

MODUL PEMROGRAMAN 5

**Disusun oleh :
Tim Pengajar PRG5**

**LP2M Politeknik Manufaktur Astra
JAKARTA**

MODUL Mata Kuliah

Penulis :

Dosen Pengampu

Instruktur

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 1 topik praktikum yang membahas tentang Framework Laravel. 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, 25 September 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI.....	iii
M11 Create pada Framework Laravel	4
1. TUJUAN.....	4
2. DURASI WAKTU	4
3. DASAR TEORI.....	4
4. PERCOBAAN	5
5. LATIHAN	10

M11 Create pada Framework Laravel

1. TUJUAN

CPMK: Create pada Framework Laravel

Sub-CPMK:

- Mahasiswa mampu menerapkan framework PHP

2. DURASI WAKTU

1 pertemuan x 3 jam

3. DASAR TEORI

Modul Pemrograman 5 Week 11

4. PERCOBAAN

Percobaan 1 : Menambahkan data ke dalam database

1. Langkah pertama jika ingin menambahkan data ke dalam database adalah menambahkan method create dan store di controller. Maka silakan buka file app/Http/Controllers/ProductController.php
2. Import return type RedirectResponse dan Request pada file app/Http/Controllers/ ProductController.php seperti berikut ini

```
use Illuminate\Http\Request;  
  
use Illuminate\Http\RedirectResponse;
```

3. Tambahkan fungsi create pada file app/Http/Controllers/ ProductController.php seperti berikut ini

```
public function create(): View  
{  
    return view('products.create');  
}
```

4. Tambahkan fungsi store pada file app/Http/Controllers/ ProductController.php seperti berikut ini

```
public function store(Request $request): RedirectResponse  
{  
    $request->validate([  
        'image'      => 'required|image|mimes:jpeg,jpg,png|max:2048',  
        'title'      => 'required|min:5',  
        'description' => 'required|min:10',  
        'price'      => 'required|numeric',  
        'stock'      => 'required|numeric'  
    ]);  
  
    $image = $request->file('image');  
    $imageName = $image->hashName();  
    $image->move('storage/products', $imageName);  
    $image = $request->file('image');
```

```

Product::create([
    'image'      => $imageName,
    'title'       => $request->title,
    'description' => $request->description,
    'price'       => $request->price,
    'stock'       => $request->stock
]);

return redirect()->route('products.index')->with(['success' => 'Data Berhasil Disimpan!']);
}

```

5. Membuat view form create product dengan cara membuat file baru dengan nama create.blade.php pada folder resources/view/products.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Add New Products</title>
8     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body style="background: lightgray">
11
12     <div class="container mt-5 mb-5">
13         <div class="row">
14             <div class="col-md-12">
15                 <div class="card border-0 shadow-sm rounded">
16                     <div class="card-body">
17                         <form action="{{ route('products.store') }}" method="POST" enctype="multipart/form-data">
18
19                             @csrf
20
21                             <div class="form-group mb-3">
22                                 <label class="font-weight-bold">IMAGE</label>
23                                 <input type="file" class="form-control @error('image') is-invalid @enderror" name="image"
24                                     accept="image/*" id="imgInp">
25                                 <img id="blah" src="" style="display: none;" alt="your image" width="150">
26                                 <!-- error message untuk image -->
27                                 @error('image')
28                                     <div class="alert alert-danger mt-2">
29                                         {{ $message }}
30                                     </div>
31                                 @enderror
32                             </div>
33
34                             <div class="form-group mb-3">
35                                 <label class="font-weight-bold">TITLE</label>
36                                 <input type="text" class="form-control @error('title') is-invalid @enderror" name="title"
37                                     value="{{ old('title') }}" placeholder="Masukkan Judul Product">
38
39                                 <!-- error message untuk title -->
40                                 @error('title')
41                                     <div class="alert alert-danger mt-2">
42                                         {{ $message }}
43                                     </div>

```

```

44 |         @enderror
45 |     </div>
46 |
47 |     <div class="form-group mb-3">
48 |         <label class="font-weight-bold">DESCRIPTION</label>
49 |         <textarea class="form-control @error('description') is-invalid @enderror" name="description"
50 |             rows="5" placeholder="Masukkan Description Product">{{ old('description') }}</textarea>
51 |
52 |         <!-- error message untuk description -->
53 |         @error('description')
54 |             <div class="alert alert-danger mt-2">
55 |                 {{ $message }}
56 |             </div>
57 |         @enderror
58 |     </div>
59 |
60 |     <div class="row">
61 |         <div class="col-md-6">
62 |             <div class="form-group mb-3">
63 |                 <label class="font-weight-bold">PRICE</label>
64 |                 <input type="number" class="form-control @error('price') is-invalid @enderror" name="price"
65 |                     value="{{ old('price') }}" placeholder="Masukkan Harga Product">
66 |
67 |                 <!-- error message untuk price -->
68 |                 @error('price')
69 |                     <div class="alert alert-danger mt-2">
70 |                         {{ $message }}
71 |                     </div>
72 |                 @enderror
73 |             </div>
74 |         </div>
75 |         <div class="col-md-6">
76 |             <div class="form-group mb-3">
77 |                 <label class="font-weight-bold">STOCK</label>
78 |                 <input type="number" class="form-control @error('stock') is-invalid @enderror" name="stock"
79 |                     value="{{ old('stock') }}" placeholder="Masukkan Stock Product">

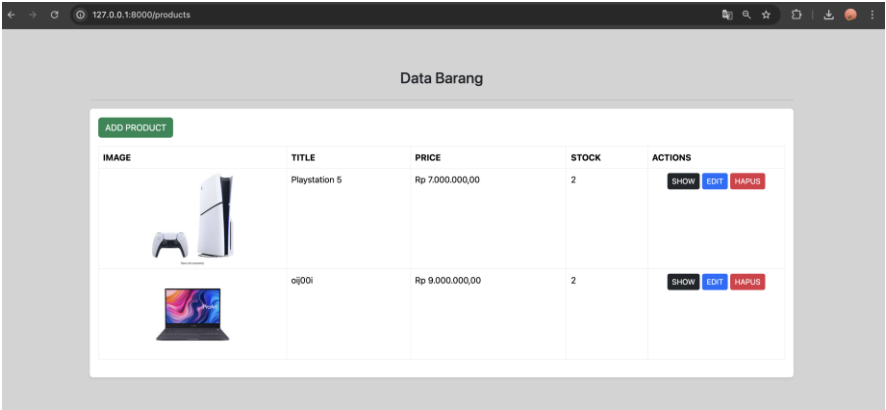
```

```

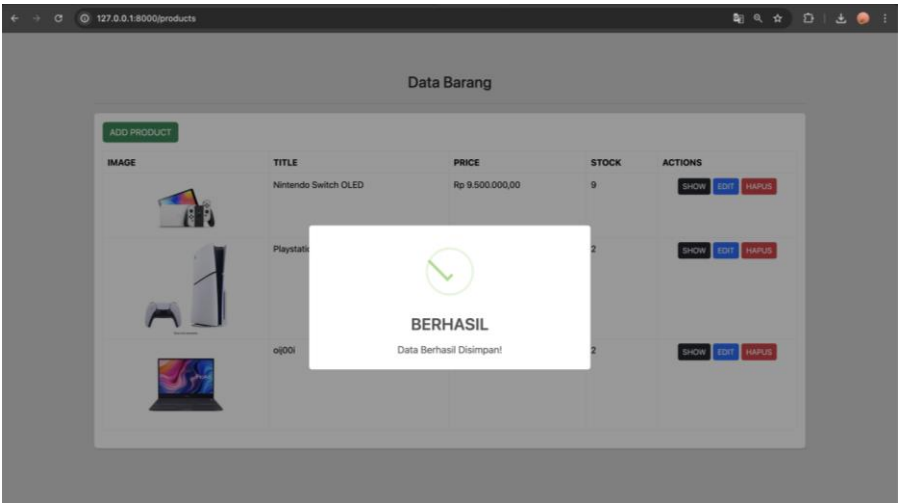
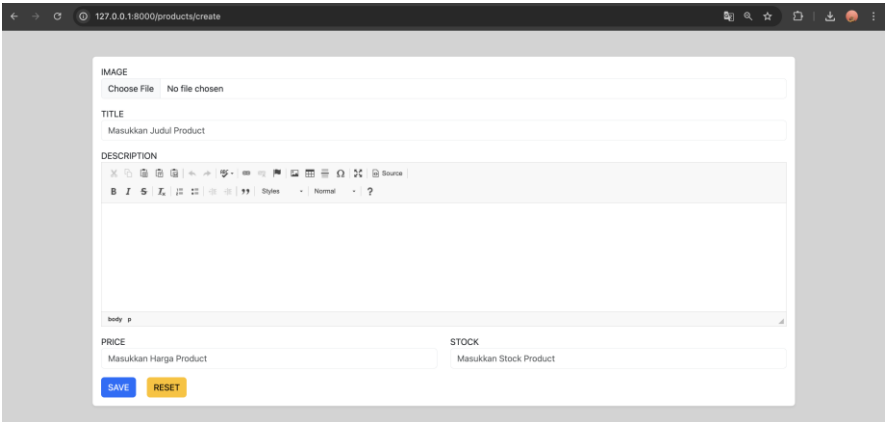
80 |
81 |                 <!-- error message untuk stock -->
82 |                 @error('stock')
83 |                     <div class="alert alert-danger mt-2">
84 |                         {{ $message }}
85 |                     </div>
86 |                 @enderror
87 |             </div>
88 |         </div>
89 |         <button type="submit" class="btn btn-md btn-primary me-3">SAVE</button>
90 |         <button type="reset" class="btn btn-md btn-warning">RESET</button>
91 |     </form>
92 | </div>
93 | </div>
94 | </div>
95 | </div>
96 | </div>
97 | <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
98 | <script src="https://cdn.ckeditor.com/4.13.1/standard/ckeditor.js"></script>
99 | <script>
100 |     CKEDITOR.replace( 'description' );
101 |     const imgInp = document.getElementById('imgInp');
102 |     const blah = document.getElementById('blah');
103 |     imgInp.onchange = evt => {
104 |         const [file] = imgInp.files;
105 |         if (file) {
106 |             blah.src = URL.createObjectURL(file); // Atur src menjadi URL gambar yang dipilih
107 |             blah.style.display = 'block'; // Tampilkan gambar
108 |         } else {
109 |             blah.style.display = 'none'; // Sembunyikan jika tidak ada file yang dipilih
110 |         }
111 |     }
112 | </script>
113 | </body>
114 | </html>

```


6. Hasil uji coba



Jika memilih add product, maka tampil halaman berikut



Percobaan 2 : Menampilkan detail data

1. Pada routes/web.php telah menggunakan resource sehingga menampilkan detail data termasuk dalam routes tersebut. Resource tersebut sama saja seperti code berikut. Sehingga dibebaskan mau menggunakan resource maupun ditulis satu persatu.

```
14 Route::get('/products', [ProductController::class, 'index'])->name('products.index'); // Menampilkan daftar produk
15 Route::get('/products/create', [ProductController::class, 'create'])->name('products.create'); // Menampilkan form tambah produk
16 Route::post('/products', [ProductController::class, 'store'])->name('products.store'); // Menyimpan produk baru
17 Route::get('/products/{id}', [ProductController::class, 'show'])->name('products.show'); // Menampilkan detail produk
18 Route::get('/products/{id}/edit', [ProductController::class, 'edit'])->name('products.edit'); // Menampilkan form edit produk
19 Route::put('/products/{id}', [ProductController::class, 'update'])->name('products.update'); // Mengupdate produk
20 Route::delete('/products/{id}', [ProductController::class, 'destroy'])->name('products.destroy'); // Menghapus produk
```

2. Tambahkan method show pada app/Http/Controllers/ProductController.php.

```
public function show(string $id): View
{
    $product = Product::findOrFail($id);
    return view('products.show', compact('product'));
}
```

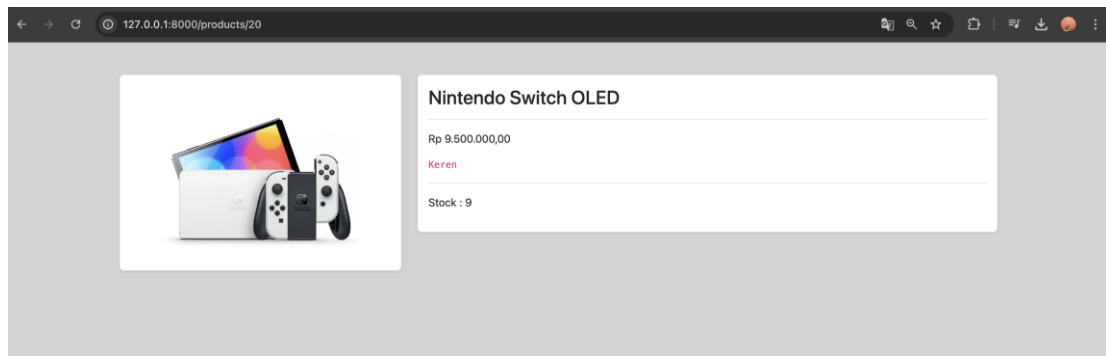
3. Membuat file view detail data product

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Show Products - SantriKoding.com</title>
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
9  </head>
10 <body style="background: lightgray">
11
12     <div class="container mt-5 mb-5">
13         <div class="row">
14             <div class="col-md-4">
15                 <div class="card border-0 shadow-sm rounded">
16                     <div class="card-body">
17                         
18                     </div>
19                 </div>
20             </div>
21             <div class="col-md-8">
22                 <div class="card border-0 shadow-sm rounded">
23                     <div class="card-body">
24                         <h3>{{ $product->title }}</h3>
25                         <hr/>
26                         <p>{{ "Rp " . number_format($product->price,2,',','.') }}</p>
27                         <code>
28                             <p>{!! $product->description !!}</p>
29                         </code>
30                         <hr/>
31                         <p>Stock : {{ $product->stock }}</p>
32                     </div>
33                 </div>
34             </div>
35         </div>
36     </div>
37     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
38 </body>
39 </html>

```

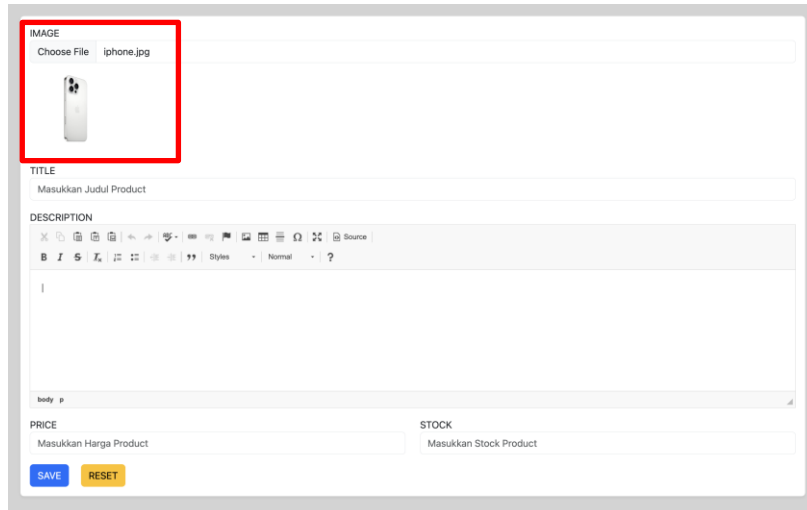
4. Uji coba



5. LATIHAN

Latihan 1 : Buat preview gambar setelah memilih file image

1. Tambahkan code agar gambar dapat dipreview sebelum disimpan pada database. Contoh tampilan dapat sebagai berikut.



The screenshot shows a web form for adding a product. A red rectangular box highlights the 'IMAGE' section at the top left. Inside this box, there is a 'Choose File' button and a preview of an image labeled 'iphone.jpg'. Below the image preview is a small thumbnail of an iPhone. The rest of the form includes fields for 'TITLE' (Masukkan Judul Product), 'DESCRIPTION' (with a rich text editor), 'PRICE' (Masukkan Harga Product), and 'STOCK' (Masukkan Stock Product). At the bottom are 'SAVE' and 'RESET' buttons.

Latihan 2 : Buat fitur tambahan untuk search dan export PDF

Catatan:

Kumpulkan project anda dalam satu file zip dengan nama PRG5_M11_P1_XX.zip (XX adalah 2 angka NIM terakhir Anda)