

MODUL PEMROGRAMAN 5

Disusun oleh :

Tim Pengajar PRG 5

LP2M Politeknik Manufaktur Astra

JAKARTA

MODUL Mata Kuliah

Penulis :

Tim Pengajar PRG 5

Penerbit :

LP2M Politeknik Manufaktur Astra

Jl. Gaya Motor Raya No.8 Sunter II Jakarta 14330

Telepon: (021) 6519555 Fax: (021) 6519821

Email: sekretariat@polman.astra.ac.id

KATA PENGANTAR

Modul praktikum ini dibuat sebagai pedoman mahasiswa dalam melakukan kegiatan praktikum mata kuliah Pemrograman 5 di Politeknik Astra. Modul praktikum ini terdiri dari topik praktikum yang membahas tentang Membuat Model dan Migration dan Menampilkan Data. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan tujuan pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum mata kuliah Pemrograman 5 ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang. Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Jakarta, 10 November 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
M9 Migrasi Model dan Data.....	1
1. TUJUAN.....	1
2. PERCOBAAN.....	1
3. LATIHAN	7

M9 Migrasi Model dan Data

1. TUJUAN

Sub-CPMK : Mahasiswa mampu membuat web sederhana dengan menggunakan Laravel

2. PERCOBAAN

1. Konfigurasi Koneksi Database

Buka file .env dari project yang sudah dibuat sebagai berikut:

DB_CONNECTION	= mysql
DB_HOST	= 127.0.0.1
DB_PORT	= 3306
DB_DATABASE	= db_app11 (nama db disesuaikan masing-masing)
DB_USERNAME	= root
DB_PASSWORD	=

*note: jika belum ada db, silahkan buat db terlebih dahulu

2. Membuat Model dan Migrasi

a. Buat model dan migrasi dengan syntax berikut:

```
php artisan make:model Product -m
```

Keterangan

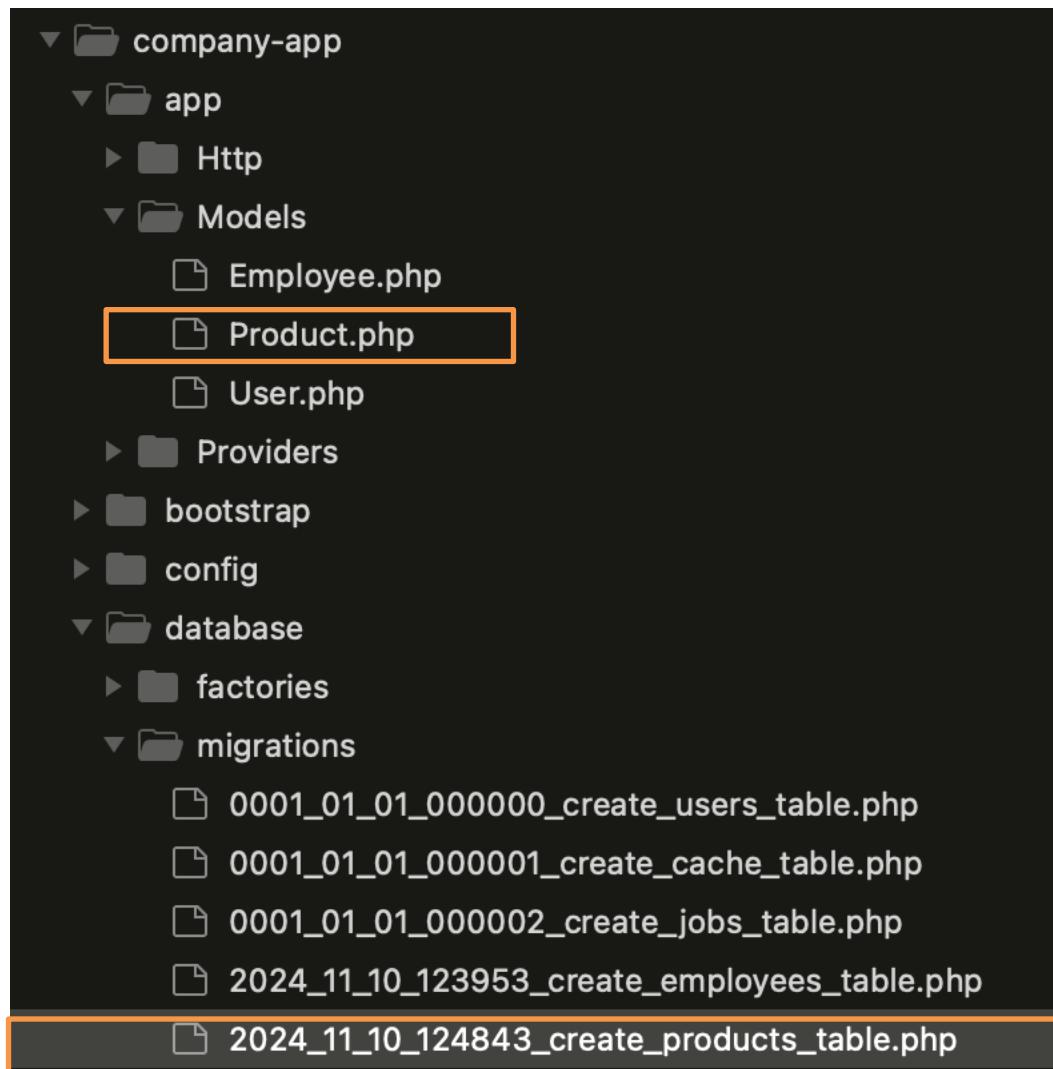
make:model : Perintah ini digunakan untuk membuat model baru dalam Laravel. Model di Laravel adalah representasi dari tabel di database yang memungkinkan kita untuk berinteraksi dengan data dalam tabel tersebut.

Product : Ini adalah nama model yang akan dibuat. Dalam hal ini, model tersebut akan diberi nama Product, yang biasanya sesuai dengan entitas "produk" dalam aplikasi.

-m : Opsi -m ini memberitahukan Laravel untuk **membuat file migrasi** secara otomatis saat model dibuat. File migrasi ini akan digunakan untuk mendefinisikan struktur tabel di database (misalnya, nama kolom dan tipe data).

b. Jika perintah berhasil, akan ada 2 file baru di dalam folder :

- App/Models/Produk.php
- database/migration/2024_11_10_124843_create_products_table.php



c. Tambah Field / kolom didalam migration

```
public function up(): void
{
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->string('image');
        $table->string('title');
        $table->text('description');
        $table->bigInteger('price');
        $table->integer('stock')->default(0);
        $table->timestamps();
    });
}
```

3. Menambahkan mass Assignment

Mass Assignment di Laravel memungkinkan untuk mengisi banyak kolom dalam sebuah tabel di database dengan sekali perintah, tanpa harus menyebutkan setiap kolom secara individual. Mass Assignment juga merupakan teknik dalam Laravel (dan framework lainnya) untuk mengisi banyak kolom dalam sebuah tabel database sekaligus, dan memungkinkan untuk memasukkan banyak data ke dalam model dalam satu perintah.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;

    /**
     * fillable
     *
     * @var array
     */
    protected $fillable = [
        'image',
        'title',
        'description',
        'price',
        'stock',
    ];
}
```

4. Menjalankan proses migrate

```
php artisan migrate
```

Jika muncul pertanyaan “**Would you like to create it?**”. Silahkan pilih YES.

Cek pada database, jika proses berhasil maka table-table akan berhasil digenerate beserta field-field didalamnya.

5. (Opsional) Rollback migration

Apabila ingin mengubah atau membatalkan proses migrasi, maka bisa dilakukan dengan cara rollback. Rollback dapat dilakukan dengan 2 cara.

- Rollback step sebelumnya

```
php artisan migrate:rollback
```

- Rollback step custom

```
php artisan migrate:rollback --step=3
```

6. Menampilkan data dari Database

- Membuat controller

```
php artisan make:controller ProductController
```

Maka file controller akan ditambahkan secara otomatis pada **app/Http/Controllers/ProductController.php**

- Ubah file controller sebagai berikut:

```
<?php

namespace App\Http\Controllers;

//import model product
use App\Models\Product;

//import return type View
use Illuminate\View\View;

class ProductController extends Controller
{
    /**
     * index
     *
     * @return void
     */
    public function index() : View
    {
        //get all products
        $products = Product::latest()->paginate(10);

        //render view with products
        return view('products.index', compact('products'));
    }
}
```

- Membuat route pada **routes/web.php**

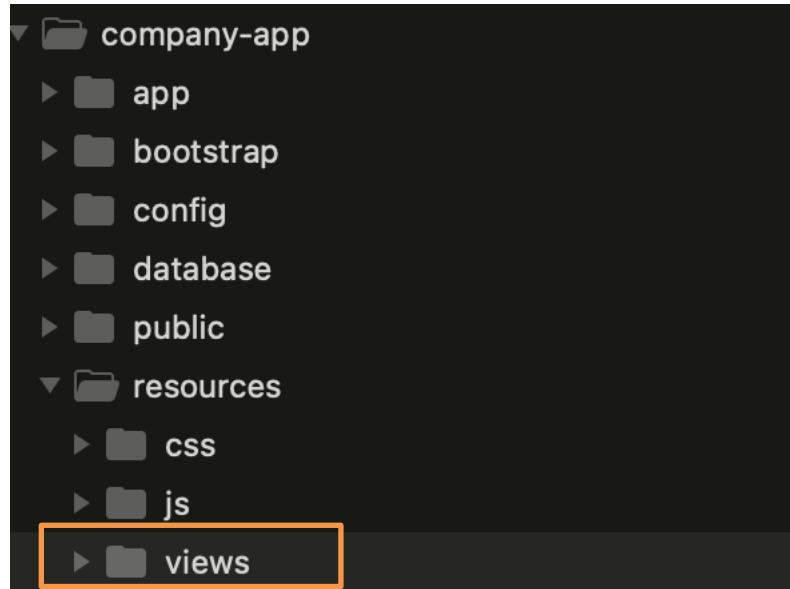
```
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

//route resource for products
Route::resource('/products', \App\Http\Controllers\ProductController::class);
```

- d. Membuat tampilan view buat file baru pada resource/views dengan nama **index.blade.php**.



Isi file view sebagai berikut:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Data Products - NIM(XXX) NAMA:</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body style="background-color: #f0f0f0">
  <div class="container mt-5">
    <div class="row">
      <div class="col-md-12">
        <div class="text-center my-4">Data Produk Warung 2A</div>
        <div class="text-center"><a href="#">Mata Kuliah Pemrograman 5</a></div>
      </div>
      <div class="card border-0 shadow-sm rounded">
        <div class="card-body">
          <form method="POST" action="{{ route('products.create') }}" class="mt-3">ADD PRODUCT</form>
          <table class="table-bordered">
            <thead>
              <tr>
                <th>IMAGE</th>
                <th>TITLE</th>
                <th>PRICE</th>
                <th>STOCK</th>
                <th>ACTIONS</th>
              </tr>
            </thead>
            <tbody>
              @foreach ($products as $product)
                <tr>
                  <td class="text-center">
                    
                  </td>
                  <td>{{ $product->title }}</td>
                  <td>{{ "Rp ". number_format($product->price,2,',','.') }}</td>
                  <td>{{ $product->stock }}</td>
                  <td class="text-center">
                    <form onsubmit="return confirm('Apakah Anda Yakin ?'); "> action="{{ route('products.destroy', $product->id) }}" method="POST">
                      <input type="hidden" name="route" value="{{ route('products.show', $product->id) }}"/>
                      <a href="#" class="btn btn-sm btn-dark">SHOW</a>
                      <a href="{{ route('products.edit', $product->id) }}" class="btn btn-sm btn-primary">EDIT</a>
                    @csrf
                    @method('DELETE')
                    <button type="submit" class="btn btn-sm btn-danger">HAPUS</button>
                  </form>
                </td>
              @endforeach
            </tbody>
          </table>
        <div class="alert alert-danger">
          Data Products belum tersedia.
        </div>
      </div>
    </div>
  </div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
  
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

<script>
    //message with sweetalert
    @if(session('success'))
        Swal.fire({
            icon: "success",
            title: "BERHASIL",
            text: "{{ session('success') }}",
            showConfirmButton: false,
            timer: 2000
        });
    @elseif(session('error'))
        Swal.fire({
            icon: "error",
            title: "GAGAL!",
            text: "{{ session('error') }}",
            showConfirmButton: false,
            timer: 2000
        });
    @endif
</script>

</body>
</html>
```

7. Silahkan jalankan project dengan perintah php artisan serve.

3. LATIHAN

1. Ubah view yang sudah dibuat menggunakan template sekreatif mungkin.
 2. Tambahkan data produk pada database sebanyak 10 baris, tampilkan list daftar produk tersebut.
-

Catatan:

Kumpulkan project anda dalam satu file zip dengan nama PRG5_20241_Laravel2_XXXXXXXXXX.zip (XXXXXXXXXX adalah NIM lengkap Anda)