

Praktikum 9 - Pemrograman Web

Program Studi: Teknik Informatika

Lakukan praktikum dibawah ini, dan buat screenshot untuk pembuktian mengerjakan setiap poin dengan mengisi tabel dibawah, kemudian tunjukkan hasil akhir dari men-share repository github yang telah dibuat.

Praktikum 9 adalah lanjutan dari praktikum 8, anda harus menyelesaikan praktikum 8 agar bisa lanjut ke praktikum 9.

A. Membuat view dan Form input

1. Aktifkan Apache, dan database mysql anda pada xampp control panel,
2. Pada laravel edit view **list_produk.blade.php** (Lanjutan kode minggu yang lalu)
3. Buatlah form input sederhana seperti pada gambar dibawah ini

```
26 <div> <h1>Input Produk</h1></div>
27 <form method="POST" action="{{ route('produk.simpan') }}">
28 <table class="table">
29 <tr>
30 <td>Nama:</td>
31 <td colspan="3"><input type="text" class="form-control" id="nama" name="nama"></td>
32 </tr>
33 <tr>
34 <td>Deskripsi:</td>
35 <td colspan="3"><textarea class="form-control" id="deskripsi" name="deskripsi"></textarea></td>
36 </tr>
37 <tr>
38 <td>Harga:</td>
39 <td colspan="3"><input type="number" class="form-control" id="harga" name="harga"></td>
40 <td></td>
41 <td></td>
42 </tr>
43 </table>
44 <button type="submit" class="btn btn-primary">Simpan</button>
45 </form>
```

Pada bagian `<form method="POST">` perhatikan action menuju **route produk.simpan** ini nanti akan dibuatkan kedalam routes dengan nama produk.simpan

4. Pastikan akan menghasilkan view seperti ini di browser

Input Produk

Nama:

Deskripsi:

Harga:

B. Membuat Routes

1. Buatlah routes baru untuk menerima request post dengan produk.simpan, seperti pada gambar dibawah ini

```
routes > web.php
You, 3 hours ago | 2 authors (You and others) | Click here to ask Blackbox to help you code faster
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\HomeController;
5  use App\Http\Controllers\SalesController;
6  use App\Http\Controllers\ListProdukController;
7
8
9  Route::get('/listproduk', [ListProdukController::class, 'show'] );
10 Route::post('/listproduk', [ListProdukController::class, 'simpan'])->name('produk.simpan');
11
```

2. Tulis route baru di line 10 pada gambar dengan method post, dan memanggil function 'simpan' kemudian set name produk.simpan sama seperti pada view

C. Mengangkap Request Post pada Controller

1. Buatlah function simpan pada controller ListProdukController untuk menangkap request POST yang masuk dari view yang sebelumnya sudah dibuat

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Produk;
class ListProdukController extends Controller
{
    public function simpan(Request $request)
    {
        $produk = new Produk;
        $produk->nama = $request->input('nama');
        $produk->deskripsi = $request->input('deskripsi');
        $produk->harga = $request->input('harga');
        $produk->save();

        return redirect()->back()->with('success', 'Data berhasil disimpan!');
    }
}
```

2. Tambah model use App\Models\Produk
3. Buat function simpan dengan argument Request
4. Create new produk sebagai inisiasi menginsert data baru ke database
5. Kemudian Save.
6. Jika sudah coba lewat browser untuk menambahkan data baru


Input Produk

Nama:

Deskripsi:

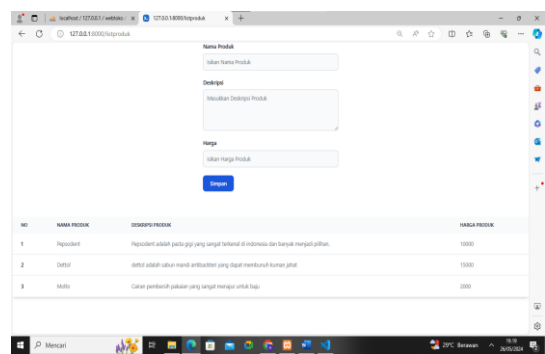
Harga:

7. Pada saat menekan tombol simpan, kemungkinan akan mengalami error karena 'created_at' dan 'update at' tidak ada field di dalam database.
8. Untuk menangani ini maka perlu menonaktifkan timestamp pada Laravel dengan cara
9. Masuk ke model Produk pada file /app/Models/Produk.php yang sebelumnya dibuat tambahkan \$timestamp buat menjadi false.
10. Setelah itu ulangi langkah ke-6 untuk update.

```
app > Models >  Produk.php
  ⚡ Click here to ask Blackbox to help you code faster
1  <?php
2  namespace App\Models;
3  use Illuminate\Database\Eloquent\Factories\HasFactory;
4  use Illuminate\Database\Eloquent\Model;
5  class Produk extends Model
6  {
7      use HasFactory;
8      protected $table = 'tblproduk';
9      public $timestamps = false;
10 }
```

1. Pelajari penyajian view diatas, dan sajikan data produk anda dengan menggunakan template tailwind dengan tampilan yang lebih menarik dengan menggunakan template.
2. SCREENSHOT hasil tabel produk dengan template, akan dijadikan penilaian akhir di praktikum ke-8 ini.

Hasil Pengerjaan

No.	Instruksi	Screenshot	Kendala/Saran
A.	Extends - Layout		
1.	A. Membuat view dan Form input B. Membuat Routes C. Mengangkap Request Post pada Controller		
2.			
...			
B.	Components → Include		
1.			
2.			
...			