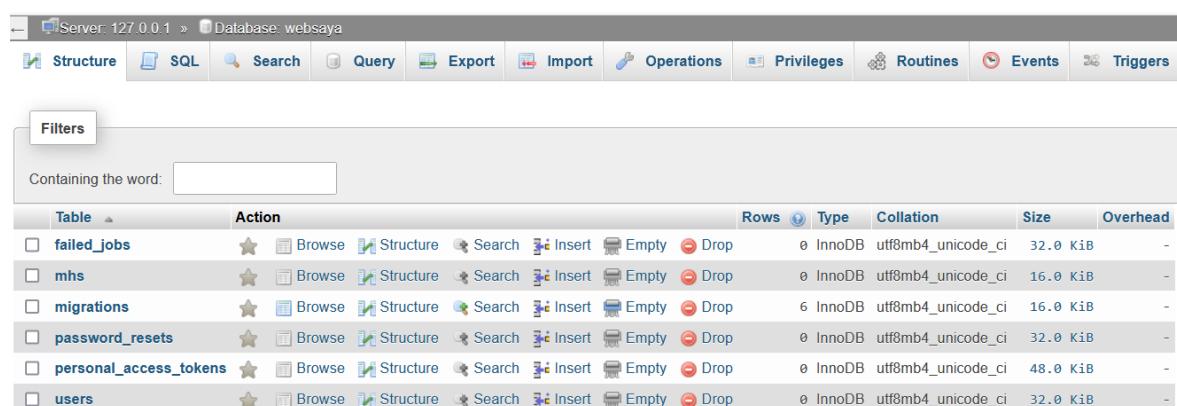


Table	Action	Rows	Type	Collation	Size	Overhead
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
mhs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-

Selain method `create()`, `drop()` dan `rename()`, masih ada method lain yang dapat digunakan untuk memodifikasi tabel, antara lain dapat dilihat pada tabel dibawah.



Tabel Method-method Untuk Memodifikasi Tabel Pada Migration

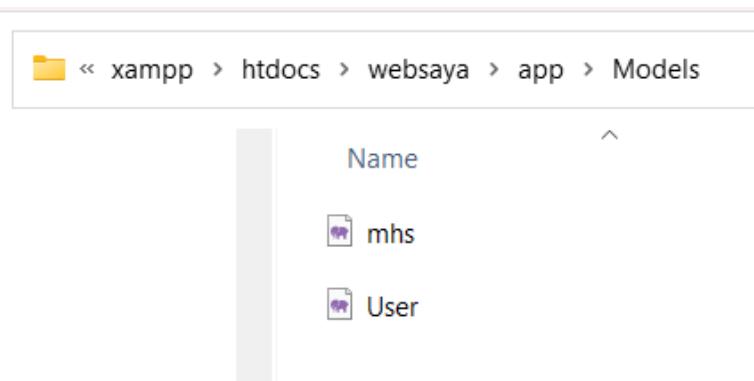
Contoh Script	Keterangan
If (Schema::hasTable('mhs')) { }	Untuk mengecek keberadaan suatu tabel
If (Schema::hasColumn('mhs', 'nama')) { }	Untuk mengecek keberadaan kolom pada suatu tabel
Schema::dropIfExists('mhs');	Untuk menghapus nama tabel Ketika nama tabel ditemukan

MEMBUAT MODEL

Model merupakan bagian dari konsep MVC pada Laravel yang fungsinya untuk berinteraksi dengan database. Model dibuat berdasarkan nama tabel pada aplikasi yang dibangun. Misalnya sebelumnya kita telah membuat satu tabel yang bernama barang. Untuk membuat model untuk table barangs kita dapat menjalankan perintah artisan dengan scipt sebagai berikut:

```
php artisan make:model mhs
```

hasil dari perintah di atas, akan ada satu file baru pada folder websaya/app/Models dengan nama **mhs.php** hasil dari perintah artisan di atas dapat dilihat pada gambar dibawah.





Laravel mengamsumsikan bahwa nama dari file model yang dibuat berarti nama tabel yang ada dalam database adalah plural dari nama model tersebut yang berarti jika nama file model disini adalah mhs maka nama tabel yang ada di dalam database adalah mhs. Namun hal tersebut dapat dikonfigurasi dengan cara membuka file model mhs.php yang baru saja dibuat dan tambahkan script berikut:

```
op > Models > mhs.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class mhs extends Model
9  {
10     protected $table = 'mhs';
11 }
12
```

Di dalam class model mhs kita mendefinsikan satu variabel bernama table, value dari variabel ini akan dianggap sebagai nama tabel yang diwakili oleh model ini.

SEEDER

Seeder digunakan untuk membuat contoh data pada database. Fitur ini sangat bermanfaat saat melakukan pengembangan suatu sistem yang dimana kita memerlukan suatu contoh data. Apalagi ketika membutuhkan contoh data yang banyak, fitur ini dapat membantu dari pada memasukan data satu persatu secara manual melalui PhpMyAdmin.

KEGIATAN PRAKTIKUM MEMBUAT SEEDER

Untuk membuat seeder, gunakan perintah artisan pada command prompt sebagai berikut:

```
php artisan make:seeder mhsTableSeeder
```

Perintah tersebut akan menghasilkan satu file baru pada folder database/seeds dengan nama **mhsTableSeeder.php** atau yang diketikan pada perintah. Hasil dari perintah artisan diatas dapat dilihat pada gambar dibawah.



File Explorer Screenshot:

- database > seeders > DatabaseSeeder.php
- database > seeders > mhsTableSeeder.php

Moh Ali Fikri

Selanjutnya buka file mhsTableSeeder yang telah dibuat dan tambahkan script pada method run() sehingga menjadi seperti berikut:

```
database > seeders > mhsTableSeeder.php > ...
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Seeder;
6  use Illuminate\Support\Facades\DB;
7
8  class mhsTableSeeder extends Seeder
9  {
10     /**
11      * Run the database seeds.
12      *
13      * @return void
14      */
15     public function run()
16     {
17         DB::table('mhs')->insert(array(
18             ['nim' => '22030001',
19              'nama' => 'Nadia',
20              'prodi' => 'D3TI',
21              'angkatan' => '2022'],
22              ['nim' => '22030002',
23              'nama' => 'Zahra',
24              'prodi' => 'D3TI',
25              'angkatan' => '2022']
26          ));
27     }
28 }
```

Langkah berikutnya, buka file DatabaseSeeder.php yang ada pada folder database/seeds dan panggil seeder yang baru dibuat pada method run dengan menambahkan script sebagai berikut:



```
atabase > seeders > 🐘 DatabaseSeeder.php > ...
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     * @return void
13     */
14    public function run()
15    {
16        $this->call(mhsTableSeeder::class);
17    }
18 }
```

Sekarang kita bisa menjalankan perintah artisan pada command prompt untuk mengeksekusi seeder yang telah dibuat dengan perintah sebagai berikut:

```
c:\xampp\htdocs\websaya>php artisan db:seed
Seeding: Database\Seeders\mhsTableSeeder
Seeded: Database\Seeders\mhsTableSeeder (15.00ms)
Database seeding completed successfully.
```

Untuk melihat hasil dari seeder yang dibuat bukalah localhost/phpmyadmin pada browser lalu pilih database penjualan dan table produks maka hasilnya akan ada 2 data baru yang dihasilkan dari pembuatan seeder yang telah kita melakukan seperti pada gambar dibawah:



**MODUL PRAKTIKUM
PEMROGRAMAN WEB LANJUT
FITUR PADA LARAVEL**

JTI

Hal 11 of 11

Server: 127.0.0.1 » Database: websaya » Table: mhs

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#)

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

```
SELECT * FROM `mhs`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this

+ Options

	nim	nama	prodi	angkatan
<input type="checkbox"/> Edit Copy Delete	22030001	Nadia	D3TI	2022
<input type="checkbox"/> Edit Copy Delete	22030002	Zahra	D3TI	2022

TUGAS

- Buatlah sebuah tabel **prodi** menggunakan **migration**
- Input data sesuai program studi POLINDRA pada tabel prodi menggunakan **seeder**

Catatan:

- Langkah-langkah pembuatannya dapat mengikuti praktikum
- Laporan praktikum tugas dan sourcode dikumpulkan ke elearning