

LAPORAN PRAKTIKUM
PEMROGRAMAN APLIKASI BERBASIS WEB II
KELAS A1



Di Kerjakan Oleh :
228160049 / Nugraha Ramadan Diyanto
228160073 / Tyas Adrian
Instruktur : Rizki Muliono,S.Kom,M.Kom

PRODI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2025

DAFTAR ISI

DAFTAR ISI.....	i
INSTALASI DAN KONFIGURASI ENVIRONMENT LARAVEL	1
Task: Modul 1	1
Kode Program	1
Screen Shoot / Output Hasil Kode Program:	2
Kesimpulan Hasil / Solusi:.....	2
RUTE DAN CONTROLLER.....	4
Task: Modul 2	4
Kode Program	4
Screen Shoot / Output Hasil Kode Program:	5
Kesimpulan Hasil / Solusi:.....	5
VIEWS DAN BLADE TEMPLATING	6
Task: Modul 3	6
Kode Program	6
Screen Shoot / Output Hasil Kode Program:	9
Kesimpulan Hasil / Solusi:.....	10
MODEL, MIGRASI DAN INPUT FORM.....	11
Task: Modul 4	11
Kode Program	11
Screen Shoot / Output Hasil Kode Program:	14
Kesimpulan Hasil / Solusi:.....	16
Database dan Eloquent ORM.....	17
Task: Modul 5	17
Kode Program	17
Screen Shoot / Output Hasil Kode Program:	21
Kesimpulan Hasil / Solusi:.....	22
Autentikasi	23
Task: Modul 6	23
Kode Program	23
Screen Shoot / Output Hasil Kode Program:	26
Kesimpulan Hasil / Solusi:.....	28
Middleware dan Keamanan	29
Task: Modul 7	29
Kode Program	29

Screen Shoot / Output Hasil Kode Program:	30
Kesimpulan Hasil / Solusi:	31
Manajemen File Uploads	32
Task: Modul 8	32
Kode Program	32
Screen Shoot / Output Hasil Kode Program:	35
Kesimpulan Hasil / Solusi:	37
Scan QR & API Development	38
Task: Modul 9	38
Kode Program	38
Screen Shoot / Output Hasil Kode Program:	47
Kesimpulan Hasil / Solusi:	49
SWEETALERT DAN DATA TABLES	50
Task: Modul 10	50
Kode Program	50
Screen Shoot / Output Hasil Kode Program:	53
Kesimpulan Hasil / Solusi:	58
LEMBAR PERNYATAAN	59

INSTALASI DAN KONFIGURASI ENVIRONMENT LARAVEL

Task: Modul 1

1. Instalasi Laravel pada Sistem Kita:

- Lakukan instalasi Laravel dan Composer pada sistem Kita jika belum dilakukan. Pastikan Kita dapat menjalankan Laravel development server dan mengaksesnya melalui browser.

2. Eksplorasi Struktur Direktori Laravel:

- Eksplorasi dan buatlah laporan singkat mengenai struktur direktori Laravel dan fungsi utama dari masing-masing direktori tersebut.

3. Konfigurasi Database:

- Buatlah database baru pada sistem manajemen database yang Kita gunakan (misalnya MySQL). Kemudian, konfigurasikan file .env Laravel Kita untuk terhubung dengan database yang baru dibuat tersebut.

4. Kirim Project kalian ke github masing-masing mahasiswa.

5. Dokumentasikan ke dalam bentuk laporan praktikum.

Kode Program

konfigurasikan file .env Laravel

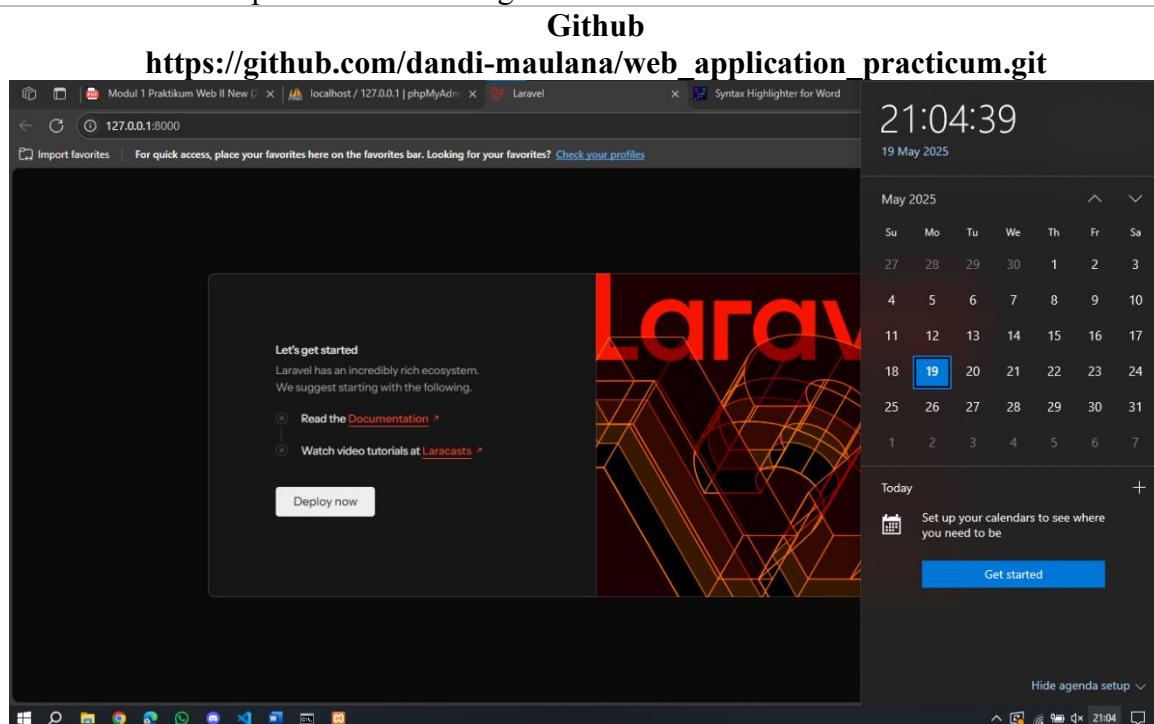
```
1 | APP_NAME=Laravel
2 | APP_ENV=local
3 | APP_KEY=base64:CinHeTBEMBWJzOCeLOPk6v4h/NmjOE9JhIr3Q0YSNoU=
4 | APP_DEBUG=true
5 | APP_URL=http://localhost
6 |
7 | APP_LOCALE=en
8 | APP_FALLBACK_LOCALE=en
9 | APP_FAKE_LOCALE=en_US
10|
11| APP_MAINTENANCE_DRIVER=file
12| # APP_MAINTENANCE_STORE=database
13|
14| PHP_CLI_SERVER_WORKERS=4
15|
16| BCRYPT_ROUNDS=12
17|
18| LOG_CHANNEL=stack
19| LOG_STACK=single
20| LOG_DEPRECATED_CHANNEL=null
21| LOG_LEVEL=debug
22|
23| DB_CONNECTION=mysql
24| DB_HOST=127.0.0.1
25| DB_PORT=3306
26| DB_DATABASE=laravel
27| DB_USERNAME=root
28| DB_PASSWORD=
29|
30| SESSION_DRIVER=database
31| SESSION_LIFETIME=120
32| SESSION_ENCRYPT=false
33| SESSION_PATH=/
34| SESSION_DOMAIN=null
35|
36| BROADCAST_CONNECTION=log
37| FILESYSTEM_DISK=local
38| QUEUE_CONNECTION=database
39|
```

```

40 | CACHE_STORE=database
41 | # CACHE_PREFIX=
42 |
43 | MEMCACHED_HOST=127.0.0.1
44 |
45 | REDIS_CLIENT=phpredis
46 | REDIS_HOST=127.0.0.1
47 | REDIS_PASSWORD=null
48 | REDIS_PORT=6379
49 |
50 | MAIL_MAILER=log
51 | MAIL_SCHEME=null
52 | MAIL_HOST=127.0.0.1
53 | MAIL_PORT=2525
54 | MAIL_USERNAME=null
55 | MAIL_PASSWORD=null
56 | MAIL_FROM_ADDRESS="hello@example.com"
57 | MAIL_FROM_NAME="${APP_NAME}"
58 |
59 | AWS_ACCESS_KEY_ID=
60 | AWS_SECRET_ACCESS_KEY=
61 | AWS_DEFAULT_REGION=us-east-1
62 | AWS_BUCKET=
63 | AWS_USE_PATH_STYLE_ENDPOINT=false
64 |
65 | VITE_APP_NAME="${APP_NAME}"
66 |

```

Screen Shoot / Output Hasil Kode Program:



Kesimpulan Hasil / Solusi:

Struktur Direktori Laravel dan Fungsi Utamanya

Laravel memiliki struktur direktori yang terorganisir dengan baik, yang memudahkan pengembang dalam mengelola kode. Berikut adalah penjelasan mengenai direktori utama dalam Laravel dan fungsi masing-masing:

1. Folder 'app/'

- Berisi logika bisnis aplikasi, termasuk model, controller, dan middleware.
- Sub-folder penting:
 - Console/: Menyimpan perintah artisan kustom.
 - Http/: Berisi controller dan middleware.
 - Models/: Menyimpan model yang berinteraksi dengan database.
- 2. Folder 'bootstrap/'
 - Berisi file yang digunakan untuk bootstrap framework dan menginisialisasi aplikasi.
 - File penting:
 - app.php: File utama untuk mem-bootstrap Laravel.
- 3. Folder 'config/'
 - Menyimpan file konfigurasi aplikasi, seperti pengaturan database, autentikasi, dan caching.
 - Contoh file konfigurasi:
 - app.php: Konfigurasi dasar aplikasi.
 - database.php: Konfigurasi database.
- 4. Folder 'database/'
 - Menyimpan semua yang berkaitan dengan database, termasuk:
 - migrations/: File migrasi untuk mengelola struktur database.
 - seeders/: File untuk mengisi database dengan data awal.
 - factories/: Class untuk membuat data dummy.
- 5. Folder 'public/'
 - Merupakan root akses aplikasi melalui browser.
 - Berisi:
 - index.php: Entry point utama aplikasi.
 - Folder untuk menyimpan file statis seperti CSS, JavaScript, dan gambar.
- 6. Folder 'resources/'
 - Menyimpan file sumber daya aplikasi, seperti tampilan Blade dan asset mentah.
 - Contoh isi:
 - views/: File tampilan untuk aplikasi.
 - js/dan css/: File JavaScript dan CSS.
- 7. Folder 'routes/'
 - Berisi semua rute aplikasi.
 - File utama:
 - web.php: Rute untuk aplikasi berbasis web.
 - api.php: Rute untuk API.
- 8. Folder 'storage/'
 - Menyimpan file yang dihasilkan oleh aplikasi saat runtime, termasuk log, cache, dan sesi.
 - Sub-folder:
 - logs/: Menyimpan log aplikasi.
 - app/: Tempat menyimpan file yang diunggah pengguna.
- 9. Folder 'tests/'
 - Digunakan untuk menyimpan file uji unit dan fitur aplikasi menggunakan PHPUnit.
- 10. Folder 'vendor/'
 - Berisi semua dependensi yang diunduh melalui Composer, termasuk Laravel itu sendiri.

File Penting di Root Proyek
 - .env: Berisi variabel lingkungan aplikasi.
 - composer.json: File konfigurasi untuk Composer.
 - artisan: File CLI untuk menjalankan perintah Laravel.

RUTE DAN CONTROLLER

Task: Modul 2

1) Membuat Rute Dasar:

- Buatlah rute dasar pada aplikasi Laravel Anda yang mengarah ke halaman selamat datang.
- Pastikan rute tersebut menampilkan pesan "Selamat datang di aplikasi Laravel!".

2) Membuat Rute dengan Parameter lebih dari 1:

- Buatlah rute baru yang menggunakan parameter untuk menyapa pengguna.
- Rute tersebut harus menerima parameter nama pengguna, npm dan menampilkan pesan "Halo, [nama_anda] [NPM]!".

3) Membuat Kontroler:

- Buatlah kontroler baru dengan nama GreetingsController.
- Kontroler ini akan memiliki dua metode: welcome() untuk menangani rute dasar, dan greet(\$name, \$npm) untuk menangani rute dengan parameter.
- Pastikan metode welcome() mengembalikan pesan "Selamat datang di aplikasi Laravel!" dan metode greet(\$name, \$npm) mengembalikan pesan "Halo, [nama_anda] [NPM_anda]!".

4) Menghubungkan Kontroler dengan Rute:

- Hubungkan rute dasar ke metode welcome() dalam kontroler GreetingsController.
- Hubungkan rute dengan parameter ke metode greet(\$name) dalam kontroler GreetingsController.

5) Uji Coba Aplikasi:

- Jalankan aplikasi Laravel Anda dan pastikan rute yang telah Anda definisikan berfungsi dengan baik.
- Pastikan Anda menerima respons yang sesuai sesuai dengan yang diharapkan. Update ke Git Hub anda.
- Dokumentasikan Kedalam Laporan Praktikum

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git

web.php

```
1 | <?php
2 |
3 | use App\Http\Controllers\GreetingsController;
4 | use App\Http\Controllers\HelloController;
5 | use Illuminate\Support\Facades\Route;
6 |
7 | Route::get('/', [HelloController::class, 'index']);
8 | Route::get('/hello/{name}', [HelloController::class, 'sayHello']);
9 |
10| Route::get('/welcome', [GreetingsController::class, 'welcome']);
11| Route::get('/hello/{name}/{npm}', [GreetingsController::class,
12| 'greet']);
13| Route::get('/', function () {
14|     return view('welcome');
15| });
16|
17| Route::get('/', function () {
18|     return 'Selamat Datang Di Aplikasi Laravel';
19| });
```

```

20|
21| Route::get('/hello/{name}', function ($name) {
22|     return 'Hello, ' . $name . '!';
23| });
24|
25|
26| Route::get('/hello/{name}/{npm}', function ($name , $npm) {
27|     return 'Hello, ' . $name . ' ' . $npm . '!';
28| });

```

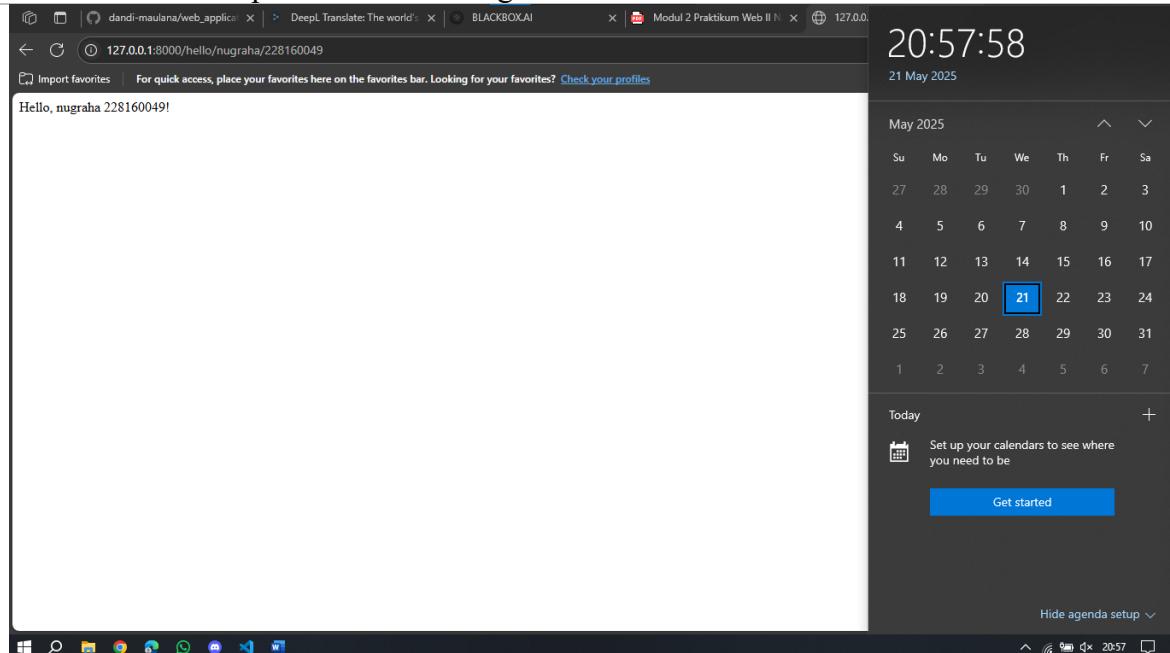
GreetingsController.php

```

1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 |
7 | class GreetingsController extends Controller
8 | {
9 |     public function welcome()
10|    {
11|        return 'Selamat Datang di aplikasi laravel!';
12|    }
13|
14|    public function greet($name, $npm)
15|    {
16|        return 'Hello, ' . $name . ' ' . $npm . '!';
17|    }
18| }
19|

```

Screen Shoot / Output Hasil Kode Program:



Kesimpulan Hasil / Solusi:

Membahas implementasi rute (route) dan controller dalam Laravel sebagai penghubung antara permintaan HTTP dengan logika aplikasi, di mana rute berfungsi menentukan endpoint URL sementara controller menangani proses bisnis, memungkinkan pengembangan aplikasi web yang terstruktur dan mudah dikelola dengan pemisahan concern yang jelas.

VIEWS DAN BLADE TEMPLATING

Task: Modul 3

Silahkan mengerjakan view untuk halaman website portfolio yang terdiri dari halaman:

- Home.blade.php
- Profil.blade.php
- Pendidikan.blade.php
- Keahlian.blade.php

dengan ketentuan :

1. profil menampilkan nilai variabel Nama dan NPM anda, nilai variabel berasal dari kontroller
2. pendidikan menampilkan variabel Kampus dan Prodi anda, nilai variabel berasal dari kontroller
3. keahlian menampilkan variabel skill anda, nilai variabel berasal dari kontroller
4. Buat masing-masing function / method untuk menampilkan tiap halaman dan buatkan route untuk meng akses tiap view tersebut.
5. Dokumentasikan hasil uji coba setiap URL dari tiap-tiap view tersebut ke dalam laporan praktikum.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git

Membuat Layout Utama (app.blade.php)

```
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 |   <title>@yield('title') - My Portfolio</title>
7 |   <link
8 |     href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.
.css" rel="stylesheet">
9 | </head>
10 | <body>
11 |   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
12 |     <div class="container">
13 |       <a class="navbar-brand" href="#">My Portfolio</a>
14 |       <div class="navbar-nav">
15 |         <a class="nav-link" href="/home">Home</a>
16 |         <a class="nav-link" href="/profil">Profil</a>
17 |         <a class="nav-link" href="/pendidikan">Pendidikan</a>
18 |         <a class="nav-link" href="/keahlian">Keahlian</a>
19 |       </div>
20 |     </div>
21 |
22 |     <div class="container mt-4">
23 |       @yield('content')
24 |     </div>
25 |
26 |     <footer class="bg-dark text-white text-center py-3 mt-5">
27 |       &copy; 2024 My Portfolio - Teknik Informatika UMA
28 |     </footer>
29 |
```

```

30|     <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
31|   </body>
32| </html>
                                Membuat Controller PortfolioController
1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 |
7 | class PortfolioController extends Controller
8 |
9 | {
10|     public function home()
11|     {
12|         return view('home');
13|
14|     public function profil()
15|     {
16|         $data = [
17|             'nama' => 'Nama Anda',
18|             'npm' => 'NPM Anda'
19|         ];
20|         return view('profil', $data);
21|     }
22|
23|     public function pendidikan()
24|     {
25|         $data = [
26|             'kampus' => 'Universitas Medan Area',
27|             'prodi' => 'Teknik Informatika'
28|         ];
29|         return view('pendidikan', $data);
30|     }
31|
32|     public function keahlian()
33|     {
34|         $data = [
35|             'skill' => ['PHP', 'Laravel', 'JavaScript', 'HTML/CSS',
'Bootstrap']
36|         ];
37|         return view('keahlian', $data);
38|     }
39| }

```

home.blade.php

```

1 | @extends('layouts.app')
2 |
3 | @section('title', 'Home')
4 |
5 | @section('content')
6 |     <h1>Selamat Datang di Portfolio Saya</h1>
7 |     <p>Ini adalah halaman utama portfolio saya yang dibuat
menggunakan Laravel Blade Templating.</p>
8 | @endsection

```

profil.blade.php

```

1 | @extends('layouts.app')
2 |
3 | @section('title', 'Profil')

```

```

4 | @section('content')
5 |     <h1>Profil Saya</h1>
6 |     <div class="card">
7 |         <div class="card-body">
8 |             <p><strong>Nama:</strong> {{ $nama }}</p>
9 |             <p><strong>NPM:</strong> {{ $npm }}</p>
10|         </div>
11|     </div>
12| 
```

13| @endsection

pendidikan.blade.php

```

1 | @extends('layouts.app')
2 |
3 | @section('title', 'Pendidikan')
4 |
5 | @section('content')
6 |     <h1>Riwayat Pendidikan</h1>
7 |     <div class="card">
8 |         <div class="card-body">
9 |             <p><strong>Kampus:</strong> {{ $kampus }}</p>
10|             <p><strong>Program Studi:</strong> {{ $prodi }}</p>
11|         </div>
12|     </div>
13| 
```

13| @endsection

keahlian.blade.php

```

1 | @extends('layouts.app')
2 |
3 | @section('title', 'Keahlian')
4 |
5 | @section('content')
6 |     <h1>Keahlian Saya</h1>
7 |     <div class="card">
8 |         <div class="card-body">
9 |             <h5 class="card-title">Skill:</h5>
10|             <ul class="list-group">
11|                 @foreach($skill as $s)
12|                     <li class="list-group-item">{{ $s }}</li>
13|                 @endforeach
14|             </ul>
15|         </div>
16|     </div>
17| 
```

17| @endsection

Membuat Route

```

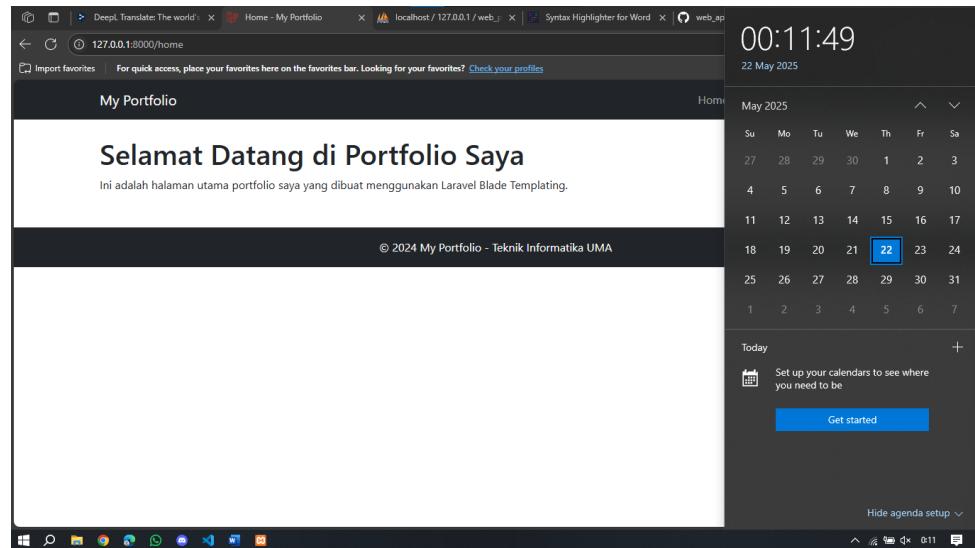
1 | <?php
2 |
3 | use App\Http\Controllers\GreetingsController;
4 | use App\Http\Controllers\HelloController;
5 | use Illuminate\Support\Facades\Route;
6 | use App\Http\Controllers\HomeController;
7 | Route::get('/home', [HomeController::class, 'index']);
8 |
9 | Route::get('/', [HelloController::class, 'index']);
10| Route::get('/hello/{name}', [HelloController::class, 'sayHello']);
11|
12| Route::get('/welcome', [GreetingsController::class, 'welcome']);
13| Route::get('/hello/{name}/{npm}', [GreetingsController::class,
14| 'greet']);
14| #-----
15| #Practicum #3
16| 
```

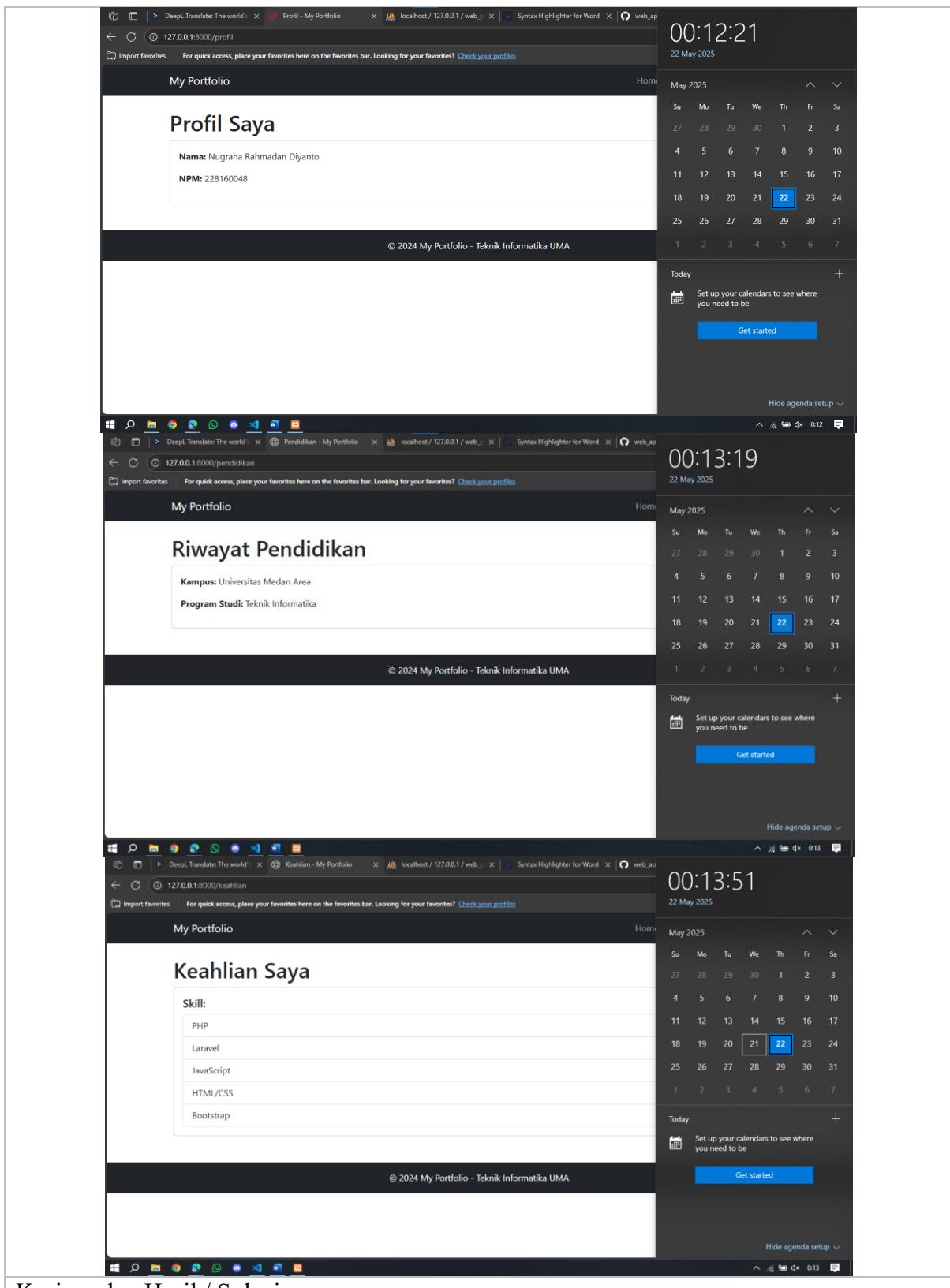
```

17| use App\Http\Controllers\PortfolioController;
18|
19| Route::get('/home', [PortfolioController::class, 'home']);
20| Route::get('/profil', [PortfolioController::class, 'profil']);
21| Route::get('/pendidikan', [PortfolioController::class,
22| 'pendidikan']);
23|
24| // Route untuk halaman utama
25| Route::redirect('/', '/home');
26| #-----
27|
28|
29|
30|
31| Route::get('/', function () {
32|     return view('welcome');
33| });
34|
35| Route::get('/', function () {
36|     return 'Selamat Datang Di Aplikasi Laravel';
37| });
38|
39| Route::get('/hello/{name}', function ($name) {
40|     return 'Hello, ' . $name . '!';
41| });
42|
43|
44| Route::get('/hello/{name}/{npm}', function ($name, $npm) {
45|     return 'Hello, ' . $name . ' ' . $npm . '!';
46| });

```

Screen Shoot / Output Hasil Kode Program:





Kesimpulan Hasil / Solusi:

Pembuatan tampilan web menggunakan Blade Templating Engine Laravel, yang memungkinkan pengembangan antarmuka yang modular melalui fitur template inheritance, komponen reusable, dan sintaks yang ringkas untuk menampilkan data dinamis, sehingga mempercepat proses development sekaligus menjaga konsistensi desain.

MODEL, MIGRASI DAN INPUT FORM

Task: Modul 4

Silahkan mengerjakan view untuk halaman website portfolio yang terdiri dari halaman:

- Home.blade.php
- Profil.blade.php
- Pendidikan.blade.php
- Keahlian.blade.php

dengan ketentuan :

1. profil menampilkan nilai variabel Nama dan NPM anda, nilai variabel berasal dari kontroller
2. pendidikan menampilkan variabel Kampus dan Prodi anda, nilai variabel berasal dari kontroller
3. keahlian menampilkan variabel skill anda, nilai variabel berasal dari kontroller
4. Buat masing-masing function / method untuk menampilkan tiap halaman dan buatkan route untuk meng akses tiap view tersebut.
5. Dokumentasikan hasil uji coba setiap URL dari tiap-tiap view tersebut ke dalam laporan praktikum.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git
Migrasi Tabel Provinsi

```
1 | <?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|     public function up()
11|     {
12|         Schema::create('provinsi', function (Blueprint $table) {
13|             $table->id();
14|             $table->string('nama');
15|             $table->timestamps();
16|         });
17|
18|     public function down()
19|     {
20|         Schema::dropIfExists('provinsi');
21|     }
22| };
```

Migrasi Tabel Kota

```
1 | <?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 |     public function up()
```

```

10|     {
11|         Schema::create('kota', function (Blueprint $table) {
12|             $table->id();
13|             $table->string('nama');
14|             $table->unsignedBigInteger('provinsi_id');
15|             $table->timestamps();
16|
17|             $table->foreign('provinsi_id')
18|                 ->references('id')
19|                     ->on('provinsi')
20|                         ->onDelete('cascade');
21|         });
22|     }
23|
24|     public function down()
25|     {
26|         Schema::dropIfExists('kota');
27|     }
28| };

```

Migrasi Tabel Kecamatan

```

1 | <?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|     public function up()
11|     {
12|         Schema::create('kecamatan', function (Blueprint $table) {
13|             $table->id();
14|             $table->string('nama');
15|             $table->unsignedBigInteger('kota_id');
16|             $table->timestamps();
17|
18|             $table->foreign('kota_id')
19|                 ->references('id')
20|                     ->on('kota')
21|                         ->onDelete('cascade');
22|         });
23|     }
24|
25|     public function down()
26|     {
27|         Schema::dropIfExists('kecamatan');
28|     }
29| };

```

Migrasi Tabel Kelurahan

```

1 | <?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|     public function up()
11|     {
12|         Schema::create('kelurahan', function (Blueprint $table) {
13|             $table->id();
14|             $table->string('nama');
15|             $table->unsignedBigInteger('kecamatan_id');
16|             $table->timestamps();
17|
18|             $table->foreign('kecamatan_id')
19|                 ->references('id')
20|                     ->on('kecamatan')
21|                         ->onDelete('cascade');
22|         });
23|     }
24|
25|     public function down()
26|     {
27|         Schema::dropIfExists('kelurahan');
28|     }
29| };

```

```

12|         $table->id();
13|         $table->string('nama');
14|         $table->unsignedBigInteger('kecamatan_id');
15|         $table->timestamps();
16|
17|         $table->foreign('kecamatan_id')
18|             ->references('id')
19|             ->on('kecamatan')
20|             ->onDelete('cascade');
21|     });
22|
23|
24|     public function down()
25|     {
26|         Schema::dropIfExists('kelurahan');
27|     }
28| };

```

Model Provinsi

```

1 | <?php
2 |
3 | namespace App\Models;
4 |
5 | use Illuminate\Database\Eloquent\Factories\HasFactory;
6 | use Illuminate\Database\Eloquent\Model;
7 |
8 | class Provinsi extends Model
9 | {
10|     use HasFactory;
11|
12|     protected $fillable = ['nama'];
13|
14|     public function kota()
15|     {
16|         return $this->hasMany(Kota::class);
17|     }
18| }

```

Model Kota

```

1 | <?php
2 |
3 | namespace App\Models;
4 |
5 | use Illuminate\Database\Eloquent\Factories\HasFactory;
6 | use Illuminate\Database\Eloquent\Model;
7 |
8 | class Kota extends Model
9 | {
10|     use HasFactory;
11|
12|     protected $fillable = ['nama', 'provinsi_id'];
13|
14|     public function provinsi()
15|     {
16|         return $this->belongsTo(Provinsi::class);
17|     }
18|
19|     public function kecamatan()
20|     {
21|         return $this->hasMany(Kecamatan::class);
22|     }
23| }

```

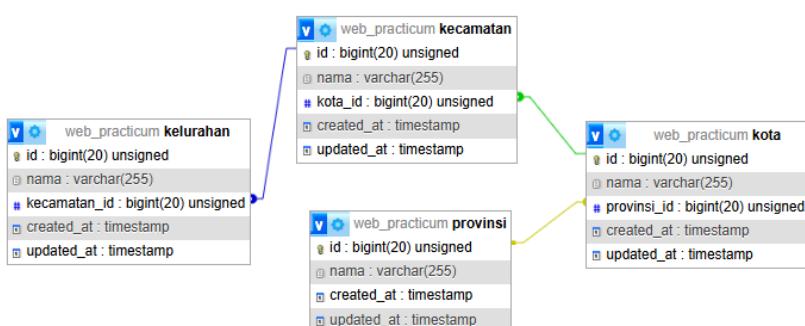
Model Kecamatan

```
1 | <?php
2 |
3 | namespace App\Models;
4 |
5 | use Illuminate\Database\Eloquent\Factories\HasFactory;
6 | use Illuminate\Database\Eloquent\Model;
7 |
8 | class Kecamatan extends Model
9 | {
10|     use HasFactory;
11|
12|     protected $fillable = ['nama', 'kota_id'];
13|
14|     public function kota()
15|     {
16|         return $this->belongsTo(Kota::class);
17|     }
18|
19|     public function kelurahan()
20|     {
21|         return $this->hasMany(Kelurahan::class);
22|     }
23| }
```

Model Kelurahan

```
1 | <?php
2 |
3 | namespace App\Models;
4 |
5 | use Illuminate\Database\Eloquent\Factories\HasFactory;
6 | use Illuminate\Database\Eloquent\Model;
7 |
8 | class Kelurahan extends Model
9 | {
10|     use HasFactory;
11|
12|     protected $fillable = ['nama', 'kecamatan_id'];
13|
14|     public function kecamatan()
15|     {
16|         return $this->belongsTo(Kecamatan::class);
17|     }
18| }
```

Screen Shoot / Output Hasil Kode Program:



23:42:24
21 May 2025

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<input type="checkbox"/> id	bigint(20)	unsigned	No	None		AUTO_INCREMENT		<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> Move
2	<input type="checkbox"/> nama	varchar(255)	utf8mb4_unicode_ci	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> Move
3	<input type="checkbox"/> created_at	timestamp		Yes	NULL				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> Move
4	<input type="checkbox"/> updated_at	timestamp		Yes	NULL				<input type="checkbox"/> Change <input type="checkbox"/> Drop <input type="checkbox"/> Move

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="checkbox"/> Edit	<input type="checkbox"/> Rename	<input type="checkbox"/> Drop	PRIMARY	BTREE	Yes	No	id	0	A No

Create an index on 1 columns

Partitions

No partitioning defined

Console

23:42:54
21 May 2025

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<input type="checkbox"/> id	bigint(20)	unsigned	No	None		AUTO_INCREMENT		<input type="checkbox"/> Change <input type="checkbox"/> Drop
2	<input type="checkbox"/> nama	varchar(255)	utf8mb4_unicode_ci	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop
3	<input type="checkbox"/> provinsi_id	bigint(20)	unsigned	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop
4	<input type="checkbox"/> created_at	timestamp		Yes	NULL				<input type="checkbox"/> Change <input type="checkbox"/> Drop
5	<input type="checkbox"/> updated_at	timestamp		Yes	NULL				<input type="checkbox"/> Change <input type="checkbox"/> Drop

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="checkbox"/> Edit	<input type="checkbox"/> Rename	<input type="checkbox"/> Drop	PRIMARY	BTREE	Yes	No	id	0	A No
<input type="checkbox"/> Edit	<input type="checkbox"/> Rename	<input type="checkbox"/> Drop	kota_provinsi_id_foreign	BTREE	No	No	provinsi_id	0	A No

Create an index on 1 columns

Partitions

Console

23:43:35
21 May 2025

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<input type="checkbox"/> id	bigint(20)	unsigned	No	None		AUTO_INCREMENT		<input type="checkbox"/> Change <input type="checkbox"/> Drop
2	<input type="checkbox"/> nama	varchar(255)	utf8mb4_unicode_ci	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop
3	<input type="checkbox"/> kota_id	bigint(20)	unsigned	No	None				<input type="checkbox"/> Change <input type="checkbox"/> Drop
4	<input type="checkbox"/> created_at	timestamp		Yes	NULL				<input type="checkbox"/> Change <input type="checkbox"/> Drop
5	<input type="checkbox"/> updated_at	timestamp		Yes	NULL				<input type="checkbox"/> Change <input type="checkbox"/> Drop

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<input type="checkbox"/> Edit	<input type="checkbox"/> Rename	<input type="checkbox"/> Drop	PRIMARY	BTREE	Yes	No	id	0	A No
<input type="checkbox"/> Edit	<input type="checkbox"/> Rename	<input type="checkbox"/> Drop	kecamatan_kota_id_foreign	BTREE	No	No	kota_id	0	A No

Create an index on 1 columns

Partitions

Console

Kesimpulan Hasil / Solusi:

Implementasi model dan migrasi Laravel untuk membangun struktur database relasional, mencakup pembuatan tabel (provinsi, kota, kecamatan, kelurahan) melalui migrasi, pendefinisian relasi one-to-many antar model, serta operasi CRUD dengan Eloquent ORM, yang memungkinkan pengelolaan data secara efisien dan terstruktur dalam aplikasi web.

Database dan Eloquent ORM

Task: Modul 5

- Buatlah tampilan data yang menampilkan data nilai berdasarkan id satu mahasiswa.Buatlah kode dibawah ini pada file NilaiController.php

```
1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 | use App\Models\Mahasiswa;
7 |
8 | public function showNilaiMahasiswa($mahasiswaId)
9 | {
10|     $mahasiswa = Mahasiswa::find($mahasiswaId);
11|
12|     if (!$mahasiswa) {
13|         echo "Data mahasiswa tidak ditemukan";
14|     } else {
15|         echo "Nama Mahasiswa: $mahasiswa->nama <br>";
16|
17|         foreach ($mahasiswa->nilais as $nilai) {
18|             $matakuliah = $nilai->matakuliah;
19|             echo "Matakuliah: $matakuliah->nama | Nilai: $nilai-
>nilai <br>";
20|         }
21|     }
22| }
```

Route :

```
1 | use App\Http\Controllers\NilaiController;
2 |
3 | Route::get('/nilai/{mahasiswaId}', [NilaiController::class, 'showNilaiMahasiswa'])-
>name('tampilnilai');
```

Akses hasilnya dengan : <http://localhost:8000/nilai/1>

- Buatlah file View Untuk menampilkan tugas 1 tersebut dengan kode html nya.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git
NilaiController.php

```
1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 | use App\Models\Mahasiswa;
7 | use App\Models\Nilai;
8 |
9 | class NilaiController extends Controller
10| {
11|     public function index()
12|     {
13|         $dataNilai = Nilai::with(['mahasiswa', 'matakuliah'])->get();
14|         return view('datanilai', compact('dataNilai'));
15|     }
16|
17|     public function showNilaiMahasiswa($mahasiswaId)
18|     {
```

```

19|         $mahasiswa = Mahasiswa::with('nilai.matakuliah')-
>find($mahasiswaId);
20|
21|         if (!$mahasiswa) {
22|             return view('nilai.not_found');
23|         }
24|
25|         return view('nilai.show', compact('mahasiswa'));
26|     }
27| }

```

Route Web.php

```

1 | <?php
2 |
3 | use App\Http\Controllers\GreetingsController;
4 | use App\Http\Controllers\HelloController;
5 | use Illuminate\Support\Facades\Route;
6 | use App\Http\Controllers\HomeController;
7 | Route::get('/home', [HomeController::class, 'index']);
8 |
9 | Route::get('/', [HelloController::class, 'index']);
10| Route::get('/hello/{name}', [HelloController::class, 'sayHello']);
11|
12| Route::get('/welcome', [GreetingsController::class, 'welcome']);
13| Route::get('/hello/{name}/{npm}', [GreetingsController::class,
'greet']);
14|
15| #-----
16| #Practicum #3
17| use App\Http\Controllers\PortfolioController;
18|
19| Route::get('/home', [PortfolioController::class, 'home']);
20| Route::get('/profil', [PortfolioController::class, 'profil']);
21| Route::get('/pendidikan', [PortfolioController::class,
'pendidikan']);
22| Route::get('/keahlian', [PortfolioController::class, 'keahlian']);
23|
24| // Route untuk halaman utama
25| Route::redirect('/', '/home');
26| #-----
27|
28| #Practicum #5.1
29| use App\Http\Controllers\NilaiController;
30|
31| Route::get('/nilai', [NilaiController::class, 'index'])-
>name('nilai.index');
32| #-----
33| #Practicum #5.2
34| use App\Http\Controllers\MahasiswaController;
35|
36| Route::resource('mahasiswa', MahasiswaController::class);
37| #-----
38| #Practicum #task
39| Route::get('/nilai/{mahasiswaId}', [NilaiController::class,
'showNilaiMahasiswa'])->name('tampilnilai');
40| #-----
41| Route::get('/', function () {
42|     return view('welcome');
43| });
44|

```

```

45| Route::get('/', function () {
46|     return 'Selamat Datang Di Aplikasi Laravel';
47| });
48|
49| Route::get('/hello/{name}', function ($name) {
50|     return 'Hello, ' . $name . '!';
51| });
52|
53| Route::get('/hello/{name}/{npm}', function ($name , $npm) {
54|     return 'Hello, ' . $name . ' ' . $npm . '!';
55| });

```

Show.blade.php

```

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 |     <title>Nilai Mahasiswa</title>
7 |     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css" rel="stylesheet">
8 | </head>
9 | <body>
10|     <div class="container mt-5">
11|         <h1 class="mb-4">Nilai Mahasiswa</h1>
12|
13|         <div class="card mb-4">
14|             <div class="card-header">
15|                 <h2>Informasi Mahasiswa</h2>
16|             </div>
17|             <div class="card-body">
18|                 <p><strong>Nama:</strong> {{ $mahasiswa->nama }}</p>
19|                 <p><strong>NIM:</strong> {{ $mahasiswa->nim }}</p>
20|                 <p><strong>Alamat:</strong> {{ $mahasiswa->alamat ??
'-' }}</p>
21|             </div>
22|         </div>
23|
24|         <div class="card">
25|             <div class="card-header">
26|                 <h2>Daftar Nilai</h2>
27|             </div>
28|             <div class="card-body">
29|                 @if($mahasiswa->nilai->isEmpty())
30|                     <p class="text-muted">Mahasiswa ini belum
memiliki nilai.</p>
31|                 @else
32|                     <table class="table table-striped">
33|                         <thead>
34|                             <tr>
35|                                 <th>No</th>
36|                                 <th>Mata Kuliah</th>
37|                                 <th>Kode</th>
38|                                 <th>SKS</th>
39|                                 <th>Nilai</th>
40|                             </tr>
41|                         </thead>
42|                         <tbody>

```

```

43|                                     @foreach($mahasiswa->nilai as $index =>
$nilai)
44|                                         <tr>
45|
46|
47|                                         <td>{{ $index + 1 }}</td>
48|                                         <td>{{ $nilai->matakuliah->nama
}}</td>
49|                                         <td>{{ $nilai->matakuliah->kode
}}</td>
50|                                         <td>{{ $nilai->matakuliah->sks
}}</td>
51|                                         <td>{{ $nilai->nilai }}</td>
52|                                     </tr>
53|                                 @endforeach
54|                             </tbody>
55|                         </table>
56|                     </div>
57|
58|                     <a href="{{ url('/nilai') }}" class="btn btn-primary mt-
3">Kembali ke Daftar Nilai</a>
59|                 </div>
60|
61|             <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
62|         </body>
63|     </html>

```

Not_found.blade.php

```

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 |         <title>Data Tidak Ditemukan</title>
7 |         <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min
.css" rel="stylesheet">
8 |     </head>
9 |     <body>
10|         <div class="container mt-5">
11|             <div class="alert alert-danger">
12|                 <h4 class="alert-heading">Data Mahasiswa Tidak
Ditemukan</h4>
13|                 <p>Mahasiswa dengan ID yang diminta tidak ditemukan dalam
database.</p>
14|                 <hr>
15|                 <a href="{{ url('/nilai') }}" class="btn btn-
primary">Kembali ke Daftar Nilai</a>
16|             </div>
17|         </div>
18|     </body>
19| </html>

```

Screen Shoot / Output Hasil Kode Program:

The image displays three screenshots of a web application interface, likely built with Laravel, showing student information and grade lists.

Screenshot 1: This screenshot shows the student information page. It includes a header "Nilai Mahasiswa" and a section titled "Informasi Mahasiswa" containing the student's name, NIM, and address. Below this is a table titled "Daftar Nilai" showing one record: "Aplikasi Berbasis Web" with a code of "01" and 3 SKS, and a grade of "100". A blue button at the bottom left says "Kembali ke Daftar Nilai".

No	Mata Kuliah	Kode	SKS	Nilai
1	Aplikasi Berbasis Web	01	3	100

Screenshot 2: This screenshot shows the same student information page as the first one, but for a different student named Tyas Adrian. The "Informasi Mahasiswa" section shows her name, NIM, and address. The "Daftar Nilai" table shows the same record as the first screenshot.

No	Mata Kuliah	Kode	SKS
1	Aplikasi Berbasis Web	01	3

Screenshot 3: This screenshot shows the "Data Nilai Mahasiswa" page, displaying a table of student grades. The table has columns: No, NPM, Nama Mahasiswa, Matakuliah, and Nilai. It contains two records: one for Nugraha Rahmadan Diyanto and one for Tyas Adrian, both with an NPM of 228160049 and a grade of 100.

No	NPM	Nama Mahasiswa	Matakuliah	Nilai
1	228160049	Nugraha Rahmadan Diyanto	Aplikasi Berbasis Web	100
2	228160037	Tyas Adrian	Aplikasi Berbasis Web	100

Kesimpulan Hasil / Solusi:

Database dan Eloquent ORM dalam Laravel menyederhanakan pengelolaan data dengan mengubah tabel database menjadi objek PHP, memungkinkan operasi CRUD (Create, Read, Update, Delete) yang intuitif tanpa menulis kueri SQL manual. Eloquent ORM menyediakan metode seperti all(), find(), where(), dan relasi antar model (sepertihasMany atau belongsTo), sehingga mempercepat pengembangan aplikasi web dengan kode yang lebih rapi, mudah dibaca, dan terstruktur.

Autentikasi

Task: Modul 6

1: Kerjakan Praktikum 1 dan Praktikum 2 pada modul 6 ini, kemudian dokumentasikan hasilnya kedalam laporan praktikum.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git

Praktikum 1

install ui dan auth

```
0 | composer require laravel/ui  
1 | php artisan ui bootstrap --auth
```

Memasukkan ke database

```
0 | php artisan migrate
```

Praktikum 2 login.blade.php

```
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 |     <title>Login</title>  
7 |     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">  
8 |   </head>  
9 |   <body>  
10|     <div class="container mt-5">  
11|       <div class="row justify-content-center">  
12|         <div class="col-md-6">  
13|           <div class="card">  
14|             <div class="card-header">Login</div>  
15|             <div class="card-body">  
16|               @if(session('success'))  
17|                 <div class="alert alert-info">{{ session('success') }}</div>  
18|               @endif  
19|               @if(session('error'))  
20|                 <div class="alert alert-danger">{{ session('error') }}</div>  
21|               @endif  
22|               @endif  
23|               <form action="{{ route('login') }}" method="POST">  
24|                 @csrf  
25|                 <div class="mb-3">  
26|                   <label for="email" class="form-label">Email</label>  
27|                   <input type="email" class="form-control" id="email" name="email" required>  
28|                   </div>  
29|                   <div class="mb-3">  
30|                     <label for="password" class="form-label">Password</label>  
31|                     <input type="password" class="form-control" id="password" name="password" required>  
32|                   </div>  
33|                   <button type="submit" class="btn btn-primary">Login</button>  
34|                 </form>  
35|               </div>  
36|             </div>  
37|           </div>  
38|         </div>  
39|       </div>  
40|     </body>  
41| </html>
```

Model User

```
1 | <?php
2 |
3 | namespace App\Models;
4 |
5 | use Illuminate\Contracts\Auth\MustVerifyEmail;
6 | use Illuminate\Database\Eloquent\Factories\HasFactory;
7 | use Illuminate\Foundation\Auth\User as Authenticatable;
8 | use Illuminate\Notifications\Notifiable;
9 |
10| class User extends Authenticatable
11| {
12|     use HasFactory, Notifiable;
13|
14|     /**
15|      * The attributes that are mass assignable.
16|      *
17|      * @var array
18|      */
19|     protected $fillable = [
20|         'name',
21|         'email',
22|         'password',
23|     ];
24|
25|     /**
26|      * The attributes that should be hidden for arrays.
27|      *
28|      * @var array
29|      */
30|     protected $hidden = [
31|         'password',
32|         'remember_token',
33|     ];
34|
35|     /**
36|      * The attributes that should be cast to native types.
37|      *
38|      * @var array
39|      */
40|     protected $casts = [
41|         'email_verified_at' => 'datetime',
42|     ];
43|
44|     /**
45|      * Get the email address of the user.
46|      *
47|      * @return string
48|      */
49|     public function email()
50|     {
51|         return $this->email;
52|     }
53| }
```

Dashboard

```
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 |     <title>Dashboard</title>
7 |     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
8 | </head>
9 | <body>
10|     <div class="container mt-5">
11|         <div class="row justify-content-center">
12|             <div class="col-md-8">
13|                 <div class="card">
```

```

14|             <div class="card-header">Dashboard</div>
15|             <div class="card-body">
16|                 @if(session('success'))
17|                     <div class="alert alert-info">{{ session('success') }}</div>
18|                 @endif
19|                 <p>Welcome, {{ auth() ->user() ->email }}</p>
20|                 <h1>Halaman Dashboard</h1>
21|             </div>
22|         </div>
23|     </div>
24| </div>
25| </body>
26| </html>
27|

```

Route

```

1 | use App\Http\Controllers\AuthController;
2 | use Illuminate\Support\Facades\Auth;
3 | use Illuminate\Support\Facades\Route;
4 |
5 | Route::get('/login', function () {
6 |     return view('auth.login');
7 | })->name('login');
8 |
9 | Route::post('/login', [AuthController::class, 'cekLogin'])->name('cek-login');
10|
11| Route::middleware('auth')->group(function () {
12|     Route::get('/dashboard', function () {
13|         return view('dashboard');
14|     })->name('dashboard');
15|
16|     Route::get('/logout', function () {
17|         Auth::logout();
18|         return redirect()->route('login')->with('success', 'Anda logged out.');
19|     })->name('logout');
20| });

```

AuthController

```

1 | <?php
2 |
3 | namespace App\Http\Controllers;
4 |
5 | use Illuminate\Http\Request;
6 | use Illuminate\Support\Facades\Auth;
7 |
8 | class AuthController extends Controller
9 | {
10|     public function cekLogin(Request $request)
11|     {
12|         $credentials = $request->only('email', 'password');
13|
14|         if (Auth::attempt($credentials)) {
15|             return redirect()
16|                 ->route('dashboard')
17|                 ->with('success', 'Login successful.');
18|         }
19|
20|         return redirect()
21|             ->route('login')
22|             ->with('error', 'Login failed. Periksa Username/Password.');
23|     }
24| }

```

Screen Shoot / Output Hasil Kode Program:

Praktikum 1 Tampilan awal login

Login

Email Address tyasadrian33@gmail.com

Password
 Remember Me

[Login](#) [Forgot Your Password?](#)

Tampilan awal register

Register

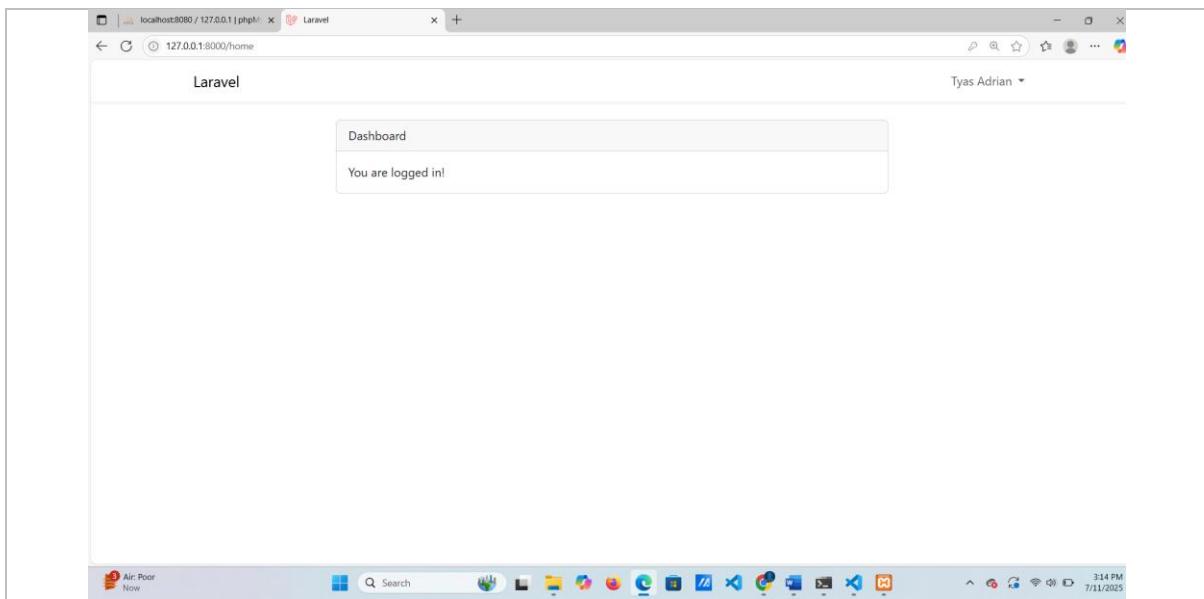
Name Tyas Adrian

Email Address tyasadrian33@gmail.com

Password
Confirm Password

[Register](#)

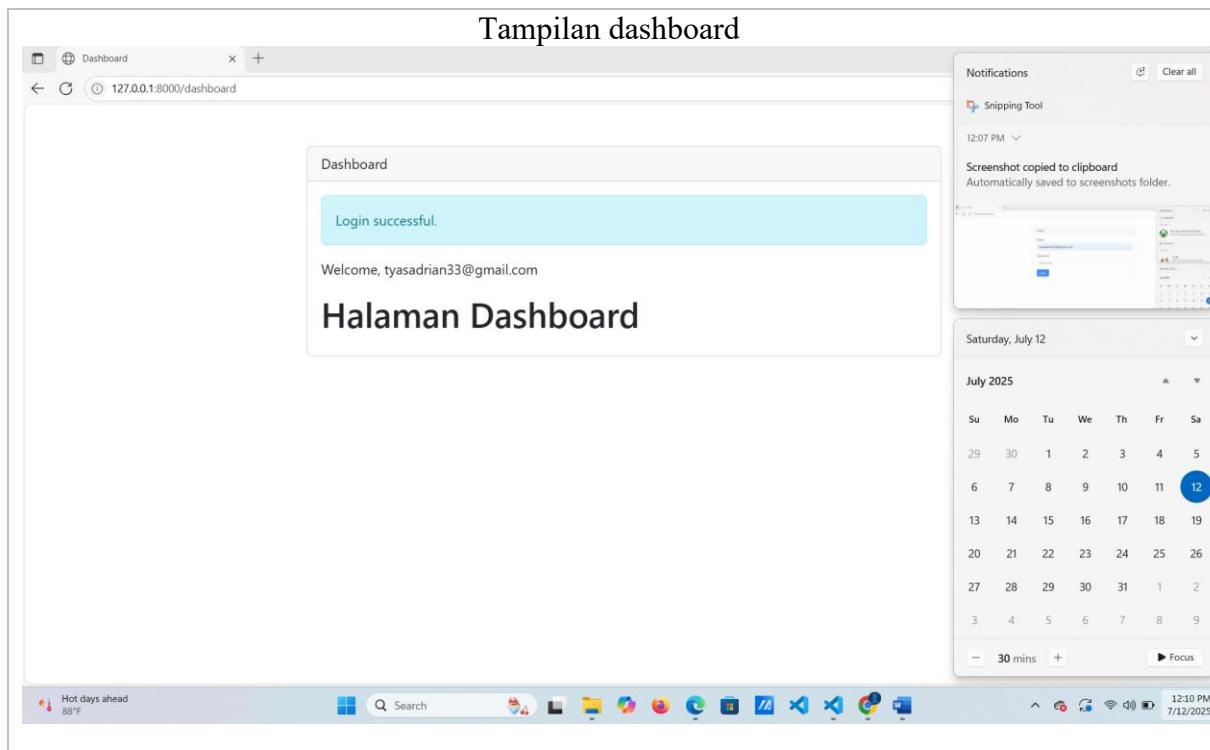
Tampilan home



Praktikum 2

Tampilan awal login

A screenshot of a web browser window titled "Login". The address bar shows "127.0.0.1:8000/login". The main content area displays a "Login" form with fields for "Email" (containing "tyasadrian33@gmail.com") and "Password" (containing "*****"). A "Login" button is visible. To the right of the login form is a sidebar. The sidebar includes a "Notifications" section with a "Suggested" item from Xbox about playing day one with Game Pass. It also shows a WhatsApp message from "ais" with the text "kok kamu kayak ga senang gtu". Below the notifications is a calendar for July 2025, with the 12th highlighted. The bottom of the sidebar shows a "30 mins" timer and a "Focus" button. The browser interface includes a taskbar at the bottom with various icons and a system tray showing the date and time.



Kesimpulan Hasil / Solusi:

Kesimpulan: Program-program di atas membangun sistem login sederhana menggunakan Laravel yang terdiri dari: (1) View Login (Blade) dengan form email/password dan notifikasi error/success, (2) Model User untuk autentikasi dengan proteksi password, (3) View Dashboard yang menampilkan email user setelah login, (4) Routes yang mengatur akses halaman login/dashboard/logout dengan middleware auth, serta (5) AuthController yang memproses login menggunakan Auth::attempt(). Sistem ini menggunakan Bootstrap untuk tampilan dan menerapkan komentar Blade ({{-- --}}) yang aman (tidak muncul di HTML).

Middleware dan Keamanan

Task: Modul 7

1. Buat middleware custom yang memeriksa apakah pengguna memiliki atribut tertentu (misalnya, usia minimal). Maka anda harus menambahkan kolom usia pada tabel User.
2. Implementasikan middleware tersebut pada rute tertentu untuk melindungi 1 halaman route page. Buat 1 halaman views yang hanya bisa diakses jika usia user >= 18 tahun, untuk isi konten page menampilkan Auth::user()->name dan Auth::user()->role
3. Buat hasil praktikum kedalam Laporan Praktikum sesuai format.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git
program tugas
adult.blade.php

```
1 | <!DOCTYPE html>
2 | <html>
3 | <head>
4 |   <title>Adult Content</title>
5 | </head>
6 | <body>
7 |   <h2>Welcome to Adult Content Page</h2>
8 |   <p>Name: {{ Auth::user()->name }}</p>
9 |   <p>Role: {{ Auth::user()->role }}</p>
10|   <p>Age: {{ Auth::user()->age }}</p>
11|   <a href="{{ route('home') }}">Back to Home</a>
12| </body>
13| </html>
```

Route web.php

```
1 |
2 | Route::middleware(['auth', 'check.age'])->group(function () {
3 |   Route::get('/adult-content', function () {
4 |     return view('adult');
5 |   })->name('adult.content');
6 |});
```

migrasi

```
1 | <?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 |   public function up(): void
10|   {
11|     Schema::table('users', function (Blueprint $table) {
12|       $table->integer('age')->nullable()->after('role');
13|     });
14|   }
15|
16|   public function down(): void
17|   {
18|     Schema::table('users', function (Blueprint $table) {
19|       $table->dropColumn('age');
20|     });
21|   }
22| };
```

Middleware

```
1 | <?php
2 |
3 | namespace App\Http\Middleware;
4 |
```

```

5 | use Closure;
6 | use Illuminate\Http\Request;
7 | use Symfony\Component\HttpFoundation\Response;
8 |
9 | class CheckAge
10| {
11|     public function handle(Request $request, Closure $next): Response
12|     {
13|         if (auth()->check() && auth()->user()->age >= 18) {
14|             return $next($request);
15|         }
16|
17|         return redirect('home')->with('error', 'You must be at least 18 years old to
access this page.');
18|     }
19| }

```

Bootstrap app.php

```

1 | <?php
2 |
3 | use Illuminate\Foundation\Application;
4 | use Illuminate\Foundation\Configuration\Exceptions;
5 | use Illuminate\Foundation\Configuration\Middleware;
6 |
7 | return Application::configure(basePath: dirname(__DIR__))
8 |     ->withRouting(
9 |         web: __DIR__.'/../routes/web.php',
10|         commands: __DIR__.'/../routes/console.php',
11|         health: '/up',
12|     )
13|     ->withMiddleware(function (Middleware $middleware) {
14|         $middleware->alias([
15|             'auth' => \App\Http\Middleware\AuthMiddleware::class,
16|             'role' => \App\Http\Middleware\RoleMiddleware::class,
17|             'check.age' => \App\Http\Middleware\CheckAge::class,
18|         ]);
19|     })
20|     ->withExceptions(function (Exceptions $exceptions) {
21|         //
22|     })->create();
23|

```

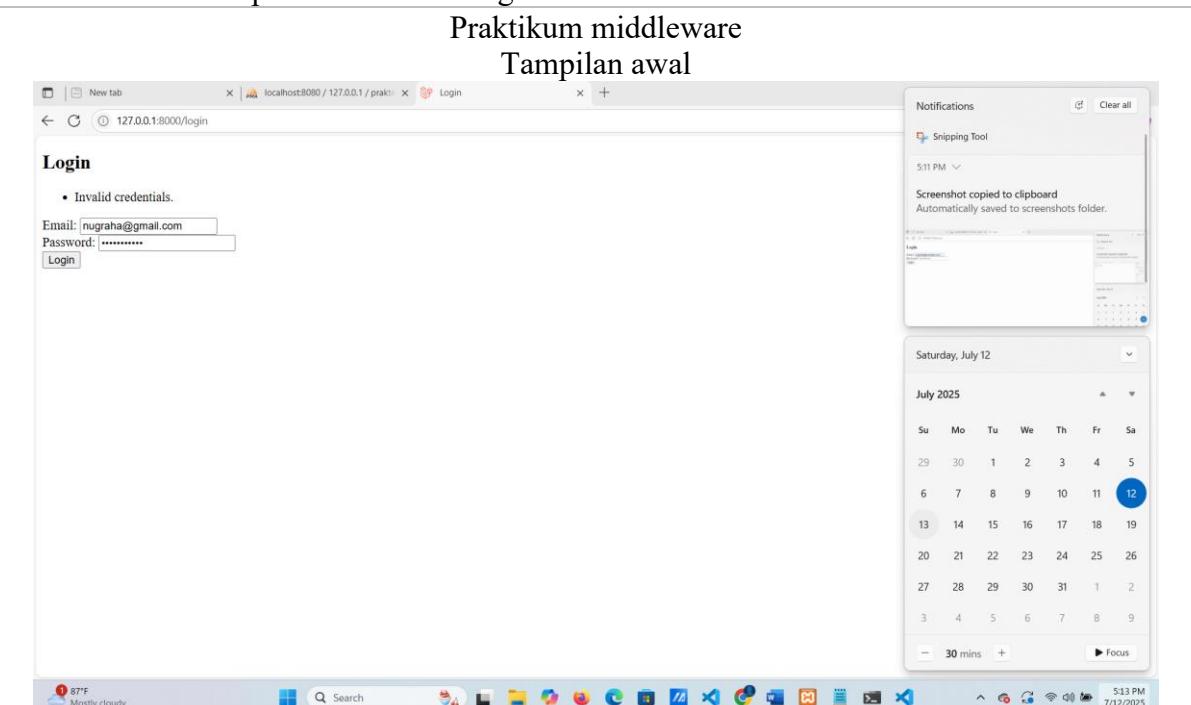
User model

```

1 |     protected $fillable = [
2 |         'name', 'email', 'password', 'role', 'age'
3 |     ];

```

Screen Shoot / Output Hasil Kode Program:



Tampilan admin

Welcome, Nugraha

[Logout](#)

Tampilan detail

Welcome to Adult Content Page

Name: Nugraha

Role: admin

Age: 20

[Back to Home](#)

Kesimpulan Hasil / Solusi:

Sistem yang dibangun merupakan aplikasi web berbasis Laravel dengan fitur autentikasi user yang membedakan akses berdasarkan role (admin dan owner), dilengkapi dengan middleware custom untuk verifikasi usia (minimal 18 tahun) guna mengakses halaman khusus dewasa (adult content), menggunakan Bootstrap 5 untuk tampilan yang responsif dan modern, serta memiliki mekanisme keamanan seperti CSRF protection dan session management, dimana seluruh halaman (login, admin dashboard, owner dashboard, dan adult content) telah didesain secara konsisten dengan navigasi yang intuitif dan penampilan data user yang jelas.

Manajemen File Uploads

Task: Modul 8

1. Validasi Tambahan: Tambahkan validasi untuk memastikan hanya file gambar (jpg, jpeg, png) yang dapat diupload dan ukuran maksimal file adalah 2MB.
2. Tampilkan Daftar File: Buat halaman yang menampilkan daftar semua file yang diupload dan tautan untuk mendownload file tersebut.
3. Hapus File: Tambahkan fitur untuk menghapus file yang diupload dari storage.
4. Dokumentasikan semua Kegiatan Modul 8 dan Tugas ke dalam laporan Praktikum.

No	Foto	Aksi
1		<button>del</button>
2		<button>del</button>

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git

route web.php

```
0 | use App\Http\Controllers\FileUploadController;
1 |
2 | Route::get('/upload', [FileUploadController::class, 'showUploadForm'])-
>name('upload.form');
3 | Route::post('/upload', [FileUploadController::class, 'storeFile'])-
>name('upload.store');
4 | Route::get('/files', [FileUploadController::class, 'listFiles'])->name('files.list');
5 | Route::delete('/files/{filename}', [FileUploadController::class, 'deleteFile'])-
>name('files.delete');
6 | Route::get('/files/download/{filename}', [FileUploadController::class, 'downloadFile'])->name('files.download');
```

Controller

```
0 | <?php
1 |
2 | namespace App\Http\Controllers;
3 |
4 | use Illuminate\Http\Request;
5 | use Illuminate\Support\Facades\Storage;
6 |
7 | class FileUploadController extends Controller
8 | {
9 |     public function showUploadForm()
10|    {
11|        return view('upload');
12|    }
13|
14|    public function storeFile(Request $request)
15|    {
16|        // Validasi dengan ketentuan tugas: hanya gambar dan maksimal 2MB
17|        $request->validate([
18|            'file' => 'required|file|image|mimes:jpg,jpeg,png|max:2048',
19|        ]);
20|
21|        $file = $request->file('file');
```

```

22|         // Simpan file dengan nama asli
23|         $fileName = time() . '_' . $file->getClientOriginalName();
24|         $path = $file->storeAs('public/uploads', $fileName);
25|
26|         return back()->with('success', 'File uploaded successfully!')
27|                         ->with('file', $path)
28|                         ->with('fileName', $fileName);
29|
30|     }
31|
32|     public function listFiles()
33|     {
34|         $files = Storage::files('public/uploads');
35|         $fileDetails = [];
36|
37|         foreach ($files as $file) {
38|             $fileDetails[] = [
39|                 'name' => basename($file),
40|                 'path' => $file,
41|                 'size' => Storage::size($file),
42|                 'url' => Storage::url($file),
43|                 'lastModified' => Storage::lastModified($file)
44|             ];
45|         }
46|
47|         return view('file_list', compact('fileDetails'));
48|     }
49|
50|     public function deleteFile($filename)
51|     {
52|         $filePath = 'public/uploads/' . $filename;
53|
54|         if (Storage::exists($filePath)) {
55|             Storage::delete($filePath);
56|             return back()->with('success', 'File deleted successfully!');
57|         }
58|
59|         return back()->with('error', 'File not found!');
60|     }
61|
62|     public function downloadFile($filename)
63|     {
64|         $filePath = 'public/uploads/' . $filename;
65|
66|         if (Storage::exists($filePath)) {
67|             return Storage::download($filePath);
68|         }
69|
70|         return back()->with('error', 'File not found!');
71|     }
72| }

```

View upload

```

0 | <!DOCTYPE html>
1 | <html>
2 | <head>
3 |     <title>File Upload</title>
4 |     <link rel="stylesheet"
5 | href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
6 | </head>
7 | <body>
8 |     <div class="container mt-5">
9 |         <div class="row justify-content-center">
10|             <div class="col-md-8">
11|                 <div class="card">
12|                     <div class="card-header">
13|                         <h2>File Upload</h2>
14|                     </div>
15|                     <div class="card-body">
16|                         @if (session('success'))
17|                             <div class="alert alert-success">
18|                                 {{ session('success') }}
19|                                 @if(session('file'))

```

```

20|                                     <a href="{{ Storage::url(session('file')) }}" target="_blank" class="btn btn-sm btn-info mt-2">View File</a>
21|                                     @endif
22|                               </div>
23|                           @endif
24|
25|                           @if (session('error'))
26|                               <div class="alert alert-danger">
27|                                   {{ session('error') }}
28|                               </div>
29|                           @endif
30|
31|                               <form action="{{ route('upload.store') }}" method="POST"
32| enctype="multipart/form-data">
33|                                 @csrf
34|                                 <div class="form-group">
35|                                   <label for="file">Choose Image File (JPG, JPEG, PNG -
Max 2MB)</label>
36|                                   @error('file') is-invalid @enderror accept="image/*">
37|                                       @error('file')
38|                                           <div class="invalid-feedback">{{ $message }}</div>
39|                                       @enderror
40|                                   <div>
41|                                     <button type="submit" class="btn btn-primary">Upload</button>
42|                                     <a href="{{ route('files.list') }}" class="btn btn-secondary">View All Files</a>
43|                                   </div>
44|                               </div>
45|                           </div>
46|                       </div>
47|                   </div>
48|               </body>
49|           </html>

```

Files list view

```

0 | <!DOCTYPE html>
1 | <html>
2 |   <head>
3 |     <title>File List</title>
4 |     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
5 |   </head>
6 |   <body>
7 |     <div class="container mt-5">
8 |       <div class="row justify-content-center">
9 |         <div class="col-md-10">
10|           <div class="card">
11|             <div class="card-header">
12|               <h2>File List</h2>
13|               <a href="{{ route('upload.form') }}" class="btn btn-primary btn-sm float-right">Upload New File</a>
14|             </div>
15|             <div class="card-body">
16|               @if (session('success'))
17|                 <div class="alert alert-success">
18|                   {{ session('success') }}
19|                 </div>
20|               @endif
21|
22|               @if (session('error'))
23|                 <div class="alert alert-danger">
24|                   {{ session('error') }}
25|                 </div>
26|               @endif
27|
28|               @if(count($fileDetails) > 0)
29|                 <div class="table-responsive">
30|                   <table class="table table-striped">
31|                     <thead>
32|                       <tr>
33|                         <th>No</th>
34|                         <th>Preview</th>
35|                         <th>File Name</th>
36|                         <th>Size</th>

```

```

37|                                     <th>Last Modified</th>
38|                                     <th>Actions</th>
39|
40|                                     </thead>
41|                                     <tbody>
42|                                         @foreach ($fileDetails as $index => $file)
43|                                         <tr>
44|                                             <td>{{ $index + 1 }}</td>
45|                                             <td>
46|                                                 
48|                                             </td>
49|                                             {{ round($file['size'] / 1024, 2)}}
50|                                             KB</td>
51|                                         <td>{{ date('Y-m-d H:i:s',
52|                                         $file['lastModified']) }}</td>
53|                                         <td>
54|                                             <a href="{{ $file['url'] }}" target="_blank" class="btn btn-info btn-sm">View</a>
55|                                             <a href="{{ route('files.download', $file['name']) }}" class="btn btn-success btn-sm">Download</a>
56|                                             <form action="{{ route('files.delete', $file['name']) }}" method="POST" class="d-inline" onsubmit="return confirm('Are you sure you want to delete this file?')">
57|                                                 @csrf
58|                                                 @method('DELETE')
59|                                                 <button type="submit" class="btn btn-danger btn-sm">Delete</button>
60|                                             </form>
61|                                         </td>
62|                                         @endforeach
63|                                     </tbody>
64|                                 </div>
65|                                 @else
66|                                     <p>No files uploaded yet.</p>
67|                                 @endif
68|                             </div>
69|                         </div>
70|                     </div>
71|                 </div>
72|             </div>
73|         </body>
74|     </html>

```

Config file system

```

0|     'disks' => [
1|
2|         'local' => [
3|             'driver' => 'local',
4|             'root' => storage_path('app'),
5|         ],
6|
7|         'public' => [
8|             'driver' => 'local',
9|             'root' => storage_path('app/public'),
10|            'url' => env('APP_URL').'/storage',
11|            'visibility' => 'public',
12|        ],

```

Screen Shoot / Output Hasil Kode Program:

Tampilan awal dan memasukkan file gambar

The screenshot shows a Windows desktop environment with two separate browser windows open, both displaying a "File Upload" interface from a local host application.

Top Browser Window:

- Title: File Upload
- Content: "Choose Image File (JPG, JPEG, PNG - Max 2MB)" and a file input field containing "fauzi.jpeg".
- Buttons: "Upload" and "View All Files".
- Right Panel: A "Notifications" section showing "No new notifications" and a calendar for July 2025.
- Taskbar: Shows various pinned icons like File Explorer, Edge, and File Explorer, along with system status icons.

Bottom Browser Window:

- Title: Tampilan sukses upload gambar
- Content: "File uploaded successfully!" and a green "View File" button.
- Content: "Choose Image File (JPG, JPEG, PNG - Max 2MB)" and a file input field showing "No file chosen".
- Buttons: "Upload" and "View All Files".
- Right Panel: A "Notifications" section showing "Screenshot copied to clipboard" and a calendar for July 2025.
- Taskbar: Shows various pinned icons and system status icons.

Tampilan semua files yang sudah di upload

No	Preview	File Name	Size	Last Modified	Actions
1		1752374156_topi.jpg	107.93 KB	2025-07-13 02:35:56	View Download Delete
2		1752374218_ridho.jpg	102.64 KB	2025-07-13 02:36:58	View Download Delete
3		1752374347_fauzi.jpeg	6.78 KB	2025-07-13 02:39:07	View Download Delete

Kesimpulan Hasil / Solusi:

Berdasarkan analisis seluruh kode program yang diberikan (FileUploadController.php, upload.blade.php, file_list.blade.php, dan web.php), masalah utama terletak pada penanganan path file yang tidak konsisten antara sistem penyimpanan Laravel dan symlink storage, khususnya pada struktur direktori yang mengandung backslash (\) dan slash (/) secara campuran, serta ketidaksesuaian antara URL publik (/storage/uploads/) dengan lokasi fisik file (storage/app/public/uploads/), yang diperparah oleh kurangnya validasi nama file dan konfigurasi symlink yang tidak optimal, sehingga menyebabkan gambar gagal ditampilkan meskipun file tersimpan di server

Scan QR & API Development

Task: Modul 9

1. Kerjalan Task Praktikum 1, 2 & 3 hasilnya dokumentasikan pada laporan Praktikum.
2. **Membangun API CRUD untuk Entitas Lain:**
 1. Buat entitas lain seperti kategori atau pengguna dan bangun API CRUD untuk entitas tersebut.
 2. Uji API menggunakan Postman.
3. **Membuat Dokumentasi API:**
 1. Buat dokumentasi API yang menjelaskan setiap endpoint, parameter, metode HTTP, dan contoh respons.
 2. Dokumentasi dapat berupa file teks, PDF, atau menggunakan alat seperti Swagger.
4. **Mengintegrasikan SCAN QR:**
 1. Implementasikan fitur untuk menghasilkan dan memindai kode QR yang terkait dengan produk.
 2. Tambahkan endpoint API yang menghasilkan kode QR untuk setiap produk dan endpoint yang menerima hasil pemindaian kode QR untuk mendapatkan detail produk.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git
route web.php

```
0 | //modul 9
1 | use App\Http\Controllers\ScanController;
2 |
3 | Route::get('/scankode', [ScanController::class, 'scanKode']);
4 | Route::post('/scan', [ScanController::class, 'processScan']);
5 | Route::get('/scan-data-produk', [ScanController::class, 'scanDataProduk'])->name('scan.data.produk');
6 | Route::post('/scan-produk', [ScanController::class, 'processScanProduk']);
7 |
8 | // New route for getting last scanned product
9 | Route::get('/api/last-scanned-product', [ScanController::class, 'getLastScannedProduct']);
10|
```

Controller scan

```
0 | <?php
1 |
2 | namespace App\Http\Controllers;
3 |
4 | use Illuminate\Http\Request;
5 | use App\Models\Product;
6 |
7 | class ScanController extends Controller
8 | {
9 |     public function scanKode()
10|     {
11|         return view('scankode');
12|     }
13|
14|     public function processScan(Request $request)
15|     {
16|         $request->validate([
17|             'code' => 'required|string',
```

```

18|         ]);
19|
20|         $scanData = $request->input('code');
21|         $product = Product::where('sku', $scanData)->first();
22|
23|         if ($product) {
24|             // Store scan result in session for transfer to cart
25|             session(['last_scanned_product' => [
26|                 'id' => $product->id,
27|                 'name' => $product->name,
28|                 'sku' => $product->sku,
29|                 'price' => $product->price,
30|                 'description' => $product->description,
31|                 'image' => $product->image_url
32|             ]]);
33|
34|             return response()->json([
35|                 'success' => true,
36|                 'code' => $scanData,
37|                 'product' => [
38|                     'id' => $product->id,
39|                     'name' => $product->name,
40|                     'sku' => $product->sku,
41|                     'price' => $product->price,
42|                     'description' => $product->description,
43|                     'image' => $product->image_url
44|                 ],
45|                 'redirect_url' => route('scan.data.produk')
46|             ]);
47|         } else {
48|             return response()->json([
49|                 'success' => false,
50|                 'code' => $scanData,
51|                 'message' => 'Produk tidak ditemukan.'
52|             ], 404);
53|         }
54|     }
55|
56|     public function scanDataProduk()
57|     {
58|         return view('scandataproduk');
59|     }
60|
61|     public function processScanProduk(Request $request)
62|     {
63|         $request->validate([
64|             'code' => 'required|string',
65|         ]);
66|
67|         $code = $request->input('code');
68|         $product = Product::where('sku', $code)->orWhere('id', $code)->first();
69|
70|         if ($product) {
71|             return response()->json([
72|                 'success' => true,
73|                 'code' => $code,
74|                 'product' => [
75|                     'id' => $product->id,
76|                     'name' => $product->name,
77|                     'sku' => $product->sku,
78|                     'price' => $product->price,
79|                     'description' => $product->description,
80|                     'image' => $product->image_url,
81|                 ]
82|             ]);
83|         }
84|
85|         return response()->json([
86|             'success' => false,
87|             'message' => 'Produk tidak ditemukan.',
88|             'code' => $code
89|         ], 404);
90|     }
91|
92|     // Get last scanned product from session
93|     public function getLastScannedProduct()

```

```

94|     {
95|         $product = session('last_scanned_product');
96|
97|         if ($product) {
98|             // Clear the session after retrieving
99|             session()->forget('last_scanned_product');
100|
101|             return response()->json([
102|                 'success' => true,
103|                 'product' => $product
104|             ]);
105|
106|
107|             return response()->json([
108|                 'success' => false,
109|                 'message' => 'No scanned product found'
110|             ]);
111|
112|
113|             // New method to get camera permissions status
114|             public function checkCameraPermission()
115|             {
116|                 return response()->json([
117|                     'hasPermission' => $this->checkBrowserCameraSupport()
118|                 ]);
119|
120|
121|             private function checkBrowserCameraSupport()
122|             {
123|                 return !! (navigator.mediaDevices && navigator.mediaDevices.getUserMedia);
124|             }
125|     }

```

View scancode

```

0 | <!DOCTYPE html>
1 | <html lang="en">
2 | <head>
3 |     <meta charset="UTF-8">
4 |     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 |     <title>Barcode Scan Produk</title>
6 |     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet">
7 | </head>
8 | <body class="container py-4">
9 |
10|     <h1 class="mb-4">Scan Barcode / QR Code</h1>
11|
12|     <div class="row">
13|         <div class="col-md-6">
14|             <div id="reader" style="width: 100%;"></div>
15|
16|         <div class="col-md-6">
17|             <p>Product Code: <span id="code" class="fw-bold"></span></p>
18|             <div id="result" class="mt-3"></div>
19|
20|             <!-- Add to Cart button -->
21|             <div id="addToCartSection" style="display: none;" class="mt-3">
22|                 <button id="addToCartBtn" class="btn btn-success me-2">
23|                     Tambah ke Shopping Cart
24|                 </button>
25|                 <button id="goToCartBtn" class="btn btn-primary">
26|                     Lihat Shopping Cart
27|                 </button>
28|             </div>
29|         </div>
30|     </div>
31|
32|     <script src="https://unpkg.com/html5-qrcode"></script>
33|     <script>
34|         let currentProduct = null;
35|
36|         function onScanSuccess(decodedText, decodedResult) {
37|             fetch('/scan', {
38|                 method: 'POST',

```

```

39|         headers: {
40|             'Content-Type': 'application/json',
41|             'Accept': 'application/json',
42|             'X-CSRF-TOKEN': `{{ csrf_token() }}`
43|         },
44|         body: JSON.stringify({
45|             code: decodedText
46|         })
47|     })
48|     .then(response => response.json())
49|     .then(data => {
50|         const resultDiv = document.getElementById('result');
51|         const resultCode = document.getElementById('code');
52|         const addToCartSection = document.getElementById('addToCartSection');
53|
54|         resultCode.textContent = data.code;
55|
56|         if (data.success) {
57|             const product = data.product;
58|             currentProduct = product;
59|
60|             resultDiv.innerHTML = `
61|                 <div class="alert alert-success">
62|                     <h5>Produk Ditemukan!</h5>
63|                     <p><strong>Nama:</strong> ${product.name}</p>
64|                     <p><strong>SKU:</strong> ${product.sku}</p>
65|                     <p><strong>Harga:</strong> Rp
${parseInt(product.price).toLocaleString()}</p>
66|                     <p><strong>Deskripsi:</strong> ${product.description}</p>
67|                     ${product.image} ? `` : ''
68|                 </div>
69|             `;
70|
71|             addToCartSection.style.display = 'block';
72|         } else {
73|             resultDiv.innerHTML = `<div class="alert alert-
danger">${data.message}</div>`;
74|             addToCartSection.style.display = 'none';
75|         }
76|     })
77|     .catch(error => {
78|         document.getElementById('result').innerHTML = `<div class="alert alert-
danger">Terjadi kesalahan.</div>`;
79|         console.error('Error:', error);
80|     });
81| }
82|
83| // Add to cart functionality
84| document.getElementById('addToCartBtn').addEventListener('click', function() {
85|     if (!currentProduct) return;
86|
87|     // Get existing cart from localStorage
88|     let cart = JSON.parse(localStorage.getItem('scanCart') || '[]');
89|
90|     // Check if product already exists in cart
91|     const existingIndex = cart.findIndex(item => item.sku ===
currentProduct.sku);
92|
93|     if (existingIndex !== -1) {
94|         cart[existingIndex].qty += 1;
95|     } else {
96|         cart.push({
97|             name: currentProduct.name,
98|             sku: currentProduct.sku,
99|             price: parseFloat(currentProduct.price),
100|             qty: 1
101|         });
102|     }
103|
104|     // Save to localStorage
105|     localStorage.setItem('scanCart', JSON.stringify(cart));
106|
107|     // Show success message
108|     const alertDiv = document.createElement('div');
109|     alertDiv.className = 'alert alert-success alert-dismissible fade show';

```

```

110|         alertDiv.innerHTML = ` 
111|             <strong>Berhasil!</strong> ${currentProduct.name} ditambahkan ke
shopping cart.
112|             <button type="button" class="btn-close" data-bs-
dismiss="alert"></button>
113|         `;
114|         document.querySelector('.container').insertBefore(alertDiv,
document.querySelector('.row'));
115|
116|         // Auto dismiss after 3 seconds
117|         setTimeout(() => {
118|             alertDiv.remove();
119|         }, 3000);
120|     });
121|
122|     // Go to cart page
123|     document.getElementById('goToCartBtn').addEventListener('click', function() {
124|         window.location.href = '/scan-data-produk';
125|     });
126|
127|     // Inisialisasi scanner
128|     let html5QrcodeScanner = new Html5QrcodeScanner("reader", {
129|         fps: 10,
130|         qrbox: 250
131|     });
132|     html5QrcodeScanner.render(onScanSuccess);
133| 
```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>
136|
137| </body>
138| </html>
```

View scan data kode atau tampilan list cart

```

0 | <!DOCTYPE html>
1 | <html lang="en">
2 | <head>
3 |     <meta charset="UTF-8">
4 |     <meta name="viewport" content="width=device-width, initial-scale=1.0">
5 |     <title>Scan Data Produk</title>
6 |     <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
7 | </head>
8 | <body>
9 |     <div class="container mt-4">
10|         <h1 class="text-center mb-4">Scan Data Produk - Shopping Cart</h1>
11|
12|         <!-- Navigation buttons -->
13|         <div class="row mb-3">
14|             <div class="col-md-6">
15|                 <a href="/scankode" class="btn btn-outline-primary">
16|                     <i class="fas fa-camera"></i> Scan dengan Kamera
17|                 </a>
18|             </div>
19|             <div class="col-md-6 text-end">
20|                 <button id="clearCartBtn" class="btn btn-outline-danger">
21|                     <i class="fas fa-trash"></i> Clear Cart
22|                 </button>
23|             </div>
24|         </div>
25|
26|         <div class="row">
27|             <div class="col-md-8">
28|                 <div class="mb-3">
29|                     <label for="barcodeInput" class="form-label">Barcode Scanner
Input</label>
30|                     <input type="text" id="barcodeInput" class="form-control"
placeholder="Scan barcode atau ketik manual (tekan Enter)">
31|                     <input type="text" id="barcodeInput" class="form-control"
placeholder="Scan barcode atau ketik manual (tekan Enter)">
32|                     </div>
33|                 <div class="card">
34|                     <div class="card-body">
```

```

35|             <div class="card-header">
36|                 <h5>Shopping Cart</h5>
37|             </div>
38|             <div class="card-body">
39|                 <div class="table-responsive">
40|                     <table class="table table-striped">
41|                         <thead>
42|                             <tr>
43|                                 <th>No</th>
44|                                 <th>Name</th>
45|                                 <th>SKU</th>
46|                                 <th>Price</th>
47|                                 <th>Qty</th>
48|                                 <th>Subtotal</th>
49|                                 <th>Aksi</th>
50|                             </tr>
51|                         </thead>
52|                         <tbody id="cartTableBody">
53|                             <tr>
54|                                 <td colspan="7" class="text-center text-muted">
55|                                     Cart kosong. Scan produk untuk menambahkan.
56|                                 </td>
57|                             </tr>
58|                         </tbody>
59|                         <tfoot>
60|                             <tr>
61|                                 <td colspan="5" class="text-
end"><strong>Total:</strong></td>
62|                                     <td id="sumTotal" class="fw-bold">Rp 0</td>
63|                                     <td></td>
64|                             </tr>
65|                         </tfoot>
66|                     </table>
67|                 </div>
68|             </div>
69|         </div>
70|
71|         <div class="col-md-4">
72|             <div class="card">
73|                 <div class="card-header">
74|                     <h5>Detail Produk</h5>
75|                 </div>
76|                 <div class="card-body">
77|                     <div id="productDetails">
78|                         <p><strong>Product Code:</strong> <span id="productCode">-
79| </span></p>
80|                         <p><strong>Nama:</strong> <span id="productName">-
81| </span></p>
82|                         <p><strong>SKU:</strong> <span id="productSku">--</span></p>
83|                         <p><strong>Harga:</strong> <span id="productPrice">-
84| <span id="productDescription">--</span></p>
85|                         <p><strong>Deskripsi:</strong> <span
86| <img id="productImage" src="" style="max-width: 100%; height: auto; display: none;" alt="Product Image">
87| </div>
88|                     </div>
89|                 </div>
90|             </div>
91|         </div>
92|     </div>
93|
94|     <script>
95|         let cart = [];
96|
97|         // Load cart from localStorage on page load
98|         function loadCartFromStorage() {
99|             const savedCart = localStorage.getItem('scanCart');
100|             if (savedCart) {
101|                 cart = JSON.parse(savedCart);
102|                 updateCartTable();
103|             }
104|         }

```

```

105| // Save cart to localStorage
106| function saveCartToStorage() {
107|     localStorage.setItem('scanCart', JSON.stringify(cart));
108| }
109|
110|
111| function updateCartTable() {
112|     const tbody = document.getElementById('cartTableBody');
113|     tbody.innerHTML = '';
114|     let total = 0;
115|
116|     if (cart.length === 0) {
117|         tbody.innerHTML = `
118|             <tr>
119|                 <td colspan="7" class="text-center text-muted">
120|                     Cart kosong. Scan produk untuk menambahkan.
121|                 </td>
122|             </tr>
123|         `;
124|     } else {
125|         cart.forEach((item, index) => {
126|             const subtotal = item.price * item.qty;
127|             total += subtotal;
128|
129|             const row = document.createElement('tr');
130|             row.innerHTML = `
131|                 <td>${index + 1}</td>
132|                 <td>${item.name}</td>
133|                 <td>${item.sku}</td>
134|                 <td>Rp ${parseFloat(item.price).toLocaleString('id-ID')}</td>
135|                 <td>
136|                     <input type="number" min="1" value="${item.qty}"
137|                           data-index="${index}" class="form-control qty-
input" style="width: 80px;"/>
138|                 </td>
139|                 <td>Rp ${subtotal.toLocaleString('id-ID')}</td>
140|                 <td>
141|                     <button class="btn btn-sm btn-danger delete-btn" data-
index="${index}">
142|                         Hapus
143|                     </button>
144|                 </td>
145|             `;
146|             tbody.appendChild(row);
147|         });
148|     }
149|
150|     document.getElementById('sumTotal').textContent = `Rp
${total.toLocaleString('id-ID')}`;
151|     saveCartToStorage();
152| }
153|
154| // Event listeners
155| document.addEventListener('input', function(e) {
156|     if (e.target.classList.contains('qty-input')) {
157|         const index = e.target.getAttribute('data-index');
158|         const newQty = parseInt(e.target.value) || 1;
159|         cart[index].qty = newQty;
160|         updateCartTable();
161|     }
162| });
163|
164| document.addEventListener('click', function(e) {
165|     if (e.target.classList.contains('delete-btn')) {
166|         const index = e.target.getAttribute('data-index');
167|         cart.splice(index, 1);
168|         updateCartTable();
169|     }
170| });
171|
172| // Clear cart button
173| document.getElementById('clearCartBtn').addEventListener('click', function() {
174|     if (confirm('Apakah Anda yakin ingin menghapus semua item dari cart?')) {
175|         cart = [];
176|         localStorage.removeItem('scanCart');
177|         updateCartTable();

```

```

178|         }
179|     });
180|
181|     document.getElementById('barcodeInput').addEventListener('keypress',
182|     function(e) {
183|         if (e.key === 'Enter') {
184|             const code = this.value.trim();
185|             if (code === '') return;
186|
187|             fetch('/scan-produk', {
188|                 method: 'POST',
189|                 headers: {
190|                     'Content-Type': 'application/json',
191|                     'Accept': 'application/json',
192|                     'X-CSRF-TOKEN': '{{ csrf_token() }}'
193|                 },
194|                 body: JSON.stringify({ code })
195|             })
196|             .then(res => res.json())
197|             .then(data => {
198|                 if (data.success) {
199|                     const p = data.product;
200|
201|                     // Check if product already exists in cart
202|                     const existingIndex = cart.findIndex(item => item.sku ===
203|                         p.sku);
204|
205|                     if (existingIndex !== -1) {
206|                         cart[existingIndex].qty += 1;
207|                     } else {
208|                         cart.push({
209|                             name: p.name,
210|                             sku: p.sku,
211|                             price: parseFloat(p.price),
212|                             qty: 1
213|                         });
214|
215|                     updateCartTable();
216|
217|                     // Display product details
218|                     document.getElementById('productCode').textContent =
219|                         data.code;
220|
221|                     document.getElementById('productName').textContent = p.name;
222|                     document.getElementById('productSku').textContent = p.sku;
223|                     document.getElementById('productPrice').textContent =
224|                         'Rp ' + parseFloat(p.price).toLocaleString('id-ID');
225|                     document.getElementById('productDescription').textContent =
226|                         p.description;
227|
228|                     const imgElement = document.getElementById('productImage');
229|                     if (p.image) {
230|                         imgElement.src = p.image;
231|                         imgElement.style.display = 'block';
232|                     } else {
233|                         imgElement.style.display = 'none';
234|                     }
235|
236|                     this.value = '';
237|                 } else {
238|                     alert(data.message);
239|                     this.value = '';
240|                 }
241|             })
242|             .catch(err => {
243|                 console.error('Error:', err);
244|                 alert('Terjadi kesalahan saat memproses scan');
245|             });
246|
247|         }
248|     };
249| </script>

```

```
250| </body>
251| </html>
```

Migrasi products

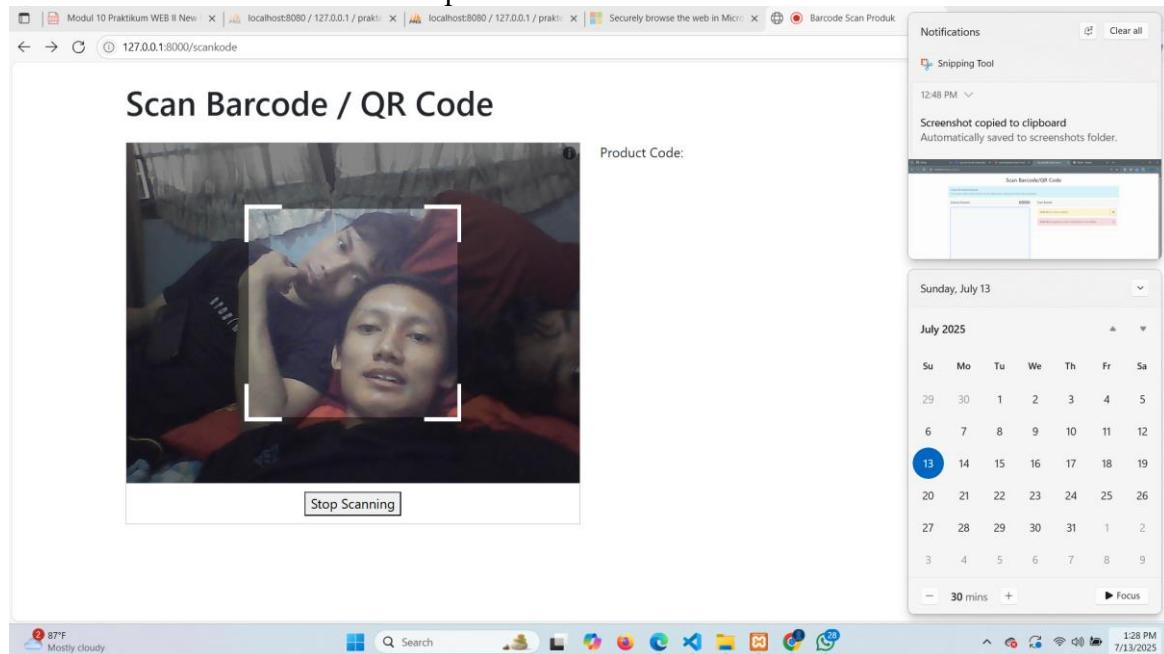
```
0 | <?php
1 |
2 | use Illuminate\Database\Migrations\Migration;
3 | use Illuminate\Database\Schema\Blueprint;
4 | use Illuminate\Support\Facades\Schema;
5 |
6 | return new class extends Migration
7 | {
8 |     public function up(): void
9 |     {
10|         Schema::create('products', function (Blueprint $table) {
11|             $table->id();
12|             $table->string('name');
13|             $table->string('sku')->unique();
14|             $table->decimal('price', 10, 2);
15|             $table->text('description')->nullable();
16|             $table->string('image')->nullable();
17|             $table->timestamps();
18|         });
19|     }
20|
21|     public function down(): void
22|     {
23|         Schema::dropIfExists('products');
24|     }
25| };
```

Model products

```
0 | <?php
1 |
2 | namespace App\Models;
3 |
4 | use Illuminate\Database\Eloquent\Model;
5 |
6 | class Product extends Model
7 | {
8 |     protected $fillable = [
9 |         'name',
10|         'sku',
11|         'price',
12|         'description',
13|         'image'
14|     ];
15|
16|     protected $casts = [
17|         'price' => 'decimal:2',
18|     ];
19|
20|     public function getImageUrlAttribute()
21|     {
22|         return $this->image ? asset('storage/' . $this->image) : null;
23|     }
24| }
```

Screen Shoot / Output Hasil Kode Program:

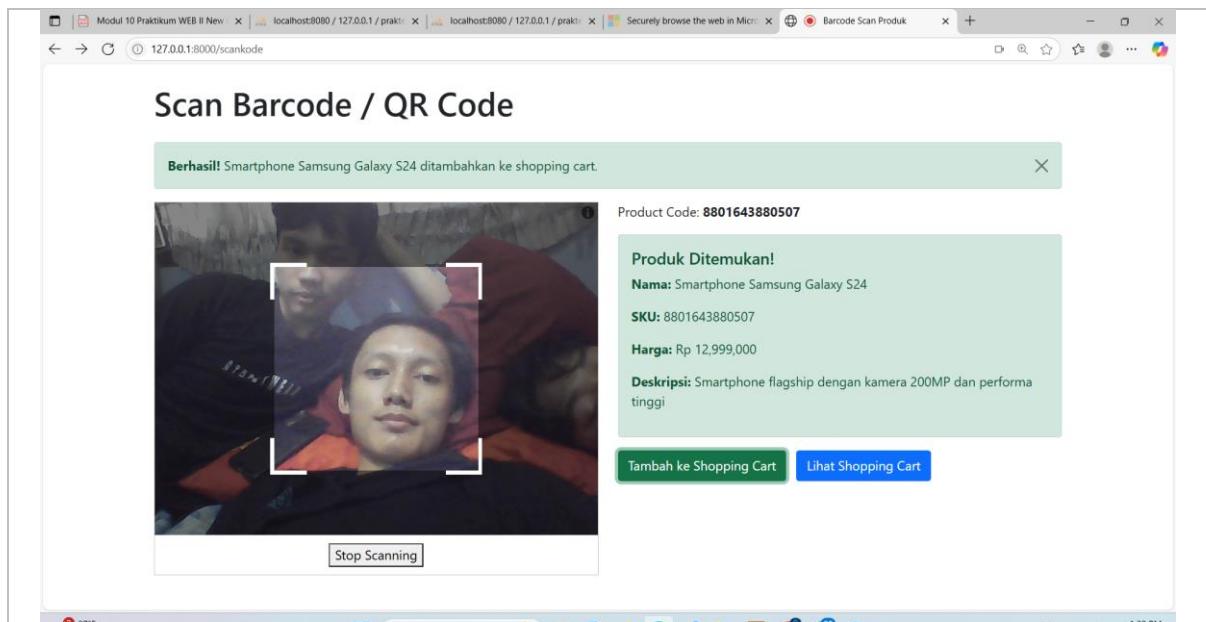
Tampilan awal untuk scan kode



Tampilan proses scan kode data

This screenshot shows the process of scanning a barcode. The camera viewfinder is focused on a smartphone displaying a barcode. The phone's screen shows the barcode and the number '8801643880507'. To the right of the viewfinder, a green box displays the scanned product information: 'Produk Ditemukan!', 'Nama: Smartphone Samsung Galaxy S24', 'SKU: 8801643880507', 'Harga: Rp 12,999,000', and 'Deskripsi: Smartphone flagship dengan kamera 200MP dan performa tinggi'. Below this box are two buttons: 'Tambah ke Shopping Cart' (green) and 'Lihat Shopping Cart' (blue).

Tampilan proses tambah ke cart dengan alert



87°F Mostly cloudy 1:32 PM 7/13/2025

Tampilan isi dari cart

The screenshot shows a web browser window titled "Scan Data Produk - Shopping Cart". It features a "Scan dengan Kamera" button and a "Barcode Scanner Input" field. The main area displays a "Shopping Cart" table:

No	Name	SKU	Price	Qty	Subtotal	Aksi
1	Smartphone Samsung Galaxy S24	8801643880507	Rp 12,999,000	1	Rp 12,999,000	Hapus

Total: Rp 12,999,000

To the right, there is a "Detail Produk" panel with fields for Product Code, Name, SKU, Harga, and Deskripsi, all currently empty. A "Clear Cart" button is also present.

87°F Mostly cloudy 1:32 PM 7/13/2025

Tampilan tambah data produk manual dan produk bertambah

The screenshot shows a web-based shopping cart application. At the top, there's a header "Scan Data Produk - Shopping Cart". On the left, there's a button "Scan dengan Kamera" and a "Barcode Scanner Input" field with placeholder text "Scan barcode atau ketik manual (tekan Enter)". Below this is a "Shopping Cart" section with a table:

No	Name	SKU	Price	Qty	Subtotal	Aksi
1	Smartphone Samsung Galaxy S24	8801643880507	Rp 12.999.000	2	Rp 25.998.000	<button>Hapus</button>

At the bottom of the cart table, it says "Total: Rp 25.998.000". To the right, there's a "Detail Produk" panel with the following information:

Product Code: 8801643880507
Nama: Smartphone Samsung Galaxy S24
SKU: 8801643880507
Harga: Rp 12.999.000
Deskripsi: Smartphone flagship dengan kamera 200MP dan performa tinggi

Kesimpulan Hasil / Solusi:

Terdapat berbagai website penyedia generator barcode dan QR code online seperti Barcode.tec-it.com, QRCode Monkey, dan BarcodeAPI.org yang menawarkan layanan gratis dengan beragam fitur, mulai dari pembuatan barcode sederhana hingga QR code kustom dengan logo dan warna, cocok untuk kebutuhan cepat tanpa instalasi software, meskipun untuk produksi massal atau integrasi sistem, solusi berbasis library seperti milon/barcode di Laravel lebih efisien dan terintegrasi langsung dengan database produk.

SWEETALERT DAN DATA TABLES

Task: Modul 10

1. Kerjakan instruksi pada kegiatan praktikum modul 10 ini dan dokumentasikan hasilnya pada laporan praktikum.
2. Buat custom alert untuk menampilkan pesan sukses setelah data berhasil ditambahkan atau diperbarui dan dokumentasikan hasilnya pada laporan praktikum.
3. -Tambahkan DataTables pada tabel produk buat aplikasi CRUD sederhana untuk tabel produk.
 - Konfigurasikan DataTables untuk menggunakan server-side processing dengan Ajax.
 - dan dokumentasikan hasilnya pada laporan praktikum.

Kode Program

Github

https://github.com/dandi-maulana/web_application_practicum.git

KodeSoalNo1
UserController.php

```
0 | <?php
1 |
2 | namespace App\Http\Controllers;
3 |
4 | use App\Models\User;
5 | use Illuminate\Http\Request;
6 | use Illuminate\Support\Facades\Hash;
7 |
8 | class UserController extends Controller
9 |
10| {
11|     public function index()
12|     {
13|         $users = User::all();
14|         return view('users.index', compact('users'));
15|     }
16|
17|     public function create()
18|     {
19|         return view('users.create');
20|     }
21|
22|     public function store(Request $request)
23|     {
24|         $request->validate([
25|             'name'      => 'required|max:255',
26|             'email'     => 'required|email|unique:users',
27|             'level'     => 'required',
28|             'password'  => 'required|min:8',
29|         ]);
30|
31|         User::create([
32|             'name'      => $request->name,
33|             'email'     => $request->email,
34|             'level'     => $request->level,
35|             'password'  => Hash::make($request->password),
36|         ]);
37|
38|         return redirect()->route('users.index')->with('success', 'User created
successfully.');
39|
40|     public function show(User $user)
41|     {
42|         return view('users.show', compact('user'));
43|     }
44|
45|     public function edit(User $user)
46|     {
```

```

47|         return view('users.edit', compact('user'));
48|
49|
50|     public function update(Request $request, User $user)
51|     {
52|         $request->validate([
53|             'name'      => 'required|max:255',
54|             'email'     => 'required|email|unique:users,email,' . $user->id,
55|             'level'     => 'required',
56|             'password'  => 'nullable|min:8',
57|         ]);
58|
59|         $user->update([
60|             'name'      => $request->name,
61|             'email'     => $request->email,
62|             'level'     => $request->level,
63|             'password'  => $request->password ? Hash::make($request->password) : $user-
>password,
64|         ]);
65|
66|         return redirect()->route('users.index')->with('success', 'User updated
successfully.');
67|     }
68|
69|     public function destroy(User $user)
70|     {
71|         $user->delete();
72|         return redirect()->route('users.index')->with('success', 'User deleted
successfully.');
73|     }
74| }
75|

```

Web.php

```

0 | use App\Http\Controllers\UserController;
1 | Route::resource('users', UserController::class);

```

App.blade.php

```

0 | <!DOCTYPE html>
1 | <html lang="en">
2 | <head>
3 |     <meta charset="utf-8">
4 |     <link rel="stylesheet"
5 |         href="https://cdn.datatables.net/1.11.3/css/jquery.dataTables.min.css">
6 |     <meta name="csrf-token" content="{{ csrf_token() }}">
7 | </head>
8 | <body>
9 |     @yield('content')
10|     <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
11|     <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
12|     <script
src="https://cdn.datatables.net/1.11.3/js/jquery.dataTables.min.js"></script>
13|     <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
14|
15|     <script>
16|         axios.defaults.headers.common['X-CSRF-TOKEN'] =
17|             document.querySelector('meta[name="csrf-token"]').content;
18|     </script>
19|
20|     @stack('scripts')
21| </body>
22| </html>
23|

```

Index.blade.php

```

0 | @extends('layouts.app')
1 |
2 | @section('content')
3 | <div class="container">

```

```

4 |     <h1>Users</h1>
5 |
6 |     {{-- flash alert sukses (setelah store / update / delete) --}}
7 |     @if (session('success'))
8 |         <div id="flash-success" data-msg="{{ session('success') }}"></div>
9 |     @endif
10|
11|     <table class="table" id="users-table">
12|         <thead>
13|             <tr>
14|                 <th>Name</th>
15|                 <th>Email</th>
16|                 <th>Actions</th>
17|             </tr>
18|         </thead>
19|         <tbody>
20|             @foreach ($users as $user)
21|             <tr>
22|                 <td>{{ $user->name }}</td>
23|                 <td>{{ $user->email }}</td>
24|                 <td>
25|                     <button class="btn btn-danger btn-sm"
26|                         onclick="confirmDelete('{{ $user->id }}")>
27|                         Delete
28|                     </button>
29|                 </td>
30|             </tr>
31|         @endforeach
32|     </tbody>
33| </table>
34| </div>
35| @endsection
36|
37| @push('scripts')
38| <script>
39| $(function () {
40|     // aktifkan DataTables
41|     $('#users-table').DataTable();
42|
43|     // jika ada flash sukses, tampilkan SweetAlert
44|     const flash = $('#flash-success').data('msg');
45|     if (flash) swal("Berhasil", flash, "success");
46| });
47|
48| /**
49| * Konfirmasi hapus + kirim DELETE via Axios
50| */
51| function confirmDelete(id) {
52|     swal({
53|         title: "Yakin ingin menghapus?",
54|         text: "Data user akan hilang permanen!",
55|         icon: "warning",
56|         buttons: ["Batal", "Hapus!"],
57|         dangerMode: true,
58|     })
59|     .then((willDelete) => {
60|         if (willDelete) {
61|             axios.delete(`/users/${id}`)
62|                 .then(() => {
63|                     swal("Terhapus!", "User berhasil dihapus.", "success")
64|                         .then(() => location.reload());
65|                 })
66|                 .catch(() => {
67|                     swal("Gagal", "Terjadi kesalahan.", "error");
68|                 });
69|         }
70|     });
71| }
72| </script>
73| @endpush
74|

```

KodeSoalNo2.
Create.blade.php

```
0 | @extends('layouts.app')
```

```

1 | @section('content')
2 | <div class="container" style="max-width:600px">
3 |     <h2>Tambah User</h2>
4 |
5 |     {{-- tampilkan validasi --}}
6 |     @if ($errors->any())
7 |         <div class="alert alert-danger">
8 |             <ul class="mb-0">
9 |                 @foreach ($errors->all() as $e) <li>{{ $e }}</li>@endforeach
10 |             </ul>
11 |         </div>
12 |     @endif
13 |
14 |
15 |     <form action="{{ route('users.store') }}" method="POST">
16 |         @csrf
17 |         <div class="mb-3">
18 |             <label class="form-label">Nama</label>
19 |             <input type="text" name="name" class="form-control"
20 |                 value="{{ old('name') }}" required>
21 |         </div>
22 |
23 |         <div class="mb-3">
24 |             <label class="form-label">Email</label>
25 |             <input type="email" name="email" class="form-control"
26 |                 value="{{ old('email') }}" required>
27 |         </div>
28 |
29 |         <div class="mb-3">
30 |             <label class="form-label">Level</label>
31 |             <select name="level" class="form-select" required>
32 |                 <option value="" disabled selected>-- pilih level --</option>
33 |                 <option value="admin">Admin</option>
34 |                 <option value="user">User</option>
35 |             </select>
36 |         </div>
37 |
38 |         <div class="mb-3">
39 |             <label class="form-label">Password</label>
40 |             <input type="password" name="password" class="form-control" required>
41 |         </div>
42 |
43 |         <button class="btn btn-success">Simpan</button>
44 |         <a href="{{ route('users.index') }}" class="btn btn-secondary">Batal</a>
45 |     </form>
46 | </div>
47 | @endsection
48 |

```

Index.blade.php

```

0 |     <a href="{{ route('users.create') }}" class="btn btn-primary mb-3">
1 |         + Tambah User
2 |     </a>

```

Screen Shoot / Output Hasil Kode Program:

Output Soal nomor 1

Screenshot of a web browser showing a users management interface. The URL is `localhost:8000/users`.

Users

Show 10 entries

Name	Email	Actions
German Feil	warren.will@example.net	Delete
Lois Maggio	yadira.nader@example.com	Delete
Mortimer Larson DVM	fbernier@example.com	Delete
Wayne Corwin	kovacek.carol@example.com	Delete
Willow Schoen	ankunding.elmira@example.org	Delete

Showing 1 to 5 of 5 entries

Search:

Previous 1 Next

Screenshot of the same web browser showing a confirmation dialog box.



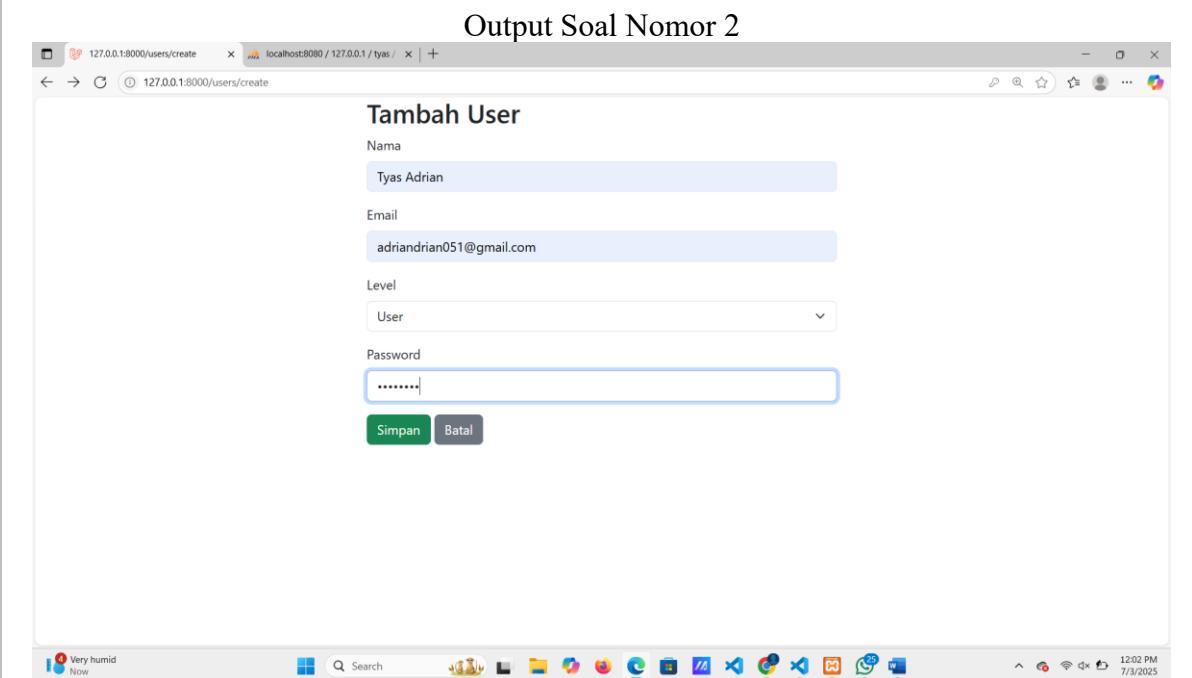
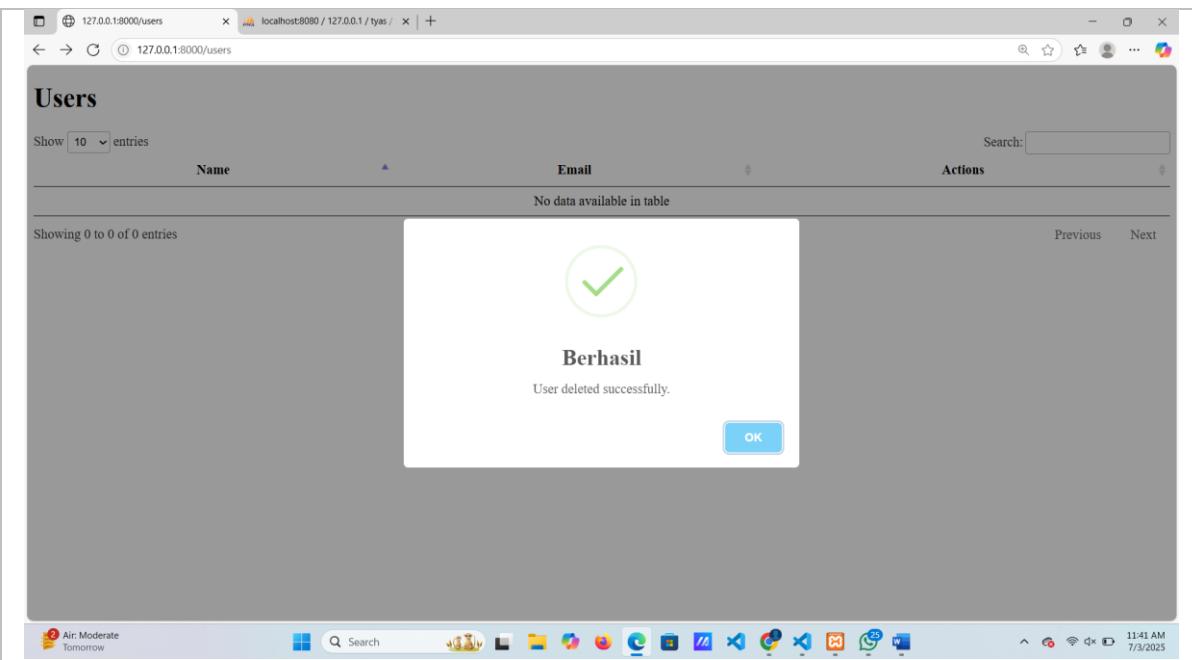
Yakin ingin menghapus?

Data user akan hilang permanen!

Showing 1 to 5 of 5 entries

Search:

Previous 1 Next



The image displays two screenshots of a web-based user management application. The top screenshot shows a modal dialog box centered over a table of users. The modal contains a green checkmark icon, the word "Berhasil", and the message "User created successfully." with an "OK" button. The background table lists one user: Tyas Adrian with email adriandrian051@gmail.com. The bottom screenshot shows the same table after the new user has been added, with the same data visible.

Name	Email
Tyas Adrian	adriandrian051@gmail.com

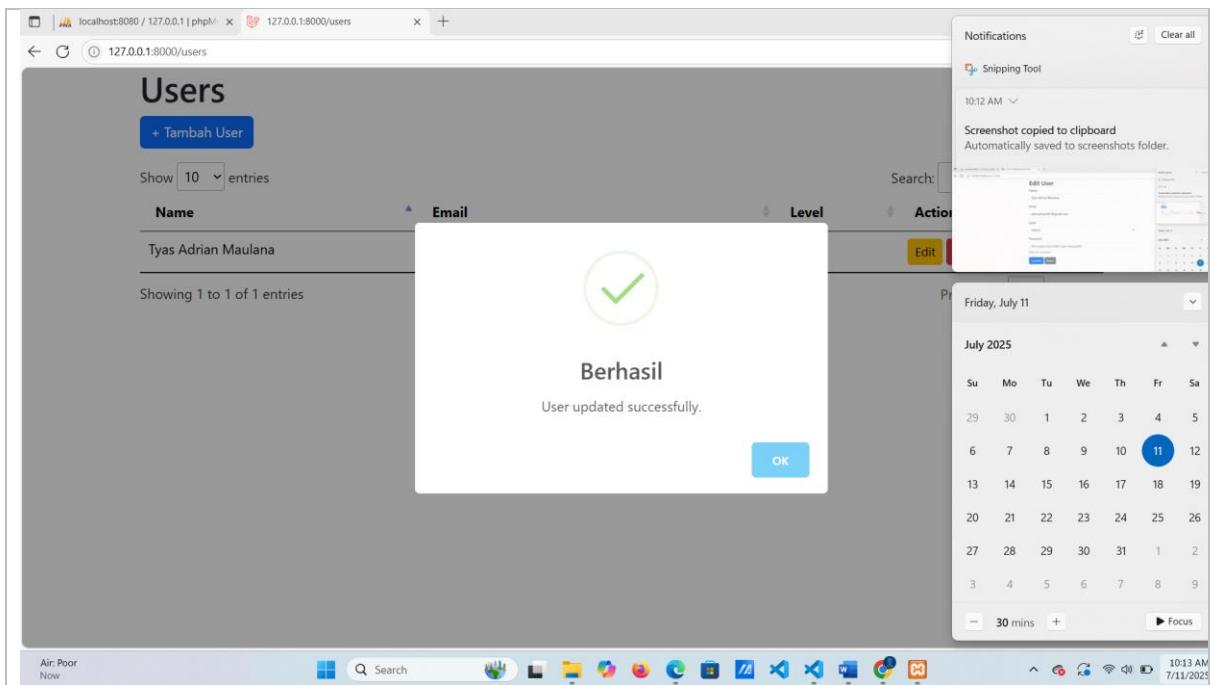
Untuk CRUD ini tambahan update atau edit users

The screenshot shows a web application interface for managing users. At the top, there is a header bar with tabs for 'localhost:8080 / 127.0.0.1 | phpMyAdmin' and '127.0.0.1:8000/users'. Below the header, the main content area has a title 'Users' and a button '+ Tambah User'. A table lists one user entry:

Name	Email	Level	Actions
Tyas Adrian	adriandrian051@gmail.com		Edit Delete

Below the table, a message says 'Showing 1 to 1 of 1 entries'. To the right of the table, there is a 'Notifications' sidebar with a 'Storage settings' notification about low storage space. At the bottom of the screen, there is a taskbar with various icons and a system tray showing the date and time as 'Friday, July 11'.

In the second part of the screenshot, a new window titled 'Edit User' is open. It contains fields for 'Nama' (Tyas Adrian Maulana), 'Email' (adriandrian051@gmail.com), 'Level' (Admin), and a 'Password' field with placeholder text 'Kosongkan jika tidak ingin mengubah'. There are 'Update' and 'Batal' buttons at the bottom. The background of this window shows a preview of the user list table from the first screenshot. The taskbar and system tray are also visible at the bottom.



Kesimpulan Hasil / Solusi:

Program-program tersebut membangun sistem manajemen pengguna (CRUD) dengan Laravel yang mencakup tampilan daftar pengguna (index), form tambah pengguna (create), dan form edit pengguna (edit), dilengkapi validasi input, fitur hapus dengan konfirmasi, serta penggunaan DataTables untuk tampilan tabel yang interaktif, dengan dukungan autentifikasi level pengguna (admin/user) dan manajemen password yang aman.

LEMBAR PERNYATAAN

Kita yang tangan tanda di bawah ini, menyatakan dan melaporkan hasil praktikum Matakuliah Pemrograman Web II ini adalah murni hasil karya kami tanpa menciplak / copy paste laporan lain, di kerjakan dengan penuh tanggung jawab dan bersungguh-sungguh dan saling berkontribusi sebagai anggota kelompok di dalam penggerjaan, penyusunan hingga penyelesaian laporan ini.

Yang Menyusun Laporan

Nugraha Rahmadan Diyanto
228160049

Tyas Adrian
228160073